

Use of Modal Interval Analysis in Early Engineering Design

1 Part I: Modal Interval Analysis

In order to describe possible applications of modal interval analysis in early engineering design, let us start by briefly explaining what is interval computation and what is modal interval analysis.

Direct and indirect measurements: uncertainty is ubiquitous. In practice, how do we obtain the numerical values of different physical quantities?

For some quantities, we can obtain these values directly, either by performing a measurement or by eliciting the value from an expert.

Measurements are never absolutely accurate; as a result, the result \tilde{x} of the measurement is somewhat different from the actual (unknown) value x of the desired physical quantity: $\tilde{x} \neq x$. In other words, from measurements, we can only determine the value x with uncertainty: the approximation error $\Delta x \stackrel{\text{def}}{=} \tilde{x} - x$ is, in general, different from 0.

Expert estimates are usually even less accurate than measurements, so the values \tilde{x} obtained from the experts also always contain uncertainty.

For some quantities y , it is difficult (or even impossible) to measure them directly. For example, while we can directly measure the temperature on the Earth surface, it is not usually possible to directly measure the temperature on the surface of a distant planet or a star. To estimate the values of such difficult-to-measure quantities, we perform *indirect measurements*, i.e., we:

- find easier-to-measure quantities x_1, \dots, x_n which are related to the desired quantity y by a known algorithm $y = f(x_1, \dots, x_n)$;
- measure the values $\tilde{x}_1, \dots, \tilde{x}_n$ of these auxiliary quantities; and then
- apply the algorithm f to the results of direct measurements, thus producing the estimate $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$ for the desired quantity y .

Since direct measurements are never absolutely accurate, the measurement results \tilde{x}_i are, in general, different from the actual (unknown) values x_i : $\Delta x_i \stackrel{\text{def}}{=} \tilde{x}_i - x_i \neq 0$. Thus, our estimate $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$ is, in general, different from the actual value $y = f(x_1, \dots, x_n)$ of the desired quantity: $\Delta y \stackrel{\text{def}}{=} \tilde{y} - y \neq 0$.

In other words, indirectly measured quantities are also only known with uncertainty.

It is worth mentioning that the situation is usually even more complex because the dependence between y and x_1, \dots, x_n is also only known with uncertainty: even if we knew the exact values of x_1, \dots, x_n , the estimate $f(x_1, \dots, x_n)$ would still be somewhat different from the actual value y of the desired quantity.

Traditional engineering approach to describing uncertainty: using probabilities. The need to take uncertainty into account is well understood in engineering practice. Uncertainty processing techniques are part of the standard engineering education and of the standard engineering practice; see, e.g., [9].

In this approach, it is usually assumed that we know the probability distribution of different measurement errors Δx . For the analysis of the actual measurement results, this assumption is realistic: In principle, to find out the desired probability distribution, we can take a more accurate (“standard”) measuring instrument, and perform several measurements of the same quantity by both the tested and the standard instrument. From the resulting samples of Δx , we can then determine the desired probability distribution.

Such a “calibration” of a measuring instrument is indeed frequently performed. It is not always done:

- sometimes (like with a state-of-the-art instrument), we do not have a better (standard) instrument to compare;
- sometimes, calibration is too expensive,

but usually, it is performed.

However, there are situations when we do not know these probabilities: situations of early engineering design.

Situations of early engineering design: it is difficult to estimate probabilities. In the early stages of engineering design, we do not know the probability distributions because we are still planning the corresponding system – and in particular, its measuring instruments.

An additional problem is often that the distribution of the measurement errors depends on the environment, and we do not have enough information about this environment: actually, the whole purpose of the design is often to plan a mission that will provide us with information about this environment. This is a reasonably typical situation for NASA missions to planets and to the outer space.

Bounds and interval uncertainty. Instead of the *probabilities* of different values of x , in the early engineering design problems, we often only know the *bounds* \underline{x} and \bar{x} on the possible values of x : $\underline{x} \leq x \leq \bar{x}$, and some partial information about the probabilities of different possible values $x \in [\underline{x}, \bar{x}]$.

Let us start our description with the simplest situation in which we do not have any information about the probabilities, we only know the bounds \underline{x}

and \bar{x} . In such situation, the only information that we have about the actual (unknown) value of the desired quantity x is that x belongs to the interval $[\underline{x}, \bar{x}]$. This situation is usually called the situation of *interval uncertainty*.

A typical example of such a situation is direct measurement, when we only know the upper bound Δ on the (absolute value of) the measurement error Δx : $|\Delta x| \leq \Delta$. In this case, after we perform the measurement and get the measurement result \tilde{x} , we know that the actual value x is in the interval

$$[\tilde{x} - \Delta, \tilde{x} + \Delta]. \quad (1)$$

Processing interval uncertainty for indirect measurements: interval computations. As we have mentioned, the uncertainty in the direct measurements leads to the uncertainty in the result $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$ of the indirect measurement. In particular, if all the results of direct measurements are known with interval uncertainty, i.e., if we know the intervals $[\underline{x}_i, \bar{x}_i]$ that contain the actual values x_i , then the only information that we have about the actual value $y = f(x_1, \dots, x_n)$ is that y should be equal to $f(x_1, \dots, x_n)$ for some $x_i \in [\underline{x}_i, \bar{x}_i]$, i.e., in mathematical terms, that y belongs to the *range* \mathbf{y} of the function f over these intervals:

$$\mathbf{y} = f([\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_n, \bar{x}_n]) \stackrel{\text{def}}{=} \{f(x_1, \dots, x_n) : x_1 \in [\underline{x}_1, \bar{x}_1], \dots, x_n \in [\underline{x}_n, \bar{x}_n]\}. \quad (2)$$

Usually, the function $f(x_1, \dots, x_n)$ is continuous; thus, its range is also an interval $[\underline{y}, \bar{y}]$. The problem of computing the endpoints \underline{y} and \bar{y} of the range \mathbf{y} of a known function f over known intervals is known as the problem of *interval computations*; see, e.g., [6] and [7].

Interval computations: monotonic case. In general, the interval computation problem is NP-hard; see, e.g., [8]. However, in many practical cases, the dependence $f(x_1, \dots, x_n)$ is monotonic (increasing or decreasing) in each of the variables x_i . In such cases, it is possible to easily find the values x_i at which the function $f(x_1, \dots, x_n)$ attains its largest possible value \bar{y} :

- for each variable x_i with respect to which f is increasing, we should take the largest possible value of x_i : $x_i = \bar{x}_i$;
- for each variable x_i with respect to which f is decreasing, we should take the smallest possible value of x_i : $x_i = \underline{x}_i$.

It is similarly possible to find the values x_i at which the function $f(x_1, \dots, x_n)$ attains its smallest possible value \underline{y} :

- for each variable x_i with respect to which f is increasing, we should take the smallest possible value of x_i : $x_i = \underline{x}_i$;
- for each variable x_i with respect to which f is decreasing, we should take the largest possible value of x_i : $x_i = \bar{x}_i$.

Monotonic case: formal description. In the following text, we will expand this simple idea to the case of different design problems. To make this expansion easier, let us describe this idea in precise terms. For each variable x_i , let ε_i describe the sign of monotonicity, i.e.,

- we take $\varepsilon_i = 1$ if the function f is increasing with respect to x_i , and
- we take $\varepsilon_i = -1$ if the function f is decreasing with respect to x_i .

In this case, the above advice means that

- to find the value \bar{y} , we must take $x_i = \bar{x}_i$ if $\varepsilon_i = 1$ and $x_i = \underline{x}_i$ if $\varepsilon_i = -1$; and
- to find the value \underline{y} , we must take $x_i = \underline{x}_i$ if $\varepsilon_i = 1$ and $x_i = \bar{x}_i$ if $\varepsilon_i = -1$.

Let us make the following notations:

- we define $x_i^+ \stackrel{\text{def}}{=} \bar{x}_i$ if $\varepsilon_i = 1$ and $x_i^+ \stackrel{\text{def}}{=} \underline{x}_i$ if $\varepsilon_i = -1$;
- we define $x_i^- \stackrel{\text{def}}{=} \underline{x}_i$ if $\varepsilon_i = 1$ and $x_i^- \stackrel{\text{def}}{=} \bar{x}_i$ if $\varepsilon_i = -1$.

In this case, we have $\bar{y} = f(x_1^+, \dots, x_n^+)$ and $\underline{y} = f(x_1^-, \dots, x_n^-)$ and therefore, the desired range $[\underline{y}, \bar{y}]$ of $y = f(x_1, \dots, x_n)$ has the form

$$[\underline{y}, \bar{y}] = [f(x_1^-, \dots, x_n^-), f(x_1^+, \dots, x_n^+)]. \quad (3)$$

Interval arithmetic. An important particular case is the case of arithmetic operations $y = f(x_1, x_2)$. For example, addition $y = f(x_1, x_2) = x_1 + x_2$ is increasing with respect to both variables x_i ; as a result, the range $[\underline{y}, \bar{y}]$ of possible values of $y = x_1 + x_2$ is equal to

$$[\underline{y}, \bar{y}] = [\underline{x}_1, \bar{x}_1] + [\underline{x}_2, \bar{x}_2] = [\underline{x}_1 + \underline{x}_2, \bar{x}_1 + \bar{x}_2]. \quad (4)$$

For example, if we know that $x_1 \in [1, 2]$ and that $x_2 \in [3, 5]$, we can conclude that

$$x_1 + x_2 \in [1, 2] + [3, 5] = [1 + 3, 2 + 5] = [4, 7]. \quad (5)$$

The subtraction function $y = f(x_1, x_2) = x_1 - x_2$ is increasing with respect to x_1 and decreasing with respect to x_2 ; as a result, the range $[\underline{y}, \bar{y}]$ of possible values of $y = x_1 - x_2$ is equal to

$$[\underline{y}, \bar{y}] = [\underline{x}_1, \bar{x}_1] - [\underline{x}_2, \bar{x}_2] = [\underline{x}_1 - \bar{x}_2, \bar{x}_1 - \underline{x}_2]. \quad (6)$$

For non-negative values x_1 and x_2 , the product $y = f(x_1, x_2) = x_1 \cdot x_2$ is a (non-strictly) increasing function of both x_1 and x_2 . Thus, we have

$$[\underline{y}, \bar{y}] = [\underline{x}_1, \bar{x}_1] \cdot [\underline{x}_2, \bar{x}_2] = [\underline{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2]. \quad (7)$$

For $x_1 \geq 0$ and $x_2 > 0$, the ratio $y = f(x_1, x_2) = x_1/x_2$ is increasing in x_1 and decreasing in x_2 . Thus, we have

$$[\underline{y}, \bar{y}] = [\underline{x}_1, \bar{x}_1]/[\underline{x}_2, \bar{x}_2] = [\underline{x}_1/\bar{x}_2, \bar{x}_1/\underline{x}_2]. \quad (8)$$

For general (not necessarily positive) values x_i , the product is increasing for some values and decreasing for others. Thus, in the general case, for the range, we have a more complicated formula

$$[\underline{x}_1, \bar{x}_1] \cdot [\underline{x}_2, \bar{x}_2] = [\min(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \bar{x}_2, \bar{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2), \max(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \bar{x}_2, \bar{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2)]. \quad (9)$$

Similarly, when $0 \notin [\underline{x}_2, \bar{x}_2]$, we have

$$[\underline{x}_1, \bar{x}_1]/[\underline{x}_2, \bar{x}_2] = [\min(\underline{x}_1/\underline{x}_2, \underline{x}_1/\bar{x}_2, \bar{x}_1/\underline{x}_2, \bar{x}_1/\bar{x}_2), \max(\underline{x}_1/\underline{x}_2, \underline{x}_1/\bar{x}_2, \bar{x}_1/\underline{x}_2, \bar{x}_1/\bar{x}_2)]. \quad (10)$$

The formulas (4), (6), (7), (8), (9), (10) for the range of the results of arithmetic operations are known as formulas of *interval arithmetic*.

From interval arithmetic to general interval computations. Interval arithmetic enables us to compute the range of arithmetic operations. As we have mentioned, for more complex (not necessarily monotonic) functions $f(x_1, \dots, x_n)$, computing the exact range $\mathbf{y} = f(\mathbf{x}_1, \dots, \mathbf{x}_n)$ over given intervals $\mathbf{x}_1, \dots, \mathbf{x}_n$ is, in general, an NP-hard (computationally intractable) problem.

However, we can use interval arithmetic to efficiently find an *enclosure* $\mathbf{Y} \supseteq \mathbf{y}$ for the desired range. This possibility is related to the fact that inside a computer, every expression or algorithm $f(x_1, \dots, x_n)$ is compiled into a sequence of arithmetic operations (and elementary functions such as $\exp(x)$, $\min(x_1, x_2)$, etc.). It turns out that if we simply replace each arithmetic operation with the corresponding operation of interval arithmetic, we get an enclosure for the desired range (for a proof by induction, see, e.g., [6]). This procedure is called *straightforward interval computations*. For example, for a function $f(x_1) = x_1 \cdot (1 - x_1)$, a compiler would represent the expression as a sequence of two arithmetic operations:

- first, we compute the intermediate result $r_1 = 1 - x_1$;
- then, we compute $y = x_1 \cdot r_1$.

Thus, to estimate the range \mathbf{y} of this function on the interval $\mathbf{x}_1 = [\underline{x}_1, \bar{x}_1] = [0, 1]$, we can do the following:

- first, we compute the range \mathbf{r}_1 as

$$\mathbf{r}_1 = 1 - \mathbf{x}_1 = [1, 1] - [0, 1] = [1 - 1, 1 - 0] = [0, 1]; \quad (11)$$

- then, we compute

$$\mathbf{Y} = \mathbf{x}_1 \cdot \mathbf{r}_1 = [0, 1] \cdot [0, 1] = [0 \cdot 0, 1 \cdot 1] = [0, 1]. \quad (12)$$

This clearly is a proper enclosure for the range, because for our quadratic function $f(x_1) = x_1 - x_1^2$, the maximum is attained for when $\frac{df}{dx_1} = 1 - 2x_1 = 0$, i.e., for $x_1 = 0.5$, and is equal to $0.5 - 0.5^2 = 0.25$, and the minimum 0 is attained at both endpoints $x_1 = 0$ and $x_1 = 1$. Thus, the range is $[0, 0.25] \subset [0, 1]$.

For some simple expressions (generalizing arithmetic operations), e.g., for *single use* expressions like $(x_1 + x_2) \cdot x_3$, in which each variable occurs only once, straightforward interval computations lead to the exact range. However, in general, we have a proper enclosure.

There exist many techniques for reducing the width of the enclosure; most of them are based on either changing the expression or repeating the computation for several subintervals of the original intervals – and all use straightforward interval computations as the main underlying tool; see, e.g., [6] and [7].

Design problems: direct application of interval computations. One of the objectives of design is to find the values of the design parameters for which the designed system satisfies given objectives.

Usually, we know the dependence of each of the corresponding quantities y on the parameters x_1, \dots, x_n describing the design and the environment in which the system is supposed to operate. Namely, we know an algorithm (or even an explicit formula) f for which $y = f(x_1, \dots, x_n)$. The desired constraints are usually formulated as bounds \underline{y} and \bar{y} on y : $\underline{y} \leq y \leq \bar{y}$.

If we know the exact values of the design parameters and of the parameters which characterize the environment, then we can simply compute the corresponding value $y = f(x_1, \dots, x_n)$ and check whether it satisfies the desired inequality $\underline{y} \leq y \leq \bar{y}$.

In practice, we usually know the parameters characterizing the environment with some uncertainty; also, in manufacturing, it is not possible to maintain the exact values of the design parameters, these parameters can only be maintained within a certain tolerance. As a result, for each of the parameters x_i , instead of the exact value we only know the interval $\mathbf{x}_i = [\underline{x}_i, \bar{x}_i]$ that contains the (unknown) exact value. In this situation, we can use interval computations to find the range $f(\mathbf{x}_1, \dots, \mathbf{x}_n)$ of possible values of y – or, more generally, the enclosure $[\underline{Y}, \bar{Y}]$ for this range. Thus, we are guaranteed that the actual (unknown) value of the characteristic $y = f(x_1, \dots, x_n)$ belongs to this enclosure $[\underline{Y}, \bar{Y}]$.

If this enclosure is within the desired interval: $[\underline{Y}, \bar{Y}] \subseteq [\underline{y}, \bar{y}]$, i.e., if

$$\underline{y} \leq \underline{Y} \leq \bar{Y} \leq \bar{y}, \quad (13)$$

then we can be sure that the actual (unknown) value $y \in [\underline{Y}, \bar{Y}]$ is also within the desired interval $[\underline{y}, \bar{y}]$ and thus, that for this design, the constraint is satisfied.

If the enclosure is *not* fully within the desired range $[y, \bar{y}]$ – and cannot be placed into the desired range by applying better interval computation techniques – then there is a possibility that the actual value y is outside the desired range. In this case, the original design is not satisfactory.

For example, if $f(x_1, x_2) = x_1 + x_2$, $x_1 \in [1, 2]$, $x_2 \in [3, 5]$, and the desired range is $[0, 10]$, then interval computations enable us to compute the range of $x_1 + x_2$ as $[1, 2] + [3, 5] = [4, 7]$. Since $[4, 7] \subseteq [0, 10]$, this design is satisfactory.

On the other hand, if in the same example, the desired range is $[0, 6]$, then the range $[4, 7]$ of $x_1 + x_2$ is *not* contained in the desired range $[0, 6]$ and thus, the design is not satisfactory.

The problem of generating a design. In the previous section, we considered one specific design-related problem, when the design is already given, and all we wanted was to check whether this design is satisfactory or not.

This checking is important, but in real-life design problems, the most important problem is not *checking* whether a proposed design is satisfactory, but rather *generating* a design.

This design generation problem is usually simple to solve if we make an idealized assumption that there is no uncertainty. Let us consider the simplest possible example when:

- the quality $y = f(x_1, d)$ of the design depends on a single environmental parameter x_1 and a single design parameter d ;
- we know the exact value of the environmental parameter x_1 ;
- the dependence is described by a simple formula $f(x_1, d) = x_1 + d$, and
- we would like the quality to have the desired value y .

In this case, the problem is simple: we know x_1 , we know y , and we want to find the value d for which $x_1 + d = y$. The solution is straightforward: we should take $d = y - x_1$.

Generating a design: case of interval uncertainty. In practice, of course, all the values are known with uncertainty:

- instead of the exact value of the parameter x_1 that describes the environment, we only know the bounds \underline{x}_1 and \bar{x}_1 on x_1 , i.e., in other words, we only know the interval $[\underline{x}_1, \bar{x}_1]$ of possible values of x_1 ;
- similarly, instead of a single allowed value of the characteristic y , we require that y is within the given bounds \underline{y} and \bar{y} ; in other words, we are given the interval $[\underline{y}, \bar{y}]$ of possible values of y .

In this case, we would like to describe all possible values of the design parameter d with the following property: for every $x_1 \in [\underline{x}_1, \bar{x}_1]$, we want to guarantee that $x_1 + d \in [\underline{y}, \bar{y}]$. The set of all such d is called the *tolerance solution*.

Computing the tolerance solution for problem for the case of $y = x_1 + d$.

For the case when $y = f(x_1, d) = x_1 + d$, the problem of finding the tolerance solution is relatively easy to solve. Indeed, for a given value d , the set of possible values of $x_1 + d$ when $x_1 \in [\underline{x}_1, \bar{x}_1]$ is the interval

$$[\underline{x}_1, \bar{x}_1] + [d, d] = [\underline{x}_1 + d, \bar{x}_1 + d]. \quad (14)$$

Thus, the requirement that all the values from this interval be within the desired range $[\underline{y}, \bar{y}]$ mean that we should have

$$[\underline{x}_1 + d, \bar{x}_1 + d] \subseteq [\underline{y}, \bar{y}], \quad (15)$$

i.e.,

$$\underline{y} \leq \underline{x}_1 + d \text{ and } \bar{x}_1 + d \leq \bar{y}. \quad (16)$$

The conditions (16) can be reformulated as

$$\underline{y} - \underline{x}_1 \leq d \leq \bar{y} - \bar{x}_1. \quad (17)$$

Thus, the conclusion is that the correct range of the design parameter d is the interval

$$[\underline{y} - \underline{x}_1, \bar{y} - \bar{x}_1]. \quad (18)$$

For example, when $[\underline{x}_1, \bar{x}_1] = [1, 2]$ and $[\underline{y}, \bar{y}] = [4, 7]$, then we get the possible range $[\underline{d}, \bar{d}] = [4 - 1, 7 - 2] = [3, 5]$ of the design parameter d .

The operation that transforms the intervals \mathbf{x}_1 and \mathbf{y} for x_1 and y into the interval for d is called a *backcalculation*.

Case of no solution. It should be mentioned that in general, it is possible that this problem does not have a solution: e.g., if the width $\bar{y} - \underline{y}$ of the desired interval $[\underline{y}, \bar{y}]$ is smaller than the width $\bar{x}_1 - \underline{x}_1$ of the environmental interval $[\underline{x}_1, \bar{x}_1]$. In this case, we can still prove that if d is the allowed value, then $\underline{y} - \underline{x}_1 \leq d \leq \bar{y} - \bar{x}_1$, but since $\underline{y} - \underline{x}_1 > \bar{y} - \bar{x}_1$, there is no value d which would satisfy that inequality and therefore, the interval $[\underline{y} - \underline{x}_1, \bar{y} - \bar{x}_1]$ of possible solutions is simply empty.

We can say that the expression $[\underline{y} - \underline{x}_1, \bar{y} - \bar{x}_1]$ covers this case:

- indeed, the interval $[a, b]$ is usually defined as a set of all the numbers which are larger than or equal to a and smaller than or equal to b :

$$[a, b] \stackrel{\text{def}}{=} \{x : a \leq x \leq b\}; \quad (19)$$

- thus, for $a > b$, we can still define an “interval” $[a, b] = \{x : a \leq x \leq b\}$, but since no such x exists, the interval (19) will be simply an empty set.

For example, when $[\underline{y}, \bar{y}] = [5, 6]$ and $[\underline{x}_1, \bar{x}_1] = [1, 3]$, we get

$$\underline{y} - \underline{x}_1 = 5 - 1 = 4 > \bar{y} - \bar{x}_1 = 6 - 3 = 3, \quad (20)$$

hence the interval $[\underline{y} - \underline{x}_1, \bar{y} - \bar{x}_1] = [4, 3]$ of allowed values of d is empty – meaning that this design problem does not have solutions.

Backcalculation: additional cases. In the previous text, we considered the case when $y = x_1 + d$. A similar formula can be derived when the desired characteristic y is equal to the product $y = x_1 \cdot d$ of the environmental parameter $x_1 > 0$ and the design parameter $d > 0$. In the idealized exact case, we have $d = y/x_1$. In the interval case, once we know the range $[\underline{x}_1, \bar{x}_1]$ for x_1 and the desired range $[\underline{y}, \bar{y}]$ for y , we conclude that the interval of possible values of d is

$$[\underline{y}/\underline{x}_1, \bar{y}/\bar{x}_1]. \quad (21)$$

For other arithmetic expressions f , we can also get explicit formulas for the range of d :

- when $y = f(x_1, d) = x_1 - d$ (and $d = x_1 - y$), then the range of possible values of d is

$$[\bar{x}_1 - \bar{y}, \underline{x}_1 - \underline{y}]; \quad (22)$$

- when $y = f(x_1, d) = d - x_1$ (and $d = x_1 + y$), then the range of possible values of d is

$$[\bar{x}_1 + \underline{y}, \underline{x}_1 + \bar{y}]; \quad (23)$$

- when $y = f(x_1, d) = x_1/d$ (and $d = x_1/y$), then the range of possible values of d is

$$[\bar{x}_1/\bar{y}, \underline{x}_1/\underline{y}]; \quad (24)$$

- when $y = f(x_1, d) = d/x_1$ (and $d = x_1 \cdot y$), then the range of possible values of d is

$$[\bar{x}_1 \cdot \underline{y}, \underline{x}_1 \cdot \bar{y}]. \quad (25)$$

Modal intervals as a natural way to describe backcalculation. The above section provides a large number of different formulas, each of which can be deduced. However, in contrast to easier-to-understand interval computations, this deduction did not leads us to an easier general way to generate these formulas. Such a general way would be welcome. And it would be also desirable to have such general guidance on how to generate such formulas in more general situations.

Such a general approach was indeed discovered, under the name of *Kaucher arithmetic*, or, as it was called later, *modal interval analysis*; see, e.g., [5, 10, 11, 13]. To explain this approach, let us go back to the our very first example of a tolerance solution. In this first example, we have a dependence $d = y - x_1$ between the value y of the quality, the parameter x_1 characterizing the environment, and the parameter d which characterizes the design. We assume that:

- we know the interval $[\underline{y}, \bar{y}]$ of allowed values of y ,
- the interval $[\underline{x}_1, \bar{x}_1]$ of possible values of x_1 , and
- we want to find the interval $[\underline{d}, \bar{d}]$ of all allowed values of d – i.e., all the values for which, for every $x_1 \in [\underline{x}_1, \bar{x}_1]$, we have $y \in [\underline{y}, \bar{y}]$.

Our objective is to produce some description for the tolerance solution which is similar to the formulas of interval arithmetic. Our (“tolerance”) solution to this problem is the interval

$$[\underline{y} - \underline{x}_1, \bar{y} - \bar{x}_1]. \quad (26)$$

For the expression $d = y - x_1$, we can also describe what interval arithmetic would produce if we knew the interval $[\underline{y}, \bar{y}]$ of possible values of y and the interval $[\underline{x}_1, \bar{x}_1]$ of possible values of x_1 ; the resulting interval (6) has the following form:

$$[\underline{y} - \bar{x}_1, \bar{y} - \underline{x}_1]. \quad (27)$$

By comparing the two expressions (26) and (27), one can see that formally, the expression for the tolerance set can be obtained from the standard interval expression (27) if we simply (formally) replace the interval $[\underline{x}_1, \bar{x}_1]$ with an expression $[\bar{x}_1, \underline{x}_1]$.

Let us emphasize the word “formally” because, as we have mentioned in the previous section, if we simply plug in the values $a > b$ into a usual mathematical definition of an interval $[a, b]$ (in our case, $a = \bar{x}_1$ and $b = \underline{x}_1$), we simply get an empty set.

Formal expressions of the type $[\bar{x}_1, \underline{x}_1]$ are called *dual* intervals; we will also say that the dual interval $[\bar{x}_1, \underline{x}_1]$ is dual to the original interval $[\underline{x}_1, \bar{x}_1]$. Both original intervals and dual intervals are together called *modal intervals*, and techniques for processing such intervals is known as *modal interval analysis*.

The term “modal” comes from the fact that we need such intervals in design situations, when

- some intervals – in our case, the interval $[\underline{x}_1, \bar{x}_1]$ – describe the set of *possible* values; while
- some other intervals – in our case, the interval $[\underline{y}, \bar{y}]$ – describe the set of values which are *necessary* to achieve.

The area of mathematical logic which formalizes the notions of “possible” and “necessary” is known as *modal logic*; thus, the version of interval analysis which takes into account that some intervals describe possibility and some necessity was called modal interval analysis.

Backcalculation by using modal interval analysis: case of $y = x_1 + d$. With modal interval analysis, the solution to the tolerance problem $d = y - x_1$ looks as follows: once we have the intervals $[\underline{y}, \bar{y}]$ and $[\underline{x}_1, \bar{x}_1]$, we find the dual interval $[\bar{x}_1, \underline{x}_1]$ and use the standard interval arithmetic formula (6) for subtraction to compute the range of d as

$$[\underline{d}, \bar{d}] = [\underline{y}, \bar{y}] - [\bar{x}_1, \underline{x}_1] = [\underline{y} - \underline{x}_1, \bar{y} - \bar{x}_1]. \quad (28)$$

For example, when $[\underline{x}_1, \bar{x}_1] = [1, 2]$ and $[\underline{y}, \bar{y}] = [4, 7]$, we find the dual interval $[\bar{x}_1, \underline{x}_1] = [2, 1]$ and then compute the desired range of d as

$$[\underline{d}, \bar{d}] = [\underline{y}, \bar{y}] - [\bar{x}_1, \underline{x}_1] = [4, 7] - [2, 1] = [4 - 1, 7 - 2] = [3, 5]. \quad (29)$$

Backcalculation: additional cases of arithmetic functions $f(x_1, d)$. One may ask: is there any benefit in using this somewhat complicated and not-easy-to-understand way to compute the same tolerance solution – that we already know how to easily compute by backcalculation? If this reformulation only worked for this particular case $d = y - x_1$, then the above skepticism would be completely justified. The good news is that this same simple idea works for *all* cases for which we have described the backcalculation solution.

Thus, we do have a “magic” unified way of describing all these backcalculation solution: all we do is use the standard interval arithmetic (4), (6), (7), (8), but with a dual interval for x_1 . Indeed:

- when $d = y/x_1$, we get

$$[\underline{d}, \overline{d}] = [\underline{y}, \overline{y}]/[\overline{x_1}, \underline{x_1}] = [\underline{y}/\underline{x_1}, \overline{y}/\overline{x_1}]; \quad (30)$$

- when $d = x_1 - y$, we get

$$[\underline{d}, \overline{d}] = [\overline{x_1}, \underline{x_1}] - [\underline{y}, \overline{y}] = [\overline{x_1} - \overline{y}, \underline{x_1} - \underline{y}]; \quad (31)$$

- when $d = x_1 + y$, we get

$$[\underline{d}, \overline{d}] = [\overline{x_1}, \underline{x_1}] + [\underline{y}, \overline{y}] = [\overline{x_1} + \underline{y}, \underline{x_1} + \overline{y}]; \quad (32)$$

- when $d = x_1/y$, we get

$$[\underline{d}, \overline{d}] = [\overline{x_1}, \underline{x_1}]/[\underline{y}, \overline{y}] = [\overline{x_1}/\overline{y}, \underline{x_1}/\underline{y}]; \quad (33)$$

- when $d = x_1 \cdot y$, we get

$$[\underline{d}, \overline{d}] = [\overline{x_1}, \underline{x_1}] \cdot [\underline{y}, \overline{y}] = [\overline{x_1} \cdot \underline{y}, \underline{x_1} \cdot \overline{y}]. \quad (34)$$

Let us show that this technique works for the general case of a monotonic dependence as well – and moreover, it is true if we have several variables describing the environment.

Modal arithmetic as a general way to describe backcalculation for a monotonic dependence: a proof. Let x_1, \dots, x_m denotes variables describing the environment, let d denote the design parameter, and let y is a numerical measure of quality of the given design in the given environment.

We assume that we know how exactly the quality depends on the design d and on the environment, i.e., we know the algorithm f for which $y = f(x_1, \dots, x_m, d)$. We assume that with respect to each of the variables x_i , the dependence is monotonic: i.e., either increasing or decreasing. We also assume that with respect to d , the dependence is *strictly monotonic*, i.e., either strictly increasing or strictly decreasing.

For simplicity, let us start with the case when the function f is increasing with respect to each of the unknowns x_i and with respect to d .

If we knew the exact values x_1, \dots, x_m of the environmental variables, and the desired value y of the quality, monotonicity with respect to d enables us to conclude that there can be at most one value of d for which $y = f(x_1, \dots, x_m, d)$. Let us denote this value by $F(x_1, \dots, x_m, y)$.

Since the function f is increasing with respect to all its variables, we can conclude that the dependence $F(x_1, \dots, x_m, y)$ increases with respect to y and decreases with respect to all the environmental variables x_1, \dots, x_m .

Indeed, if we increase the value y without changing the values of the environmental variables, then we need to increase d to accordingly increase the value of $f(x_1, \dots, x_m, d)$. Thus, the dependence F of d on y is increasing.

On the other hand, for given y , if we increase the value of one of the environmental variables x_i without changing the values of all the other environmental variables, then the value of $y = f(x_1, \dots, x_m, d)$ increase. Thus, to get to the same value of the quality y , we must decrease the corresponding value d . So, the dependence F of d on x_i is indeed decreasing.

In the above particular cases, when $f(x_1, d)$ is a simple arithmetic operation, we have the following expressions for F :

- when $y = f(x_1, d) = x_1 + d$, we have $d = y - x_1$, so $F(x_1, y) = y - x_1$;
- when $y = f(x_1, d) = x_1 \cdot d$, we have $d = y/x_1$, so $F(x_1, y) = y/x_1$;
- etc.

In practice, we do not know the exact values of x_i and y . Instead, for each of the environmental variables x_1, \dots, x_m , we know the interval $[\underline{x}_i, \bar{x}_i]$ of its possible values, and we also know the interval $[\underline{y}, \bar{y}]$ of allowed values of the quality y . Our objective is to find all the values \underline{d} of the design parameter for which, for all possible values $x_i \in [\underline{x}_i, \bar{x}_i]$ of the environmental parameters, the resulting value y is within the allowed interval.

For each value d , the set of all possible values of $y = f(x_1, \dots, x_m, d)$ when $x_i \in [\underline{x}_i, \bar{x}_i]$ can be computed by using interval computations. Specifically, since the dependence of f on each of the variables x_1, \dots, x_m is increasing, this range is equal to $[f(\underline{x}_1, \dots, \underline{x}_m, d), f(\bar{x}_1, \dots, \bar{x}_m, d)]$. To check that all the values from this range are within the desired interval $[\underline{y}, \bar{y}]$, it is sufficient to check that the endpoints of this range are within the desired interval, i.e., that $\underline{y} \leq f(\underline{x}_1, \dots, \underline{x}_m, d)$ and $f(\bar{x}_1, \dots, \bar{x}_m, d) \leq \bar{y}$.

Since the function f is increasing with respect to d as well, the inequality $\underline{y} \leq f(\underline{x}_1, \dots, \underline{x}_m, d)$ is equivalent to $d \geq F(\underline{x}_1, \dots, \underline{x}_m, \underline{y})$, where, by the definition of the inverse function F , the value $\underline{d} \stackrel{\text{def}}{=} F(\underline{x}_1, \dots, \underline{x}_m, \underline{y})$ is the one for which $f(\underline{x}_1, \dots, \underline{x}_m, \underline{d}) = \underline{y}$.

Similarly, the inequality $f(\bar{x}_1, \dots, \bar{x}_m, d) \leq \bar{y}$ is equivalent to $d \leq F(\bar{x}_1, \dots, \bar{x}_m, \bar{y})$, where, by the definition of the inverse function F , the value $\bar{d} \stackrel{\text{def}}{=} F(\bar{x}_1, \dots, \bar{x}_m, \bar{y})$ is the one for which $f(\bar{x}_1, \dots, \bar{x}_m, \bar{d}) = \bar{y}$.

Thus, the range of allowed values of d has the form

$$[\underline{d}, \bar{d}] = [F(\underline{x}_1, \dots, \underline{x}_m, \underline{y}), F(\bar{x}_1, \dots, \bar{x}_m, \bar{y})]. \quad (35)$$

How does this compare with the result of applying interval computations to the function $F(x_1, \dots, x_m, y)$ over the intervals $[\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_m, \bar{x}_m], [\underline{y}, \bar{y}]$? Since the function F is decreasing with respect to x_1, \dots, x_m and increasing with respect to y , the interval range (3) is equal to

$$F([\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_m, \bar{x}_m], [\underline{y}, \bar{y}]) = [F(\bar{x}_1, \dots, \bar{x}_m, \underline{y}), F(\underline{x}_1, \dots, \underline{x}_m, \bar{y})]. \quad (36)$$

By comparing this expression (36) with the design-related interval (35) of possible values of d , we conclude that the interval (35) can be indeed obtained from the general range expression (36) if we formally replace each of the environmental intervals $[\underline{x}_i, \bar{x}_i]$ with the corresponding dual interval $[\bar{x}_i, \underline{x}_i]$:

$$\begin{aligned} [\underline{d}, \bar{d}] &= F([\bar{x}_1, \underline{x}_1], \dots, [\bar{x}_m, \underline{x}_m], [\underline{y}, \bar{y}]) = \\ &= [F(\underline{x}_1, \dots, \underline{x}_m, \underline{y}), F(\bar{x}_1, \dots, \bar{x}_m, \bar{y})]. \end{aligned} \quad (37)$$

We have proved this fact only for the case when the function $f(x_1, \dots, x_m, d)$ is increasing with respect to all of its variables x_1, \dots, x_m, d . However, one can check that a similar result holds also if it is decreasing with respect to some (or all) of these variables.

Example. Let us assume that the quality y is determined by the distance $y = \sqrt{x_1^2 + d^2}$ from the desired point $(0, 0)$ to the actual relative location (x_1, d) of the object with respect to the desired location relative to the space station. For simplicity, let us assume that in one direction, the deviation x_1 is determined exclusively by the environment (e.g., by the pressure of the atmosphere), and that the deviation $x_2 = d$ is determined exclusively by the design (i.e., by the design inaccuracy).

This function f is increasing with respect to each of its variables, so the above analysis is applicable. Here, $d = F(x_1, y) = \sqrt{y^2 - x_1^2}$ is a function which is increasing in y and decreasing in x_1 . Thus, for the standard interval range computation, we get

$$F([\underline{x}_1, \bar{x}_1], [\underline{y}, \bar{y}]) = [F(\bar{x}_1, \underline{y}), F(\underline{x}_1, \bar{y})] = \left[\sqrt{\underline{y}^2 - (\bar{x}_1)^2}, \sqrt{(\bar{y})^2 - \underline{x}_1^2} \right]. \quad (38)$$

Thus, if we know the interval $[0, \bar{y}]$ of allowed values of the distance and the interval $[0, \Delta]$ of possible environmental deviations, we can find the interval of allowed values d by substituting the dual interval $[\Delta, 0]$, with $\underline{x}_1 = \Delta$ and $\bar{x}_1 = -\Delta$, into the above formula. As a result, we get

$$[\underline{d}, \bar{d}] = \left[\sqrt{\underline{y}^2 - \underline{x}_1^2}, \sqrt{(\bar{y})^2 - (\bar{x}_1)^2} \right] = \left[0, \sqrt{(\bar{y})^2 - \Delta^2} \right]. \quad (39)$$

In particular, for $\bar{y} = 5$ and $\Delta = 4$, we get $[\underline{d}, \bar{d}] = [0, 3]$.

Reformulation without the use of modal intervals. Since modal intervals are a new (and somewhat counterintuitive) tool, it may be advantageous to reformulate the formula for the interval $[\underline{d}, \bar{d}]$ in terms that do not explicitly include modal intervals. This reformulation can be done as follows.

Let us recall the problem:

- we know the dependence $y = f(x_1, \dots, x_m, d)$; we know that this dependence is monotonic in terms of each of the variables and strictly monotonic in d ; the value d for which the above equality holds is denoted by $d = F(x_1, \dots, x_m, y)$;
- we know the desired interval $[\underline{y}, \bar{y}]$ of quality values, and
- we know the intervals $[\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_m, \bar{x}_m]$ that characterize the environment.

Our objective is now to describe the set $[\underline{d}, \bar{d}]$ of all the values d which have the following property:

- for all possible values $x_1 \in [\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_m, \bar{x}_m]$,
- the quality $y = f(x_1, \dots, x_m, d)$ is within the given interval $[\underline{y}, \bar{y}]$.

The solution to this problem is as follows. Since the original function f is monotonic with respect to each of its variables, the inverse function F is also monotonic with respect to each variable.

For each variable x_i ($1 \leq i \leq m$),

- if F is increasing in x_i , we define $x_i^- \stackrel{\text{def}}{=} \underline{x}_i$ and $x_i^+ \stackrel{\text{def}}{=} \bar{x}_i$;
- if F is decreasing in x_i , we define $x_i^- \stackrel{\text{def}}{=} \bar{x}_i$ and $x_i^+ \stackrel{\text{def}}{=} \underline{x}_i$.

Similarly,

- if F is increasing in y , we define $y^- \stackrel{\text{def}}{=} \underline{y}$ and $y^+ \stackrel{\text{def}}{=} \bar{y}$;
- if F is decreasing in y , we define $y^- \stackrel{\text{def}}{=} \bar{y}$ and $y^+ \stackrel{\text{def}}{=} \underline{y}$.

Then,

$$[\underline{d}, \bar{d}] = [F(x_1^+, \dots, x_m^+, y^-), F(x_1^-, \dots, x_m^-, y^+)]. \quad (40)$$

When tolerances are small, most dependencies are monotonic. The monotonicity requirement is not as restrictive as it may sound, because it is usually satisfied when the tolerances are small and the intervals are narrow. Indeed, in many practical situations, for each environmental parameter x_i , we know the approximate value \tilde{x}_i and we know that the possible deviations $\Delta x_i \stackrel{\text{def}}{=} x_i - \tilde{x}_i$ are small: $\Delta x_i \ll \tilde{x}_i$. Similarly, we know the desired value \tilde{y} of the quality, and we require that the deviations $\Delta y = y - \tilde{y}$ be small: $\Delta y \ll \tilde{y}$.

Since $x_1 \approx \tilde{x}_1, \dots, x_m \approx \tilde{x}_m$, and $y \approx \tilde{y}$, we can therefore conclude that, for every i , we have:

$$\frac{\partial F}{\partial x_i}(x_1, \dots, x_m, y) \approx \frac{\partial F}{\partial x_i}(\tilde{x}_1, \dots, \tilde{x}_m, \tilde{y}), \quad (41)$$

and thus, for all possible values, the partial derivative has the same sign as for the “nominal” values $\tilde{x}_1, \dots, \tilde{x}_m, \tilde{y}$. So:

- if the partial derivative at the nominal values is positive, it is everywhere positive and thus, the function F is increasing with respect to x_i ;
- if the partial derivative at the nominal values is negative, it is everywhere negative and thus, the function F is decreasing with respect to x_i .

In both cases, we conclude that the function F is monotonic with respect to x_i . Similarly, we can conclude that the function F is monotonic with respect to y .

This monotonicity can be explained in a different way: when the deviations Δx_i and Δy are small, we can apply the usual engineering linearization technique: namely, we can expand the value

$$F(x_1, \dots, x_m, y) = F(\tilde{x}_1 + \Delta x_1, \dots, \tilde{x}_m + \Delta x_m, \tilde{y} + \Delta y) \quad (42)$$

in Taylor series in terms of Δx_i and Δy and ignore quadratic and higher order terms in this expansion. The resulting linear dependence

$$F(x_1, \dots, x_m, y) = F(\tilde{x}_1 + \Delta x_1, \dots, \tilde{x}_m + \Delta x_m, \tilde{y} + \Delta y) \approx F(\tilde{x}_1, \dots, \tilde{x}_m, \tilde{y}) + \sum_{i=1}^m c_i \cdot \Delta x_i + c \cdot \Delta y, \quad (43)$$

where

$$c_i \stackrel{\text{def}}{=} \frac{\partial F}{\partial x_i}(\tilde{x}_1, \dots, \tilde{x}_m, \tilde{y}) \text{ and } c \stackrel{\text{def}}{=} \frac{\partial F}{\partial y}(\tilde{x}_1, \dots, \tilde{x}_m, \tilde{y}) \quad (44)$$

is clearly monotonic.

What if the dependence is not monotonic? If the dependence $f(x_1, \dots, x_m, d)$ is not monotonic with respect to some variables, the situation is much more complicated – just like it is more complicated when we find the interval range for a non-monotonic function. And just like interval arithmetic helps in estimating the range of non-monotonic functions (see, e.g., [6]), modal interval analysis helps in finding tolerance solutions in non-monotonic cases as well; see, e.g., [5, 10, 11] and especially [13].

From design of passive systems to design on active (controllable) systems. All above design examples were *passive* in the following sense: we assumed that the quality y of the resulting system is uniquely determined by the environment (parameters x_1, \dots, x_m) and by the design (parameter d). In these

examples, once the design has been selected, the quality is uniquely determined by the environment.

To explain this passivity, let us give a simple example. Suppose that for some special experiments, we want to maintain a certain orientation y of a spaceship relative to its orbit (a similar example can be produced by using torque and other characteristics). We want the corresponding angle y not to exceed a given value Δ , i.e., we want to make sure that $y \in [y, \bar{y}] = [-\Delta, \Delta]$. Let us assume that in the idealized environment, this orientation is determined by the design; let us denote the corresponding angle by x_2 .

In reality, external forces may slightly change the actual orientation. As a result, the actual angle y has the form $y = x_1 + x_2$, where x_1 denotes the angle deviation caused by the environment. In line with the above description, we assume that we know the interval $[x_1, \bar{x}_1] = [-\Delta_1, \Delta_1]$ of possible values of x_1 . In this example, we need to limit the original orientation x_2 is such a way that no matter what is the environmental contribution $x_1 \in [-\Delta_1, \Delta_1]$, the resulting spaceship orientation will always be within the given bounds.

Such an accurate passive design may be possible for a crude orientation, but for a very accurate orientation (e.g., if we want to perform some precise astronomical observations), we often need an accuracy which is even higher than the environmental deviations in the case of a perfect design: $\Delta \ll \Delta_1$. Clearly, the only way to achieve such an accurate orientation is to be *active*, i.e., to bring the space station to the desired position by actively rotating the spaceship.

Active systems: formulation of the design problem. For such active systems, the design problem becomes more complicated. For such systems, the quality y of the resulting system depends not only on the parameters x_1, \dots, x_m characterizing the environment and on the parameter d characterizing the design, it also depends on the parameters c_1, \dots, c_p describing the control. We assume that we know how the quality y depends on these parameters, i.e., that we know the dependence $y = f(x_1, \dots, x_m, d, c_1, \dots, c_p)$.

Similarly to the design of passive systems,

- we know the desired interval $[y, \bar{y}]$ of quality values, and
- we know the intervals $[x_1, \bar{x}_1], \dots, [x_m, \bar{x}_m]$ that characterize the environment;
- in addition, we also know the intervals $[c_1, \bar{c}_1], \dots, [c_p, \bar{c}_p]$ of possible values of control.

Our objective is now to describe all the values d of the design parameter which have the following property: no matter what the environment, we should be able to find controls for which the desired in within the desired range. To be more precise:

- for all possible values $x_1 \in [x_1, \bar{x}_1], \dots, [x_m, \bar{x}_m]$,

- there exist control parameters $c_1 \in [\underline{c}_1, \bar{c}_1], \dots, c_p \in [\underline{c}_p, \bar{c}_p]$
- for which the quality $y = f(x_1, \dots, x_m, d, c_1, \dots, c_p)$ is within the given interval $[\underline{y}, \bar{y}]$.

In interval computations, this is called a *control solution*. Following ideas outlined in [15], we will show that modal intervals can be helpful in computing the control solutions as well.

Example: gap-fitting problem and flexible materials. Let us start with an example analyzed in [15]. A typical example of a *passive* design problem in manufacturing is the following gap-fitting problem: We have a gap of a given size that is designed to contain several components. The total width y of all these components should not exceed the width of the gap. Due to manufacturing variability, the values of this gap may be slightly different. We want to make sure that the total width y is always smaller than or equal to the size of the gap. Thus, we must require that y does not exceed the smallest allowable gap width. Let us denote this upper bound for y by \bar{y} .

On the other hand, this total width cannot be too much smaller than the width, because then, these components may fall off; thus, a lower bound \underline{y} is also given, with the requirement that $y \geq \underline{y}$.

In general, the actual width must always be between the two given bounds \underline{y} and \bar{y} .

Some of these components have been pre-selected as off-the-shelf (OTS) components. For these components, we have no control over widths, but we know the bounds on their widths guaranteed by the manufacturer: we know that:

- the width x_1 of the first OTS component is within a given interval $[\underline{x}_1, \bar{x}_1]$;
- the width x_2 of the second OTS component is within a given interval $[\underline{x}_2, \bar{x}_2]$;
- ...
- and the width x_m of the m -th OTS component is within a given interval $[\underline{x}_m, \bar{x}_m]$.

If all the components are OTS, then the total width of all the components is equal to $y = x_1 + \dots + x_m$. In this case, the problem of estimating the range of possible values of y is a typical problem of interval computations, and so, the interval $[\underline{Y}, \bar{Y}]$ of possible values of the resulting range is equal to

$$[\underline{Y}, \bar{Y}] = [\underline{x}_1, \bar{x}_1] + \dots + [\underline{x}_m, \bar{x}_m] = [\underline{x}_1 + \dots + \underline{x}_m, \bar{x}_1 + \dots + \bar{x}_m]. \quad (45)$$

If this range $[\underline{Y}, \bar{Y}]$ is completely within the desired interval $[\underline{y}, \bar{y}]$, then our problem is solved. But in practice, such perfect fittings are rare. Often, to be able to provide a fit, we must specifically design at least one of the components.

If we denote the width of this component by d , then we have the previously discussed passive design problem:

- we have $y = x_1 + \dots + x_m + d$,
- we know the desired interval $[y, \bar{y}]$ for y ,
- we know the intervals $[x_1, \bar{x}_1], \dots, [x_m, \bar{x}_m]$ of possible values of the OTS component widths, and
- we must find all possible values d for which for all possible widths of the OTS components, the overall width is within the given range.

We already know how to solve this problem: the set of all such values d is the interval

$$[\underline{d}, \bar{d}] = [\underline{y} - \underline{x}_1 - \dots - \underline{x}_m, \bar{y} - \bar{x}_1 - \dots - \bar{x}_m]. \quad (46)$$

In some practical situations, this interval is non-empty, so we can indeed solve the gap-fitting problem by designing an appropriate component. However, often, the widths of the OTS components vary a lot; as a result, the interval of possible values of $x_1 + \dots + x_m$ is very wide, and no matter what d we add, we will not be able to guarantee that the result is always within the (relatively narrow) desired interval $[y, \bar{y}]$.

For example, in a realistic case when we have two OTS components with 10% width uncertainty

$$[\underline{x}_1, \bar{x}_1] = [\underline{x}_2, \bar{x}_2] = [1 - 0.1, 1 + 0.1] = [0.9, 1.1], \quad (47)$$

and for the desired overall width $[2.9, 3.0]$, the above formula produces an empty interval

$$\begin{aligned} [\underline{d}, \bar{d}] &= [\underline{y} - \underline{x}_1 - \dots - \underline{x}_m, \bar{y} - \bar{x}_1 - \dots - \bar{x}_m] = \\ &= [2.9 - 0.9 - 0.9, 3.0 - 1.1 - 1.1] = [1.1, 0.8], \end{aligned} \quad (48)$$

meaning that in this formulation, the fitting problem cannot be solved.

To provide a fit, we allow *flexible* components: e.g., we may add spring-type components which can shrink to cover the desired gap, or components which are somewhat flexible so that they can be hammered in even if they are somewhat larger.

For instance, in the above example, we can add a spring-type flexible component whose width c_1 can take any value from 0 to 0.5. For such a component, we can take, e.g., $d = x_3 = 0.7$. In this case, no matter what values $x_1 \in [0.1, 1.1]$ and $x_2 \in [0.9, 1.1]$ are, the sum $x_1 + x_2 + d$ is always in the interval $[0.9, 1.1] + [0.9, 1.1] + 0.7 = [2.5, 2.9]$. Thus, by taking $c_1 = 3.0 - x_1 - x_2 - d \in 3 - [2.5, 2.9] = [0.1, 0.5]$, we can not only guarantee that the resulting width $y = x_1 + x_2 + d + c_1$ is within the desired interval $[2.9, 3.0]$ – we can actually guarantee the perfect fit $y = 3.0$.

General case of a monotonic dependence: description. In the flexible fit example, the dependence of y on the parameters x_i , d , and c_j was linear – and therefore, monotonic with respect to each of the variables.

Let us now consider the general case when this dependence is monotonic, i.e., when the function $f(x_1, \dots, x_m, d, c_1, \dots, c_p)$ is monotonic (strictly increasing or strictly decreasing) with respect to each of the variables.

In the idealized case when we know the exact values of all the parameters x_i , c_j , and y , there exists at most one value d for which $y = f(x_1, \dots, x_m, d, c_1, \dots, c_p)$. This value d will be denoted by $F(x_1, \dots, x_m, y, c_1, \dots, c_p)$.

Design of an active system: derivation of the formulas. Similarly to the case of passive design, let us first consider the case when the function $f(x_1, \dots, x_m, d, c_1, \dots, c_p)$ is increasing with respect to all its variables x_i , d , and c_j .

In this case, similar to the passive design, the function

$$F(x_1, \dots, x_m, y, c_1, \dots, c_p) \quad (49)$$

is increasing in y and decreasing in all other variables x_i and c_j .

We must select the parameter d from the condition that for each combination of possible values $x_i \in [\underline{x}_i, \bar{x}_i]$, there should be controls $c_j \in [\underline{c}_j, \bar{c}_j]$ for which $f(x_1, \dots, x_m, d, c_1, \dots, c_p) \in [y, \bar{y}]$.

Let us first consider the case when a certain value d is selected and certain values $x_i \in [\underline{x}_i, \bar{x}_i]$ are also selected. In this case, we need

- to find the range of possible values of $f(x_1, \dots, x_m, d, c_1, \dots, c_p)$ when $c_j \in [\underline{c}_j, \bar{c}_j]$, and
- to make sure that this range has at least one common point with the desired interval $[y, \bar{y}]$.

Since the function $f(x_1, \dots, x_m, d, c_1, \dots, c_p)$ is increasing with respect to all the parameters c_j , we can use monotonic interval computations (3) to provide an explicit expression for the range of y :

$$[f(x_1, \dots, x_m, d, \underline{c}_1, \dots, \underline{c}_p), f(x_1, \dots, x_m, d, \bar{c}_1, \dots, \bar{c}_p)]. \quad (50)$$

We need to check that this interval has a common point with the desired interval $[y, \bar{y}]$.

In general, the two intervals $[\underline{a}, \bar{a}]$ and $[\underline{b}, \bar{b}]$ have a common point if $\underline{a} \leq \bar{b}$ and $\underline{b} \leq \bar{a}$. Indeed:

- If the two intervals do have a common point c , then:
 - $\underline{a} \leq c$ (since $c \in [\underline{a}, \bar{a}]$) and $c \leq \bar{b}$ (since $c \in [\underline{b}, \bar{b}]$) and thus, by transitivity, $\underline{a} \leq \bar{b}$;
 - similarly, one can show that $\underline{b} \leq \bar{a}$.
- Vice versa, if $\underline{a} \leq \bar{b}$ and $\underline{b} \leq \bar{a}$, then for $c \stackrel{\text{def}}{=} \max(\underline{a}, \underline{b})$, we have:
 - $\underline{a} \leq \max(\underline{a}, \underline{b}) = c$,

- and $c \leq \bar{a}$ follows from the fact that $\underline{a} \leq \bar{a}$ and $\underline{b} \leq \bar{a}$ (by assumption).

Thus, $c \in [\underline{a}, \bar{a}]$. Similarly, we can prove that $c \in [\underline{b}, \bar{b}]$ and thus, c is indeed a common point of the two given intervals.

So, the interval (50) and $[y, \bar{y}]$ have a common point if and only if the following two inequalities are satisfied:

$$f(x_1, \dots, x_m, d, \underline{c}_1, \dots, \underline{c}_p) \leq \bar{y} \quad (51)$$

and

$$\underline{y} \leq f(x_1, \dots, x_m, d, \bar{c}_1, \dots, \bar{c}_p). \quad (52)$$

The first inequality (51) must be satisfied for all possible values $x_i \in [\underline{x}_i, \bar{x}_i]$. For this to be true, the largest possible values of $f(x_1, \dots, x_m, d, \underline{c}_1, \dots, \underline{c}_p)$ (when $x_i \in [\underline{x}_i, \bar{x}_i]$) must satisfy the inequality (51). The function f is increasing with respect to x_i ; thus, its largest value is attained when each of the parameters x_i attained its largest possible value \bar{x}_i . So, the fact that the inequality (51) is satisfied for all possible values $x_i \in [\underline{x}_i, \bar{x}_i]$ is equivalent to a single inequality

$$f(\bar{x}_1, \dots, \bar{x}_m, d, \underline{c}_1, \dots, \underline{c}_p) \leq \bar{y}. \quad (53)$$

Since the function f is increasing, this inequality is equivalent to

$$d \leq \bar{d} \stackrel{\text{def}}{=} F(\bar{x}_1, \dots, \bar{x}_m, \bar{y}, \underline{c}_1, \dots, \underline{c}_p), \quad (54)$$

where, by definition of the inverse function F , the number \bar{d} is the value for which $f(\bar{x}_1, \dots, \bar{x}_m, \bar{d}, \underline{c}_1, \dots, \underline{c}_p) = \bar{y}$.

Similarly, the second inequality (52) must be satisfied for all possible values $x_i \in [\underline{x}_i, \bar{x}_i]$. For this to be true, the smallest possible values of $f(x_1, \dots, x_m, d, \bar{c}_1, \dots, \bar{c}_p)$ (when $x_i \in [\underline{x}_i, \bar{x}_i]$) must satisfy the inequality (52). The function f is increasing with respect to x_i ; thus, its smallest value is attained when each of the parameters x_i attained its smallest possible value \underline{x}_i . So, the fact that the inequality (52) is satisfied for all possible values $x_i \in [\underline{x}_i, \bar{x}_i]$ is equivalent to a single inequality

$$\underline{y} \leq f(\underline{x}_1, \dots, \underline{x}_m, d, \bar{c}_1, \dots, \bar{c}_p). \quad (55)$$

Since the function f is increasing, this inequality is equivalent to

$$d \geq \underline{d} \stackrel{\text{def}}{=} F(\underline{x}_1, \dots, \underline{x}_m, \underline{y}, \bar{c}_1, \dots, \bar{c}_p), \quad (56)$$

where, by definition of the inverse function F , the number \underline{d} is the value for which $f(\underline{x}_1, \dots, \underline{x}_m, \underline{d}, \bar{c}_1, \dots, \bar{c}_p) = \underline{y}$.

By combining the inequalities (54) and (56), we conclude that the range of possible values of d is the interval

$$[\underline{d}, \bar{d}] = [F(\underline{x}_1, \dots, \underline{x}_m, \underline{y}, \bar{c}_1, \dots, \bar{c}_p), F(\bar{x}_1, \dots, \bar{x}_m, \bar{y}, \underline{c}_1, \dots, \underline{c}_p)]. \quad (57)$$

To see how this expression is related to modal intervals, let us compare this result with the range

$$\mathbf{F} \stackrel{\text{def}}{=} F([\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_m, \bar{x}_m], [\underline{y}, \bar{y}], [\underline{c}_1, \bar{c}_1], \dots, [\underline{c}_p, \bar{c}_p]) \quad (58)$$

of the same function F on the same intervals obtained by interval computations. We considered the case when the function f is increasing in all the variables x_i , d , and c_j and thus, the function F is increasing with respect to y and decreasing with respect to x_i and c_j . Thus, due to the formula (3), the range is equal to

$$\mathbf{F} = [F(\bar{x}_1, \dots, \bar{x}_m, \underline{y}, \bar{c}_1, \dots, \bar{c}_p), F(\underline{x}_1, \dots, \underline{x}_m, \bar{y}, \underline{c}_1, \dots, \underline{c}_p)]. \quad (59)$$

So, we conclude that to get the desired set (57), we must use the normal intervals for the quality y and for the control parameters c_j , but dual variables for the environmental variables x_i :

$$[\underline{d}, \bar{d}] = F([\bar{x}_1, \underline{x}_1], \dots, [\bar{x}_m, \underline{x}_m], [\underline{y}, \bar{y}], [\underline{c}_1, \bar{c}_1], \dots, [\underline{c}_p, \bar{c}_p]). \quad (60)$$

One can show that a similar formula holds in the general case, when the dependence f may be decreasing in some of the variables.

Resulting algorithm: use of modal interval analysis. Let us recall the problem:

- we know the dependence $y = f(x_1, \dots, x_m, d, c_1, \dots, c_p)$; we know that this dependence is monotonic (strictly increasing or strictly decreasing) in terms of each of the variables; the value d for which the above equality holds is denoted by $d = F(x_1, \dots, x_m, y, c_1, \dots, c_p)$;
- we know the desired interval $[\underline{y}, \bar{y}]$ of quality values, and
- we know the intervals $[\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_m, \bar{x}_m]$ that characterize the environment;
- in addition, we also know the intervals $[\underline{c}_1, \bar{c}_1], \dots, [\underline{c}_p, \bar{c}_p]$ of possible values of control.

Our objective is now to describe the set $[\underline{d}, \bar{d}]$ of all the values d which have the following property:

- for all possible values $x_1 \in [\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_m, \bar{x}_m]$,
- there exist control parameters $c_1 \in [\underline{c}_1, \bar{c}_1], \dots, c_p \in [\underline{c}_p, \bar{c}_p]$
- for which the quality $y = f(x_1, \dots, x_m, d, c_1, \dots, c_p)$ is within the given interval $[\underline{y}, \bar{y}]$.

The solution to this problem is as follows: we take the original intervals for y and c_j and the dual intervals $[\bar{x}_i, \underline{x}_i]$ for x_i , and apply interval computations to these intervals:

$$[\underline{d}, \bar{d}] = F([\bar{x}_1, \underline{x}_1], \dots, [\bar{x}_m, \underline{x}_m], [\underline{y}, \bar{y}], [\underline{c}_1, \bar{c}_1], \dots, [\underline{c}_p, \bar{c}_p]). \quad (61)$$

Example. In the above example, we have:

- $y = f(x_1, x_2, d, c_1) = x_1 + x_2 + d + c_1$ hence

$$d = F(x_1, x_2, y, c_1) = y - x_1 - x_2 - c_1; \quad (62)$$

- $[\underline{x}_1, \bar{x}_1] = [\underline{x}_2, \bar{x}_2] = [0.9, 1.1]$;
- $[\underline{y}, \bar{y}] = [2.9, 3.0]$; and
- $[\underline{c}_1, \bar{c}_1] = [0, 0.5]$.

For this function F , we the formula for the interval computations range has the form

$$\begin{aligned} F([\underline{x}_1, \bar{x}_1], [\underline{x}_2, \bar{x}_2], [\underline{y}, \bar{y}], [\underline{c}_1, \bar{c}_1]) &= [\underline{y}, \bar{y}] - [\underline{x}_1, \bar{x}_1] - [\underline{x}_2, \bar{x}_2] - [\underline{c}_1, \bar{c}_1] = \\ &= [\underline{y} - \bar{x}_1 - \bar{x}_2 - \bar{c}_1, \bar{y} - \underline{x}_1 - \underline{x}_2 - \underline{c}_1]. \end{aligned} \quad (63)$$

In our case, for y and c_1 , we take the original intervals, and for x_i , we take dual intervals $[1.1, 0.9]$. For this combination of intervals, we get the following set

$$\begin{aligned} F([1.1, 0.9], [1.1, 0.9], [2.9, 3.0], [0, 0.5]) &= \\ [2.9, 3.0] - [1.1, 0.9] - [1.1, 0.9] - [0, 0.5] &= \\ [2.9 - 0.9 - 0.9 - 0.5, 3.0 - 1.1 - 1.1 - 0] &= [0.6, 0.8]. \end{aligned} \quad (64)$$

Reformulation without the use of modal intervals. Since, as we have mentioned, modal intervals are a new (and somewhat counterintuitive) tool, it may be advantageous to reformulate the formula for the interval $[d, \bar{d}]$ in terms that do not explicitly include modal intervals. This reformulation can be done as follows.

For each variable x_i ($1 \leq i \leq m$),

- if F is increasing in x_i , we define $x_i^- \stackrel{\text{def}}{=} \underline{x}_i$ and $x_i^+ \stackrel{\text{def}}{=} \bar{x}_i$;
- if F is decreasing in x_i , we define $x_i^- \stackrel{\text{def}}{=} \bar{x}_i$ and $x_i^+ \stackrel{\text{def}}{=} \underline{x}_i$.

Similarly,

- if F is increasing in y , we define $y^- \stackrel{\text{def}}{=} \underline{y}$ and $y^+ \stackrel{\text{def}}{=} \bar{y}$;
- if F is decreasing in y , we define $y^- \stackrel{\text{def}}{=} \bar{y}$ and $y^+ \stackrel{\text{def}}{=} \underline{y}$.

For each variable c_j ($1 \leq j \leq p$),

- if F is increasing in c_j , we define $c_j^- \stackrel{\text{def}}{=} \underline{c}_j$ and $c_j^+ \stackrel{\text{def}}{=} \bar{c}_j$;
- if F is decreasing in c_j , we define $c_j^- \stackrel{\text{def}}{=} \bar{c}_j$ and $c_j^+ \stackrel{\text{def}}{=} \underline{c}_j$.

Then,

$$[d, \bar{d}] = [F(x_1^+, \dots, x_m^+, y^-, c_1^-, \dots, c_p^-), F(x_1^-, \dots, x_m^-, y^+, c_1^+, \dots, c_p^+)]. \quad (65)$$

Comment. In the above analysis, we made a simplifying assumption that we can always implement the exact value of the desired control. In real life, of course, actuators and controllers are not perfect, the control can only be implemented with a certain accuracy. The inaccuracy of the control implementation must also be taken into account in the design.

From the computational viewpoint, the parameters describing corresponding inaccuracies can be treated as additional environmental parameters.

2 Part II: Combination of Modal Intervals with Probabilistic Uncertainty

In Part I, we describe modal interval analysis and its applications to situations when we have no information about the probabilities. In Part II, we will show how modal interval analysis can be extended to situations when we do have some partial information about the probabilities.

It turns out that, in contrast to the no-probabilities case where modal intervals have been successful in many different applications, in the probabilities case the only promising application is to take control into account.

From intervals to p-boxes. As we have mentioned earlier, traditional engineering approach assumes that we know the exact probability distribution of the desired parameter. In practice, we do not have complete information about these probabilities.

Sometimes, we only know the bounds \underline{x} and \bar{x} on each quantity x : $\underline{x} \leq x \leq \bar{x}$. In these situations, we know an *interval* $[\underline{x}, \bar{x}]$ that contains the actual (unknown) value x . In Part I, we analyzed how to solve the problem of early engineering design under such interval uncertainty.

In practice, in addition to the bounds, we often have some *partial* information about the probabilities. How to represent this partial information? In principle, there are many different ways to represent a probability distribution: we can use a probability density function, a cumulative distribution function, moments of different order, characteristic functions, etc.; see, e.g., [12]. Different representations have their advantages and disadvantages. In each practical application, it is reasonable to select a representation that is the most adequate for this particular application.

For the design problems, as we have mentioned, one of the most important requirements is that the values of the important characteristics are within the required range. For example, in a spaceflight, the level of radiation or the acceleration should never exceed the threshold after which the corresponding exposure can be damaging to the astronauts' health.

In practice, it is not possible to guarantee any such bounds with 100% reliability: no matter how reliable the systems are, there is always a small but positive probability that systems and devices may fail or malfunction. In this case, it is important to make sure that the probability of exceeding the critical

threshold is as small as possible. Since the probability of exceeding the critical potential-health-damaging threshold cannot be made exactly zero, it is important to guarantee at least that the probability of exceeding the larger critical threshold, e.g., of irreversible health damage or even a lethal effect, is as small as possible.

In all such situation, it is important to know the probability $\text{Prob}(x > t)$ of exceeding different thresholds t . This probability has a straightforward relation to the cumulative distribution function $F(t) \stackrel{\text{def}}{=} \text{Prob}(X \leq t)$, namely

$$\text{Prob}(x > t) = 1 - F(t). \quad (66)$$

Thus, in design problems, it is important to know the cumulative distribution function (cdf) $F(t)$.

In real life, we usually have a partial information about the probability distribution, and thus, only a partial information about the cdf $F(t)$. Thus, for each real number t , instead of the exact value $F(t)$, we usually only know the bounds $\underline{F}(t)$ and $\overline{F}(t)$ for which $\underline{F}(t) \leq F(t) \leq \overline{F}(t)$. In this case, the only information that we have about the actual (unknown) cumulative distribution function $F(t)$ is that $F(t) \in [\underline{F}(t), \overline{F}(t)]$. The set of all such probability distributions is called a *probability box*, or a *p-box*, for short.

Alternative representation: quantiles and quantile intervals. Alternatively, we can view the same design situation in a somewhat different way. As we have mentioned, ideally, desirable situations should occur with probability 100%, but in practice, the probability of the desirable situation cannot be made equal exactly to 1. Usually, depending on the level of undesirability, the users set up probabilities of not exceeding a threshold; for example, there are allowable probabilities of risk in spaceflight planning:

- a higher one for unmanned flights and
- a much lower one on the manned flights.

From this viewpoint, it makes sense to fix one or several probabilities p and to consider thresholds which can be guaranteed with these allowed probabilities. In other words, we want to know the values $t(p)$ for which, for different probabilities p , we have $\text{Prob}(x \leq t) = F(t) = p$. The corresponding dependence $t(p)$ is an inverse function to cdf $F(t)$ in the sense that $t = t(p)$ if and only if $p = F(t)$ – in the same way as

- $x = \log(y)$ is the inverse function to $y = \exp(x)$,
- $x = \sqrt{y}$ is the inverse function to $y = x^2$, and
- $x = \arcsin(y)$ is the inverse function to $y = \sin(x)$.

The values of these inverse function are known as *quantiles*:

- for $p = 0.5$, we have a median;

- for $p = 0.25$ and $p = 0.75$, we have *quartiles*,
- etc.

From this viewpoint, a natural representation of a probability distribution is via the quantiles.

Since we only have a partial information about the probability distribution, we do not know the exact values $t(p)$ of the quantiles, we only know intervals of $[\underline{t}(p), \bar{t}(p)]$ of possible values.

Mathematical viewpoint: p-boxes are equivalent to quantile intervals.

From the mathematical viewpoint, this quantile representation is equivalent to the p-box representation. Indeed, since the function $t(p)$ is an inverse function to $F(t)$, for each p :

- the *smallest* possible value $\underline{t}(p)$ of $t(p)$ corresponds to the *largest* possible value $\bar{F}(t)$ of the inverse function $F(t)$, and
- the *largest* possible value $\bar{t}(p)$ of $t(p)$ corresponds to the *smallest* possible value $\underline{F}(t)$ of the inverse function $F(t)$.

Thus:

- the values $\underline{t}(p)$ are quantiles for the cdf $\bar{F}(t)$, while
- the values $\bar{t}(p)$ are quantiles for the cdf $\underline{F}(t)$.

Computational viewpoint: quantile interval representations are, in general, more efficient.

A natural way to represent a generic function $f(x)$ in a computer is to store its values $f(x_i)$ at different points – usually, at points forming a uniform grid: $x_0, x_1 = x_0 + h, x_2 = x_0 + 2h, \dots, x_k = x_0 + k \cdot h$.

From this viewpoint, the above two mathematically equivalent representations of a cdf $F(t)$ (and of a p-box $[\underline{F}(t), \bar{F}(t)]$) lead to the following two computationally different computer representations:

- if we start with the the cdf $F(t)$, then we store the values $F(t_0), F(t_1), \dots, F(t_k), \dots$ (or the intervals $[\underline{F}(t_0), \bar{F}(t_0)], [\underline{F}(t_1), \bar{F}(t_1)], \dots, [\underline{F}(t_k), \bar{F}(t_k)], \dots$) corresponding to different thresholds $t_0, t_1 = t_0 + h, \dots, t_k = t_0 + k \cdot h, \dots$
- if we start with the quantile representation $t(p)$, then we end up with values $t_0 = t(0), t_1 = t(1/n), t_2 = t(2/n), \dots$ (or the intervals $[\underline{t}_0, \bar{t}_0], [\underline{t}_1, \bar{t}_1], [\underline{t}_2, \bar{t}_2], \dots$).

One can see that for the same number of parameters, the second representation provides a much better picture of a distribution than the first one.

For example, in the case when we have an almost full certainty, i.e., a distribution which is located in a small vicinity of a single point x , then

- the quantile representation gives us this point x as a median, while

- in the first representation, we will only know that this x is somewhere between the consequent values t_i and t_{i+1} for which $F(t_i) = 0$ and $F(t_{i+1}) = 1$.

Similarly, when we have an interval uncertainty, i.e., when we only know that the actual value x is somewhere on the interval $[\underline{x}, \bar{x}]$ but we have no information on the probabilities of different values $x \in [\underline{x}, \bar{x}]$, then the interval $[\underline{F}(t), \overline{F}(t)]$ changes:

- from $[\underline{F}(t), \overline{F}(t)] = [0, 0]$ for $t < \underline{x}$
- to $[\underline{F}(t), \overline{F}(t)] = [0, 1]$ for $\underline{x} \leq x \leq \bar{x}$
- to $[\underline{F}(t), \overline{F}(t)] = [1, 1]$ for $t > \bar{x}$.

In this case,

- the cdf representation merely tells us where \underline{x} and \bar{x} are in relation to the values t_i , while
- the median interval gives us exactly the interval $[\underline{x}, \bar{x}]$.

In view of this computational advantage, p-boxes are usually represented in the computer as quantile intervals; see, e.g., [2].

Comment: how to future improve the computational representation of a p-box. The existing computer representation of a p-box is via the interval $[\underline{x}_i, \bar{x}_i]$ of possible values of quantiles corresponding to probabilities i/n , with $i = 0, 1, \dots, n$.

For $n = 10$, we thus represent a p-box by 11 intervals, a representation that describes all the probabilities with accuracy $\approx 10\%$. This is a very small amount of information, taking a small space in computer memory and easy- and fast-to-process.

In some practical applications, a 10% accuracy in describing probabilities is reasonable. In many design problems, however, we want to estimate the probabilities of low-probability events, with probabilities as low as 1%, 0.1%, or even 0.001%. To describe such probabilities, we need to use larger values of n . For example:

- to get the accuracy of 1%, we need to use $n = 100$;
- to get the accuracy of 0.1%, we need to use $n = 10^3$;
- to get the accuracy of 0.001%, we need to use $n = 10^5$.

In the existing representation, this means that we need $n+1$ intervals (and thus, twice as many real numbers) to represent each p-box.

This is still doable, especially for early design problems – where there is no pressure to deliver results in real time. However, the fact that we are looking at a factor of 10^4 increase in the memory size – and thus, at a similar increase

in computation times – shows that there is a need to decrease the number of values and thus, speed up the computations.

A natural way to such a decrease comes from the fact that while we want to know small probabilities with a high accuracy, there is no need to compute the reasonable-size probabilities like 50% with a similar accuracy. Thus, instead of using the quantiles $[\underline{p}(r), \bar{p}(r)]$ corresponding to *all* the probability values $r = 0, 1/n, 2/n, \dots$, it is sufficient to use the quantiles corresponding to only *some* of these values. For example, when we are interested in the accuracy of 0.001%, instead of storing all 10^5 values for all $r = i/n$, it makes sense to store only the quantiles corresponding to the following $5 \cdot 9 + 2 = 47$ values:

- value $r = 0$;
- 9 values 0.001%, 0.002%, ..., 0.009%;
- 9 values 0.01%, 0.002%, ..., 0.09%;
- 9 values 0.1%, 0.2%, ..., 0.9%;
- 9 values 1%, 2%, ..., 9%;
- 9 values 10%, 20%, ..., 90%;
- value $r = 1$.

In this manner, we can get all the probabilities with a reasonable relative accuracy.

In principle, the existing algorithms can be easily adjusted for this representation, because, due to the monotonicity, the “skipped” values can be automatically reconstructed from the existing ones. For example, if we store the values $[\underline{p}(r_1), \bar{p}(r_1)]$ and $[\underline{p}(r_2), \bar{p}(r_2)]$ for $r_1 = i_1/n$ and $r_2 = i_2/n$ and skip all the intermediate values $r = i/n$ with $i_1 < i < i_2$, then due to monotonicity, we know that

$$\underline{p}(r_1) \leq \underline{p}(r) \leq p(r) \leq \bar{p}(r) \leq \bar{p}(r_2). \quad (67)$$

Thus, we can know that $p(r) \in [\underline{p}(r_1), \bar{p}(r_2)]$. So, we can use the interval $[\underline{p}(r_1), \bar{p}(r_2)]$ as the interval quantiles corresponding to each missing value r .

Of course, to gain the desired speed-up, we must accordingly transform all the algorithms which are currently used for processing p-boxes.

Towards the use of p-boxes in design: simple analysis case. Let us start with the simplest case when:

- only one parameter x_1 described the environment,
- only one parameter d described the design, and
- the quality $y = f(x_1, d)$ of the design is described by a simple formula $y = x_1 + d$.

We will start with the *analysis* problem, when

- we know a p-box for x_1 ,
- we have already selected a p-box corresponding to d , and
- we want to describe the resulting p-box corresponding to y .

It is worth mentioning that this case is more general than it may seem at first glance, for two reasons:

- In some cases, the dependence $y = f(x_1, d)$ has the form $y = x_1 \cdot d$ or a similar form; such a dependence can be reduced to the sum if we use a logarithmic scale: $Y = X_1 + D$, where $y \stackrel{\text{def}}{=} \ln(y)$, $X_1 \stackrel{\text{def}}{=} \ln(x_1)$, and $D \stackrel{\text{def}}{=} \ln(d)$.
- In other cases, the intervals of possible values of x_1 and d are narrow. In this case, we can expand the dependence f in Taylor series and only keep linear terms in this expansion, resulting in a linear dependence $y \approx y_0 + c_1 \cdot x_1 + c_d \cdot d$. By an appropriate linear re-scaling, we can also reduce this expression to the case of a simple sum: $Y = X_1 + D$, where $Y \stackrel{\text{def}}{=} y - y_0$, $X_1 \stackrel{\text{def}}{=} c_1 \cdot x_1$, and $D \stackrel{\text{def}}{=} c_d \cdot d$.

For the case of interval uncertainty, we assumed that there are no constraints between x_1 and d , i.e., potentially high values of x_1 may be correlated with high values of d or with low values of d , or not correlated at all. Similarly, it is reasonable to start with the assumption that we have no information about the possible dependence between the probability distributions of x_1 and d . (If we have additional information about the dependence, e.g., if we know that x_1 and d are independent, then we get narrower bounds on y [2].)

Simplest case of p-boxes in design analysis: formulas for \underline{y}_i and their justification. For the case of unknown dependence, we want to transform the interval quantiles $[\underline{q}_i, \bar{q}_i]$ for x_1 and the interval quantiles $[\underline{d}_i, \bar{d}_i]$ for d into interval quantiles $[\underline{y}_i, \bar{y}_i]$ for y . The formulas for describing \underline{y}_i and \bar{y}_i were first deduced in [17].

Specifically, for computing \underline{y}_i , we have the following formula:

$$\underline{y}_i = \max_j (\underline{q}_{i-j} + \underline{d}_j). \quad (68)$$

This formula may sound strange, but it actually has a good intuitive explanation. Indeed, let us fix some i and $j \leq i$. By definition of a quantile, the value \underline{q}_{i-j} means that

$$\bar{F}_{x_1}(\underline{q}_{i-j}) = \frac{i-j}{n}, \quad (69)$$

i.e., that for the actual (unknown) value $F_{x_1}(\underline{q}_{i-j}) = \text{Prob}(x_1 \leq \underline{q}_{i-j})$, we have

$$\text{Prob}(x_1 \leq \underline{q}_{i-j}) = F_{x_1}(\underline{q}_{i-j}) \leq \bar{F}(\underline{q}_{i-j}) = \frac{i-j}{n}. \quad (70)$$

The probability $\text{Prob}(x_1 > \underline{q}_{i-j})$ of the complement event, that x_1 exceeds the value \underline{q}_{i-j} , is equal to $1 - \text{Prob}(x_1 \leq \underline{q}_{i-j})$. Thus, for this exceeding probability, we get the inequality

$$\text{Prob}(x_1 > \underline{q}_{i-j}) \geq 1 - \frac{i-j}{n}. \quad (71)$$

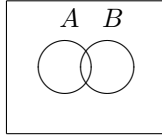
Similarly, from the fact that the lower bound for the j -th quantile for d is \underline{d}_j , we conclude that

$$\text{Prob}(d > \underline{d}_j) \geq 1 - \frac{j}{n}. \quad (72)$$

If both equalities $x_1 > \underline{q}_{i-j}$ and $d > \underline{d}_j$ are satisfied, then we have $y = x_1 + d > \underline{q}_{i-j} + \underline{d}_j$. The probability that both equalities are satisfied can be bounded from below if we take into account that for any two events A and B , we have

$$\text{Prob}(A \vee B) = \text{Prob}(A) + \text{Prob}(B) - \text{Prob}(A \& B). \quad (73)$$

This formula can be easily deduced from the corresponding Venn diagram



From this diagram, one can see that when we add $\text{Prob}(A)$ and $\text{Prob}(B)$, we count all the cases when A happened and all the cases when B happened – and we count each cases when both A and B happened twice. So, if we subtract from the sum $\text{Prob}(A) + \text{Prob}(B)$ all the events when both A and B occur, we thus count all the cases when A or B holds – i.e., we get exactly $\text{Prob}(A \vee B)$.

Formula (73) implies that

$$\text{Prob}(A \& B) = \text{Prob}(A) + \text{Prob}(B) - \text{Prob}(A \vee B). \quad (74)$$

Since every probability is bounded by 1, we have $\text{Prob}(A \vee B) \leq 1$ and thus,

$$\text{Prob}(A \& B) \geq \text{Prob}(A) + \text{Prob}(B) - 1. \quad (75)$$

In particular, in the above case, we conclude that

$$\begin{aligned} \text{Prob}(y = x_1 + d > \underline{q}_{i-j} + \underline{d}_j) &\geq \text{Prob}((x_1 > \underline{q}_{i-j}) \& (d > \underline{d}_j)) \geq \\ \text{Prob}(x_1 > \underline{q}_{i-j}) + \text{Prob}(d > \underline{d}_j) - 1 &\geq \left(1 - \frac{i-j}{n}\right) + \left(1 - \frac{j}{n}\right) - 1. \end{aligned} \quad (76)$$

The right-hand side of this formula is equal to $1 - \frac{i}{n}$, so we conclude that

$$\text{Prob}(y > \underline{q}_{i-j} + \underline{d}_j) \geq 1 - \frac{i}{n}. \quad (77)$$

For the complementary event $y \leq \underline{q}_{i-j} + \underline{d}_j$, we thus have

$$F_y(q_{i-j} + \underline{d}_j) = \text{Prob}(y \leq \underline{q}_{i-j} + \underline{d}_j) = 1 - \text{Prob}(y > \underline{q}_{i-j} + \underline{d}_j) \leq 1 - \left(1 - \frac{i}{n}\right) = \frac{i}{n}. \quad (78)$$

This is true for all j , in particular, for j for which the sum $q_{i-j} + \underline{d}_j$ is the largest, hence

$$F_y\left(\max_j(q_{i-j} + \underline{d}_j)\right) \leq \frac{i}{n}. \quad (79)$$

This inequality is true for all possible values of $F_y(\max_j(q_{i-j} + \underline{d}_j))$ corresponding to all possible distributions $F(t)$. In particular, it should be true for the largest possible value $\bar{F}_y(\max_j(q_{i-j} + \underline{d}_j))$:

$$\bar{F}_y\left(\max_j(q_{i-j} + \underline{d}_j)\right) \leq \frac{i}{n}. \quad (80)$$

By definition, the quantile \underline{y}_i is the value for which $\bar{F}_y(\underline{y}_i) = \frac{i}{n}$. Thus, the inequality (80) leads to

$$\bar{F}_y\left(\max_j(q_{i-j} + \underline{d}_j)\right) \leq \bar{F}_y(\underline{y}_i). \quad (81)$$

Since the function \bar{F}_y is a cdf and is, thus increasing, we thus conclude that

$$\max_j(q_{i-j} + \underline{d}_j) \leq \underline{y}_i. \quad (82)$$

A more technical part of the proof from [17] is that there exist distributions for which y_i is exactly equal to the maximum and thus, that the formula (68) is true.

Simplest case of p-boxes in design analysis: formulas for \bar{y}_i and their justification. To get similar formulas for \bar{y}_i , one can use the same computational trick that is often used in interval computations: that an upper bound a for x can be easily computed based on a lower bound for $-x$. Indeed, if we know that $x \leq a$, then $-x \geq -a$, and vice versa.

In the probability case, we thus have

$$\text{Prob}(x \leq a) = \text{Prob}(-x \geq -a). \quad (83)$$

The left-hand side probability $\text{Prob}(x \leq a)$ is the cdf: $\text{Prob}(x \leq a) = F_x(a)$. The right-hand side probability $\text{Prob}(-x \geq -a)$ can be expressed as

$$\text{Prob}(-x \geq -a) = 1 - \text{Prob}(-x \leq -a) = 1 - F_{-x}(-a). \quad (84)$$

Thus, the equality (83) takes the form

$$F_x(a) = 1 - F_{-x}(-a). \quad (85)$$

Since $1 - \dots$ is a decreasing function, we conclude that $F_x(a)$ attains its smallest value $\underline{F}_x(a)$ when $F_{-x}(-a)$ attains its largest possible value $\overline{F}_{-x}(-a)$:

$$\underline{F}_x(a) = 1 - \overline{F}_{-x}(-a). \quad (86)$$

In terms of the quantiles, this means that if $\underline{x}_{i'}$ is the i' -th quantile for $-x$, i.e., if $\overline{F}_{-x}(x_{i'}^-) = \frac{i'}{n}$, then we have

$$\underline{F}_x(-\underline{x}_{i'}^-) = 1 - \frac{i'}{n} = \frac{n - i'}{n}. \quad (87)$$

In particular, for $i' = n - i$, we have $n - i' = i$ and hence

$$\underline{F}_x(-\underline{x}_{n-i}^-) = \frac{i}{n}. \quad (88)$$

By definition of the upper quantile \bar{x}_i as the value for which $\underline{F}_x(\bar{x}_i) = \frac{i}{n}$, this means that

$$\bar{x}_i = -\underline{x}_{n-i}^-. \quad (89)$$

For $i' = n - i$, this means that

$$\underline{x}_{i'}^- = -\bar{x}_{n-i'}. \quad (90)$$

This is true for every quantity, in particular, this is true for $y = x_1 + d$, hence

$$\bar{y}_i = -\underline{y}_{n-i}^-. \quad (91)$$

Here, $-y = (-x_1) + (-d)$. Thus, we can use the formula (68) to compute

$$\underline{y}_{n-i}^- = \max_j (\underline{q}_{(n-i)-j}^- + \underline{d}_j^-). \quad (92)$$

Due to (90), we have

$$\underline{q}_{(n-i)-j'}^- = -\bar{q}_{n-((n-i)-j)} = -\bar{q}_{j-i} \quad (93)$$

and $\underline{d}_{j'}^- = -\bar{d}_{n-j}$. Thus, the formula (92) takes the form

$$\underline{y}_{n-i}^- = \max_j (-(\bar{q}_{j-i} + \bar{d}_{n-j})). \quad (94)$$

The opposite value $-z$ is the largest if and only if z is the smallest, so

$$\underline{y}_{n-i}^- = -\min_j (\bar{q}_{j-i} + \bar{d}_{n-j}). \quad (95)$$

From (91), we can now conclude that

$$\bar{y}_i = \min_j (\bar{q}_{j-i} + \bar{d}_{n-j}). \quad (96)$$

This is the formula deduced in [17].

Summary: formulas for y_i and \bar{y}_i . Let us summarize the above derivation. We consider the case when $y = x_1 + d$. In the *design analysis* problem, we assume that

- we know the interval quantiles $[q_i, \bar{q}_i]$ for x_1 ;
- we know the interval quantiles $[d_i, \bar{d}_i]$ for d , and
- we want to compute interval quantiles $[y_i, \bar{y}_i]$ for y .

For computing y_i and \bar{y}_i , we have the following formulas:

$$y_i = \max_j(q_{i-j} + d_j); \quad (97)$$

$$\bar{y}_i = \min_j(\bar{q}_{j-i} + \bar{d}_{n-j}). \quad (98)$$

Backcalculation: formulation of the problem. In the previous section, we described the formulas that can be used to gauge the quality of a given design. While this analysis is important, a more challenging problem is to analyze the existing design, but to come up with a design that satisfies the given constraints.

In the case when $y = x_1 + d$ and uncertainty is characterized by p-boxes, this means that:

- we know the interval quantiles $[q_i, \bar{q}_i]$ for the environmental variable x_1 ;
- we are given the interval quantiles $[y_i, \bar{y}_i]$ for y , and
- we want to find the interval quantiles $[d_i, \bar{d}_i]$ for d for which y satisfies the given constraints.

In other words, we want to make sure that for our choice of the design, the resulting quantiles for y – which are described by the formula (97) and (98) – are always within the desired bounds $[y_i, \bar{y}_i]$:

$$y_i \leq \max_j(q_{i-j} + d_j); \quad (99)$$

$$\bar{y}_i \geq \min_j(\bar{q}_{j-i} + \bar{d}_{n-j}). \quad (100)$$

This problem is known as the problem of *backcalculation*; see, e.g., [1, 3, 4, 14].

As we can see,

- the inequalities containing y_i only involve the lower bounds d_i , and
- the inequalities containing \bar{y}_i only involve the upper bounds \bar{d}_i .

Thus, we have, in effect, two independent problems:

- finding d_i and

- finding \bar{d}_i .

Of course, these problems are not completely independent, since we need to make sure that $\underline{d}_i \leq \bar{d}_i$. However, for simplicity, we will consider these two problems separately.

By going from x to $-x$, as we have mentioned, we can always reduce the computation of the upper bounds to the computation of the lower bounds. Thus, once we know how to solve the first problem, we can easily solve the second one as well. In view of this observation, we will concentrate on the solution of the first problem.

Before we start describing how to solve the general problem, let us first describe a simple example.

A simple example. In the general p-box case, to describe the uncertainty of a variable x , we use $n + 1$ quantile intervals $[\underline{x}_i, \bar{x}_i]$ for $i = 0, 1, \dots, n$.

P-box uncertainty is a generalization of the case of interval uncertainty, in which the uncertainty in each variable x is characterized by a single interval $[\underline{x}, \bar{x}]$. This case corresponds to $n = 0$. In this case, the inequality for \underline{d}_0 takes the form $\underline{y}_0 \leq \underline{q}_0 + \underline{d}_0$ or, equivalently, $\underline{y}_0 - \underline{q}_0 \leq \underline{d}_0$ – the same form as for interval uncertainty.

We are looking for the simplest possible example which is different from interval uncertainty. After a single interval, the next simplest example is when we use two intervals, i.e., when $n = 1$.

In this case, we need to find two values $\underline{d}_0 \leq \underline{d}_1$ that satisfy the following inequalities:

$$\underline{y}_0 \leq \underline{q}_0 + \underline{d}_0; \quad (101)$$

$$\underline{y}_1 \leq \max(\underline{q}_0 + \underline{d}_1, \underline{q}_1 + \underline{d}_0). \quad (102)$$

Depending on which term in the max expression is the largest, we have two cases: $\underline{q}_0 + \underline{d}_1 \leq \underline{q}_1 + \underline{d}_0$ and $\underline{q}_0 + \underline{d}_1 \geq \underline{q}_1 + \underline{d}_0$.

In the first case, the following inequalities must be satisfied:

$$\begin{aligned} \underline{y}_0 &\leq \underline{q}_0 + \underline{d}_0; \\ \underline{q}_0 + \underline{d}_1 &\leq \underline{q}_1 + \underline{d}_0; \\ \underline{y}_1 &\leq \underline{q}_1 + \underline{d}_0. \end{aligned} \quad (103)$$

The first of these three inequalities from (103) is equivalent to

$$\underline{y}_0 - \underline{q}_0 \leq \underline{d}_0. \quad (104)$$

Similarly, the third inequality from (103) is equivalent to

$$\underline{y}_1 - \underline{q}_1 \leq \underline{d}_0. \quad (105)$$

Thus, these two inequalities are equivalent to

$$\max(\underline{y}_0 - \underline{q}_0, \underline{y}_1 - \underline{q}_1) \leq \underline{d}_0. \quad (106)$$

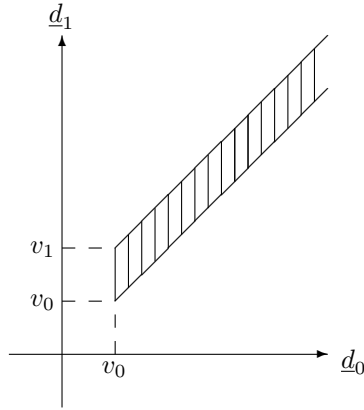
The second inequality from (103) is equivalent to

$$\underline{d}_1 - \underline{d}_0 \leq \underline{q}_1 - \underline{q}_0. \quad (107)$$

Thus, in the first case, the values \underline{d}_0 and \underline{d}_1 must satisfy the following two inequalities:

$$\begin{aligned} \underline{d}_0 &\geq \max(\underline{y}_0 - \underline{q}_0, \underline{y}_1 - \underline{q}_1); \\ 0 &\leq \underline{d}_1 - \underline{d}_0 \leq \underline{q}_1 - \underline{q}_0. \end{aligned} \quad (108)$$

Graphically, the values that satisfy these inequalities fill the following area:



where $v_0 \stackrel{\text{def}}{=} \max(\underline{y}_0 - \underline{q}_0, \underline{y}_1 - \underline{q}_1)$ and $v_1 \stackrel{\text{def}}{=} v_0 + \underline{q}_1 - \underline{q}_0$.

In the second case, the following inequalities must be satisfied:

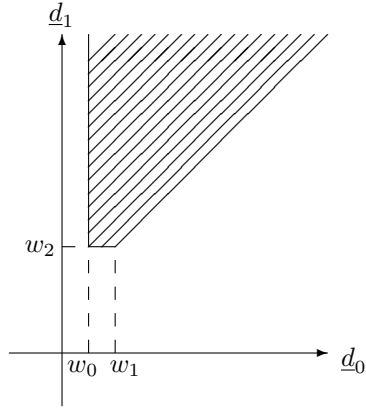
$$\begin{aligned} \underline{y}_0 &\leq \underline{q}_0 + \underline{d}_0; \\ \underline{q}_0 + \underline{d}_1 &\geq \underline{q}_1 + \underline{d}_0; \\ \underline{y}_1 &\leq \underline{q}_0 + \underline{d}_1. \end{aligned} \quad (109)$$

By moving the unknowns to one side and all the other terms to the other side, we conclude that we must satisfy the following inequalities:

$$\begin{aligned} \underline{d}_0 &\geq \underline{y}_0 - \underline{q}_0; \\ \underline{d}_1 - \underline{d}_0 &\geq \underline{q}_1 - \underline{q}_0; \\ \underline{d}_1 &\geq \underline{y}_1 - \underline{y}_0. \end{aligned} \quad (110)$$

(Since $\underline{q}_1 - \underline{q}_0 \geq 0$, the second inequality automatically implies that $\underline{d}_1 \geq \underline{d}_0$.)

Graphically, we have the following representation:



where $w_0 \stackrel{\text{def}}{=} \underline{y}_0 - \underline{q}_0$, $w_1 \stackrel{\text{def}}{=} w_0 + (\underline{q}_1 - \underline{q}_0)$, and $w_2 \stackrel{\text{def}}{=} \underline{y}_1 - \underline{y}_0$.

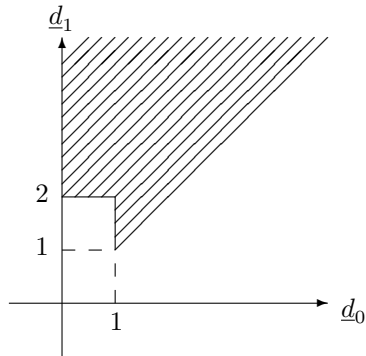
Overall, the set of all possible design values is the union of these two sets. Let us illustrate this union on a simple numerical example when $\underline{q}_0 = \underline{y}_0 = 0$, $\underline{q}_1 = 1$, and $\underline{y}_1 = 2$. In this case, the first case corresponds to the following inequalities

$$\begin{aligned} \underline{d}_0 &\geq 1; \\ 0 &\leq \underline{d}_1 - \underline{d}_0 \leq 1. \end{aligned} \tag{111}$$

and the second case leads to the following inequalities:

$$\begin{aligned} \underline{d}_0 &\geq 0; \\ \underline{d}_1 - \underline{d}_0 &\geq 1; \\ \underline{d}_1 &\geq 2. \end{aligned} \tag{112}$$

Thus, the union of the two corresponding sets has the following form:



Optimal backcalculation: towards the formulation of the problem in precise terms. As we can see from the above example, there are many combinations of the values \underline{d}_i that satisfy the desired constraints. Which combination should we choose?

In the design case, a natural requirement is to select the values \underline{d}_i which will be the easiest to implement. To describe this idea in precise terms, we must analyze how easy is it to implement different values of \underline{d}_i .

In general, the value \underline{d}_i is the value for which $\bar{F}(\underline{d}_i) = \frac{i}{n}$, i.e., we which we must guarantee that

$$\text{Prob}(d \leq \underline{d}_i) = F(\underline{d}_i) \leq \bar{F}(\underline{d}_i) = \frac{i}{n}. \quad (113)$$

In particular:

- once we select the value \underline{d}_0 , we must guarantee that the probability $\text{Prob}(d \leq \underline{d}_0)$ of the actual value d being below \underline{d}_0 is 0: $\text{Prob}(d \leq \underline{d}_0) = 0$;
- once we select \underline{d}_1 , we must guarantee that the probability $\text{Prob}(d \leq \underline{d}_1)$ of the actual value d being below \underline{d}_1 does not exceed $1/n$: $\text{Prob}(d \leq \underline{d}_1) \leq 1/n$;
- once we select \underline{d}_2 , we must guarantee that the probability $\text{Prob}(d \leq \underline{d}_2)$ of the actual value d being below \underline{d}_2 does not exceed $2/n$: $\text{Prob}(d \leq \underline{d}_2) \leq 2/n$;
- etc.

When we motivated the need to take into account partial information about the probabilities, we have mentioned that the most difficult task is to guarantee that the actual d *never* gets below a threshold. The corresponding restriction is related to the value \underline{d}_0 : we must guarantee that $d \geq \underline{d}_0$. The smaller this value \underline{d}_0 , the weaker this constraint and thus, the easiest to satisfy. Thus, it is reasonable to select the smallest possible value \underline{d}_0 .

Once this value is selected and the corresponding bound is guaranteed, we must guarantee that the inequalities $d \geq \underline{d}_i$ be guaranteed with the probabilities $\geq 1 - i/n$. The closer this probability to 1, the more stringent is the corresponding requirement, and thus, the more difficult the corresponding task. So, after we have selected \underline{d}_0 , the most difficult of the remaining tasks is to select the value \underline{d}_1 , the value for which the guaranteed probability of violating the restriction $d \geq \underline{d}_1$ is the smallest (probability = $1/n$). Thus, to make the design as easy to implement as possible, we should make this restriction the least difficult to implement – i.e., we should select the value \underline{d}_1 as small as possible.

Once we have fixed \underline{d}_0 and \underline{d}_1 , the most difficult of the remaining tasks is to guarantee that $d \geq \underline{d}_2$ with probability $\geq 1 - (2/n)$. Thus, we must select the corresponding threshold \underline{d}_2 to be as small as possible.

As a result, we arrive at the following formulation of the backcalculation problem.

Formulation of the optimal backcalculation problem in precise mathematical terms. Out of all the tuples $\underline{d}_0 \leq \dots \leq \underline{d}_n$ that satisfy the inequalities (99),

- we first select all the tuples for which the value \underline{d}_0 is the smallest possible;
- out of the selected tuples, we select all the tuples for which the value \underline{d}_1 is the smallest possible;
- then, out of the newly selected tuples, we select those for which the value \underline{d}_2 is the smallest possible;
- etc.

In mathematical terms, we can say that a tuple $\underline{d} = (\underline{d}_0, \dots, \underline{d}_n)$ is *better* than a tuple $\underline{d}' = (\underline{d}'_0, \dots, \underline{d}'_n)$ if one of the following conditions hold:

- either $\underline{d}_0 < \underline{d}'_0$;
- or $\underline{d}_0 = \underline{d}'_0$ and $\underline{d}_1 < \underline{d}'_1$;
- or $\underline{d}_0 = \underline{d}'_0$, $\underline{d}_1 = \underline{d}'_1$, and $\underline{d}_2 < \underline{d}'_2$;
- ...
- or $\underline{d}_0 = \underline{d}'_0, \dots, \underline{d}_{i-1} = \underline{d}'_{i-1}$, and $\underline{d}_i < \underline{d}'_i$;
- ...
- or $\underline{d}_0 = \underline{d}'_0, \dots, \underline{d}_{n-1} = \underline{d}'_{n-1}$, and $\underline{d}_n < \underline{d}'_n$.

In computer science, this relation is known as a *lexicographic* (alphabetic) order, since this is exactly how words are placed in a dictionary or in a lexicon: a word $w = \ell_0 \ell_1 \dots$ consisting of the letters ℓ_0, ℓ_1, \dots , is placed before a word $w' = \ell'_0 \ell'_1 \dots$ consisting of the letters ℓ'_0, ℓ'_1, \dots if one of the following conditions hold:

- either the letter ℓ_0 precedes the letter ℓ'_0 (e.g., *apple* goes before *zebra*);
- or $\ell_0 = \ell'_0$ and the next letter ℓ_1 precedes the corresponding letter ℓ'_1 (e.g., *abuse* goes before *alpha*);
- ...
- or $\ell_0 = \ell'_0, \dots, \ell_{i-1} = \ell'_{i-1}$, and the next letter ℓ_i precedes the corresponding letter ℓ'_i ;
- ...

In these terms, we must select the tuple which is the smallest in the lexicographic order.

3 Towards a solution to the optimal backcalculation problem.

To find the optimal solution, let us start with the value \underline{d}_0 . For $i = 0$, the condition (99) becomes $\underline{q}_0 + \underline{d}_0 \leq \underline{y}_0$, i.e., equivalently, $\underline{d}_0 \geq \underline{y}_0 - \underline{q}_0$. The smallest real number that satisfies this inequality is the value $\underline{d}_0 = \underline{y}_0 - \underline{q}_0$. Thus, we should take

$$\underline{d}_0 = \underline{y}_0 - \underline{q}_0. \quad (114)$$

Let us now assume that we have already selected the values $\underline{d}_0, \dots, \underline{d}_{i-1}$, and that we are now selecting the value \underline{d}_i . The i -th condition (99) has the form

$$\underline{y}_i \leq \max_j (\underline{q}_{i-j} + \underline{d}_j) = \max \left[\max_{j \leq i-1} (\underline{q}_{i-j} + \underline{d}_j), \underline{q}_0 + \underline{d}_i \right]. \quad (115)$$

If

$$\underline{y}_i \leq \max_{j \leq i-1} (\underline{q}_{i-j} + \underline{d}_j), \quad (116)$$

then the condition (115) is already satisfied. In this case, the only restriction on \underline{d}_i is that $\underline{d}_i \geq \underline{d}_{i-1}$; thus, the smallest possible value of \underline{d}_i is $\underline{d}_i = \underline{d}_{i-1}$.

If the inequality (116) is not satisfied, then to satisfy (115), we must satisfy the inequality $\underline{y}_i \leq \underline{q}_0 + \underline{d}_i$, i.e., equivalently, $\underline{d}_i \geq \underline{y}_i - \underline{q}_0$. Thus, the smallest possible value here is $\underline{d}_i = \underline{y}_i - \underline{q}_0$. So, we arrive at the following algorithm:

Algorithm for optimal backcalculation. ([1, 3, 4, 14]) We know:

- the interval quantiles $[\underline{q}_i, \bar{q}_i]$ for the environmental variable x_1 ; and
- the interval quantiles $[\underline{y}_i, \bar{y}_i]$ for y .

We want to find the lexicographically optimal interval quantiles $[\underline{d}_i, \bar{d}_i]$ for d for which y satisfies the given constraints.

In this algorithm, we compute the values $\underline{d}_0, \underline{d}_1, \dots$ one by one.

- First, we compute $\underline{d}_0 = \underline{y}_0 - \underline{q}_0$.
- Once the values $\underline{d}_0, \dots, \underline{d}_{i-1}$ are computed, we check whether

$$\underline{y}_i \leq \max_{j \leq i-1} (\underline{q}_{i-j} + \underline{d}_j). \quad (117)$$

If this inequality is satisfied, we take $\underline{d}_i = \underline{d}_{i-1}$; otherwise, we take $\underline{d}_i = \underline{y}_i - \underline{q}_0$.

Comment. In the derivation of the algorithm, we, in fact, proved that the result \underline{d} of applying this algorithm always satisfies the inequalities (99): indeed, we have selected each value \underline{d}_i in such a way that the i -th inequality (99) is satisfied.

We have also shown that no tuple satisfying (99) can be (lexicographically) better than the result \underline{d} of using this algorithm – and thus, this result \underline{d} is indeed optimal.

Algorithm illustrated on the above simple example. In the above example when $\underline{q}_0 = \underline{y}_0 = 0$, $\underline{q}_1 = 1$, and $\underline{y}_1 = 2$, we do the following:

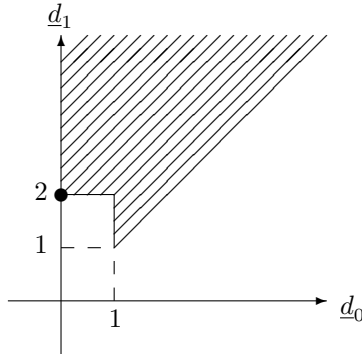
- First, we compute $\underline{d}_0 = \underline{y}_0 - \underline{q}_0 = 0 - 0 = 0$.
- Next, we check whether

$$\underline{y}_1 \leq \max_{j \leq 0} (\underline{q}_{i-j} + \underline{d}_j) = \underline{q}_1 + \underline{d}_0. \quad (118)$$

Here, we have $2 = \underline{y}_1 \not\leq \underline{q}_1 + \underline{d}_0 = 1 + 0 = 1$, so we take $\underline{d}_1 = \underline{y}_1 - \underline{q}_0 = 2 - 0 = 2$.

One can easily check that the resulting tuple $(\underline{d}_0, \underline{d}_1) = (0, 2)$ is indeed lexicographically optimal:

- it has the smallest possible value of $\underline{d}_0 = 0$, and
- out of all the tuples with this value of \underline{d}_0 , it has the smallest possible value of \underline{d}_1 .



The use of modal intervals: reminder. In [16], it was proposed to use modal intervals to solve the backcalculation problem. To explain the idea, let us briefly recall, from Part I, how modal intervals can be used in design in the case of interval uncertainty.

In the absence of uncertainty, when we know the exact value of the environmental parameter x_1 and the exact desired value of the quality y , the solution to the design problem is straightforward: take $d = y - x_1$. In the computer, subtraction $y - x_1$ is often implemented as adding $-x_1$ to y , so this formula is equivalent to $d = y + (-x_1)$.

In the case of interval uncertainty, we know:

- an interval $[\underline{x}_1, \bar{x}_1]$ of possible values of x_1 and

- an interval $[\underline{y}, \bar{y}]$ of the desired values of y .

In this case, in principle, we can:

- compute the interval $[-\bar{x}_1, -\underline{x}_1]$ of possible values of $-x_1$, and then
- compute the interval of possible values of $d = y + (-x_1)$ as

$$[\underline{y} + (-\bar{x}_1), \bar{y} + (-\underline{x}_1)] = [\underline{y} - \bar{x}_1, \bar{y} - \underline{x}_1]. \quad (119)$$

As we have discussed in Part I, the resulting interval will *not* solve the original design problem. We can, however, compute the solution to the design problem if instead of the original interval $[\underline{x}_1, \bar{x}_1]$, we plug in the “dual interval” $[\bar{x}_1, \underline{x}_1]$ into the above formula.

Towards the use of modal intervals in p-boxes. How can this idea be extended to p-boxes? For the p-boxes, we can, in principle, also:

- compute the p-box corresponding to $-x_1$, and then
- use the known p-boxes for y and for $-x_1$ to compute the p-box corresponding to $d = y + (-x_1)$.

Once we know the lower bound \underline{y}_i of the quantile intervals for y and the lower bounds \underline{q}_i^- of the quantile intervals for $x_1^- \stackrel{\text{def}}{=} -x_1$, we can compute the corresponding bounds for $y + (-x_1)$ as

$$\underline{d}_i = \max_j (\underline{y}_{i-j} + \underline{q}_j^-). \quad (120)$$

Here, according to the formula (90), we have $\underline{q}_j^- = -\bar{q}_{n-j}$, so the formula (120) takes the form

$$\underline{d}_i = \max_j (\underline{y}_{i-j} - \bar{q}_{n-j}). \quad (121)$$

Just like in the interval case, the resulting p-box will *not* solve the original design problem – it already does not solve this problem even for the interval case $n = 0$. The idea of modal interval analysis is that to get a solution, we formally replace the bounds \bar{q}_{n-j} corresponding to $x_1^- = -x_1$ with some “dual” bounds.

How can we define this duality? In the interval case, we started with a formula

$$\underline{x}^- = -\bar{x} \quad (122)$$

that related the lower bound \underline{x}^- for $x^- = -x$ with the bound \bar{x} for x , and defined duality as replacing \underline{x} with \bar{x} (and vice versa). In the p-box case, a similar formula relating the quantile intervals is the formula (90): $\underline{x}_{i'}^- = -\bar{x}_{n-i}$. Thus, it is natural to define the “dual” as replacing \underline{x}_i with \bar{x}_{n-i} and, vice versa, \bar{x}_i with \underline{x}_{n-i} .

This duality is a natural generalization of the interval duality. Indeed, as we have mentioned, the value \underline{x}_0 is the guaranteed lower bound for the quantity

x , and similarly, \bar{x}_n is the guaranteed upper bound. Thus, the only thing we can conclude with 100% guarantee is that the quantity x belongs to the interval $[\underline{x}_0, \bar{x}_n]$. If we want bounds which are valid with probability $\geq 1 - (i/n)$, then we can take a narrower interval $[\underline{x}_i, \bar{x}_{n-i}]$. In all these cases, a natural transition from the lower bound to an upper bound means going from \underline{x}_i to \bar{x}_{n-i} and vice versa.

With this notion of duality, we arrive at the following definition.

Modal p-boxes: a formal definition. Once we have a p-box b_x described by the quantile intervals

$$[\underline{x}_0, \bar{x}_0], [\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_n, \bar{x}_n], \quad (123)$$

we can define its *dual* as

$$[\bar{x}_n, \underline{x}_n], [\bar{x}_{n-1}, \underline{x}_{n-1}], \dots, [\bar{x}_0, \underline{x}_0]. \quad (124)$$

Modal p-boxes provide a solution to the design problem: a proof. If we formally replace, in the formula (120) describing a p-box for $y + (-x_1)$, the original p-box for x_1 with its dual, we get the following formula for d_i :

$$\underline{d}_i = \max_j (y_{i-j} - \underline{q}_j). \quad (125)$$

Let us show that, similar to the interval case, this “modal” solution is indeed a solution to the original design problem. In other words, we need to prove that for the values \underline{d}_i as described by the formula (125), the desired inequalities (99) are satisfied, i.e., that for every i , we have

$$\underline{y}_i \leq \max_j (\underline{q}_{i-j} + \underline{d}_j). \quad (126)$$

To prove this result, we will first show that the above expression (125) can be further simplified. Indeed, the expression (125) has the form

$$\underline{d}_i = \max(\underline{y}_i - \underline{q}_0, \underline{y}_{i-1} - \underline{q}_1, \dots, \underline{y}_0 - \underline{q}_i). \quad (127)$$

Both the values \underline{y}_i and \underline{q}_i are increasing with i . Thus,

$$\underline{y}_i \geq \underline{y}_{i-1} \geq \dots \geq \underline{y}_0 \quad (128)$$

and, similarly,

$$\underline{q}_0 \leq \underline{q}_1 \leq \dots \leq \underline{q}_n. \quad (129)$$

Therefore,

$$\underline{y}_i - \underline{q}_0 \geq \underline{y}_{i-1} - \underline{q}_1 \geq \dots \geq \underline{y}_0 - \underline{q}_n, \quad (130)$$

and thus, the maximum in (127) is always attained at the first term. So, the modal solution has the simplified form

$$\underline{d}_i = \underline{y}_i - \underline{q}_0. \quad (131)$$

For $j = i$, we have

$$\max_j (\underline{q}_{i-j} + \underline{d}_j) \geq \underline{q}_0 + \underline{d}_i. \quad (132)$$

Due to (127), we have

$$\max_j (\underline{q}_{i-j} + \underline{d}_j) \geq \underline{q}_0 + \underline{d}_i = \underline{q}_0 + (\underline{y}_i - \underline{q}_0) = \underline{y}_i, \quad (133)$$

i.e., the desired inequality (126). The statement is proven.

The solution produced by modal p-boxes is not always optimal: an example. In the interval case, the modal interval solution was equivalent to backcalculation. In other words, the modal interval analysis simply provided an alternative (and sometimes easier-to-describe) way to describe the same solution.

In the p-box case, the formula for the modal p-box solution is different. Since the backcalculation solution that we have described earlier is optimal, this difference means that whenever these solutions differ, the modal solution is *not* optimal. Indeed, let us give a simple numerical example when the modal solution is not optimal.

Take $n = 1$, $\underline{q}_0 = \underline{y}_0 = 0$, and $\underline{q}_1 = \underline{y}_1 = 1$.

- Let us first use the above algorithm to compute the optimal solution. Here, $\underline{d}_0^{\text{opt}} = \underline{y}_0 - \underline{q}_0 = 0 - 0 = 0$. Since

$$\max_{j \leq i-1} (\underline{q}_{i-j} + \underline{d}_j) = \underline{q}_1 + \underline{d}_0 = 1 + 0 = 1 \geq 1 = \underline{y}_1, \quad (134)$$

we take $\underline{d}_1^{\text{opt}} = \underline{d}_0 = 0$.

- In contrast, for the modal solution, we take $\underline{q}_0^{\text{mod}} = \underline{y}_0 - \underline{q}_0 = 0 - 0 = 0$ and $\underline{q}_1^{\text{mod}} = \underline{y}_1 - \underline{q}_0 = 1 - 0 = 1$.

So, here $\underline{q}_0^{\text{opt}} = \underline{q}_0^{\text{mod}}$ but $\underline{q}_1^{\text{opt}} < \underline{q}_1^{\text{mod}}$. Thus, the optimal solution is better – and hence, the modal solution is indeed not optimal.

Extension to control. In the interval case, one of the advantages of using modal intervals was that it enabled us to take into account the possibility to consider *active* (controlled) design, where we can achieve the desired values of the quality y by applying an appropriate control. Let us show how we can extend this idea to p-boxes.

We will consider the simple *additive* case of control, when:

- there is only one control parameter c ,
- we know the range $[\underline{c}, \bar{c}]$ and
- the dependence on the desired quality y on this control parameter has the additive form $y = f(x_1, \dots, x_m, d) + c$.

A *multiplicative* control, when $y = f(x_1, \dots, x_m, d) \cdot c$, can be handled similarly.

In this case, if we need to maintain the exact value of the quality y , it is sufficient to select d in such a way that

$$f(x_1, \dots, x_m) \in y - [\underline{c}, \bar{c}] = [y - \bar{c}, y - \underline{c}]. \quad (135)$$

Indeed, for each value y' from this interval, the difference $y - y'$ belongs to the interval $[\underline{c}, \bar{c}]$ and thus, can be achieved by a possible control.

If we need to maintain the value y within a given interval $[\underline{y}, \bar{y}]$, then similarly, it is sufficient to select d in such a way that

$$f(x_1, \dots, x_m) \in [\underline{y}, \bar{y}] - [\underline{c}, \bar{c}] = [\underline{y} - \bar{c}, \bar{y} - \underline{c}]. \quad (136)$$

Indeed, for each value y from this interval $[\underline{y} - \bar{c}, \bar{y} - \underline{c}]$, it is possible to find the value $c \in [\underline{c}, \bar{c}]$ for which $y + c \in [\underline{y}, \bar{y}]$.

Similarly, if we need to maintain a p-box for y , it is sufficient to select d in such a way that the value $f(x_1, \dots, x_m, d)$ is guaranteed to fit into the p-box corresponding to $y - c$. In other words, instead of fitting into the original p-box described by quantile intervals $[\underline{y}_i, \bar{y}_i]$, it is sufficient to fit the value $f(x_1, \dots, x_m, d)$ into the p-box described by wider quantile intervals

$$[\underline{y}_i, \bar{y}_i] - [\underline{c}, \bar{c}] = [\underline{y}_i - \bar{c}, \bar{y}_i - \underline{c}]. \quad (137)$$

In this case, by applying appropriate controls – to fit into the range $[\underline{y}_i, \bar{y}_{n-i}]$ with the largest possible i – we will get the desired p-box.

References

- [1] S. Ferson, “Using approximate deconvolution to estimate cleanup targets in probabilistic risk analyses”, In: P. T. Kostecki, E. J. Calabrese, and M. Bonazountas (eds.), *Hydrocarbon Contaminated Soils*, Amherst Scientific Publishers, Amherst, Massachusetts, 1995, pp. 245–254.
- [2] S. Ferson. *RAMAS Risk Calc 4.0*. CRC Press, Boca Raton, Florida, 2002.
- [3] S. Ferson, V. Kreinovich, and W. T. Tucker, “Untangling equations involving uncertainty: deconvolutions, updates, and backcalculations”, *Proceedings of the NSF Workshop on Reliable Engineering Computing*, Savannah, Georgia, September 15–17, 2004.
- [4] S. Ferson and T. F. Long, “Deconvolution can reduce uncertainty in risk analyses”, In: M. Newman and C. Stojan (eds.), *Risk Assessment: Measurement and Logic*, Ann Arbor Press, Ann Arbor, Michigan, 1997.
- [5] E. Gardeñes, M. A. Sainz, L. Jorba, R. Calm, R. Estela, H. Mielgo, and A. Trepast, “Modal intervals”, *Reliable Computing*, 2001, Vol. 7, No. 2, pp. 77–111.

- [6] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag, London, 2001.
- [7] V. Kreinovich, “Interval Computations and Interval-Related Statistical Techniques: Tools for Estimating Uncertainty of the Results of Data Processing and Indirect Measurements”, In: F. Pavese and A. B. Forbes (eds), *Advances in Data Modeling for Measurements in the Metrology and Testing Fields*, Birkhauser-Springer, Boston, 2008, pp. 119–147.
- [8] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational complexity and feasibility of data processing and interval computations*, Kluwer, Dordrecht, 1998.
- [9] S. Rabinovich, *Measurement Errors and Uncertainties: Theory and Practice*, Springer-Verlag, New York, 2005.
- [10] M. A. Sainz, E. Gardeñes, and L. Jorba, “Formal solution to systems of interval linear or non-linear equations”, *Reliable Computing*, 2002, Vol. 8, No. 3, pp. 189–211.
- [11] M. A. Sainz, E. Gardeñes, and L. Jorba, “Interval estimations of solution sets to real-valued systems of linear or non-linear equations”, *Reliable Computing*, 2002, Vol. 8, No. 4, pp. 283–305.
- [12] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman & Hall/CRC, Boca Raton, Florida, 2004.
- [13] S. P. Shary, “A New Technique in Systems Analysis under Interval Uncertainty and Ambiguity”, *Reliable Computing*, 2001, Vol. 8, pp. 321–418.
- [14] W. T. Tucker and S. Ferson, “Setting cleanup targets in a probabilistic assessment”, In: S. Mishra (ed.), *Groundwater Quality Modeling and Management under Uncertainty*, American Society of Civil Engineers, Reston, Virginia, 2003.
- [15] Y. Wang, “Semantic Tolerance Modeling based on Modal Interval Analysis”, *Proceedings of the Second International Workshop on Reliable Engineering Computing*, Savannah, Georgia, February 22–24, 2006, pp. 293–318.
- [16] Y. Wang, “Semantic Tolerance Modeling based on Modal Interval Analysis”, *Proceedings of the International Workshop on Reliable Engineering Computing REC’08*, Savannah, Georgia, February 20–22, 2008, pp. 46–59.
- [17] R. C. Williamson and T. Downs, “Probabilistic arithmetic I: Numerical methods for calculating convolutions and dependency bounds”, *International Journal of Approximate Reasoning*, 1990, Vol. 4, pp. 89–158.