

A Fast Line-Fitting Algorithm under Bounded Errors

Journal Title
XX(X):1–10
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/

SAGE

Ahmet Agaoglu¹ and Namik Ciblak¹

Abstract

Line fitting is widely applied across various fields to model relationships between variables, often in contexts where measurement error bounds are associated with sensor accuracy. These error bounds define a feasible parameter set (FPS), represented as a convex polygon in parameter space. This study introduces the Corridor Method to efficiently and exactly describe the FPS and proposes using the polygon's centroid as an optimal estimate when no distributional assumptions are available.

In a second scenario, where errors are bounded and assumed to follow a normal distribution, a common occurrence in practice, finding the most likely line within the FPS becomes a convex quadratic optimization problem, typically solved by computationally intensive iterative methods. Here, we expand the solution by leveraging the geometry of the likelihood function, which transforms the convex quadratic optimization problem into a sequence of one-dimensional optimizations with analytical solutions along the polygon edges. This approach provides an exact and non-iterative solution.

In both scenarios, extensive testing across diverse error distributions and sample sizes demonstrates that the proposed methods provide fast and accurate estimates, outperforming Ordinary Least Squares, Constrained Linear Least Squares, and the Regularized Chebyshev Center. A MATLAB implementation of the proposed method is publicly available, providing a practical tool for precise and efficient line fitting in bounded-error settings.

Keywords

Line fitting, Bounded error, Regression, Convex optimization

Introduction

Linear regression remains a cornerstone of statistical analysis and data modeling, with applications spanning virtually every quantitative discipline. Among its variants, simple linear regression, or line fitting, represents the most fundamental approach to characterizing relationships between two variables. The ubiquity of line fitting in modern applications, from real-time signal processing to high-precision manufacturing, necessitates methods that are both computationally efficient and statistically robust.

Line Fitting Methods: Evolution and Current State

The landscape of line fitting methods has evolved considerably to address different error structures and modeling assumptions. Ordinary Least Squares (OLS) minimizes the sum of squared vertical residuals and remains the most widely used approach when errors are confined to the response variable and follow a normal distribution. However, OLS demonstrates well-documented vulnerability to outliers and model misspecification, limiting its applicability in practical scenarios where data quality cannot be guaranteed.

When errors affect both variables, errors-in-variables (EIV) models become necessary. Total Least Squares (TLS), also known as orthogonal regression, addresses this scenario by minimizing orthogonal distances to the fitted line rather than vertical distances. Recent theoretical advances by Aishima [Aishima \(2024\)](#) have extended TLS consistency

properties to constrained problems with specific row and column structures, while McKittrick et al. [McKittrick and Christy \(2024\)](#) have examined TLS performance in high-dimensional settings with heterogeneous noise variances, revealing significant biases when standard assumptions are violated. Deming regression, a special case of TLS where the ratio of error variances is known, offers a maximum likelihood solution under normally distributed errors.

The challenge of outliers has motivated extensive research in robust regression methods. Modern approaches include Huber regression, which employs a hybrid loss function combining squared and absolute terms [Tong \(2023\)](#); RANSAC (Random Sample Consensus), which identifies inlier subsets through iterative sampling [Martínez-Otzeta et al. \(2023\)](#); and least median of squares methods with breakdown points up to 50% [Rousseeuw and Leroy \(1987\)](#).

Deep learning approaches have recently entered the line fitting domain, particularly for image processing tasks. Boyer et al. [Boyer et al. \(2024\)](#) introduced LineFit, which combines predicted image gradient maps with geometric optimization to fit line segments in computer vision applications, demonstrating competitive performance against traditional detectors on various benchmarks. Similarly, Teplyakov et

¹Yeditepe University, Turkey

Corresponding author:

Ahmet Agaoglu, Department of Mechanical Engineering, Yeditepe University, Kayisdagi Cad. 326A Atasehir, Istanbul, 34755, Turkey
Email: ahmet.agaoglu@yeditepe.edu.tr

al. [Teplyakov et al. \(2022\)](#) proposed LSDNet, a deep-learning-augmented variant of the classical LSD detector that enhances the accuracy and robustness of line segment extraction in images.

Bounded Error Estimation and Set-Membership Approaches

In many practical applications, measurement errors arise primarily from sensor limitations with known specifications, making it unrealistic to assume unbounded error distributions. The Unknown-But-Bounded (UBB) error model provides a more appropriate framework for such scenarios, where errors are constrained within intervals defined by sensor accuracy or expert knowledge. This assumption naturally leads to set-membership estimation, which seeks to characterize the feasible parameter set (FPS)—the set of all parameter values consistent with the observations and error bounds.

The geometric representation of the FPS presents computational challenges. While exact vertex enumeration provides complete characterization, its computational complexity scales poorly with problem size [Avis et al. \(1997\)](#). Consequently, much research has focused on efficient approximations using simpler geometric shapes.

Beck and Eldar [Beck and Eldar \(2007\)](#) proposed the Regularized Chebyshev Center (RCC) approach, which determines the center of the smallest radius ball enclosing an approximation of the FPS formed by the intersection of two ellipses. While computationally tractable, the RCC may produce estimates outside the true FPS depending on the set's geometry, and the ellipsoidal approximation introduces additional conservatism.

Casini et al. [Casini et al. \(2024\)](#) advanced methods for discrete-time linear systems with binary sensor measurements, while Wang et al. [Wang et al. \(2023\)](#) developed zonotopic set-membership estimation for linear time-varying descriptor systems with optimized observer parameters. Tang et al. [Tang et al. \(2024\)](#) revisited minimum enclosing ellipsoids for set-membership estimation, introducing computational enhancements including constraint pruning and generalized relaxed Chebyshev centers that make the framework practical for modern control and perception problems. Zhang et al. [Zhang et al. \(2024\)](#) investigated fault detection and isolation for switched linear parameter-varying systems using zonotopic methods.

Zonotopic approaches offer an attractive compromise, allowing exact computation of linear maps and Minkowski sums while maintaining reasonable computational complexity (Alamo et al. [Alamo et al. \(2005\)](#)). Recent comparisons by Rego et al. [Rego et al. \(2024\)](#) across interval arithmetic, ellipsoids, zonotopes, constrained zonotopes, polytopes, and constrained convex generators reveal varying trade-offs depending on the specific application context—state estimation, fault detection, or collision avoidance.

Interval regression methods, which specify explicit upper and lower bounds on observations, provide another perspective on bounded error problems [Billard and Diday \(2000, 2002\)](#); [Souza et al. \(2017\)](#); [de A. Lima Neto and de A.T. de Carvalho \(2010\)](#); [Wang et al. \(2012\)](#). While effective in certain applications, interval regression

often assumes errors in both variables and can produce broad solution spaces that reduce precision. When only the response variable contains bounded error, interval regression reduces to OLS, lacking the capacity to directly incorporate bounded constraints.

When errors follow a normal distribution within known bounds, parameter estimation becomes a constrained linear least squares (CLLS) problem which is a convex quadratic programming challenge. Standard solution approaches employ iterative solvers such as interior-point methods and active-set algorithms. While effective, these methods incur substantial computational costs, particularly for large datasets, and provide approximate solutions dependent on optimization tolerances [Cimini and Bemporad \(2017\)](#); [Rodriguez-Lujan et al. \(2010\)](#).

Real-Time Applications and Performance Requirements

Modern applications demand line fitting methods that combine accuracy with computational efficiency. In power systems, line fitting enables real-time zero-crossing detection in sinusoidal signals, critical for synchronizing operations in systems with dynamic frequencies [Patil and Ghorai \(2016\)](#). The rapid transients in high-speed power electronics require processing times on the order of microseconds, making iterative optimization methods impractical.

Similarly, additive manufacturing systems employ line fitting for real-time anomaly detection within video-imaging data, where swift identification of quality control issues is essential for maintaining production standards [Yan et al. \(2020\)](#). Ultra-precise positioning systems in nanotechnology rely on line fitting to define trajectories with sub-micrometer accuracy, fundamental to reliable nano-manufacturing [Pérez et al. \(2019\)](#).

Like many real-world applications, these systems share common characteristics. They involve time-based data collection, where timing precision is high, but measurement accuracy is limited by sensor specifications. In such cases, the predictor variable can be considered error-free, while the response variable has known bounds. Moreover, the need for both high accuracy and computational efficiency precludes iterative optimization methods. To contextualize these developments, Table 1 provides a comprehensive comparison of existing methods in relation to these critical performance requirements.

Research Gap and Contributions

The analysis presented in Table 1 reveals a critical gap in the current methodological landscape. Methods that ignore error bounds sacrifice accuracy in bounded error scenarios, while methods that acknowledge bounds either sacrifice exactness through geometric approximations or incur prohibitive computational costs through iterative optimization. The performance requirements of modern applications, combined with the limitations of existing methods, motivate the need for a new approach that delivers exact characterization of the feasible parameter set with computational efficiency suitable for real-time implementation. This study addresses this need through two main contributions.

Table 1. Comparison of line fitting methods for bounded error scenarios.

Method	FPS	Exactness	Solution Approach	Distributional Assumption	Primary Limitations
OLS	Not considered	Exact for unbounded	Closed-form Fast	Normal, unbounded	Ignores error bound
TLS/Deming	Not considered	Exact for unbounded	Closed-form Fast	Normal, unbounded	Assumes errors in both variables
Robust	Not considered	Approximate	Iterative High cost	Arbitrary with outliers	High computational cost; no bound incorporation
Interval Regression	Interval-based	Approximate	Closed-form Fast	None, bounded	Broad solution space; reduces to OLS for single-variable error
Approximation Methods	Ellipsoid, zonotope	Approximate	Iterative Moderate cost	None	Estimate may lie outside true FPS
CLLS	Exact polygon (implicit)	Exact	Iterative High cost	Normal, bounded	High computational cost
Proposed Method	Exact polygon	Exact	Closed-form Fast	None or Normal, bounded	Requires bounded error specification

First, we introduce a fast algorithm, the **Corridor Method**, to delineate the FPS exactly by leveraging its boundary geometry. A preliminary geometric formulation of this idea appeared in the author's thesis [Agaoglu \(2023\)](#). In scenarios where no distributional information about the error is available, we propose using the centroid of the FPS as a robust estimator of the regression parameters, as it lies within the FPS due to the convexity of the set. This approach is validated across synthetic datasets incorporating diverse error distributions and observation counts, consistently outperforming OLS, Constrained Linear Least Squares (CLLS), and the RCC, except when errors follow a strict normal distribution.

Second, we extend the Corridor Method to identify the most likely estimate within the FPS under the assumption of normally distributed errors. This extension leverages the geometry of the likelihood function, transforming the quadratic optimization problem into a sequence of one-dimensional optimizations with analytical solutions along the FPS boundary. This exact, non-iterative approach reduces computational demands while enhancing precision. Experimental results show that this method achieves both high computational efficiency and accuracy, consistently outperforming the same set of competitor methods.

Corridor Method to Exactly Describe Feasible Parameter Set

Consider the following simple linear regression model.

$$y_i = \beta_0^* + \beta_1^* x_i + \varepsilon_i \quad (1)$$

where y_i and x_i are values of response and predictor variables, respectively, measured in the i^{th} observation ($i = 1, \dots, N$). $\beta_0^*, \beta_1^* \in \mathbb{R}$ are the true model parameters to be estimated and $\varepsilon_i \in [\varepsilon_i^L, \varepsilon_i^U]$ denotes the bounds of the error. In each observation the true response is constrained by two inequalities.

$$\beta_0^* + \beta_1^* x_i \geq y_i - \varepsilon_i^U = L_i \quad (2)$$

$$\beta_0^* + \beta_1^* x_i \leq y_i - \varepsilon_i^L = U_i \quad (3)$$

where L_i and U_i represent the lower and upper limits of the true response.

If multiple observations are considered, then L_i and U_i form an uncertainty corridor as shown in Figure 1. Further, $\beta_0^* + \beta_1^* x_i$ now represents the true line which must be in the corridor, assuming that error bounds and the measurements conform to the model. In this way, the problem becomes a membership problem of finding full solution set containing all feasible pairs of (β_0, β_1) , namely all feasible lines in the corridor, as described in the following definition.

Definition 1. The solution set of all feasible lines (FPS) is given by

$$S = \{\beta_0, \beta_1 \in \mathbb{R} \mid L_i \leq \beta_0 + \beta_1 x_i \leq U_i, \forall i\} \quad (4)$$

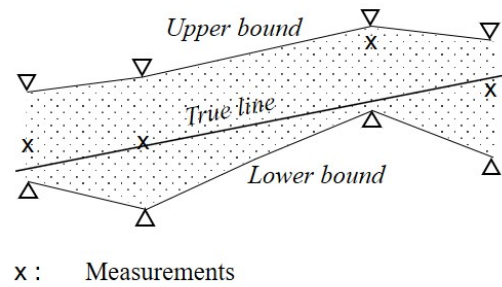


Figure 1. The uncertainty corridor.

Assumptions:

1. $x_i \neq x_j$ for $i \neq j$
2. $\varepsilon_i^U - \varepsilon_i^L > 0$.
3. The image of S in the parameter space is a non-empty, 2D region.

These assumptions and Equation 4 force β_0 and β_1 to be bounded. Further, it is well known that the solution set for a system of linear inequalities in parameter space always forms a convex polytopic region [Bosse et al. \(2005\)](#). Hence, the following lemma.

Lemma 1. S is a bounded, convex polygonal region in the parameter space.

Next, the geometry of S is analyzed to construct an algorithm that precisely characterizes S .

Definition 2. Define the following subsets of S :

1. S_F : The set of all lines through the corridor that do not pass through any lower or upper bound points (free lines).
2. S_L : The set of all lines through the corridor that pass through at least one lower, but none of the upper bound points.
3. S_U : The set of all lines through the corridor that pass through at least one upper, but none of the lower bound points.
4. S_{UL} : The set of all lines through the corridor that pass through at least one lower and at least one upper bound points.

By this definition, $S = S_F \cup S_L \cup S_U \cup S_{UL}$ and $S_i \cap S_j = \emptyset$ (mutually disjoint subsets).

Corollary 1. S_F is an open set. The boundary of S_F , denoted by $\partial S = S_L \cup S_U \cup S_{UL}$, defines the limit points of S_F .

Proof. Any line in S_F can be shifted downward (or upward) until it contacts a lower (or upper) boundary point, thereby becoming an element of S_L (or S_U). Similarly, rotations of lines in S_F eventually intersect both lower and upper boundaries, resulting in membership in S_{UL} . These transformations can be regarded as infinite sequences that start in one set yet end, in the limit, in another that is disjoint. In set theoretical terms, the end points are the limit (accumulation) points of the original sets. Since S_F does not contain any of its limit points it is an open set. S_L , S_U , and S_{UL} contain the limit points of S_F . Therefore, their union forms the boundary of S_F .

Since S is a convex polygonal region in the parameter space, it can be described by its vertices which are the elements of S_L , S_U , and S_{UL} by Corollary 1. Let the space of (x, y) be denoted by \mathbb{M} (measurement space) and that of (β_0, β_1) by \mathbb{P} (parameter space), both of which are \mathbb{R}^2 spaces.

Consider continuous rotations of a line about a fixed point (x_P, y_P) in \mathbb{M} . The resulting set of lines is called a pencil of lines, or a subset of it (partial pencil), with a concurrent point (x_P, y_P) . This leads to $\beta_0 = y_P - x_P \beta_1$, a line in \mathbb{P} . Figure 2 illustrates the mapping of a generic partial pencil of lines.

A linear map between \mathbb{M} and \mathbb{P}

$$y = \beta_0 + \beta_1 x \quad (5)$$

can be interpreted in two ways that are duals of one another:

1. Given a point (β_0, β_1) in \mathbb{P} , Equation 5 describes a line in \mathbb{M} .
2. Given a point (x, y) in \mathbb{M} , it describes a line in \mathbb{P} ($\beta_0 = y - x\beta_1$).

Theorem 1. Any line in the corridor that passes through at least two bound points is a vertex of S .

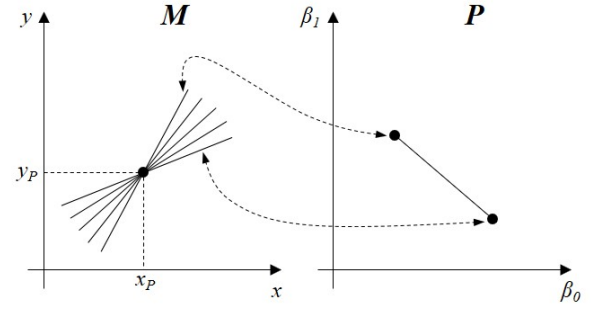


Figure 2. A constrained rotation of lines in \mathbb{M} and its image in \mathbb{P} .

Proof. Let $l \in S_L$ be a line passing through two lower bound points, P and Q . In the parameter space \mathbb{P} , this line corresponds to a single point, say Λ . Rotating l about P while freeing Q yields a line $l' \in S_L$ that only passes through P , representing a line segment in \mathbb{P} with endpoint Λ .

Similarly, if l is rotated about Q , fixing Q while freeing P , a different line segment in \mathbb{P} is generated, converging again at Λ but with a different dependency between β_0 and β_1 . Thus, l forms the join of two distinct segments in \mathbb{P} at Λ , which is a vertex (see Figure 3).

The same reasoning holds for lines passing through at least two points in S_U or S_{UL} , establishing that such lines form vertices on the boundary of S .

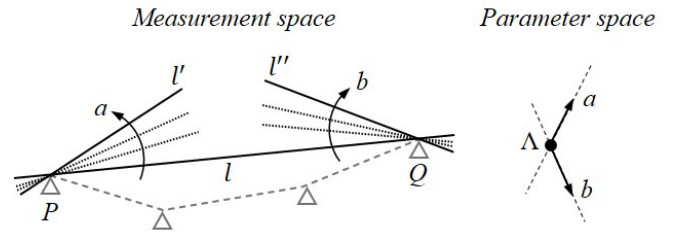


Figure 3. Formation of a vertex in \mathbb{P} by a line in \mathbb{M} contacting two points in the corridor.

Consider a line $l \in S_L$ that passes through a lower bound point P_0 . Image of l corresponds to a point in \mathbb{P} , say B . Since the boundary is a polygon, one can trace the boundary by starting from B and returning back to B , in one full cycle. A visual illustration of this cycle is depicted in Figure 4.

One rotates l by keeping P_0 fixed and **decreasing the slope**. This transformation traces the image of S_L in \mathbb{P} . The rotation continues until either of two things happens: i) the rotated line contacts one or more lower bound points, say P_1 , without contacting the upper bound, marking a vertex in S_L or, ii) the line contacts one or more upper bound points, say P_2 , giving a vertex in S_{UL} (Theorem 1).

In both cases no further rotation in the same sense is possible. If case (i) occurs, then one continues the rotation in the same sense by freeing P_0 and pivoting on P_1 . Thus, tracing the image of S_L in \mathbb{P} continues. Again, either a vertex in S_L (case i) or S_{UL} (case ii) is found. The process continues until a vertex in S_{UL} is reached eventually, say l_{\min} .

After l_{\min} is reached, one can continue rotation by freeing the lower contact point and pivoting on the upper contact

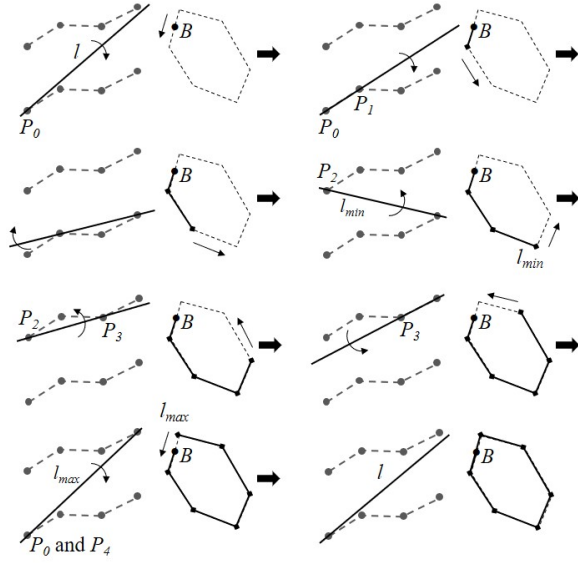


Figure 4. Illustration of the cycle in which the vertices are determined sequentially.

point P_2 in the opposite direction, namely **increasing the slope**. This makes l_{\min} as the vertex with the minimum slope. Rotations about P_2 trace the image of S_U in \mathbb{P} that continue until either of two things happens: a) the rotated line contacts one or more upper bound points P_3 , without contacting the lower bound, marking a vertex in S_U or, b) the rotated line contacts one or more lower bound points P_4 , giving $l_{\max} \in S_{UL}$, the vertex with the maximum slope. The rest is similar.

The process is terminated when P_0 becomes the pivot again. In this way, the vertices are determined sequentially as follows.

vertices in $S_L \rightarrow l_{\min} \rightarrow$ vertices in $S_U \rightarrow l_{\max} \rightarrow$ vertices in S_L (6)

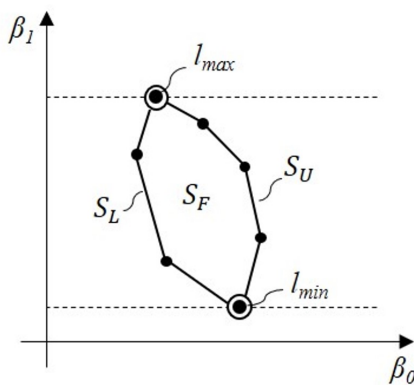


Figure 5. Structure of S and partition of ∂S by l_{\min} and l_{\max} .

Corollary 2. $S_{UL} = \{l_{\min}, l_{\max}\}$ and these vertices partition ∂S into four disjoint components: $\{l_{\min}\}$, $\{l_{\max}\}$, S_L , S_U , the latter two of which are infinite and connected sets (See Figure 5).

While this method enables the exact determination of S , the regression problem necessitates selecting a single

representative estimate from the infinitely many possibilities within S .

In the absence of distributional information, all members of S are equally likely to represent the true line. This lack of preference among the solutions within S allows for different approaches in determining the best estimate. A popular choice is the Chebyshev center, which is defined as the center of the smallest radius ball that encloses the S [Beck and Eldar \(2007\)](#). However, the Chebyshev center may reside outside S , depending on the geometry of the set in the parameter space. In contrast, the proposed method exactly delineates S , ensuring that the centroid, which lies within S due to its convexity, serves as a robust and consistent estimate. Thus, we propose using the centroid of S as the optimal estimate in scenarios where no distributional information is available.

An Algorithm to Enumerate the Vertices

In this section we propose an algorithm based on the method outlined in Section . The algorithm expects on input the lower and upper bound data, $L_i(x_i)$ and $U_i(x_i)$, sorted with respect to $\{x_i\}$ in ascending order. The algorithm first determines vertices in S_L and S_U , then l_{\min} and l_{\max} .

The algorithm starts with (x_1, L_1) as the first pivot, say P . In order to find the other contact point, say Q , which would form a vertex together with P , the lines that pass through P and all succeeding points are created. The point Q is determined by the line with the maximum slope, since the other points would stay below this line and so are not feasible (Figure 6).

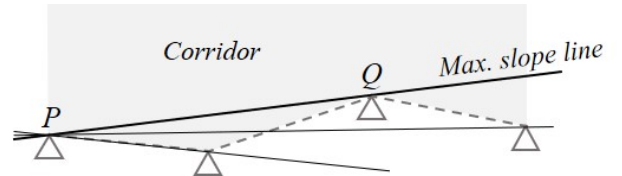


Figure 6. Determination of a vertex in S_L

Subsequently, the line passing through P and Q is tested against the upper bound. If the line is *below* the upper bound then it is a vertex in S_L . In the next iteration, Q becomes the new pivot point and the same operations continue until the index of the pivot point reaches N . This method identifies the vertices in S_L and is presented in Algorithm 1. Note that the algorithm also outputs the indices of the left and right contact points of the first and last vertices found, a and b respectively, which will later be used to determine l_{\min} and l_{\max} .

A similar algorithm, with minor adjustments, can be applied to determine the vertices in S_U . In this case, the algorithm processes the upper bound points, where Q is determined by the line with the minimum slope. The feasibility check is performed against the lower bound points. The output consists of V_U (the set of vertices in S_U), along with c and d , representing the indices of the left and right contact points of the first and last identified vertices.

Once the vertices in S_L and S_U have been identified, l_{\min} and l_{\max} are determined by the lines passing through the points (x_b, L_b) and (x_c, U_c) for l_{\min} , and (x_a, L_a) and (x_d, U_d) for l_{\max} , respectively.

Algorithm 1 Determination of vertices in S_L

```

1: Input:  $x_p, L_p, U_p$  (for  $p = 1$  to  $N$ )
2: Initialize  $V_L$  (set of vertices in  $S_L$ ),  $a$  and  $b$  as empty
3: Set pivot index  $i = 1$ 
4: while  $i < N$  do
5:   for  $j = i + 1$  to  $N$  do
6:     Calculate slope  $m_{ij} = \frac{L_j - L_i}{x_j - x_i}$  of line through
        $(x_i, L_i)$  and  $(x_j, L_j)$ 
7:   end for
8:   Find the index of maximum slope  $J = \arg \max_j (m_{ij})$ 
9:   Calculate intercept  $n_i = L_i - m_{iJ}x_i$ 
10:  if  $m_{iJ}x_k + n_i < U_k$  for  $k = 1, \dots, N$  then
11:    Append  $n_i, m_{iJ}$  to  $V_L$ 
12:    if  $a$  is empty then
13:       $a = i$   $\triangleright$  Record the first contact point
14:    end if
15:     $b = J$   $\triangleright$  Update the rightmost contact point
16:  end if
17:  Set pivot index  $i = J$ 
18: end while
19: Output:  $V_L, a, b$ 

```

Identifying the Most Probable Line within S under Normal Error Distribution

Having established the Corridor Method to precisely delineate S , we now focus on identifying the most probable line within S when errors are normally distributed. Assuming that the errors ε_i are distributed with zero mean and known variance σ^2 , $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, the likelihood function $L(\beta_0, \beta_1)$ can be used to measure the probability of the observed data given the parameters (β_0, β_1) .

$$L(\beta_0, \beta_1) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}\right) \quad (7)$$

Maximizing $L(\beta_0, \beta_1)$ is equivalent to maximizing the log-likelihood function:

$$\ln L(\beta_0, \beta_1) = -\frac{N}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_i)^2 \quad (8)$$

Ignoring constants, maximizing the log-likelihood reduces to minimizing the sum of squared residuals:

$$\min_{\beta_0, \beta_1} \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_i)^2 \quad (9)$$

This is the classical OLS, the estimates of which $(\hat{\beta}_0, \hat{\beta}_1)$ are calculated as:

$$\begin{aligned} \hat{\beta}_1 &= \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^N (x_i - \bar{x})^2}, \\ \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x}, \end{aligned} \quad (10)$$

where \bar{x} and \bar{y} are the sample means of x_i and y_i , respectively.

If $(\hat{\beta}_0, \hat{\beta}_1) \in S$, the OLS solution represents the most probable line within S , and no further calculations are necessary. When $(\hat{\beta}_0, \hat{\beta}_1) \notin S$, one needs to solve the following constrained optimization problem:

$$\min_{\beta_0, \beta_1} \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_i)^2, \quad (11)$$

$$\text{subject to } L_i \leq \beta_0 + \beta_1 x_i \leq U_i, \quad i = 1, \dots, N.$$

The problem defined by Equation 11 is a convex quadratic programming (QP) problem, more specifically a constrained linear least-squares problem. Since the objective function is strictly convex and the feasible set S is convex (Lemma 1), the problem admits a unique global minimizer $(\hat{\beta}_0, \hat{\beta}_1)$.

Iterative QP algorithms, such as interior point methods, active set methods, and gradient projection methods, provide approximate solutions with a certain level of precision based on predetermined optimization tolerances and also operate with relatively inefficient speeds. However, a non-iterative and exact solution is possible. We propose the following geometric solution to identify the optimal point $(\hat{\beta}_0, \hat{\beta}_1)$.

The contour curves of the objective function, representing equal objective function values, are ellipses on the (β_0, β_1) plane, centered at the OLS solution $(\hat{\beta}_0, \hat{\beta}_1)$. These ellipses expand outward as the objective function increases, corresponding to a decrease in likelihood (see Figure 7). The optimal solution $(\hat{\beta}_0, \hat{\beta}_1)$ is found where the smallest ellipse (corresponding to the lowest objective function value) first touches the polygon boundary, since further expansion of the ellipse would encompass points with higher objective function values. Both the ellipses and the polygon are convex, ensuring that the ellipses touch the boundary at a single point. This guarantees the uniqueness of the solution, while the existence of the solution is assured by the space-filling property of the ellipses.

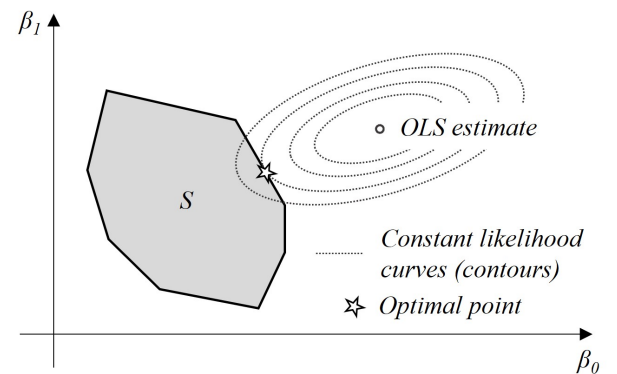


Figure 7. Feasible parameter set S with the OLS estimate, likelihood contours, and the optimal point maximizing the likelihood function within S .

To determine the optimal solution, we evaluate the minimum objective function value for each edge of the polygon, which may occur at an internal point along the edge or at one of its vertices.

For each edge connecting vertices V_k and V_{k+1} , we parameterize the line collinear to the edge as $\beta_1 = p_k \beta_0 + q_k$, where p_k and q_k are derived from the coordinates

of V_k and V_{k+1} . Substituting this expression into the objective function reduces the problem to a one-dimensional optimization along the line, with the solution given by:

$$\begin{aligned}\bar{\beta}_0 &= \frac{\sum_{i=1}^N (y_i - q_k x_i)(1 + p_k x_i)}{\sum_{i=1}^N (1 + p_k x_i)^2}, \\ \bar{\beta}_1 &= p_k \bar{\beta}_0 + q_k.\end{aligned}\quad (12)$$

If $(\bar{\beta}_0, \bar{\beta}_1)$ lies within the edge $[V_k, V_{k+1}]$, it is recorded as a candidate solution; otherwise, the vertex with the lower objective function value between V_k and V_{k+1} is recorded. This process is repeated for all edges, and the candidate with the lowest objective function value across all edges is chosen as the optimal solution $(\hat{\beta}_0, \hat{\beta}_1)$. This method is implemented in Algorithm 2.

In addition, a MATLAB implementation of the proposed method is publicly available at <https://github.com/Ahmet-Agaoglu/CMFit>. This implementation generates the vertex list in sequential order as described in Equation 6, calculates the centroid as the optimal estimate for cases where no distributional information is available, and determines the optimal point $(\hat{\beta}_0, \hat{\beta}_1)$ for scenarios where the error is assumed to follow a normal distribution.

Algorithm 2 Determination of Optimal Solution for Normally Distributed Errors

```

1: Input:  $\{V_L, l_{\min}, V_U, l_{\max}\}$   $\triangleright$  Vertices of  $S$  in sequential order
2: Initialize optimal point  $(\tilde{\beta}_0, \tilde{\beta}_1)$  as empty
3: Initialize candidate solution list  $C$  as empty
4: Calculate  $(\beta_0, \beta_1)$  (OLS estimate)  $\triangleright$  Using Equation 10
5: if  $L_i \leq \beta_0 + \beta_1 x_i \leq U_i, \quad i = 1, \dots, N$  then
6:    $(\beta_0, \beta_1) = (\beta_0, \beta_1)$   $\triangleright$  OLS estimate is within  $S$ 
7: else
8:   for each edge connecting vertices  $V_k$  and  $V_{k+1}$  do
9:     Calculate  $(\beta_0, \beta_1)$   $\triangleright$  Using Equation 12
10:    if  $(\beta_0, \beta_1)$  lies within the edge  $[V_k, V_{k+1}]$  then
11:      Append  $(\beta_0, \beta_1)$  to  $C$ 
12:    else
13:      Calculate the objective function value for  $V_k$ 
14:      and  $V_{k+1}$ 
15:      Append  $V_k$  or  $V_{k+1}$ , whichever has the lower
16:      objective function value, to  $C$ 
17:    end if
18:  end for
19:  Set  $(\tilde{\beta}_0, \tilde{\beta}_1)$  to the element in  $C$  with the minimum
20:  objective function value
21: end if
22: Output:  $(\tilde{\beta}_0, \tilde{\beta}_1)$ 

```

Synthetic Experiments

This section examines the performance of the proposed method under various experimental conditions. Two scenarios are considered: one where the error is unknown but bounded, and another where the error follows a known normal distribution with specified bounds.

Performance Under Unknown but Bounded Error

In this subsection, the proposed method is tested under the assumption that the error is unknown but bounded. The observed response vector y_n was generated using the following equation:

$$y_n = \beta_0 + \beta_1 x + \varepsilon \quad (13)$$

Both β_0 and β_1 were set to 1, defining the true line. The predictor vector x was randomly drawn from a uniform distribution in the range $[-10, 10]$. The error term ε_i was randomly generated from one of four distributions, selected to explore a broad range of real-world sensor errors.

The standard normal distribution ($\mu = 0, \sigma = 1$) represents symmetric, unbiased errors. The extreme value distribution (location = -1, scale = 2) introduces asymmetry and systematic bias with a non-zero mean. Weibull distribution (scale = 1, shape = 5) models positive-only, skewed errors. These three distributions, which have infinite support, were truncated to capture 95% of the distribution and scaled to provide error bounds $\varepsilon_i = [-1, 1]$. Finally, a uniform distribution with bounds $[-1, 1]$ was included to model evenly distributed errors.

The number of observations N was varied between 20, 50, 100, 200, and 500 to assess the performance of the proposed method across different data sizes. To ensure the reliability of the results and minimize the influence of randomness, each experiment was repeated 100 times. Finally, the competitors evaluated in the experiments are as follows:

- **Ordinary Least Squares (OLS):** OLS is one of the most widely used methods for linear regression. It minimizes the sum of squared residuals without considering any constraints or bounds on the error, making it a standard baseline for comparison.
- **Regularized Chebyshev Center (RCC):** This method Beck and Eldar (2007) approximates the FPS and selects the Chebyshev center as the estimate.
- **Constrained Linear Least Squares (CLLS):** CLLS restricts its estimates to lie within the S . It assumes that the errors follow a normal distribution, which is a reasonable assumption in many real-world scenarios, making it a practical competitor.
- **Proposed Method (Centroid of S):** Since no information about the distribution is assumed to be available, every line within S has an equal likelihood of being the true line. Therefore, the centroid of the polygon S is used as the estimate, as discussed in Section .

All methods were implemented in MATLAB R2024a Inc. (2024). For OLS, we employed the MATLAB built-in function `polyfit` The MathWorks, Inc. (2024b), while for CLLS, we used the `lsqlin` function The MathWorks, Inc. (2024a). The computational environment used in the experiments includes an Intel(R) Core(TM) i7-13700H 2.40 GHz processor and 32GB DDR4 3200MHz RAM.

Accuracy is assessed using the Mean Absolute Error (MAE), which quantifies the average magnitude of residuals. Table 2 presents the MAE values of the competitor methods for each value of N .

Table 2. Comparison of methods under different distributions with varying N values. Values are multiplied by 10^3 for better visualization. The best results are indicated in **bold**.

N	Normal				Extreme Value			
	OLS	RCC	CLLS	Ours	OLS	RCC	CLLS	Ours
20	109	143	105	121	361	367	338	260
50	72	91	65	76	355	321	292	174
100	47	57	41	49	359	264	244	126
200	34	36	29	30	355	152	144	65
500	21	18	14	15	362	95	87	38

N	Uniform				Weibull			
	OLS	RCC	CLLS	Ours	OLS	RCC	CLLS	Ours
20	141	135	115	80	580	243	182	65
50	99	87	81	37	578	147	109	40
100	71	61	56	22	576	72	55	21
200	46	36	32	9	576	51	37	15
500	27	17	15	4	574	19	14	5

In all distributions except the normal, the proposed method consistently achieves the lowest MAE values across all data sizes, underscoring its robustness across various error structures. For the normal distribution, CLLS achieves the lowest MAE values, outperforming the proposed method. This outcome is anticipated since CLLS is specifically tailored for normally distributed errors. Notably, despite being optimized for normal errors, CLLS performs better than RCC even with biased, skewed, or one-sided errors. This is likely because CLLS ensures that its estimates remain within the S , whereas RCC's approximation of S can yield estimates outside of the actual feasible set.

Furthermore, increasing the data size enhances accuracy across all methods, with the RCC, CLLS and the proposed method benefiting most consistently. Each additional observation introduces two constraints to the system, thereby incrementally shrinking the S . Consequently, the feasible region contracts with increasing observations, narrowing the solution space and elevating the likelihood of the estimated solution aligning closely with the true values. This contraction in S as N grows reflects the greater precision achieved with larger sample sizes.

Given that the proposed method performs robustly across various distributions, it is a reasonable choice to employ the centroid of S when the error distribution is unknown. Even for normally distributed errors, where CLLS has a slight advantage, the proposed method remains competitive. Its adaptability to different error structures provides a practical advantage in applications where distributional assumptions may not be possible or reliable.

To further evaluate the practical utility of each method, we compared their computational speeds by calculating the ratios of the mean processing times of OLS, RCC and CLLS relative to our proposed method, as shown in Table 3. Since processing time is primarily influenced by the sample size N rather than the error distribution type, we conducted this speed comparison using a single distribution, the normal distribution.

These speed ratios highlight a clear computational advantage for the proposed method over RCC and CLLS. RCC consistently exhibits slower performance, with processing times over 100 times greater than our method across all sample sizes N . While CLLS demonstrates better

Table 3. Processing time ratios of OLS, RCC and CLLS relative to the proposed method for different sample sizes N .

N	OLS/Ours	RCC/Ours	CLLS/Ours
20	0.82	116.2	12.2
50	0.63	145.6	37.0
100	0.56	124.1	82.4
200	0.50	133.5	287.1
500	0.42	139.5	1629.1

efficiency than RCC, it still incurs considerably higher processing times, especially as N increases, with ratios rising sharply to over 1600 times the speed of our method at $N = 500$. Although OLS is faster than our method across all sample sizes, with a time ratio decreasing from 0.82 to 0.42 as N increases, this speed advantage comes at a significant accuracy cost (see Table 2). Overall, these results illustrate the balance our approach achieves between computational speed and accuracy, making it a highly practical choice for applications where both precision and efficiency are essential.

Performance of the Proposed Method with Known Normal Distribution and Specified Bounds

In this section, we assess the performance of the method under the assumption that the error follows a known normal distribution with specified bounds. Rather than using the centroid of S , we employ the solution introduced in Section

In the experiments, errors were generated from a truncated and scaled standard normal distribution with bounds $\varepsilon_i = [-1, 1]$. The number of observations N was varied and each experiment was repeated 100 times to ensure reliable results. Table 4 presents the MAE values for each method across varying sample sizes N .

Table 4. Mean Absolute Error (MAE) values for OLS, RCC, CLLS, and the proposed method across different sample sizes N . Values are multiplied by 10^3 for better visualization. The best results are indicated in **bold**.

N	OLS	RCC	CLLS	Ours
20	114	141	100	100
50	70	92	62	62
100	48	59	40	40
200	35	34	25	25
500	23	19	15	14

The performance results across different values of N reveal distinct differences among the methods. OLS performs better than RCC for smaller values of N , as RCC does not utilize distributional information. However, RCC surpasses OLS when $N \geq 200$, likely due to the contraction of the feasible parameter set S as the sample size increases. Both OLS and RCC yield higher MAE values compared to CLLS and the proposed method across all sample sizes. CLLS provides strong estimates, with MAE values closely matching those of our proposed method.

Recall that once S is determined, Algorithm 2 is applied to find the optimal solution that maximizes the likelihood function. The additional computational cost of this step is minimal due to the limited number of polygon edges, making the speed comparison nearly consistent with previous results (Table 3). Table 5 presents the ratios of the mean processing times of OLS, RCC, and CLLS relative to our proposed method.

Table 5. Processing time ratios of OLS, RCC, and CLLS relative to the proposed method for different sample sizes N .

N	OLS/Ours	RCC/Ours	CLLS/Ours
20	0.77	110	11
50	0.61	140	36
100	0.54	121	80
200	0.49	132	283
500	0.42	139	1618

Under normal distribution, the proposed method demonstrates computational efficiency, being only slower than OLS, which, however, suffers from significantly lower accuracy compared to our method. It is worth noting that OLS benefits from highly optimized native MATLAB functions, which provide a relative speed advantage compared to our method, implemented as a custom M-file.

On the other hand, RCC and CLLS achieve higher accuracy, with CLLS exhibiting results slightly better than ours. However, they exhibit substantially higher computational costs, particularly as N increases. This notable difference underscores the computational burden of iterative solvers like CLLS in large-scale problems, positioning our method as a more practical choice for applications demanding both speed and accuracy. These results affirm that our method achieves an effective balance between computational speed and precision, making it well-suited for real-time applications and large datasets where other techniques may be either too slow or inaccurate.

Conclusion

This study presents an efficient approach for line fitting under bounded measurement errors, introducing the Corridor Method as a fast algorithm to delineate the feasible parameter set (FPS) exactly. For cases without distributional assumptions, we propose using the centroid of the FPS as a robust estimate, demonstrating high accuracy and computational efficiency across a range of error distributions and sample sizes.

We further extend the approach to handle normally distributed errors by identifying the most likely estimate within the FPS, solving a convex quadratic optimization problem in a non-iterative, exact manner. This extension avoids the computational demands of iterative optimization solvers, making it suitable for real-time applications.

In both settings, the performance of the proposed approach consistently surpassed competitor methods including Ordinary Least Squares, Constrained Linear Least Squares and the Regularized Chebyshev Center in terms of both speed and accuracy. Overall, this work provides a practical and precise solution for real-time line fitting applications across

diverse bounded error scenarios and dataset sizes. The publicly available MATLAB implementation offers researchers and practitioners an efficient tool for bounded-error environments.

Future Work

Future research may extend the proposed exact bounded-error line fitting framework in several directions. A promising avenue is the integration of the method into deep-learning pipelines, enabling differentiable bounded-error constraints for tasks such as line segment detection or geometric regression. Another direction is the application of the algorithm to real-world domains where strict error bounds naturally arise, including spectroscopy, nano-positioning, and manufacturing metrology. Finally, generalizing the polygonal feasible-set characterization to higher-dimensional or nonlinear models remains an open challenge.

Acknowledgements

The authors received no specific acknowledgments for this work.

Declaration of conflicting interests

The authors declare that they have no conflicts of interest.

Funding

This research received no external funding.

References

- Agaoglu A (2023) *Novel Vertex Enumeration Algorithms with Mechanical Engineering Applications*. PhD thesis, Yeditepe University, Istanbul, Turkey. Available at: https://tez.yok.gov.tr/UlusalTezMerkezi/TezGoster?key=G_oJlrKE4SgJUkomyAKpRyeIr45ekUiFSWLKI2jhQ77SAj3Hu0TT3
- Aishima K (2024) Total least squares problem with a rank-deficient constraint matrix. *Numerical Linear Algebra with Applications* 31(2): e2531. DOI:10.1002/nla.2531.
- Alamo T, Bravo JM and Camacho EF (2005) Guaranteed state estimation by zonotopes. *Automatica* 41(6): 1035–1043. DOI: 10.1016/j.automatica.2004.12.008.
- Avis D, Bremner D and Seidel R (1997) How good are convex hull algorithms? *Computational Geometry* 7(5): 265–301. DOI:[https://doi.org/10.1016/S0925-7721\(96\)00023-5](https://doi.org/10.1016/S0925-7721(96)00023-5). URL <https://www.sciencedirect.com/science/article/pii/S0925772196000235>.
- Beck A and Eldar YC (2007) Regularization in regression with bounded noise: A chebyshev center approach. *SIAM Journal on Matrix Analysis and Applications* 29(2): 606–625. DOI:10.1137/060656784. URL <https://doi.org/10.1137/060656784>.
- Billard L and Diday E (2000) Regression analysis for interval-valued data. In: *Data Analysis, Classification, and Related Methods*. Berlin: Springer, pp. 369–374.
- Billard L and Diday E (2002) Symbolic regression analysis. In: Arya KV and Arbelaitz M (eds.) *Classification, Clustering, and Data Analysis: Recent Advances and Applications*. Berlin: Springer, pp. 281–288.

- Bosse H, Grötschel M and Henk M (2005) Polynomial inequalities representing polyhedra. *Mathematical Programming* 103(1): 35–44. DOI:10.1007/s10107-004-0563-2. URL <https://doi.org/10.1007/s10107-004-0563-2>.
- Boyer E, Kalogeiton V et al. (2024) Linefit: A geometric approach for fitting line segments in images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 12345–12354.
- Casini M, Garulli A and Vicino A (2024) Set-membership estimation for discrete-time linear systems with binary sensor measurements. *Automatica* 159: 111358. DOI:10.1016/j.automatica.2023.111358.
- Cimini G and Bemporad A (2017) Exact complexity certification of active-set methods for quadratic programming. *IEEE Transactions on Automatic Control* 62(12): 6094–6109. DOI: 10.1109/TAC.2017.2696742.
- de A Lima Neto E and de AT de Carvalho F (2010) Constrained linear regression models for symbolic interval-valued variables. *Computational Statistics and Data Analysis* 54(2): 333–347. DOI:<https://doi.org/10.1016/j.csda.2009.08.010>. URL <https://www.sciencedirect.com/science/article/pii/S0167947309003065>.
- Inc TM (2024) Matlab version: 24.1.0 (r2024a). URL <https://www.mathworks.com>.
- Martínez-Otzeta JM, Rodríguez-Moreno I, Mendiadua I and Sierra B (2023) Ransac for robotic applications: A survey. *Sensors* 23(1). DOI:10.3390/s23010327. URL <https://www.mdpi.com/1424-8220/23/1/327>.
- McKittrick R and Christy JR (2024) Biases in errors-in-variables estimation of the climate feedback parameter. *Journal of Climate* 37(8): 2453–2468. DOI:10.1175/JCLI-D-23-0436.1.
- Patil T and Ghorai S (2016) Robust zero-crossing detection of distorted line voltage using line fitting. In: *2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECOT)*. pp. 92–96. DOI:10.1109/ICEECOT.2016.7955192.
- Pérez LD, Serrano BR, García JAA, Fabra JAYE, Maza EM and Gracia MT (2019) Trajectory definition with high relative accuracy (hra) by parametric representation of curves in nano-positioning systems. *Micromachines (Basel)* 10(9): 597. DOI:10.3390/mi10090597. URL <https://doi.org/10.3390/mi10090597>.
- Rego BS, Raffo GV and Scott JK (2024) A comparison of set-membership techniques for parameter estimation and state estimation. *Automatica* 160: 111434. DOI:10.1016/j.automatica.2023.111434.
- Rodriguez-Lujan I, Huerta R, Elkan C and Cruz CS (2010) Quadratic programming feature selection. *The Journal of Machine Learning Research* 11: 1491–1516.
- Rousseeuw PJ and Leroy AM (1987) Robust regression and outlier detection. *Wiley Series in Probability and Mathematical Statistics*.
- Souza LC, Souza RM, Amaral GJ and Silva Filho TM (2017) A parametrized approach for linear regression of interval data. *Knowledge-Based Systems* 131: 149–159. DOI:<https://doi.org/10.1016/j.knosys.2017.06.012>. URL <https://www.sciencedirect.com/science/article/pii/S0950705117302861>.
- Tang W, Wang Z and Shen Y (2024) Revisiting minimum enclosing ellipsoids for set-membership estimation. *IEEE Transactions on Automatic Control* 69(5): 3215–3222. DOI:10.1109/TAC.2023.3312345.
- Teplyakov L, Erlygin L and Shvets E (2022) Lsdnet: Trainable modification of lsd algorithm for real-time line segment detection. *IEEE Access* 10: 45256–45265. DOI:10.1109/ACCESS.2022.3169177.
- The MathWorks, Inc (2024a) *lsqlin - Solve constrained linear least-squares problem*. URL <https://www.mathworks.com/help/optim/ug/lsqlin.html>. Accessed: 2024-11-04.
- The MathWorks, Inc (2024b) *polyfit - Fit polynomial to data*. URL <https://www.mathworks.com/help/matlab/ref/polyfit.html>. Accessed: 2024-11-04.
- Tong H (2023) Functional linear regression with huber loss. *Journal of Complexity* 74: 101696. DOI:<https://doi.org/10.1016/j.jco.2022.101696>. URL <https://www.sciencedirect.com/science/article/pii/S0885064X22000619>.
- Wang H, Guan R and Wu J (2012) Linear regression of interval-valued data based on complete information in hypercubes. *Journal of Systems Science and Systems Engineering* 21(4): 422–442. DOI:10.1007/s11518-012-5203-4. URL <https://doi.org/10.1007/s11518-012-5203-4>.
- Wang Y, Puig V and Cembrano G (2023) Zonotopic set-membership estimation for linear time-varying descriptor systems. *IEEE Transactions on Automatic Control* 68(12): 7589–7596. DOI:10.1109/TAC.2023.3265102.
- Yan H, Grasso M, Paynabar K and Colosimo BM (2020) Real-time detection of clustered events in video-imaging data with applications to additive manufacturing. URL <https://arxiv.org/abs/2004.10977>.
- Zhang K, Jiang B and Shi P (2024) Fault detection and isolation for switched linear parameter-varying systems based on zonotopic set-membership estimation. *International Journal of Robust and Nonlinear Control* 34(3): 1987–2003. DOI:10.1002/rnc.7012.