

Messunsicherheitsbestimmung in imc FAMOS – Die Monte-Carlo-Methode

White Paper

Was ist eine Monte-Carlo-Simulation?

Die Monte-Carlo-Simulation ist ein Verfahren aus der Stochastik, bei dem sehr häufig durchgeführte Zufallsexperimente die Basis darstellen. Mit Hilfe der Wahrscheinlichkeitstheorie gilt es analytisch unlösbare Probleme numerisch zu lösen. Die Zufallsexperimente werden über geeignete Zufallszahlen erzeugt. Die Monte-Carlo-Methode (MCM) benötigt in der Regel eine große Anzahl von Zufallsexperimenten, um aussagekräftig sein.

Monte-Carlo-Simulationen werden in vielen Bereichen angewendet (z.B. in der Finanzmathematik, im Risikomanagement, in der Kern- und Teilchenphysik, oder in der Prozess- und Regelungstechnik). Darüber hinaus eignet sich die Monte-Carlo-Simulation auch zur Bestimmung der Messunsicherheit. Die Monte-Carlo-Simulation kann sowohl aufwendige und teure Laborexperimente ersetzen als auch bei der Erhebung großer Datenmengen helfen.

Vorteile des Verfahrens

- Bearbeitung von Nichtlinearitäten, komplexe Algorithmen, bei von Normalverteilungen abweichende Verteilungen
- Häufig die einzige Simulationsmethode, die in vernünftiger Rechenzeit brauchbare Resultate liefert
- Unter Einsatz von mehr Rechenzeit systematische Verbesserung der Lösung

Grundsätzliches Vorgehen

In der Regel werden bis zu 10 Millionen (zwischen 10⁴-10⁷) Simulationen des gleichen Problems unter Variation der Werte der Einflussgrößen durchgeführt. Anschließend sind die einzelnen Simulationsergebnisse nach statistischen Gesichtspunkten auszuwerten, d.h. es wird der Erwartungswert und Standardabweichung berechnet.

Der Kern von Monte-Carlo-Simulationen ist das Beaufschlagen von Zufallsgrößen. Man beaufschlagt diese Zufallsgrößen auf das zu untersuchende Problem (in unserem Fall die Messreihe) und erzeugt so eine Anzahl an ungenauen Kopien des zu untersuchenden Sachverhaltes. Die Zufallszahl wird dabei jedes Mal „gewürfelt“, man spricht von einem random walk. Über das Ensemble an Kopien wird nun der Mittelwert und die Standardabweichung gebildet. Bei manchen Systemen ist es erforderlich, einen „gezinkten“ Würfel zu benutzen, der eine Folge von Zufallszahlen mit vorgegebenen statistischen Eigenschaften generiert (z.B. eine bestimmte Art der Verteilung: Rechteckverteilung, Dreieckverteilung, Normalverteilung).

Die Monte-Carlo-Methode in imc FAMOS

Famos wendet die Monte-Carlo Methode (MCM) an. Im GUM sind mehrere Berechnungsmethoden angegeben. Die direkte analytische Berechnung ist für viele einfache Formeln direkt möglich. Aber für die meisten Befehle von Famos praktisch nicht durchzuführen. Die MCM Methode hat dabei keine Probleme. Damit ergibt sich ein systematischer Weg für beliebige Algorithmen.

Das MCM Verfahren beruht darauf, die Eingangsdaten (leicht) zu verrauschen. Dazu werden sie mit Zufallszahlen beaufschlagt, um Abweichungen zu simulieren. Diese Abweichungen wirken sich als Abweichungen der Ergebnisse des zu rechnenden Algorithmus aus. Die Standardabweichung wird ermittelt und ergibt die Messunsicherheit der Ergebnisse.

Es sei angemerkt, dass das Verrauschen nicht unbedingt heißt, dass es sich um das Addieren eines Rauschens im engeren Sinn handelt. Es kann das Beaufschlagen mit diversen Formen von Störeinflüssen sein.

Das MCM Verfahren ist ein statistisches Verfahren und verlangt eine hinreichend große Anzahl von Zufallszahlen, um aussagekräftig zu sein.

Das GUM uncertainty framework wird berücksichtigt, aber nicht benutzt.

Da der GUM für den Zahlenwert der erweiterten Messunsicherheit eine symmetrische Verteilung annimmt, bestimmt auch imc FAMOS das kürzeste Überdeckungsintervall als symmetrisches. Wenn eine feinere Analyse, vor allem bei unsymmetrischen Verteilungen, erforderlich ist, kann dies durch eine erweiterte Analyse geschehen.

Anzahl der Monte-Carlo-Versuche

Die Anzahl der Monte-Carlo-Versuche kann über die Funktion „Uncertainty_LOOP“ eingegeben werden. Es sei angemerkt, dass das Verrauschen nicht unbedingt heißt, dass es sich um das Addieren eines Rauschens im engeren Sinn handelt. Es kann das Beaufschlagen mit diversen Formen von Störgrößen sein (Rauschen, Netzbrummen, Offset-Drift). Die Ergebnisse des Algorithmus erhalten die anwenderdefinierte Eigenschaft „Uncertainty“, diese ist als Standardabweichung angegeben. Die Schleife Uncertainty_LOOP in Famos ist nötig zum Durchführen aller M Monte-Carlo Versuche. Die Schleife wird M+2 mal durchlaufen: Ein erster Durchlauf berechnet das unbeeinflusste Ergebnis aus den unbeeinflussten Eingangsdaten. Es folgen die M Monte-Carlo Durchläufe. Dabei werden die Eingangsdaten verrauscht. Von Durchlauf zu Durchlauf wird der Schätzwert für die Messunsicherheit verbessert. In einem letzten Durchlauf werden alle unbeeinflussten Werte wieder hergestellt. Die Berechnung der Messunsicherheit wird umso zuverlässiger und genauer, je mehr Stichproben in die Berechnung einfließen. Wichtig ist aber auch, dass die Stichproben selbst gut verteilt sind, um den möglichen Raum der Stichproben gut abzudecken.

Die erweiterten Analyse-Funktionen (uc, mean, pdf, pdf0) helfen bei der Feststellung, ob die Anzahl M der Monte-Carlo Versuche ausreichend hoch ist. Falls zu niedrig, ist die Verteilungsdichte keine schöne glatte Kurve. Auch die Berechnungen von „uc“ und „mean“ schwanken dann stark. Das Schwanken der Berechnungen von „min/max“ sollte nicht wesentlich zur Beurteilung beitragen, da bei diesen Größen keine mittelnde Wirkung mit wachsender Anzahl M eintritt.

In den folgenden Abbildungen können Sie erkennen, wie die Verteilungsdichte mit der Anzahl der Monte-Carlo-Versuche immer glatter wird.

Erstes Beispiel

Der Temperaturdatensatz hat beispielsweise 5.000 Messwerte (Abtastrate 100 Hz, Abbildung 1). Aus dieser Zeitreihe wird ein Ergebnis (Anstiegszeit) ermittelt. Es werden hierbei einige Einzelwert-Hilfsvariablen erzeugt.

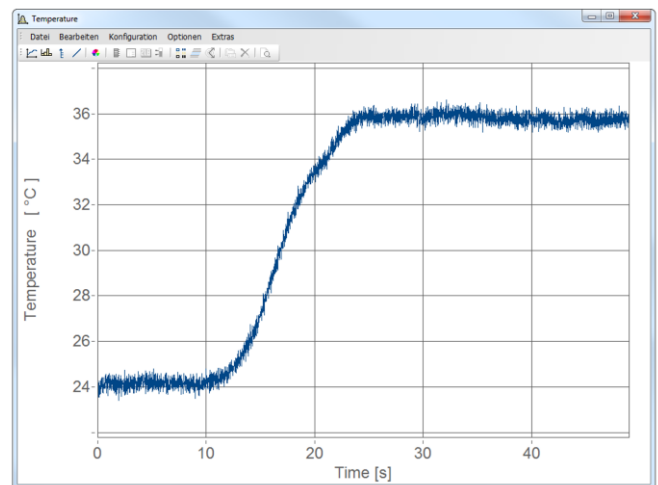


Abbildung 1

In den folgenden Abbildungen sind die Wahrscheinlichkeitsdichtefunktion für M=100, 1000 und 10.000 Versuche dargestellt. Das zur Berechnung der Hilfsvariablen schon sehr viele Werte eingeflossen sind, hat hier keine Auswirkung. Es zeigt sich, dass 10.000 Versuche nötig sind, um eine ausreichend glatte Kurve zu erhalten (siehe Abbildungen 2).

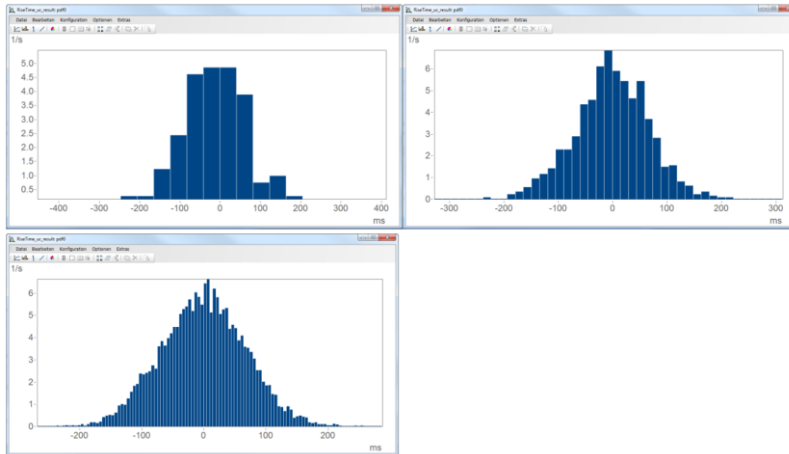
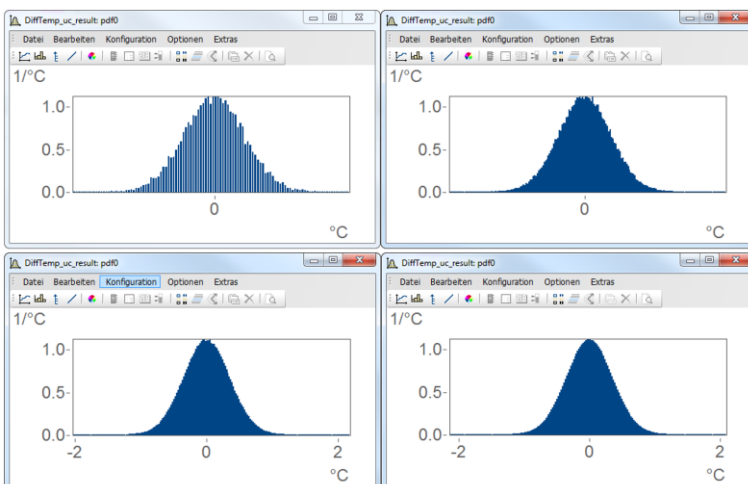


Abbildung 2

Zweites Beispiel

Differenz zweier Temperaturen: Gegeben sind zwei Temperaturmessungen, die voneinander subtrahiert werden. Die Temperaturmessungen enthalten beide 1000 Werte. Beiden Kanälen wird eine Messungenauigkeit zugeordnet. Bei jedem Monte-Carlo-Versuch wird auf jeden der beiden Kanäle ein Rauschen beaufschlagt und jedes Wertepaar wird bei jedem Durchlauf miteinander verrechnet. Wie man den folgenden Abbildungen entnehmen kann, können hier $M=10$ Durchläufe ausreichend sein, um eine ausreichend glatte Wahrscheinlichkeitsdichte-Funktion zu erhalten. Grund dafür ist, dass in diesem Fall $10 \cdot 1000$ Werte in die Berechnung einfließen.



Fazit

Wie viele Schleifendurchläufe letztlich zu durchlaufen sind, ist von der Aufgabenstellung abhängig. Doch zeigt es sich, dass mit der Verrechnung von Datenreihen schon eine kleine Anzahl an Monte-Carlo-Versuchen zu einem guten Ergebnis führen kann.

Qualitative Erklärung

Das MCM-Verfahren wird in imc FAMOS nicht benutzt, um den Messwert selbst zu berechnen, sondern nur dessen Messunsicherheit. Die Messunsicherheit muss i.a. nicht mit vielen Nachkommastellen ermittelt werden. In diesem Sinn ist es sicher relevant, ob die Messunsicherheit 2mV oder 3mV beträgt. Eine Angabe der Messunsicherheit von 2.3971mV ist sicher nicht angemessen.

z.B. soll aus vorhandenen Einzelwerten für Kraft und Weg die Arbeit in einer Formel bestimmt werden. Eine Anzahl $M = 10$ Monte-Carlo Versuche ist sicher zu wenig. Wenn die komplette Bestimmung der Messunsicherheit ausgeführt wird, erhält man jedes Mal ein stark abweichendes Ergebnis. Daran ist zu erkennen, dass $M = 10$ Stichproben zur Bestimmung der Standardabweichung nicht ausreichen.

Bei $M = 100$ werden die Schwankungen deutlich geringer, aber noch gut sichtbar. Bei $M = 1000$ scheint das Ergebnis stabil zu sein.

Z.B. liegt eine Zeitreihe vor, die 10000 Messwerte enthält. Aus dieser Zeitreihe wird ein Ergebnis durch Filterung etc. ermittelt. Das Ergebnis selbst ist ein Einzelwert. Hier kann sich zeigen, dass bereits nach $M = 50$ Monte-Carlo Versuchen ein recht gutes Ergebnis erzielt wird.

Denn in die Berechnung der Standardabweichung fließen $n=10000 \cdot 50$ Werte ein. Es gibt Algorithmen, die abhängig vom Messwert unterschiedlich verzweigen. Für eine ausreichende Genauigkeit ist es wichtig, dass jeder Zweig der Berechnung ausreichend oft durchlaufen wird.

Für eine ausreichende Genauigkeit der ermittelten Messunsicherheit ist die ausreichend genaue Bestimmung der Messunsicherheiten und Verteilungsdichten der Eingangsdaten Voraussetzung. In vielen Fällen ist trotz vermeintlicher Präzision und korrekt ausgefülltem Messunsicherheitsbudget die Schätzung der Messunsicherheiten der Eingangsdaten eher grob. Die Bestimmung der Verteilungsdichten kann in vielen praktischen Fällen nicht erfolgen. Dann wird auf eine Normalverteilung ausgewichen. Das ist aber oft eben auch nur eine (grobe) Näherung. Daraus folgt, dass eine vermeintlich hohe Präzision einer ermittelten Messunsicherheit nicht wirklich präzise sein muss.

Das Monte-Carlo Verfahren konvergiert mit \sqrt{M} . Für eine deutlich stabilere Messunsicherheit muss die Anzahl der Monte-Carlo Versuche quadratisch gesteigert werden. Also z.B. Verbesserung um Faktor 10 (eine dezimale Stelle genauer) über eine Erhöhung der Anzahl um dem Faktor 100. Das ist i.a. praktisch nur für kleine Algorithmen bzw. Einzelwertrechnungen möglich.

Für die Bestimmung der erweiterten Messunsicherheit gilt noch die folgende Überlegung: Die erweiterte Messunsicherheit wird aus der empirisch ermittelten Verteilungsdichte geschätzt. Ist z.B. eine Überdeckungswahrscheinlichkeit von 99% gefordert, so darf 1% der Werte der Verteilungsdichte außerhalb des Überdeckungsintervalls liegen. Dieses 1% darf aber nicht nur aus einem Wert bestehen, sondern aus vielen. Wird z.B. mit $n=1000$ Werten gearbeitet, so wird das Überdeckungsintervall so gewählt, dass 10 Werte außerhalb liegen. Der Wert der erweiterten Messunsicherheit

hängt also von diesen 10 zufälligen Werten ab. Da kann noch nicht von hoher Präzision gesprochen werden. Allerdings sind $n=10000$ Werte nötig, damit 100 Werte außerhalb liegen, was schon deutlich besser ist.

Quantitative Erklärung

Sei n die Anzahl der an der Berechnung der Messunsicherheit beteiligten Werte. Diese ergibt sich aus: $n = [\text{Anzahl Monte-Carlo Versuche } M] \cdot [\text{Anzahl Messwerte im Ergebnis}]$
Beispiel: Bei $M=10$ Monte-Carlo Versuchen und einer Länge des Ergebnisses von 10 Werten, ergibt sich $n=100$. Aus der Anzahl n lässt sich abschätzen, wie zuverlässig die Berechnung der Messunsicherheit ist. In Abbildung 4 sehen Sie eine beispielhafte Verteilung für $n=100$ und einen wahren Wert von 1.

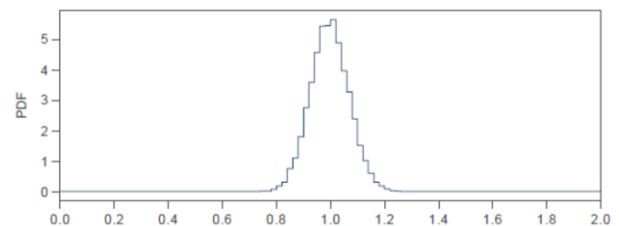


Abbildung 4: PDF (Abk. Power Density Function)

Dazu wird die Messunsicherheitsbestimmung viele Male wiederholt. Die präzise Messunsicherheit des Ergebnisses sei 1.0. Jedes Mal ist das Ergebnis leicht anders. Bei $n=100$ z.B. liegen die Werte meist im Bereich ganz grob $[0.8, 1.2]$, siehe Bild. Die meisten natürlich sehr nahe an 1. Die Messunsicherheit ist auf etwa ± 0.2 genau bestimmt. Zu beachten ist, dass der präzise Wert von 1.0 unbekannt ist. Bei einer durchgeführten Messunsicherheitsbestimmung erhält man z.B. den Wert 0.8. Das ist dann so zu deuten:

Der Wert kann um den Faktor $1/0.8$ zu klein. Würde man bei einem anderen Mal einen Wert von 1.2 errechnen, dann wäre er um den Faktor 1.2 zu groß. Also ist das Intervall = $[0.8 \cdot 1/1.2, 0.8 \cdot 1/0.8]$ bzw. mit einer gewissen Wahrscheinlichkeit liegt der wahre Wert in dem Intervall. Wie auch beim Überdeckungsintervall wird das so gedeutet: Wenn man

ganz viele Messunsicherheitsbestimmungen durchführt, dann liegt bei dem Anteil von Durchführungen (der der gegebenen Wahrscheinlichkeit entspricht) der wahre Wert im Intervall. So lässt sich z.B. für n=100 sagen: Mit einer Wahrscheinlichkeit von 95% liegt der Wert im Intervall [0.88· [Berechnete Messunsicherheit], 1.16· [Berechnete Messunsicherheit]]. Voraussetzung ist die Normalverteilung der Messwerte. Mit größerer Anzahl n wird das Ergebnis genauer. Folgende Tabelle gibt Aufschluss:

n	Wahrscheinlichkeit	Min	Max
10	95%	0,7	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

Tabelle:
Konfidenzintervall für Messunsicherheit von 1.0 bei zugrunde liegender Normalverteilung

Die Tabelle wird folgendermaßen gedeutet:
Bei Berechnung über n Werte liegt mit der angegebenen Wahrscheinlichkeit der präzise Wert für die Messunsicherheit im Intervall [Min· [Berechnete Messunsicherheit], Max· [Berechnete Messunsicherheit]]

Beispiel:

Bei Berechnung über 100 Werte und errechneter Messunsicherheit von 1000Nm liegt mit der Wahrscheinlichkeit von 99% die Messunsicherheit im Bereich: $[0.84 \cdot 1000\text{Nm}, 1.22 \cdot 1000\text{Nm}] = [840\text{Nm}, 1220\text{Nm}]$

Aus der Tabelle können Sie entnehmen, wie ungenau die Messunsicherheit bei n = 10 ist. Das ist i.a. nicht ausreichend, da der wahre Wert mehr als doppelt so groß wie der errechnete sein kann! Bei n = 100 ist Messunsicherheit auf ca. 20% genau. Damit kann man in vielen Anwendungen schon gut auskommen. Bei n=1000 ist sie schon auf ca. 6% genau. Auch ist deutlich zu erkennen, dass bei Erhöhung von 100 auf 10000 Stichproben (also um den Faktor 100) das Konfidenzintervall ca. 10mal so klein wird, also die Standardmessunsicherheit 10mal so genau ermittelt wird.

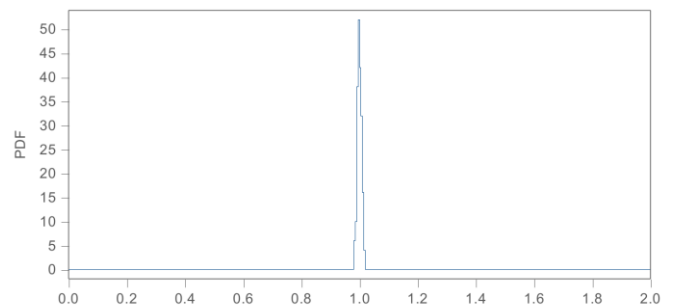


Abb. 5:
Beispielhafte Verteilung der ermittelten Messunsicherheit. Wahrer Wert = 1 und n = 10000

Die Tabelle ergibt sich aus der Bestimmung des Konfidenzintervalls für die unbekannte Varianz bei bekanntem Mittelwert. Dazu wird die Chi-Quadrat-Verteilung mit n Freiheitsgraden angewendet. Konfidenzintervall =

$$\left[\sqrt{\frac{n}{c_1}} \cdot uc, \sqrt{\frac{n}{c_2}} \cdot uc \right]$$

mit uc = errechnete Messunsicherheit, $c_1 = 1 - \alpha/2$ Quantil, $c_2 = \alpha/2$ Quantil. Zum Beispiel ist mit $n = 100$ und Wahrscheinlichkeit 95% der Wert $\alpha=0.05$, damit die Quantile 129.6 und 74.2, damit das Konfidenzintervall = $[0.88 \cdot uc, 1.16 \cdot uc]$

Mittelwert

imc FAMOS orientiert sich am GUM, folgt diesem aber nicht in allen Punkten. Eine Unterscheidung gibt es beispielsweise bei der Bestimmung von Mittelwert und Standardabweichung.

imc FAMOS geht davon aus, dass das beste Ergebnis des Algorithmus im ersten (bzw. äquivalent auch im letzten) Durchlauf der Schleife errechnet wird. In allen anderen Monte-Carlo Versuchen wird ein zusätzliches Rauschen addiert. Der Sinn ist das Simulieren der Auswirkung des Rauschens auf das Ergebnis und letztendlich das Ermitteln der Messunsicherheit. Aber das Ergebnis in jedem einzelnen Monte-Carlo Versuch ist sicher nicht besser als das unbeeinflusste Ergebnis des ersten Durchlaufs. Im Gegenteil, es ist schlechter. Der GUM sagt eindeutig: „Das Messergebnis selbst ist der beste Schätzwert für den wahren Wert.“ imc FAMOS folgt dieser Aussage, indem der erste/letzte Durchlauf der Schleife das beste Ergebnis liefert.

Der GUM schlägt andererseits vor, allein aus den Monte-Carlo Versuchen Mittelwert und Standardabweichung zu ermitteln. Der Mittelwert ist nach GUM die beste Schätzung. imc FAMOS bietet dem Anwender diesen Mittelwert an, lässt aber die Berechnung der Standardabweichung nicht auf diesem Mittelwert basieren. Aus Sicht von imc FAMOS würde man den Mittelwert der verrauschten Ergebnisse nur nutzen, wenn man das unbeeinflusste Ergebnis nicht hätte.

imc FAMOS bietet den Mittelwert über alle Monte-Carlo Versuche als eine der erweiterten Analyse-Funktionen. Der Mittelwert ist dann zu deuten als das mittlere Ergebnis im Fall von hinzugefügtem Rauschen. Weicht der Mittelwert vom unbeeinflussten Ergebnis ab, ist das zu deuten als eine Empfindlichkeit des Algorithmus gegen die eingepprägten Störungen.

Wenn imc FAMOS zum Zweck der Berechnung der Messunsicherheit die Standardabweichung berechnet, dann taucht in der Formel der Standardabweichung der Mittelwert auf. imc FAMOS benutzt an dieser Stelle die beste Schätzung: Also nicht den Mittelwert der Monte-Carlo Versuche, sondern den Wert aus dem ersten Schleifendurchlauf, das unbeeinflusste Ergebnis.

Das Ziel der Messunsicherheitsbestimmung in imc FAMOS ist nicht die Bestimmung des Mittelwertes, sondern die Bestimmung der Messunsicherheit für das vorhandene, eben das unbeeinflusste Ergebnis.

Einige Rechenoperationen erhalten den Mittelwert, d.h. bei hoher Anzahl von Monte-Carlo Versuchen strebt der Mittelwert gegen das unbeeinflusste Ergebnis. Das sind z.B. lineare Operationen, wie das Multiplizieren mit festem Faktor oder Addieren eines konstanten Offsets. Wenn das Rauschen selbst mittelwertfrei ist, wird auch das linear verrechnete Rauschen mittelwertfrei sein. Viele auch durchaus sogar einfache Rechenoperationen verursachen einen abweichenden Mittelwert. Z.B. Quadrieren oder Gleichrichten: Der Mittelwert von gleichgerichtetem vorher mittelwertfreien Rauschen ist nicht mehr Null.

Falls der Mittelwert aus den Monte-Carlo Versuchen vom unbeeinflussten Ergebnis abweicht, schlägt sich das auch in der ermittelten Messunsicherheit nieder. Denn egal, aus welchem Grund das Ergebnis eines Monte-Carlo Versuchs vom unbeeinflussten Ergebnis abweicht, wird stets eben diese Abweichung vom unbeeinflussten Ergebnis zum verrauschten Ergebnis verwendet.

Steuerung des Zufalls

Der Befehl `UNCERTAINTY_LOOP` in imc FAMOS hat 2 Parameter, mit denen gesteuert werden kann, wie zufällig oder doch gar reproduzierbar die Messunsicherheitsbestimmung abläuft und wie genau sie letztendlich wird.

UNCERTAINTY_LOOP M EwInit

M Anzahl der Monte-Carlo Versuche.
EwInit legt die Initialisierung des Zufallszahlengenerators fest.

Man kann wählen, ob eine Neuinitialisierung des Zufallszahlengenerators erfolgen soll. Mit einem Zahlenwert > 0 wird der Zufallszahlengenerator auf einen festen Wert initialisiert. Stets liefert die Sequenz dann exakt dieselben Ergebnisse. Wählen Sie einen anderen positiven Initialisierungswert, wird eine andere Zufallszahlenfolge generiert. Die Ergebnisse werden also anders sein, aber wiederum reproduzierbar.

Lediglich ein Wert von 0 verhindert die Neuinitialisierung. Mit jedem Durchlauf der Sequenz werden andere Folgen generiert. Beim Start von imc FAMOS wird der Zufallszahlengenerator mit einem festen Wert initialisiert.

Beispielhafter Vergleich uncertainty framework und MCM innerhalb von imc FAMOS

Beispielhaft wird eine einfache Auswertung benutzt, zunächst ohne Bestimmung der Messunsicherheit:

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

Result hat am Ende den Wert 49.

Nun dieselbe Auswertung, aber mit Bestimmung der Messunsicherheit. Die Eingangsdaten Input1 haben eine Messunsicherheit von 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")
```

Die Messunsicherheit von Result ist uc und hat den Wert 1.44. Wird der Befehl UNCERTAINTY_LOOP anders parametrisiert, ergeben sich andere Werte! Dem GUM uncertainty Framework folgend würde sich nach Bestimmung der partiellen Ableitung ein Wert von

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

ergeben.

Beide sind erwartungsgemäß ähnlich, aber nicht gleich.

Einflussgrößen und Verursachung von Abweichungen

Wesentlich für den Erfolg der Monte-Carlo-Methode ist die sinnvolle Beaufschlagung mit Zufallszahlen. Der GUM konzentriert sich dabei auf die statistische Verteilung, die Wahrscheinlichkeitsdichte. Damit lässt sich auch ein schlüssiges mathematisches Gebäude erstellen. Die Wahrscheinlichkeitsdichte modelliert nicht das zeitliche Verhalten der Zufallsgröße, sondern eben die Verteilung bei großer Anzahl von Zufallsexperimenten. Typische und häufig im GUM zitierte Vertreter sind die Normalverteilung, die Rechteck- und Dreieckverteilung. Wird nun ein Zufallszahlengenerator angeworfen, der Zahlen mit solch einer Verteilung produziert, so sind das ausschließlich Zahlenfolgen, die wie Rauschen aussehen. Die Abfolgen sollen wie der Name impliziert eben zufällig sein. Im Fall einer Gleichverteilung liefert kein Zufallszahlengenerator (0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, ...), sondern eher so etwas wie (3, 0, 1, 2, 2, 4, 0, 2, 3, 1, ...). Wohlgermerkt, bei Folgen weisen dieselbe Wahrscheinlichkeitsdichte auf.

Mit der Monte-Carlo-Methode sollen nun verschiedene Messungen simuliert werden, die in Wirklichkeit nicht stattgefunden haben. Mit den oben genannten Zufallsgeneratoren und Verteilungen würden sich die einzelnen Messungen stets um ein anderes Rauschen voneinander unterscheiden.

Die messtechnische Realität sieht aber anders aus: Nicht nur Rauschen, sondern Offsetdrift, eingekoppeltes Netzbrummen, Spikes etc. beeinflussen das Messergebnis. Mal sind die Spikes da, mal nicht. Mal kann das Brummen größer, mal kleiner ausfallen. Diese messtechnisch relevanten Beeinflussungen lassen sich nicht hinreichend durch eine Wahrscheinlichkeitsdichte modellieren. Dennoch spielt auch bei diesen Phänomenen der Zufall eine wichtige Rolle.

Der GUM schließt nicht aus, dass ein Zufallszahlengenerator ein gewisses Zeitverhalten erhält und Daten liefert, die eine kompliziertere Wahrscheinlichkeitsdichte aufweisen. Damit sind alle folgenden Ausarbeitungen komplett konform mit dem GUM.

In imc FAMOS werden diese besonderen Störeinflüsse über anwenderdefinierte Eigenschaften an den betroffenen Kanälen bekannt gegeben. Die Funktion `UncertaintyModify()` beachtet sie dann und führt die entsprechende Beaufschlagung durch. Die Beaufschlagung verändert den Wert, meist additiv, aber auch multiplikativ. Mehrere verschiedene Störungen sind sinnvoll kombinierbar.

Im Folgenden sind die wichtigsten Störeinflüsse gelistet:

Normalverteiltes Rauschen

Alle unbekanntes Störeinflüsse lassen sich am besten mit dieser Eigenschaft modellieren.

Dazu gehört unbekanntes Rauschen.

Weist ein Kanal lediglich die Standardmessunsicherheit auf, so wird (mangels Zusatzinformation) normalverteiltes Rauschen angewendet.

Die y-Werte des Eingangskanals werden mit einer normalverteilten Störung beaufschlagt:

$$[\text{Neuer Wert}] := [\text{Alter Wert}] + [\text{Zufallszahl}] * [\text{"Uncertainty"}]$$

Die Zufallszahl ist normalverteilt mit Standardabweichung 1.

Die Standardabweichung der Störung entspricht dem Wert der Eigenschaft.

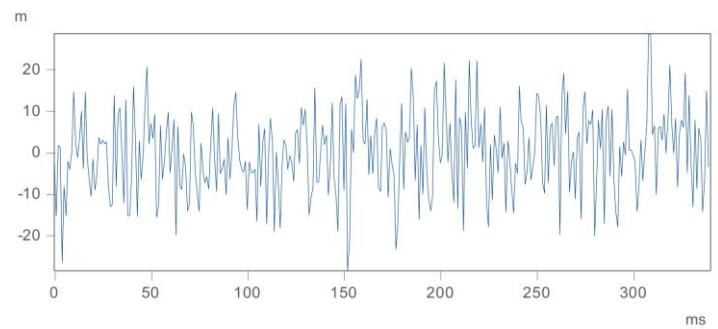


Abb. 6: normalverteilte Störung zur Messunsicherheit 10.0

Rauschen mit Rechteckverteilung

Die y-Werte des Eingangskanals werden mit einer gleichverteilten Störung beaufschlagt:

$$[\text{Neuer Wert}] := [\text{Alter Wert}] + [\text{Zufallszahl}] * [\text{"Uncertainty.Source.Rectangular"}]$$

Die Zufallszahl ist gleichverteilt im Bereich $[-1.0, +1.0]$.

Die Eigenschaft gibt die halbe Breite der symmetrischen Gleichverteilung an.

Gleichverteilte Störungen ergeben sich z.B. bei der Diskretisierung, also der Analog-Digitalwandlung. Aber auch bei einer Rundung auf eine gewisse Anzahl von Nachkommastellen, etwa beim Ablesen eines Digitalmultimeters oder Lineals oder inkrementalen Gebers.

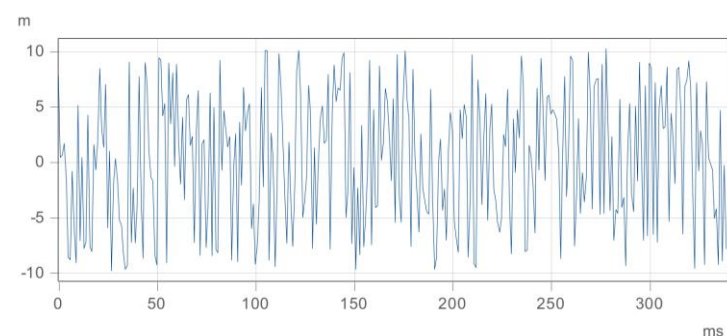


Abb. 7: gleichverteiltes Rauschen mit Parameter 10

LSB-Klappern

Die y-Werte des Eingangskanals werden mit einer Störung beaufschlagt, die dem Bitklappern eines Analog-Digitalwandler (ADC) entspricht. Das Verfahren kann nur bei Datensätzen angewendet werden, die intern in einem ganzzahligen Format (plus ggf. Skalierungsinformation) gehalten werden. Die Störung selbst ist ein ganzzahliges Vielfaches eines Bits bzw. LSB (Least Significant Bit), das der Amplitudenauflösung entspricht. $[Neuer\ Wert] := [Alter\ Wert] + Zufallszahl * [„Uncertainty\ Source.LSBs“]$ Die Zufallszahl ist ganzzahliges Vielfaches eines LSB und schwankt um Null herum. Die Standardabweichung der Störung entspricht dem Wert der Eigenschaft.

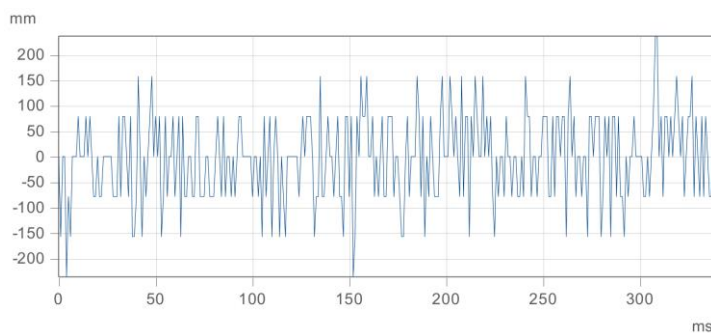


Abb. 8:
Störung von 1 LSB effektiv. 1LSB entspricht ca. 80mm

Verstärkungsfehler

Eine Abweichung der Amplitude wird nachgebildet. Damit wird z.B. ein unbekannter Verstärkungsfehler eines Messverstärkers nachgebildet. Im Fall eines Sensors ist das eine Abweichung in der Empfindlichkeit. Diese Abweichung ist während der gesamten Messung konstant. Eine Messung in diesem Sinn entspricht einem Durchlauf der Schleife UNCERTAINTY_LOOP.

Die Abweichung wird relativ zum Wert von 1.0 angegeben, also als dimensionsloser Anteil. Ein Wert von z.B. 0.01 bedeutet, dass die Amplitude im Mittel über viele Messungen um ca. 1% abweicht. Es erfolgt eine Veränderung

des Messwertes (y-Wert der Eingangsdaten) mittels der folgenden Formel:

$$A = 1 + [Zufallszahl] * [„Uncertainty\ Source.Amplitude“]$$
$$[Neuer\ Wert] := [Alter\ Wert] * A$$

Die Zufallszahl ist normalverteilt mit Standardabweichung 1.

A wird einmalig für die gesamte Messung bestimmt.

Offsetfehler

Eine Nullpunktabweichung wird nachgebildet. Diese Abweichung ist während der gesamten Messung konstant. Eine Messung in diesem Sinn entspricht einem Durchlauf der Schleife UNCERTAINTY_LOOP.

Der Offset wird angegeben als reelle Zahl in der y-Einheit des Kanals.

Es erfolgt eine Veränderung des Messwertes (y-Wert der Eingangsdaten) entsprechend der folgenden Formel:

$$Off = [Zufallszahl] * [„Uncertainty\ Source.\ Offset“]$$

$$[Neuer\ Wert] := [Alter\ Wert] + Off$$

Die Zufallszahl ist normalverteilt mit Standardabweichung 1.

Off wird einmalig für die gesamte Messung bestimmt.

Offset-Drift

Unter der Drift wird eine schleichende Offset-Änderung während der Messung verstanden.

"Uncertainty Source.Drift Offset": Maß für die Höhe der Drift, angegeben als reelle Zahl in der y-Einheit des Kanals.

"Uncertainty Source.Drift Time": Maß für die zeitliche Änderung des driftenden Offsets, angegeben als reelle Zahl in der x-Einheit des Kanals. Innerhalb dieser Zeit findet eine merkliche Änderung des Offsets statt. Die Angabe selbst ist eher qualitativ. Aber das Zeitverhal-

ten ändert sich proportional zu dieser Eigenschaft.

Es erfolgt eine Veränderung des Messwertes entsprechend der folgenden Formel:

[Neuer Wert] := [Alter Wert] + ["Uncertainty Source.Drift Offset"] * Driftwert

Der Driftwert ist eine Zufallszahl und weist eine Standardabweichung von ca. 1 auf.

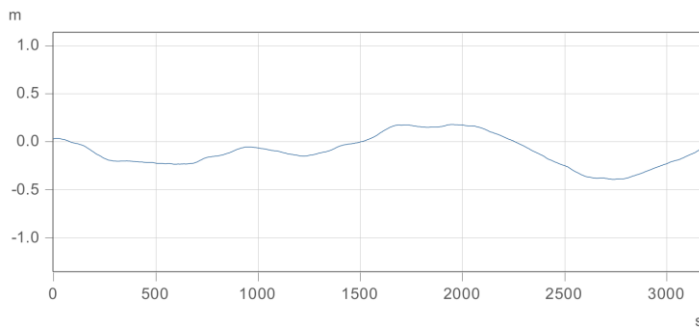


Abb. 9: Drift mit Offset 1m und Time = 100s

Brummen, Netzbrummen

Das Signal wird mit einem Netzbrummen beaufschlagt.

"Uncertainty Source.Hum Amplitude": Die Amplitude der Grundschwingung, angegeben als reelle Zahl in der y-Einheit des Kanals, größer Null. Die Amplitude ist während der gesamten Messung konstant. Eine Messung in diesem Sinn entspricht einem Durchlauf der Schleife UNCERTAINTY_LOOP. Über ganz viele Messungen gesehen ist die Amplitude normalverteilt. Die Standardabweichung dieser Normalverteilung wird hier als Eigenschaft angegeben. Die Phase der Grundschwingung variiert von Messung zu Messung und unterliegt einer Gleichverteilung.

"Uncertainty Source.Hum Frequency": Die Frequenz der Grundschwingung, angegeben im Kehrwert der x-Einheit des Kanals.

Die Frequenz ist über alle Messungen gleich. Z.B. 50Hz oder 60Hz.

"Uncertainty Source.Hum Harmonics": Verhältnis der Leistung aller Oberschwingungen zur Leistung der Grundschwingung. So

bedeutet z.B. ein Wert von 0.01, dass die Gesamtleistung aller Oberschwingungen 1% der Leistung der Grundschwingung beträgt. Ist diese Eigenschaft nicht gegeben, wird 0.1 angenommen.

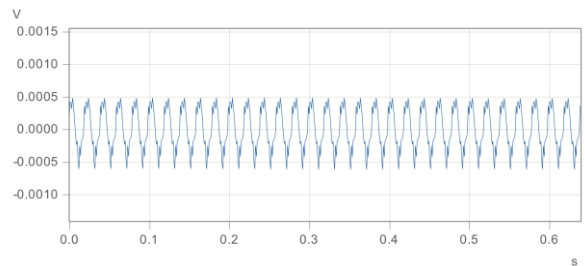


Abb. 9: Netzbrummen mit Parameter Amplitude = 0.001V, Frequenz = 50Hz, Harmonics=0.1

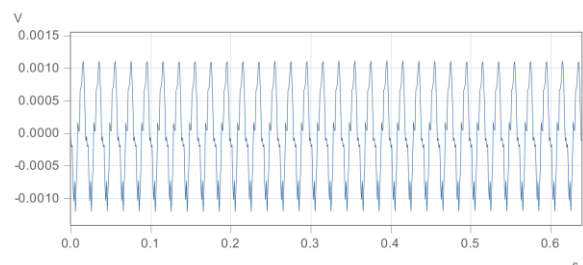


Abb. 10: Ein nächster Monte-Carlo Durchlauf bei gleichen Parametern

Störpulse, Spikes

Seltene Störpulse, Störspitzen, Spikes können dem Signal überlagert werden. Höhe und zeitliches Verhalten wird dabei definiert.

Die Höhe der Störpulse liegt zwischen einem Minimal- und einem Maximalwert. Jeder einzelne Störpuls und jeder einzelne Messpunkt eines Störpulses erhält einen individuellen zufälligen Wert. Die Werte sind gleichverteilt. Zwischen den einzelnen Störpulsen bleibt das Signal unverändert.

"Uncertainty Source.Spikes Max": Der Maximalwert der Amplitude des Störpulses, angegeben als reelle Zahl in der y-Einheit des Kanals.

"Uncertainty Source.Spikes Min": Der Minimalwert der Amplitude des Störpulses, angegeben als reelle Zahl in der y-Einheit des Kanals.

"Uncertainty Source.Spikes Width": Die Breite des Störpulses, angegeben als reelle Zahl in der x-Einheit des Kanals. Die maximale Breite ist angegeben. Die Breite selbst ist gleichverteilt und wird für jeden Puls zufällig ermittelt.

"Uncertainty Source.Spikes Time": Abstand der Störpulse, angegeben als reelle Zahl in der x-Einheit des Kanals. Gemessen wird von Beginn eines bis zum Beginn des nächsten Störpulses.

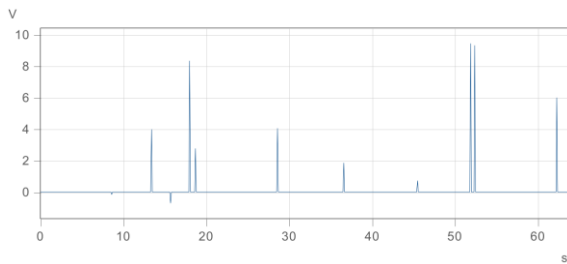


Abb. 11:
Spikes mit Min=-2V, Max=10V, Width=0.1s, Time=10s

Höherfrequentes Rauschen

Rauschen, das vor allem höhere Frequenzanteile enthält.

"Uncertainty Source.Noise RMS": Die y-Werte des Eingangskanals werden mit einem Rauschen beaufschlagt. Die Standardabweichung der Störung entspricht dem Wert der Eigenschaft.

"Uncertainty Source.Noise Frequency": Die untere Grenzfrequenz des Rauschens, angegeben im Kehrwert der x-Einheit des Kanals. Die untere Grenzfrequenz ist über alle Messungen gleich.

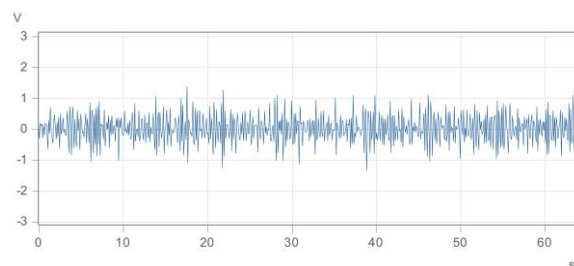


Abb. 12:
Rauschen oberhalb von 2Hz, RMS = 0.01V, Abtastfrequenz 10Hz

Testlauf

Es ist empfohlen, einen Testlauf durchzuführen, um die Form der Störung zu überprüfen. Die meisten Störungen lassen sich mit einem Signal, das aus [Eingangssignal * 0.0] gebildet wird, gut beurteilen. Im Fall des Verstärkungsfehlers ist es das [Eingangssignal * 0.0 + 1.0].

Beispiel: Spannungsspitzen

Ein gemessenes Spannungssignal enthält einige Spannungsspitzen. Für die weiterführende Berechnung soll die Auswirkung dieser Spitzen auf das Ergebnis der Analyse untersucht werden.

UncertaintySet(Spannung, "Uncertainty Source.Spikes Max",300) ; in [V]

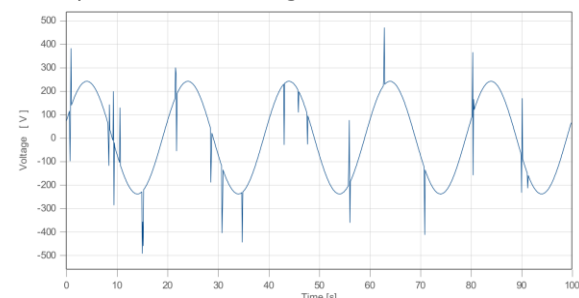
UncertaintySet(Spannung, "Uncertainty Source.Spikes Min",-300) ; in [V]

UncertaintySet(Spannung, "Uncertainty Source.Spikes Width",0.001) ; in [sec]

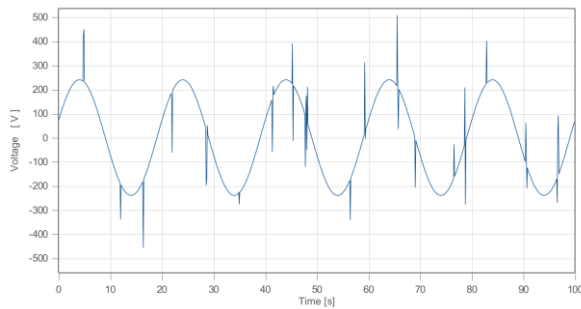
UncertaintySet(Spannung, "Uncertainty Source.Spikes Time",0.01) ; in [sec]

Eingangsdaten werden mit Spannungsspitzen beaufschlagt. Über die Funktion „Uncertainty-Set()“ werden die Maximal- und Minimalwerte der Spitzen festgelegt, sowie die Breite und die Häufigkeit.

Das Signal selbst ist sinusförmig und wird nun in den verschiedenen Monte-Carlo-Versuchen gemäß Vorgabe mit verschiedenen Störspitzen beaufschlagt.



Ein weiterer Durchlauf liefert:



Selbst definierte Störmuster

Es können auch komplett selbst definierte Störmuster erstellt werden. Die Notwendigkeit besteht z.B. dann, wenn eine Verteilungsdichte vorliegt, die über die anwenderdefinierten Eigenschaften nicht abgebildet wird. Z.B. ist das der Fall bei begrenzten Verteilungen, weil aus physikalischen Gründen Grenzen vorliegen. Z.B.

Masse ≥ 0 kg

Temperatur ≥ 0 K

Wirkungsgrad < 1 .

Das ist auch der Fall bei bestimmten Formen des Störsignals.

Die Beaufschlagung mit der Störung wird durchgeführt, indem mit der Funktion „random“ Pseudo-Zufallszahlen vorgegebener Verteilung erzeugt werden:

Beispiel:

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

Mit der letzten Zeile erfolgt die Begrenzung auf 0K für die Temperatur, unter der Annahme, dass der Kanal in K skaliert ist.

Ebenso ist die Überlagerung von automatischer und selbstdefinierter Störung möglich:

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

Sie müssen darauf achten, dass Sie stets nicht korrelierte Zufallszahlen benutzen. Das gilt für alle Kanäle innerhalb eines Schleifendurchlaufs wie auch zwischen den Schleifendurchläufen.

Fazit:

Mit imc FAMOS sind Sie in der Lage, alle am Eingangssignal beteiligten Störgrößen für die Weiterverrechnung abzubilden. Die Auswirkungen solcher Einflüsse auf einen mathematischen Algorithmus können untersucht werden. Die resultierende Messunsicherheit kann ermittelt werden.

Weitere Informationen erhalten Sie unter:

imc Test & Measurement GmbH

Voltastr. 5
D-13355 Berlin

Telefon: +49 (0)30-46 7090-0
Fax: +49 (0)30-46 31 576
E-Mail: hotline@imc-tm.de
Internet: <http://www.imc-tm.de>

Die imc Test & Measurement GmbH ist Hersteller und Lösungsanbieter von produktiven Mess- und Prüfsystemen für Forschung, Entwicklung, Service und Fertigung. Darüber hinaus konzipiert und produziert imc schlüsselfertige Elektromotorenprüfstände. Passgenaue Sensor- und Telemetriesysteme ergänzen unser Produktportfolio.

Unsere Anwender kommen aus den Bereichen Fahrzeugtechnik, Maschinenbau, Bahn, Luftfahrt und Energie. Sie nutzen die imc-Messgeräte, Softwarelösungen und Prüfstände, um Prototypen zu validieren, Produkte zu optimieren, Prozesse zu überwachen und Erkenntnisse aus Messdaten zu gewinnen. Rund um die imc Geräte steht dafür ein umfassendes Dienstleistungsspektrum zur Verfü-

gung, das von der Beratung bis zur kompletten Prüfstandsautomatisierung reicht. Auf diese Weise verfolgen wir konsequent das imc Leistungsversprechen „produktiv messen“.

National wie international unterstützen wir unsere Kunden und Anwender mit einem starken Kompetenz- und Vertriebsnetzwerk.

Wenn Sie mehr über die imc Produkte und Dienstleistungen in Ihrem Land erfahren wollen oder selbst Distributor werden möchten, finden Sie auf unserer Webseite alle Informationen zum imc Partnernetzwerk:

<http://www.imc-tm.de/partner/>



Nutzungshinweis:

Dieses Dokument ist urheberrechtlich geschützt. Alle Rechte sind vorbehalten. Dieser Bericht darf ohne Genehmigung weder bearbeitet, abgewandelt noch in anderer Weise verändert werden. Ausdrücklich gestattet ist das Veröffentlichung und Vervielfältigen des Dokuments. Bei Veröffentlichung bitten wir darum, dass der Name des Autors, des Unternehmens und eine Verlinkung zur Homepage www.imc-tm.de genannt werden. Trotz inhaltlicher sorgfältiger Ausarbeitung, kann dieser Bericht Fehler enthalten. Sollten Ihnen unzutreffende Informationen auffallen, bitten wir um einen entsprechenden Hinweis an: marketing@imc-tm.de. Eine Haftung für die Richtigkeit der Informationen wird grundsätzlich ausgeschlossen.