

Dynamische Messunsicherheit – Software und Leitfäden als Brücke zwischen Theorie und Praxis

Dr. Sascha Eichstädt¹, Dr. Clemens Elster¹, Dr. Ian Smith², Dr. Trevor J Esward¹
¹Physikalisch-Technische Bundesanstalt, Braunschweig und Berlin, Deutschland
²National Physical Laboratory, Teddington, Grossbritannien

Zusammenfassung

Die Analyse dynamischer Messungen stellt für viele Anwender eine große Herausforderung dar, was eine breite Anwendung vorhandener Kalibrierdienste und mathematischer Werkzeuge einschränkt. So erfordern dynamische Messungen stets die Anwendung von Methoden der digitalen Signalverarbeitung, System- und Kontrolltheorie sowie der multivariaten Statistik. Insbesondere das Design geeigneter digitaler Filter und die Bestimmung der Messunsicherheit der im Allgemeinen hochdimensionalen dynamischen Messgröße sind eine Herausforderung in der Praxis.

In einer Reihe von internationalen Forschungsprojekten wurden mathematische und messtechnische Verfahren zum Umgang mit dynamischen Messungen entwickelt und auf Ebene der nationalen Metrologieinstitute erfolgreich umgesetzt. Ziel des europäischen Projekts EMPIR 14SIP08 ist es, durch die Entwicklung von Leitfäden und Software diese Verfahren einer breiten Gruppe von Anwender verfügbar zu machen.

Dieser Beitrag stellt ausgewählte Methoden aus dem Bereich dynamischer Messungen vor, für welche im Rahmen dieses Projekts mathematische Methoden in einem Softwarepaket zusammengefasst und als Open-Source-Projekt veröffentlicht werden. Das Softwarepaket ist öffentlich zugänglich, für die nicht-kommerzielle Nutzung und Verbreitung freigegeben und transparent dokumentiert. Die Programmiersprache Python wurde gewählt, da sie einfach zugänglich ist und ohne großen Aufwand in vielen gängigen Softwareumgebungen, wie zum Beispiel MATLAB®, Excel® oder LabView®, eingebunden werden kann.

Keywords: dynamische Messungen, digitale Signalverarbeitung, Messunsicherheit, Software

Einleitung

Die Analyse von Messungen, bei denen mindestens eine der beteiligten Größen zeitabhängige Werte aufweist, spielen eine immer größere Rolle in der Kalibrierung und Anwendung. Dabei reichen dynamische Messungen von einzelnen Sensoren bis hin zu komplexen Netzwerken, wie zum Beispiel Strom- oder Gasnetzen, sie beinhalten eine große Bandbreite von physikalischen Größen und decken ebenso Messgrößen ab, die auf Zeitskalen von wenigen Pikosekunden als auch von mehreren Minuten variieren.

Beispiele von dynamischen Messungen finden sich etwa bei mechanischen Größen wie Beschleunigung [1], Kraft [2-3], Drehmoment [4] und Druck [5-6], die häufig unter dynamischen Bedingungen durchgeführt werden [7]. Ebenso finden sich dynamische Messungen bei der Charakterisierung von elektrischen Größen [8-11] oder Ultraschallgeräten [12].

Trotz ihrer großen Verbreitung besteht ein akuter Mangel an einheitlichen Regelwerken und Standards zum Umgang, der Durchführung und der Analyse dynamischer Messungen. Im Gegensatz dazu gibt es im Bereich der statischen Messungen eine große Auswahl an etablierten Standards, Leitfäden und ausgereifter Software. Insbesondere im Hinblick auf die Fortpflanzung von Messunsicherheiten hat sich zum Beispiel der „Guide to the Expression of Uncertainty in Measurements“ (GUM) [13] weitestgehend als Quasi-Standard durchgesetzt. In Zuge dessen wurden eine Reihe von Softwarepaketen verschiedener Komplexität entwickelt, die auf vielen Anwendungsebenen eine einfache Umsetzung der Methodik des GUM ermöglichen.

Die Situation für statische Messungen kann daher als sehr gut bezeichnet werden, während im Bereich dynamischer Messungen noch immer viele grundlegende Fragen unbeantwortet sind [14].

In dem kürzlich abgeschlossenen europäischen Forschungsprojekt IND09 „*Traceable dynamic measurement of mechanical quantities*“ [15] wurden für die Messgrößen Kraft, Drehmoment und Druck Verfahren zur dynamischen Primärkalibrierung von Sensoren und entsprechende Methoden zur Fortpflanzung von Messunsicherheiten entwickelt. Damit ist für diese Messgrößen im Bereich der nationalen Metrologieinstitute (NMI) das Fundament für eine Behandlung dynamischer Messungen gelegt.

In den letzten Jahren wurden außerdem verschiedene Verfahren für die Analyse dynamischer Messungen entwickelt, wie zum Beispiel on-line Verfahren für die Berechnung der Messunsicherheit bei der Anwendung digitaler Filter [16-17] effiziente Monte-Carlo Methoden für dynamische Messungen [18], sowie Entfaltungsmethoden für Modelle im Frequenzbereich [9].

Bisher sind diese mathematischen Methoden und die entwickelten Kalibrierverfahren noch nicht ausreichend in entsprechende Leitfäden, Standards oder allgemein nutzbare Software überführt worden, welche auf industrielle Aufgabenstellungen anwendbar sind. Zu diesem Zweck verfolgen die Physikalisch-Technische Bundesanstalt (PTB) und das National Physical Laboratory (NPL) im 2015 gestarteten EMPIR Projekt 14SIP08 [19] das Ziel, Anleitungen und Software zu entwickeln, welche die Anwendung vorhandener Kalibrierverfahren und mathematischer Methoden außerhalb der NMIs vereinfachen sollen. Die Zusammenarbeit mit Partnern aus der Industrie sichert dabei die angestrebte Eignung der Leitfäden und Software in für die Industrie relevanten Anwendungssituationen.

Während Leitfäden zitierbare Referenzen und Beispiele bereitstellen, welche die Entwicklung von anwendungsspezifischen Standards unterstützen, fördert frei verfügbare Software die Anwendung und Verbreitung komplexer mathematischer Methoden. Die Bereitstellung von Software als Implementierung aktueller mathematischer und statistischer Verfahren gehört seit langem für viele NMIs zur selbstverständlichen Praxis. So bieten das NPL und die PTB inzwischen für viele Anwendungsbereiche entsprechende Software zur kostenfreien Nutzung an.

Dieser Beitrag stellt das Softwarepaket *PyDynamic* vor, welches in dem Projekt 14SIP08 entwickelt wird. Ziel dieses Pakets ist es, die vom NPL und der PTB entwickelten Methoden zur Analyse dynamischer Messungen in einer umfassenden Bibliothek von Programmen zusammenzufassen und mit Beispielen zu dokumentieren. Die Programmiersprache Py-

thon wurde gewählt, da sie einfach zugänglich ist und ohne großen Aufwand in vielen gängigen Softwareumgebungen, wie zum Beispiel MATLAB®, EXCEL® oder LabView®, eingebunden werden kann. Zur besseren Dokumentation wird das Softwarepaket auf der Open-Source-Plattform GitHub bereitgestellt¹. Dort kann es entweder kostenfrei direkt heruntergeladen oder in lokalen Versionsverwaltungssystemen eingebunden werden.

Im Folgenden werden die Struktur des Softwarepakets und ausgewählte Methoden näher erläutert.

Allgemeiner Aufbau von *PyDynamic*

Das Softwarepaket *PyDynamic* ist in folgende Unterpakete aufgeteilt:

- *identification*,
- *deconvolution*,
- *uncertainty*,
- *misc*,

welche jeweils speziellen Module und Methoden beinhalten.

Das Paket *identification* stellt Methoden zum Entwurf eines digitalen Filters oder einer Übertragungsfunktion ausgehend von einer gemessenen Frequenzantwort bereit [20]. Die Methoden beinhalten dabei auch explizit die Möglichkeit, die Unsicherheiten der Werte der Frequenzantwort auf die Koeffizienten des Filters oder der Übertragungsfunktion zu propagieren. Das Paket *deconvolution* verwendet ähnliche Methoden wie das Paket *identification*, um für eine gegebene Frequenzantwort ein digitales Filter zur Kompensation des zugehörigen dynamischen Systems zu entwerfen. Auch hier spielt die Fortpflanzung von Unsicherheiten eine große Rolle.

Das Paket *uncertainty* beinhaltet Methoden zur Berechnung von Messunsicherheiten bei der Anwendung digitaler Filter mit unsicheren Koeffizienten. Dabei werden on-line taugliche Verfahren zur Anwendung des GUM als auch effiziente Implementierungen der Monte-Carlo Methode des GUM Supplement 1 [21] bereitgestellt [16-18].

Das Paket *misc* besteht aus verschiedenen Hilfsmodulen zur Erzeugung von Testsignalen und der Bereitstellung von allgemeinen Hilfsmethoden.

In der Software enthalten ist außerdem ein Paket von Beispielen, welche für anwendungsnahe Situationen die Verwendung der Methoden dokumentieren. Im Folgenden werden ausgewählte Beispiele aus diesem Paket

¹ <https://github.com/eichstaedtPTB/PyDynamic>

näher erläutert. Der hier verwendete Stand der Software ist dabei der vom 20. Februar 2016.

Beispiel: Entwurf eines digitalen Kompensationsfilters

In diesem Abschnitt betrachten wir die Bestimmung des Wertes der Messgröße, wenn der Zusammenhang zwischen Beobachtung und Messgröße durch eine Faltung modelliert werden kann

$$y(t) = (h * x)(t) \quad (1)$$

wobei x das Eingangssignal (die Messgröße), y das Ausgangssignal und h die Impulsantwort des als linear und zeitinvariant angenommenen Messsystems bezeichnet.

Für die Schätzung des Eingangssignals aus der Kenntnis des Ausgangs und des Messsystems soll hier ein digitales Filter konstruiert werden. In Abhängigkeit von der Art der Kenntnis über das Messsystem können dafür verschiedene Methoden angewendet werden. Eine Übersicht und Einführung dazu ist in [22] gegeben.

Die grundsätzliche Herangehensweise besteht in dem Fit eines digitalen Filters an das Reziproke der Frequenzantwort des Messsystems. Eine Herausforderung stellt dabei die Stabilität des Filters dar, da der Filterentwurf für ein eigentlich unphysikalisches System erfolgt. Typischerweise wird die Stabilität des Filters durch die Einführung einer kleinen Zeitverzögerung erreicht [22].

Hier betrachten wir den Entwurf eines Filters mit endlicher Impulsantwortlänge (FIR) mit Hilfe der *PyDynamic* Routine *LSFIR_unc* aus dem Paket *deconvolution*. Als Eingangsvariablen für die Routine wird ein Array mit den Werten der Frequenzantwort als Real- und Imaginärteil benötigt

$$\mathbf{H} = (\Re_0, \dots, \Re_N, \Im_0, \dots, \Im_N)^T \quad (2)$$

zusammen mit der beigeordneten Messunsicherheit U_H . Zu gegebener Filterordnung M und Zeitverzögerung τ bestimmt die Routine dann ein FIR Filter G mit Filterkoeffizienten \mathbf{b} als kleinste-Quadrate-Anpassung an das Reziproke der Frequenzantwort, so dass

$$G(e^{j\omega F_s}) \approx H^{-1}(j\omega)e^{-j\omega\tau} \quad (3)$$

für einen gewählten Frequenzbereich und Abtastrate F_s , siehe Abb. 1.

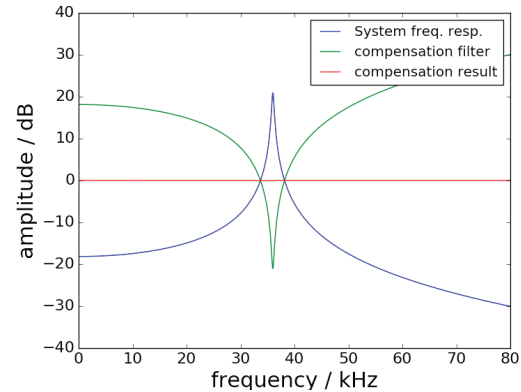


Abb. 1: Betrag der komplexwertigen Frequenzantwort des Messsystems (blau), des angepassten Kompensationsfilters (grün) und dem Produkt (rot).

Der Aufruf der *PyDynamic* Routine mit

$$\mathbf{b}, U_b = \text{FIR_unc}(H, U_H, M, \tau, f, F_s)$$

liefert dann die Filterkoeffizienten \mathbf{b} und die beigeordnete Kovarianzmatrix U_b , siehe Abb. 2. Die Unsicherheit, welche der Kenntnis des Messsystems beigeordnet ist, überträgt sich auf eine Unsicherheit der gefitteten Filterkoeffizienten. Die Fortpflanzung der Unsicherheit kann dabei entweder mit Hilfe der Monte-Carlo Methode (GUM Supplement 1) oder durch Anwendung der Gauß-Formel (GUM) ausgeführt werden [16]. Beide Methoden stehen in *PyDynamic* zur Verfügung.

Da es sich bei der Kompensation mathematisch um ein schlecht gestelltes inverses Problem handelt, muss eine Regularisierung angewendet werden. Eine Vielzahl von Regularisierungsmethoden existieren in der Literatur mit anwendungsspezifischen Ansätzen zu ihrer Umsetzung. Ohne Regularisierung ergibt sich bei der Kompensation linearer dynamischer Systeme eine (fast beliebig große) Rauschverstärkung. Daher ist ein häufig verwendeter Regularisierungsansatz der Entwurf eines geeigneten Tiefpassfilters, welches zusätzlich zu dem Kompensationsfilter angewendet wird [23]. Dabei sollte das Tiefpassfilter möglichst so entworfen werden, dass der entstehende systematische Schätzfehler für die Messgröße vernachlässigbar klein im Vergleich zu den übrigen Unsicherheitsfaktoren ist. Andererseits muss Vorwissen über die Messgröße verwendet werden, um den Unsicherheitsbeitrag des Tiefpassfilters zu berücksichtigen [24].

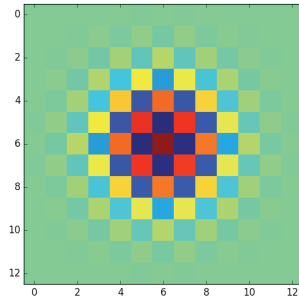


Abb. 2: Illustration der Kovarianzmatrix U_b , welche dem Vektor der gefitteten Filterkoeffizienten b beigeordnet ist.

Beispiel: Unsicherheitsfortpflanzung bei der Anwendung eines FIR Filters

Die Anwendung eines FIR Filters mit Koeffizientenvektor b auf eine Sequenz von Signalwerten y ist gegeben als

$$x[n] = \sum_{k=0}^{M-1} b_k y[n-k] \quad (4)$$

Für die Anwendung des angepassten FIR Filters aus dem vorherigen Abschnitt ist eine geeignete Methode zur Fortpflanzung der Messunsicherheit notwendig. Dafür stellt im *PyDynamic*-Paket *uncertainty* das Modul *propagate_FIR* eine Implementierung der Gleichung [16]

$$u^2(\hat{x}[n-\tau]) = b^T U_{y_{low}[n]} b + y_{low}[n] U_b y_{low}[n] + Tr(U_{y_{low}[n]} U_b) \quad (5)$$

bereit, wobei y_{low} das Ergebnis der Anwendung eines optionalen zusätzlichen FIR-Tiefpassfilters auf das beobachtete Ausgangssignal y und $Tr()$ die Summe der Diagonaleinträge bezeichnet, siehe [16]. Zur Illustration wird hier das FIR-Kompensationsfilter aus dem vorherigen Abschnitt verwendet, und für das Ausgangssignal y aus Gl. (1) wird ein weißes Rauschen mit bekannter Standardabweichung angenommen.

Der Aufruf der *PyDynamic* Routine mit

$$xh, Uxh = FIRuncFilter(y, noise, b, Ub, shift, blow)$$

liefert dann für gegebenes Signal y mit Rauschstandardabweichung $noise$, Zeitverzögerung $shift$ und FIR Tiefpassfilterkoeffizienten $blow$ das Ergebnis der Faltung als

$$\hat{x}[n-d] = ((y * b_{low}) * b)[n], \quad (6)$$

mit shift d sowie die beigeordnete (punktweise) Messunsicherheit u_x berechnet nach Gleichung (5). Das Ergebnis für das Beispiel aus dem vorherigen Abschnitt ist in Abb. 3 und Abb. 4 dargestellt.

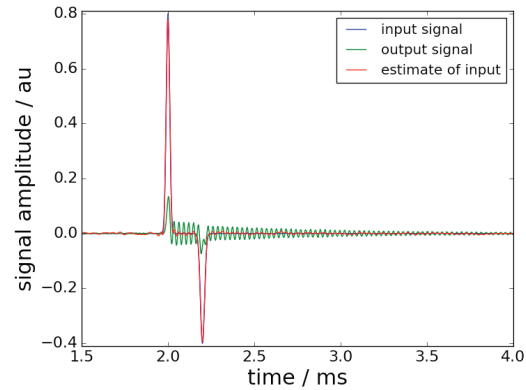


Abb.3 Simulierte Messgröße (blau), Ausgangssignal des Messsystems (grün) und Ergebnis der Entfaltung (rot).

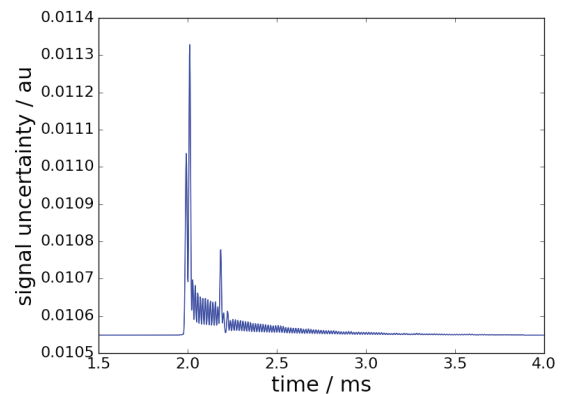


Abb. 4 Zeitabhängige Messunsicherheit, welche dem Ergebnis der Entfaltung beigeordnet ist.

Der Schätzwert ist in diesem Beispiel bis auf den Spitzenwert nahezu deckungsgleich mit dem simulierten Eingangssignal des Messsystems. Die beigeordnete Messunsicherheit zeigt eine deutliche Zeitabhängigkeit, welche maßgeblich durch den Einfluss der unsicheren Filterkoeffizienten verursacht wird.

Beispiel: Unsicherheitsfortpflanzung bei der Anwendung eines IIR Filters

Die Anwendung eines Filters mit unendlich langer Impulsantwort (IIR) auf eine Signalsequenz y ist gegeben als

$$x[n] = \sum_{k=0}^{M_b} b_k y[n-k] - \sum_{k=1}^{M_a} a_k x[n-k] \quad (5)$$

Durch die rekursive Form kann die resultierende Unsicherheit des Filterausgangssignals x nicht mehr einfach analytisch ausgedrückt werden. Darüber hinaus ist Gl. (5) nichtlinear in den Filterkoeffizienten und den Signalsequenzen.

In [17] wurde daher eine Transformation von Gl. (5) in eine Zustandsraumdarstellung verwendet:

$$\begin{aligned} z[n+1] &= Az[n] + by[n] \\ x[n] &= c^T z[n] + b_0 y[n] \end{aligned} \quad (6)$$

wobei

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & 1 \\ -a_{M_a} & -a_{M_a-1} & -a_{M_a-2} & \dots & -a_1 \end{pmatrix}$$

Diese Darstellung hat den Vorteil, dass die rekursive Berechnung von $x[n]$ basierend auf den vorhergehenden M_a Schätzwerten der Signalsequenz x auf eine rekursive Schätzung nur basierend auf dem letzten Schätzwert verringert wird. Das „Gedächtnis“ des Filters (5) wird dabei in einen Zustandsvektor der Dimension M_a übertragen.

Darauf aufbauend ist in [17] eine GUM-konforme Linearisierung der rekursiven Zustandsraumgleichung angegeben, welche es erlaubt, die (punktweise) Messunsicherheit on-line zu bestimmen. Diese Methode ist im *PyDynamic*-Paket *uncertainty* im Modul *propagate_IIR* implementiert und wird für eine Sequenz y mit weißem Rauschen bekannter Varianz aufgerufen durch

$$x, ux = IIR_uncFilter(y, noise, b, a, Uab)$$

wobei *noise* die Standardabweichung des Rauschens, b und a die Filterkoeffizienten mit beigeordneter Messunsicherheit Uab bezeichnen.

Da jedoch auch das mathematische Modell (6) nichtlinear in den Eingangsgrößen ist, ist häufig die Anwendung der Monte Carlo Methode aus GUM Supplement 2 [25] zu empfehlen. Dafür wird die Kenntnis der Eingangsgrößen in einer gemeinsamen multivariaten Wahrscheinlichkeitsdichtefunktion (PDF) ausgedrückt und mit Hilfe der Monte Carlo Methode durch das mathematische Modell (5) oder (6) propa-

giert. Für die Monte Carlo Methode werden Samples aus der PDF der Eingangsgrößen gezogen und für jedes Sample das Modell ausgewertet. Das Ergebnis ist dann eine Menge von Samples der PDF, welche der Messgröße beigeordnet werden kann. Aus diesen Samples werden der Schätzwert als Mittelwert und die Unsicherheit als Kovarianz berechnet.

In der Regel ist jedoch die Länge der Signalsequenz y zu groß, um eine direkte Anwendung der Monte Carlo mit ausreichender Anzahl von Samples durchführen zu können. Zu diesem Zweck wurde in [18] basierend auf Gl. (6) eine effizientere Implementierung der Monte Carlo Methode für dynamische Messungen entwickelt. Diese erlaubt es eine sehr große Genauigkeit für die Monte Carlo Methode auch auf Standard-Computern zu erhalten. Im *PyDynamic*-Paket *uncertainty* stellt das Modul *propagate_MonteCarlo* dafür die Routine *SMC* bereit, welche mittels

$$x, ux = SMC(y, noise, b, a, Uba)$$

den Schätzwert x und seine (punktweise) Unsicherheit ux bereitstellt. Optional werden zusätzlich auch punktweise Vertrauensintervalle bestimmt.

Zur Illustration verwenden wir die Anwendung eines IIR Filters mit unsicheren Koeffizienten auf ein Rechtecksignal, siehe Abb. 5. Bei dem Filter handelt es sich um ein Butterworth Tiefpassfilter mit einer Unsicherheit bezüglich der Wahl der Abschnidefrequenz. Aus dieser Unsicherheit über die Designparameter des Filters ergibt sich eine Unsicherheit der Filterkoeffizienten, welche mit Hilfe der Monte Carlo Methode berechnet wurde. Die Unsicherheit, welche dem Ausgangssignal x des Filters zugeordnet werden kann, wenn für das Eingangssignal ein weißes Rauschen mit bekannter Standardabweichung angenommen wird, wurde zum einen mit der Linearisierungsmethode und zum Vergleich mit der effizienten Monte Carlo Implementierung bestimmt, siehe Abb. 6.

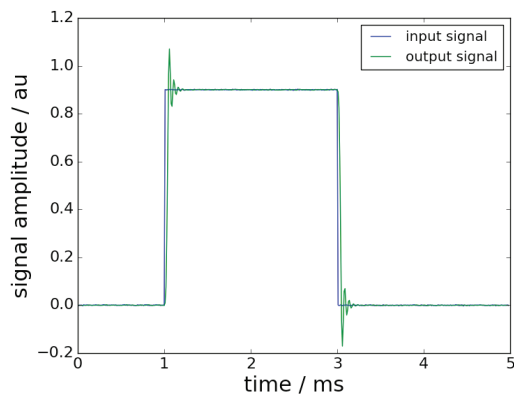


Abb. 5 Ein- und Ausgangssignal des IIR Filters

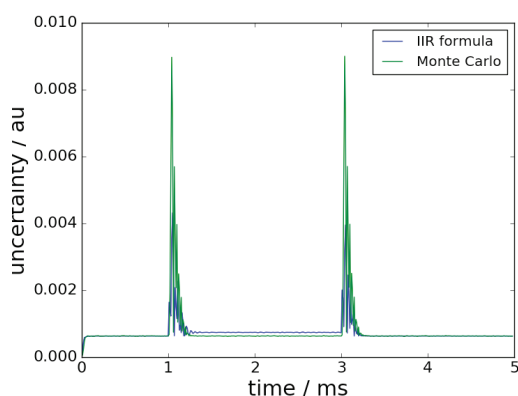


Abb. 6 Punktweise Unsicherheit des Ausgangssignals des IIR-Filter

Der Unterschied zwischen dem Ergebnis der Linearisierungsmethode („IIR formula“) und der Monte Carlo Methode begründet sich durch die Nichtlinearität des Modells.

Zusammenfassung und Ausblick

Das gemeinsame EMPIR Projekt 14SIP08 *Dynamic* des NPL und der PTB hat zum Ziel, von NMIs entwickelte Methoden und mathematische Werkzeuge im Bereich der Analyse dynamischer Messungen für industrielle Aufgabenstellungen anwendbar zu machen. Dazu werden einführende Veröffentlichungen, ein Leitfaden erarbeitet und das freie quelloffene Softwarepaket *PyDynamic* entwickelt.

Die hier gezeigten Anwendungsbeispiele für die Software illustrieren die zurzeit implementierten mathematischen Werkzeuge zum Entwurf digitaler Kompensationsfilter und der Fortpflanzung von Messunsicherheiten bei deren Anwendung. Durch die Bereitstellung von Software werden die mathematischen Methoden einer breiten Gruppe von Anwendern zugänglich ge-

macht. Im Laufe des Projekts sollen diese Methoden in ihrem Umfang erweitert und um weitere Methoden ergänzt werden. Beispielsweise werden Methoden zur Fortpflanzung der Messunsicherheit bei der Anwendung der diskreten Fouriertransformation (DFT) und der Beurteilung der systematischen Abweichung der Regularisierung entwickelt. Die zu erarbeitenden Dokumente sollen dann die Anwendung der Software in industriellen Aufgabenstellungen illustrieren.

Danksagung

Teile dieser Arbeiten wurden innerhalb des „European Metrology Programme for Innovation and Research“ (EMPIR) Projekts 14SIP08 durchgeführt. Die EMPIR-Initiative wird kofinanziert von dem Europäischen Union Programm „2020 Research and Innovation“ und den an EMPIR teilnehmenden Staaten.

Literaturnachweis

- [1] A. Link, W. Wabinski, and H. J. von Martens, *tm - Technisches Messen*, 72 (3), 153–160 (2005); doi: 10.1524/teme.72.3.153.60277
- [2] C. Schlegel, G. Kieckenap, B. Glöckner, A. Buß, and R. Kümme, *Metrologia*, 49 (3), 224–235 (2012); doi: 10.1088/0026-1394/49/3/224
- [3] M. Kobusch, S. Eichstädt, L. Klaus, and T. Bruns, *ACTA IMEKO* 4 (2), (2015)
- [4] L. Klaus, B. Arendacká, M. Kobusch, and T. Bruns, *ACTA IMEKO*, 4 (2) 39–44 (2015)
- [5] R. M. Gardner, *Anesthesiology* 54 (3) 227–236 (1981); doi: 10.1097/0000542-198103000-00010
- [6] C. Matthews, F. Pennecci, S. Eichstädt, A. Malengo, T. Esward, I. Smith, C. Elster, A. Knott, F. Arrhen, and A. Lakka, *Metrologia*, 51 (3) 326–338 (2014); doi: 10.1088/0026-1394/51/3/326
- [7] A. Schäfer, *PTB-Mitteilungen Traceable Dynamic Measurement of Mechanical Quantities*, no. 9, (2015)
- [8] N. H. Younan, A. B. Kopp, and D. B. Miller, *IEEE Transactions on Power Delivery*, 6 (2) 501–506 (1991); doi: 10.1109/61.131103
- [9] P. D. Hale, A. Dienstfrey, J. Wang, D. F. Williams, A. Lewandowski, D. A. Keenan, and T. S. Clement, *IEEE Trans. Instrum. Meas.*, 58 (10) 3554–3568 (2009); doi: 10.1109/TIM.2009.2018012
- [10] P. D. Hale, D. F. Williams, A. Dienstfrey, J. Wang, J. Jargon, D. Humphreys, M. Harper, H. Fuser, and M. Bieler, Conference on Precision Electromagnetic Measurements (CPEM), 2012
- [11] D. A. Humphreys, P. M. Harris, M. Rodriguez-Higuero, F. A. Mubarak, D. Zhao, and K. Ojasa-

- Io, *TIM*, 64 (2) 356–365, (2015); doi: 10.1109/TIM.2014.2340640
- [12] V. Wilkens and C. Koch, *J. Acoust. Soc. Am.*, 115 (6) 2892–12 (2004); doi: 10.1121/1.1707087
- [13] BIPM, IFCC, IUPAC, and ISO, *Evaluation of measurement data—guide for the expression of uncertainty in measurement. JCGM 100: 2008*
- [14] S. Eichstädt, V. Wilkens, A. Dienstfrey, P. D. Hale, B. Hughes, and C. Jarvis, submitted to *Metrologia* (2016)
- [15] “IND09 Traceable Dynamic Measurement of Mechanical Quantities,” *IND09*. Available: www.ptb.de/emrp/ind09.html. [Accessed: 05-Feb-2016].
- [16] C. Elster and A. Link, *Metrologia*, 45 (4) 464–473 (2008); doi: 10.1088/0026-1394/45/4/013
- [17] A. Link and C. Elster, *Meas. Sci. Technol.* 20(5) 1–6, (2009); doi: 10.1088/0957-0233/20/5/055104
- [18] S. Eichstädt, A. Link, P. Harris, and C. Elster, *Metrologia*, 49 (3) 401–410 (2012); doi: 10.1088/0026-1394/49/3/401
- [19] EMPIR 14SIP08 “Standards and software to maximise end user uptake of NMI calibrations of dynamic force, torque and pressure sensors Available: <http://mathmet.org/projects/14SIP08/>. [Accessed: 05-Feb-2016].
- [20] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-time signal processing*. 1989.
- [21] BIPM, IFCC, and IUPAC, OIML, *Evaluation of measurement data—Supplement 1 to the GUM: propagation of distributions using a Monte Carlo method*. JCGM 101:2008.
- [22] S. Eichstädt, C. Elster, T. J. Esward, and J. P. Hessling, *Metrologia* 47(5) 522-533 (2010); doi: 10.1088/0026-1394/47/5/003
- [23] S. Eichstädt, “Analysis of Dynamic Measurements,” PhD Thesis, TU Berlin 2012.
- [24] S. Eichstädt, A. Link, T. Bruns, and C. Elster, *Measurement*, 43 (5) 708–713 (2010); doi: 10.1016/j.measurement.2009.12.028
- [25] BIPM, IFCC, and IUPAC, OIML. *Evaluation of measurement data Supplement 2 to the “Guide to the expression of uncertainty in measurement”—Extension to any number of output quantities* JCGM 102:2011.