Modeling for the Widths of Corridors

Siqi Pan

Abstract

This paper investigates the minimum width $G(d, l, p, w)$ of a corridor, given the known values of door width d, wall thickness p, object length l, and object width w. By establishing and analyzing the motion equation, the function G is fitted using both linear and quadratic functions through computer simulation. The quadratic fit is then optimized, leading to the final analytical expression for the function G.

1 Problem

In interior design, designers may encounter the following problem, as shown in Fig. 1: there is a door with width d leading to a corridor, and the thickness of the wall where the door is located is p. To allow an object with a length of l and width of w to be moved into the corridor, what is the minimum required width m of the corridor? To determine the minimum width of the corridor, we need to find a function G that relates the door width d, wall thickness p, object length l, and object width w, such that:

$$\min\{m\} = G(d, l, p, w) \tag{I}$$

The main purpose of this paper is to find the function G.

Unless otherwise specified, the unit of length used in this paper is meters.

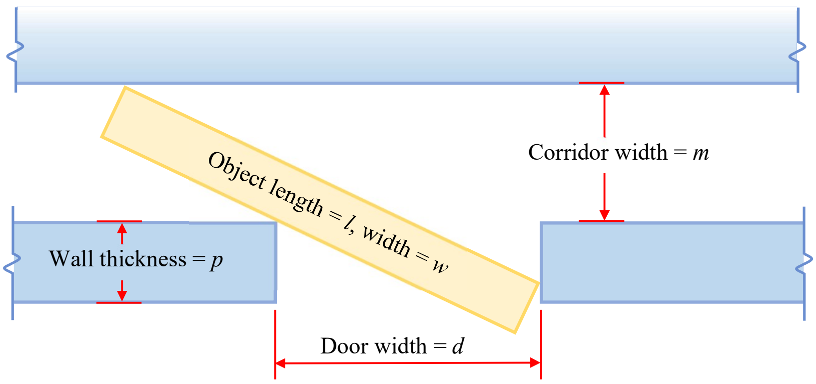


Fig. 1 The Problem

2 Background

There have been related studies on similar problems, which investigate the maximum length and width of an object that can be moved into an L-shaped corner formed by two corridors of equal width (as shown in Fig. 2). This problem has several differences from the one addressed in this paper.

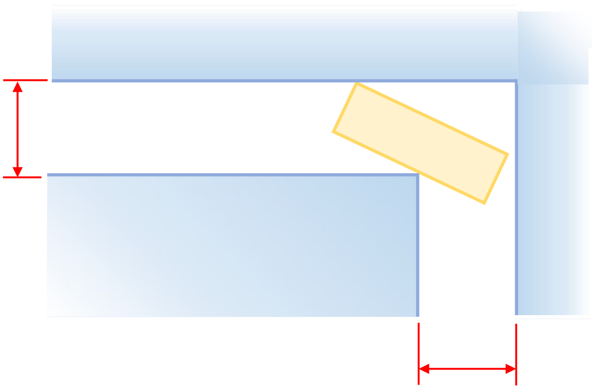


Fig. 2 Related Research

1. Previous studies considered the allowable size of an object when the corridor has already been constructed, while this paper focuses on how to design the corridor width in order to move an object of a given size.
2. If the wall thickness p of the door in this paper is assumed to be infinitely large, the door opening can be considered as a part of the corridor. The corner formed by two corridors is a special case of the corner formed by a door and a corridor. The research in this paper is more general than previous studies.
3. Previous studies considered the case where the corridor widths are the same, while this paper assumes that, in general, the corridor width m could differ from the door opening width d.

3 Assumptions

1. The object is a rectangular cuboid with dimensions in three directions: length l, width w, and height. It is assumed that the height is greater than the length l, and the length l is greater than the width w.
2. It is assumed that the height of the door opening is greater than the height of the object, so the effect of the object's height on the movement is not considered in this paper.
3. It is assumed that the width of the door opening d is greater than the width of the object w.
4. The object is considered to be a rigid body, meaning that bending or folding of the object is not considered.

4 Motion Equation

4.1 Critical Conditions

During the process of moving the object, we need to position it as shown in Fig. 3 to pass through the door opening, which roughly involves four stages. The first stage is when point B on the object is positioned exactly against the bottom of the door frame (Fig. 3-①). At this point, in order to minimize the vertical distance, the right end of the object is pressed against the wall (Fig. 3-②) and the object slides until point B reaches the top of the door frame (Fig. 3-③). At this stage, the object can rotate around point A and smoothly enter the door until it reaches the horizontal position, allowing the object to enter the door successfully (Fig. 3-④).

Due to limitations of the door frame, there are two critical situations during the moving process.

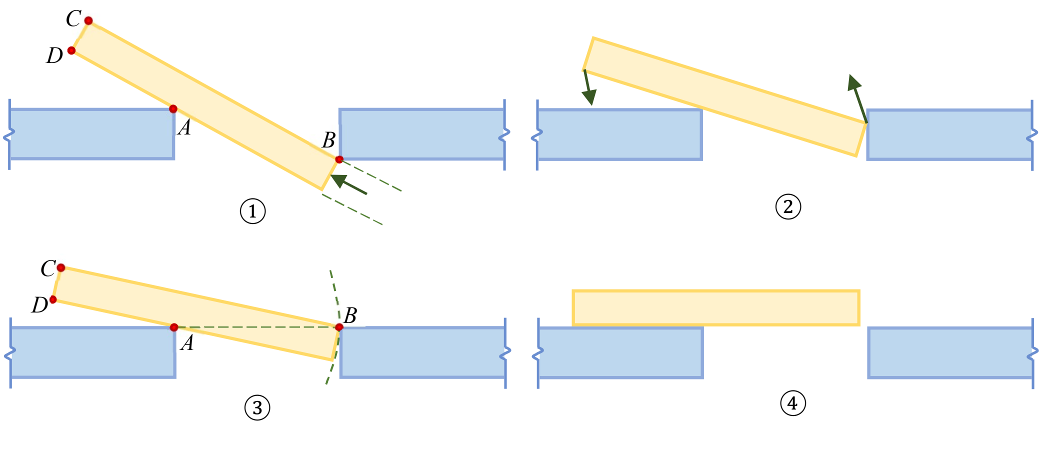


Fig. 3 Moving Process

The first critical condition is shown in Fig. 3-①: point B on the object is exactly against the bottom of the door frame. During the process of moving the object into the door before reaching this position, the object is not yet restricted by the door frame and can be slid in the direction indicated by the arrow in the figure. Once the object reaches this position, it starts to press against the door frame and moves inward. This state is considered as the initial condition.

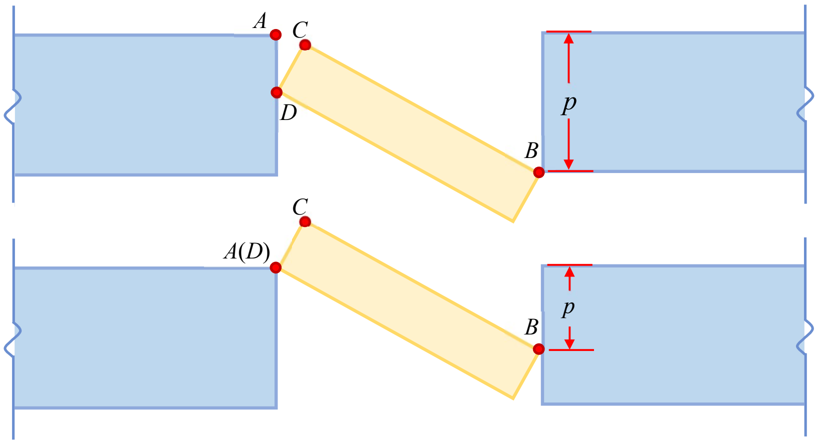


Fig. 4 Special Case in the First Critical Condition

In the first critical condition, if the wall is thick and the object is relatively short (as shown in Fig. 4), it is possible that when point B of the object reaches the bottom of the door frame, point D on the object may not have yet reached point A at the top of the opposite side of the door frame. In this case, we move the object along the door frame to align point D on the object with point A on the opposite side of the door frame. This alignment is considered as the first critical condition. For convenience in calculations, we can replace the original door frame thickness p with $p = \min\{p, \sqrt{l^2 + w^2 - d^2}\}$, thus transforming this special case into a more general one. In the following sections, unless otherwise specified, the door frame thickness p will be defined using the formula above.

The second critical condition is shown in Fig. 3-③: point B on the object is exactly against the top of the door frame. When the object reaches this position, it can simply rotate around point A to enter the door frame. During this rotation, the new position of point C will not be higher than the position of point C in this critical condition. Therefore, this second critical condition is considered the termination point. As long as the object can move from the first critical situation to the second, it can be moved through the door.

4.2 Motion Equation

To determine the minimum corridor width m, it is necessary to consider the maximum space required by the object during the moving process. Therefore, we need to establish the equations of motion for the object and determine the minimum corridor width required in all cases (with different door widths d, wall thicknesses p, object lengths l, and object widths w, in order to ultimately derive the function G.

When the mover stands outside the door to move the object, it is easier to observe the change in the angle between the object and the door frame. On the other hand, when the mover stands inside the door to move the object, it is easier to notice the change in the distance between the object's corner and the wall. Therefore, the following two approaches are used to establish the equations of motion: one considers the angle as the independent variable, and the other considers the displacement as the independent variable.

4.2.1 Angle as the Independent Variable

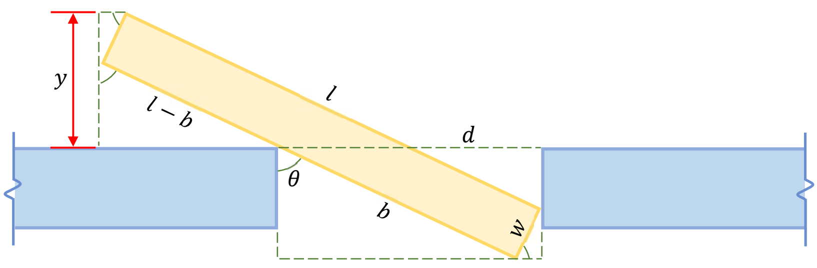


Fig. 5 Angle as the Independent Variable

As shown in the figure, let θ be the angle between the vertical wall and the object's length, and let y be the distance from the top-left corner of the object to the wall. The relationship between y and θ can be expressed as $y = f(\theta)$:

$$\min\{m\} = \max\{y\} = \max\{f(\theta)\} \tag{1}$$

In the two critical conditions mentioned in Section 4.1, θ is given by $\arcsin\frac{\sqrt{d^2-w^2}}{d}$ and $\left(\arcsin\frac{d}{\sqrt{d^2+p^2}} - \arcsin\frac{w}{\sqrt{d^2+p^2}}\right)$. Therefore, in equation (1), the domain of $f(\theta)$ is the set $\left\{\theta \,\middle|\, \arcsin\frac{p}{\sqrt{d^2+p^2}} - \arcsin\frac{w}{\sqrt{d^2+p^2}} \le \theta \le \arcsin\frac{\sqrt{d^2-w^2}}{d}\right\}$.

As shown in Figure 5,

$$l = \frac{d - w\cos\theta}{\sin\theta} + \frac{y - w\sin\theta}{\cos\theta}.$$

After simplification, we obtain

$$y = f(\theta) = \frac{l\sin\theta\cos\theta - d\cos\theta + w}{\sin\theta}$$

$$\left(\arcsin\frac{d}{\sqrt{d^2+p^2}} - \arcsin\frac{w}{\sqrt{d^2+p^2}} \le \theta \le \arcsin\frac{w}{d}\right). \tag{II}$$

4.2.2 Displacement as the Independent Variable

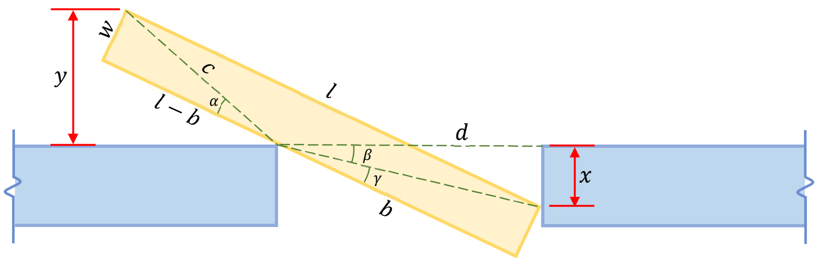


Fig. 6 Displacement as the Independent Variable

As shown in the figure, let x be the distance from the object's top-right corner to the wall, and let y be the distance from the object's top-left corner to the wall. The relationship between y and x can be expressed as $y = g(x)$, thus we have

$$\min\{m\} = \max\{y\} = \max_{0\le x\le p}\{g(x)\}.$$

Next, we establish the relationship $y = g(x)$ $(0 \le x \le p)$.

As can be seen from the above figure,

$$\sin\beta = \frac{x}{\sqrt{x^2+d^2}}, \tag{4}$$

$$\cos\beta = \frac{d}{\sqrt{x^2+d^2}}, \tag{5}$$

$$b = \sqrt{x^2 + d^2 - w^2}, \tag{6}$$

$$\sin\gamma = \frac{w}{\sqrt{x^2 + d^2}}, \tag{7}$$

$$\cos\gamma = \frac{b}{\sqrt{x^2 + d^2}}, \tag{8}$$

$$\sin\alpha = \frac{w}{c}, \tag{9}$$

$$\cos\alpha = \frac{l - b}{c}. \tag{10}$$

Therefore,

$$\begin{aligned} y &= g(x) \\ &= c\sin(\alpha + \beta + \gamma) \\ &= c(\sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\beta\cos\gamma + \cos\alpha\cos\beta\sin\gamma - \sin\alpha\sin\beta\sin\gamma). \end{aligned}$$

Substituting equations (4), (5), (6), (7), (8), (9), and (10) into the above equation, we get

$$\begin{aligned} y &= g(x) \\ &= c\left[\frac{wbd}{c(x^2 + d^2)} + \frac{(l - b)bx}{c(x^2 + d^2)} + \frac{(l - b)dw}{c(x^2 + d^2)} - \frac{w^2 x}{c(x^2 + d^2)}\right] \\ &= \frac{ldw + lx\sqrt{x^2 + d^2 - w^2}}{x^2 + d^2} - x. \end{aligned}$$

Thus, we obtain the following equation:

$$\boxed{y = g(x) = \frac{ldw + lx\sqrt{x^2 + d^2 - w^2}}{x^2 + d^2} - x \qquad (0 \le x \le p)} \tag{III}$$

4.2.3 Comparison of the Motion Equations

We will now prove that the formulas (II) and (III) obtained above are equivalent. As shown in Fig. 5 and Fig. 6,

$$\sin\theta = \frac{bd - wx}{x^2 + d^2}, \tag{11}$$

$$\cos\theta = \frac{bx + dw}{x^2 + d^2}, \tag{12}$$

Substituting equations (11) and (12) into (II) gives:

$$y = f(\theta) = \frac{dwl + blx}{x^2 + d^2} - \frac{bdx + d^2w}{bd - wx} + \frac{wx^2 + wd^2}{bd - wx}$$

$$= \frac{dwl + lx\sqrt{x^2 + d^2 - w^2}}{x^2 + d^2} - x = g(x)$$

Therefore, the forms of the equations are equivalent.

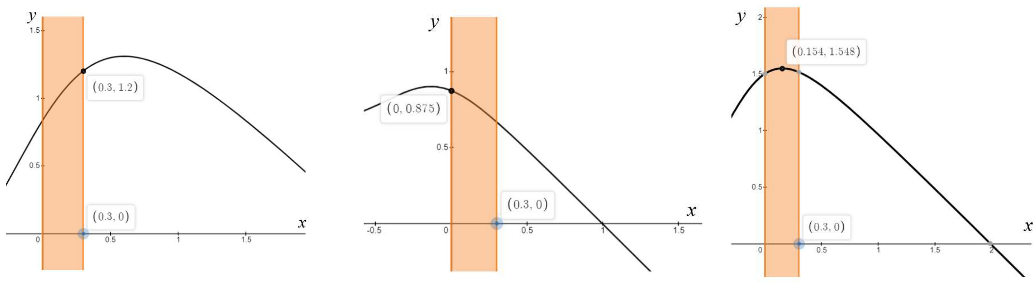
At the same time, when $x = 0$ and $x = p$, the values of θ are $\arcsin\frac{\sqrt{d^2-w^2}}{d}$ and $\left(\arcsin\frac{d}{\sqrt{d^2+p^2}} - \arcsin\frac{w}{\sqrt{d^2+p^2}}\right)$, respectively. Therefore, the domains of $f(\theta)$ and $g(x)$ are also equivalent.

Thus, the proof is complete.

By observing equations (II) and (III), we can see that the domain of equation (II) is more complex. Therefore, this paper chooses equation (III) as the focus of study.

5 Analysis of the Motion Equation

Since calculating the extreme values of $g(x)$ is relatively complex, this paper selects different combinations of d, l, p, and w, and uses a computer to plot the graph of $g(x)$. This approach is used to determine the monotonicity and maximum value of the function within the range $0 \le x \le p$.



(d, l, p, w) = (0.9, 2.5, 0.3, 0.3) **(d, l, p, w) = (0.8, 1, 0.3, 0.7)** **(d, l, p, w) = (0.8, 2, 0.3, 0.6)**

Fig. 7 Graph of *g(x)*

As seen in Fig. 7, the monotonicity and the position of the maximum value of $y = g(x)$ $(0 \le x \le p)$ vary with changes in d, l, p, and w. Sometimes, the maximum value of y occurs at $x = 0$, other times at $x = p$, and in some cases, it appears somewhere in between. Therefore, it is difficult to obtain an analytical expression for the function $G(d, l, p, w)$ using conventional methods. In this paper, a data fitting approach using computer simulation is adopted to obtain an approximate analytical expression for $G(d, l, p, w)$.

6 Data Fitting

6.1 Data

In order to fit the function *G*, data were collected based on the value ranges shown in Table 1, resulting in a total of 16,098 data points. These data points satisfy the conditions provided in the basic assumptions and critical conditions.

Table 1 Data

Parameter	Min. Value	Max. Value	Step
d	0.8	1.5	0.1
l	$d + 0.1$	2.0	0.1
w	0.1	d	0.1
p	0.25	$\sqrt{l^2 + w^2 - d^2}$	0.05

At the same time, for each set of parameters (*d, l, p, w*), a computer program was used to substitute the parameters into equation (III) to solve for the exact value of $\min\{m\}$. These exact values were then stored in a column vector **b**. The source code of the computer program for calculating the maximum value of the function is provided in the appendix.

6.2 Linear Function Fitting

The equation $G_1(d, l, p, w) = c_1 d + c_1 l + c_3 p + c_4 w + c_5$ is a linear fit for the function *G*, where (d_k, l_k, p_k, w_k) represents the *k-th* set of data obtained earlier, and there are a total of *n* sets of data. Let

$$A = \begin{pmatrix} d_1 & l_1 & p_1 & w_1 & 1 \\ d_2 & l_2 & p_2 & w_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_n & l_n & p_n & w_n & 1 \end{pmatrix}, \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix}, \mathbf{s} = \begin{pmatrix} G_1(d_1, l_1, p_1, w_1) \\ G_1(d_2, l_2, p_2, w_2) \\ \vdots \\ G_1(d_n, l_n, p_n, w_n) \end{pmatrix}$$

where *A* is the matrix formed by the parameter set $(d_k, l_k, p_k, w_k, 1)$, **c** is the coefficient vector of the fitting function, and **s** is the vector of the fitted approximate values.

Thus, we have the equation $A\mathbf{c} = \mathbf{s}$.

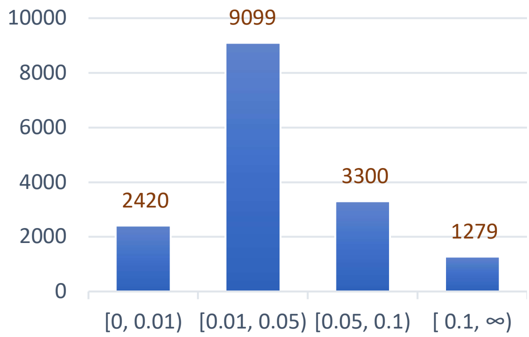
To find the best fitting function, we need to minimize $||\mathbf{b} - \mathbf{s}||$, i.e., the distance between the vector **b** (the exact values) and the vector **s** (the fitted approximate values). Let $R(A)$ be the column space of matrix *A*, and $N(A^T)$ be the null space of matrix A^T. It is easy to see that $\mathbf{b} - \mathbf{s} \in R(A)^{\perp}$, meaning $\mathbf{b} - A\mathbf{c} \in N(A^T)$. Therefore, we have $A^T(\mathbf{b} - A\mathbf{c}) = \mathbf{0}$

This is a linear system, which can be solved using a computer (the source code can be found in the Appendix). Solving this system, we obtain: $\mathbf{c} \approx (-0.942, 0.728, -0.003, 1.312, 0.003)^T$.

Thus, we have:

$$G_1(d, l, p, w) = -0.942d + 0.728l - 0.003p + 1.312w + 0.003$$

To examine the error after fitting, we define the error vector $\mathbf{r} = \mathbf{b} - \mathbf{s}$. By solving for **r** using a computer, we can obtain the following statistical chart. The horizontal axis of the chart represents the error range, divided into four categories: Less than 1 cm, between 1 cm and 5 cm, between 5 cm and 10 cm, and greater than 10 cm.



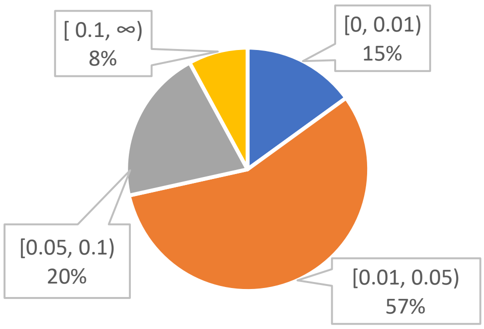


Fig. 8 Linear Fit Error Statistical Chart

From the statistical data, it can be seen that 28% of the data points have a linear fit error exceeding 5 cm, which indicates that the results are not particularly satisfactory. Therefore, a quadratic fitting approach is attempted next.

6.3 Quadratic Function Fitting

The equation $G_2(d,l,p,w) = c_1 d^2 + c_2 l^2 + c_3 p^2 + c_4 w^2 + c_5 dl + c_6 dp + c_7 dw + c_8 lp + c_9 lw + c_{10} pw + c_{11} d + c_{12} l + c_{13} p + c_{14} w + c_{15}$ is a quadratic fit for the function G. Let

$$A = \begin{pmatrix} d_1^2 & l_1^2 & p_1^2 & w_1^2 & d_1 l_1 & d_1 p_1 & d_1 w_1 & l_1 p_1 & l_1 w_1 & p_1 w_1 & d_1 & l_1 & p_1 & w_1 & 1 \\ d_2 & l_2 & p_2 & w_2 & d_2 l_2 & d_2 p_2 & d_2 w_2 & l_2 p_2 & l_2 w_2 & p_2 w_2 & d_2 & l_2 & p_2 & w_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_n & l_n & p_n & w_n & d_n l_n & d_n p_n & d_n w_n & l_n p_n & l_n w_n & p_n w_n & d_n & l_n & p_n & w_n & 1 \end{pmatrix},$$

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{15} \end{pmatrix}, \mathbf{s} = \begin{pmatrix} G_2(d_1, l_1, p_1, w_1) \\ G_2(d_2, l_2, p_2, w_2) \\ \vdots \\ G_2(d_n, l_n, p_n, w_n) \end{pmatrix}.$$

The meanings of the symbols are the same as in the linear fit, and using the same approach as in the linear fit, we can obtain the following equation:

$$\begin{aligned} G_2(d,l,p,w) = {} & 0.624d^2 + 0.066l^2 - 0.010p^2 + 0.148w^2 \\ & -0.538dl + 0.001dp - 0.872dw + 0.023lp + 0.531lw - 0.021pw \\ & -0.904d + 0.722l - 0.005p + 1.270w \end{aligned}$$

Using the same method to calculate the error vector, the following statistical chart can be obtained:

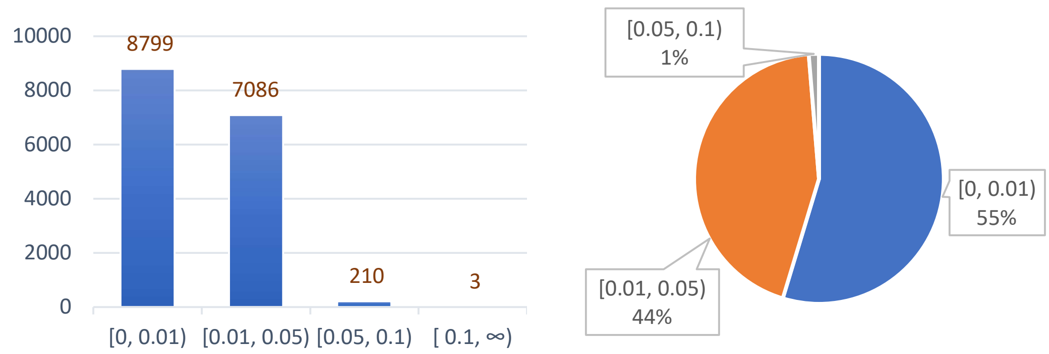


Fig. 9 Quadratic Fit Error Statistical Chart

From the statistical chart, it can be seen that the quadratic fit yields satisfactory results, with 99% of the data having an error less than 5 cm. However, by examining the specific data, it is observed that the fit performs poorly when d and w are close to each other. Therefore, the quadratic function fit can be further optimized.

6.5 Optimization of Quadratic Function Fitting

$G_3(d, l, p, w)$ is the improved quadratic fitting function.

When $\frac{w}{d} = 1$, $\min\{m\} = l$. To improve the fitting performance when w is close to d, when $\frac{w_k}{d_k} \geq 0.95$, the data set (d_k, l_k, p_k, w_k) is not stored in matrix A, and the value of $G_3(d_k, l_k, p_k, w_k)$ is directly set to l.

After performing the fitting using the new matrix A, the fitted equation is obtained as:

$$\boldsymbol{G_3(d, l, p, w)} = \begin{cases} \boldsymbol{l} & \boldsymbol{\frac{w_k}{d_k} \geq 0.95} \\ \begin{aligned} &0.545d^2 + 0.076l^2 - 0.012p^2 + 0.142w^2 \\ &-0.501dl - 0.003dp - 0.774dw + 0.028lp + 0.480lw - 0.022pw \\ &-0.823d + 0.668l - 0.005p + 1.244w \end{aligned} & \frac{w_k}{d_k} < 0.95 \end{cases}$$

By using $G_3(d, l, p, w)$ to calculate the approximate values and the error vector, the following statistical chart can be obtained:

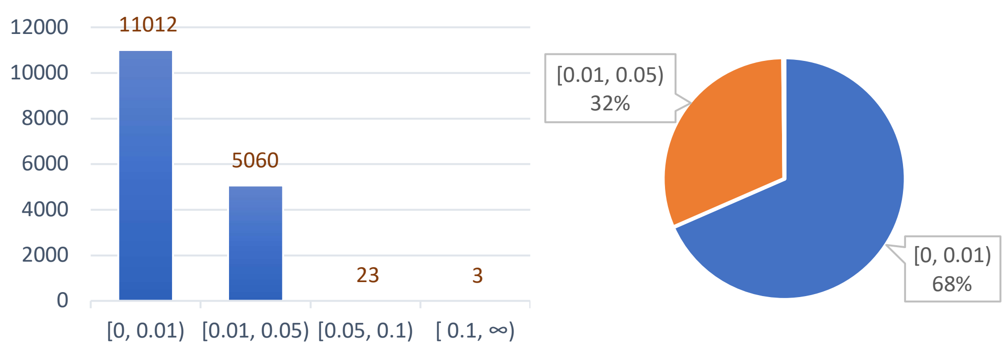


Fig. 10 Optimized Quadratic Fit Error Statistical Chart

From the statistical chart, it can be seen that the optimized quadratic fit performs very well, with the error largely controlled within 5 cm.

7 Conclusion

In conclusion, by taking G_3 as the final fitting function, the following function G can be obtained:

Door width = d, wall thickness = p, object length = l, and object width = w Minimum corridor width = $\boldsymbol{G(d,l,p,w)}$
$\boldsymbol{p = \min\left\{p, \sqrt{l^2 + w^2 - d^2}\right\}}$ $\boldsymbol{l = \max\{l, w\}}$ $\boldsymbol{w = \min\{l, w\}}$
$\boldsymbol{G(d,l,p,w)} = \begin{cases} w & d \geq l \\ \text{No solution} & d < w \\ l & \frac{w_k}{d_k} \geq 0.95 \\ \begin{array}{l} 0.545d^2 + 0.076l^2 - 0.012p^2 + 0.142w^2 \\ -0.501dl - 0.003dp - 0.774dw + 0.028lp + 0.480lw - 0.022pw \\ -0.823d + 0.668l - 0.005p + 1.244w \end{array} & \frac{w_k}{d_k} < 0.95 \end{cases}$

8 Application

Let's illustrate how to apply the fitted function G_3 to calculate the required corridor width with an example.

Suppose there is a mattress with a length of 1.8 meters and a width of 0.3 meters, and it needs to pass through a door that is 0.8 meters wide with 0.3-meter-thick walls. By substituting (d, l, p, w) = (0.8, 1.8, 0.3, 0.3) into the function $G_3(d,l,p,w)$, we get $G_3(d,l,p,w) \approx 0.884$. This means the minimum corridor width is approximately 0.884 meters. (Using a computer to solve for the exact value gives about 0.895 meters, with an error of approximately 1 cm.)

When designers use the formula derived in this paper, they simply need to input the four data points into the formula to obtain the approximate minimum corridor width. The error in the obtained value is generally within ±5 cm.

9 References

[1] 埃伯哈德·蔡德勒，《数学指南——实用数学手册》，科学出版社，2012
[2] Jake VanderPlas，《Python 数据科学手册》，人民邮电出版社，2018
[3] Ivan Idris，《Python 数据分析基础教程：NumPy 学习指南》，人民邮电出版社，2014
[4] Al Sweigart，《Python 编程快速上手——让繁琐工作自动化》，人民邮电出版社，2016
[5] Steven J. Leon，《Linear Algebra with Applications, Ninth Edition》，机械工业出版社，2018
[6] 《病床转弯问题的数学模型》，https://www.doc88.com/p-703867146611.html

10 Appendix

10.1 Calculate the exact value of $\min\{m\}$

```
xPrecision = 0.01

def getY(d, l, w, x):
        return (l * d * w + l * x * sqrt(x * x + d * d - w * w)) / (x * x + d * d) - x

def maxY(d, l, p, w):
    realP = sqrt(w * w + l * l - d * d)
    if p > realP:
        p = realP
    x = 0
    result = -1
    while x < p + xPrecision:
        y = getY(d, l, w, x)
        if y > result:
            result = y
        x += xPrecision
         return result
```

10.2 Solve $A^T A\mathbf{c} = A^T\mathbf{b}$

```
def leastSquareSolve(dataName, residualName):
    dataFile = open(dataName, encoding = "UTF-8")
    dataReader = csv.reader(dataFile)
    datas = array(list(dataReader), dtype = float)
    matrixShape = datas.shape
    AArray, bArray = hsplit(datas, [matrixShape[1] - 1])
    A = mat(AArray)
    b = mat(bArray)
    ATA = matmul(A.T, A)
    ATb = matmul(A.T, b)
    x = linalg.solve(ATA, ATb)
    for i in range(matrixShape[1] - 1):
        x[i, 0] = int(x[i, 0] * 1000) / 1000
    print(x)
    print("\n")
    residualFile = open(residualName, "w", encoding='UTF-8')
    residualVector = b - matmul(A, x)
    for i in range(matrixShape[0]):
        for j in range(matrixShape[1]):
            residualFile.write(str(datas[i, j]) + ", ")
        residualFile.write(str(residualVector[i, 0]) + "\n")
```