Determination of measurement uncertainty in imc FAMOS

Monte Carlo method

What is a Monte Carlo simulation?

The Monte Carlo simulation is a technique from the stochastic where very frequently performed random experiments constitute the basis. The aim is to numerically solve analytically insolvable problems using probability theory. The random experiments are generated via random numbers. The Monte Carlo method (MCM) usually requires a large number of random experiments in order to be meaningful.

Monte Carlo simulations are employed in many areas (e.g. in financial mathematics, in risk management, in nuclear and particle physics, or in process and control technology). Moreover, the Monte Carlo simulation is also suitable for determining the measurement uncertainty. The Monte Carlo simulation can replace complex and expensive laboratory experiments as well as assisting in the collection of large amounts of data.

Advantages of the method:

- Processing of non-linearities, complex algorithms, with non-normal distributions
- Often the only simulation method that delivers useful results within a reasonable computation time
- Systematic improvement of the solution using more computing time

General procedure:

Usually up to 10 million (between 10^4-10^7) simulations of the same problem are performed while varying the values of the influencing variables. The individual simulation results must then be evaluated according to statistical criteria, i.e. the expected value and the standard deviation are calculated.

The core of Monte Carlo simulations is the application of random variables. These random variables are applied to the problem to be examined (in our case the measurement series) and in so doing a number of inaccurate copies of the matter to be examined are created. The random number is "rolled" each time; this is referred to as a random walk. The mean value and the standard deviation are now calculated for the collection of copies. With some systems it is necessary to use a "loaded" die that generates a sequence of random numbers with prescribed statistical properties (such as a certain type of distribution: rectangular distribution, triangular distribution, normal distribution).

The Monte Carlo method in imc FAMOS

imc FAMOS uses the Monte Carlo method (MCM). Several methods of calculation are specified in the GUM. The direct analytical calculation is directly possible for many simple equations, but is virtually unsustainable for most of the commands in imc FAMOS. The MCM has no problems here. This results in a systematic way for any algorithm.

The MCM is based on adding (slight) noise to the input data. For this purpose random numbers are applied to the data to simulate deviations. These deviations take effect as deviations of the results of the algorithm to be calculated. The standard deviation is determined and produces the measurement uncertainty of the results.

It should be noted that "addition of noise" does not necessarily mean the addition of noise in a strict sense. It may be the application of various forms of interference.

The MCM is a statistical method and requires a sufficiently large number of random numbers in order to be meaningful.

The GUM uncertainty framework is taken into account, but not used.

Since the GUM assumes a symmetrical distribution for the numerical value of the expanded measurement uncertainty, imc FAMOS also determines the shortest coverage interval as a symmetrical one. If a finer analysis is required, especially with asymmetrical distributions, this can be done using an enhanced analysis.

Number of Monte Carlo experiments:

The number of Monte Carlo experiments can be input via the function "Uncertainty_LOOP". It should be noted that "addition of noise" does not necessarily mean the addition of noise in a strict sense. It can be the application of various forms of interference (noise, mains hum, offset drift). The results of the algorithm are given the user-defined property "Uncertainty"; this is specified as the standard deviation.

The Uncertainty_LOOP in imc FAMOS is necessary for the performance of all M Monte Carlo experiments. M+2 passes of the loop are performed: The first pass calculates the uninfluenced result from the uninfluenced input data. This is followed by the M Monte Carlo passes. Noise is thereby added to the input data. The estimated value for the measurement uncertainty is improved from pass to pass. In the final pass all uninfluenced values are restored.

The calculation of the measurement uncertainty becomes even more reliable and more accurate, the more random samples are incorporated into the calculation. It is also important that the samples themselves are well distributed in order to ensure good coverage of the possible range of the random samples.

The advanced analysis functions (uc, mean, pdf, pdf0) help in determining whether the number M of the Monte Carlo experiments is sufficiently high. If it is too low, the distribution density is not a nice smooth curve. Also, the calculations of "uc" and "mean" then fluctuate strongly. The fluctuations of the calculations of "min/max" should not contribute significantly to the assessment, since in the case of these variables no averaging effect occurs with an increasing number M.

In figures 2 and 3 below, you can see how the distribution density becomes increasingly smooth with the number of Monte Carlo experiments.

First example:

The temperature data record has, for example, 5,000 measurements (sampling rate 100 Hz, Figure 1). A result (rise time) is determined from this time series. Some single-value auxiliary variables are generated here.

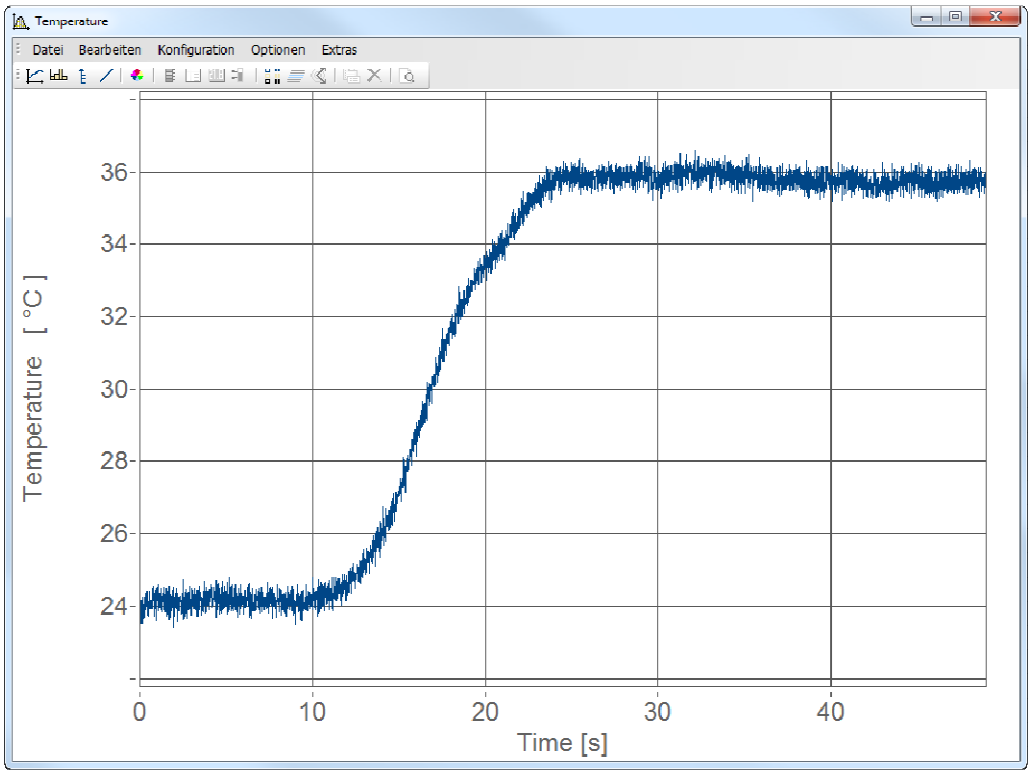


Figure 1

The probability density function for M = 100, 1000 and 10,000 experiments is shown in the following illustrations. The fact that a great many values have been incorporated for the calculation of the auxiliary variables has no effect here. It transpires that 10,000 experiments are necessary in order to obtain a sufficiently smooth curve (see Figure 2).

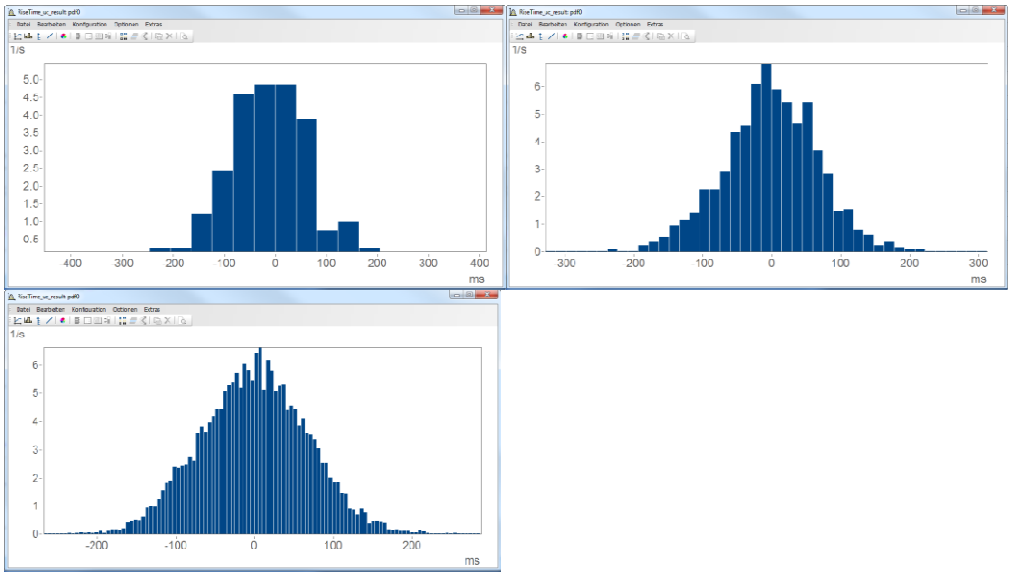


Figure 2

Second example:

Difference between two temperatures:

Two temperature measurements are given, which are subtracted from each other. The temperature measurements each contain 1000 values. A measurement uncertainty is assigned to both channels. In each Monte Carlo experiment, noise is applied to each of the two channels and the differentials are calculated from all value pairs on each pass. As can be seen in the following illustrations, M = 10 passes can be sufficient here to obtain a sufficiently smooth probability density function. The reason is that in this case 10 x 1000 values are incorporated in the calculation.

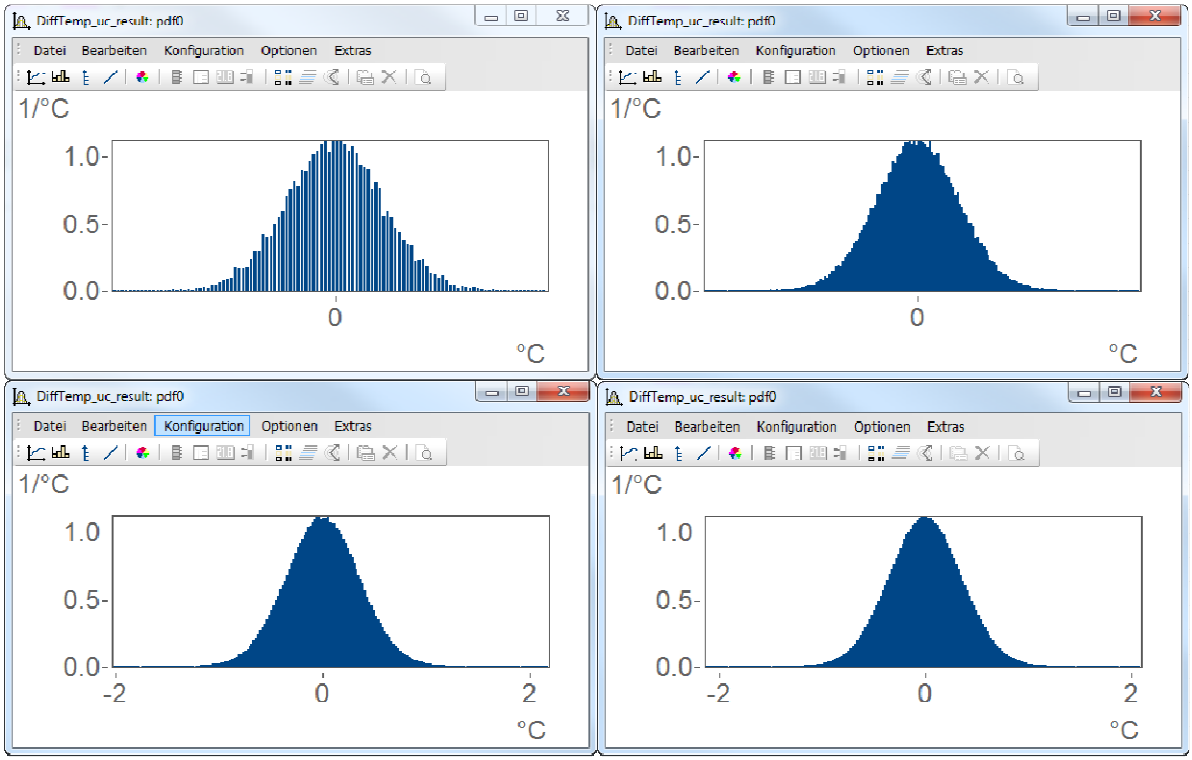


Figure 3

Conclusion: The number of passes of the loop ultimately performed depends on the task in hand. However, it can be shown that with the offsetting of data series just a small number of Monte Carlo experiments can lead to a good result.

Qualitative explanation:

The MCM is not used in imc FAMOS to calculate the measured value itself, but only its measurement uncertainty. The measurement uncertainty generally doesn't need to be determined with many decimal places. In this sense it is certainly relevant whether the measurement uncertainty is 2 mV or 3 mV. A specification of 2.3971 mV for the measurement uncertainty is certainly inappropriate.

For example, the mechanical work is to be determined in an equation taking the existing values for force and displacement. Here, an M of 10 Monte Carlo experiments is certainly too little. If the complete determination of the measurement uncertainty is executed, one obtains a strongly differing result

each time. This shows that an M of 10 random samples is insufficient for determining the standard deviation. If M = 100 the fluctuations are much smaller, but still clearly visible. With an M of 1000 the result seems to be stable.

For example, we have a time series that contains 10,000 measurements. A result is determined from this time series by filtering, etc. The result itself is a single value. It can transpire here that a reasonably good result is achieved after an M of only 50 Monte Carlo experiments, because n = 10000 x 50 values are included in the calculation of the standard deviation.

There are algorithms that branch differently, depending on the measured value. For sufficient accuracy it is important that each branch of the calculation is run through sufficiently often.

The sufficiently accurate determination of measurement uncertainty and the distribution density of the input data is a prerequisite for sufficient accuracy of the determined measurement uncertainty. In many cases, despite supposed accuracy and correctly filled measuring uncertainty budget, the estimation of the measurement uncertainty of the input data is rather coarse. The determination of the distribution densities is not possible in many practical cases. In such cases one diverts to a normal distribution. However, that is often also just a (rough) approximation. It follows that a supposedly high precision of a determined measurement uncertainty doesn't have to be really precise.

The Monte Carlo method converges with $\sqrt{M}$. For a significantly more stable measurement uncertainty, the number of Monte Carlo experiments needs to be quadratically increased. Thus, for example, improvement by a factor of 10 (one decimal place more accurate) by increasing the number by a factor of 100. This is practically possible only for small algorithms or single value calculations.

The following consideration still applies to the determination of the expanded measurement uncertainty: the expanded measurement uncertainty is estimated from the empirically determined distribution density. For example, if a coverage probability of 99 % is required, then 1 % of the values for the distribution density may lie outside the coverage interval. However, this 1 % may not consist only of one value – it must consist of many. If n = 1000 values are used, for example, then the coverage interval is chosen so that 10 values lie outside it. The value of the expanded measurement uncertainty thus depends on these 10 random values. This still cannot be called high precision. However, n = 10,000 values are needed so that 100 values lie outside, which is already much better.

Quantitative explanation:

Let n be the number of values included in the calculation of the measurement uncertainty. This is calculated from: n = [number of Monte Carlo experiments M] · [number of measured values in the result]

Example: If the number of Monte Carlo experiments M = 10 and the length of the result is 10 values, then n = 100. From the number n it is possible to estimate the reliability of the calculation of the measurement uncertainty. Figure 4 shows a typical distribution for n = 100 and a true value of 1.

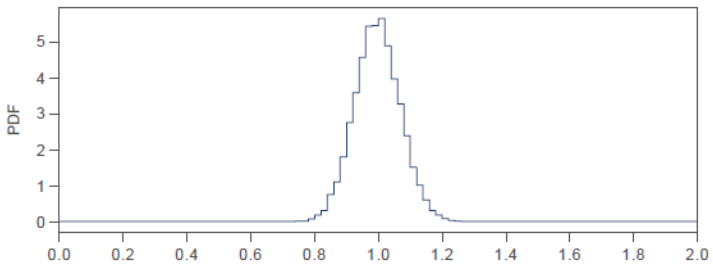


Figure 4, PDF stands for Power Density Function

To do this, the measurement uncertainty calculation is repeated many times. Let the precise measurement uncertainty of the result be 1.0. The result is slightly different each time. If n = 100, for example, the values usually lie in the very rough range [0.8, 1.2], see illustration. Of course, most are very close to 1. The measurement uncertainty is accurately determined to be about ±0.2. Note that the precise value of 1.0 is unknown. If the measurement uncertainty is calculated, a value of 0.8 is obtained, for example. That must be interpreted as follows: The value can be too small by the factor 1/0.8. If one were to calculate a value of 1.2 at another time, then it would be too large by the factor 1.2. Thus the interval = [0.8·1/1.2, 0.8·1/0.8], or: the true value lies within the interval with a certain probability. As with the coverage interval this is interpreted as follows: if a large number of measuring uncertainty calculations are performed, then the true value lies within the interval for the proportion of performances (which corresponds to the given probability). So, for example, if n = 100, one can say: the value lies within the interval

[0.88 · [Calculated measurement uncertainty], 1.16 · [Calculated measurement uncertainty]] with a probability of 95 %. The prerequisite is the normal distribution of the measured values. The result will be more accurate with a larger number n. The following table provides information:

n	Probability	Min	Max
10	95 %	0.70	1.75
	99 %	0.63	2.15
	99.7 %	0.59	2.48
100	95 %	0.88	1.16
	99 %	0.84	1.22
	99.7 %	0.82	1.26
1000	95 %	0.96	1.05
	99 %	0.95	1.06
	99.7 %	0.94	1.07
10000	95 %	0.986	1.014
	99 %	0.982	1.019
	99.7 %	0.979	1.021

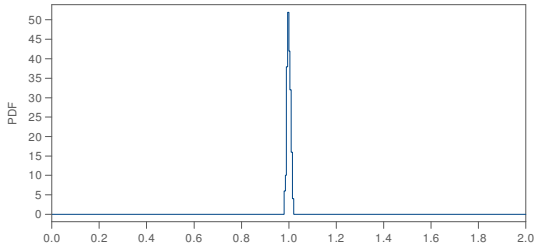
Table: Confidence interval for a measurement uncertainty of 1.0 with underlying normal distribution

The table is interpreted as follows: if n values are taken for the calculation, the precise value of the measurement uncertainty lies within the interval [Min · [calculated measurement uncertainty], Max · [calculated measurement uncertainty]] with the specified probability.

Example:

If 100 values are taken for the calculation and the calculated measurement uncertainty is 1000 Nm, the measurement uncertainty lies within the range [0.84 · 1000 Nm, 1.22 · 1000 Nm] = [840 Nm, 1220 Nm] with a probability of 99 %

From the table you can see how inaccurate the measurement uncertainty is when n = 10. That is generally inadequate, since the true value could be more than twice as large as the calculated value! If n = 100, the measurement uncertainty is accurate to approx. 20 %. That could be acceptable in many applications. If n = 1000 it is already accurate to about 6 %. It can also clearly be seen that increasing the number of random samples from 100 to 10,000 (i.e. a factor of 100) makes the confidence interval about 10 times smaller, i.e. the standard measurement uncertainty is determined 10 times more accurately.



Example distribution of the calculated measurement uncertainty. True value = 1 and n = 10,000

The table results from the determination of the confidence interval for the unknown variance with a known mean value. To do this, the chi-squared distribution with n degrees of freedom is used. Confidence interval = $[\sqrt{n/c1} \cdot uc, \sqrt{n/c2} \cdot uc]$

where uc = calculated measurement uncertainty, c1 = 1-α/2 quantile, c2 = α/2 quantile. For example, if n = 100 and the probability is 95 %, the value α = 0.05, hence the quantiles 129.6 and 74.2, hence the confidence interval = [0.88 · uc, 1.16 · uc]

Mean value:

imc FAMOS is based on the GUM, but doesn't follow it in all respects. There is a distinction, for example, between the determination of the mean value and that of the standard deviation.

imc FAMOS assumes that the best result of the algorithm is calculated in the first (or, equivalently, also in the last) pass of the loop. In all other Monte Carlo experiments additional noise is added. The purpose is to simulate the effect of noise on the result and ultimately to determine the measurement uncertainty. However, the result in each individual Monte Carlo experiment is certainly not better than the uninfluenced result of the first pass. On the contrary, it is worse.

The GUM clearly states: "The measured result itself is the best estimate of the true value." imc FAMOS follows this statement in that the first/last pass of the loop supplies the best result.

The GUM on the other hand proposes that the mean value and standard deviation be determined from the Monte Carlo experiments alone. The mean value is the best estimate according to GUM. imc FAMOS offers the user this mean value, but doesn't allow the calculation of the standard deviation to be based on it. From the point of view of imc FAMOS, one would only use the mean value of the noisy results if one didn't have the uninfluenced result.

imc FAMOS offers the mean value of all Monte Carlo experiments as one of the advanced analysis functions. The mean value is then to be interpreted as the average result in the case of added noise. If the mean value deviates from the uninfluenced result, that is to be interpreted as sensitivity of the algorithm to the superimposed interference.

If imc FAMOS calculates the standard deviation for the purpose of calculating the measurement uncertainty, then the mean value appears in the equation for the standard deviation. imc FAMOS uses the best estimate at this point: i.e. not the mean value of the Monte Carlo experiments, but the value from the first pass of the loop – the uninfluenced result.

The aim of the measurement uncertainty calculation in imc FAMOS is not the determination of the mean value, but the determination of the measurement uncertainty for the existing result, i.e. precisely the uninfluenced result.

Some arithmetic operations obtain the mean value, i.e. with a high number of Monte Carlo experiments the mean value strives towards the uninfluenced result. These are, for example, linear operations such as multiplication by a fixed factor or the addition of a constant offset. If the noise itself has no mean value, then the linearly calculated noise will also have no mean value.

Many even quite simple arithmetic operations cause a deviating mean value. For example, squaring or rectification: the mean value of rectified noise that was previously without a mean value is no longer zero.

If the mean value of the Monte Carlo experiments deviates from the uninfluenced result, this is also reflected in the calculated measurement uncertainty, because regardless of the reason why the result of a Monte Carlo experiment deviates from the uninfluenced result, it is precisely this deviation of the uninfluenced result from the noisy result that is always used.

Control of coincidence

The command UNCERTAINTY_LOOP in imc FAMOS has two parameters that can be used to control how randomly or even how reproducibly the measurement uncertainty calculation takes place and ultimately how accurate it is.

```
UNCERTAINTY_LOOP M EwInit
```

M is the number of Monte Carlo experiments.

EwInit sets the initialisation of the random number generator. You can choose whether a re-initialisation of the random number generator should take place. With a numerical value > 0 the random number generator is initialised to a fixed value. The sequence then always produces exactly the same results. If you choose a different positive initialisation value, a different sequence of random numbers is generated. The results will thus be different and yet reproducible.

Only a value of 0 prevents the re-initialisation. Other number sequences are generated with each pass of the sequence. The random number generator is initialised to a fixed value when imc FAMOS starts.

Example of the comparison of uncertainty framework and MCM within imc FAMOS

As an example, a simple evaluation is used, initially without calculation of the measurement uncertainty:

```
Input1 = 7
_In1 = Input1
Result =_In1 ^ 2
```

The result at the end is 49.

Now the same evaluation, but with calculation of the measurement uncertainty. The input data Input1 have a measurement uncertainty of 0.1:

```
Input1 = 7
UncertaintySet( Input1, "Uncertainty", 0.1)
UNCERTAINTY_LOOP 1000 1
        _In1 =UncertaintyModify( Input1)
        Result=_In1^2
        UncertaintyCalc( Result )
END
uc = UncertaintyGet( Result, "Uncertainty")
```

The measurement uncertainty of the result is uc and has the value 1.44. If the UNCERTAINTY_LOOP command is parameterised differently, other values result! If the GUM uncertainty framework were to be followed, this would result, after determination of the partial derivation, in a value of

$$2 \cdot 7 \cdot 0.1 = 1.4$$

The two are similar, as expected, but not equal.

Influencing factors and cause of deviations

The sensible application of random numbers is essential for the success of the Monte Carlo method. The GUM concentrates on the statistical distribution: the probability density function. Hence, a coherent mathematical model can also be created. The probability density doesn't model the time behaviour of the random variables, but precisely the distribution with a large number of random experiments. Typical examples of this that are frequently mentioned in the GUM are the normal distribution, the rectangular distribution and the triangular distribution.

If a random number generator that produces numbers with such a distribution is now started, then these are exclusively sequences of numbers that look like noise. As the name implies, the sequences should be random. In the case of a uniform distribution, a random number generator does not produce (0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4,...), but something more like (3, 0, 1, 2, 2, 4, 0, 2, 3, 1,...). Note that sequences exhibit the same probability density.

The Monte Carlo method is now to be used to simulate different measurements that have not taken place in reality. With the above random number generators and distributions, the individual measurements would always differ from one another by a different noise.

However, the metrological reality is different: the measurement result is influenced not only by noise, but by offset drift, induced mains hum, spikes, etc. Sometimes the spikes are there, sometimes they aren't. The hum may be larger at one time and smaller another time. These relevant influences cannot be adequately modelled by a probability density function. Nevertheless, coincidence plays an important role in these phenomena, too.

The GUM does not rule out that a random number generator is given a certain time behaviour and supplies data that exhibit a more complex probability density. The following elaborations are thus fully compliant with the GUM.

In imc FAMOS, these particular types of interference are specified by user-defined properties on the channels affected. The UncertaintyModify() function then takes note of them and carries out the corresponding application. The application changes the value – usually additively, but also multiplicatively. Several different types of interference can be meaningfully combined.

The main types of interference are listed below:

Gaussian noise

All unknown types of interference can be modelled best with this property. These includes unknown noise.

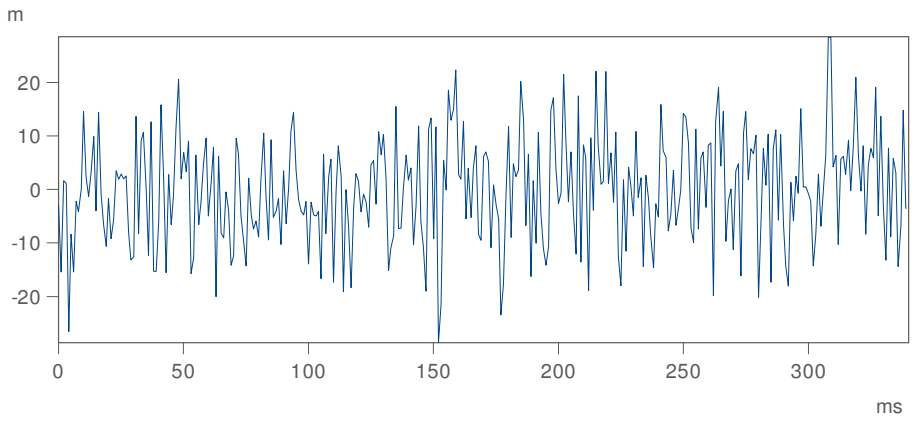
If a channel exhibits merely the standard measurement uncertainty, then (in the absence of additional information) Gaussian noise is applied.

A Gaussian interference is applied to the y-values of the input channel:

[new value]: = [old value] + random number * ["Uncertainty"]

The random number is normally distributed with a standard deviation of 1.

The standard deviation of the interference is equivalent to the value of the property.



Gaussian interference for the measurement uncertainty 10.0

Noise with rectangular distribution

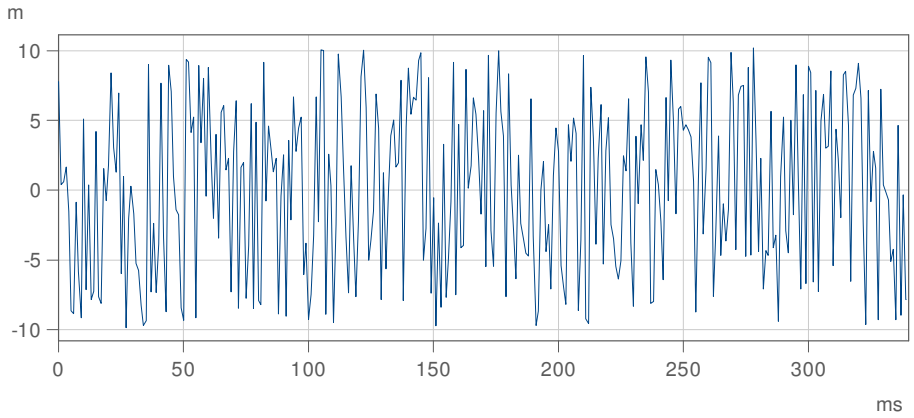
A uniformly distributed interference is applied to the y-values of the input channel:

[new value]: = [old value] + random number * ["Uncertainty Source.Rectangular"]

The random number is uniformly distributed in the range [- 1.0, + 1.0].

The property specifies the half width of the symmetric uniform distribution.

Uniformly distributed interference arises, for example, from the discretisation, i.e. the analogue-to-digital conversion and also when rounding to a certain number of decimal places, such as reading a digital multimeter or ruler or incremental encoder.



uniformly distributed noise with a parameter 10

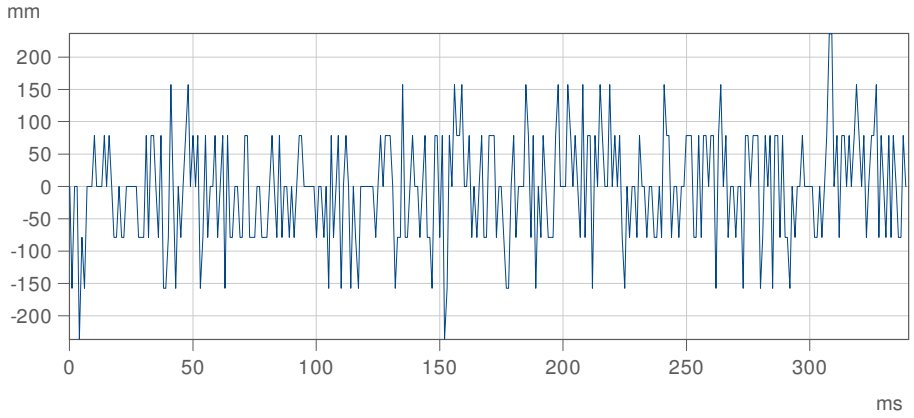
LBS noise

Interference corresponding to the bit noise of an analogue-to-digital converter (ADC) is applied to the y-values of the input channel. The procedure can be used only with data records that are stored internally in an integer format (with the possible addition of scaling information). The interference itself is an integral multiple of a bit or LSB (least significant bit), which corresponds to the amplitude resolution.

[new value]: = [old value] + random number * ["Uncertainty Source.LSBs"]

The random number is an integral multiple of an LSB and fluctuates around zero.

The standard deviation of the interference is equivalent to the value of the property.



Error of 1 LSB effective. 1 LSB corresponds to approx. 80 mm

Gain error

A deviation of the amplitude is considered. Thus, for example, an unknown gain error of a measurement amplifier is modelled. In the case of a sensor this is a deviation in the sensitivity. This deviation is constant during the whole measurement. A measurement in this sense corresponds to a pass of the UNCERTAINTY_LOOP.

The deviation is specified relative to the value of 1.0, i.e. as a dimensionless proportion. For example, a value of 0.01 means that the amplitude deviates by approx. 1 % on average over many measurements. The measured value (y value of the input data) is changed by means of the following equation:

A = 1 + random number * ["Uncertainty Source.Amplitude"]

[new value]: = [old value] * A

The random number is normally distributed with a standard deviation of 1.

A is determined once only for the whole measurement.

Offset error

A zero point deviation is considered. This deviation is constant during the whole measurement. A measurement in this sense corresponds to a pass of the UNCERTAINTY_LOOP.

The offset is specified as a real number in the y-unit of the channel.

The measured value (y value of the input data) is changed according to the following equation:

Off = random number * ["Uncertainty Source.Offset"]

[new value]: = [old value] + Off

The random number is normally distributed with a standard deviation of 1.

Off is determined once only for the whole measurement.

Offset drift

Drift means a gradual change in the offset during the measurement.

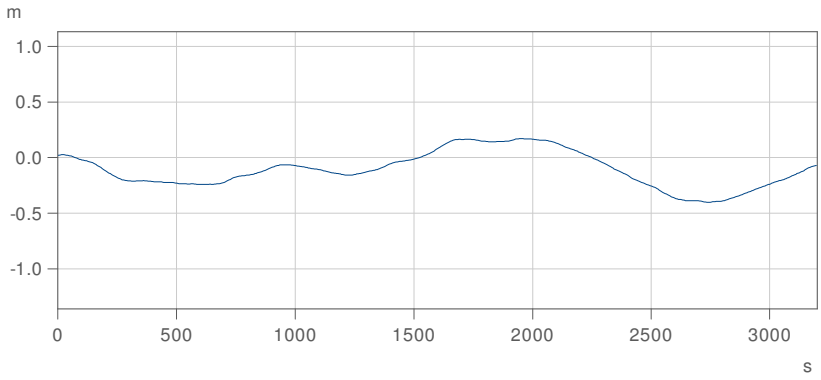
"Uncertainty Source.Drift Offset": measure of the size of the drift, specified as a real number in the y-unit of the channel.

"Uncertainty Source.Drift Time": measure of the change of the drifting offset over time, specified as a real number in the x-unit of the channel. During this time, a noticeable change in the offset takes place. The specification itself is more of a qualitative nature, but the time behaviour changes in proportion to this property.

The measured value changes according to the following equation:

[new value]: = [old value] + ["Uncertainty Source.Drift Offset"] * drift value

The value of the drift is a random number and has a standard deviation of approximately 1.



Drift with offset 1 m and time = 100 s

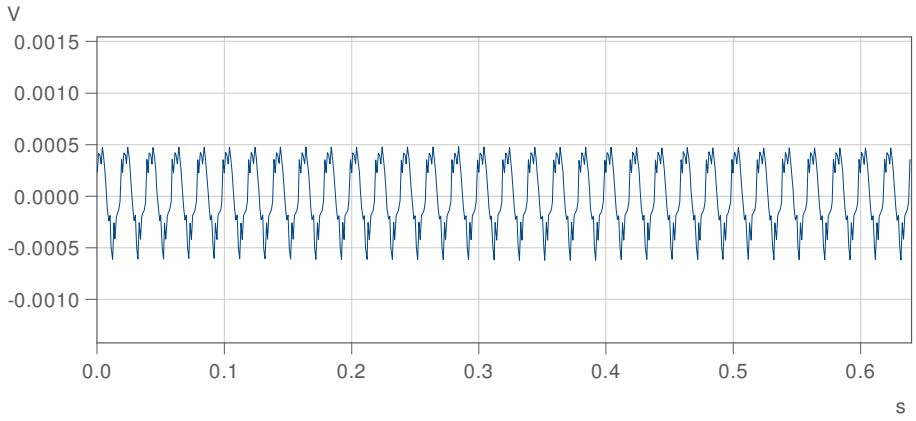
Hum, mains hum

A mains hum is applied to the signal.

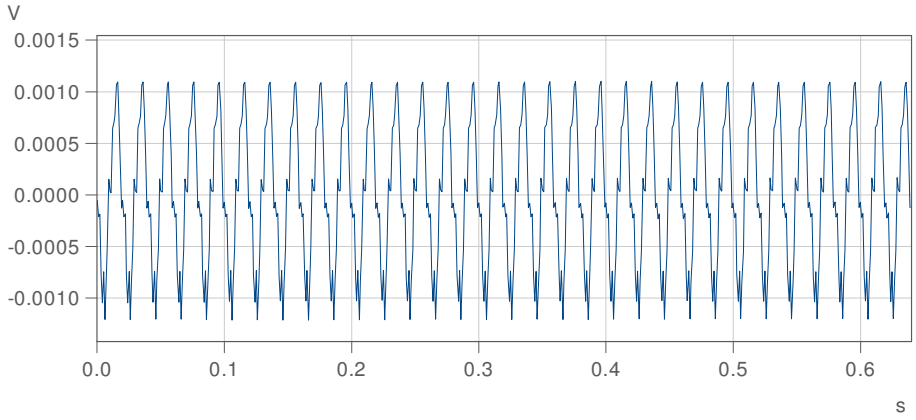
"Uncertainty Source.Hum Amplitude": the amplitude of the fundamental oscillation, specified as a real number in the y-unit of the channel, greater than zero. The amplitude is constant during the whole measurement. A measurement in this sense corresponds to a pass of the UNCERTAINTY_LOOP. Viewed over a great many measurements, the amplitude is normally distributed. The standard deviation of the normal distribution is specified here as a property. The phase of the fundamental oscillation varies from measurement to measurement and is subject to a uniform distribution.

"Uncertainty Source.Hum Frequency": the frequency of the fundamental oscillation, specified in the reciprocal value of the x-unit of the channel. The frequency is the same during all measurements. For example, 50 Hz or 60 Hz.

"Uncertainty Source.Hum Harmonics": the ratio of the power of all harmonics to the power of the fundamental oscillation. Hence, for example, a value of 0.01 means that the total power of all harmonics is 1 % of the power of the fundamental oscillation. If this property is not specified, then 0.1 is assumed.



Mains hum where parameter amplitude = 0.001 V, frequency = 50 Hz, harmonics = 0.1



A following Monte Carlo pass with the same parameters

Interference pulses, spikes

Rare interference pulses, noise peaks and spikes can be superimposed on the signal. Size and temporal behaviour are defined.

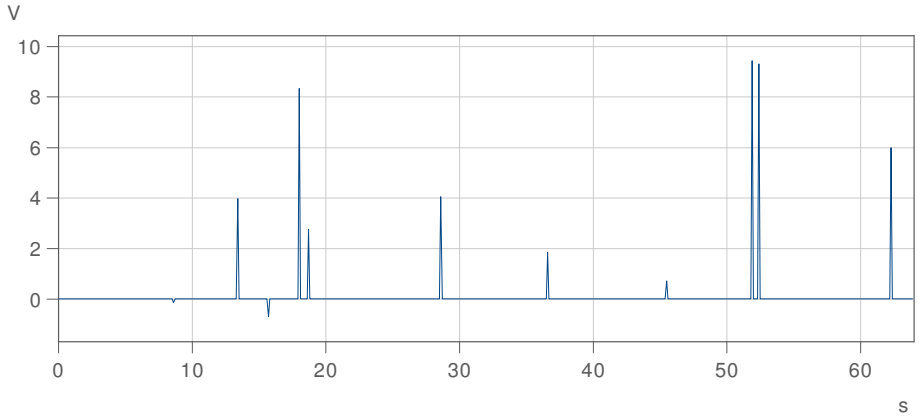
The size of the interference pulses lies between a minimum and a maximum value. Each individual interference pulse and every single measuring point of an interference pulse is given an individual random value. The values are uniformly distributed. The signal remains unchanged between the individual interference pulses.

"Uncertainty Source.Spikes Max": the maximum value of the amplitude of the interference pulse, specified as a real number in the y-unit of the channel.

"Uncertainty Source.Spikes Min": the minimum value of the amplitude of the interference pulse, specified as a real number in the y-unit of the channel.

"Uncertainty Source.Spikes Width": the width of the interference pulse, specified as a real number in the x-unit of the channel. The maximum width is specified. The width itself is uniformly distributed and randomly determined for each pulse.

"Uncertainty Source.Spikes Time": distance between the interference pulse, specified as a real number in the x-unit of the channel. This is measured from the start of one interference pulse to the start of the next.



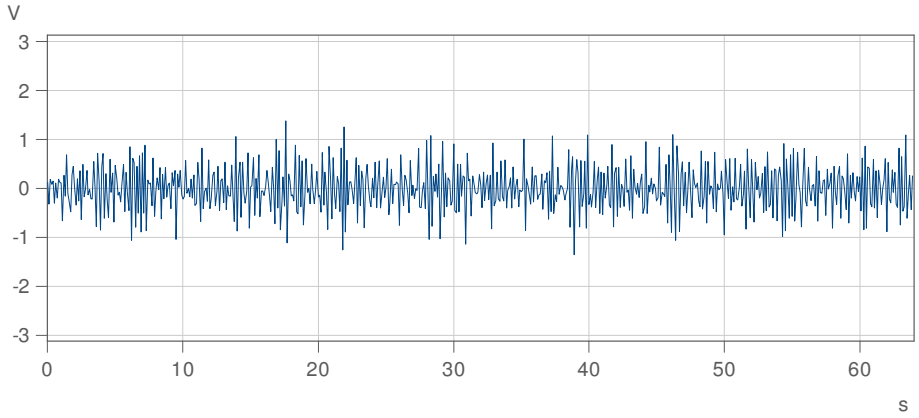
Spikes where Min = -2 V, Max = 10 V, Width = 0.1 s, Time = 10 s

Higher-frequency noise

Noise that contains all higher frequencies.

"Uncertainty Source.Noise RMS": noise is applied to the y-values of the input channel. The standard deviation of the interference is equivalent to the value of the property.

"Uncertainty Source.Noise Frequency": The lower limit frequency of the noise, specified in the reciprocal value of the x-unit of the channel. The lower limit frequency is the same during all measurements.



Noise above 2 Hz, RMS = 0.01 V, sampling frequency = 10 Hz

Test run

It is recommended to carry out a test run to check the form of the interference. Most interference can be assessed well using a signal formed from [input signal * 0.0]. In the case of the gain error, it is [input signal * 0.0 + 1.0].

Example: Voltage peaks

A measured voltage signal contains some voltage peaks. The impact of these peaks on the result of the analysis will be examined for further calculation.

UncertaintySet(Voltage, “Uncertainty Source.Spikes Max”,300) ; in [V]

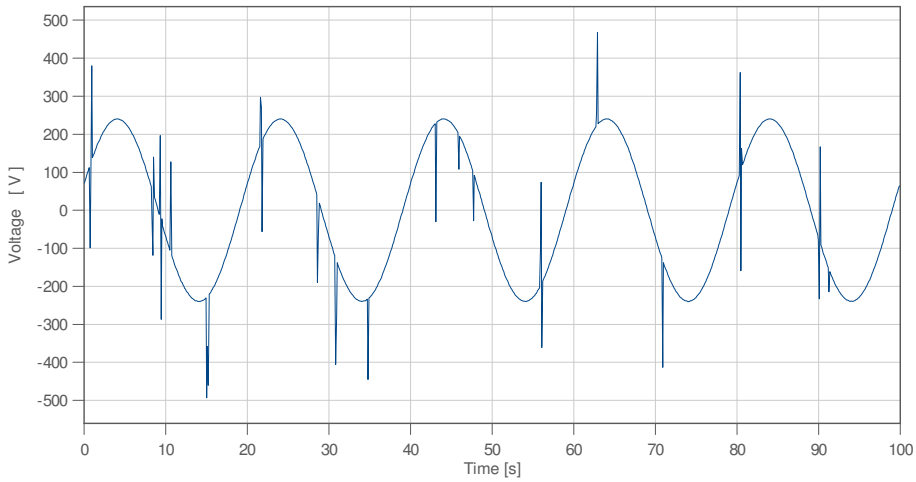
UncertaintySet(Voltage, “Uncertainty Source.Spikes Min”,-300) ; in [V]

UncertaintySet(Voltage, “Uncertainty Source.Spikes Width”,0.001) ; in [sec]

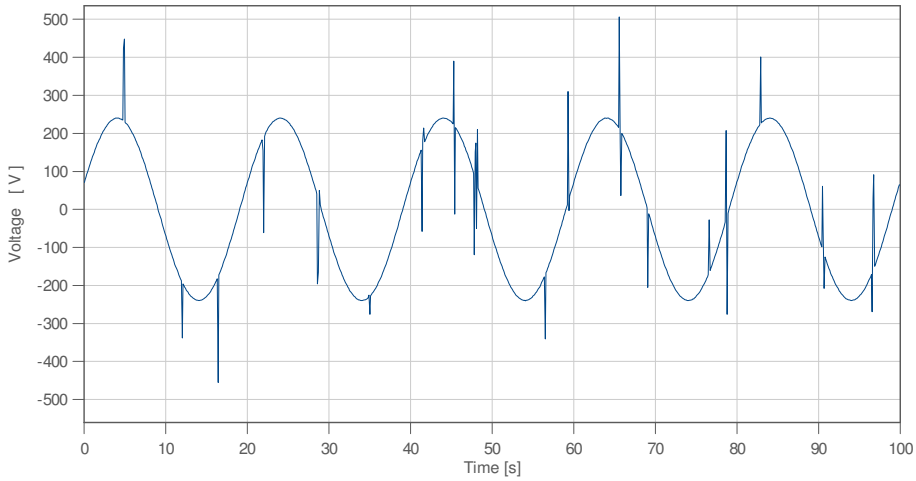
UncertaintySet(Voltage, “Uncertainty Source.Spikes Time”,0.01) ; in [sec]

Voltage peaks are applied to input data. Using the "UncertaintySet()" function, the maximum and minimum values of the peaks are set, together with the width and frequency.

The signal itself is sinusoidal and various interference peaks are now applied to it in the various Monte Carlo experiments in accordance with specifications.



A further pass supplies:



User-defined interference patterns

Completely user-defined interference patterns can also be created. This is necessary, for example, if a distribution density exists that is not covered by the user-defined properties. For example, this is the case with limited distributions, because there are limits for physical reasons. For example

Mass ≥ 0 kg

Temperature ≥ 0 K

Efficiency < 1.

This is also the case with certain forms of the interference signal.

The interference is applied by generating pseudo-random numbers with a specified distribution using the “random” function:

Example:

```
_In1 = Temperature + random (leng?( Temperature),3,0,7,24 )
_In1 = uppervalue (_In1, 0 )
```

In the last line the temperature is limited to 0 K, assuming that the channel is scaled in K.

Similarly, the overlaying of automatic and user-defined interference is possible:

```
_In1 = UncertaintyModify (Input1) + random (leng?(Input1),2,0,0,0)
```

You must ensure that you always use non-correlated random numbers. This applies to all channels within a loop pass and also between the loop passes.

Conclusion:

imc FAMOS enables you to model all different types of interference variables involved in the input signal for further calculation. The effects of such influences on a mathematical algorithm can be investigated. The resulting measurement uncertainty can be determined.

Additional information:

imc Test & Measurement GmbH
Voltastr. 5
13355 Berlin, Germany

Telephone: +49 (0)30-46 7090-0
Fax: +49 (0)30-46 31 576
E-mail: hotline@imc-tm.de
Internet: http://www.imc-tm.com

imc Test & Measurement GmbH is a manufacturer and solution provider of productive test and measurement systems. imc implements metrological solutions for research, development, service and production. imc has particular expertise in the design and production of turnkey electric motor test benches. Precisely outfitted sensor and telemetry systems complement our customer applications.

Our customers from the fields of automotive engineering, mechanical engineering, railway, aerospace and energy use imc measurement devices, software solutions and test stands to validate prototypes, optimize products, monitor processes and gain insights from measurement data. As a solution provider, imc offers their customers an attractive and comprehensive range of services. These include project consulting, contracted measurements, data evaluation, specialist deployment, customer-specific software development and system integration. imc consistently pursues its claim of providing services for “productive testing”.

If you would like to find out more specific information about imc products or services in your particular location, or if you are interested in becoming an imc distributor yourself, please go to our website where you will find both a world-wide distributor list and more details about becoming an imc distributor yourself:

http://www.imc-tm.com/our-partners/