Modulated Mathematical Formulation of MSE for Assessment of a Positioning System in Live-status

Abbas Al baidhani
Member IEEE, Dr, Technical Department, Ministry of transport, Iraq
Email: abbasabbar.ghali@e-campus.uab.cat

Abstract. In wireless sensor network (WSN), it is a substantial affair to obtain a convenient assessment metric for evaluating created positioning systems. Mean square error (MSE) is one of very paramount assessment metrics utilized for such problem. The conventional mathematical formula of MSE such as the built in MSE function in matlab is utilized only when the real and measured values of the position are available, which means utilizing the MSE is potential only after building the system not while building it. In this paper, for positioning purpose, we offer a mathematical derivation of an MSE formulation based on least square (LS) approach for three-dimensional positioning system without utilizing the real and measured position. The created MSE uses only the measured distance between anchor nodes and a mobile station, that needs its position to be calculated. The formed MSE has the ability to assess a positioning system online based only on the estimated distances, and then the user can modulate his system until obtaining a system ambidextrous to match the requested positioning accuracy. The derived MSE has accuracy 100% equals to the conventional mathematical formulation such as the built in matlab MSE function.

1. Introduction
Recently, wireless information access is widely available, there is an increment demand for accurate positioning algorithms for both outdoor and indoor environments [1]. Positioning approach in wireless sensor network is classified into two types, there are: range based and range free. Range based is used the range based on the distance or angle as information from anchor node to estimate the location of an object which has been installed with sensor [2]. The possibility of gathering localization information with wireless LAN applications helps the improvement of available mobile applications and the proposal of several innovative services [3]. If a commander has the ability to monitor the wireless users coordinates and feed location data to different application software types, different possibility applications might be easily envisaged. Usually, researchers evaluate the positioning accuracy of their created systems off-line using different evaluation metrics such as mean square error (MSE), geometric dilution of positioning (GDOP), etc. But, if we would like to enhance the positioning accuracy of a created system online, these mentioned metrics are useless. For example, the conventional MSE such as built in MSE function has concentrates which needs for the true and estimated position of a target, so it is not able to be implemented for online evaluation of the positioning accuracy of the created system when this system is utilized in real life. In real life, the main idea from enhancing the positioning accuracy online is to make the created positioning system more flexible for different environment [4]. As we solved the aforementioned problem in our previous work [5], but the solution was limited to two dimensions’ system, the coordinates of the reference anchor node (x1; y1) should be zeros, the positioning system should be implemented first to obtain the estimated position, and the MSE equation
is in a matrix format. In this paper, the contribution is to derive an MSE based on LS method for three dimensions mathematically with no previous information of the true and estimated position of the target. In the work, the created MSE is computed in a generic formula for three dimensions’ system using only estimated distances between anchor nodes and target. The obtained accuracy of the created MSE equals 100% to the conventional MSE such as the built-in MSE matlab function which is two times better than the previous MSE created by [4], also, we avoid using matrix operations when computing the MSE to make it simpler and cost less time than the previous MSE. Section 2 presents the mathematical module of MSE and the modified LS. Result and discussion are presented in section 3. Finely, section 4 offers a conclusion.

2. Mathematical Module

Target positioning in space (3 dimensions), we need at least four anchor (fixed) nodes with known coordinates \((x; y; z)\) connected to a target having unknown coordinates \((Tx; Ty; Tz)\) using a radio frequency technology to measure the distances between every anchor node and the target. The distance between an anchor node such as the first anchor node and a target is computed as presented in equation 1.

\[
\begin{align*}
    r_1 &= (T_x - x_1)^2 + (T_y - y_1)^2 + (T_z - z_1)^2
\end{align*}
\]

(1)

Where, \(r\) denotes the real distance. The generic equation of the distance is shown in equation 2.

\[
\begin{align*}
    r_i &= (T_x - x_i)^2 + (T_y - y_i)^2 + (T_z - z_i)^2
\end{align*}
\]

(2)

To linearize the non-linearity of the LS method, we need to determine one of the aforementioned equations as a reference equation and subtract it from others, then implementing matrix operations. So, in a generic way, we obtain equation 3.

\[
\begin{align*}
    v_i &= \frac{(x_i - x_1)^2 + (y_i - y_1)^2 + (z_i - z_1)^2 + r_1^2 - r_i^2}{2} \rightarrow v = \begin{bmatrix} v_2 \\ v_3 \\ \vdots \\ v_n \end{bmatrix}
\end{align*}
\]

(3)

Where, \(i\) denotes the index of an anchor node and \(i = 2 \text{ to } n\), where \(n\) is the total number of the anchor nodes. The coordinates matrix of the anchor nodes \((H)\) is presented below.

\[
H = \begin{bmatrix}
    x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\
    x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\
    \vdots & \vdots & \vdots \\
    x_n - x_1 & y_n - y_1 & z_n - z_1
\end{bmatrix}
\]

In real experiment, the estimated distance by a wireless sensor has a residual error due to the attenuation occurred on the transmitted signal because of different parameters. So, we express the estimated distance as shown in equation 4.

\[
\begin{align*}
    d_i &= r_i + e_i
\end{align*}
\]

(4)

Where, \(e\) denotes the residual error. As we computed the generic equation of the real distances, we compute it to the estimated distances denoted \(b\) as shown in equation 5.

\[
\begin{align*}
    b_i &= \frac{(x_i - x_1)^2 + (y_i - y_1)^2 + (z_i - z_1)^2 + d_1^2 - d_i^2}{2} \rightarrow b = \begin{bmatrix} b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}
\end{align*}
\]

(5)

The compact equation of the estimated target position is expressed in equation 6.
\[
\hat{T} = \begin{bmatrix}
\hat{T}_x \\
\hat{T}_y \\
\hat{T}_z
\end{bmatrix} = (H^TH)^{-1}H^Tb + \begin{bmatrix}
x_1 \\
y_1 \\
z_1
\end{bmatrix}
\] (6)

Now, let us compute \((H^TH)^{-1}\) matrix separately as shown in equation 7. The invers matrix of the three dimensions system is \(3 \times 3\) matrix whatever the total number of the anchor nodes is.

\[
(H^TH)^{-1} = \frac{1}{\text{det}} \text{adj}(H^TH)
\] (7)

Where, \(\text{det}\) denotes the determinant of \(H^TH\) and \(\text{adj}\) denotes the adjugate matrix of \(H^TH\). Let us express the \(\text{adj}\) as shown below.

\[
\text{adj}(H^TH) = \begin{bmatrix}
k_{11} & -k_{12} & k_{13} \\
-k_{12} & k_{22} & -k_{23} \\
k_{13} & -k_{23} & k_{33}
\end{bmatrix}
\] (8)

where, \(k\) is explained in appendix A below.

2.1 MSE Module
In this subsection, we present the compact equations of the derived MSE and leave the entire derivations including \(\text{det}\) and \(\text{adj}\) to be in appendix 1. MSE for \(x\) coordinates is presented in equation 9.

\[
\text{MSE}(\hat{T}_x) = \frac{\sum_{i=1}^{n} E(e_i)}{\text{det}} \left[ (k_{11} \sum_{i=1}^{n} x_i - x_i) - (k_{12} \sum_{i=1}^{n} y_i - y_i) + (k_{13} \sum_{i=1}^{n} z_i - z_i) \right]^2
\] (9)

Also, the MSEs for \(y\) and \(z\) coordinates, Eq.10 and Eq. 11, are similar to \(x\) coordinate and only will have change with the index of \(k\) in equation 8.

\[
\text{MSE}(\hat{T}_y) = \frac{\sum_{i=1}^{n} E(e_i)}{\text{det}} \left[ (-k_{12} \sum_{i=1}^{n} x_i - x_i) + (k_{22} \sum_{i=1}^{n} y_i - y_i) - (k_{23} \sum_{i=1}^{n} z_i - z_i) \right]^2
\] (10)

\[
\text{MSE}(\hat{T}_z) = \frac{\sum_{i=1}^{n} E(e_i)}{\text{det}} \left[ (k_{13} \sum_{i=1}^{n} x_i - x_i) - (k_{23} \sum_{i=1}^{n} y_i - y_i) + (k_{33} \sum_{i=1}^{n} z_i - z_i) \right]^2
\] (11)

Where, \(e_i = e_i^2 - \delta_i^2 + 2d_i e_i - 2d_i \delta_i\) and it is computed in Appendix 1 subsection B for every coordinates. Where, \(i = 2 \ldots n\) denotes the index of the anchor node, and \(n\) denotes the total number of anchor nodes. Also, \(E\) denotes the expectation value.

Finely, the compact equation of the derived MSE is expressed in equation 12.

\[
\text{MSE}(\hat{T}) = \frac{\text{MSE}(\hat{T}_x) + \text{MSE}(\hat{T}_y) + \text{MSE}(\hat{T}_z)}{3}
\] (12)

2.2 Localization Module
In this subsection, we present the created localization module to compute the estimated coordinates of the target without using matrix operations. Also, we present in this subsection only the compact equations of the coordinates and leave the entire mathematical derivation to appendix 1. Based on equation 6, we rewrite this equation as presented in equation 13.

\[
\begin{bmatrix}
\hat{T}_x \\
\hat{T}_y \\
\hat{T}_z
\end{bmatrix} = \left(\frac{1}{\text{det}} \text{adj}(H^TH)\right)H^Tb + \begin{bmatrix}
x_1 \\
y_1 \\
z_1
\end{bmatrix}
\] (13)

Then, the estimated coordinates of the target are shown in equations 14, 15, and 16.
\[ \hat{x}_1 = x_1 + \sum_{i=2}^{n} \frac{b_i}{\det} \left[ \left( k_{11} \sum_{i=2}^{n} x_i - x_1 \right) - \left( k_{12} \sum_{i=2}^{n} x_i - x_1 \right) + \left( k_{13} \sum_{i=2}^{n} x_i - x_1 \right) \right] \tag{14} \]

\[ \hat{y}_1 = y_1 + \sum_{i=2}^{n} \frac{b_i}{\det} \left[ \left( -k_{21} \sum_{i=2}^{n} x_i - x_1 \right) + \left( k_{22} \sum_{i=2}^{n} x_i - x_1 \right) - \left( k_{23} \sum_{i=2}^{n} x_i - x_1 \right) \right] \tag{15} \]

\[ \hat{y}_1 = z_1 + \sum_{i=2}^{n} \frac{b_i}{\det} \left[ \left( k_{13} \sum_{i=2}^{n} x_i - x_1 \right) - \left( k_{23} \sum_{i=2}^{n} x_i - x_1 \right) + \left( k_{33} \sum_{i=2}^{n} x_i - x_1 \right) \right] \tag{16} \]

3. Result and Discussion

The created MSE is mathematically proofed with step-by-step calculation, so we think no need to assess it by other way. But we also assessed it using matlab simulation. More than one thousand points are evaluated using the created MSE for real and estimated distances, then it is compared to built-in matlab function. We compute the relative error (\( R \)) between both MSEs for real and estimated distances as shown in equation 17 below.

\[ R = \frac{\text{MSE}_{\text{derived}} - \text{MSE}_{\text{matlab}}}{\text{MSE}_{\text{matlab}}} \tag{17} \]

We observed that when adding0 random error (real distances), which means the result of the estimated values are exactly equal to the true values, the derived MSE equals to zero, and for MSE matlab function is slightly higher than zero (approximately around \(10^{-22}\)). Also, we add random error to the computed distances and compute the relative error between the two MSEs, the relative error is around \(10^{-14}\).

4. Conclusion

We present a novel mathematical formula of MSE could be utilized in live situations based on LS positioning method without involving the true and real values of a target position, but only the measured distances. The purpose of this MSE to make an evaluation to a positioning system live in status, accurate, and flexible that a user can use it to modify his positioning system until reaching the required accuracy, for example add or remove some anchor nodes, or enhance the measured distances. The created MSE is exactly equal to the built in MSE matlab function. It is valuable and could be used for outdoor and indoor positioning applications.

Acknowledgments

The authors would like to acknowledge and thank Iraqi government for funding and supporting this work.

Appendices

In the appendices A and B below, we offer the entire derivation of the created localization algorithm and the MSE module.

A. localization Algorithm

As we mentioned in the mathematical module section, equation 7 as stated below in equation A.1.

\[ (H^TH)^{-1} = \frac{1}{\det} \text{adj}(H^TH) \tag{A.1} \]

And, \( \text{adj}(H^TH) = \begin{bmatrix} k_{11} & -k_{12} & k_{13} \\ -k_{21} & k_{22} & -k_{23} \\ k_{31} & -k_{32} & k_{33} \end{bmatrix} \)

So, we start with computing \( k \)

From Matrix \( H = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ \vdots & \vdots & \vdots \\ x_n - x_1 & y_n - y_1 & z_n - z_1 \end{bmatrix} \), we can compute \( H^TH \)

\[ 3. Result and Discussion \]

The created MSE is mathematically proofed with step-by-step calculation, so we think no need to assess it by other way. But we also assessed it using matlab simulation. More than one thousand points are evaluated using the created MSE for real and estimated distances, then it is compared to built-in matlab function. We compute the relative error (\( R \)) between both MSEs for real and estimated distances as shown in equation 17 below.

\[ R = \frac{\text{MSE}_{\text{derived}} - \text{MSE}_{\text{matlab}}}{\text{MSE}_{\text{matlab}}} \tag{17} \]

We observed that when adding0 random error (real distances), which means the result of the estimated values are exactly equal to the true values, the derived MSE equals to zero, and for MSE matlab function is slightly higher than zero (approximately around \(10^{-22}\)). Also, we add random error to the computed distances and compute the relative error between the two MSEs, the relative error is around \(10^{-14}\).

4. Conclusion

We present a novel mathematical formula of MSE could be utilized in live situations based on LS positioning method without involving the true and real values of a target position, but only the measured distances. The purpose of this MSE to make an evaluation to a positioning system live in status, accurate, and flexible that a user can use it to modify his positioning system until reaching the required accuracy, for example add or remove some anchor nodes, or enhance the measured distances. The created MSE is exactly equal to the built in MSE matlab function. It is valuable and could be used for outdoor and indoor positioning applications.

Acknowledgments

The authors would like to acknowledge and thank Iraqi government for funding and supporting this work.

Appendices

In the appendices A and B below, we offer the entire derivation of the created localization algorithm and the MSE module.

A. localization Algorithm

As we mentioned in the mathematical module section, equation 7 as stated below in equation A.1.

\[ (H^TH)^{-1} = \frac{1}{\det} \text{adj}(H^TH) \tag{A.1} \]

And, \( \text{adj}(H^TH) = \begin{bmatrix} k_{11} & -k_{12} & k_{13} \\ -k_{21} & k_{22} & -k_{23} \\ k_{31} & -k_{32} & k_{33} \end{bmatrix} \)

So, we start with computing \( k \)

From Matrix \( H = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ \vdots & \vdots & \vdots \\ x_n - x_1 & y_n - y_1 & z_n - z_1 \end{bmatrix} \), we can compute \( H^TH \)

\[ 3. Result and Discussion \]

The created MSE is mathematically proofed with step-by-step calculation, so we think no need to assess it by other way. But we also assessed it using matlab simulation. More than one thousand points are evaluated using the created MSE for real and estimated distances, then it is compared to built-in matlab function. We compute the relative error (\( R \)) between both MSEs for real and estimated distances as shown in equation 17 below.

\[ R = \frac{\text{MSE}_{\text{derived}} - \text{MSE}_{\text{matlab}}}{\text{MSE}_{\text{matlab}}} \tag{17} \]

We observed that when adding0 random error (real distances), which means the result of the estimated values are exactly equal to the true values, the derived MSE equals to zero, and for MSE matlab function is slightly higher than zero (approximately around \(10^{-22}\)). Also, we add random error to the computed distances and compute the relative error between the two MSEs, the relative error is around \(10^{-14}\).

4. Conclusion

We present a novel mathematical formula of MSE could be utilized in live situations based on LS positioning method without involving the true and real values of a target position, but only the measured distances. The purpose of this MSE to make an evaluation to a positioning system live in status, accurate, and flexible that a user can use it to modify his positioning system until reaching the required accuracy, for example add or remove some anchor nodes, or enhance the measured distances. The created MSE is exactly equal to the built in MSE matlab function. It is valuable and could be used for outdoor and indoor positioning applications.

Acknowledgments

The authors would like to acknowledge and thank Iraqi government for funding and supporting this work.

Appendices

In the appendices A and B below, we offer the entire derivation of the created localization algorithm and the MSE module.

A. localization Algorithm

As we mentioned in the mathematical module section, equation 7 as stated below in equation A.1.

\[ (H^TH)^{-1} = \frac{1}{\det} \text{adj}(H^TH) \tag{A.1} \]

And, \( \text{adj}(H^TH) = \begin{bmatrix} k_{11} & -k_{12} & k_{13} \\ -k_{21} & k_{22} & -k_{23} \\ k_{31} & -k_{32} & k_{33} \end{bmatrix} \)

So, we start with computing \( k \)

From Matrix \( H = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ \vdots & \vdots & \vdots \\ x_n - x_1 & y_n - y_1 & z_n - z_1 \end{bmatrix} \), we can compute \( H^TH \)
Let \( H^T H = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \)

Where,

\[
\begin{align*}
t_{11} &= (x_2 - x_1)^2 + (x_3 - x_1)^2 + \cdots (x_n - x_1)^2, \\
t_{12} &= (x_2 - x_1)(y_2 - y_1) + (x_3 - x_1)(y_3 - y_1) + \cdots (x_n - x_1)(y_n - y_1), \\
t_{13} &= (x_2 - x_1)(z_2 - z_1) + (x_3 - x_1)(z_3 - z_1) + \cdots (x_n - x_1)(z_n - z_1), \\
t_{21} &= t_{21} + t_{23} = (y_2 - y_1)^2 + (y_3 - y_1)^2 + \cdots (y_n - y_1)^2, \\
t_{22} &= t_{22} = (y_2 - y_1)(z_2 - z_1) + \cdots (y_n - y_1)(z_n - z_1), \\
t_{31} &= t_{31}, t_{32} = t_{33} = (z_2 - z_1)^2 + (z_3 - z_1)^2 + \cdots (z_n - z_1)^2.
\end{align*}
\]

And, \( n \) denotes the total number of the anchor nodes.

Then, \( det = t_{11}(t_{22}t_{33} - t_{23}^2) - t_{12}(t_{31}t_{33} - t_{32}t_{31}) + t_{13}(t_{21}t_{32} - t_{22}t_{31}) \).

So, \( adj(H^T H) = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \rightarrow \begin{bmatrix} + & + & + \\ - & - & - \end{bmatrix} = \begin{bmatrix} -k_{12} & k_{22} & -k_{32} \\ -k_{13} & k_{23} & -k_{33} \end{bmatrix} \)

Where, \( k_{11} = t_{22}t_{33} - t_{23}^2, k_{12} = t_{21}t_{33} - t_{32}t_{31}, k_{13} = t_{21}t_{32} - t_{22}t_{31}, k_{21} = k_{12}, k_{22} = t_{11}t_{33} - t_{13}t_{31}, k_{23} = t_{11}t_{32} - t_{12}t_{31}, k_{31} = k_{13}, k_{32} = k_{23}, k_{33} = t_{11}t_{22} - t_{12}t_{21} \).

While \( k_{12} = k_{21}, k_{13} = k_{31}, and k_{23} = k_{32} \), we can rewrite the \( (adj(H^T H)) \) matrix as shown in equation A.2 below.

\[
adj(H^T H) = \begin{bmatrix} k_{11} & -k_{12} & k_{13} \\ -k_{12} & k_{22} & -k_{32} \\ k_{13} & -k_{23} & k_{33} \end{bmatrix}
\] (A.2)

Therefore, to complete reforming the localization formula mentioned in the mathematical module section (equation 13), we need to modify part \( (H^T b) \) of the aforementioned equation.

\[
H^T b = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 & \cdots & x_n - x_1 \\ y_2 - y_1 & y_3 - y_1 & \cdots & y_n - y_1 \\ z_2 - z_1 & z_3 - z_1 & \cdots & z_n - z_1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} (x_2 - x_1)b_1 + (x_3 - x_1)b_2 + \cdots (x_n - x_1)b_n \\ (y_2 - y_1)b_1 + (y_3 - y_1)b_2 + \cdots (y_n - y_1)b_n \\ (z_2 - z_1)b_1 + (z_3 - z_1)b_2 + \cdots (z_n - z_1)b_n \end{bmatrix}
\]

So, Equation 13 presented in the mathematical module section could be written as shown in equation A.3.

\[
\hat{T} = \begin{bmatrix} \hat{T}_x \\ \hat{T}_y \\ \hat{T}_z \end{bmatrix} = \frac{1}{det} \begin{bmatrix} k_{11} & -k_{12} & k_{13} \\ -k_{12} & k_{22} & -k_{32} \\ k_{13} & -k_{23} & k_{33} \end{bmatrix} \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & (x_2 - x_1)b_1 + (x_3 - x_1)b_2 + \cdots (x_n - x_1)b_n \\ y_2 - y_1 & (y_2 - y_1)b_1 + (y_3 - y_1)b_2 + \cdots (y_n - y_1)b_n \\ z_2 - z_1 & (z_2 - z_1)b_1 + (z_3 - z_1)b_2 + \cdots (z_n - z_1)b_n \end{bmatrix}
\] (A.3)

Finely, the compact equations of the three target coordinates are presented in equations A.4, A.5, and A.6.

\[
\hat{T}_x = x_1 + \sum_{i=2}^{n} \frac{b_i}{det} \left[ (k_{11} \Sigma_{l=2}^{n} x_l - x_1) - (k_{12} \Sigma_{l=2}^{n} x_l - x_1) + (k_{13} \Sigma_{l=2}^{n} x_l - x_1) \right] \] (A.4)
\[
\hat{T}_y = y_1 + \sum_{i=2}^{n} \frac{b_i}{det} \left[ (-k_{12} \Sigma_{l=2}^{n} x_l - x_1) + (k_{22} \Sigma_{l=2}^{n} x_l - x_1) - (k_{23} \Sigma_{l=2}^{n} x_l - x_1) \right] \] (A.5)
\[
\hat{T}_z = z_1 + \sum_{i=2}^{n} \frac{b_i}{det} \left[ (k_{13} \Sigma_{l=2}^{n} x_l - x_1) - (k_{23} \Sigma_{l=2}^{n} x_l - x_1) + (k_{33} \Sigma_{l=2}^{n} x_l - x_1) \right] \] (A.6)

**B. MSE Formula**

The generic form of an MSE is shown in equation B.1

Where, \( E \) denotes the expectation value. Where, \( \hat{T} \) and \( T \) denote the estimated and true value of a target. Let us take the \( x \) coordinate as a target and find its MSE. The estimated value of \( x \) coordinate is
presented in equation A.4. So, we need to find the real value of the x coordinate. As we mentioned equation 4 in section 2, the generic formula of the estimated distance is

\[ d_i = r_i + e_i \rightarrow r_i = d_i - e_i \]  
\[ \text{(B.2)} \]

Where, \( r; d; \) and \( e \) denote the real, estimated, and residual error of distance between two points (a target and anchor node) respectively. Then, equation B.2 presents the real distance. The equation 3 in section 2 will become as equation B.3 below.

\[ v_i = \frac{(x_i - x_1)^2 + (y_i - y_1)^2 + (z_i - z_1)^2 + (d_1 - e_1)^2 - (d_i - e_i)^2}{2} \]

\[ = \frac{(x_i - x_1)^2 + (y_i - y_1)^2 + (z_i - z_1)^2 + d_i^2 - d_1^2}{2} \]

\[ + \frac{e_i^2 - e_1^2 + 2e_1d_i - 2e_id_1}{2} \]  
\[ \text{(B.3)} \]

In equation B.3, the highlighted part is the value of \( b \) in equation 5 section 2, so, let denote the other part of equation B.3 as shown in equation B.4.

\[ \varepsilon = \frac{e_1^2 - e_i^2 + 2d_ie_i - 2d_i e_i}{2} \rightarrow \varepsilon = \left[ \begin{array}{c} \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{array} \right] \]  
\[ \text{(B.4)} \]

Then, reforming equation B.3 as shown in equation B.5.

\[ v_i = b_i + \varepsilon_i \rightarrow v = \left[ \begin{array}{c} v_2 \\ \vdots \\ v_n \end{array} \right] \]  
\[ \text{(B.5)} \]

Based on equations A.4 A.5, and A.6, we replace the \( b \) vector by \( v \) vector (Eq. B5). We will proceed and compute the target coordinates, but we will proceed only with \( x \) coordinate, because \( y \) and \( z \) coordinate have the same procedure of the \( x \) computation, and only the change with \( k \) parameter. So, the real \( x \) coordinate \( T_x \) will be as presented in equation B.6.

\[ T_x = x_1 + \sum_{i=2}^{n} \frac{b_i + \varepsilon_i}{\det} \left[ \begin{array}{c} k_{11} \sum_{i=2}^{n} x_i - x_1 - \sum_{i=2}^{n} y_i - y_1 + \sum_{i=2}^{n} z_i - z_1 \\ k_{12} \sum_{i=2}^{n} y_i - y_1 + \sum_{i=2}^{n} z_i - z_1 \\ k_{13} \sum_{i=2}^{n} z_i - z_1 \end{array} \right] \]

\[ = x_1 + \sum_{i=2}^{n} \frac{b_i}{\det} \left[ k_{11} \sum_{i=2}^{n} x_i - x_1 - \sum_{i=2}^{n} y_i - y_1 + \sum_{i=2}^{n} z_i - z_1 \right] 
\]

\[ + \sum_{i=2}^{n} \frac{\varepsilon_i}{\det} \left[ k_{11} \sum_{i=2}^{n} x_i - x_1 - \sum_{i=2}^{n} y_i - y_1 + \sum_{i=2}^{n} z_i - z_1 \right] \]  
\[ \text{(B.6)} \]

The yellow highlighted part in Eq. B.6 equals to the estimated \( x \) coordinate (Eq. A.4), then we obtain equation B.7 for the true value of the \( x \) coordinate.

\[ T_x = \tilde{T}_x + \sum_{i=2}^{n} \frac{\varepsilon_i}{\det} \left( k_{11} \sum_{i=2}^{n} x_i - x_1 - \sum_{i=2}^{n} y_i - y_1 + \sum_{i=2}^{n} z_i - z_1 \right) \]  
\[ \text{(B.7)} \]

Then, by modifying equation B.1, we obtain
\[
\text{MSE}(\hat{T}_x) = E\left\{ \hat{T}_x^2 \right\} - 2E[\hat{T}_x] \left(E[\hat{T}_x] + \sum_{l=2}^n \frac{E[e_l]}{\det} \langle k_{11} \sum_{i=2}^n x_i - x_1 \rangle - \langle k_{12} \sum_{i=2}^n y_i - y_1 \rangle + \langle k_{13} \sum_{i=2}^n z_i - z_1 \rangle \rangle \right) + \left[ \sum_{l=2}^n \frac{E[e_l]}{\det} \langle k_{11} \sum_{i=2}^n x_i - x_1 \rangle - \langle k_{12} \sum_{i=2}^n y_i - y_1 \rangle + \langle k_{13} \sum_{i=2}^n z_i - z_1 \rangle \rangle \right]
\]

We proceed with the \text{MSE}(\hat{T}_x)

\[
\text{MSE}(\hat{T}_x) = E\left\{ \hat{T}_x^2 \right\} - 2E[\hat{T}_x] \hat{T}_x - 2E[\hat{T}_x] \sum_{l=2}^n \frac{E[e_l]}{\det} \langle k_{11} \sum_{i=2}^n x_i - x_1 \rangle - \langle k_{12} \sum_{i=2}^n y_i - y_1 \rangle + \langle k_{13} \sum_{i=2}^n z_i - z_1 \rangle \rangle + \left[ \sum_{l=2}^n \frac{E[e_l]}{\det} \langle k_{11} \sum_{i=2}^n x_i - x_1 \rangle - \langle k_{12} \sum_{i=2}^n y_i - y_1 \rangle + \langle k_{13} \sum_{i=2}^n z_i - z_1 \rangle \rangle \right]^2 \quad (B.8)
\]

It is clear from equation B.8 above that highlighted parts of this equation will be eliminated, and the compact equation of the MSE(\text{T}_x) will be only the non-highlighted part of the equation as shown in equation B.9 below.

\[
\text{MSE}(\hat{T}_x) = \left[ \sum_{l=2}^n \frac{E[e_l]}{\det} \langle k_{11} \sum_{i=2}^n x_i - x_1 \rangle - \langle k_{12} \sum_{i=2}^n y_i - y_1 \rangle + \langle k_{13} \sum_{i=2}^n z_i - z_1 \rangle \rangle \right]^2 \quad (B.9)
\]

Using the same way, we find \text{MSE}(\hat{T}_y) and \text{MSE}(\hat{T}_z) as presented by equations B.10 and B.11 respectively.

\[
\text{MSE}(\hat{T}_y) = \left[ \sum_{l=2}^n \frac{E[e_l]}{\det} \langle \neg k_{12} \sum_{i=2}^n x_i - x_1 \rangle + \langle k_{22} \sum_{i=2}^n y_i - y_1 \rangle - \langle k_{23} \sum_{i=2}^n z_i - z_1 \rangle \rangle \right]^2 \quad (B.10)
\]

\[
\text{MSE}(\hat{T}_z) = \left[ \sum_{l=2}^n \frac{E[e_l]}{\det} \langle k_{13} \sum_{i=2}^n x_i - x_1 \rangle - \langle k_{23} \sum_{i=2}^n y_i - y_1 \rangle + \langle k_{33} \sum_{i=2}^n z_i - z_1 \rangle \rangle \right]^2 \quad (B.11)
\]

Then, the compact equation of \text{MSE}(\hat{T}) is shown in equation B.12 below.

\[
\text{MSE}(\hat{T}) = \frac{\text{MSE}(\hat{T}_x) + \text{MSE}(\hat{T}_y) + \text{MSE}(\hat{T}_z)}{3} \quad (B.12)
\]

References
[1] Ferreira A G, Fernandes D, Catarino A P, and Monteiro J L 2017 Localization and Positioning Systems for Emergency Responders: a Survey IEEE DOI 10.1109/COMST, 2703620.
[2] Rahmadini A P, Kristalina P and Sudarsono A 2017 An improved fingerprint method based on K-NN algorithm for indoor multiple object tracking international Electronics Symposium on engineering technology and applications (IESETA) (Surabaya, 2017) pp 239-244
[3] Rashid H, Turuk A K 2015 Dear reckoning localization technique for mobile wireless sensor Network IET Wireless Sensor System 2015 87-98
[4] Albadhahi A, Morell A, Vicario J L 2019 Anchor selection for UWB indoor positioning system Emerging Telecommunication Technologies. 30(6) 1-17
[5] Albaidhani A 2020 Modified mathematical formula of MSE for live evaluation of a positioning system. *Preprint on Authorea*. **2020** 1-6