

# Euclidean Distance Matrix-Based Rapid Fault Detection and Exclusion

Derek Knowles<sup>1</sup> | Grace Gao<sup>2</sup>

<sup>1</sup> Department of Mechanical Engineering,  
Stanford University

<sup>2</sup> Department of Aeronautics and  
Astronautics, Stanford University

## Correspondence

Grace Gao

Department of Aeronautics and  
Astronautics, Stanford University  
Stanford, CA, USA, 94305

Email: [gracegao@stanford.edu](mailto:gracegao@stanford.edu)

## Abstract

Faulty signals from global navigation satellite systems (GNSSs) often lead to erroneous position estimates. A variety of fault detection and exclusion (FDE) methods have been proposed in prior research to both detect and exclude faulty measurements. This paper introduces a new technique for the FDE of GNSS measurements using Euclidean distance matrices. After a brief introduction to Euclidean distance matrices, both the detection and exclusion strategy is explained in detail. Euclidean distance matrix-based FDE is verified in two separate real-world data sets and proven to accurately detect and exclude GNSS faults on an average of 1.4-times faster than residual-based FDE and 70-times faster than solution separation FDE.

## Keywords

GNSS, Euclidean distance matrix, fault detection, fault exclusion, fault isolation

## 1 | INTRODUCTION

Signals from global navigation satellite systems (GNSSs) that are used for navigation must be monitored for potential faults in order to consistently provide an accurate position estimate. Common GNSS signal faults include receiving GNSS signals from satellites that are non-line-of-sight (NLOS), receiving signals that have reflected off of foliage or buildings (referred to as *multipath signals*; Enge, 1994), or faults caused by atmospheric effects. This paper focuses on the detection of these faults in which the GNSS signal is received with a time delay and does not address other types of GNSS signal faults such as when satellites are transmitting incorrect signals. One of the methods for fault detection and fault exclusion (FDE) uses the signal-to-noise ratio (SNR) to eliminate faulty signals (Bilich & Larson, 2007). However, the front end of most GNSS receivers contains automatic gain control that increases the gain of low strength signals and makes the SNR difference between faulty and non-faulty signals difficult to distinguish (Borowski et al., 2012; Wang et al., 2015). Beyond using SNR values alone, there are two common categories for more advanced FDE algorithms: solution separation and residual based.

Solution separation computes the least-squares position estimate using all measurements and also the position estimate using subsets of measurements (Blanch et al., 2012; Joerger et al., 2014; Ma et al., 2019; Pullen & Joerger, 2021;

**TABLE 1**  
Time Complexity of Solution Separation, Residual-Based, and EDM-Based Fault Detection and Exclusion

	<b>Solution Separation</b>	<b>Residual-based</b>	<b>EDM-based</b>
<b>Fault Detection</b>	$\mathcal{O}\left(\frac{m^f}{f!}\right)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
<b>Fault Exclusion</b>	$\mathcal{O}\left(\frac{m^f}{f!}\right)$	$\mathcal{O}(m)$	$\mathcal{O}(f)$

Note:  $m$  is the number of measurements in the measurement epoch and  $f$  is the number of faults hypothesized. All derivations for time complexity approximations included in Appendix A.

Zhu et al., 2018). A fault is detected if the normalized distance between the position estimate using all measurements and the position estimate using a subset of measurements is greater than the pre-determined threshold. Since solution separation requires the calculation of the position estimate using all measurement subset combinations, the time complexity of solution separation increases in a combinatorial manner as the number of faults hypothesized increases. Since the exact time complexity of solution separation varies widely due to the least-squares implementation and programming language, we simplify the time complexity of solution separation in Table 1 to a first-order approximation that accounts for the number of times the least-squares algorithm must run. The complete derivation for the time complexity approximation of solution separation is included in Appendix A.

Residual-based methods calculate the difference between measured and expected pseudorange values (Ma et al., 2019; Parkinson & Axelrad, 1988; Pullen & Joerger, 2021). If the normalized difference is larger than the pre-determined threshold, then a fault is detected. When a fault is detected, residual-based FDE creates measurement subsets in which a single measurement is removed and iterates throughout the subsets, recalculating the difference between measured and expected pseudoranges to determine which measurement to exclude (Parkinson & Axelrad, 1988). For this reason, residual-based fault detection is essentially constant with respect to the number of measurements and the number of faults hypothesized; however, residual-based fault exclusion scales linearly with respect to the number of measurements. Similarly to solution separation, in Table 1, we simplify the time complexity of residual-based FDE to a first-order approximation that accounts for the number of iterative detection checks that must be performed. The full approximation derivation is included in Appendix A.

This paper is based on our recent ION GNSS+ conference paper (Knowles & Gao, 2021b) and proposes a novel FDE algorithm that uses Euclidean distance matrices (EDMs), which are formally defined in Section 2, to rapidly detect and exclude GNSS signal faults. Like the residual-based and solution separation FDE methods discussed above, EDM-based FDE is also a snapshot method, meaning that it detects and excludes signals by analyzing measurements in a single snapshot of time. The proposed FDE method has lower computational complexity than both solution separation and residual-based FDE since it has a constant time fault detection solution and its fault exclusion step only scales linearly with respect to the number of faults hypothesized. A summary of the time complexity for each fault detection and exclusion strategy is included in Table 1. Additionally, EDM-based FDE does not require an explicit estimate of the receiver's location (like residual-based FDE requires) and is shown to detect and exclude measurement faults while using a broader range of thresholding parameters than either solution separation or residual-based FDE.

The contributions of this paper are as follows:

1. We propose a novel fault detection and exclusion method that leverages Euclidean distance matrices.
2. We use two real-world data sets to quantitatively validate time complexity and its ability to detect and exclude faults with a broad range of thresholding parameters when compared to both solution separation and residual-based FDE methods.

This paper begins by introducing the relevant theory of Euclidean distance matrices (EDMs) in Section 2. Next, in Section 3, we explain how to create an EDM using GNSS measurements. In Section 4, we present our methodology for fault detection and fault exclusion that leverages EDMs constructed from GNSS measurements. Finally, in Section 5, we present experimental results from two real-world data sets. We compare the fault detection sensitivity and robustness to changes in the thresholding parameters for solution separation, residual-based, and EDM-based FDE.

## 2 | EUCLIDEAN DISTANCE MATRIX PRELIMINARIES

Previously, EDMs were used to solve a wide range of relative positioning and alignment problems. EDMs have been used in acoustics for surveying rooms with echoes (Dokmanic, 2015), biology for protein structure alignment (Holm & Sander, 1993), and machine learning for dimensionality reduction (Tenenbaum et al., 2000). EDM-based algorithms have also been used for sensor network localization (Destino et al., 2007; Drineas et al., 2006; Oguz-Ekim et al., 2011; Wu et al., 2019) and multi-robot formation control (Ahn & Oh, 2010). EDM-based tools are much less prevalent in the GNSS navigation community despite the wealth of information they provide about ranging measurements. This section will present foundational theory about EDMs prior to Section 3, which explains how to create an EDM with GNSS measurements.

Consider  $p$  points in  $\mathbb{R}^n$ ,  $\{x_1, x_2, \dots, x_p\}$ , stacked and compactly represented as  $\mathbf{X} \in \mathbb{R}^{n \times p}$ . EDMs consist of the squared Euclidean distances between all points (i.e.,  $d_{ij} = \|x_i - x_j\|^2$ ). An EDM,  $\mathbf{D}$ , is analytically constructed with the following formula (Dokmanic et al., 2015):

$$\mathbf{D} = \mathbf{1} \text{diag}(\mathbf{X}^\top \mathbf{X})^\top - 2\mathbf{X}^\top \mathbf{X} + \text{diag}(\mathbf{X}^\top \mathbf{X}) \mathbf{1}^\top \quad (1)$$

where  $\text{diag}(\cdot)$  returns the diagonal of a matrix as a column vector and  $\mathbf{1}$  is a column vector of ones. This formula is further simplified by the following rewritten EDM construction formula in terms of the Gram matrix,  $\mathbf{G} = \mathbf{X}^\top \mathbf{X}$ :

$$\mathbf{D} = \mathbf{1} \text{diag}(\mathbf{G})^\top - 2\mathbf{G} + \text{diag}(\mathbf{G}) \mathbf{1}^\top \quad (2)$$

Gram matrices are positive semidefinite by construction. Most importantly for this proposed work, the rank of a Gram matrix,  $\mathbf{G}$ , equals the dimension of the space spanned by the vectors in  $\mathbf{X}$  (i.e.,  $\text{rank}(\mathbf{G}) = n$ ; Horn & Johnson, 2013).

While Equation (1) and Equation (2) show how an EDM can be constructed with knowledge of known point positions, EDMs can also be constructed without any known point positions, but rather with the relative distances between pairs of points. Figure 1 provides an illustrative example of how EDMs are constructed with known relative distances between pairs of points. In Figure 1(a), we illustrate

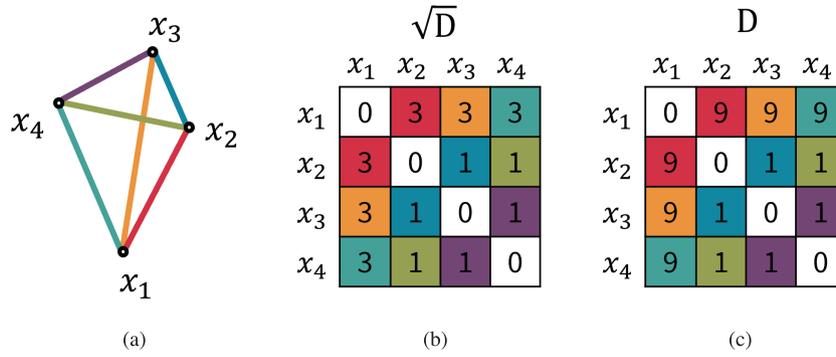


FIGURE 1 Visualization of how relative distances between points are used to construct a Euclidean distance matrix (EDM): subfigure (a) shows a fully connected point configuration, subfigure (b) is the matrix  $\sqrt{\mathbf{D}}$ , and subfigure (c) is the Euclidean distance matrix,  $\mathbf{D}$ .

a fully connected tetrahedron point configuration. Point  $x_1$  is at a distance three units away from all other points, and points  $x_2$ ,  $x_3$ , and  $x_4$  have a distance of one unit between one another. Figure 1(b) shows the matrix,  $\sqrt{\mathbf{D}}$ , is constructed by writing down the distances between all pairs of points in the system. There is a distance of zero between any point and itself. Figure 1(c) exhibits the construction of the Euclidean distance matrix,  $\mathbf{D}$ , by squaring the distances between all pairs of points in the system.

If an EDM is constructed with the relative distances between measurements, then the corresponding Gram matrix is recovered through double centering the EDM as follows (see Section 5.4.2.2 in Dattorro [2015] for a more thorough explanation):

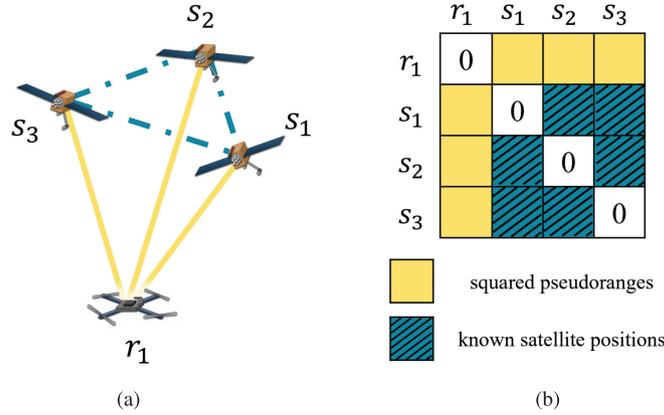
$$\mathbf{G} = -\frac{1}{2}\mathbf{J}\mathbf{D}\mathbf{J} \quad \text{where } \mathbf{J} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top \quad (3)$$

The matrix  $\mathbf{J}$  is called the geometric centering matrix,  $\mathbf{I}$  represents the identity matrix, and  $\mathbf{1}$  is a column vector of ones.

In this section, we showed that an EDM could be constructed from either known point locations or the relative distances between pairs of points. In Section 3, we will use a combination of both of these methods to construct an EDM for a GNSS application. We additionally showed that it is possible to recover a Gram matrix,  $\mathbf{G}$ , from an EDM and that, by construction, Gram matrices have a rank equal to the state space (i.e.,  $\text{rank}(\mathbf{G}) = n$ ).

### 3 | GNSS EDM FORMULATION

In this section, we leverage the theory about EDMs from Section 2 to create an EDM directly from GNSS measurements. We present an example scenario with a single GNSS receiver,  $r_1$ , that is receiving pseudorange measurements from  $m$  satellites (see Figure 2[a]). Figure 2(b) shows the sources from which we construct an EDM,  $\mathbf{D}_c$ , using GNSS measurements. The top row and leftmost column are comprised of the squared measured pseudoranges. The remaining bottom-right square of the EDM is filled utilizing the known positions of the satellites in view that are obtained from the navigation ephemeris message or cellular networks. This part of the EDM is filled according to Equation (1), where  $\mathbf{X} \in \mathbb{R}^{n \times m}$  is a matrix of the known satellite positions for each of the satellites from which the receiver has



**FIGURE 2** A combination of squared pseudoranges and known satellite positions are used to construct the EDM shown in subfigure (b) for a single receiver and for multiple satellites in subfigure (a).

obtained measurements. For satellite positioning,  $n = 3$ , representing the fact that satellite positions are provided in three-dimensional coordinates. Thus, the combination of both measured pseudoranges and known satellite positions are used to construct the EDM,  $\mathbf{D}_c \in \mathbb{R}^{(m+1) \times (m+1)}$ . In this section, we define the EDM as  $\mathbf{D}_c$  instead of  $\mathbf{D}$  to represent the fact that the GNSS EDM is constructed not from truth values, but from measured pseudorange and satellite positions that may both contain noise. Note that the construction of  $\mathbf{D}_c$  does not require any knowledge about the absolute position of the receiver.

The construction of the EDM,  $\mathbf{D}_c$ , may remind some readers of the dilution of precision (DOP) metric commonly used to describe the role of user-satellite geometry on measurement error (Enge, 1994). However, while DOP is based on the relative direction to satellites in view,  $\mathbf{D}_c$  is constructed using the distances to each satellite. The fault detection method illustrated in Section 4 is unaffected by the relative direction to satellites in view. Additionally, unlike DOP, our metric does not require an explicit estimate for the receiver's position.

In this section, we demonstrated how to construct an EDM directly from measured pseudoranges and known satellite positions.

## 4 | APPROACH

We now propose an algorithm that uses the constructed EDM,  $\mathbf{D}_c$ , for rapid fault detection and exclusion of GNSS signals. First, we explain our proposed method for fault detection.

The key to EDM-based fault detection is the property of Gram matrices defined in Section 2 in which the Gram matrices have a rank equal to the state space (i.e.,  $\text{rank}(\mathbf{G}) = n$ ; Horn & Johnson, 2013). If localizing in a three-dimensional state space (i.e.,  $n = 3$ ), this means that when we recover a Gram matrix,  $\mathbf{G}_c$ , from our observed GNSS EDM,  $\mathbf{D}_c$ , the Gram matrix,  $\mathbf{G}_c$ , would have a rank of three.

As a method to check the rank of the Gram matrix,  $\mathbf{G}_c$ , we perform singular value decomposition and inspect the calculated singular values. The singular value decomposition of  $\mathbf{G}_c$  is written as:

$$\text{SVD}(\mathbf{G}_c) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (4)$$

where  $\mathbf{U}$  are the left singular vectors, the diagonal of  $\Sigma$  contains the singular values of  $\mathbf{G}_c, \sigma_1, \sigma_2, \dots, \sigma_{m+1}$  sorted by value, and the columns of  $\mathbf{V}$  are the right singular vectors.

In the GNSS EDM formulation, we perform fault detection and exclusion (FDE) in a three-dimensional state space representing the three coordinate axes of our position solution. As will be discussed later in Equation (6) of Section 5, we assume that, prior to FDE, the receiver clock bias is solved independently (such as with an extended Kalman filter). In a three-dimensional state space without any faults present, we find that the first three singular values of  $\mathbf{G}_c$  are nonzero and all subsequent singular values (no matter how many there are) will be zero (Horn & Johnson, 2013). Having three nonzero singular values means that the Gram matrix,  $\mathbf{G}_c$ , is rank three, as expected, in the no-fault scenario. However, in a three-dimensional state space where faults *are* present, we find that there are more than three nonzero singular values of  $\mathbf{G}_c$ . The larger number of nonzero singular values indicates that the Gram matrix,  $\mathbf{G}_c$ , is higher than rank three.

Thus, our fault detection strategy is to inspect the singular values of the Gram matrix,  $\mathbf{G}_c$ , that we reconstructed from the observed EDM,  $\mathbf{D}_c$ , specifically looking at the singular values that would be zero if the EDM,  $\mathbf{D}_c$ , had been constructed with perfect measurements. In reality, because of measurement noise and numerical precision, the singular values of  $\mathbf{G}_c$  are likely small numbers, but not exactly zero. We use a thresholding technique to check how close the singular values are to zero. If the singular values of  $\mathbf{G}_c$  are near zero, then we predict that no faults are present in the measurements; however, if those singular values are much larger than zero (and above the chosen thresholding parameter), then we would predict that faults do exist in the measurements. Concretely, a fault is detected in EDM-based FDE if the test statistic shown in Equation (5) holds true.

$$\frac{\sigma_{n+1} \times \frac{1}{m-n} \sum_{i=n+1}^m \sigma_i}{\sigma_1} > \text{detection threshold} \quad (5)$$

The first element,  $\sigma_{n+1}$ , of the test statistic is the first singular value that we would expect to be zero if we had perfect measurements. In a three-dimensional state space (i.e.,  $n=3$ ), this would correspond to  $\sigma_4$ . The second element of the test statistic,  $\frac{1}{m-n} \sum_{i=n+1}^m \sigma_i$ , is the average of all singular values that would be zero if we had perfect measurements. In a three-dimensional state space, this would be the average of  $\sigma_4, \dots, \sigma_{m+1}$ . We normalize the test statistic by  $\sigma_1$ . Inherently, the magnitude of the singular values changes depending on the noise magnitude in the measurement data. Normalizing the test statistic means that the threshold of EDM-based fault detection is relatively agnostic to the noise magnitude of the measurements.

Modern computers perform the above fault detection steps rapidly so no iterative computations or comparisons are necessary. The first-order approximation of EDM-based fault detection runs in  $\mathcal{O}(1)$  time since it is essentially independent of the number of measurements and completely independent of the number of faults hypothesized. There is a small increase in computation time with an increase in the number of measurements since the Euclidean distance matrix,  $\mathbf{D}_c$ , would be larger with more measurements. Matrix multiplication and other similar operations take slightly more time with larger matrices. These computational time factors are minor when compared with iterative detection checks. As a first-order approximation shown in Table 1, the computation time of fault detection reduces to  $\mathcal{O}(1)$ .

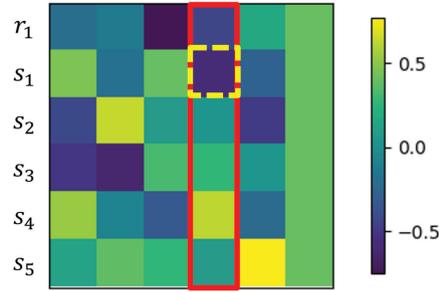


FIGURE 3 The left singular vectors,  $\mathbf{U}$ , of  $\mathbf{G}_c$  with  $u_4$  is highlighted in red while  $\mathbf{U}_{24}$  is highlighted in dashed yellow and is the maximum absolute value in the column indicating a fault from the  $s_1$  measurement.

We now present our method for EDM-based fault exclusion. If we detect a fault, then we look at the left and right singular vectors associated with  $\sigma_{n+1}$ , the first singular value that would be zero if we had perfect measurements. In a three-dimensional state space, this would mean looking at the fourth columns of  $\mathbf{U}$  and  $\mathbf{V}$ , which we write as  $u_4$  and  $v_4$ . Intuitively, these columns give us quantitative information on how much each measurement contributes to  $\sigma_{n+1}$  being nonzero. We sort  $u_4$  and  $v_4$  by absolute value and predict that the largest absolute value in the columns correspond to a potential measurement fault. We remove the potentially faulty measurement from the EDM,  $\mathbf{D}_c$ , and recheck whether a fault is still detected using the fault detection test statistic in Equation (5). This process can be repeated to detect multiple faults, meaning that EDM-based FDE scales linearly with respect to the number of faults hypothesized,  $\mathcal{O}(f)$ .

Figure 3 illustrates an example of the left singular vectors,  $\mathbf{U}$ , after a fault is detected. The largest absolute value in the  $u_{n+1}$  column is highlighted in dashed yellow and represents the successful identification of a fault from the  $s_1$  satellite.

Algorithm 1 outlines the proposed EDM-based FDE algorithm. The algorithm needs at least one more satellite than the localization solution requires to be able to exclude a faulty satellite. Since GNSS localization requires a minimum of four satellites for a position solution, the receiver must receive signals from at least five satellites in order to be able to exclude a faulty satellite signal. If there is more than one faulty signal, EDM-based FDE will, first, detect the largest faulty signal and, then, EDM-based FDE can be run again to detect the next-largest faulty signal. While Algorithm 1 only removes a single faulty satellite in each iteration, the algorithm can be repeated quickly to remove multiple faulty satellites until faults are no longer detected. For a detailed explanation of the theory behind multiple fault hypotheses, we direct the reader to Knight et al. (2010). An open-source implementation of EDM-based FDE is available in Knowles and Gao (2021a).

## 5 | RESULTS

We now compare the proposed EDM-based fault detection and exclusion strategy from Algorithm 1 with both residual-based and solution separation FDE using two real-world data sets. The first data set was used to test detection sensitivity and the second data set was used to test detection robustness among noisy measurements.

In addition to the steps in Algorithm 1, for a real-world implementation, the measured pseudoranges should be conditioned to remove known biases. Since EDM-based FDE works by matching distances with pairs of points, it works best if those distances are as close to their truth value as possible. We calculate the

conditioned pseudorange,  $\rho_c$ , by removing known biases from the raw measured pseudorange,  $\rho_m$ , according to the model:

$$\rho_c = \rho_m + r_b - I + T - s_b \quad (6)$$

where  $r_b$  is the receiver clock bias,  $I$  is the ionosphere delay,  $T$  is the troposphere delay, and  $s_b$  is the inter-signal range bias. In the results that follow, we estimated the receiver clock bias as a state in an extended Kalman filter (Kalman, 1960) and the atmospheric and inter-signal biases were either provided by the data set (Fu et al., 2020) or estimated from ground truth data (Reisdorf et al., 2016).

For each FDE method, we analyzed the corresponding average computation time and accuracy in excluding faults. To calculate the average computation time, we timed how long a Python implementation of each algorithm took to complete both fault detection and exclusion. We also calculated accuracy metrics derived from the confusion matrix shown in Figure 4.

We compare EDM-based FDE with residual-based and solution separation FDE using the quantitative accuracy rates in Equations (7)–(11). Emphasis is placed on the balanced accuracy metric (Equation [9]) over true positive rate (Equation [7]) or true negative rate (Equation [8]) since the real-world data sets we tested had an overall small percentage of ground truth faults in relation to all of the measurements in the data set. Comparing balanced accuracy prevents an FDE method from seemingly performing well by simply predicting every measurement to be non-faulty. Now that we have outlined our quantitative performance measures, we present the two real-world data sets used to validate EDM-based FDE. In the results we show, the thresholding parameters have been optimized to increase

#### ALGORITHM 1 EDM-Based FDE

- 1: **Inputs:**
- 2:  $\mathbf{D}_c$ : Euclidean distance matrix
- 3:  $n$ : dimensions of state space
- 4: **Initialize:**
- 5:  $\mathbf{G}_c \leftarrow -\frac{1}{2}\mathbf{J}\mathbf{D}_c\mathbf{J}$
- 6:  $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \text{SVD}(\mathbf{G}_c)$
- 7: **while** # satellites > 4 and detect\_fault( $\mathbf{\Sigma}$ ) **do**
- 8: index  $\leftarrow$  predict( $u_{n+1}, v_{n+1}$ )
- 9: exclude index from  $\mathbf{D}_c$
- 10:  $\mathbf{G}_c \leftarrow -\frac{1}{2}\mathbf{J}\mathbf{D}_c\mathbf{J}$
- 11:  $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \text{SVD}(\mathbf{G}_c)$
- 12: **end while**

		Predicted	
		Fault	No Fault
Ground Truth	Fault	TP: True Positive	MD: Missed Detection
	No Fault	FA: False Alarm	TN: True Negative

FIGURE 4 This confusion matrix depicts the relationship between the ground truth fault status of measurements and the measurement fault status as predicted by each FDE method.

the balanced accuracy metric in Equation (9) due to the reasons stated above. For other use cases, the thresholding parameters can be chosen to optimize the false alarm rate or missed detection rate in an approach similar to Yang et al. (2013). The open-source implementation of our EDM-based FDE is available online and includes all code needed to reproduce the result figures shown in this section (Knowles & Gao, 2021a).

$$\text{True Positive Rate} = \frac{\text{True Positive}}{\text{True Positive} + \text{Missed Detection}} \quad (7)$$

$$\text{True Negative Rate} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Alarm}} \quad (8)$$

$$\text{Balanced Accuracy} = \frac{1}{2}(\text{True Positive Rate} + \text{True Negative Rate}) \quad (9)$$

$$\text{Missed Detection Rate} = \frac{\text{Missed Detection}}{\text{Missed Detection} + \text{True Positive}} \quad (10)$$

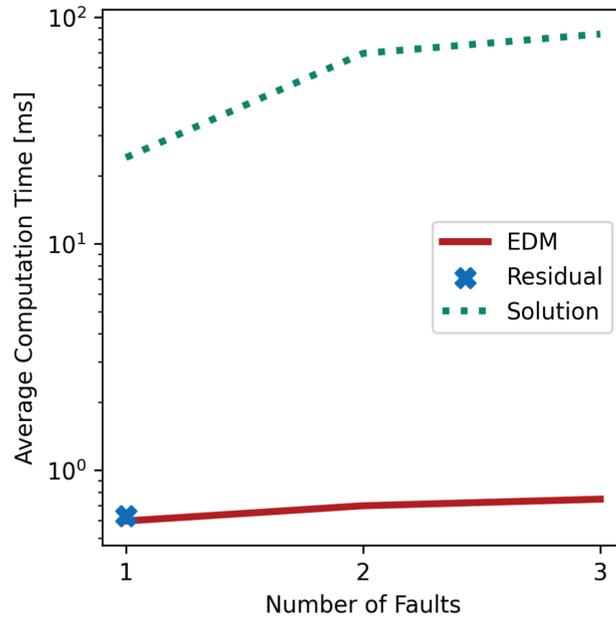
$$\text{False Alarm Rate} = \frac{\text{False Alarm}}{\text{False Alarm} + \text{True Negative}} \quad (11)$$

## 5.1 | Detection Sensitivity

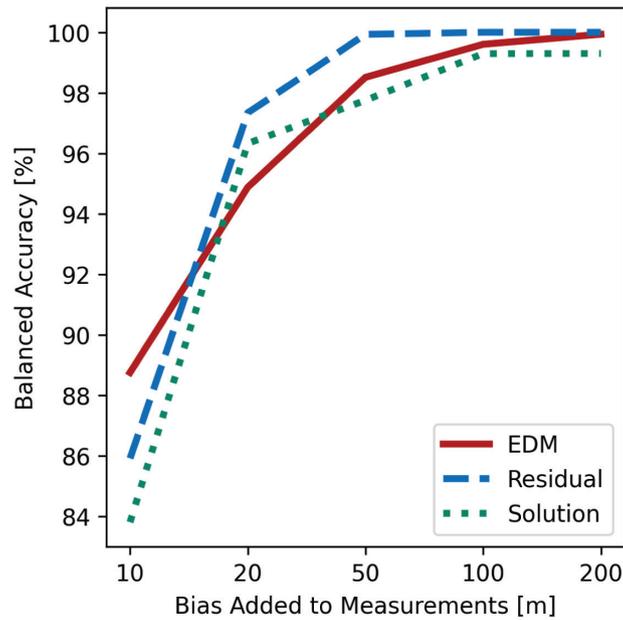
The first real-world data set we used was a relatively clean data set of about a 2,000-m trajectory with low average residuals (about 1.5 m) in an urban environment (Reisdorf et al., 2016). To test the detection sensitivity of each FDE method relative to the bias magnitude, we performed separate tests for fault magnitudes of 10, 20, 50, 100, and 200 m. For each of the five bias magnitude tests, we added the tested bias magnitude to the conditioned pseudorange,  $\rho_c$ , with a probability of 25% to a random satellite measurement (or set of satellite measurements depending on the test) at each timestep. Each FDE method used the same measurements at each timestep.

We, first, use this data set to compare how the average computation time of each FDE method was affected by increasing the number of faults hypothesized. Figure 5 summarizes the experimental results. Not only does residual-based FDE run faster than solution separation on an absolute scale, but it also scales better computationally as the number of faults hypothesized increases. This is due to the fact that fault exclusion of EDM-based FDE runs in  $\mathcal{O}(f)$  time compared to solution separation fault exclusion that runs in  $\mathcal{O}\left(\frac{m^f}{f!}\right)$  time, where  $f$  is the number of faults hypothesized and  $m$  is the number of measurements in the measurement epoch. As implemented in Parkinson and Axelrad (1988), residual-based FDE is only valid for a single fault hypothesis.

While solution separation and EDM-based FDE are compared using strict fault hypotheses, an additional advantage of EDM-based FDE is that a fault hypothesis is not strictly necessary. A fault hypothesis, in the case of EDM-based FDE, is only used to force termination of Algorithm 1 after the number of measurements excluded has reached the fault hypothesis. Without a fault hypothesis, EDM-based FDE would continue to remove measurements one after another until either a fault is no longer detected or only four satellite measurements are left. In practice, a maximum fault hypothesis can be used in EDM-based FDE to limit the maximum computation time allowed.

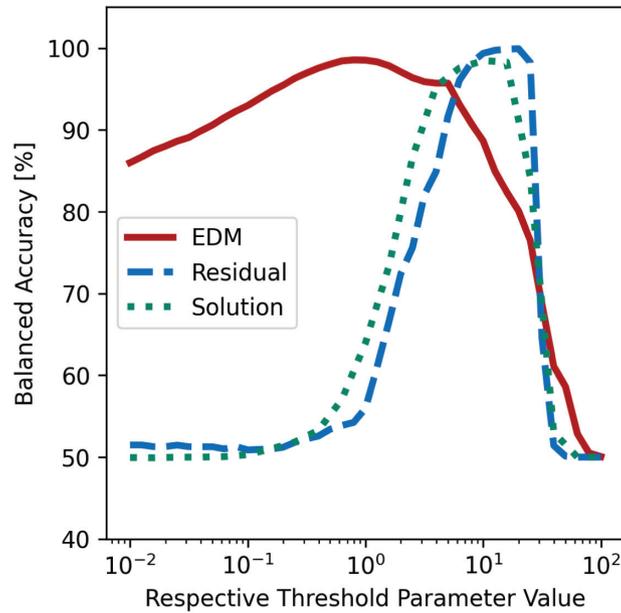


**FIGURE 5** Average computation time of each FDE method with an increasing number of faults hypothesized; the computation time of EDM-based FDE increases more slowly than solution separation as the number of faults hypothesized increases.



**FIGURE 6** Balanced accuracy (see Equation [9]) with respect to the magnitude of the faults added shows that EDM-based FDE rivals the balanced accuracy of the other two FDE methods with large magnitude biases excelling at low bias magnitudes.

Figure 6 shows the results of how balanced accuracy for each FDE method changes with respect to the magnitude of added biases. Not only does EDM-based FDE rival residual-based and solution separation FDE when it encounters measurements with large biases, but EDM-based FDE excels over the other two methods when measurements only have 10-m biases. EDM-based FDE excelling at the 10-m bias threshold is due to the fact that our choice in thresholding values was made with the intent of maximizing the balanced accuracy from



**FIGURE 7** The balanced accuracy (see Equation [9]) of each FDE method swept across a wide range of FDE thresholding parameters shows that EDM-based FDE is much more robust to changes in its thresholding parameter than the other two FDE methods.

Equation (9). As discussed below, EDM-based FDE is able to decrease the missed detection rate by  $\sim 15\%$  over residual-based or solution separation FDE at the expense of a  $\sim 6\%$  worse false alarm rate and, thus, achieves a better balanced accuracy metric overall.

Figure 7 shows that EDM-based FDE is able to maintain a high balanced accuracy across a wide range of thresholding parameters. Figure 7 was created by sweeping over a large range of thresholding parameters from  $10^{-2}$  to  $10^2$  for each FDE method. The plot shows that, while residual-based and solution separation FDE have high balanced accuracy for only a narrow range of thresholding parameters, EDM-based FDE maintains a reasonable balanced accuracy across a wide range of thresholding parameters. This means that EDM-based FDE is much less sensitive to small changes in its thresholding parameter and the user would not need to spend as much time parameter tuning to achieve reasonable results.

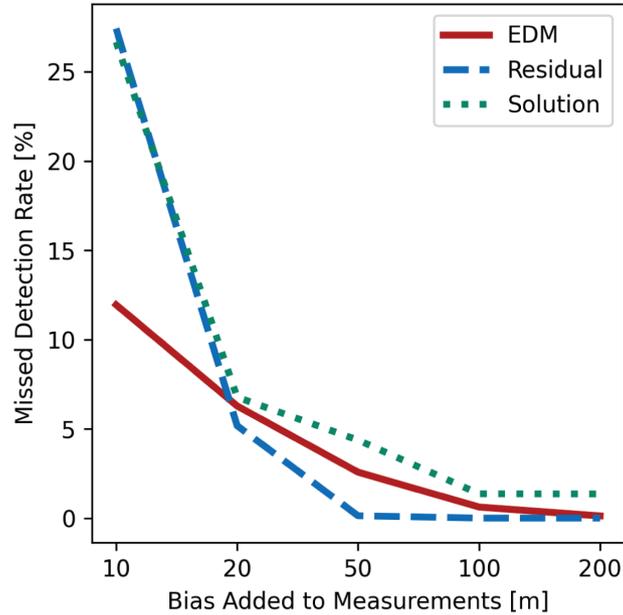
In Figure 8, we plot the missed detection rate with respect to the magnitude of added biases. EDM-based FDE is able to detect many more of the low-bias faults than residual-based or solution separation FDE and, thus, has a lower missed detection rate at low fault biases. However, from Figure 9, we see that EDM-based FDE has a much higher rate of false alarm than either of the other two FDE methods. This means that EDM-based FDE is more likely to predict a fault when no faults are present. This trade-off between missed detection rate and false alarm rate is due to our engineering decision to optimize the thresholding parameters based on the balanced accuracy but, depending on the use case, could be chosen to optimize either the missed detection rate or false alarm rate instead.

## 5.2 | Detection Robustness Among Noisy Measurements

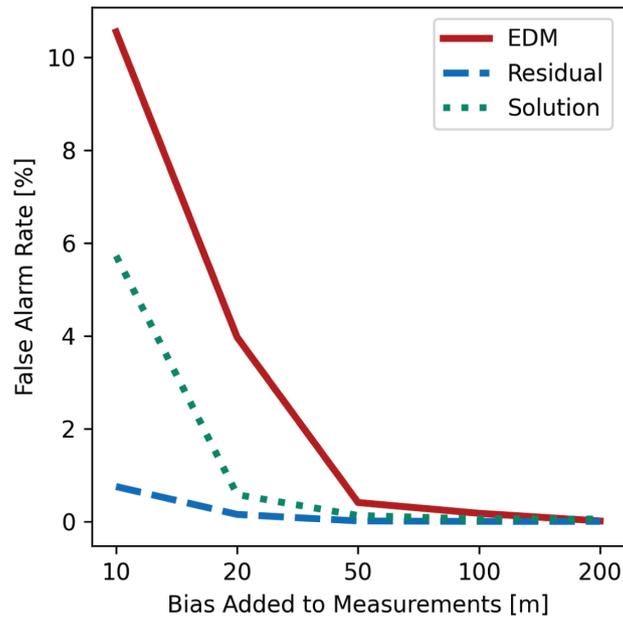
The second real-world data set that we used contained many more measurements (about 3.3 million measurements) representing roughly 36 hr of driving throughout the San Francisco Bay area and larger average residuals of about

20 m (Fu et al., 2020). We used this data set to test the robustness to noisier data across a wide range of open sky to urban conditions for each FDE method.

Using this data set, we plotted in Figure 10 the average computation time of each FDE method as a factor of the number of measurements received in the measurement epoch. Computation time was computed as the length of time it took a Python implementation of each algorithm to complete the fault detection and exclusion steps. Since this second real-world data set includes few faults relative to the total size of the data set, the computation time shown in Figure 10 is mostly representative of the computation time of the fault detection step of each FDE



**FIGURE 8** Missed detection rate (see Equation [10]) with respect to the magnitude of the faults added



**FIGURE 9** False alarm rate (see Equation [11]) with respect to the magnitude of the faults added

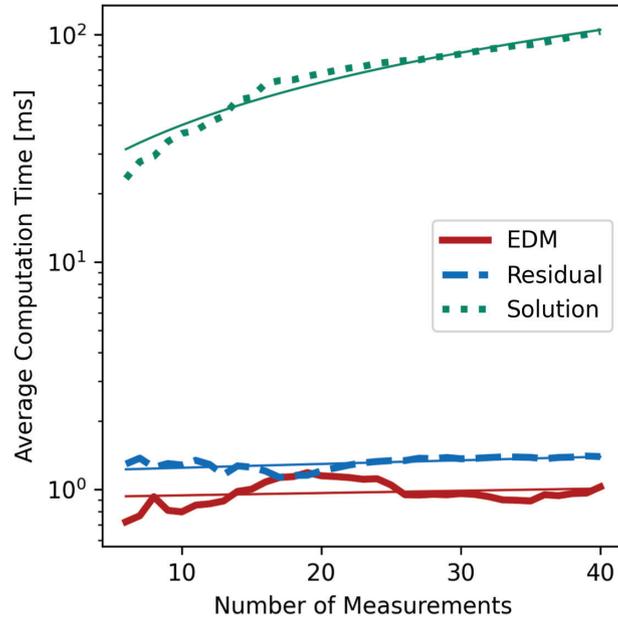


FIGURE 10 Average computation time of each FDE method with increasing measurements in each measurement epoch; also shown is the best line fit for each FDE method.

TABLE 2 Average Accuracy Metrics for the Larger, Noisier Android Data Set

	Solution Separation	Residual-Based	EDM-Based
Balanced Accuracy %	84.9	88.7	<b>91.6</b>
Missed Detection Rate %	29.9	18.4	<b>11.7</b>
False Alarm Rate %	<b>0.2</b>	4.2	5.1

algorithm with a fewer number of fault exclusion steps. We see that EDM-based FDE, on average, is 1.4-times faster than residual-based FDE and nearly 70-times faster than solution separation due to its lower computational complexity.

In Table 2, we present the accuracy results for each FDE method on this larger, noisier data set. For each FDE method, we used the FDE thresholding parameter that achieved the highest balanced accuracy rate across the entire data set. EDM-based FDE had a superior balanced accuracy and missed detection rate across the data set, but solution separation FDE had the lowest false alarm rate.

## 6 | CONCLUSION

This paper introduced the theory behind Euclidean distance matrices (EDMs) and then used that theory to propose a novel EDM-based method for GNSS signal fault detection and exclusion. Using two real-world data sets, we validated the theory that EDM-based FDE has superior computational complexity when increasing the number of faults hypothesized and measurements received compared with solution separation and residual-based FDE (see Table 1). We also showed that EDM-based FDE rivals solution separation and residual-based FDE in terms of exclusion accuracy. EDM-based FDE was shown to detect and exclude measurement faults while using a broader range of thresholding parameters than either solution separation or residual-based FDE.

## ACKNOWLEDGEMENTS

The authors thank all members of the Stanford Navigation and Vehicles Laboratory for their insightful discussion and especially Ashwin Kanhere and Shubh Gupta for contributing abundant feedback.

## REFERENCES

- Ahn, H. -S., & Oh, K. -K. (2010). Command coordination in multi-agent formation: Euclidean distance matrix approaches. *ICCAS 2010 - International Conference on Control, Automation and Systems*, 1592–1597. <https://doi.org/10.1109/iccas.2010.5669671>
- Bilich, A., & Larson, K. M. (2007). Mapping the GPS multipath environment using the signal-to-noise ratio (SNR). *Radio Science*, 42(6). <https://doi.org/10.1029/2007RS003652>
- Blanch, J., Walter, T., & Enge, P. (2012). Satellite navigation for aviation in 2025. *Proc. of the IEEE*, 100, 1821–1830. <https://doi.org/10.1109/JPROC.2012.2190154>
- Borowski, H., Isoz, O., Eklöf, F., Lo, S., & Akos, D. (2012). Detecting false signals with automatic gain control. *GPS World*, 23(4), 38–43. <https://www.gpsworld.com/detecting-false-signals-automatic-gain-control-12804>
- Dattorro, J. (2015). *Convex optimization: Euclidean distance geometry*. Meboo.
- Destino, G., Macagnano, D., & Abreu, G. (2007). A clusterized WLS localization algorithm for large scale WSNs. *4th Workshop on Positioning, Navigation and Communication 2007*, Hannover, Germany. <https://doi.org/10.1109/wpnc.2007.353643>
- Dokmanic, I. (2015). *Listening to distances and hearing shapes: inverse problems in room acoustics and beyond* [Doctoral dissertation, École Polytechnique Fédérale de Lausanne]. EPFL. <https://doi.org/10.5075/EPFL-THESIS-6623>
- Dokmanic, I., Parhizkar, R., Ranieri, J., & Vetterli, M. (2015). Euclidean distance matrices: essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6), 12–30. <https://doi.org/10.1109/MSP.2015.2398954>
- Drineas, P., Javed, A., Magdon-Ismail, M., Pandurangan, G., Virrankoski, R., & Savvides, A. (2006). Distance matrix reconstruction from incomplete distance information for sensor network localization. *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, Reston, VA. <https://doi.org/10.1109/SAHCN.2006.288510>
- Enge, P. K. (1994). The Global Positioning System: signals, measurements, and performance. *International Journal of Wireless Information Networks*, 1(2), 83–105. <https://doi.org/10.1007/BF02106512>
- Fu, G., Khider, M., & van Diggelen, F. (2020). Android raw GNSS measurement datasets for precise positioning. *Proc. of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020)*, 1925–1937. <https://doi.org/10.33012/2020.17628>
- Holm, L., & Sander, C. (1993). Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233(1), 123–138. <https://doi.org/10.1006/jmbi.1993.1489>
- Horn, R. A., & Johnson, C. R. (2013). *Matrix analysis* (2nd edition). Cambridge University Press. <https://doi.org/10.1017/cbo9780511810817>
- Joerger, M., Chan, F. -C., & Pervan, B. (2014). Solution separation versus residual-based RAIM. *NAVIGATION*, 61(4), 273–291. <https://doi.org/10.1002/navi.71>
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering, Transactions of the ASME*, 82(1), 35–45. <https://doi.org/10.1115/1.3662552>
- Knight, N. L., Wang, J., & Rizos, C. (2010). Generalised measures of reliability for multiple outliers. *Journal of Geodesy*, 84(10), 625–635. <https://doi.org/10.1007/s00190-010-0392-4>
- Knowles, D., & Gao, G. (2021a). *EDM-FDE* [Code repository]. <https://github.com/betaBison/edm-fde>
- Knowles, D., & Gao, G. (2021b). Euclidean distance matrix-based rapid fault detection and exclusion. *Proc. of the 34th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2021)*, St. Louis, MO, 3382–3391. <https://www.ion.org/publications/abstract.cfm?articleID=17973>
- Ma, X., Yu, K., Montillet, J. -P., & He, X. (2019). Equivalence proof and performance analysis of weighted least squares residual method and weighted parity vector method in RAIM. *IEEE Access*, 7, 97803–97814. <https://doi.org/10.1109/ACCESS.2019.2929073>
- Oguz-Ekim, P., Gomes, J. P., Xavier, J., & Oliveira, P. (2011). Robust localization of nodes and time-recursive tracking in sensor networks using noisy range measurements. *IEEE Transactions on Signal Processing*, 59(8), 3930–3942. <https://doi.org/10.1109/TSP.2011.2153848>
- Parkinson, B. W., & Axelrad, P. (1988). Autonomous GPS integrity monitoring using the pseudorange residual. *NAVIGATION*, 35(2), 255–274. <https://doi.org/10.1002/J.2161-4296.1988.TB00955.X>
- Pullen, S., & Joerger, M. (2021). GNSS integrity. In Y. J. Morton, F. van Diggelen, J. J. Spilker Jr., B. W. Parkinson, S. Lo, & G. X. Gao (Eds.), *Position, navigation, and timing technologies in the 21st century: integrated satellite navigation, sensor systems,*

- and civil applications*, Vol. 2 (pp. 591–617). Wiley. <https://www.wiley.com/en-us/Position,+Navigation,+and+Timing+Technologies+in+the+21st+Century:+Integrated+Satellite+Navigation,+Sensor+Systems,+and+Civil+Applications,+Volume+2-p-9781119458494>
- Reisdorf, P., Pfeifer, T., Breßler, J., Bauer, S., Weissig, P., Lange, S., Wanielik, G., & Protzel, P. (2016). The problem of comparable GNSS results—an approach for a uniform dataset with low-cost and reference data. *5th International Conference on Advances in Vehicular Systems, Technologies and Applications (VEHICULAR)*. <https://www.tu-chemnitz.de/projekt/smartLoc/paper/reisdorf2016.pdf>
- Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2319–2323. <https://doi.org/10.1126/science.290.5500.2319>
- Wang, L., Groves, P. D., & Ziebart, M. K. (2015). Smartphone shadow matching for better cross-street GNSS positioning in urban environments. *Journal of Navigation*, 68(3), 411–433. <https://doi.org/10.1017/S0373463314000836>
- Wu, Z., Yao, Z., & Lu, M. (2019). Coordinate establishment of ground-based positioning systems with anchor information. *IEEE Transactions on Vehicular Technology*, 68(3), 2517–2525. <https://doi.org/10.1109/TVT.2019.2893456>
- Yang, L., Wang, J., Knight, N. L., & Shen, Y. (2013). Outlier separability analysis with a multiple alternative hypotheses test. *Journal of Geodesy*, 87(6), 591–604. <https://doi.org/10.1007/s00190-013-0629-0>
- Zhu, N., Marais, J., Bétaille, D., & Berbineau, M. (2018). GNSS position integrity in urban environments: a review of literature. *IEEE Transactions on Intelligent Transportation Systems*, 19(9), 2762–2778. <https://doi.org/10.1109/TITS.2017.2766768>

**How to cite this article:** Knowles, D., & Gao, G. (2023). Euclidean distance matrix-based rapid fault detection and exclusion. *NAVIGATION*, 70(1). <https://doi.org/10.33012/navi.555>

## APPENDIX A | TIME COMPLEXITY DERIVATIONS

In this section, we outline the derivations of the first-order time complexities for each algorithm shown in Table 1. Exact computation time is dependent on numerous parameters such as programming language, specific function implementations, programming libraries used, CPU and GPU quality, etc. A detailed computation time estimate based on all of these parameters is beyond the scope of this work, so instead we provide a first-order approximation of the computational complexity based on the number of times the most time-consuming part of the algorithm needs to run.

For solution separation, the most time-consuming part of the algorithm is the least-squares position estimate (Blanch et al., 2012; Joerger et al., 2014; Ma et al., 2019; Pullen & Joerger, 2021; Zhu et al., 2018). For the fault detection step of solution separation, the least-squares position estimate is calculated using all measurement subset combinations with a single measurement removed. The number of combinations needed can be calculated using the standard combination formula:

$${}_m C_f = \frac{m!}{f!(m-f)!} \quad (\text{A1})$$

where  $m$  is the number of measurements in the measurement epoch and  $f$  is the number of faults hypothesized. The fault exclusion step of solution separation

removes the worst satellites depending on the number of faults hypothesized and does another solution-separation check to verify that no additional faults are detected. Hence, the solution separation exclusion step also has a time complexity according to the combination equation shown in Equation (A1). For the sake of a more intuitive and simple understanding, instead of using the combination formula from Equation (A1) directly, we use an upper bound for Equation (A1) that is derived below.

$$\frac{m!}{f!(m-f)!} = \frac{1}{f!} \frac{m!}{(m-f)!} = \frac{1}{f!} \frac{m \cdot (m-1) \cdot (m-2) \cdots (1)}{(m-f) \cdot (m-f-1) \cdots (1)} \leq \frac{1}{f!} m^f = \frac{m^f}{f!} \quad (\text{A2})$$

Using the upper bound approximation in Equation (A2) results in a first-order time complexity approximation of  $\mathcal{O}\left(\frac{m^f}{f!}\right)$  for both the fault detection and fault exclusion steps of solution separation.

The most time-consuming part of residual-based FDE is calculating the difference between the measured and expected pseudoranges (Ma et al., 2019; Parkinson & Axelrad, 1988; Pullen & Joerger, 2021). This step runs only once for residual-based fault detection, resulting in a time complexity approximation of  $\mathcal{O}(1)$ . For fault exclusion, residual-based FDE creates all measurement subset combinations with a single measurement removed and then computes the test statistic again based on the difference between the expected and measured pseudoranges (Parkinson & Axelrad, 1988). Since the test statistic calculation is performed for each subset, the first-order approximation for residual-based fault exclusion is  $\mathcal{O}(m)$ .

The most time-consuming part of EDM-based FDE shown in Algorithm 1 is the singular value decomposition (SVD) step. SVD is performed only once for fault detection resulting in a first-order time complexity approximation of  $\mathcal{O}(1)$  for EDM-based fault detection. For EDM-based fault exclusion, the SVD step is repeated at most one time for each fault hypothesized resulting in a time complexity approximation of  $\mathcal{O}(f)$ .

To conclusively demonstrate the lower time complexity of EDM-based FDE, we present computation time comparisons between all three methods when tested on real-world data in Figures 5 and 10 of the main text.