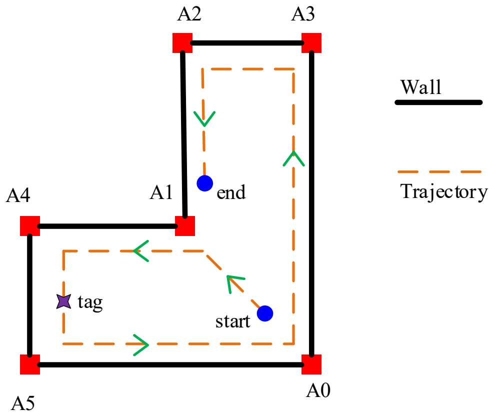
Supplementary Materials for
"Denoising, Outlier/Dropout Correction, and Sensor Selection in Range-Based Positioning"

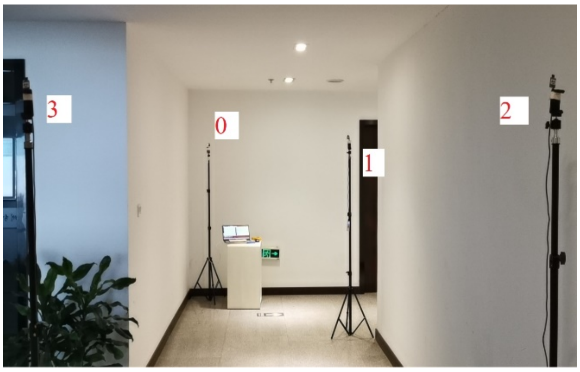
Abstract
In this supplementary materials, we are providing another example for sensor anomaly in Fig. 2. And, we are also answering the possible questions from readers. See contents below.

For any other confusions, critiques, comments, suggestions, and/or insights, the corresponding author is open and happy to hear from readers to discuss together.

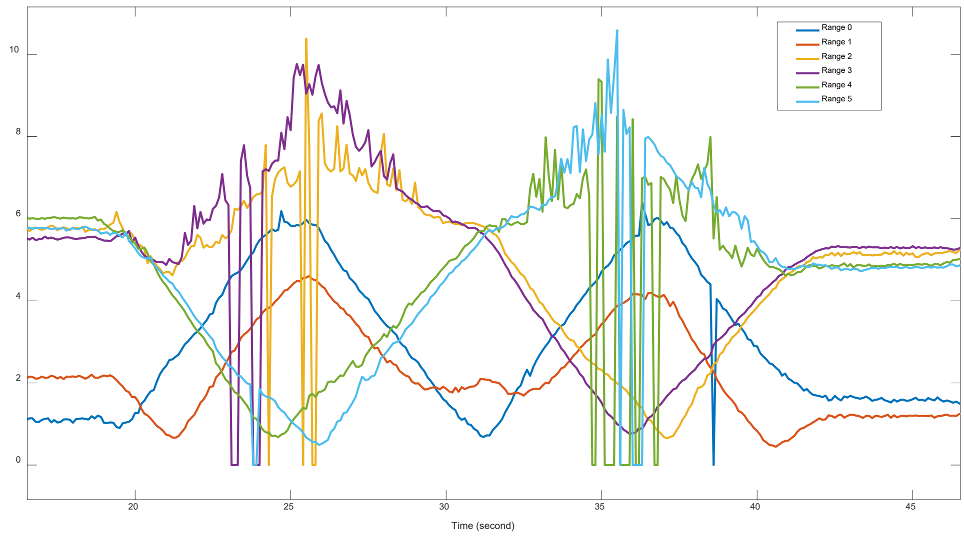
1 Another Example for Sensor Anomaly

From indoor tests, we found that the type of anomaly in Fig. 2 is frequent in NLOS (non-light-of-sight) environments. Below is an example.





As we can see, when the tag is in the area surrounded by A0-A1-A4-A5, the anchors 2 and 3 are NLOS anchors; while when in the area surrounded by A0-A1-A2-A3, the anchors 4 and 5 are NLOS anchors. When we follow the illustrated trajectory, we have the ranges as shown below:



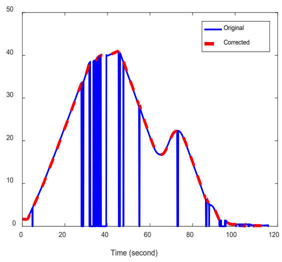
As we can see, when the tag is in the area surrounded by A0-A1-A4-A5, i.e., time from 20s to 32s, the anchors 2 and 3 are suffering from the same anomalies as in Fig. 2. When the tag is in the area surrounded by A0-A1-A2-A3, i.e., time from 32s to 40s, the anchors 4 and 5 are suffering from the same anomalies as in Fig. 2. In this indoor environment, we can know that the anomalies are due to NLOS sensing conditions (i.e., signal obstructions). But in the previous outdoor environment (Fig. 2), there is no signal obstruction and the anomaly still shows.

Since this type of anomaly is much easier to understand from Fig. 2 (because of figure clearness compared to the example above) and there are no essential differences, we do not mention this indoor test results in the paper to control paper length.

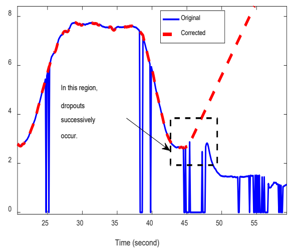
The source data collected in this indoor test field is available online at GitHub. See the sub-folder "[9] Data - Supplementary" in the folder "Codes and Data".

2 If the abnormality (especially continuous anomalies) occurs when the real measurement suddenly changes, can the algorithm cope with it?

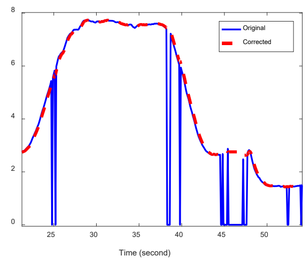
This is hard to answer because it depends a lot on how suddenly the real measurement changes, how abnormal the uncertainty is, and what parameters are used in the algorithm. We do some other experiments below to intuitively explain this point. In (a), the algorithm works well. In (b), the algorithm with K=3 fails for successive dropouts during times around 45s~47.5s. The data in (c) is the same as that in (b). But in (c) the algorithm with K=2 can work well. We confess this is a drawback (maybe a potential danger) of the proposed method. But at present, the authors have no idea to adaptively adjust parameters to address this problem. We expect improvements for the proposed algorithm in the future.



(a) ***Can*** handle

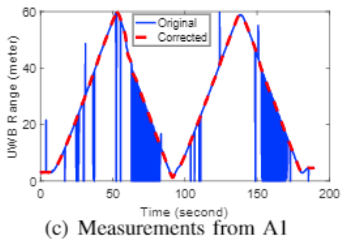


(b) ***Cannot*** handle

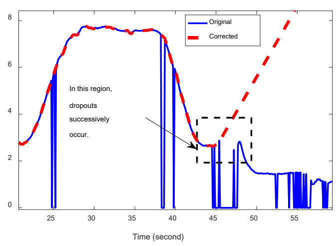


(c) ***Can*** handle (b) with different parameter

Also, as we can see from Fig. 8 (c) [also available below, the left], the proposed method is also efficient for the case of bursts of dropouts sometimes, e.g., when the range time series changes slowly. See, for example, times from 150s~180s. But experiments show that if the bursts of dropouts happen when the target is quickly maneuvering, the proposed method can no longer handle (e.g., see the right figure below). Because we need to use some useful information of range measurements in the past to predict the future's range. But when the information in the past is not reliable as well (e.g., bursts of dropouts), the prediction performance for the future is limited. However, for the case in the right figure below, we think that it challenges all the existing literature. Because in the time window 45s~47.5s, no one knows what will happen in the future. And, from 47.5s on, when measurements are again available, we can no longer judge whether the newly collected measurements are believable or not (but if believable, reset and restart the algorithm). So this is another drawback of the proposed method. We are happy to see possible improvements in the future. Please see also the Section IV (title: Conclusion) in the right column of page 11 (highlighted in blue).



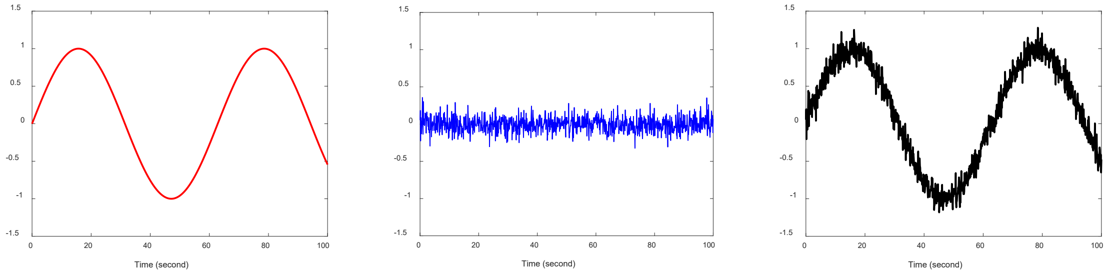
(a) Can handle



(b) ***Cannot*** handle

3 What's the rationale of the model (1)?

Now we discuss the model (1), i.e., x(t) = r(t) + s(t). Since r(t) is not guaranteed to be smooth, we used a polynomial f(t) to approximate it. By the Weierstrass approximation theorem, r(t) and f(t) could be close enough for any t. This approximation is for the convenience of theoretical analysis, but it does not cause significant precision problem. In our model (1), f(t) is the true value of range measurement, which is deterministic at any time t because the range between a receiver and an anchor is deterministic although we do not know the true value. See Figure (a) below. For a real sensor, more or less, it must have measurement noises which is usually stationary white Gaussian. See Figure (b) below. Therefore, the really measured range measurements from a sensor may be like Figure (c), i.e., the true values in (a) ***plus*** the noises in (b). This gives the rationale of the model (1).



(a) True information r(t) (b) Noise s(t) (c) Integrated range x(t)

As we can see from the Figure (c), it is actually a non-stationary (i.e., the mean function is not constant) stochastic process. Its mean function is given in Figure (a). That is, the expectation of x(t) is given by f(t). Since f(t) is also a time function, we denote this function as the **mean function** of x(t). For shortness, people usually call f(t) the **mean** of x(t), discarding the word "function". Since the definition of the mean (function) of a stochastic process is common in any stochastic process book, we do not repeat it in this paper.

4 What's the rationale behind the model (12)? The noise process W(n) affects x(n+1) in a correlated way. Why is it reasonable?

To use W(n) as the process noise (i.e., modeling error) of (12) is actually a common trick in modeling a linear Gauss-Markov system [1,2,3], which is suitable to be handled by the Kalman filter. For example, recall the Constant-Acceleration (CA) model in target tracking [4], in which the CA model is used to model the maneuvering dynamics of the moving target, and a noise process W(n) is used to represent the modeling errors (between the true maneuvering dynamics and the CA-modelled maneuvering dynamics). The authors think that this fact is well-known for readers who know the fundamentals of linear-system modeling [1,2], and the background and basics of the Kalman filter [2,3]. Because when readers who are not familiar with our field try to use the Algorithm 1 in the future, he/she will need to read the literature of the Kalman filter (e.g., 2, 3), through which this idea of using W(n) is easy to follow. So, we think maybe there is no need to discuss more on W(n) in this paper.

1 Kitagawa G., Gersch W. (1996) *Linear Gaussian State Space Modeling*. In: Smoothness Priors Analysis of Time Series. Lecture Notes in Statistics, vol 116. Springer, New York, NY. https://doi.org/10.1007/978-1-4612-0761-0_5 . See also references therein.
2 Kalman, R. (1961). New results in linear prediction and filtering theory. Trans. AMSE. J. Basic Eng., 83, 95-108.
3 Simon, D. (2006). Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons. See Chapter 5.
4 Li, X. R., & Jilkov, V. P. (2003). Survey of maneuvering target tracking. Part I. Dynamic models. IEEE Transactions on Aerospace and Electronic Systems, 39(4), 1333-1364.

5 Why the authors did not record the time stamps of dropouts and try to recover these locations?

Yes, it is also possible to record the discrete time stamps at which the message packages are missing. But we still need to find some reasonable values to replace the missing range values. Therefore, the prediction for a range time series is unavoidable. For this reason, in this paper, we directly treated the missing values as zeroes and designed the prediction-and-replace mechanism to recover the missing ranges. As we can see, if we use our method, whether to record the time stamps at which packages are missing becomes not necessary. Therefore, we did not record those time stamps.

On the other hand, the designed method is also used for outlier detection and correction. But the time stamps for outliers are un-observable. This means that our method is workable for not only outliers but also dropouts.

6 How to obtain (9)?

(9) can be directly obtained from (6) and (8) after some straightforward calculus and algebraic manipulations, especially, by taking derivatives.
In detail, (6) gives

$$f(n+1) = f(n)\cdot 1 + \frac{f^{(1)}(n)}{1!}T + \cdots + \frac{f^{(K-1)}(n)}{(K-1)!}T^{(K-1)} + \frac{f^{(K)}(n)}{K!}T^{K}$$

the derivative of above admits

$$f^{(1)}(n+1) = f(n)\cdot 0 + f^{(1)}(n)\cdot 1 + \frac{f^{(2)}(n)}{1!}T + \cdots + \frac{f^{(K)}(n)}{(K-1)!}T^{(K-1)} + \frac{f^{(K+1)}(n)}{K!}T^{K}$$

the derivative of above admits

$$f^{(2)}(n+1) = f(n)\cdot 0 + f^{(1)}(n)\cdot 0 + f^{(2)}(n)\cdot 1 + \cdots + \frac{f^{(K)}(n)}{(K-2)!}T^{(K-2)} + \frac{f^{(K+1)}(n)}{(K-1)!}T^{(K-1)}$$

…
…
…

Since we are truncating the polynomial at the order of K, we have $f^{(K+1)}(n), f^{(K+2)}(n), \ldots$equal to zeroes. So, all the blue terms disappear. We write the above results in compact form, we obtain (9). The entries of the matrix in (9) have been highlighted in red in above derivations.

7 Why the previously estimated position has not been considered?

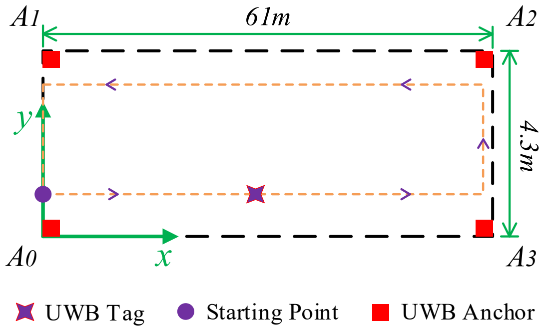
Question Explanation: The proposed approach makes sense if the localization algorithm resets its estimate at each time step. Nevertheless, usually, the Kalman filtering solutions are employed, gathering all information from the past. The information used to correct the measurement is the same as that used to get the position estimate. There is clearly a problem in using twice the same information when the corrected measurement is fed in the Kalman filter used to track the moving object.

Yes, the information used to correct the measurement is the same as that used to get the position estimate. Before we formally study this issue, we first make a short note. This paper is concerned with the target positioning problem, not the target tracking problem. In fact, for a moving target, the positioning problem is slightly different from the tracking problem. The former is a metrological problem (so related papers usually appear on, e.g., the IEEE Transactions on Instrumentation and Measurement [1]), while the latter is an automation problem (so related papers always appear on, e.g., the IEEE Transactions on Aerospace and Electronic Systems [2]). In the context of target positioning, the efforts are made to estimate the position of the target at the current time instant, without considering the positions in the past. In the context of target tracking, the efforts try to find the relation among the positions in the past and at present. Usually, we consider the target tracking problem because the directly measured positions are noises contaminated (or corrupted), not reliable enough. So, we need to use the target tracking methods (e.g., the CV, CA, Singer model [2]) to further improve the position-estimation accuracy and precision.

So, motivated by this question, the authors realize that here are two different philosophies about position-estimation for a moving target: 1) Directly improve the positioning accuracy (i.e., measurement accuracy) so that there is no need to do further processing like target tracking; 2) Obtain the position measurements first with raw range measurements (no matter whether they are reliable or not), then use the proper target tracking methods (which also takes into considerations the positions in the past) to further improve the accuracy and precision of the estimated positions.

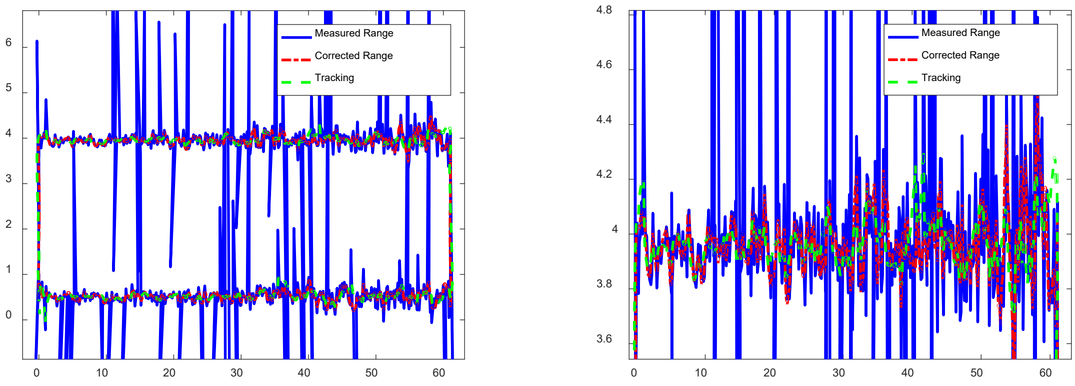
At present, the authors are not sure whether the two ideas are theoretically equivalent (because they indeed share the same information for the same position-estimation problem), or which one is theoretically better. Maybe this is a new good research topic to be further studied.

But, practically, we can design some simulated experiments to compare the MSE (mean square error) of position-estimation using the two different philosophies. Since we need to calculate the MSE, we must do simulated experiments rather than real experiments. We still use the following settings:



But this time, the trajectory is simulated with a constant moving speed of 1.5m/s, not a real trajectory. Because only in this way can we obtain the true position series against time to calculate MSE.

From the experiments results (see figures below), we found that any one of the two philosophies ***cannot outperform*** the other, depending a lot on parameters and tracking methods used. For some values of the parameters (i.e., K, Q and Δ), the first works better than the second; for some other values, the second works better than the first. In the figures below, "Measured Range" means that the trajectory is directly calculated with the raw un-corrected ranges; "Corrected Range" means that the trajectory is calculated with the corrected ranges (Algorithm 1); "Tracking" means that the trajectory is calculated by a tracking method with the raw un-corrected ranges. The tracking method used is the Constant-Acceleration (CA) model, and the tracking filter is the particle filter. In the figures below, the MSE of "Corrected Range" method is 3.3617m^2, and of "Tracking" method is 3.7792 m^2. For the technical details of the Constant-Acceleration model and the particle filter, they can be checked out from google with those keywords. Since they are common knowledge, we omit the details here.



(a) Results

(b) Closeup of (a)

But the authors still suggest the first philosophy, i.e., correcting the ranges first and then doing positioning (during which the target tracking algorithms are no longer needed). Because it helps lower the calculation burden of the remote server:

- For the first philosophy, the range correction step could be done in anchors, and only the positioning step is done in the remote server. If we use the pseudo-TDOA method for range-based positioning in Ref. 1, the positioning step only needs to solve a linear algebraic equations, which is very easy to calculate;
- But if we choose to use the target tracking method (i.e., the second philosophy), the anchors only collect the ranges and directly send them to the remote server to do further processing. Suppose we use the Constant-Acceleration (CA) Model as the target's maneuvering dynamics. Since the range measurement equations are nonlinear, in order to guarantee the tracking accuracy, the particle filtering must be used to do this nonlinear filtering problem. As we all

known, the particle filtering requires a lot of computation resources.

As we can see, in this case, the first philosophy is much more computationally efficient. Because the complexity of the range-correction algorithm (i.e., Algorithm 1) is $O(D^3)$ [since the complexity of the Kalman filter is $O(D^3)$. This fact is well-known for the Kalman filter] where D is the dimension of the state vector $\mathbf{X}(n)$ in (12). Since K usually equals to 3 or 4 (so D=4 or 5), i.e., very small, the calculation burden for anchors is omittable.

Since this paper is mainly concerned with the positioning problem, we do not include the discussions above in the main manuscript to save paper space.

1 S. Cao, X. Chen, X. Zhang, and X. Chen, "Combined weighted method for TDOA-based localization," IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 5, pp. 1962–1971, 2019

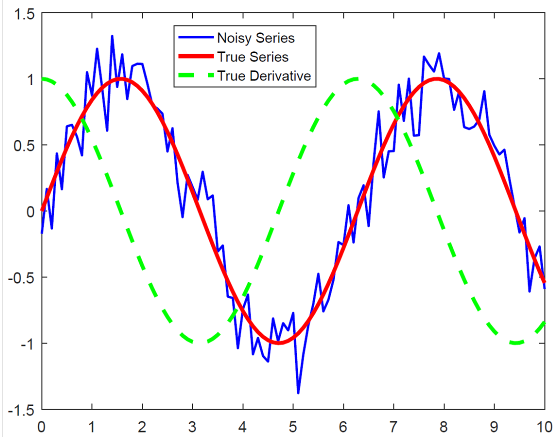
2 Li, X. R., & Jilkov, V. P. (2003). Survey of maneuvering target tracking. Part I. Dynamic models. IEEE Transactions on Aerospace and Electronic Systems, 39(4), 1333-1364.

8 Why the difference method is not suitable for estimating the derivative of a time series?

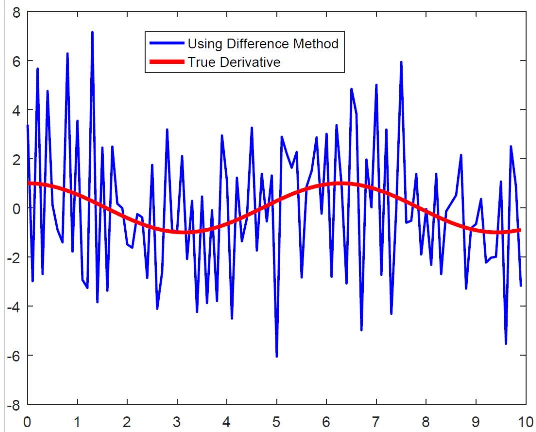
Because the noise contained in the time series would be amplified by the difference operator. Nevertheless, if the time series ***only*** contains the true information, ***without*** any noises (or significant noises), the difference method would be suitable.

For an intuitive understanding, please refer to the subsection "*4.1. A Simple Nonsmooth Example*" (pages 4-5) of Ref. 1, in which an illustration experiment is conducted. But we should mention that the designed numerical differentiation (ND) method in Ref. 1 is not applicable for our problem because ND works only for block data, not a time series. That is, ND is an offline method which is only applicable for *post hoc* signal analysis in the case that a segment of signal has already been collected.

For another intuitive understanding, see the figures below.



(a) Time Series and its true value and true derivative



(b) Using difference method to estimate the derivative

As we can see, the difference operator is indeed not reliable to estimate the derivative of a noised time series.

This fact is easy to be checked out from the frequency response [2] of the difference operator. Let

$$y(n) := [x(n) - x(n-1)]/T.$$

We have the transfer function of this linear system as

$$H(z) = [1 - z^{-1}]/T,$$

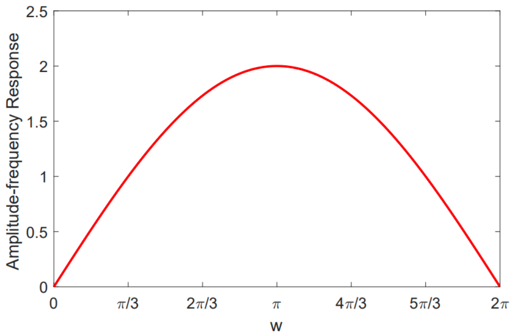
whose frequency response is

$$H(e^{jw}) = [1 - e^{-jw}]/T,$$

admitting the amplitude-frequency response as

$$|H(e^{jw})| = |[1 - e^{-jw}]/T| = [\sqrt{2 - 2cos(w)}]/T,$$

i.e.,



Note that in the digital frequency domain [2], π stands for the high-frequency, and 0 and 2π for the low-frequency. Therefore, the high-frequency components (n.b., usually high-frequency components are corresponding to noises) have been significantly amplified (i.e., amplitude-frequency responses are larger than 1).

This fact is common and well-known for researchers in signal processing community.

1 R. Chartrand, "Numerical differentiation of noisy, nonsmooth data," ISRN Applied Mathematics, vol. 2011, 2011.
2 Diniz, P. S., Da Silva, E. A., and Netto, S. L. (2010). Digital signal processing: system analysis and design. Cambridge University Press.