Consensus-based Online Co-Calibration for Networks of Homogeneous Sensors in IIoT Environments under Consideration of Semantic Knowledge

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von
Maximilian Gruber

Berlin 2023

First Reviewer :	Prof. Dr. Adrian Paschke
Second Reviewer :	Prof. Dr. Markus Bär

Date of Submission :	29th August 2023
Date of Disputation :	1st March 2024

Selbstständigkeitserklärung

Name:	Gruber
Vorname:	Maximilian

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.

Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

Berlin, den 29. August 2023

....

Abstract

Large scale sensor networks form an important part of the Industrial Internet of Things. To maintain the operation of such networks over time, quality of the sensor readings needs to be ensured. This leads to the development of a metrological traceable in-situ calibration method based on a Bayesian framework which leverages local sensor redundancy. Furthermore, automation of such in-situ calibration tasks is a key feature. To this end, an extension of existing sensor-related ontologies is proposed to cover relevant metrological terms. Sensor self-descriptions based on these knowledge representations allow for support of in-situ calibration by finding suitable reference sensors and initialization the mathematical method presented here. The mathematical method is evaluated in simulation studies against a state of the art in-situ calibration. The evaluation results show good estimation performance in cases of time-depending input signals or sensors of comparable uncertainty levels, but also reveal higher computational costs. The developed ontologies are evaluated by a corpus comparison, ontology metrics as well as logical checks of the taxonomic backbone and indicate a good agreement with existing ontology quality standards.

Acknowledgments

There are some people I want to acknowledge in this thesis for their support, guidance, work and advice.

First off, I am deeply indebted to Prof. Dr. Adrian Paschke for giving me the opportunity to work on the topic of this thesis, the continuous support and the fruitful discussions on semantics and other equally exiting topics.

I could not have undertaken this journey without Sascha Eichstädt and Clemens Elster who not only opened up this current and interesting research opportunity, but were also always available to provide valuable feedback throughout my work as well as share their great technical knowledge with me.

Further I would like to thank Sascha for letting me develop and follow my own ideas while still providing guidance and always being available for any questions. I am grateful for the great work atmosphere which allowed for high detailed technical conversation as well as equally enjoyable non-technical lunch breaks and coffee chats. On this note, my great thanks also goes to Tanja, Björn and Anupam for the engaging discussions, collaborations and general helpfulness.

Additionally, I would like to thank the other current and former group members Julia, Maik, Julia, Cliff, Daniel, Anke, Jens, Lara, Cathi, Barbara, Alex, Max, Samuel, Antje, Elisabeth, Maxim, Alex, Wafa, Jasper, Helga, Maher, Paul and Katy as well as all the colleges at the PTB, especially Benedikt and Joachim, who supported me with discussions, feedback, encouragement and great ideas.

I'm extremely grateful to my wife, parents, parents-in-law, my grandfather and extended family for supporting me throughout my journey.

Contents

List of Figures

List of Tables

List of Code Listings

Nomenclature

Notation

Symbol	Description
$\underline{\phi}$	underlined symbol denotes vector
$\mathbf{\Phi}$	bold capital symbol denotes matrix
ϕ^T	transpose of ϕ
$\hat{\phi}$	estimate of ϕ
$\boldsymbol{U}_{\underline{\phi}}$	covariance matrix of $\underline{\phi}$
u_ϕ	uncertainty of ϕ
$\phi(t)$	ϕ at time t
$\phi[k]$	ϕ at discrete time k
ϕ_i	i-th element of $\underline{\phi}$
$\phi_{(i)}$	all, except i-th element of $\underline{\phi}$

Operators

Symbol	Name	Read As
$\forall$	universal quantifier	for all
$\exists$	existence quantifier	there exists
$\mathbb{E}(\bullet)$	expectancy	
$\mathbb{V}(\bullet)$	covariance	
$u(\bullet)$	uncertainty	
$\Gamma(\bullet)$	Gamma function	
$\lvert\bullet\rvert$	absolute value	
$\lVert\bullet\rVert$	Euclidean norm (p-2-norm)	
$\bullet^{\mathcal{I}}$	interpretation	
$I_\bullet$	identity matrix of size $\bullet$	
$p(\bullet\vert\circ)$	conditional probability	probability of $\bullet$ given $\circ$
$\mathcal{N}(\bullet,\circ)$	Gaussian distribution	. . . with mean $\bullet$ and covariance $\circ$
$\bullet = \circ$	equality	is equal to
$\bullet \approx \circ$	approximation	is approximately
$\bullet \propto \circ$	proportionality	is proportional to
$\bullet \sim \circ$	distribution	distributed according to

$\bullet \land \circ$	logical and	
$\bullet \lor \circ$	logical or	
$\bullet \in \circ$	set membership	is in
$\bullet \notin \circ$	negation of set membership	is not in
$\bullet \subseteq \circ$	subset	is part of
$\bullet \cap \circ$	set intersection	overlapped with
$\bullet \cup \circ$	set union	joined with
$\bullet \setminus \circ$	set difference	without
$\{\bullet \mid \circ\}$	set expression	all elements $\bullet$ fulfilling $\circ$

Acronyms, Abbreviations and Prefixes

Short form	**Description**
AMD	Advanced Micro Devices (CPU manufacturer)
CGS	Centimetre–Gram–Second System of Units
CPU	Central Processing Unit
CQ	Competency Question
D-SI	Digital-SI
DCC	Digital Calibration Certificate
DGSM	De Gruyter Series in Measurement Sciences
DIN	Deutsches Institut für Normung
DL	Description Logic
DUT	Device under Test
EIV	Error in Variables
FU Berlin	Freie Universität Berlin
GUM	Guide to the expression of uncertainty in measurement
ID	Identity
IIoT	Industrial Internet of Things
IIR	Infinite Impulse Response
IoT	Internet of Things
IQR	Interquartile Range
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LTI	Linear Time-Invariant
MAP	maximum a-posteriori estimate
MathML	Mathematical Markup Language
MCM	Monte-Carlo-Method
MCMC	Markov-Chain-Monte-Carlo
MSD	Mean Signed Difference
MSE	Mean Squared Error
NMAE	Normalized Mean Absolute Error
NMI	National Metrology Institute
NMSE	Normalized Mean Squared Error
NTP	Network Time Protocol

OGC	Open Geospatial Consortium
OM	Ontology of Units of Measure and Related Concept
`om:`	prefix used for OM
OWL	Web Ontology Language
PDF	probability density function
PDF (math)	probability density function
PDF (doc)	portable document format
PDF/A (doc)	PDF specialized for archiving
PTB	Physikalisch-Technische Bundesanstalt
QUDT	Ontology for Quantities, Units, Dimensions and Types
`qudt:`	prefix used for QUDT
RDF	Resource Description Framework
`rdf:`	prefix used for RDF
`rdfs:`	prefix used for RDF Schema
SI	Système international d'unités (International System of Units)
SOSA	Sensors, Observations, Samples and Actuators Ontology
`sosa:`	prefix used for SOSA
SPARQL	SPARQL Protocol and RDF Query Language
SSN	Semantic Sensor Network Ontology
`ssn:`	prefix used for SSN
STD	standard deviation
UI	User Interface
URI	Universal Resource Identifier
URL	Universal Resource Locator
VIM	International vocabulary of metrology
W3C	World Wide Web Consortium
WGS84	World Geodetic System (version from 1984)
XML	Extensible Markup Language

Part I

Introduction

1.1 Thesis Focus

This thesis focuses on the description of homogeneous networks of calibrated sensors and the application of metrological methods therein. This includes development of an online co-calibration method for linear affine sensor transfer behavior using a Bayesian framework as well as proposing a merge of knowledge representations to describe metrological use cases in such sensor networks with semantic expressiveness. Furthermore, it is shown how the developed mathematical method benefits from the use of semantic knowledge.

The following sections 1.1.1 to 1.1.3 introduce the context of this thesis, the problems it solves and what scientific questions are guiding it. Section 1.1.4 motivates the structure of the remaining document.

1.1.1 Context

Technical, industrial and natural processes are monitored using sensors. To ensure the quality of their measurements, these sensors need to be calibrated against reference devices that trace back to international standards provided by national metrology institutes (NMIs) [1]. Current developments, accelerated by the idea of an (industrial) internet of things ((I)IoT), lead to dense sensor networks with an increasing amount of sensors that require calibration on a regular basis [2].

As of now, in order to calibrate[1] a sensor, it is taken out of the process, send to an accredited calibration laboratory and is brought back into the process after the calibration [4, 5]. It is of interest to shorten this process by using surrounding sensors as references to *co-calibrate* the sensor in place. Moreover, available semantic knowledge about the sensors in the network (e.g., self-descriptions provided by smart sensors) needs to support the automated selection of suitable reference sensors and initialization of the method. The combination of semantics and mathematics paves the way for machine-actionable co-calibration and automated use in dense sensor networks, adding a new perspective to the inclusion of expert knowledge into a calibration process.

Sensor networks typically generate a stream of time-series data for each sensor. The internal clocks of (the network interfaces of) the sensors can be assumed to be synchronized to a common time-base, e.g., via the network time protocol (NTP). However, the data acquisition is in general not synchronized, leading to an uncertainty-aware interpolation task. Moreover, it is not guaranteed,

[1] in an NMI-sense, which is defined in the International Vocabulary of Metrology [3]

that all sensors of interest provide data within the period of interest. To provide traceable calibration, the proposed method therefore needs to operate online on data streams with non-equidistant/-synchronized time, varying amount of reference sensors due to dropouts and potential outliers, uncertainty awareness and semantic pre-knowledge.

1.1.2 Tasks and Solution Ideas

In the set context, four tasks can be identified and ideas to their solutions outlined. The formulation of these tasks requires specific terms and concepts that are introduced in part II of this thesis.

Task 1: Metrology-aware Sensor Description

Metrology and with that traceability to the International System of Units (SI) is a key component of the quality infrastructure. However, this is so far not reflected by common knowledge representations describing sensors and sensor networks.

Solution idea: Although not every aspect required to cover metrological use cases is provided by a single knowledge representation, most concepts can be found in existing ontologies. The idea is to establish a metrologoy-aware sensor (network) description by proposing a merge of existing concepts and addition of missing concepts to achieve metrological usability.

Task 2: Traceable Co-calibration

Methods for estimating the transfer behavior of linear affine type exist, but do not evaluate the uncertainty of the estimate and therefore do not continue the traceability chain.

Solution ideas: To preserve traceability, it is necessary to maintain a calibration chain from national standards to the sensor of interest. This - by definition - excludes pure blind calibration methods. Recent approaches (e.g., [6, 7]) allow to specify certain sensors as reference sensors via proper configuration. Therefore, evaluation of uncertainty according to the GUM [8] for these existing co-calibration methods would achieve traceable estimates.

An alternative approach is a Bayesian estimation method to provide probabilistic information about the estimate. This approach is preferred, as it also allows to address Task 3 and Task 4.

Task 3: Usefulness in Practical Applications

The co-calibration is assumed to be used in real process environments. Data is available in continuous streams, providing blocks of measurement data. Temporal alignment of measurement data of different streams cannot be guaranteed. Moreover, the data can include outliers from failing sensors or dropouts due to network communication errors. If multiple reference sensors are available during the co-calibration, an uncertainty-aware, traceable and robust consensus value needs to be calculated. Existing approaches cover some of these constraints, but not all at once.

Solution ideas: Temporal alignment can be achieved by uncertainty-aware interpolation to an agreed time-base. Multiple reference sensors could then be fused into a single reference using methods from analysis of interlaboratory comparisons and meta analysis, achieving robustness with regard to outliers and the number of input references. Online capabilities are enabled using iterative parameter estimation methods that calculate a posterior from the neighbourhood consensus.

Task 4: Using Pre-Knowledge in the Co-calibration

The shift towards digital technologies opens up vast automation possibilities. Although smart sensors are able to communicate their self-descriptions within the network, machine-interpretable self-descriptions are not used so far for automatic and ideal initialization of co-calibration methods.

Solution idea: A suitable transfer model structure is assumed to be deducible from a sensor's self-description and an informative prior for the parameterization can be chosen.

1.1.3 Research Questions

Dealing with the above mentioned tasks leads to the following research questions:

RQ1 How can information relevant for metrological co-calibration be stored in a semantically expressive way? What level of reasoning complexity is a fair trade-off between practicality and expressiveness?

RQ2 How can an iterative, outlier-robust and traceable co-calibration within a homogeneous sensor network be achieved using Bayesian methods?

RQ3 How can semantic pre-knowledge be incorporated to provide informative initializations of mathematical operations?

RQ4 What conditions must be fulfilled to perform a successful co-calibration in a homogeneous sensor network?

1.1.4 Content Structure

The thesis is structured as follows: The introduction (part I) establishes the scientific context by outlining the methodological approach chosen, guided by relevant publications the author of this thesis was involved in. Moreover, a small exemplary sensor network is introduced to show specific aspects of the proposed co-calibration method. The relevant background knowledge of this thesis is presented in part II by describing related fields, summarizing the state-of-the-art and introducing relevant theoretical concepts. Part III presents the proposed co-calibration method by specifying it mathematically and semantically. The methods are evaluated in part IV by implementing them and applying them to specific use cases. Moreover, the methods are also compared to an existing co-calibration method and the simulation results as well as the gained semantic expressiveness are discussed. The thesis closes with a conclusion of the presented research and provides an outlook on future topics in part V.

1.2 Approach and Contributions

This chapter provides an overview of the methods developed as part of this thesis, shows how they interconnect, mentions related publications and highlights the individual contributions to the state of the art.

1.2.1 Approach

The task of a “co-calibration under consideration of semantic knowledge” is divided into several distinct but linked subtasks, providing solutions to the problems mentioned in section 1.1.2. The general assumption is that there is a sensor network available which is supposed to be used to co-calibrate a “new” sensor [2]. The selection of a subnetwork of suitable reference sensors from the sensor network requires additional knowledge (i.e., measured quantity, calibration information). Such knowledge needs to be represented in a semantically expressive way in order to enable an automated and machine-interpretable process, similar to the idea of the semantic web [9]. The metrological relevant information is used to select homogeneous calibrated reference sensors and, if necessary, to compensate and provide uncertainty information to the data stream of each reference sensor. Moreover, all uncertainty-enhanced data streams need to be interpolated to the same time-base. Offline [10, 11] or online [12] linear interpolation can be used (with some adjustments to reject interpolation if the period between two timestamps gets too long). If multiple reference sensors are available, they are fused into an uncertainty-aware robust reference value. The fused reference value and the calibration target are then used in a co-calibration method to provide estimates of the parameters characterizing the transfer behavior of the “new” sensor. Upon successful co-calibration (e.g., based on the uncertainty of the parameter estimate) the result is transferred back into the semantically expressive form and attached to the “new” sensor, making it a part of the network of (co-)calibrated sensors. Figure 1.2.1 summarizes this in a block diagram. The method is tested on simulated datasets.

[2]The “new” sensor could be a physically new sensor or an existing sensor for which some heuristic proposes a re-calibration.

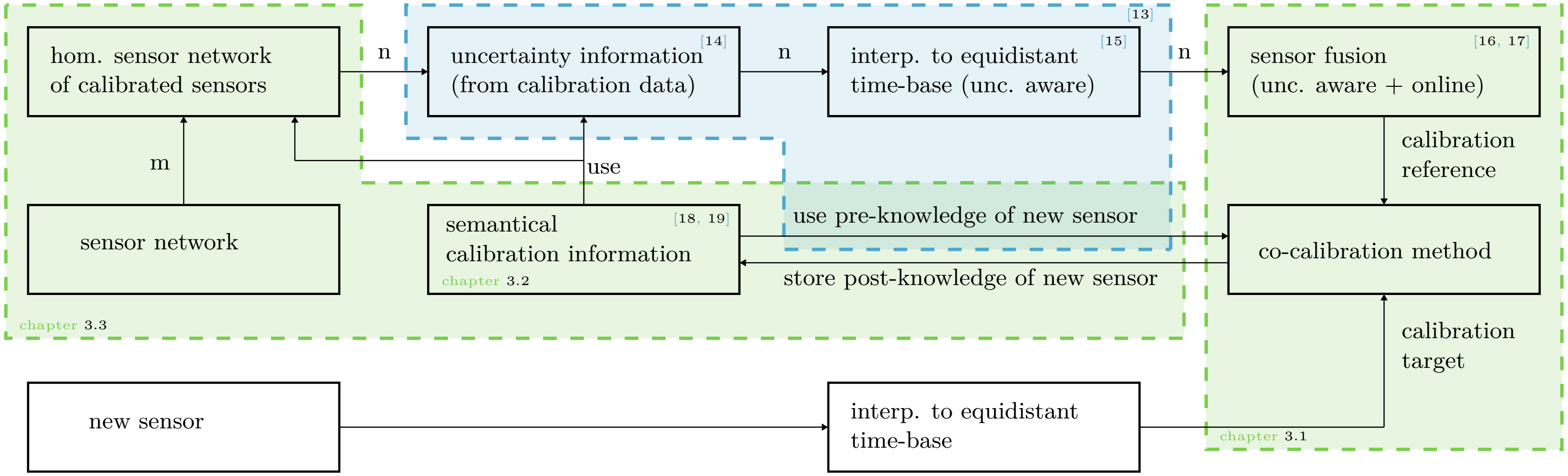

Figure 1.2.1: Distinct aspects of the proposed co-calibration method with relevant citations and links to the specific chapters in this thesis.

1.2.2 Scientific Methods

The chosen scientific method is of quantitative nature, as a co-calibration routine is proposed and evaluated against a ground truth in simulative studies and against another state-of-the-art co-calibration method [20]. The executed research is constructive, in that it proposes a solution to the identified problem and evaluates it against an existing method.

1.2.3 Contributions

Multiple aspects of the proposed methods have already been published and an overview of the corresponding publications is given in table 1.2.1.

In terms of contributions to the state-of-the-art, the following advancements have been made:

- integration of semantic information to the co-calibration process
- uncertainty aware co-calibration of linear affine transfer behavior
- explicit consideration of metrological traceability in the method design

Publication Title	Authors	Relevant contribution	Related to	Type	Status
Semantic Information in Sensor Networks: How to Combine Existing Ontologies, Vocabularies and Data Schemes to Fit a Metrology Use Case [18]	Gruber et al.	merge of existing ontologies	RQ1	Conference Proceedings	Published ⋆
Uncertainty-Aware Sensor Fusion in Sensor Networks [16]	Gruber et al.	sensor fusion method with uncertainty	RQ2	Conference Proceedings	Published ⋆
Discrete Wavelet Transform on Uncertain Data: Efficient Online Implementation for Practical Applications [14]	Gruber et al.	online uncertainty-aware IIR filter	RQ2	Book Series	Published ⋆
Application of Uncertainty-Aware Sensor Fusion in Physical Sensor Networks [17]	Gruber et al.	application of fusion method	RQ2	Conference Proceedings	Published ⋆
Modeling Dynamic Measurements in Metrology and Propagation of Uncertainties [21]	Gruber et al.	introduction to dynamic measurement uncertainty and mapping of calibration task to the GUM (part of [22])	RQ2	Book chapter	Published
Uncertainty-Aware Data Pipeline of Calibrated MEMS Sensors Used for Machine Learning [15]	Dorst et al.	application of uncertainty-aware interpolation	RQ2	Journal Paper	Published ⋆
Semantics in Sensor Networks: An Ontology for Dynamic Transfer Behavior in Calibrated Sensors [19]	Vedurmudi et al.	extension of [18] to cover dynamic transfer behavior	RQ1	Conference Proceedings	Published ⋆
Toward Smart Traceability for Digital Sensors and the Industrial Internet of Things [13]	Eichstädt et al.	use calibration information to compensate dynamic data stream	RQ3	Journal Paper	Published ⋆

Table 1.2.1: An overview of relevant publications the author of this thesis was involved in. Peer-reviewed publications are highlighted by ⋆. Additional publications the author was involved in are listed in appendix E.3.

1.3 Exemplary Use Case

This chapter briefly introduces a prototypical sensor network which will be used throughout this thesis to illustrate aspects of the development of a co-calibration method. The network consists of six sensors measuring different quantities at three locations, as summarized in table 1.3.1 and visualized in figure 1.3.1.

Despite being a very small sensor network, it is a distributed heterogeneous sensor network that contains a subset of sensors forming a homogeneous sensor network. On an abstract level, this matches a setup which could be found in an industrial setting. There, large sensor networks are deployed but can contain subsets of sensors that are suitable for the co-calibration presented in this thesis. Moreover, these large networks highlight the need for machine-interpretable sensor self-descriptions in order to automate the identification of such co-calibration capable homogeneous sub-networks.

ID	**quantity**	**location**	**calibrated**
S1	acceleration	A	yes
S2	acceleration	A	yes
S3	acceleration	B	yes
S4	temperature	A	yes
S5	temperature	C	no
S6	acceleration	A	no

Table 1.3.1: Sensors of the exemplary sensor network.

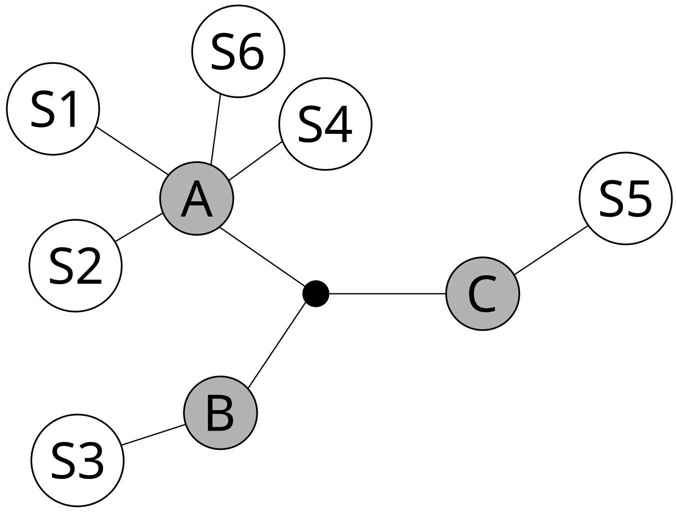

Figure 1.3.1: Topology of the exemplary use case.

Part II

Background

2.1 Related Fields and Developments

This chapter provides an overview of the domains related to this thesis and outlines their recent developments.

2.1.1 Metrology

Metrology is the science of measurement. It is concerned with providing stable, comparable and accurate measurements within a stated level of confidence [1, 3]. As such, it is a key part of any quality infrastructure, enabling and supporting scientific, environmental, technological and engineering advancements. Metrological competence is often concentrated in National Metrology Institutes (NMIs) that are concerned with the establishment of measurement units and their transfer to industry using measurement standards [23].

To meet these aims and enable trust in measurements, four interlinked concepts fundamental to metrology are at work: references, calibration, uncertainty and traceability. A reference can provide measurement results that are fit enough to assess the measurement trueness of another measurement device [3, VIM 2.7]. Primary references are typically hosted by NMIs and realized according to international standards [24] that are based on the definition of the SI units in terms of fundamental constants [25]. The quality of measurement of this reference is quantitatively characterized by stating the associated uncertainty, which is a "non-negative parameter characterizing the dispersion of the [measurement values]" [3, VIM 2.26]. The process of calibration provides uncertainty information about indications based on corresponding reference measurements [3, VIM 2.39]. Furthermore, a relation to obtain (an estimate of) the measurement result from the indication can be fitted. This is necessary, because the sensor's transfer behavior can differ from identity. Transfer behaviors can come in various degrees of complexity, ranging from linear static, dynamic to non-linear behavior and their modelling is motivated by the measurement principle. A measurement result is called traceable, if it can be related to a (primary) "reference through a documented unbroken chain of calibrations" [3, VIM 2.41].

Calibrations therefore not only allow to specify the level of confidence of a measurement device, but also link it to the primary standards and thereby to the SI. This justifies comparison of measurements taken by different devices at different locations can be compared (within bounds specified by the attributed uncertainty). A common representation of the relations between standards, calibration, traceability and uncertainty is provided in the "metrology pyramid" shown in figure 2.1.1.

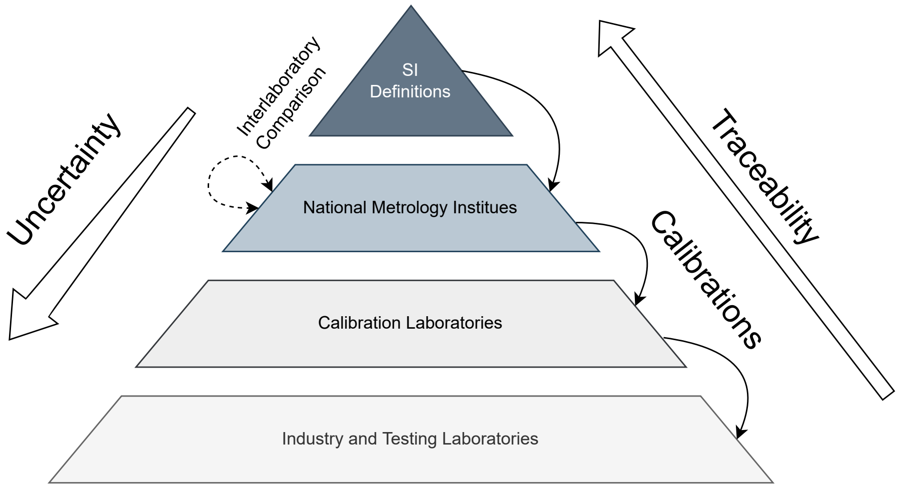

Figure 2.1.1: The metrology pyramid, based on [24, 26, 27].

More recent developments in the field of metrology are concerned with the extension of calibration models to cover dynamic transfer behavior. Moreover, it is of interest to prepare metrology for a shift towards digital technologies. This includes metrological precise and complete communication of measurement data with digital tools [28]. Additionally, transfer of existing certification documents into machine-actionable counterparts is an important part of this emerging digital transformation.

The unbroken chain of calibrations required for a traceable measurement is in practice achieved by regulating the institutions that provide calibration services, i.e., they are required to fulfill the requirements given in ISO 17025 [4] and complete an official accreditation process[3]. These institutions can then issue calibration certificates for a device that certifies the operability within certain operation ranges. In an ongoing effort to prepare this system for the digital age, a digital calibration certificate (DCC) is developed [30]. A DCC is an XML data structure, that allows to specify regulatory calibration aspects such as administrative data and measurement results as well as comments in a recognized and machine-readable format. Regulatory data fields are fixed and mandatory (e.g., calibration laboratory, calibration object or customer). Measurement data can be included in different ways, but always needs to provide unit, quantity and uncertainty information in accordance with the SI, e.g., by adopting the data format of the digital SI (D-SI) [31]. Additional information about the measurement process can be included in the unregulated comment section of a DCC. A DCC can embed a human-readable version of itself in the form of a PDF/A document in a separate section. Moreover, the XML document can be cryptographically signed, to verify its origin and enable integrity checks. With this, the DCC format provides a solid base to enable automated applications that require trusted and traceable measurement data.

[3]E.g., within the European Union this process is defined by regulation 765/2008 [29] and its transfer into national legislations of the member states.

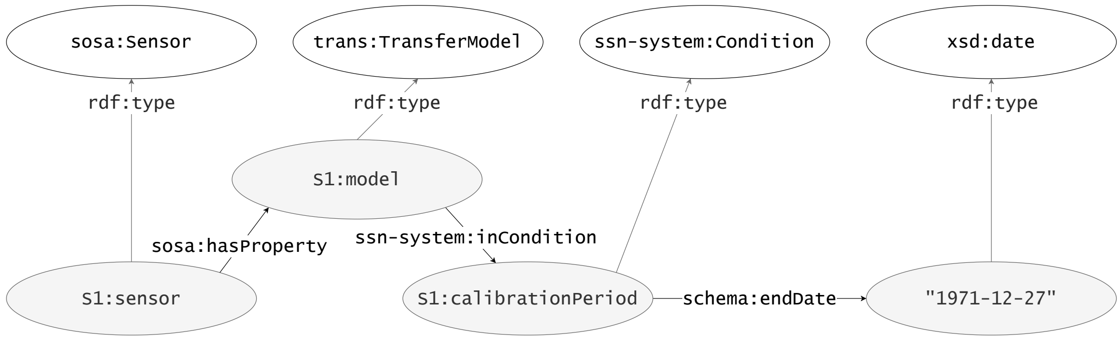

Figure 2.1.2: Knowledge graph expressed by listing 2.1.1.

2.1.2 Semantic Web

The Semantic Web is an extension of the already existing web (World Wide Web) and based on the idea of transforming human readable information into (also) machine-actionable information. It is based on three basic components [9]:

First, the addition of formal logic to the web. This translates to providing a language to express data and rules to enable machines to reason about the web's content. Moreover, this language needs to be powerful enough to describe complex properties, but also remain logically decidable within practical bounds, e.g., whether a question leads to a conflict given some specific knowledge basis.

Second, the integration with existing standards. By building atop of accepted tools that are already in use for the web development, the semantic web is an evolution rather than a parallel instance of the existing web. The extensible markup language (XML) allows to add (hidden) annotations to objects or web content, as the structure of annotations is open to custom types. The resource description framework (RDF) allows to express the meaning and relations of terms by using triples. These triples correspond to elementary sentences (subject, verb, object) and thereby allow the expression of knowledge in a very structured and fundamental way. The uniform resource identifier (URI), allows to globally and uniquely identify concepts [32] [4]. By combining all three technologies, it is possible to state relations between globally unique concepts (addressable by their URI) by using them in RDF triples. These triples can then be attached to elements of some web content by XML annotations.

Third, the availability of collections of information that capture domain specific knowledge. This is done within ontologies that define concepts and relations between them. Relations to concepts of other ontologies are possible as well, e.g., allowing to define synonyms or sub-concepts.

As a short example consider the sentence "Sensor S1 is calibrated until 12th of April 2026.". Using the XML elements <div> (block-container) and <span> (inline-container) in listing 2.1.1 it is possible to annotate information to this sentence without altering its visual rendering [33]. Although not directly observable by a user, the sentence now contains the machine-actionable knowledge graph shown in figure 2.1.2.

[4]The uniform resource locator (URL) is a subset of the URI.

```
<div about="S1:sensor">
    <span typeof="sosa:Sensor">Sensor S1 </span>
    <span property="sosa:hasProperty" content="S1:model">is calibrated </span>
</div>
<div about="S1:model">
    <span typeof="trans:TransferModel"></span>
    <span property="ssn-system:inCondition" content="S1:calibrationPeriod"></span>
</div>
<div about="S1:calibrationPeriod">
    <span typeof="ssn-system:Condition"></span>
    <span property="schema:endDate" content="2026-04-12" datatype="xsd:date">
        until 12th of April 2026.
    </span>
</div>
```

Listing 2.1.1: Expressive version of the sentence "`Sensor S1 is calibrated until 12th of April 2026.`" using Semantic Web methods.

2.1.3 Industrial Internet of Things

The industrial internet of things (IIoT) describes a new era in industry that marks a change in the use of available information and availability of computational resources [34]. Knowledge previously only used for specific tasks is made available on a much larger scale, forming networks of connected knowledge and enabling process optimization in a more global sense. This increase in connectivity and information processing requires computational resources. Use cases in this sense are, e.g., automatic selection of fallback sensors in case of sensor failure, re-calibration of drifting sensors based on redundant local information, production plan adjustments based on logistic information or adaptive maintenance schedules based on a predicted device status.

The foundation of the IIoT are data sources (i.e., databases, sensor data streams, process settings) that can communicate via standardized network interfaces [34]. This can be achieved by including such requirements during the design of a new plant or retrofitting devices in existing processes. Computational resources close to the hardware are typically referred to as edge-devices. They often act as network adapters for sensing and actuation hardware, but also provide limited computational resources capable of simple information aggregation tasks. Although this seems like an issue, it rather is an asset, leading to low power requirements. More complex analysis can be performed on more capable hardware that is no longer close to the actual process hardware, commonly referred to as a cloud-device. Both, cloud- and edge-devices, complement each other and establish prototypical concepts of IIoT-devices. Figure 2.1.3 shows an exemplary hierarchical schematic

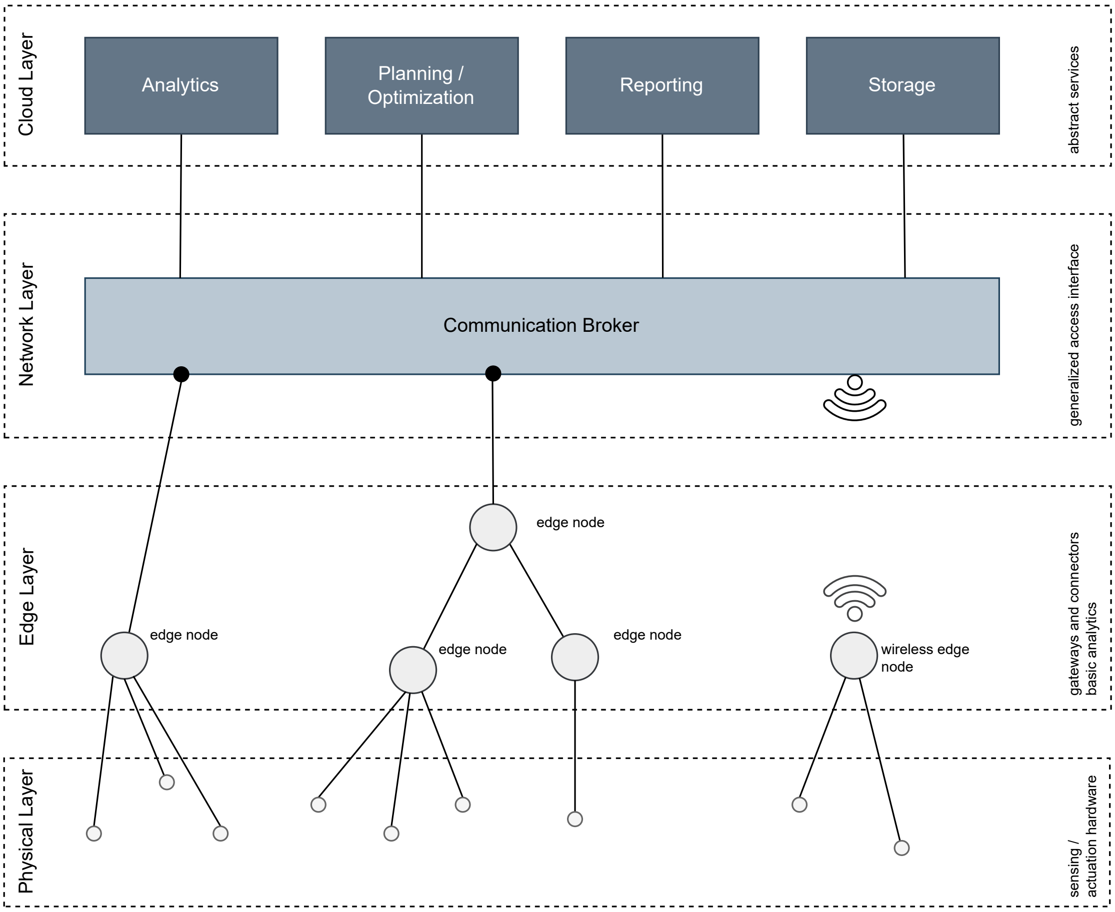

Figure 2.1.3: Exemplary components of a generic IIoT architecture, based on [34, 35, 36].

2.2 State of the Art

The main contributions of this thesis are in the domain of sensor networks. Specifically, contributions to co-calibration in homogeneous sub-networks, use of semantic information for initialing such co-calibration and representation of metrological aspects in the semantic descriptions of these networks are developed. The following sections provide an outline of the current state of the art in these fields.

2.2.1 Co-Calibration in Sensor Networks

National Metrology Institutes (NMIs) are concerned with the establishment of measurement units and their transfer to industry using measurement standards [23]. Therefore, the evaluation and interpretation of measurement uncertainty play an important role in metrology. Sensor calibration is a key element of metrology to assess the performance of a sensor in comparison to primary measurement standards. A calibration is performed at laboratories of NMIs and accredited bodies against confirmed and maintained references.

Data acquisition and measurement are key components for process control in industrial environments. Within the development of smart factories ("Industrie 4.0"), additional sensors ("Industrial Internet of Things") and exhaustive analysis of this data enables further improvements in terms of safety, efficiency and quality improvements by exploiting unused capacities [37]. The deployment of networks consisting of low-cost, small and portable sensors becomes increasingly popular [2] in this context. Recent applications of sensor networks can be found in any domain with high sensor counts, e.g., monitoring of technical equipment [38, 39, 40], object detection [41, 42], or environment surveillance [43, 44, 45, 46]. Typically, the single sensors show reduced accuracy and reliability compared to established industrial sensors, often accompanied by non-traceable or missing "calibration" data.

Thorough calibration of low-cost sensors following methods developed at NMIs is usually neither within budget limits nor feasible due to the sheer number and difficult accessibility of sensors - especially in industrial contexts. To this end, the term co-calibration is introduced to relate to a process that mimics the calibration process under conditions and resources available on-site. Therefore, co-calibration of sensor networks has recently received a lot of attention and several methods have been suggested for this purpose [45, 46]. Some of these methods make use of an overall reduction of the number of calibrations through so-called multi-hop methods [47, 48, 49]. Other approaches apply Kalman filtering for drift correction [50, 51, 52, 53].

Co-calibration (or blind calibration) in sensor networks typically refers to parameter estimation of a sensor model such that a consensus about the underlying physical phenomenon causing the sensor readings is found, e.g., the method proposed in [6]. A recent review on low-cost sensor networks highlights the need for on-site re-calibration to maintain the system's measurement performance [54]. Another need in this context is the reliability of the achieved results, which is described in [55].

Sensor networks are commonly described as communication graphs [6, 49, 56, 57], that allow flexible grouping of neighboring nodes. Changing topology is considered in [58].

The calibration models encountered in the literature are typical of linear affine type [6, 45, 49, 52, 54, 57, 59, 60, 61, 62, 63]. Non-linear calibration models are observed as well [45, 59, 64], but no indications of non-static sensor models have been found, i.e., such that the sensor output depends on the history of the input. The closest match would be a non-ideal transmission by a fixed transfer function in [56]. A physical phenomenon in [56] is assumed to follow a differential equation, but this does not seem to be further utilized within the proposed calibration scheme. In [61] the known dynamics of multiple input phenomenons are used to allow sensors to observe different phenomenons over time.

Uncertainty of input, output, parameters or estimates is considered in [65] - although without links to common concepts of metrology. Other papers do not consider uncertainty directly, but provide convergence analysis of their methods [6, 56, 60]. Another approach to evaluate uncertainty is by incorporation of Bayesian methods [52, 61], e.g., in [61] distributions for input, output, gain and offset are approximated and used. However, the mentioned methods lack the rigorous uncertainty quantification that has been established for single-sensor calibration, e.g., [**eichstÃďdt_2012**, 8, 66, 67, 68].

It is often assumed that sensor readings of all sensors are available at the same time with equidistant spacing in time. However, some authors also research the case and influence of non-ideal timestamps (delayed, lossy, non-synchronous) [6, 56] or are to some extend independent of them [69]. The majority of the publications mentioned here operate directly on the time-series data, but other approaches are exploited as well, e.g., [69] is inspired by transfer learning and adjusts model parameters by comparison of distributions.

Sensor network calibration relies on redundant information about the same physical phenomenon causing the sensor observations. Therefore, homogeneous sensor networks are of major interest and refer to sensors that measure the same quantity kind (e.g., [7]) or (in a more specific sense) measuring the same measurand, implying a certain proximity of the sensors (e.g., [6]). Initially, dense sensor networks are assumed to provide a sufficient degree of redundancy within the sensor readings, e.g., [59]. In [60] an abstract limit for the redundancy is found by utilization of subspace-methods. This idea is adopted in [52, 61, 70].

The co-calibration process can be carried out all at once (e.g., [71]) or continuously on new available data (e.g., [6]).

2.2.2 Semantics of Sensor Networks

Knowledge representation frameworks have reached a state that allows to express (for most practically relevant applications) sufficiently complex domain knowledge while staying decidable within practical limitations [72]. Within the semantic web community, knowledge is mostly represented using domain specific ontologies and vocabularies based on the web ontology language

(OWL) [73]. The adoption of a common format simplifies the linking and combination of existing ontologies, e.g., using manual or (semi-)automated merge and alignment approaches [74]. Multiple variants of OWL exist to fulfill different needs of expressiveness and reasoning complexity [75, 76]. Evaluations and reasoning over such (distributed) knowledge graphs can be accomplished using reasoners that make implicit relations explicit (e.g., [77]) and the SPARQL query language which is designed to operate on knowledge graphs [78]. Other querying languages exist as well [79, 80, 81, 82].

Most current ontologies and vocabularies related to sensors and sensor networks often build on the technologies of the semantic web [83, 84, 85, 86, 87, 88, 89]. Some knowledge representations are not (yet) compatible with OWL, but relevant to describe sensor networks in a metrological sense [30, 31, 90].

The data model of the *Digital SI* (D-SI, [31]) is the outcome of recent efforts of metrology institutes and can be used to represent values, units and uncertainties in digital communications. It defines a data model for the exchange of measurement results and complies with existing metrology standards, mainly “Le Système internationale d’unités” [24]. The concept of a quantity value (`si:realQuantityType`) is specified by a numeric value, unit, optional label, optional timestamp and optional uncertainty expression (which covers expanded uncertainty, coverage intervals and multivariate covariances). Recent discussions in the metrologic community have also identified the quantity kind as an essential part of the measurands description, as equality of units does not guarantee physically meaningful combinations of two sensor readings [91, 92].

Often, additional information about single sensors is available, e.g., in a (digital) calibration certificate. This information is so far largely neglected in the development of methods for the self-calibration of sensor networks. One reason is the use of machine-readable, but not machine-actionable representations of such knowledge. To this end, the DCC marks a turning point by designing a structure intended for machine use. The *Digital Calibration Certificate* (DCC, [30]) implements a data scheme that aims to replace paper certificates. It therefore contains administrative data, information about the calibration process, measurement data and a mathematical description of the identified transfer behavior including an uncertainty quantification. The resulting XML document is machine-readable and can be electronically signed. At its core, the DCC characterizes the results of a calibration, i.e., measurement results or mathematical descriptions of the sensor behavior. While equations are represented as LaTeX-style or MathML expressions, measurement data is provided in the D-SI format.

The D-SI and DCC cover essential aspects of metrology and with that provide a comprehensive set of metrological core knowledge. However, they do not express the interconnection between the used concepts (quantity, unit, calibration model, etc.) in a machine-understandable form. To evolve mere machine-readability into machine-interpretability, the used descriptors need to be linked together - and with that become semantically enriched concepts. To achieve this, the following knowledge representation frameworks could provide core concepts that can then be linked to achieve a semantic description of sensor networks.

The *Semantic Sensor Network* (SSN, [83]) and *Sensor, Observation, Sampling and Actuation* (SOSA, [84]) ontologies are developed by the World Wide Web Consortium (W3C) and the Open Geospatial Consortium (OGC). They complement each other by providing modular and broad knowledge representations for sensor applications. While SOSA covers core relations of sensors, observations, results, measurands and measurement procedures, SSN focusses more on the systemic aspect of sensors and sensor networks. In that SSN can be used to annotate detailed properties, system description, and deployment characteristics for instance. Alignements from SSN and SOSA to other ontologies are available, and the IoT-focused ontology IoT-Lite [85] is based on SOSA/SSN.

The *Ontology of Units of Measure and Related Concepts* (OM, [86]) defines units, quantities, quantity kinds and measures as concepts. Multiples and products of units can be defined as new units, e.g., allowing to introduce prefixed units (e.g., "milli-second") while linking the constituent concepts. Moreover, many common units and subsets of units for specific fields are predefined. Conversion to SI units is possible from the available knowledge, and quantity kinds are defined by their dimensions in the SI base kinds. Measurement results are represented by a numerical value and unit (and with that implicitly also a quantity). The concept of measurement uncertainty is not covered by OM.

The *Quantities, Units, Dimensions and Types* (QUDT, [87]) ontology has a similiar scope as OM and lists [24] as a key reference. It consists of a central ontology with accompanying vocabularies for specific unit, quantity and (physical) dimension instances. Quantity dimensions are uniquely represented using a vector storing the exponents of base dimensions in a chosen unit representation system (e.g., SI, Imperial, CGS). An extensive list of common quantities, units and systems of units are predefined and conversions are possible. Units with prefixes (e.g., *centi*-meter) are handled as a derived units, hence keeping the link between both and allowing conversion via a conversion factor. The result of a measurement is called a quantity value, which is specified by a quantity kind, numeric value, unit and an optional standard uncertainty.

The *Engineering Mathematics* (EngMath, [90]) ontology allows to describe mathematical models used in engineering contexts. As such, it provides the semantic concepts relevant to make variables, physical meaning (quantity kind), units and data dimensionality explicit. These concepts in general allow to check for dimensional consistency in an automated way. Although it is an ontology, it is not available as an OWL-ontology, as it was developed before the existence of the OWL-standard.

Representing spatial information can require both geometrical as well as topological aspects to define e.g., a proximity relation. The *Geographic Query Language* (GeoSPARQL, [88]) vocabulary covers both of these aspects by defining spatial objects and their relations in an abstract way. Based on this vocabulary, a custom ontology that provides spatial information relevant to a use case can be created.

The *Mathematical Markup Language* (MathML, [89]) is developed by the W3C intended to capture mathematical formula. Two variants exist: while the "presentation markup" aims at capturing common mathematical notation (e.g., `a*b`), the "content markup" captures the meaning of operations (e.g., `multiplication(a, b)`) and therefore reduces ambiguity from overlapping notations. MathML is not an ontology, but the content markup provides a semantically expressive way to serialize mathematical formula by providing links from operators and terms to semantic definitions through a `<semantic>` element.

2.3 Fundamentals and Definitions

This chapter provides key concepts of relevant fields, details mathematical aspects and an introduction to the required terminology. Because the developed co-calibration method is based on Bayesian inference, the fundamental concepts of this framework are also provided. Moreover, the concept of uncertainty evaluation and propagation according to metrological standards is presented. Relevant definitions from the field of sensor calibration and sensor networks are either quoted or formulated to establish a common understanding of the concepts used throughout this thesis. An introduction of semantic knowledge representation and reasoning is given to prepare the use of these methods in sensor network contexts.

2.3.1 Bayesian Framework

Bayesian statistics is characterized by modeling all (latent and observable) unknowns as random variables - which contrasts classical statistics, where only observable variables are modeled [93]. This is a consequence of the meaning that the term “probability” has in Bayesian statistics: rather than using it in the (frequentist) sense of a “rate of occurrence”, it is seen as a “degree of belief”. This fundamentally different notion allows to assign a probability to latent variables. In the majority of practically relevant cases, the knowledge about an unknown (numerical) variable $\underline{\theta}$ can then be expressed using a probability density function (PDF) $p(\underline{\theta})$ [94].[5]

Definition 1 (probability density function properties, [95])**.** *Let $\underline{\theta} \in \mathbb{R}^N$ with $N \in \mathbb{N}$ be a N-dimensional parameter. Then a probability density function p : [95]*

- *maps from $\mathbb{R}^N \to \mathbb{R}$*
- *is non-negative: $\forall_{\underline{\theta} \in \mathbb{R}^N} p(\underline{\theta}) \geq 0$*
- *covers all possibilities: $\int_{\mathbb{R}^N} p(\underline{\theta}) d\underline{\theta} = 1$*

Hence, a variable is not characterized by a single value, but by a range of possible values each with potentially different probability.

The process of updating existing beliefs based on new evidence (e.g., experimental data) is called Bayesian inference [94]. It utilizes the Bayesian theorem, which is based on conditional probabilities and provides its own interpretation of the result.

[5]Although out of scope, it should be noted that in the general case not every random variable necessarily has a PDF.

Theorem 1 (Bayes Theorem [94]). *Let $p(\underline{\theta})$ be a probability density function of a parameter $\underline{\theta} \in \mathbb{R}^N$ and $\underline{X} \in \mathbb{R}^M$ some new experimental data whose distribution depends on $\underline{\theta}$. Then the updated belief $p(\underline{\theta}|\underline{X})$ (read "θ under the condition that $\underline{X}$ was observed") is: [94]*

$$p(\underline{\theta}|\underline{X}) = \frac{p(\underline{X}|\underline{\theta}) \cdot p(\underline{\theta})}{p(\underline{X})} \tag{2.3.1}$$

$$\propto p(\underline{X}|\underline{\theta}) \cdot p(\underline{\theta}) \tag{2.3.2}$$

Where $p(\underline{\theta})$ is commonly referred to as the prior (belief), $p(\underline{X}|\underline{\theta})$ the likelihood, $p(\underline{\theta}|\underline{X})$ the posterior (belief) and $p(\underline{X})$ the model evidence.

In order to use this formula in practical applications, further assumptions are necessary. A suitable prior is required, which can include informative choices based on expert knowledge (typically leading to PDFs with narrow bands of high density) or non-informative choices (leading to very wide bands of non-zero density). The likelihood (which is seen as a function of $\underline{\theta}$) needs to provide an answer to the question "How likely are the observed data, given some value of $\underline{\theta}$?" and with that models the relation between parameters and data. Once assumptions on prior and likelihood are established, the posterior can be evaluated using analytical (preferred, but often unfeasible), numerical (e.g., evaluating an analytical solution on a discrete grid) or sampling based (e.g., Markov-Chain-Monte-Carlo (MCMC)) methods.

2.3.2 Uncertainty Evaluation

Uncertainty propagation in metrology typically refers to the "Guide to the expression of uncertainty in measurement" (GUM) and its specialized parts [8, 96, 97, 98]. The different documents have individual scopes:

- GUM-3: uncertainty evaluation for scalar quantities depending on multiple scalar input quantities using a first order approximation [8]
- GUM-7: uncertainty evaluation for scalar quantities depending on multiple scalar input quantities using Monte Carlo simulations [96]
- GUM-8: uncertainty evaluation for vector quantities depending on multiple input quantities (first order and Monte Carlo) [97]
- GUM-6: guidance on the development and use of (dynamic) measurement models [98]

Uncertainty evaluation in the GUM is a three stage process of: *formulation* of model assumptions, *propagation* of uncertainty (or of complete distributions associated with the input quantities) through the model and *summarizing* the propagation result in terms of an estimate and its associated uncertainty [97].

In the formulation stage, a measurement function $\underline{f} : \mathbb{R}^N \to \mathbb{R}^M$ is required that relates the sought quantity $\underline{Y} \in \mathbb{R}^M$ to the input quantities $\underline{X} \in \mathbb{R}^N$:

$$\underline{Y} = \underline{f}(\underline{X}) \tag{2.3.3}$$

The model $\underline{f}$ can be a placeholder for explicit mathematical equations or involved algorithms. The knowledge about $\underline{X}$ is specified by a (joint) probability density function (PDF) $g_{\underline{X}}$ or by stating an estimate $\hat{\underline{x}}$ of $\underline{X}$ along with a covariance matrix $\boldsymbol{U}_{\hat{\boldsymbol{x}}}$.

The propagation stage then yields either a PDF $g_{\underline{Y}}$ representing the knowledge about the measurand $\underline{Y}$, or an estimate $\underline{y}$ with a covariance matrix $\boldsymbol{U_y}$. The evaluation of the propagation stage can use (1) analytical methods, (2) the law of propagation of uncertainty or (3) Monte Carlo methods. Analytical methods result in non-approximative mathematical representation of $g_{\underline{Y}}$, but the required calculations only remain manageable for simple measurement models and PDFs. To keep calculations manageable and often still maintain a sufficient treatment, the "law of propagation of uncertainty" proposes to perform the propagation using a first order approximation of $\underline{f}$. The resulting estimate $\underline{y}$ of the measurand $\underline{Y}$ is obtained by inserting the estimate $\underline{x}$ of the input quantity $\underline{X}$ into the functional relationship (equation (2.3.3)). The associated covariance matrix $\boldsymbol{U_y}$ is calculated from the covariance of the inputs $\boldsymbol{U_x}$ and the sensitivity $\boldsymbol{C}$ of the measurement function $\underline{f}$ [97]:

$$\boldsymbol{U_x} = \mathbb{V}(\underline{X}) \tag{2.3.4}$$

$$\boldsymbol{C} = \begin{bmatrix} \frac{\partial f_1}{\partial X_1} & \cdots & \frac{\partial f_1}{\partial X_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_M}{\partial X_1} & \cdots & \frac{\partial f_M}{\partial X_N} \end{bmatrix} \tag{2.3.5}$$

$$\begin{aligned} \boldsymbol{U_y} &= \mathbb{V}(\underline{Y}) \\ &= \boldsymbol{C}\boldsymbol{U_x}\boldsymbol{C}^T \end{aligned} \tag{2.3.6}$$

As a second alternative to the analytical propagation, Monte Carlo methods (MCM) can be used to propagate samples of the PDF associated with the input quantity through equation (2.3.3), yielding a comprehensive empirical representation of the output PDF. The MCM draws independent samples $\underline{x}_i$ from the PDF $g_{\underline{X}}$ and for each sample the model (equation (2.3.3)) is evaluated. The resulting samples $\underline{y}_i$ then provide independent samples from the output's PDF $g_{\underline{Y}}$. This takes full account of non-linearities in the measurement model. MCMs are usually computationally intensive (depending on the chosen number of drawn samples ("runs") and the complexity of $\underline{f}$).

The summary stage provides an expected output value with corresponding uncertainty. Therefore, applying a propagation of distributions, the estimate $\underline{y}$ and covariance $\boldsymbol{U_y}$ of the output quantity $\underline{Y}$ are given by the expectation and covariance operator:

$$\underline{y} = \mathbb{E}(\underline{Y}) \tag{2.3.7}$$

$$\boldsymbol{U_y} = \mathbb{V}(\underline{Y}) \tag{2.3.8}$$

To extract the same information from the result of an MCM, expectation and covariance matrix can be estimated according to:

$$\underline{y} = \frac{1}{m} \sum_{i=1}^{m} \underline{y}_i \tag{2.3.9}$$

$$\boldsymbol{U_y} = \frac{1}{m-1} \sum_{i=1}^{m} (\underline{y}_i - \underline{y})(\underline{y}_i - \underline{y})^T \tag{2.3.10}$$

The GUM approach is mapped to measurements of sensors with dynamic transfer behavior in [66, 98, 99, 100].

2.3.3 Sensor Calibration and Transfer Behavior

The International vocabulary of metrology (VIM) distinguishes between a sensor and a measuring system [3]. While the sensor is the physical element influenced by the measurand, the measuring system can consist of multiple sensors and also covers the operation of and information retrieval from the sensor(s).

Definition 2 (measuring system, [3, VIM 3.2])**.** *set of one or more measuring instruments and often other devices, including any reagent and supply, assembled and adapted to give information used to generate measured quantity values within specified intervals for quantities of specified kinds*

Definition 3 (sensor, [3, VIM 3.8])**.** *element of a measuring system that is directly affected by a phenomenon, body, or substance carrying a quantity to be measured*

However, to adopt the common use of the term sensor (and sensor network), in this thesis a sensor also refers to what the VIM defines as a measuring system with a single measuring instrument.

Definition 4 (sensor)**.** *A sensor is a device, that is directly affected by a physical phenomenon* $x(t)$ *at a time* t *and provides an indication* $y(t)$*.*

No sensor is ideal and it is of interest to quantify the relation between actual measurand values and the indicated values by comparing them to reference measurements. Such knowledge then allows to propose a relation that maps indications to measurand values, but only at a certain level of accuracy. This task is called a calibration, which the VIM defines as:

Definition 5 (calibration, [3, VIM 2.39])**.** *operation that, under specified conditions, in a first step, establishes a relation between the quantity values with measurement uncertainties provided by measurement standards and corresponding indications with associated measurement uncertainties and, in a second step, uses this information to establish a relation for obtaining a measurement result from an indication*

To understand the importance of sensor calibration as a cornerstone of the quality infrastructure, the (metrological) concepts of uncertainty, reference and traceability are required. Metrology assesses the quality of measurements and the central indicator for this is the uncertainty:

Definition 6 (uncertainty, [3, VIM 2.26])**.** *non-negative parameter characterizing the dispersion of the quantity values being attributed to a measurand, based on the information used*

Depending on the level of uncertainty in its measurement values, a measuring device might be suitable to estimate the error of another sensor and check for consistency in terms of quoted uncertainties. This device is then called a reference device:

Definition 7 (reference measurement procedure, [3, VIM 2.7])**.** *measurement procedure accepted as providing measurement results fit for their intended use in assessing measurement trueness of measured quantity values obtained from other measurement procedures for quantities of the same kind, in calibration, or in characterizing reference materials*

A special case are primary references which are not traceable but directly constructed by convention or based on fundamental constants stated in the SI [24]. A quality anchor for these standards are key comparisons [101].

To calibrate a new reference device, an existing calibrated reference is required. To resolve this recursive logic, the chain of calibration needs to start at a primary reference. This chain is the essence of metrological traceability and another cornerstone of the quality infrastructure:

Definition 8 (metrological traceability, [3, VIM 2.41])**.** *property of a measurement result whereby the result can be related to a reference through a documented unbroken chain of calibrations, each contributing to the measurement uncertainty*

Therefore, calibrations are the links that establish traceability and with that enable trust in a measurement by relating ultimately to the same primary reference and quantifying the uncertainty.

In the literature, the term "in-situ calibration" is used to denote calibrations at the place of use of a sensor and is established thoroughly by the works of Delaine, Lebental, and Rivano [46, 102].

Definition 9 (in-situ calibration algorithm, [102])**.** *In situ calibration algorithms aim at calibrating measuring instruments while leaving them in the field, preferably without any physical intervention.*

The term "in-situ calibration" (as used in [102]) is deliberately chosen to cover a wide range of procedures. However, only a reference-based definition can maintain traceability in the calibration results. Also, it is not always possible to introduce additional metrological equipment into a sensor network. Therefore, definition 10 introduces the term *co-calibration* as an important subclass of in-situ calibrations, which allows to maintain traceability.

Definition 10 (co-calibration)**.** *An in-situ calibration relying solely on available and traceable measurement information in a sensor network. This includes cases, in which no (official) reference measurement procedure is available, but instead a virtual reference is formed from a homogeneous and collaborative sensor network. Using the taxonomy introduced in [102], this corresponds to a "reference-based group-wise in-situ calibration".*

A special case of this is a homogeneous co-calibration using only reference sensor that are spatially close (co-location) and measure the same quantity.

Definition 11 (homogeneous co-calibration)**.** *A co-calibration that is performed in a homogeneous sensor network (definition 17)).*

Mathematically, a calibration operation requires to describe the sensor's transfer behavior, measurement data and a regression task to estimate the former from the latter. Such a model is often called the transfer behavior of the sensor.

Definition 12 (sensor transfer behavior)**.** *A mathematical model that characterizes the input-output-behavior of the sensor, typically a parameterized model structure $f_{\underline{\theta}}$ with parameter $\underline{\theta}$ [103]. The input $x(t)$ is the time-resolved value of the measurand (physical phenomenon), the output $y(t)$ is a time series of indications of the sensor.*

$$y(t) = f_{\underline{\theta}}(x(t)) \tag{2.3.11}$$

Depending on whether the output of this model depends on the history of the input, the model is called dynamic (if yes) or static (if not). [6]

The calibration task can then be formulated as follows using the formalism established in the "Guide to the expression of uncertainty in measurement" (GUM [8, 96], see section 2.3.2). The same (unknown) measurand $\psi_m(t)$ is simultaneously measured using a reference sensor and the sensor under test at k discrete points in time $t_0, ..., t_{k-1}$. The reference sensor provides a discrete

[6] Note, that a time-dependent (input or) output does not imply a dynamic transfer behavior.

time series $\underline{\psi}_r$ that is taken as the expected value of an associated PDF $\underline{g}_r$ (which allows to quantify the uncertainty). Similarly, the sensor under test provides a discrete time series $\underline{\psi}_c$ that can optionally be assigned with an associated PDF $\underline{g}_c$, such that

$$\underline{\psi}_r^* \sim \underline{g}_r \quad \text{with} \quad \mathbb{E}(\underline{\psi}_r^*) = \underline{\psi}_r = [\psi_r(t_0), \ldots, \psi_r(t_{k-1})] \tag{2.3.12}$$

$$\underline{\psi}_c^* \sim \underline{g}_c \quad \text{with} \quad \mathbb{E}(\underline{\psi}_c^*) = \underline{\psi}_c = [\psi_c(t_0), \ldots, \psi_c(t_{k-1})] \tag{2.3.13}$$

A (regression based) method $\underline{f}_{\text{cal}}$ can then extract a parameter estimate $\hat{\underline{\theta}}$ for the assumed model structure $\underline{f}_{\underline{\theta}}$. In the estimation routine the initial knowledge of the sought parameter $\underline{\theta}$ can be quantified using a PDF $\underline{g}_{\theta_0}$, such that

$$\underline{\theta}_0^* \sim \underline{g}_{\theta_0} \tag{2.3.14}$$

This leads to the following choices for $\underline{X}$, $\underline{Y}$ and $\underline{f}$ of equation (2.3.3) and allows to quantify the uncertainty of the parameter estimate $\hat{\underline{\theta}}$ using the propagation stage (section 2.3.2):

$$\underline{X} = \begin{bmatrix} \underline{\psi}_r^* \\ \underline{\psi}_c^* \\ \underline{\theta}_0^* \end{bmatrix} \sim \begin{bmatrix} \underline{g}_r \\ \underline{g}_c \\ \underline{g}_{\theta_0} \end{bmatrix} \tag{formulation}$$

$$\underline{Y} = \underline{f}_{\text{cal}}(\underline{X}) \tag{propagation}$$

$$\hat{\underline{\theta}} = \mathbb{E}(\underline{Y}) \quad , \quad U_{\hat{\underline{\theta}}} = \mathbb{V}(\underline{Y}) \tag{summarizing}$$

2.3.4 Sensor Networks

Multiple definitions of the term sensor network coexist for different applications. For instance, if the use case depends on whether or not a sensor can exchange data with another sensor to form an evaluation consensus the network could be represented as a graph of the connectivity of nearby sensors and neighborhood-properties [6, 104]. In another example, all sensors might be connected via a wired network-connection, and a useful graph representation (rather than the all-to-all connectivity graph) would cover how sensors arrange along a monitored industrial process [105]. A very broad definition of a sensor network is given in definition 13. This definition covers different relations constituting the network, but also distinguishes a sensor network from a mere set of sensors by requiring that there exists the possibility to combine sensor readings of different sensors.

Definition 13 (sensor network)**.** *A set of two or more sensors (measuring devices) that are linked by purpose, connectivity, proximity, measurand, quantity kind, or other relationships. There exists at least one evaluation node that has the ability to access at least two sensors.*

Definition 14 (evaluation node)**.** *place, where an evaluation routine or method is (computationally) executed. This functionality can be part of a sensor or some specialized centralized service.*

It is physically not possible to have multiple sensors at the exact same place. However, sensor that are spatially close could still measure the same quantity in practical terms. This region is called a co-location and defined in definition 15.

Definition 15 (co-location)**.** *Two sensors are co-located, if they are in spatial proximity. The usable proximity region depends on the underlying physical process, e.g. symmetry, topology or homogeneity of the observed quantity.*

Two further specializations of a sensor network are of interest.

Definition 16 (transient sensor network)**.** *A sensor network is called transient, if the set of sensors or their relations change over time, e.g., sensors are added or removed, or the connectivity changes.*

Definition 17 (homogeneous sensor network)**.** *A sensor network is called homogeneous, if all its sensors observe the same quantity kind in direct or co-located proximity, i.e., the network observes the same measurand.*

Sensors provide measurement data. Typically, the measurement is repeated continuously at a (potentially changing) rate, yielding temporally resolved measurements. The data has no obvious beginning or end, but is available in the form of a (endless) stream. Such sensor data streams provide the most recent measurement information of each sensor in the network.

Definition 18 (sensor data stream)**.** *A time-series of sensor data that only becomes available in blocks of finite length.*

2.3.5 Knowledge Representation, Semantic Expressiveness and Reasoning

Semantics is the study of the meaning of concepts and symbols [106]. The meaning of a concept can often be captured by highlighting its relations to other concepts, forming a graph of knowledge. This can be formalized by logical expressions and with that made machine-actionable. The knowledge formalization often builds on the framework of description logic (DL), which combines concepts and roles using axioms and predefined constructors. [72]

Every DL consists of terminological axioms, which define that one concept (or role) includes or equals another concept (or role). Additionally, assertional axioms allow to specify that an element is an instance of a certain concept or relate two elements by a relation. More complex concepts and roles can be created using predefined logical constructors and every such combination is again a concept or role within this DL[7]. The chosen set of available constructors defines the expressiveness of the DL. [107]

An interpretation $\mathcal{I}$ links concepts or roles to a (set) expression that allows evaluation of matching instances within a given domain Δ. The interpretation of a concept C is given by $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and the interpretation of a role R is given by $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

The OWL 2 Web Ontology Language provides a practical implementation of a DL and builds on RDF. As such, it is an important part of the technology stack of the Semantic Web. The basic axioms, concept constructors and role constructors of the OWL2 DL are stated in tables 2.3.1 and 2.3.2 using concepts C, D, roles R, S and individuals a, b and c. Moreover, OWL2 DL

[7]Description logics have their own naming convention, which is based on the included constructors. However, the scheme is not detailed here, but can be found, e.g., in [107].

supports some axioms and constructors for datatypes. By relying on the stated constructors, the OWL2 DL provides the base for expressive domain description. However, this expressiveness can lead to undecidable or very complex (NP-Hard) computations if all allowed constructors of OWL2 DL are considered.

To increase computational efficiency for tasks that do not require full expressive power, OWL2 defines profiles which only support subsets of the OWL2 DL constructors [76]. Three profiles are provided: *OWL2 EL* for ontologies with a large number of properties or classes (but not too high expressive needs, like e.g., inverse property, symmetry or disjointness) as well as *OWL2 QL* and *OWL2 RL* for smaller ontologies from which a large number of individuals is constructed. It is beneficial to check against the full profile specifications [76] whether an application fits without a (major) loss in expressive power into one of these profiles, as computational complexity can be significantly reduced.

Based on the base axioms and constructors available within a DL, a domain of knowledge can be modeled, which is often referred to as ontology engineering. Evaluation of an ontology is (among other things) achieved by formulating competency questions (CQs) which specify design goals for an ontology [108]. These CQs define the requirements of an ontology by outlining what information needs to be available and how potential applications access or retrieve this information.

2.3.6 On the Use of Pre-Knowledge and Prior Knowledge

In this thesis, pre-knowledge and prior knowledge refer to distinct but related concepts. Pre-knowledge is the semantic information that is available in advance about sensors in a network. This includes a very broad range of properties and relations and the details of it are discussed in chapter 3.2. The terms prior, prior knowledge or prior belief refer to the mathematical object which quantifies the degree of belief of some variable. As such, it is primarily used in the description of the mathematical core of the co-calibration method in chapter 3.1. Despite the differences between these two terms, specific pre-knowledge can be used to deduce prior knowledge. This is conceptualized in chapter 3.3.

Axiom / Constructor	Notation	Interpretation
inclusion	$\texttt{SubClassOf}(C, D)$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
equivalence	$\texttt{EquivalentClasses}(C_1, \ldots, C_n)$	$\forall_{1 \leq i,j \leq n} C_i^{\mathcal{I}} = C_j^{\mathcal{I}}$
disjointedness	$\texttt{DisjointClasses}(C_1, \ldots, C_n)$	$\forall_{1 \leq i,j \leq n; i \neq j} C_i^{\mathcal{I}} \cap C_j^{\mathcal{I}} = \varnothing$
disjoint union	$\texttt{DisjointUnion}(C, C_1, \ldots, C_n)$	$C^{\mathcal{I}} = \bigcup_{1 \leq i \leq n} C_i^{\mathcal{I}}$ and $\texttt{DisjointClasses}(C_1, \ldots, C_n)$
assertion	$\texttt{ClassAssertion}(C, a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
intersection⋆	$\texttt{ObjectIntersectionOf}(C_1, \ldots, C_n)$	$\bigcap_{1 \leq i \leq n} C_i^{\mathcal{I}}$
union⋆	$\texttt{ObjectUnionOf}(C_1, \ldots, C_n)$	$\bigcup_{1 \leq i \leq n} C_i^{\mathcal{I}}$
complement⋆	$\texttt{ObjectComplementOf}(C)$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
one-of⋆	$\texttt{ObjectOneOf}(a_1, \ldots, a_n)$	$\left\{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\right\}$
existential restriction⋆	$\texttt{ObjectSomeValuesFrom}(R, C)$	$\left\{a \mid \exists b \in \Delta^{\mathcal{I}} \text{ s.t. } (a, b) \in R^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}}\right\}$
value restriction⋆	$\texttt{ObjectAllValuesFrom}(R, C)$	$\left\{a \mid \forall b \in \Delta^{\mathcal{I}}, \text{ if } (a, b) \in R^{\mathcal{I}}, \text{ then } b \in C^{\mathcal{I}}\right\}$
has value⋆	$\texttt{ObjectHasValue}(R, a)$	$\left\{b \mid (b, a) \in R^{\mathcal{I}}\right\}$
is reflexive	$\texttt{ObjectHasSelf}(R)$	$\left\{a \mid (a, a) \in R^{\mathcal{I}}\right\}$
qualified min. restriction⋆	$\texttt{ObjectMinCardinality}(n, R)$	$\left\{a \mid \left\lvert\left\{b \in \Delta^{\mathcal{I}} \mid (a, b) \in R^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}}\right\}\right\rvert \geq n\right\}$
qualified max. restriction⋆	$\texttt{ObjectMaxCardinality}(n, R)$	$\left\{a \mid \left\lvert\left\{b \in \Delta^{\mathcal{I}} \mid (a, b) \in R^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}}\right\}\right\rvert \leq n\right\}$
qualified exact restriction⋆	$\texttt{ObjectExactCardinality}(n, R)$	$\left\{a \mid \left\lvert\left\{b \in \Delta^{\mathcal{I}} \mid (a, b) \in R^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}}\right\}\right\rvert = n\right\}$
equivalence	$\texttt{SameIndividual}(a_1, \ldots, a_n)$	$\forall_{1 \leq i,j \leq n} a_i^{\mathcal{I}} = a_j^{\mathcal{I}}$
non-equivalence	$\texttt{DifferentIndividuals}(a_1, \ldots, a_n)$	$\forall_{1 \leq i,j \leq n; i \neq j} a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$

Table 2.3.1: Concept axioms and expressions supported by the OWL2 description logic. Entries marked with $\star$ are available for datatype classes. [72, 107, 109, 110]

Axiom / Constructor	Notation	Interpretation
inclusion⋆	`SubObjectPropertyOf`(R, S)	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
equivalence⋆	`EquivalentObjectProperties`$(R_1, \ldots, R_n)$	$\forall_{1 \leq i,j \leq n} R_i^{\mathcal{I}} = R_j^{\mathcal{I}}$
disjointedness⋆	`DisjointObjectProperties`$(R_1, \ldots, R_n)$	$\forall_{1 \leq i,j \leq n; i \neq j} R_i^{\mathcal{I}} \cap R_j^{\mathcal{I}} = \varnothing$
assertion⋆	`ObjectPropertyAssertion`(R, a, b)	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
negative assertion⋆	`NegativeObjectPropertyAssertion`(R, a, b)	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin R^{\mathcal{I}}$
chaining	`ObjectPropertyChain`$(R, S_1, \ldots, S_n)$	$\forall a_0, \ldots, a_n : \wedge_{1 \leq i \leq n} (a_{i-1}, a_i) \in S_i^{\mathcal{I}} \implies (a_0, a_n) \in R^{\mathcal{I}}$
inverse	`InverseObjectProperties`(R, S)	$\forall a, b : (a, b) \in R^{\mathcal{I}} \implies (b, a) \in S^{\mathcal{I}}$
domain⋆	`ObjectPropertyDomain`(R, C)	$\forall a, b : (a, b) \in R^{\mathcal{I}} \implies a \in C^{\mathcal{I}}$
range⋆	`ObjectPropertyRange`(R, C)	$\forall a, b : (a, b) \in R^{\mathcal{I}} \implies b \in C^{\mathcal{I}}$
functional⋆	`FunctionalObjectProperty`(R)	$\forall a, b, c : (a, b) \in R^{\mathcal{I}}$ and $(a, c) \in R^{\mathcal{I}} \implies b = c$
inverse functional	`InverseFunctionalObjectProperty`(R)	$\forall a, b, c : (a, c) \in R^{\mathcal{I}}$ and $(b, c) \in R^{\mathcal{I}} \implies a = b$
reflexivity	`ReflexiveObjectProperty`(R)	$\forall a : (a, a) \in R^{\mathcal{I}}$
irreflexivity	`IrreflexiveObjectProperty`(R)	$\forall a : (a, a) \notin R^{\mathcal{I}}$
symmetry	`SymmetricObjectProperty`(R)	$\forall a, b : (a, b) \in R^{\mathcal{I}} \implies (b, a) \in R^{\mathcal{I}}$
asymmetry	`AsymmetricObjectProperty`(R)	$\forall a, b : (a, b) \in R^{\mathcal{I}} \implies (b, a) \notin R^{\mathcal{I}}$
transitive closure	`TransitiveObjectProperty`(R)	$\forall a, b, c : (a, b) \in R^{\mathcal{I}}$ and $(b, c) \in R^{\mathcal{I}} \implies (a, c) \in R^{\mathcal{I}}$

Table 2.3.2: Role axioms and expressions supported by the OWL2 description logic. Entries marked with ⋆ are available for datatype properties. [72, 107, 109, 110]

Part III

Semantically enabled Co-Calibration

3.1 Development of a Consensus Based Co-Calibration Method

In this chapter, the mathematical details of the developed co-calibration methods are provided. At its core, a co-calibration provides traceable[8] estimates of the parameters defining a sensor's transfer behavior given uncertain input data. Hence, it belongs to the class of parameter estimation methods. On a high-level, the method works as shown in figure 3.1.1.

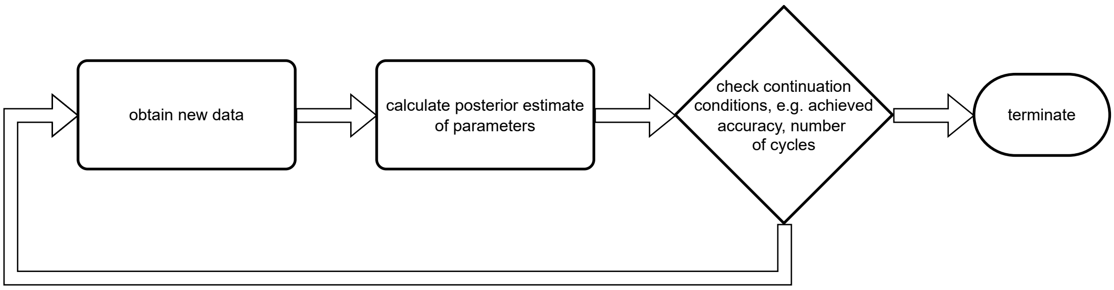

Figure 3.1.1: High-level overview of the proposed co-calibration method.

The method is supposed to have the following properties in order to fit the set context:

- online applicability
- robustness with regard to the number of input reference sensors
- robustness with regard to outliers
- robustness with regard to non-equidistant time-bases
- inclusion of pre-knowledge (by transferring it to prior knowledge, see chapter 3.3)
- traceable results

These requirements can be translated into the following development choices:

- interpolation resolves non-synchronous and non-equidistant time-bases [12, 15]
- sensor fusion allows consolidation of a varying amount of input reference sensors
- robust sensor fusion allows rejection of possible outliers [16, 17]
- working iteratively on blocks of available data and updating an internal abstract state of the method enables online capabilities [14]

[8]More details are provided in section 3.1.6

- Bayesian methods allow to include prior knowledge and quantify uncertain information
- uncertainty evaluation enables traceability

In figure 3.1.2, these guiding choices are brought together to visualize the input, output and internal structure of such a co-calibration method. The details are then given in the following sections.

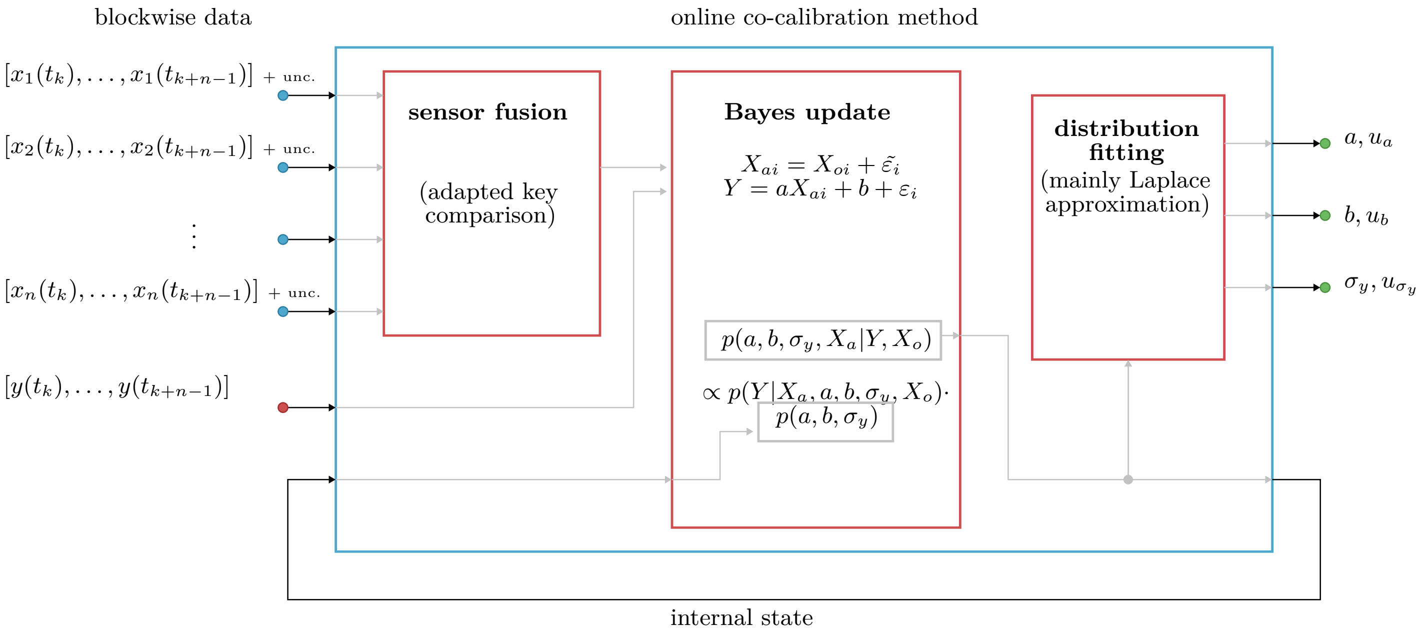

Figure 3.1.2: Outline of the mathematical method.

3.1.1 Input Structure

The online co-calibration method expects blockwise time-synced input data, i.e., an interpolation is usually required. Moreover, to maintain traceability after the interpolation to a common time-base, an uncertainty aware interpolation method needs to be chosen. Uncertainty propagation for various interpolation schemes are provided [10, 11] and have been implemented and applied by the author in [15].

Achieving online capable interpolation, however, is more involved. To achieve matching interpolation results between online and offline application, the interpolation influence needs to be locally bounded. This practically reduces the choice to zero- or first-order interpolation schemes. Smooth interpolation schemes like cubic-spline interpolation do not fulfill these criteria. Although the splines are themselves locally bounded, the calculation of their coefficients is an optimization over the whole available time-interval. Hence, smooth online interpolation can only estimate spline coefficients, e.g., using a Kalman filter [12].

After the interpolation, the blockwise data consist of n measurement values $x_j(t_i)$ (with $i = k, \dots, k+n-1$ and $j = 1, \dots, N$) of each of the N reference sensors and the indicated values $y(t_i)$ (with $i = k, \dots, k+n-1$) of the sensor to be co-calibrated. For each measurement value of the reference sensors a corresponding standard uncertainty $u(x_j(t_i))$ is known. Some of these readings can be empty, e.g., because of communication issues or the sample rate of a sensor being too low.

3.1.2 Sensor Fusion

The input reference sensor data $x_j(t_i)$ is fused into a single time series $x(t_i)$ by performing a sensor fusion at every available time t_i. The procedure is motivated by a structurally similar approach from the field of key comparison among national metrology institutes (NMIs) [101] and has been published by the author in [16, 17]. This procedure follows a frequentist approach to the sensor fusion task. Bayesian sensor fusion methods are described as well (e.g., [111, 112, 113, 114]), but were not employed in favour of the well established key-comparison-based approach. The main idea of the chosen procedure is to robustify an uncertainty-weighted mean with an uncertainty-aware outlier detection.

In a first step, the weighted mean $x_{\text{fusion}}(t_i)$ of all reference sensors with valid readings is calculated according to definition 19 and theorem 2 by setting the set of considered sensors to $C_S = \{1, \dots, N\}$. The weights γ_j are set to the inverse squared standard uncertainty, such that more certain values are weighted higher. In a second step, a χ^2-test for outlier detection is performed and the set of considered sensors C_S is adjusted as detailed in theorem 3. The chosen hypothesis allows more uncertain sensors to deviate further from the (assumed) value. If some $x_j(t_i)$ are identified as outliers by this test, $x_{\text{fusion}}(t_i)$ is recalculated with the updated C_S. If all $x_j(t_i)$ are identified as outliers by this test, $x_{\text{fusion}}(t_i)$ is set to the median of all $x_j(t_i)$.

Definition 19 (Sensor Fusion using Weighted Mean)**.** *The weighted mean of input values $x_j(t_i)$ with uncertainty $u(x_j(t_i))$ is*

$$x_{fusion}(t_i) = \frac{1}{k} \sum_{j \in C_S} \gamma_j x_j(t_i) \tag{3.1.1}$$

with weights

$$\gamma_j = \frac{1}{u(x_j(t_i))^2} \qquad\qquad k = \sum_{j \in C_S} \gamma_j \tag{3.1.2}$$

Theorem 2 (Uncertainty of Weighted Mean)**.** *The uncertainty of the fused value $x_{fusion}(t_i)$ calculated as in definition 19 is given by*

$$u(x_{fusion}(t_i))^2 = \sum_{j \in C_S} \left(\frac{\gamma_j}{k}\right)^2 u(x_j(t_i))^2 \tag{3.1.3}$$

Proof. Assuming that the $x_j(t_i)$ are not correlated, the uncertainty $u(x_{\text{fusion}}(t_i))$ of $x_{\text{fusion}}(t_i)$ is a direct application of the "law of propagation of uncertainty" given in the GUM [8]. □

Theorem 3 (Outlier Detection)**.** *Given the setting in definition 19, outliers in the input values $x_j(t_i)$ are detected using a χ^2-test with null hypothesis "The observed measurements come from normal distributions with mean $y_{fusion}(t_i)$ and variance $u(x_j(t_i))^2$.". The observed χ^2_{obs}-value is then calculated as*

$$\chi^2_{obs} = \sum_j \left(\frac{x_j(t_i) - x_{fusion}(t_i)}{u(x_j(t_i))}\right)^2 \tag{3.1.4}$$

$$p = 1 - F_{\chi^2, N-1}(\chi^2_{obs}) \tag{3.1.5}$$

with $F_{\chi^2,n}$ being the cumulative distribution function of a χ^2-distribution with n degrees of freedom. If $p < 0.05$, it is assumed that outliers are present in the data and set C_S of non-outlier values $x_j(t_i)$ is given by

$$C_S = \{j \quad if \quad |\Delta_j| \leq 2 * u_{\Delta_j}\} \tag{3.1.6}$$

with

$$\Delta_j = x_j(t_i) - x_{fusion}(t_i) \qquad\qquad u_{\Delta_j} = u(x_j(t_i))^2 - u(x_{fusion}(t_i))^2 \tag{3.1.7}$$

Remark: The hypothesis allows more uncertain sensors to deviate further from the (assumed) value.

Proof. This follows the approach presented in [16, 17] and follows the methodology of Cox [101]. □

3.1.3 Bayesian Update

The Bayesian update is the core of the co-calibration method. While the details and variants of it are described in the following subsections, the more general question "*Why is this providing traceable results?*" is of equal importance and answered in section 3.1.6.

3.1.3.1 Setup

Sensors with an ideal linear affine measurement function from measurand x to indication y are considered, as defined in definition 20. This corresponds to the most popular choice considered in the literature (see chapter 2.2) and allows to represent a first order Taylor approximation of the true transfer behavior. For sensors with such a transfer model, it is of interest to infer knowledge about the gain a and offset b based on (multiple) observations of x and y.

Definition 20 (Sensor Transfer Model)**.** *A linear affine transfer behavior is characterized by its gain a and offset b. Given some input $x(t)$, the output $y(t)$ is obtained by*

$$\begin{aligned} y(t) &= a \cdot x(t) + b \\ &= f(x(t), \underline{\theta}) \quad \textit{with} \quad \underline{\theta} = \begin{bmatrix} a \\ b \end{bmatrix} \end{aligned} \tag{3.1.8}$$

This adopts the nomenclature and notation of the GUM and VIM [3, 8].

The measurand x is not known directly, but only through observation by a reference sensor. This leads to a statistical model describing the relations between the measurand, indications of the reference sensors and indications of the device under test. The (fused) reference device does not provide the actual (true) value of the measurand X_{ai} but an observed value $X_{oi} = x_{\text{fusion}}(t_i)$ that differs by an (unknown) error term $\tilde{\varepsilon}_i$. Knowledge about the distribution of $\tilde{\varepsilon}_i$ can be derived from the uncertainties $u(t) = u(x_{\text{fusion}}(t))$ of the (fused) reference device. The indication Y_i of the DUT differs from the ideal model $f(X_{ai}, \underline{\theta})$ by an error ε_i. The distribution of ε_i is not fully known and needs to be identified as part of the calibration process. The statistical model is summarized by definition 21.

Definition 21 (Statistical Model of Sensor Indications)**.** *A statistical model for the described setting is given by [115]*

$$Y_i = f(X_{ai}, \underline{\theta}) + \varepsilon_i \tag{3.1.9}$$

$$X_{oi} = X_{ai} + \tilde{\varepsilon}_i \tag{3.1.10}$$

with (true) measurand values X_{ai}, device under test indications Y_i, reference sensor indications X_{oi} and device under test transfer behavior $f(X_{ai}, \underline{\theta})$ as in definition 20.

In this thesis, $\underline{\theta}$, $\underline{X}_a$ and σ_y are unknowns and the following assumptions on the distributions of the errors are made

$$\varepsilon_i \propto \mathcal{N}(\underline{0}, \sigma_y^2 \cdot I_n) \tag{3.1.11}$$

$$\tilde{\varepsilon}_i \propto \mathcal{N}(\underline{0}, \boldsymbol{U_x}) \tag{3.1.12}$$

$$\boldsymbol{U_x} = \begin{bmatrix} u^2(t_k) & & \\ & \ddots & \\ & & u^2(t_{k+n-1}) \end{bmatrix} \tag{3.1.13}$$

Sensor readings required for the co-calibration process become available in blocks of data Δ, which take the structure given in definition 22.

Definition 22 (Available Datapoints)**.** *Available datapoints take the form of $\underline{\delta}_i$ and the proposed method operates on a set Δ containing n datapoints $\underline{\delta}_i$ at a time.*

$$\begin{aligned}\underline{\delta}_i &= [t_i, y(t_i), x(t_i), u_x(t_i)]^T \\ &= [t_i, Y_i, X_{oi}, \sigma_{xi}]^T \qquad (3.1.14)\\ \Delta &= \{\underline{\delta}_k, \underline{\delta}_{k+1}, \dots, \underline{\delta}_{k+n-1}\} \qquad (3.1.15)\end{aligned}$$

The joint distribution of the mentioned variables is $p(\underline{\theta}, \underline{X}_a, \sigma_y, \underbrace{\underline{Y}, \underline{X}_o, \sigma_x}_{\Delta})$. The parameters $\underline{\theta}$, the actual measurand values $\underline{X}_a$ and σ_y are unknown. The main interest lies in inferring knowledge about these unknowns from the measurement data Δ via Bayesian inference. Two common solutions to this inference task are considered [116, 117, 118]. Method 1 (section 3.1.3.2) describes the details of an MCMC-based method and method 2 (section 3.1.3.3) evaluates the posterior on a discrete grid. In both approaches, the initial prior and the likelihoods are chosen in the same way.

The initial choice for the prior is given by definition 23. After that, the prior is set[9] to the posterior of the previous iteration. Note that, given the Bernstein-von Mises theorem, the influence of the initial prior diminishes given enough data [119].

Definition 23 (Informative Joint Prior Distribution)**.** *Assuming $(a, b, \sigma_y) \sim \mathcal{N}(\mu_a, \sigma_a^2) \times \mathcal{N}(\mu_b, \sigma_b^2) \times InverseGamma(\alpha, \beta)$ leads to the following PDF of the initial prior:*

$$\begin{aligned}p(\sigma_y, \underline{\theta}|\underline{X}_a) &= p_0(a, b, \sigma_y)\\ &= \frac{1}{\sqrt{2\pi\sigma_a^2}} \exp\left\{-\frac{1}{2\sigma_a^2}(a-\mu_a)^2\right\} \cdot \qquad (3.1.16)\\ &\quad \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left\{-\frac{1}{2\sigma_b^2}(b-\mu_b)^2\right\} \cdot\\ &\quad \frac{\beta^\alpha}{\Gamma(\alpha)} \sigma_y^{-\alpha-1} \exp\left\{-\frac{\beta}{\sigma_y}\right\}\end{aligned}$$

The likelihoods used in theorems 4 and 9 are given by definition 24.

Definition 24 (Likelihoods)**.** *Taking a Gaussian approach, the likelihoods used in equations* (3.1.20) *to* (3.1.22) *and* (3.1.28) *are given by:*

$$p(\underline{Y}|\underline{X}_a, \underline{\theta}, \sigma_y) \propto \frac{1}{\sqrt{(\sigma_y^2)^N}} \exp\left\{-\frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(Y_i - f(X_{ai}, \underline{\theta}))^2\right\} \qquad (3.1.17)$$

$$p(\underline{X}_a|\boldsymbol{U_x}, \underline{X}_o) \propto \frac{1}{\sqrt{|\boldsymbol{U_x}|}} \exp\left\{-\frac{1}{2}(\underline{X}_a - \underline{X}_o)^T \boldsymbol{U_x}^{-1}(\underline{X}_a - \underline{X}_o)\right\} \qquad (3.1.18)$$

3.1.3.2 Method 1: Posterior Evaluation using Block-Gibbs-Sampling

One solution to obtain an MCMC method for the inference on a, b, σ_y and $\underline{X}_a$ is given by theorem 4.

[9] Slight adjustments like a distribution fit onto the empirical Monte Carlo result or interpolation onto a new grid are used.

Theorem 4 (Equations of a Gibbs Sampler)**.** *The conditional for the posterior of the unknown variables [115]*

$$\underbrace{p(\underline{\theta}, \underline{X}_a | \underline{Y}, \underline{X}_o)}_{posterior} \propto \underbrace{p(\underline{Y} | \underline{X}_a, \underline{\theta}, \underline{X}_o)}_{likelihood} \underbrace{p(\underline{\theta}, \underline{X}_a | \underline{X}_o)}_{prior} \tag{3.1.19}$$

leads to the following conditionals of a Gibbs Sampler:

$$p(\underline{X}_a | \underline{\theta}, \sigma_y, \sigma_x, \underline{Y}, \underline{X}_o) \propto p(\underline{Y} | \underline{X}_a, \underline{\theta}, \sigma_y) p(\underline{X}_a | \sigma_x, \underline{X}_o) \tag{3.1.20}$$

$$p(\theta_j | \theta_{(j)}, \underline{X}_a, \sigma_y, \underline{Y}) \propto p(\underline{Y} | \underline{X}_a, \underline{\theta}, \sigma_y) \underbrace{p(\theta_j | \theta_{(j)}, \underline{X}_a, \sigma_y)}_{assume\ p(\theta_j)} \tag{3.1.21}$$

$$p(\sigma_y | \underline{\theta}, \underline{X}_a, \underline{Y}) \propto p(\underline{Y} | \underline{X}_a, \underline{\theta}, \sigma_y) \underbrace{p(\sigma_y | \underline{\theta}, \underline{X}_a)}_{assume\ p(\sigma_y)} \tag{3.1.22}$$

and defines an MCMC method.

Proof. Adaption of the equations provided by Dellaportas et al. [115] to the assumptions made in definitions 20 and 21. □

The Block-Gibbs-Sampling method requires a description of the posterior distribution of each unknown variable. Theorems 5 to 8 provide the explicit expressions for the posteriors given in theorem 4.

Theorem 5 (Explicit Posterior of $\underline{X}_a$)**.** *The explicit posterior of $\underline{X}_a$ is given by:*

$$p(\underline{X}_a | \underline{\theta}, \sigma_y, \sigma_x, \underline{Y}, \underline{X}_o) \propto \exp\left\{ -\frac{1}{2} (\underline{X}_a - \underline{M})^T \boldsymbol{V}^{-1} (\underline{X}_a - \underline{M}) \right\} \tag{3.1.23}$$

Which corresponds to a multivariate Gaussian $\mathcal{N}(\underline{M}, \boldsymbol{V})$ with

$$\underline{M} = \boldsymbol{V}(\boldsymbol{U_x}^{-1} \underline{X}_o + \underline{F}_2) \qquad \boldsymbol{F_1} = \frac{a^2}{\sigma_y^2} \cdot \boldsymbol{I_N}$$

$$\boldsymbol{V}^{-1} = (\boldsymbol{F_1} + \boldsymbol{U_x}^{-1}) \qquad \underline{F}_2 = \frac{a}{\sigma_y^2} \begin{bmatrix} Y_1 - b & \dots & Y_N - b \end{bmatrix}^T$$

Proof. Evaluation of equation (3.1.20) using the assumptions in definitions 20, 21 and 24 is given in appendix B.1.1. □

Theorem 6 (Explicit Posterior of a)**.** *The explicit posterior of a is given by:*

$$p(a | b, \underline{X}_a, \sigma_y, \underline{Y}) \propto \exp\left\{ A (a - \frac{B}{A})^2 \right\} \tag{3.1.24}$$

Which corresponds to a Gaussian $\mathcal{N}(\frac{B}{A}, -\frac{1}{2A})$ with

$$A = -\sum_{i=1}^{N} \frac{X_{ai}^2}{2\sigma_y^2} - \frac{1}{2\sigma_a^2} \qquad B = \sum_{i=1}^{N} \frac{(b - Y_i) X_{ai}}{2\sigma_y^2} - \frac{\mu_a}{2\sigma_a^2}$$

Proof. Evaluation of equation (3.1.21) with $\theta_i = a$ using the assumptions in definitions 20, 21 and 24 is given in appendix B.1.2. □

Theorem 7 (Explicit Posterior of b). *The explicit posterior of a is given by:*

$$p(b|a, \underline{X}_a, \sigma_y, \underline{Y}) \propto \exp\left\{A(a - \frac{B}{A})^2\right\} \tag{3.1.25}$$

Which corresponds to a Gaussian $\mathcal{N}(\frac{B}{A}, -\frac{1}{2A})$ *with*

$$A = -\frac{N}{2\sigma_y^2} - \frac{1}{2\sigma_b^2} \qquad\qquad B = \sum_{i=1}^{N} \frac{aX_{ai} - Y_i}{2\sigma_y^2} - \frac{\mu_b}{2\sigma_b^2}$$

Proof. Evaluation of equation (3.1.21) with $\theta_i = b$ using the assumptions in definitions 20, 21 and 24 is given in appendix B.1.3. □

Theorem 8 (Explicit Posterior of σ_y). *The explicit posterior of σ_y is given by:*

$$p(\sigma_y|\underline{\theta}, \underline{X}_a, \underline{Y}) \propto \exp\left\{-N\ln(|\sigma_y|) - \tilde{A}\frac{1}{\sigma_y^2} - (\alpha+1)\ln(\sigma_y - \gamma) - \frac{\beta}{\sigma_y - \gamma}\right\} \tag{3.1.26}$$

with

$$\tilde{A} = \frac{1}{2}\sum_{i=1}^{N}(Y_i - aX_{ai} - b)^2$$

Proof. Evaluation of equation (3.1.22) using the assumptions in definitions 20, 21 and 24 is given in appendix B.1.4. □

While drawing samples from the Gaussian posterior distributions of $\underline{X}_a$, a and b is typically supported by numerical toolboxes, drawing from the posterior distribution of σ_y is not. To sample from this non-classic distribution, a sample u is drawn from a uniform distribution $\mathcal{U}(0,1)$ (on the unit interval). The following expression then leads to a corresponding sample of the posterior distribution of σ_y and can be evaluated using numerical integration in conjunction with numerical optimization:

$$\tilde{\sigma}_y = \arg\min_x \| \int_{-\infty}^{x} p(\sigma_y|\underline{\theta}, \underline{X}_a, \underline{Y})d\sigma_y - u\| \tag{3.1.27}$$

In the later implementation of method 1, two simplifications to the underlying data model (definition 21) are investigated: (1) deterministic reference sensor readings and (2) known variance of the sensor to be co-calibrated. A deterministic reference sensor can be specified by setting $\underline{X}_a = \underline{X}_o$. Although this reduces the number of variables drastically, it does only affect the computational load to a small extend, as drawing samples from a multivariate Gaussian distribution is efficient. A known variance can be specified by setting $\sigma_y \equiv$ known constant. This reduces the computational load drastically, as drawing samples from the distribution of σ_y is quite involved.

3.1.3.3 Method 2: Posterior Evaluation on a Discrete Grid

Instead of using Gibbs sampling (as a variant of MCM), the joint marginalized posterior of the sought parameters a, b and σ_y can also be evaluated directly. The calculation are executed numerically on a discrete (hyper)cube. Theorem 9 provides an expression for a joint posterior $p(\sigma_y, \underline{\theta}|\underline{Y}, \underline{X}_o)$ of the unknowns with the measurand values being marginalized.

Theorem 9 (Marginalization over $\underline{X}_a$)**.** *The marginalized joint posterior is expressed by*

$$p(\sigma_y, \underline{\theta}|\underline{Y}, \underline{X}_o) \propto p(\sigma_y, \underline{\theta}) \cdot \exp\left\{-\frac{1}{2}(\underline{X}_o{}^T \boldsymbol{U_x}{}^{-1}\underline{X}_o - \underline{M}^T \boldsymbol{V}^{-1}\underline{M} + F_3)\right\} \cdot \frac{\sqrt{|\boldsymbol{V}|}}{|\sigma_y|^N} \tag{3.1.28}$$

with

$$\begin{aligned} \underline{M} &= \boldsymbol{V}(\boldsymbol{U_x}{}^{-1}\underline{X}_o + \underline{F}_2) & \qquad \boldsymbol{F_1} &= \boldsymbol{G_1}{}^T\boldsymbol{G_1} \\ \boldsymbol{V}^{-1} &= (\boldsymbol{F_1} + \boldsymbol{U_x}{}^{-1}) & \underline{F}_2 &= \boldsymbol{G_1}\underline{G}_2 \\ \boldsymbol{G_1} &= \frac{a}{\sigma_y} \cdot \boldsymbol{I_N} & F_3 &= \underline{G}_2{}^T\underline{G}_2 \\ \underline{G}_2 &= \frac{1}{\sigma_y}\begin{bmatrix} Y_1 - b & \dots & Y_N - b\end{bmatrix}^T & & \end{aligned}$$

Proof. The marginalization is written out as integration over $\underline{X}_a$. Then Bayes theorem (see theorem 1) is used to obtain a description of the joint posterior distribution $p(\sigma_y, \underline{\theta}|\underline{Y}, \underline{X}_o)$ with marginalized $\underline{X}_a$. The calculation steps are given in appendix B.2.1. □

Theorem 9 can then be used to implement a numerical update scheme. The sought joint distribution depends on three variables and is discretized on a rectilinear grid. Two different grid behaviors are used: a static grid following definition 25 and an adaptive grid that focuses the grid around the region of highest posterior density as given in definition 26. The resolution of the static grid should be chosen not too broad (meaningfulness of result), but also not too fine (computational cost). The adaptive grid avoids these problems and allows to use a computationally lighter grid that still achieves a fine resolution after some iterations. To avoid numerical overflow, the calculations and storage of the discrete distribution is done using the logarithm of the above mentioned probability distributions and likelihoods (except during integration).

Definition 25 (Static Grid)**.** *The static grid is given by a cube spanning*

$$[\mu_a - 3 * \sigma_a, \mu_a + 3 * \sigma_a]_{lin} \tag{3.1.29}$$

$$\times[\mu_b - 3 * \sigma_b, \mu_b + 3 * \sigma_b]_{lin} \tag{3.1.30}$$

$$\times[1e - 6, 1e2]_{log} \tag{3.1.31}$$

with "lin" and "log" referring to linear and logarithmic grid spacing respectively. Note: Because the grid remains the same for all iterations, the final position of the maximum a-posteriori probability (MAP) should lie within the bounds of the grid.

Definition 26 (Adaptive Grid)**.** *The initial adaptive grid is identical to definition 25. Given a posterior distribution the grid is adjusted to cover a (hyperrectangular) box where the logarithm of the posterior is greater than a threshold* τ*. To allow both contraction and expansion of the parameter space (and with that prevent early convergence), the minimal bounding box matching this criteria is expanded proportionally to the box dimensions by a "zoom out"-factor* z_{out}*. The default values are* $\tau = -1000$ *and* $z_{out} = 0.2$*. The posterior is then transferred to the new grid using a multidimensional linear interpolation.*

3.1.4 Internal State

The Block-Gibbs-Method uses parametric distributions that are fitted against the drawn samples of the posterior distribution. Due to the chosen parametric models, potential correlation between the parameters are lost. As indicated by the structure of the posterior terms, a Gaussian PDF is used to fit parametric models for a and b. The distribution for σ_y is approximated as inverse gamma distribution, to match the assumption in the next cycle's update equations.

The discrete state method stores the full hypercube as its internal state. Therefore, full correlation information in the posterior is maintained across update cycles and - apart from the chosen discretization - no parametric models are enforced in the posterior.

3.1.5 Distribution Fitting for Result Communication

To obtain parameter estimates for later use and document the current state of the method, a Laplace approximation around the maximum a posteriori (MAP) estimate of the internal state is calculated. The fitting starts by obtaining the marginalized posterior PDF of the corresponding parameter (either from its formula or via numerical integration of unused axis (trapz)). A cubic spline is fitted to the logarithmic PDF. The maximum of the spline is calculated and taken as the MAP. Moreover, the second order derivative of the spline at the position of the maximum is calculated to obtain an uncertainty measure of it. If not enough data points for fitting a spline are available, the method falls back to the maximum as MAP and grid spacing as uncertainty estimate.

3.1.6 Traceability of the Parameter Estimate

As quoted in definition 8, traceability is achieved by a chain of calibrations starting at a primary reference. In order to continue the traceability chain, the proposed co-calibration method needs to match the definition of "calibration" with respect to some agreed standard.

Bayesian uncertainty analysis (as employed in section 3.1.3) can be made equivalent to an uncertainty evaluation as defined in the GUM [8, 96]. However, this requires a very specific (non-informative) selection of the prior distribution for the sought parameters [120]. The prior distributions used in the proposed co-calibration do *not* fulfill these specific requirements by design, but use informative priors based on knowledge from the semantic sensor descriptions (see chapter 3.3). Strictly speaking, the presented co-calibration approach therefore is not evaluating the uncertainty in a GUM-compatible approach.[10] However, it is in the spirit of a GUM-7 (GUM S1) Type B uncertainty evaluation, which employs a Bayesian approach and allows for subjective priors. Even more, the proposed co-calibration method does hold up to the definition of a calibration in the VIM, which is summarized as theorem 10.

Theorem 10 (Traceability of the Proposed Co-Calibration)**.** *The proposed co-calibration method summarized by figure 3.1.2 continues the traceability chain by matching the definition of a calibration as specified by the VIM (see definition 5).*

Proof. In order to justify the compliance of the proposed co-calibration method with the definition of a calibration given in the VIM, all specified requirements are mapped to the related aspects of the developed method. According to [3, VIM 2.39], a calibration is an

- operation that, under specified conditions, in a first step,
- establishes a relation between
  - → established by equation (3.1.9)
- the quantity values with measurement uncertainties provided by measurement standards
  - → X_{oi} are obtained from a GUM-compliant sensor fusion of calibrated sensors, see section 3.1.2
- and corresponding indications
  - → Y_i, see equation (3.1.9)
- with associated measurement uncertainties
  - → equation (3.1.9) allows to provide a GUM-compliant uncertainty value for Y_i based on the estimated a, u_a, b, u_b and σ_y as well as X_{oi} and $u(X_{oi})$
- and, in a second step, uses this information to establish a relation for obtaining a measurement result from an indication
  - → an inverse model allows to obtain an estimate of X_{ai} based on Y_i (see theorem 18)

□

3.1.7 Application to the Exemplary Use Case

In the sensor network presented in chapter 1.3, it is of interest to co-calibrate the accelerometer S6 from the measurements provided by the acceleration sensors S1 and S2. The measurand, measurements taken by S1, measurements taken by S2 and the indications of S6 are shown in figure 3.1.3. In this setup, it is assumed that the data is already available as blockwise data. The blockwise structure can be observed in the zoomed plot in figure 3.1.4. Five variants of the proposed co-calibration method are listed in table 3.1.1 and executed on the same incoming data stream. The resulting estimates for a, b and σ_y are shown in figure 3.1.5. The standard uncertainty of these parameter estimates can be seen as "tube" in figure 3.1.5 or separately in figure 3.1.6. Table 3.1.2 summarizes the results of the runs by stating numerical values for the estimates after $n = 399, 599, 1999$ datapoints. As the data is simulated, the true values of the parameters are known.

[10] This also implies that the proposed co-calibration method is (in strict manner) not traceable with respect to the ISO17025 standard [4], as it states that the “measurement uncertainty [...] is evaluated according to agreed methods” and especially references the GUM [8] in that context.

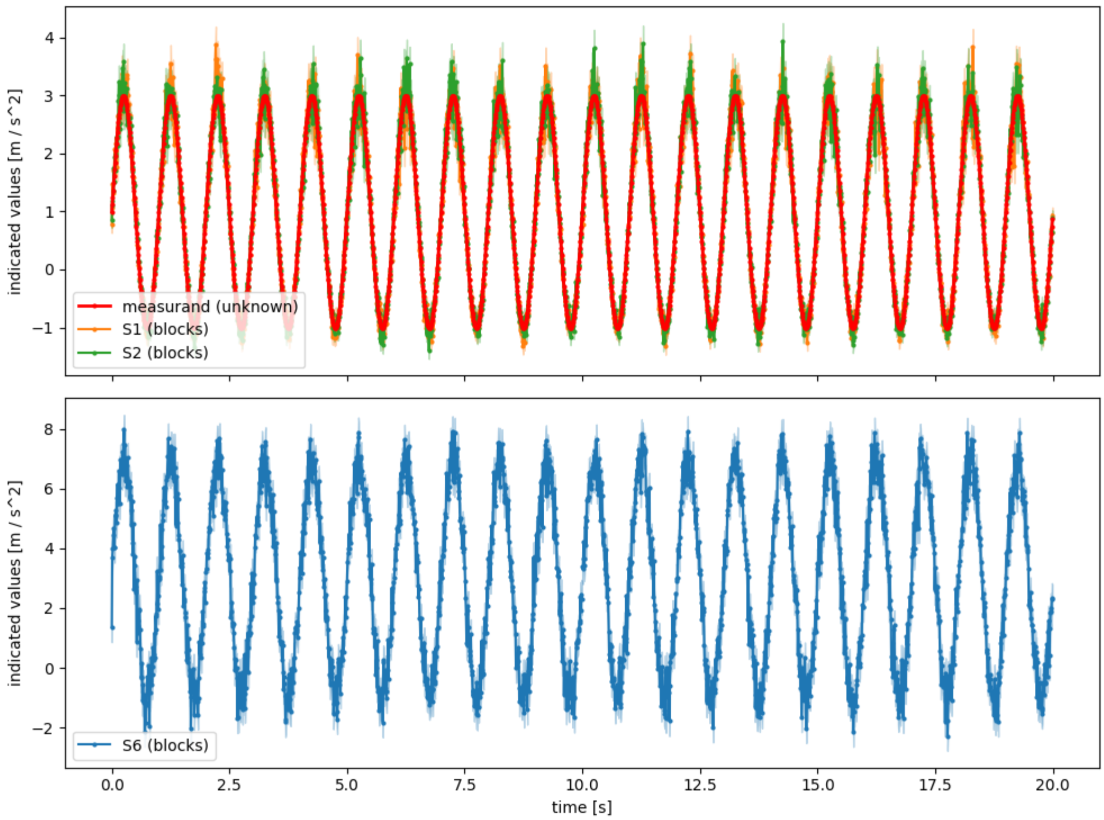

Figure 3.1.3: Simulated blockwise data of the measurand, measurement data of sensor 1 (S1), measurement data of sensor 2 (S2) and the indicated values by sensor 6 (S6).

name	**refers to**
`gibbs_minimal`	method 1
`gibbs_known_sigma_y`	method 1, but σ_y is not sampled
`gibbs_no_EIV`	method 1, but no error-in-variables model is assumed, therefore $\underline{X}_a$ is not sampled
`joint_posterior`	method 2, with fixed discrete grid
`joint_posterior_agrid`	method 2, with auto-adjusting discrete grid

Table 3.1.1: Overview of used method names and what method this name refers to.

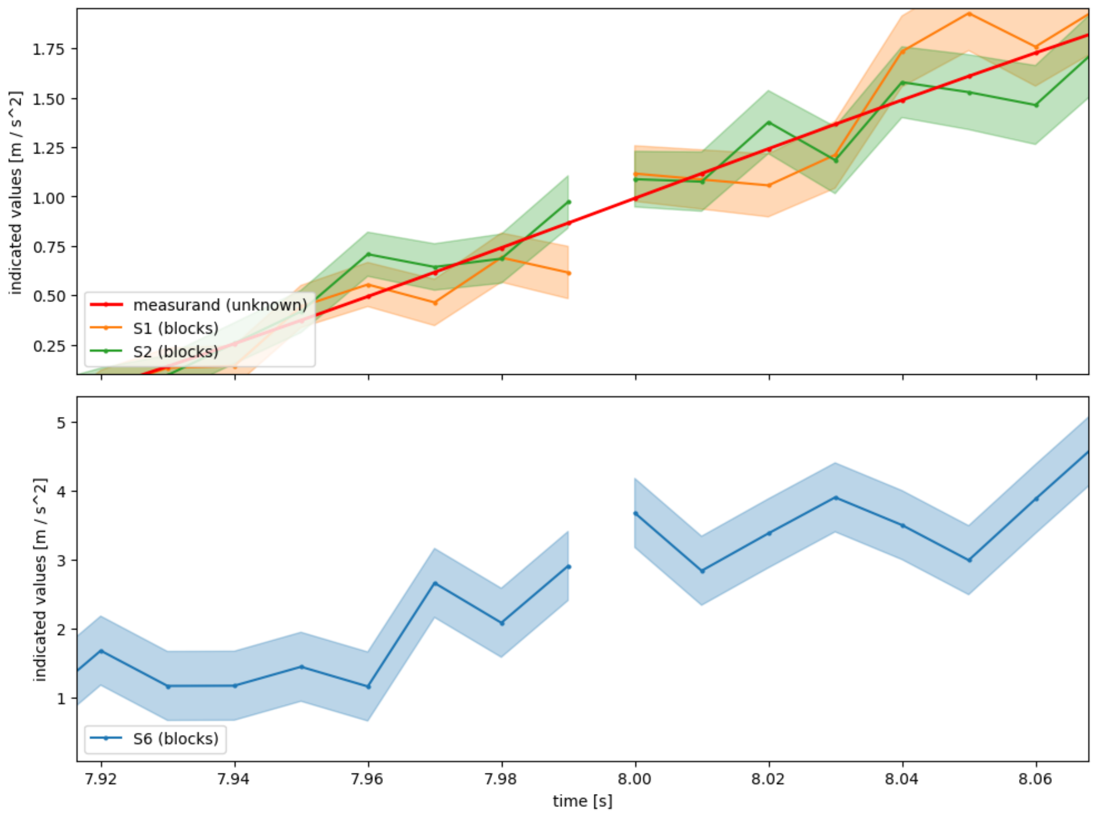

Figure 3.1.4: Detail of the blockwise structure. Zoomed-in version of figure 3.1.3, shows the blockwise data of measurand (simulation), measurement data of sensor 1 (S1), measurement data of sensor 2 (S2) and the indicated values by sensor 6 (S6).

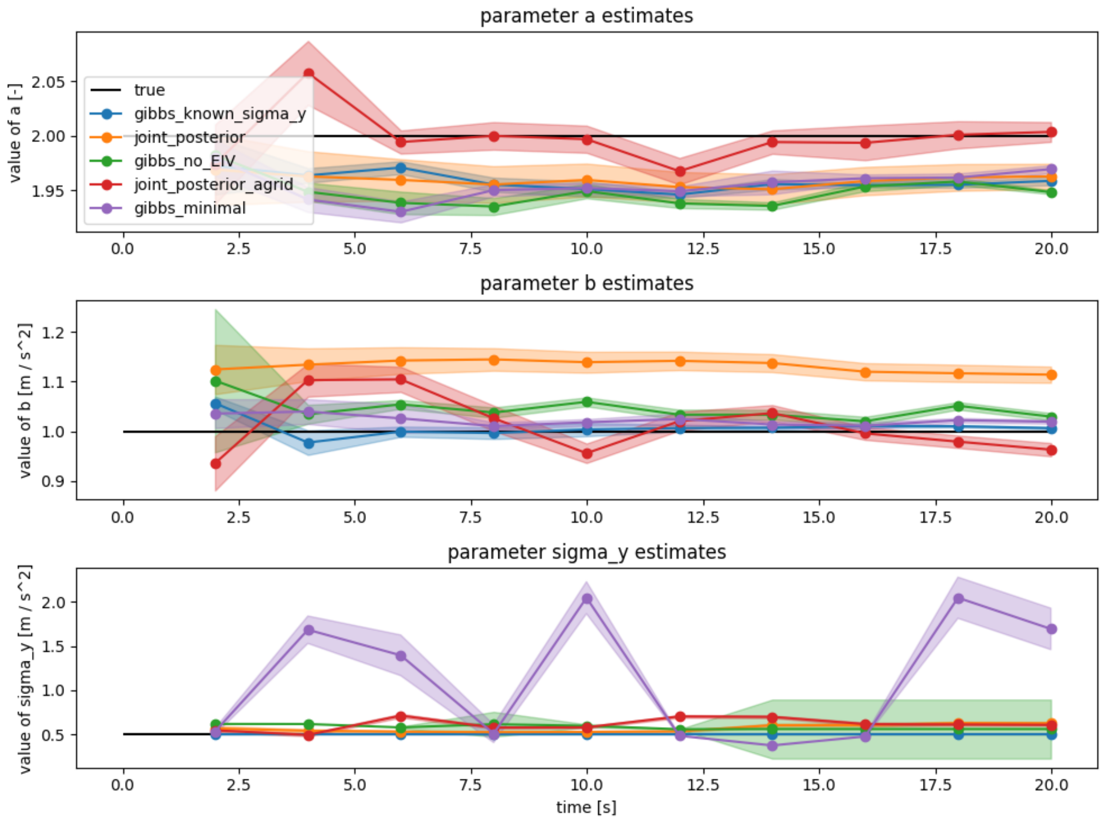

Figure 3.1.5: Estimated parameter values $\hat{a}$, $\hat{b}$ and $\hat{\sigma_y}$ of four different co-calibration methods after each processed block. The transparent bands indicates a region of one standard uncertainty above and below the estimate ($\pm u$). True values of each parameter used in the simulation are given by horizontal black lines.

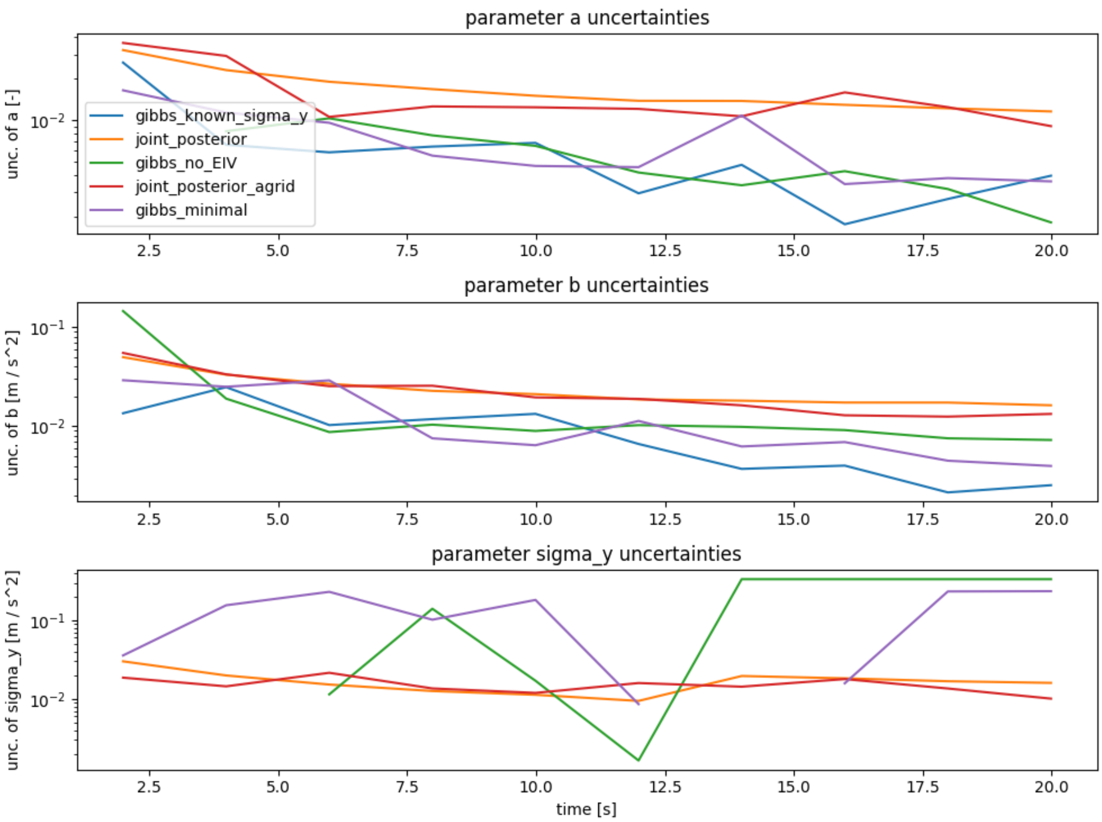

Figure 3.1.6: Estimated parameter uncertainties $u_{\hat{a}}$, $u_{\hat{b}}$ and $u_{\hat{\sigma_y}}$ of of four different co-calibration methods after each processed block.

method	parameter estimate with n datapoints	$a \pm u_a$	$b \pm u_b$	$\sigma_y \pm u_\sigma$
true		2.00e+00	1.00e+00	5.00e-01
`gibbs_known_sigma_y`	399	1.96e+00 ± 6.65e-03	9.77e-01 ± 2.47e-02	5.00e-01 ± 0.00e+00
	599	1.97e+00 ± 5.86e-03	9.99e-01 ± 1.03e-02	5.00e-01 ± 0.00e+00
	1999	1.96e+00 ± 3.97e-03	1.01e+00 ± 2.55e-03	5.00e-01 ± 0.00e+00
`gibbs_minimal`	399	1.94e+00 ± 1.14e-02	1.04e+00 ± 2.50e-02	1.69e+00 ± 1.55e-01
	599	1.93e+00 ± 9.60e-03	1.03e+00 ± 2.90e-02	1.40e+00 ± 2.30e-01
	1999	1.97e+00 ± 3.61e-03	1.02e+00 ± 3.99e-03	1.69e+00 ± 2.34e-01
`gibbs_no_EIV`	399	1.95e+00 ± 8.34e-03	1.03e+00 ± 1.90e-02	6.19e-01 ± nan
	599	1.94e+00 ± 1.03e-02	1.05e+00 ± 8.75e-03	5.80e-01 ± 1.14e-02
	1999	1.95e+00 ± 1.82e-03	1.03e+00 ± 7.28e-03	5.62e-01 ± 3.33e-01
`joint_posterior`	399	1.96e+00 ± 2.31e-02	1.13e+00 ± 3.30e-02	5.42e-01 ± 1.98e-02
	599	1.96e+00 ± 1.90e-02	1.14e+00 ± 2.68e-02	5.32e-01 ± 1.52e-02
	1999	1.96e+00 ± 1.16e-02	1.11e+00 ± 1.63e-02	6.28e-01 ± 1.59e-02
`joint_posterior_agrid`	399	2.06e+00 ± 2.93e-02	1.10e+00 ± 3.33e-02	4.96e-01 ± 1.44e-02
	599	1.99e+00 ± 1.06e-02	1.10e+00 ± 2.53e-02	7.13e-01 ± 2.14e-02
	1999	2.00e+00 ± 9.08e-03	9.63e-01 ± 1.33e-02	6.09e-01 ± 1.00e-02

Table 3.1.2: Summary of exemplary simulation results.

Chapter Summary

A consensus based co-calibration method is developed. In a first step, the general requirements, main functional components and their connections are outlined. The mathematical details of the individual blocks are then given by stating the structure of the input, providing the equations for a robust fusion of multiple reference sensors, detailing the formulas and numerical schemes for the Bayesian parameter estimation and explaining how a parameter estimate with uncertainty quantification is obtained from the internal representation. Moreover, the traceability of the method is discussed and multiple variants of the proposed update scheme are applied the sensor network previously presented in chapter 1.3.

3.2 Semantic Representation of Metrological Sensor Networks

It is of interest to represent metrological core knowledge such as calibration information, measurand specification and measurement uncertainty alongside the general semantic description of the sensors in a sensor network. This knowledge will then be used in chapter 3.3 to support and initialize the co-calibration method presented in the previous chapter 3.1. To obtain such a representation, the approach is to set the level of detail required, identify the overlap with existing knowledge representations and propose minimal extensions where necessary. Major parts of this work have been published in [18] with additions in [19], but some definitions have been adjusted.

3.2.1 Representation Requirements

The purpose of this representation is to support a co-calibration routine in sensor networks. The requirements of an ontology are typically formulated in terms of competency questions [121]. In the context of this thesis, the following competency questions are of special importance:

- Is a specific sensor currently calibrated?
- What sensors measure the same quantity?
- Which sensors are located at the same place?
- Which sensors can be used to co-calibrate a specific sensor?

Answering these questions requires to capture information about the individual sensors in the network, as well as their arrangement in the network. Following a decentralized approach, the knowledge of the sensor network is not represented explicitly, but implicitly by its constituting sensors. Requirements are therefore posed in terms of the description of individual sensors.

Addressing and identifying sensors in the network requires their (locally) unique identifiers, manufacturer information, measurand and location. The latter two are also necessary to derive the relevant topology of the sensor network. The co-calibration requires information about the validity ranges (measurand- and date-wise), the mathematical transfer behavior and traceable measurements of the sensors.

A summary of the aspects required to describe metrological use cases are provided in table 3.2.1 and figure 3.2.1.

Aspect	Detail	Covered by (proposal)
general description	identifier / name	SOSA
	manufacturer / production date	SOSA
	measurement principle	SOSA
	measurement quantity	OM
location	general concept	extension
	geometric	GeoSPARQL
	topological (hierarchical, neighbors)	GeoSPARQL, SOSA
calibration	general, date, method	XML schema datatypes
	range of validity	SSN
	model type / equation	MathML + extension
	parameter uncertainty	OM + D-SI
	accuracy, sensitivity, precision	SSN
observation result	time	SOSA
	value	OM
	uncertainty	D-SI
	unit (SI)	OM

Table 3.2.1: Overview of requirements and proposed coverage.

3.2.2 Merge of Existing Knowledge Representations

None of the knowledge frameworks mentioned in section 2.2.2 cover all required aspects. Moreover, the reviewed schemes overlap in certain aspects, especially for units and quantities. In the following, a sensible combination of relevant ontologies that each provide certain aspects of the required functionality is discussed.

Subtle but important differences between OM and QUDT are stated in [86] and [122]. Recent updates of the QUDT ontology have strengthened its expressiveness.[11] Now, both (OM and QUDT) link (sub)multiples of units to their constituent prefix and base unit, which is in line with the D-SI syntax specification for unit terms [31]. Still, OM provides a higher level of granularity in its concepts and relations and is therefore favored.

General aspects of sensors and sensor networks can be suitably and adequately captured using the SOSA and SSN ontologies. Location information can be represented using GeoSPARQL, other ontologies listed by the W3C in [123] and custom extensions.

By translating ideas from EngMath into a combination of OM and Content MathML, calibration information can be represented. If available, the required information about the sensor can be extracted from a DCC. If calibration information is available for a sensor (`sosa:Sensor`), it becomes an instance of the (new) `scal:CalibratedSensor` class. Additional concepts like system accuracy, precision, property and capability are already available within the module SSN-System, and each sensor can be linked to those.

[11]There is no recent publication highlighting these changes, but `https://github.com/qudt/qudt-public-repo/graphs/contributors` shows increased activity in the source code of QUDT since 2019.

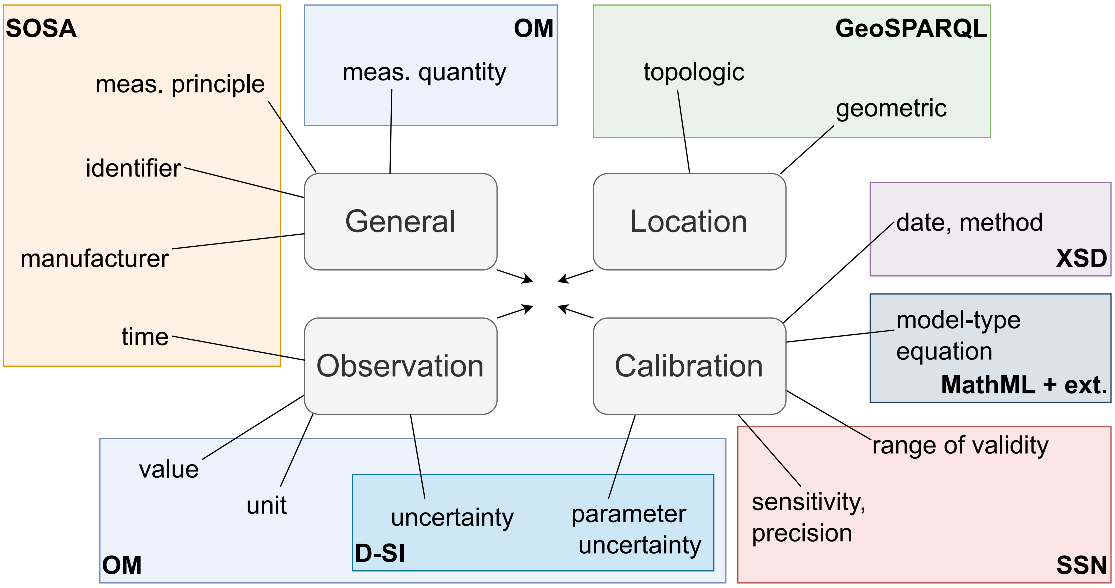

Figure 3.2.1: Required information and coverage by existing knowledge representations.

By combining time aspects from SOSA (`sosa:phenomenonTime`, `sosa:resultTime`), physical quantity kind (`om:Quantity`) and units (`om:Unit`) from OM, observations and results (`sosa:Observation`, `sosa:Result`) will be represented. To quantify the uncertainties of values, the `om:Measure` description needs to be extended. This is achieved by introducing a subclass (`scal:MeasureWithUncertainty`) which adds metrological statements following the D-SI data model. This subclass can also be used to represent parameters with uncertainties and therefore allows to capture calibration model descriptions in the same way as in the DCC.

A manual merging approach is chosen, as not all selected knowledge sources are available as ontology (D-SI) or in a common format (EngMath). Figure 3.2.1 and the last column of table 3.2.1 summarize the proposed merge. Although many requirements are already covered using existing frameworks, some new core concepts need to be defined. This is done within two new ontologies that provide the central parts interconnecting all properties from above. These ontologies are presented in the following section.

3.2.3 Extensions to the Proposed Merge

The mentioned requirements lead to the design of two distinct but closely linked ontologies.

3.2.3.1 `scal` Ontology

The `scal` ontology provides concepts relevant for the (co-)calibration of sensors that are part of a sensor network. It is mainly an extension of the SOSA/SSN ontology to cover metrological use cases.

The main contributions are

- `scal:CalibratedSensor` a sensor with calibration information
- `scal:CalibrationModel` to describe the sensor's transfer behavior
- `scal:Location` to describe the sensors position
- `scal:EquationModel` as a specific subclass of `scal:CalibrationModel`
- `scal:MeasureWithUncertainty` to communicate full metrological detail of measurement results

The SOSA ontology allows a `sosa:Sensor` to have (multiple) `ssn:Property`. A new (sub-) property `scal:CalibrationModel` is introduced to provide a generic handle to attach calibration information to a sensors description. Every sensor that has a `scal:CalibrationModel` becomes a `scal:CalibratedSensor`. The relations are visualized in figure 3.2.2. By introducing the concept of a calibration model, fundamental metrological information can be attached to a sensor's description.

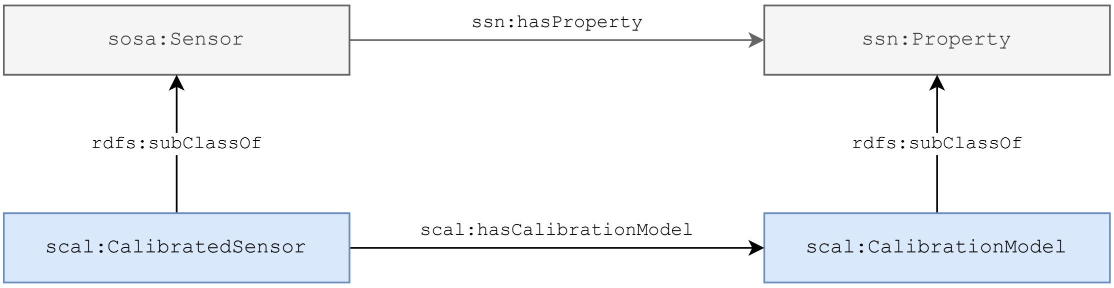

Figure 3.2.2: Details of `scal:CalibrationModel` and relations to other ontologies.

Positional information about a `sosa:Sensor` can be provided by the `sosa:isHostedBy` property which links a `sosa:Platform`. Although this allows to describe typical sensor mounting schemes in industrial plants by defining the hierarchy between platforms, other geometries and topologies are not covered [83]. Therefore a new `scal:Location` is introduced which is a subclass of `sosa:Platform` to maintain compatibility. Moreover, `scal:Location` also inherits from `geo:-SpatialObject` to open up location description vocabulary and evaluation methods provided by GeoSPARQL [123]. This enables the description of topological and geometrical spatial relations, e.g., according to the WGS84 standard [124]. The relations are shown in figure 3.2.3.

Figure 3.2.3: Details of `scal:Location` and relations to other ontologies.

Often, a calibration model is expressed as mathematical formula. Therefore, an `scal:EquationModel` is introduced as a possible representation of calibration information, which draws some inspiration from the conceptualization of the EngMath ontology [90]. The `scal:EquationModel` is specified by one `scal:Equation`, at least one `scal:Variable` and some `scal:Parameters`. The distinguishing aspect between parameters and variables is the availability of explicit numeric knowledge. Variables are provided or sought for during runtime, whereas parameters are numerically known (up to uncertainty) in advance. An `scal:Equation` is represented using a Content-MathML string, which captures the semantic structure of the equation. Because a `scal:Variable` is inheriting features of `om:Quantity`, consistency checks of the equation are enabled. `scal:Parameter` is a subclass of `om:Measure` allowing to provide a numerical value, a physical unit and a corresponding quantity kind - this also includes a `scal:MeasureWithUncertainty` (see next paragraph). The relations are shown in figure 3.2.4

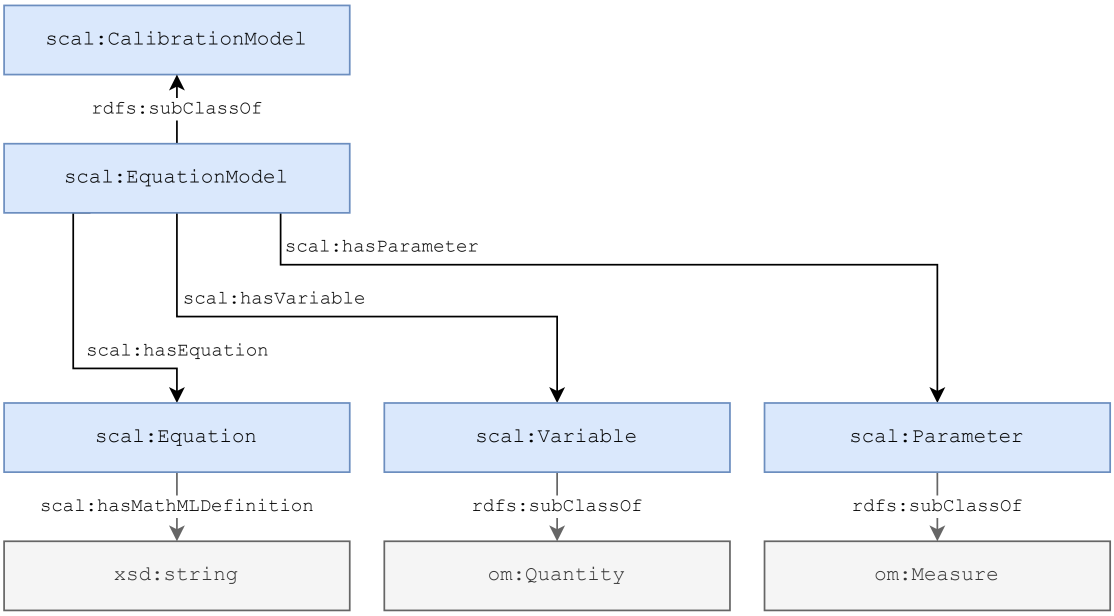

Figure 3.2.4: Details of `scal:EquationModel` and relations to other ontologies.

In SOSA a `sosa:Sensor` can make an `sosa:Observation`. To attach a metrology-aware `sosa:Result` to a given `sosa:Observation` the concept of a `scal:MeasureWithUncertainty` is introduced. It inherits from `sosa:Result` as well as `om:Measure` and extends it with the `scal:hasUncertainty` property. The relations are shown in figure 3.2.5.

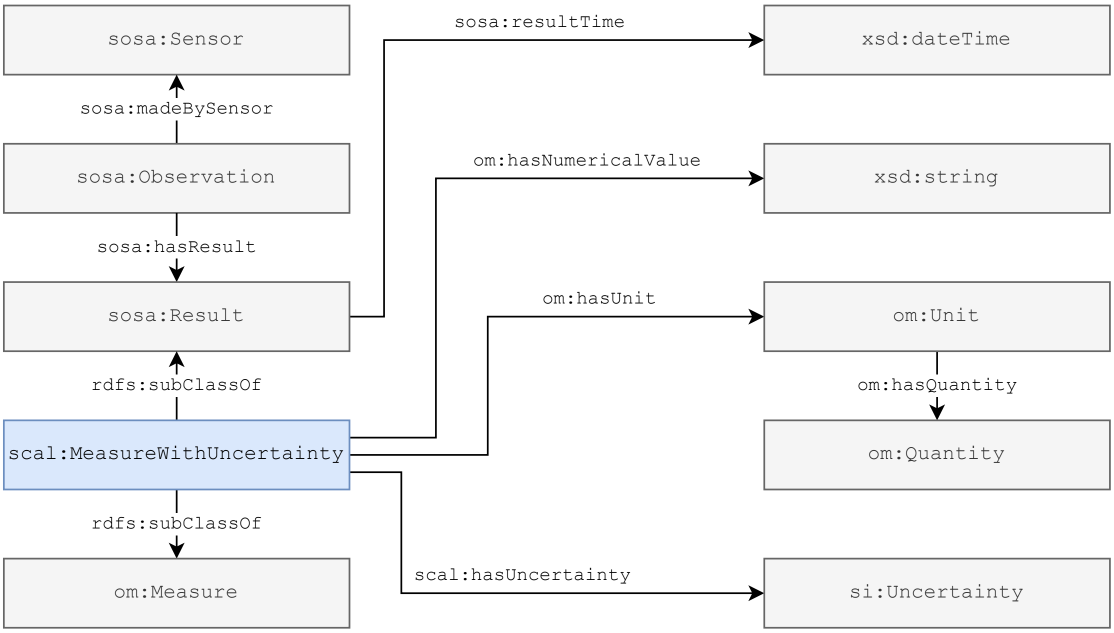

Figure 3.2.5: Details of `scal:MeasureWithUncertainty` and relations to other ontologies.

3.2.3.2 `trans` Ontology

Signal processing often uses rather involved (e.g., dynamic, non-linear) transfer behavior. A representation of this as MathML-strings is (at least in full detail) not feasible and it is beneficial to refer to transfer behavior concepts within MathML by providing links to semantically defined concepts. Therefore, the ideas of the `scal` ontology are extended to provide common representations of the mathematical description of the transfer behavior which can be used as `scal:CalibrationModel`. This leads to the `trans` ontology which enables representation of common signal processing transfer behaviors. It is conceptualized in a joint publication together with Vedurmudi et al. in [19].

The `trans` ontology introduces two main new concepts: an abstract transfer model and a mathematical representation for such models. With that, each `trans:TransferModel` `trans:isExpressedBy` a `trans:MathematicalObject`. The relations to other ontologies and some further distinguishing properties of practical relevance are shown in figure 3.2.6.

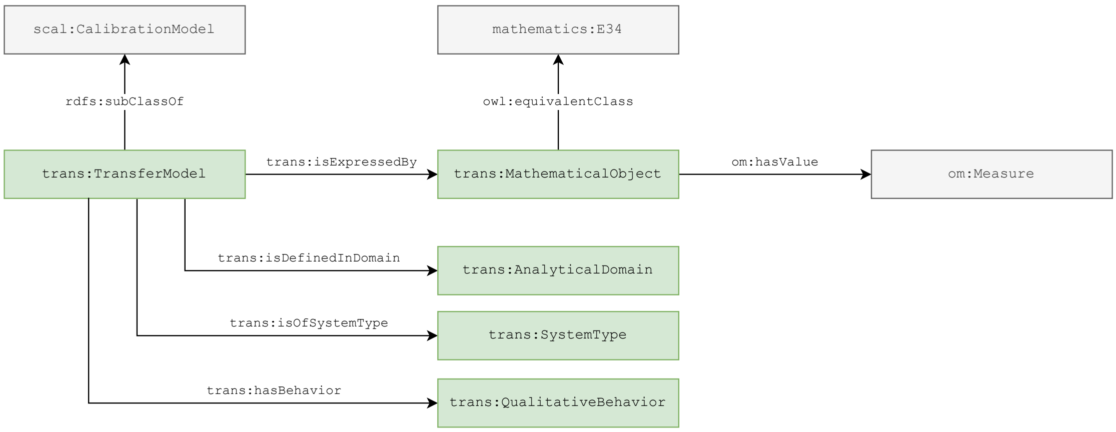

Figure 3.2.6: Overview of the `trans` ontology and relations to other ontologies.

The `trans:TransferModel` is a super-class to many transfer behaviors common in signal processing. Available sub concepts, classification and which mathematical objects are used for their representation are shown in figure 3.2.7.

Mathematical objects are often numerically represented using (multidimensional) arrays. However, the interpretation of the values in these arrays is highly dependent on the chosen mathematical object. Therefore, common mathematical objects in signal processing are provided and linked to an array representation in figure 3.2.8.

Signals and systems are often classified with regard to (e.g.) domain, type or (qualitative) behavior. To represent these classifications within the ontology, `trans:TransferModel` adds additional properties as seen in figure 3.2.6. Further sub-concepts of these properties are shown in figure 3.2.9.

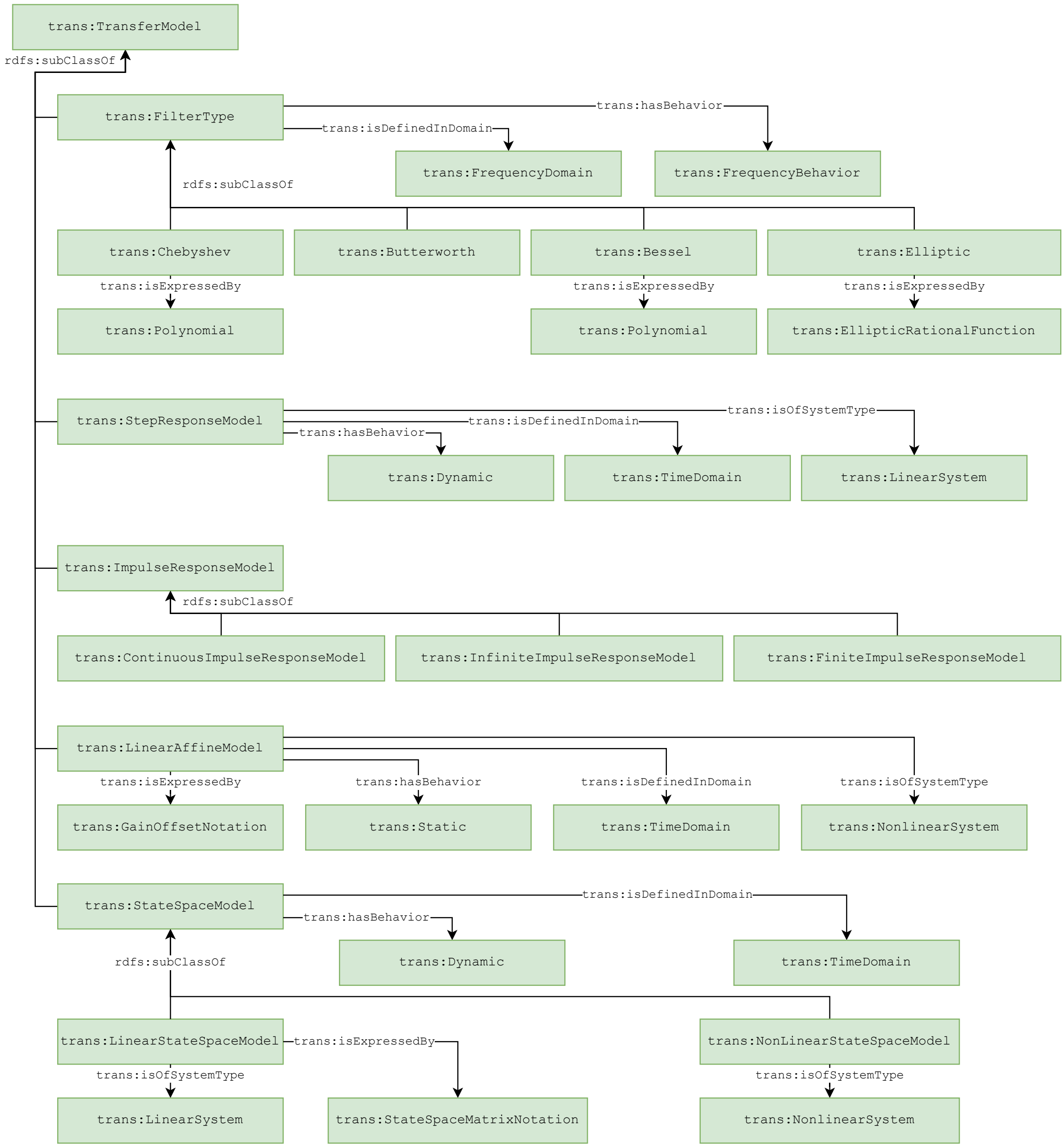

Figure 3.2.7: Details of `trans:TransferModel` and relations to other ontologies.

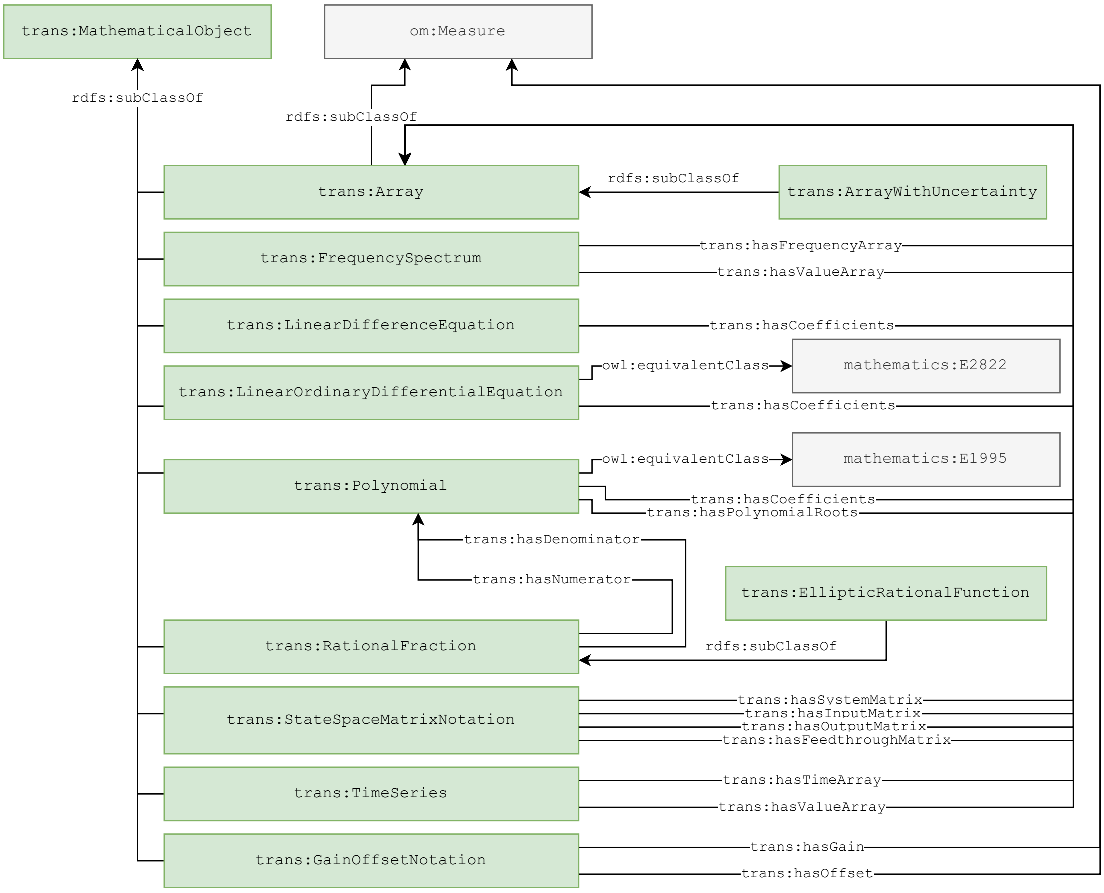

Figure 3.2.8: Details of `trans:MathematicalObject` and relations to other ontologies.

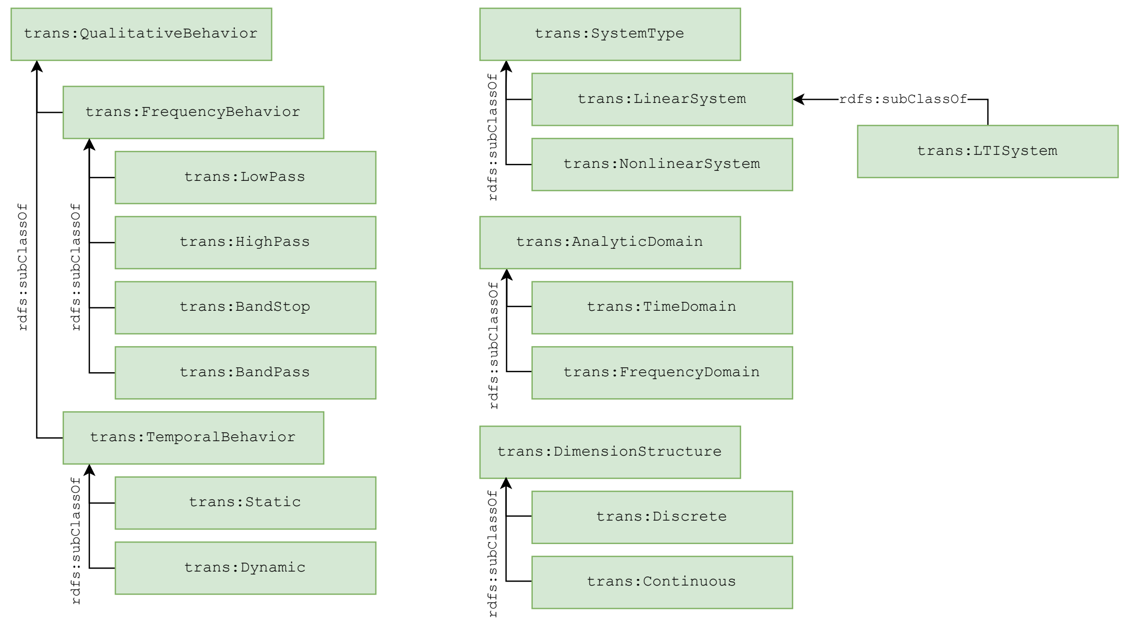

Figure 3.2.9: Additional properties of `trans:TransferModel`.

3.2.4 Representing the Exemplary Use Case

In order to describe the exemplary use case (see chapter 1.3), the self-descriptions of all sensors are required. This is exemplified for a single sensor using the RDF/Turtle syntax. A self-contained self-description of sensor S1 is given by joining listings 3.2.1 to 3.2.7, where:

- listing 3.2.1 imports the required ontologies and namespaces
- listing 3.2.2 represents the actual sensor
- listing 3.2.3 defines the location
- listing 3.2.4 defines the observed quantity
- listing 3.2.5 defines the calibration model
- listing 3.2.6 defines parameters, variables, equation and validity used in the calibration model
- listing 3.2.7 specifies, that observations made by this sensor have a specific unit

Alternatively, a visual representation of the same relations is given in figures 3.2.10 and 3.2.11.

```
# general prefixes
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix schema: <https://schema.org/> .
@prefix om: <http://www.ontology-of-units-of-measure.org/resource/om-2/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix ssn-system: <http://www.w3.org/ns/ssn/systems/> .
@prefix scal: <https://purl.org/onto/scal/> .
@prefix trans: <https://purl.org/onto/trans/> .
@prefix si: <https://ptb.de/si#> .

# specific prefixes
@prefix local: <http://www.example.com/ns/local/> .
@prefix : <http://www.example.com/ns/S1/> .
```

Listing 3.2.1: Imports of the example.

```
:sensor
   a owl:NamedIndividual , sosa:Sensor;
   sosa:isHostedBy local:location_A;
   sosa:observes local:acceleration;
   ssn:hasProperty :model .
```

Listing 3.2.2: Turtle representation of the sensor.

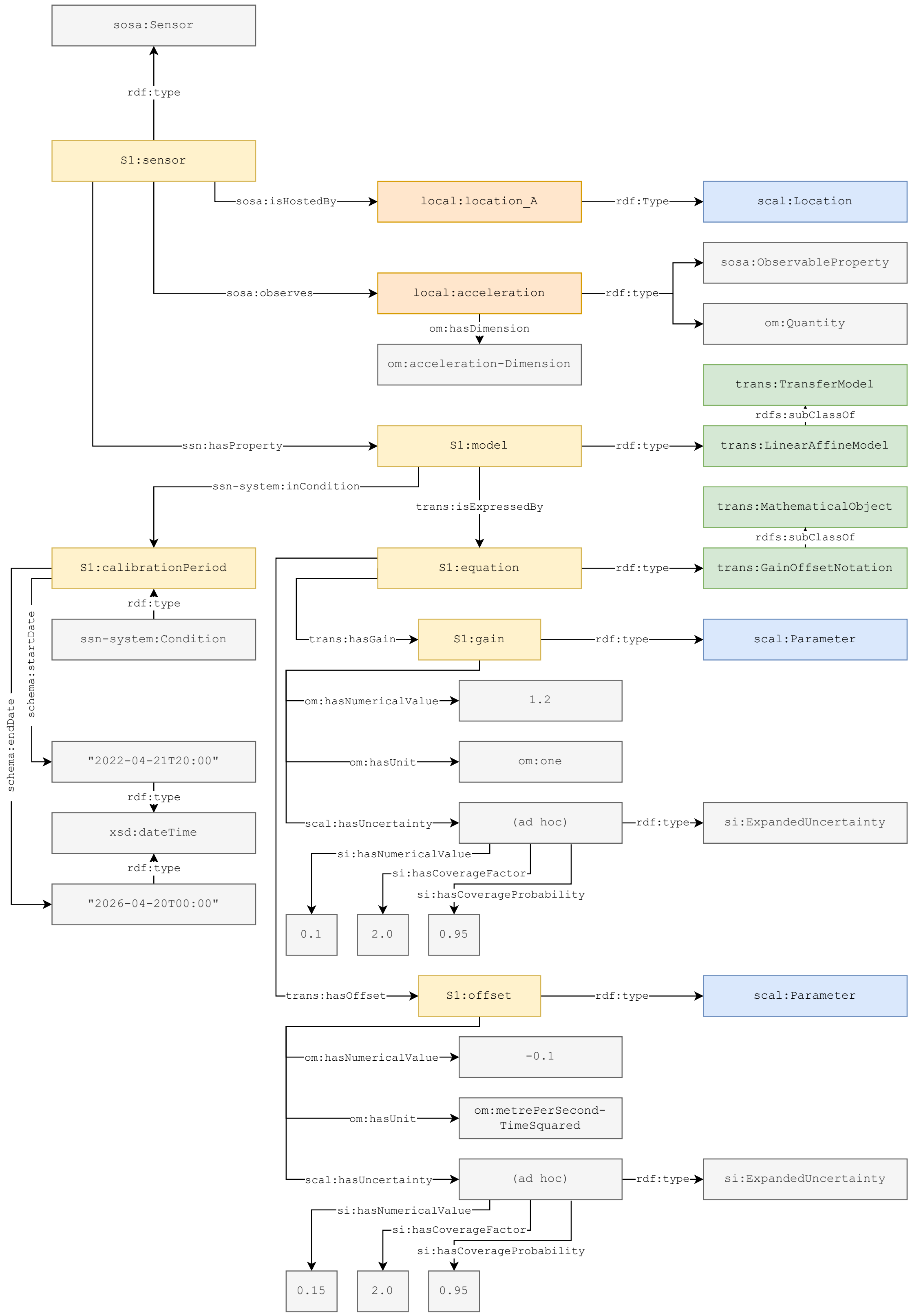

Figure 3.2.10: Main sensor self-description of S1.

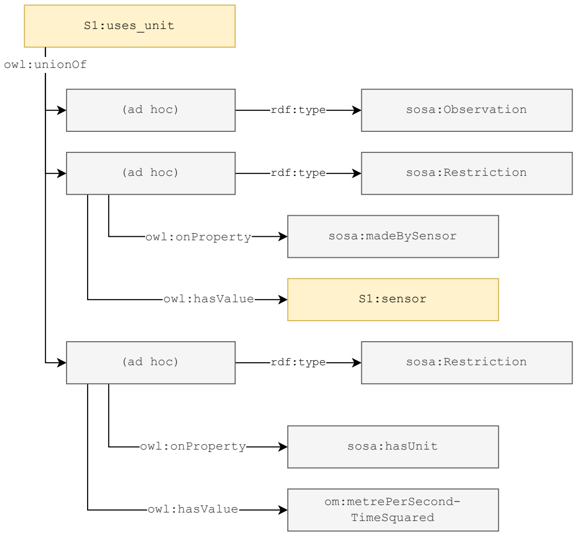

Figure 3.2.11: Additional relations to specify the unit of observations made by sensor S1.

```
local:location_A
    rdf:type owl:NamedIndividual , scal:Location .
```

Listing 3.2.3: Turtle representation of the sensor's locations.

```
local:acceleration
    a owl:NamedIndividual , sosa:ObservableProperty , om:Quantity;
    om:hasDimension om:acceleration-Dimension .
```

Listing 3.2.4: Turtle representation of observed quantities.

```
:model rdf:type owl:NamedIndividual , trans:LinearAffineModel;
       trans:isExpressedBy :equation ;
       ssn-system:inCondition :calibrationPeriod .
```

Listing 3.2.5: Turtle representation of the calibration model.

```
:gain rdf:type scal:Parameter ;
       om:hasNumericalValue 1.2 ;
       om:hasUnit om:one ;
       scal:hasUncertainty [rdf:type
       si:ExpandedUncertainty;
       si:hasNumericalValue 0.1;
       si:hasCoverageFactor 2.0;
       si:hasCoverageProbability 0.95] .

:offset rdf:type scal:Parameter ;
       om:hasNumericalValue -0.1 ;
       om:hasUnit om:metrePerSecond-TimeSquared ;
       scal:hasUncertainty [rdf:type
       si:ExpandedUncertainty;
       si:hasNumericalValue 0.15;
       si:hasCoverageFactor 2.0;
       si:hasCoverageProbability 0.95] .

:equation rdf:type trans:GainOffsetNotation;
       # y = a * x + b
       trans:hasGain :gain ;
       trans:hasOffset :offset .

:calibrationPeriod rdf:type ssn-system:Condition ;
       schema:startDate "2022-04-21T20:00"^^xsd:dateTime ;
       schema:endDate "2026-04-20T00:00"^^xsd:dateTime .
```

Listing 3.2.6: Turtle representation of objects needed for the calibration model.

```
:uses_unit
    a owl:Class;
    owl:unionOf (
        [a sosa:Observation]
        [a owl:Restriction;
            owl:onProperty sosa:madeBySensor;
            owl:hasValue :sensor
        ]
        [a owl:Restriction;
            owl:onProperty om:hasUnit;
            owl:hasValue om:metrePerSecond-TimeSquared
        ]
    ) .
```

Listing 3.2.7: Turtle representation of unit used for observations.

Chapter Summary

The available state of the art of knowledge representations in the domain of sensors and sensor networks is analyzed with regard to its metrological expressiveness. An extension to these knowledge representations is proposed to cover metrological relevant aspects of sensor networks. The additions are formulated as two distinct but connected ontologies. The `scal` ontology extends the `sosa` and `ssn` ontologies with relevant objects to describe a calibration model and enables the communication of uncertainty information in sensor observations. The `trans` ontology allows to represent common transfer behaviors and also capture the mathematical thereof. The proposed merge and extensions are applied to a sensor of the sensor network from chapter 1.3.

3.3 Initializing the Co-Calibration from Semantic Knowledge

Once knowledge of a sensor network is represented in the semantically expressive form proposed in the previous chapter (see section 3.2.4), it can be used to initialize the co-calibration method. This includes the selection of suitable calibration reference sensors and the initial prior of the parametrization of the transfer behavior. The general task is to translate the competency questions (CQs) presented earlier into logical expressions that can be evaluated using a combination of semantic querying and reasoning tools. Providing these expressions leads to machine-actionable operations on interoperable knowledge representation that initialize a mathematical method. Hence, highlighting the potential and emerging synergy of linking the semantic and mathematical domain for automated approaches.

3.3.1 Finding Suitable Calibration References in a Network

A sensor $\mathcal{S}$ (of an existing sensor network) is suitable to act as a calibration reference in a homogeneous co-calibration[12] of a another sensor $\mathcal{S}_{tbc}$ if it[13]:

- has a (valid) calibration model ($\mathcal{S} \in A_{cal}$)
- measures the same quantity[14] as the sensor to be co-calibrated ($\mathcal{S} \in A_{qty}$)
- is located at the same place[15] as the sensor to be co-calibrated ($\mathcal{S} \in A_{loc}$)

This directly corresponds to the CQs raised to design the ontology proposed in the previous chapter. In the following, these CQs are translated into logical expressions that can be evaluated on a joined knowledge graph of the sensor network. To obtain all sensors with a valid calibration model, the set expression defined in theorem 11 needs to be evaluated which makes use of the definitions 27 and 28. To obtain all sensors that observe the same quantity as $\mathcal{S}_{tbc}$, evaluate the expression in theorem 12. The set of all suitable reference sensors to co-calibrate $\mathcal{S}_{tbc}$ is then given by the intersection of the three sets from theorems 11 to 13, which is summarized as theorem 14.

[12]see also definition 11

[13]The index "tbc" stands for "to be co-calibrated". There are multiple reasons to co-calibrate a sensor, e.g. it is new and was not calibrated before, its calibration model is no longer valid or a sensor performance monitoring routine suggested an out-of-schedule re-calibration.

[14]see also definition 17

[15]see also definitions 15 and 17

Evaluating these set expressions requires some reasoning, as (e.g., use case specific) derived sub-classes and sub-properties of the mentioned concepts need to match as well. The expressions for A_{cal}, A_{qty} and A_{loc} have been designed in a way that translates directly into SPARQL-queries. The implementation details are provided in chapter 4.1.

Definition 27 (Subclass Chain Property)**.** *The* `a/rdfs:subClassOf*` *property defines a relationship that matches a specific type or any derived subclasses of this type*[16] *and is given by*

$$\begin{aligned}
&(\mathtt{a/rdfs:subClassOf*})^{\mathcal{I}} = \\
&\qquad\quad (\mathtt{rdf:type})^{\mathcal{I}} \\
&\quad \cup \quad (\mathtt{rdfs:subClassOf})^{\mathcal{I}} \\
&\quad \cup \quad (\mathtt{ObjectPropertyChain(rdfs:subClassOf,\ rdfs:subClassOf)})^{\mathcal{I}} \\
&\quad \cup \quad \ldots \\
&\quad \cup \quad (\mathtt{ObjectPropertyChain(rdfs:subClassOf,\ \ldots,\ rdfs:subClassOf)})^{\mathcal{I}} \\
&\quad \cup \quad \ldots
\end{aligned} \tag{3.3.1}$$

Definition 28 (Exemplified Temporal Validity Check)**.** *The $isvalid(p)$ check provides a boolean value whether the model p is considered valid. E.g. to check if a calibration period is available, it could be evaluated by*

$$\begin{aligned}
isvalid(p) = \quad & \mathit{if} \quad \exists\, c, t_1, t_2 : \\
&\qquad\qquad (p, c) \in (\mathtt{ssn\text{-}system:inCondition})^{\mathcal{I}} \\
&\qquad\quad \wedge\ (c, t_1) \in (\mathtt{schema:startDate})^{\mathcal{I}} \\
&\qquad\quad \wedge\ (c, t_2) \in (\mathtt{schema:endDate})^{\mathcal{I}} \\
& \mathit{then} \quad (t_1 \leq \mathtt{now()}) \wedge (\mathtt{now()} \leq t_2) \\
& \mathit{else} \quad \mathit{true}
\end{aligned} \tag{3.3.2}$$

Theorem 11 (Set of Calibrated Sensors)**.** *The set of all (known) validly calibrated sensors is given by*

$$\begin{aligned}
&A_{cal} = \\
&\Big\{ \mathcal{S} \mid \forall \mathcal{S}, p : \\
&\qquad\qquad (\mathcal{S}, \mathtt{sosa:Sensor}) \in (\mathtt{a/rdfs:subClassOf*})^{\mathcal{I}} \\
&\qquad\quad \wedge\ (\mathcal{S}, p) \in (\mathtt{ssn:hasProperty})^{\mathcal{I}} \\
&\qquad\quad \wedge\ (p, \mathtt{scal:CalibrationModel}) \in (\mathtt{a/rdfs:subClassOf*})^{\mathcal{I}} \\
&\qquad\quad \wedge\ isvalid(p) \\
&\Big\}
\end{aligned} \tag{3.3.3}$$

Rationale. The expression checks if a sensor has a calibration model and if this is valid. □

[16]Note: Using `a` instead of `rdf:type` is a common shortcut. The asterix `*` is used in the sense of common regular expression syntax.

Theorem 12 (Sensors Observing the same Quantity)**.** *The set of (known) sensors measuring the same quantity as sensor $\mathcal{S}_{tbc}$ is given by*

$$
\begin{aligned}
&A_{qty}(\mathcal{S}_{tbc}) = \\
&\Big\{\mathcal{S} \mid \forall \mathcal{S}, m_1, m_2 : \\
&\qquad\qquad \begin{array}{rlcl}
 & (\mathcal{S}, m_1) & \in & (\mathtt{sosa{:}observes})^{\mathcal{I}} \\
\wedge & (\mathcal{S}_{tbc}, m_2) & \in & (\mathtt{sosa{:}observes})^{\mathcal{I}} \\
\wedge & (m_1, m_2) & \in & (\mathtt{owl{:}sameIndividual})^{\mathcal{I}} \\
\wedge & (\mathcal{S}, \mathcal{S}_{tbc}) & \in & (\mathtt{owl{:}differentIndiviuals})^{\mathcal{I}}
\end{array} \\
&\Big\}
\end{aligned}
\tag{3.3.4}
$$

Rationale. The expression checks if a sensor's observed quantity matches the observed quantity of $\mathcal{S}_{tbc}$. □

Theorem 13 (Sensors at the same Location)**.** *The set of all sensors that are located at the same place as $\mathcal{S}_{tbc}$ is given by*

$$
\begin{aligned}
&A_{loc}(\mathcal{S}_{tbc}) = \\
&\Big\{\mathcal{S} \mid \forall \mathcal{S}, l_1, l_2 : \\
&\qquad\qquad \begin{array}{rlcl}
 & (\mathcal{S}, l_1) & \in & (\mathtt{sosa{:}isHostedBy})^{\mathcal{I}} \\
\wedge & (\mathcal{S}_{tbc}, l_2) & \in & (\mathtt{sosa{:}isHostedBy})^{\mathcal{I}} \\
\wedge & (l_1, l_2) & \in & (\mathtt{owl{:}sameIndividual})^{\mathcal{I}} \\
\wedge & (\mathcal{S}, \mathcal{S}_{tbc}) & \in & (\mathtt{owl{:}differentIndiviuals})^{\mathcal{I}}
\end{array} \\
&\Big\}
\end{aligned}
\tag{3.3.5}
$$

Rationale. The expression checks if another sensor's location is the same as the location of $\mathcal{S}_{tbc}$. □

Theorem 14 (Reference Sensors)**.** *The set of all (known) reference sensors is given by*

$$A_{ref}(\mathcal{S}_{tbc}) = A_{cal} \cap A_{qty}(\mathcal{S}_{tbc}) \cap A_{loc}(\mathcal{S}_{tbc}) \tag{3.3.6}$$

Rationale. The expression intersects the realizations of the initially stated requirements. □

3.3.2 Finding Initial Prior for the Transfer Behavior

Before any prior of the transfer behavior can be initialized, it is necessary to select a suitable model structure. Once this decision is fixed, a distribution that characterizes the initial parametrization of the chosen model structure can be specified. It can, however, not be assumed, that all information required to initialize the co-calibration is always available from the relevant sensor self-descriptions. As the co-calibration routine always requires the same level of information detail for computations, this potential information gap needs to be closed. In the following, decision flow charts are presented that provide heuristics for the model and parameter selection. The general idea is to use what is given, while reverting back to predefined defaults otherwise.

3.3.2.1 Selecting a Model Structure for the Transfer Behavior

If $\mathcal{S}_{tbc}$ already has a (potentially invalid) calibration model, the same model structure is reused. If no further model information is available from the sensor itself, it is first checked whether a similar sensor (e.g. same manufacturer, same observed quantitiy kind, etc.) is available in the sensor network. If that is not the case, another check tests, if most of the selected reference sensors share the same model structure. E.g., if more than half of all reference sensors use the same subclass of `trans:TransferModel`, it will also be selected for $\mathcal{S}_{tbc}$. If none of the above two steps lead to a model structure, a default model structure (e.g., linear affine model) is chosen. This could of course be extended to cover multiple models of varying complexity, which are co-calibrated in parallel and the best fitting model (according to an agreed upon metric) is chosen. If such a metric indicates only unsuitable results for all co-calibrated models, the co-calibration fails and manual intervention is required. Moreover, it generally needs to be checked, whether this leads to a model choice that the co-calibration method can handle, as it cannot proceed otherwise. The decision process is visualized in figure 3.3.1.

In case of the specific co-calibration method presented in chapter 3.1, the only model structure supported is `trans:LinearAffineModel`, as the method is designed for linear affine transfer behavior.[17]

3.3.2.2 Selecting a Prior of the Parametrization

Initializing the prior means to parameterize the prior probability distribution. If the model structure newly selected for $\mathcal{S}_{tbc}$ matches the model structure already in use, the parameter values can be directly reused. If $\mathcal{S}_{tbc}$ is still considered to be valid, uncertainty of these parameters can be reused as well. Otherwise, the uncertainty will be increased by a factor to account for the outdated/invalid parameter values. If the model structure was selected based on the majority of the reference sensor models, the sensor parameters can be obtained from the median of their values and the maximum of each parameters uncertainty. In all other cases, default values for the selected model structure are used for the initialization of the prior. The decision process is visualized in figure 3.3.2.

As the presented co-calibration method is suited only for linear affine models, this leads to a PDF $p_0(a, b, \sigma_y)$ as stated in equation (3.1.16). The parameters necessary to specify p_0 are μ_a, σ_a for a Gaussian distribution of a, μ_b and σ_b for a Gaussian distribution of b and α, β for an inverse gamma distribution of σ_y. The defaults are $\mu_a = 1$, $\mu_b = 0$, $\sigma_a = 1$, $\sigma_b = 1$, $\alpha = 2$, $\beta = 1$. From a sensor's self-description at most $\mu_a = a$, $\mu_b = b$, $\sigma_a = u_a$, $\sigma_b = u_b$ and (maybe) $\sigma_y \geq \sigma_b$ can be updated.

The default represents a sensor with ideal transfer behavior such that input and output are equivalent. Without any further information, this is a sensible assumption, as it represents an ideal sensor transfer characteristic and sensors are often designed to match this ideal.

For other sub-types of the `scal:CalibrationModel`, individual connectors can extract the relevant information for the priors of a method.

---

[17] The linear affine model structure is a generic and very common choice in the related literature, see also chapter 2.2 and section 3.1.3.

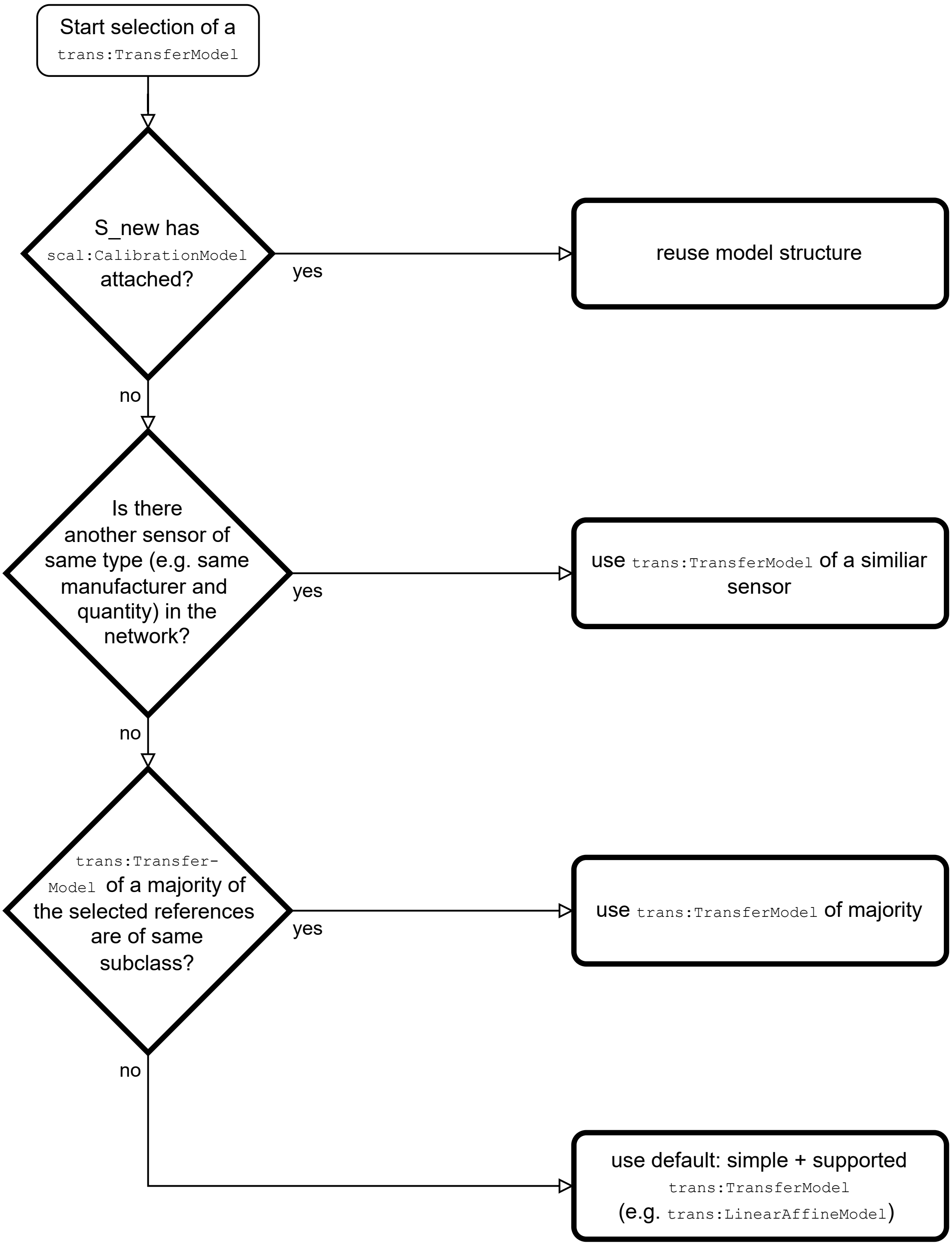

Figure 3.3.1: Decision scheme to select a specific model class.

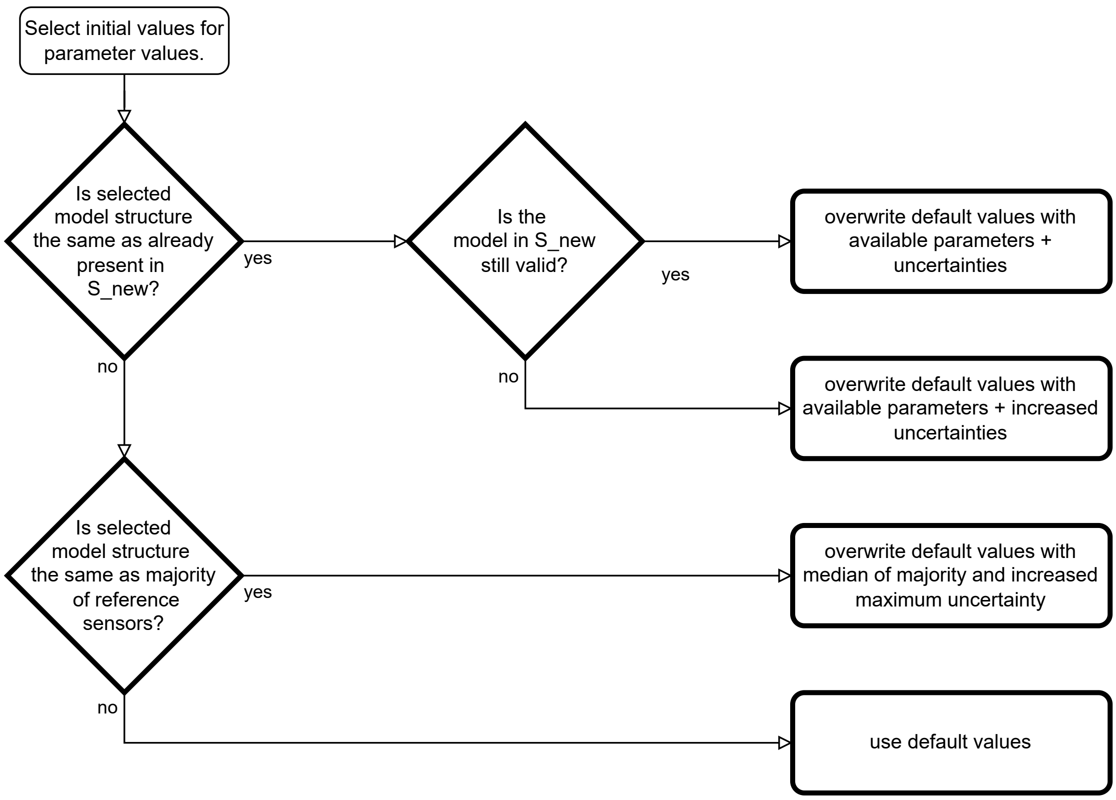

Figure 3.3.2: Decision scheme to initialize the parameter values of the chosen model class.

3.3.3 Initializing Compensation Models of Reference Sensors

The result of a (co-)calibration is a mathematical transfer behavior that quantifies the behavior from the measurand to the indication (see definition 12). From this knowledge and in line with definition 5, a relation to reconstruct the measurand value from the indications can be obtained. In order to calibrate against the best available estimates of the measurands, the reference sensor indications therefore need to be compensated using a reconstruction model. The reconstruction model is a regularized inverse model to the calibration model and specific inversion routines can be applied based on the information given in the sensor's self description. By checking the subclass of the `scal:CalibrationModel` attached to each reference sensor, specific model inverting procedures can be applied:

- `trans:LinearAffineModel` are inverted as shown in appendix A.4
- `trans:InfiniteImpulseResponseModel` are inverted using the approach shown in [**eichstÃďdt__2012**, 13, 99]. Model inversion of dynamic sensor transfer behaviors is not straightforward and requires additional assumptions (or regularization) to obtain causal inverse models (e.g., constraints with regard to the frequency response).
- many other models are invertable by mapping them to a discrete IIR filter and proceeding with the before mentioned method.

If the fitted reconstruction model is given as discrete `trans:ImpulseResponseModel`, the reconstruction filter can be applied using the online IIR filter method proposed in [14]. Moreover, the above mentioned methods return uncertainty-aware reconstruction models, leading to traceable estimated measurand values. Further information about model inversion and input reconstruction is given in an upcoming book on dynamic measurements [22].

3.3.4 Application to the Exemplary Use Case

Let S6 be an acceleration sensor which was recently introduced into the sensor network used throughout this thesis (see chapter 1.3). S6 is not calibrated and it is of interest to prepare a run of the co-calibration routine. This requires to:

- select suitable reference sensors within the sensor network
- select an appropriate model structure
- provide an initial prior of the model parametrization

3.3.4.1 Suitable Reference Sensors

The set of suitable reference sensors is given by equation (3.3.6). The relevant subsets are evaluated on the self-descriptions of all six sensors given in chapter 1.3 and using equations (3.3.3) to (3.3.5) with $\mathcal{S}_{tbc} = S6$.

$$A_{cal} = \{S1, S2, S3, S4\} \tag{3.3.7}$$
$$A_{qty}(S6) = \{S1, S2, S3\} \tag{3.3.8}$$
$$A_{loc}(S6) = \{S1, S2, S4\} \tag{3.3.9}$$

From this it follows that the set of suitable reference sensors to co-calibrate S6 is

$$A_{ref}(S6) = \{S1, S2\} \tag{3.3.10}$$

3.3.4.2 Model Structure

The schema visualized in figure 3.3.1 is applied to sensor S6 in order to select a model structure that will be used in the co-calibration. Looking at the relevant parts of the self-description of S6 in listing 3.3.1, it can be seen that the sensor has a `:model` of type `trans:LinearAffineModel` attached. The model is not parameterized and invalid, as the attached date range condition is unsatisfiable by construction. Nevertheless, following figure 3.3.1 the selected model structure for S6 is a `trans:LinearAffineModel`.

```
:sensor
   a owl:NamedIndividual , sosa:Sensor ;
   sosa:isHostedBy local:location_A ;
   sosa:observes local:acceleration ;
   ssn:hasProperty :model .
:invalid rdf:type ssn-system:Condition ;
      schema:startDate "2022-01-01T00:00"^^xsd:dateTime ;
      schema:endDate "2022-01-01T00:00"^^xsd:dateTime .

:model rdf:type owl:NamedIndividual , trans:LinearAffineModel;
      ssn-system:inCondition :invalid .
```

Listing 3.3.1: Details of the self-description of sensor S6.

3.3.4.3 Model Parametrization

Following figure 3.3.2, the selected model structure for S6 coincides with the model structure previously used for S6. Because the previous model is invalid and no parameters of `:model` are provided, no default values are overwritten. The (uninformative and default) initial prior for the co-calibration of a `trans:LinearAffineModel` is therefore (see section 3.3.2.2): $\mu_a = 1$, $\mu_b = 0$, $\sigma_a = 1$, $\sigma_b = 1$, $\alpha = 2$, $\beta = 1$.

Chapter Summary

This chapter conceptualizes, how semantic information about sensors in a sensor network can be used in the context of a co-calibration task. This is achieved by defining expressions to find co-calibration references, provides a heuristic to select a transfer behavior and then outlines how parameters can be initialized from these findings. With that, the ideas of this chapter connect the mathematical and semantic contributions presented in chapters 3.1 and 3.2.

Part IV

Evaluation and Experiments

4.1 Proof-of-Concept Implementation

In order to test and evaluate the proposed co-calibration method, proof-of-concept implementations for the mathematical and semantic part are provided. To document these implementations, the following chapter includes descriptions of the frameworks used, configuration possibilities, execution and results.

4.1.1 Simulation Environment for Homogeneous Sensor Networks

The simulation environment is centered on use cases and the evaluation of different co-calibration algorithms in these. Therefore, a use case is set up in a configuration step. The configuration is handed to the execution step, which runs the specified methods on the same input data. The results are then stored in a common format from which visualizations can be generated in a successive step.

4.1.1.1 Setup

To support a variety of use cases and methods, the simulation environment is highly configurable. Each configuration is defined using nested key-value pairs stored in one (or more) configuration file(s). The nested structure has six top level entries that are detailed in the following.

In `"random_state"` the initial state of the random number generator used by the numerical methods can be specified. The (noisy) signal creation and proposed Monte-Carlo methods rely on random number generation processes. However, for ideal recreation of simulation outcomes (e.g. for error inspection), it is necessary to fix the random state. If `"random_state" : null` is given, the initial random state typically differs between runs, leading to non-identical results.

The entry `"reference_sensors"` allows to specify how many reference sensors should be used. Initializing arguments (e.g.: model type, parameters, parameter uncertainties, dropout rate, outlier rate) are passed to the routine that creates the `CalibratedSensor` objects which are later used during the simulation. It can be selected, whether the parameters of the transfer function should be randomized according to the provided parameter uncertainty. Alternatively, a list of fully specified reference sensor descriptions can be provided, that includes three models: one for simulation, one known from calibration and one used for the compensation of indicated values (which is the inverse to the calibration model). The distinction between a simulation model and calibration model is crucial, as this matches the real world case, where only the calibration model is known - but not the true transfer characteristic of a sensor.

The `"device_under_test"` describes the sensor that will be co-calibrated by providing the simulation model and initial estimates for the calibration model and its inverse.

In `"measurand"` the time signal of the physical quantity being observed by all sensors is specified as a function that can be evaluated at arbitrary timestamps. It is possible to choose a sinusoidal process with or without (random) discontinuities, system noise, varying frequency (chirp) and constant amplitude. A discrete-time version of the measurand is evaluated at the timestamps, which are also later used for the measurement values of all sensors.

In `"sensor_readings"` the simulated indicated values of all sensors (reference sensors and device under test) are computed from the individual simulation model and the time-discrete measurand.

In `"cocalibration"` actual details of the co-calibration routine are configured. To test a method's online capabilities, the input signals can be split into predefined or random blocks which the method is called sequentially upon. Moreover, the actual methods to be used for co-calibration are initialized by providing a (Python-)class name and arguments to instantiate the class.

For all top-level settings inside the configuration file, the nested key-value pairs can either be given directly or by specifiying a path to another file, that stores the actual content. This enables e.g. reusable definitions of the methods of interest. With this, configuration aspects relevant for more than one use case (e.g., the specific method settings) can be reused across simulations. Moreover, this allows to identically recreate interesting or problematic method outcomes.

The configuration file for the example given in section 3.1.7 is listed in listing 4.1.1. The mentioned configuration of the `device_under_test` is given by listing 4.1.2 and of the `gibbs_minimal` method by listing 4.1.3.

4.1.1.2 Execution

Upon execution, the configuration is loaded and details about the simulation environment (machine, operating system, processor, Python version, installed Python packages and the commit hash of the current simulation code repository) are logged for later reference. Based on the descriptions given in the configuration file, executable objects are created.

Each co-calibration method provides (at least) an `update_parameter`-method, which receives the (blockwise) sensor readings and returns the updated parameter estimate given the new data. Once a method has consumed all available data, the collected and incrementally returned results are written into a JSON output file. The execution time of each method is logged into a separate file.

4.1.1.3 Source Code Details

The simulation environment to co-calibrate a sensor transfer behavior based on (already selected and suitable) reference sensor readings is implemented in the Python programming language using additional packages:

- Python (3.9.2) [125]

```json
{
    "random_state": null,
    "reference_sensors": "experiments/07_thesis_example/reference_sensors.json",
    "device_under_test": "experiments/07_thesis_example/device_under_test.json",
    "measurand": {
        "type": "SinusoidalMeasurand",
        "args": {
            "sigma_x": 0.01,
            "amplitude": 2.0,
            "value_offset": 1.0
        },
        "time_args": {
            "time_start": 0,
            "time_end": 20.0,
            "dt": 0.01
        }
    },
    "sensor_readings": {
        "dut_noise": "based_on_sigma_y_true",
        "ref_noise": "based_on_unc",
        "sigma_y_true": 0.1
    },
    "cocalibration": {
        "interpolate": false,
        "blockwise": true,
        "split_indices" : [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800],
        "methods": {
            "gibbs_minimal": "method_args/gibbs_minimal.json",
            "gibbs_known_sigma_y": "method_args/gibbs_known_sigma_y.json",
            "gibbs_no_EIV": "method_args/gibbs_no_EIV.json",
            "joint_posterior": "method_args/joint_posterior.json",
            "joint_posterior_agrid": "method_args/joint_posterior_grid_adjust.json"
        }
    }
}
```

Listing 4.1.1: Configuration file `config.json` for section 3.1.7.

```json
{
    "S6": {
        "hasSimulationModel": {
            "type": "LinearAffineModel",
            "params": {
                "a": 2,
                "b": 1,
                "ua": 0.01,
                "ub": 0.01,
                "uab": 0.0
            }
        },
        "hasCalibrationModel": {
            "type": "LinearAffineModel",
            "params": {
                "a": 1,
                "b": 0,
                "ua": 1,
                "ub": 1,
                "uab": 0.0
            }
        },
        "hasCompensationModel": {
            "type": "LinearAffineModel",
            "params": {
                "a": 1,
                "b": 0,
                "ua": 1,
                "ub": 1,
                "uab": 0.0
            }
        },
        "misc": {}
    }
}
```

Listing 4.1.2: Configuration file `device_under_test.json` used in listing 4.1.1.

```
{
    "class_name": "GibbsPosterior",
    "arguments": {
        "gibbs_runs": 100,
        "burn_in": 10,
        "use_every": 5,
        "sigma_y_is_given": false,
        "no_error_in_variables_model": false,
        "use_robust_statistics": false,
        "prior": {
            "a": {
                "type": "norm",
                "params": {
                    "loc": 1.0,
                    "scale": 1.0
                }
            },
            "b": {
                "type": "norm",
                "params": {
                    "loc": 0.0,
                    "scale": 1.0
                }
            },
            "sigma_y": {
                "type": "invgamma",
                "params": {
                    "a": 2.0,
                    "loc": 0.0,
                    "scale": 1.0
                }
            }
        }
    }
}
```

Listing 4.1.3: Configuration file `gibbs_minimal.json` used in listing 4.1.1.

- numpy (1.22.1) [126]
- scipy (1.7.3) [127]
- sympy (1.9) [128]
- matplotlib (3.5.1) [129]
- time-series-buffer (0.1.4b0) [130]

The source code for the implementation is part of the supplementing digital documents of this thesis, as well as available at:

`https://github.com/mgrub/phd_sensor_simulation`

4.1.2 Reasoning Environment

In order to evaluate the set expressions established in chapter 3.3, all required knowledge needs to be available in a common format and implicit relations need to be made explicit by the use of a reasoner. Therefore, existing ontologies and relevant sensor self-descriptions are loaded from their respective Turtle files with an intermediate translation step to RDF/XML. A HermitT [77] reasoner is then applied to all RDF-triples initially loaded. Once this is achieved, SPARQL-requests are build that match the sought set expressions and return potential reference sensors on execution.

4.1.2.1 Translation of Set Expressions into Queries

As mentioned in chapter 3.3, the set expressions used in theorem 14 (equations (3.3.3) to (3.3.5)) can be directly translated in to SPARQL templates given by listings 4.1.4 to 4.1.6. These templates are turned into valid SPARQL request by adding relevant prefix definitions and substituting `@@TARGET@@` with a specific sensor identifier, e.g. `sensor_S6:sensor`.

```
SELECT ?s
WHERE {
    ?s a/rdfs:subClassOf* sosa:Sensor ;
    ssn:hasProperty ?prop .
    ?prop a/rdfs:subClassOf* scal:CalibrationModel .
    OPTIONAL {
        ?prop ssn-system:inCondition ?cond .
        ?cond schema:startDate ?start .
        ?cond schema:endDate ?end .
    }
    FILTER( !BOUND(?start) || ?start <= NOW() ) .
    FILTER( !BOUND(?end) || NOW() <= ?end ) .
}
```

Listing 4.1.4: SPARQL template matching theorem 11.

```
SELECT DISTINCT ?s
WHERE {
       @@TARGET@@ sosa:observes ?m1 .
       ?s sosa:observes ?m2 .
       ?m1 om:hasDimension ?d1 .
       ?m2 om:hasDimension ?d2 .
       FILTER( SAMETERM(?d1, ?d2) && !SAMETERM(?s, @@TARGET@@) )
}
```

Listing 4.1.5: SPARQL template matching theorem 12.

```
SELECT ?s
WHERE {
       ?s sosa:isHostedBy ?l1 .
       @@TARGET@@ sosa:isHostedBy ?l2 .
       FILTER( SAMETERM(?l1, ?l2) && !SAMETERM(?s, @@TARGET@@) )
}
```

Listing 4.1.6: SPARQL template matching theorem 13.

4.1.2.2 Source Code Details

The reasoning environment to identify suitable reference sensors based on sensor self-descriptions is implemented in the Python programming language using additional packages:

- Python (3.11.2) [125]
- Owlready2 (0.41) [131]
- rdflib (6.3.2) [132]

The source code for the implementation is part of the supplementing digital documents of this thesis, and as well available at:

`https://github.com/mgrub/phd_semantic_init`

Chapter Summary

Independent proof of concept implementations are provided for the semantic and mathematical contributions of this thesis. The implementations are flexible and enable the evaluation of different use cases. For reproducibility, the Python source code is referenced and the used software environments are mentioned.

4.2 Simulation Experiments

The mathematical co-calibration method(s) presented in chapter 3.1 is evaluated by simulating different scenarios using the implementation given in chapter 4.1. The scenarios are designed to address specific aspects of the proposed co-calibration method. Moreover, metrics are presented to provide indicators for result comparison of the simulation results of different methods and scenarios.

4.2.1 Scenarios, Methods and Evaluation Metrics

The following subsections present and motivate the simulated scenarios and the applied co-calibration variants therein. Moreover, the metrics to evaluate each method's performance are introduced.

4.2.1.1 Selection of Simulation Scenarios

To evaluate specific design aspects of the co-calibration method, the scenarios cover different input signals, number of reference sensors, communication errors and uncertainty of reference sensors. An overview of the main simulated scenarios is given in table 4.2.1. The first group of scenarios (1a, 1b, 1c) evaluates the performance of the algorithms with regard to three different input signal types. No noise is added to the measurand or sensor readings, although each signal has an uncertainty associated with it. The second group (2a, 2b, 2c) is the same as the first group, but uses noisy signals by adding noise to the measurand (according to a configurable setting) and to the sensor measurements (based on the associated model variance).

Scenario	Input Signal	Noisy Inputs	Number of Reference Sensors
1a	static	no	3
1b	sinusoidal	no	4
1c	chirp + jumps	no	5
2a	static	yes	3
2b	sinusoidal	yes	4
2c	chirp + jumps	yes	5

Table 4.2.1: Design aspects covered by the main simulated scenarios.

Moreover, an extended set of scenarios is evaluated, to further highlight certain properties of the developed co-calibration methods. The extended scenarios are shown in table 4.2.2. A third group (3a, 3b, 3c) is similar to 2c, but with different partitions of the input data stream blocks and without jumps. It therefore allows to observe the influence of blocksizes onto the co-calibration routine. The fourth group (4a, 4b) evaluates the applicability of the co-calibration methods in case of dropouts or outliers in the input data streams. It evaluates the robustness of the methods against faulty or missing input data. The fifth group of scenarios (5a, 5b, 5c) shows the influence of the uncertainty-level of the reference sensors in relation to the uncertainty level of the sensor under test. In all other scenarios the uncertainty of the reference sensor indications is lower than the model error of the device under test. Moreover, suitability for different numbers of input data streams is evaluated by using between one and five reference sensors.

Scenario	Input Signal	Noisy Inputs	Number of Reference Sensors	Comm. Errors	Unc. of Ref.	Blocksize
1a	static	no	3	-	$<$ DUT	200
1b	sinusoidal	no	4	-	$<$ DUT	200
1c	chirp + jumps	no	5	-	$<$ DUT	200
2a	static	yes	3	-	$<$ DUT	200
2b	sinusoidal	yes	4	-	$<$ DUT	200
2c	chirp + jumps	yes	5	-	$<$ DUT	200
3a	chirp	yes	3	-	$<$ DUT	variable
3b	chirp	yes	3	-	$<$ DUT	100
3c	chirp	yes	3	-	$<$ DUT	200
4a	sinusoidal	yes	2	dropouts	$<$ DUT	200
4b	sinusoidal	yes	5	outliers	$<$ DUT	200
5a	sinusoidal	yes	1	-	$<$ DUT	200
5b	sinusoidal	yes	1	-	$\approx$ DUT	200
5c	sinusoidal	yes	1	-	$>$ DUT	200

Table 4.2.2: Design aspects covered by the extended simulated scenario.

4.2.1.2 Co-Calibration Methods to be Evaluated

Multiple co-calibration methods for linear affine sensor models have been presented and mentioned in this thesis. In the above scenarios the co-calibration methods presented in table 4.2.3 are executed independently on the same input data streams. This includes the proposed methods presented in sections 3.1.3.2 and 3.1.3.3 and variations thereof to evaluate computational reduction potential. Moreover, three variants of a co-calibration method proposed by Stankovic are applied to the data as well [6, 133]. Differing from the original implementation (see [6]), all three included variants are augmented with uncertainty evaluation following the GUM framework. The base, base with uncertainty-weighted calculation and the enhanced Stankovic methods are presented in appendices A.1 and A.2.

In the discussion following in chapter 4.4, the newly developed co-calibration methods (`gibbs_base`, `gibbs_minimal`, `gibbs_known_sigma_y`, `gibbs_no_EIV`, `joint_posterior`, `joint_posterior_agrid`) are referred to as the *proposed methods*, while the state-of-the-art Stankovic methods (`stankovic_base`, `stankovic_base_unc`, `stankovic_enhanced_unc`) are referred to as *reference methods*.

Method	calc a, b, u_a, u_b	calc σ_y, u_σ	use unc.	Method Documentation	Note
`gibbs_base`*	yes	yes	yes	section 3.1.3.2	Method 1 high sample count
`gibbs_minimal`	yes	yes	yes	section 3.1.3.2	Method 1 low sample count
`gibbs_known_sigma_y`	yes	no	yes	section 3.1.3.2	Method 1 with fixed σ_y
`gibbs_no_EIV`	yes	yes	yes	section 3.1.3.2	Method 1 without EIV model
`joint_posterior`	yes	yes	yes	section 3.1.3.3	Method 2 static grid
`joint_posterior_agrid`	yes	yes	yes	section 3.1.3.3	Method 2 adaptive grid
`stankovic_base`	yes	no	no	appendix A	base Stankovic method
`stankovic_base_unc`	yes	no	yes	appendix A	uncertainty weighted gradient
`stankovic_enhanced_unc`	yes	no	yes	appendix A.2	extended Stankovic method

Table 4.2.3: Co-calibration methods used in each simulated scenario. (* only used in scenarios 2a, 2b, 2c)

4.2.1.3 Metrics for Co-Calibration Evaluation

To compare the performance of the consensus-based co-calibration with other methods of same purpose, multiple metrics are introduced in definitions 29 to 36 and used in section 4.2.2. These allow to quantify the convergence to the true model parameters, convergence efficiency, computational requirements and numerical stability.

Computational requirements are captured using the runtime metric in definition 29. To evaluate the estimates of parameters a, b and σ_y; the mean signed difference (definition 30) and the normalized mean absolute error (definition 31) are used as consistency metrics. Consistency can also be evaluated in terms of the quality of the estimated measurand using the inverse f^{-1} of the fitted transfer behavior f. The details are provided in appendix A.4 and allow to obtain an estimate $\hat{X}_{ai} = f^{-1}(Y_i, \underline{\hat{\theta}})$ with uncertainty $u(\hat{X}_{ai})$ of the measurand X_{ai}. The application of the inverse model makes use of the parameter uncertainties and (if available) the estimated variance σ_y^2 of the model error ε_i (see equation (3.1.9)). The consistency of this estimated measurand is evaluated using the mean signed difference (definition 30), the mean squared error (definition 32) and normalized mean squared error (see definition 33). The corresponding metrics MSD_X, MSE_X and $NMSE_X$ are calculated over the whole available time series, hence the expectation operator is the mean over all available points in time.

To quantify the convergence behavior of a method, it can be observed how the span between maximum and minimum of all future estimates (starting from some t) develops. This is possible, because an updated parameter estimate becomes available after each evaluated block of input data. This span $s_\phi(t)$ is defined according to definition 34. Another possibility to rate the convergence is to assess after which time (of the input data) the estimated parameter uncertainty is lower than a given threshold. Two slightly different metrics are evaluated, definition 35 returns the first time the uncertainty level is reached, while definition 36 provides from when on this level is maintained until the end of the simulation.

An overview of all the metrics that are applied to each simulation result is given in table 4.2.4.

Definition 29 (Runtime Metric). *The runtime metric t_{run} is defined as*

$$\Delta t_{run} = t_{end} - t_{start} \tag{4.2.1}$$

with the logged start-time t_{start} and end-time t_{end} of a method. It is used to measure the practical computation time on the used computer system, but not the total CPU time spend.

Definition 30 (Mean Signed Difference Metric). *The mean signed difference MSD_ϕ of a (time-series of a) parameter ϕ with true (simulated) value ϕ_{true} and estimated value $\hat{\phi}$ is given by*

$$MSD_\phi = \mathbb{E}[\hat{\phi} - \phi_{true}] \tag{4.2.2}$$

and is used as a consistency metric [134]. Note: In contrast to an absolute difference metric, it allows to check for systematic under- or overestimation of a method.

Definition 31 (Normalized Mean Absolute Difference Metric). *The normalized mean absolute error $NMAE_\phi$ of a (time-series of a) parameter ϕ with true (simulated) value ϕ_{true}, estimated value $\hat{\phi}$ and estimated uncertainty $u_{\hat{\phi}}$ is defined by*

$$NMAE_\phi = \mathbb{E}[\frac{|\hat{\phi} - \phi_{true}|}{u_{\hat{\phi}}}] \tag{4.2.3}$$

and is used as a consistency metric.

Definition 32 (Mean Squared Error Metric)**.** *The normalized mean squared error $NMSE_\phi$ of a (time-series of a) parameter ϕ with true (simulated) value ϕ_{true} and estimated value $\hat{\phi}$ is defined by*

$$MSE_\phi = \mathbb{E}[(\hat{\phi} - \phi_{true})^2] \tag{4.2.4}$$

and is used as a consistency metric [134].

Definition 33 (Normalized Mean Squared Error Metric)**.** *The normalized mean squared error $NMSE_\phi$ of a (time-series of a) parameter ϕ with true (simulated) value ϕ_{true}, estimated value $\hat{\phi}$ and estimated uncertainty $u_{\hat{\phi}}$ is defined by*

$$NMSE_\phi = \mathbb{E}[\frac{(\hat{\phi} - \phi_{true})^2}{u_{\hat{\phi}}^2}] \tag{4.2.5}$$

and is used as a consistency metric.

Definition 34 (Convergence Span Metric)**.** *The convergence span $s_\phi(t)$ of a parameter ϕ with estimated value $\hat{\phi}(t)$ is defined by*

$$s_\phi(t) = \max_{t^* \geq t} \hat{\theta}(t^*) - \min_{t^* \geq t} \hat{\theta}(t^*) \tag{4.2.6}$$

and is used as a convergence metric. The time dependence of $\hat{\phi}(t)$ is meant in the sense of an updated estimate after each new evaluated block of input data. Hence, although the true parameter is constant, the estimate of it changes over the course of the simulation.

Definition 35 (Uncertainty Threshold Reached Metric)**.** *The time $t_\theta(\tau)$ after which the estimated uncertainty $u_{\hat{\phi}}$ of a parameter ϕ falls below a given threshold τ for the first time is defined by*

$$t_\phi(\tau) = \min_{t \geq t_0} t \quad s.t. \quad u_{\hat{\phi}}(t) \leq \tau \tag{4.2.7}$$

and is used as a convergence metric.

Definition 36 (Uncertainty Threshold Maintained Metric)**.** *The time $t_\theta(\tau)$ after which the estimated uncertainty $u_{\hat{\phi}}$ of a parameter ϕ falls below a given threshold τ for all future (simulated) times is defined by*

$$t^*_\theta(\tau) = \min_{t \geq t_0} t \quad s.t. \quad \forall_{s \geq t} u_{\hat{\phi}}(s) \leq \tau \tag{4.2.8}$$

and is used as a convergence metric.

4.2.1.4 Computational Execution

The methods are executed on a computer with an “AMD Ryzen 7 2700X Eight-Core Processor” and 32GB of memory. The operating system is “Debian 10” with kernel 4.19.0-23-amd64. Multiple scenarios are executed sequentially in advance. Each scenario logs software configuration and results individually into corresponding files.

Metric	Symbol	Definition	Dimension
runtime	Δt_{run}	29	`om:Time`
mean signed difference a	MSD_a	30	same as a
norm. mean abs. error a	$NMAE_a$	31	`om:dimensionOne`
mean signed difference b	MSD_b	30	same as b
norm. mean abs. error b	$NMAE_b$	31	`om:dimensionOne`
mean signed difference σ_y	MSD_{σ_y}	30	same as σ_y
norm. mean abs. error σ_y	$NMAE_{\sigma_y}$	31	`om:dimensionOne`
mean signed difference $\underaccent{\tilde}{X}_a$	MSD_X	30	`om:dimensionOne`
mean squared error $\underaccent{\tilde}{X}_a$	MSE_X	32	`om:dimensionOne`
norm. mean squared error $\underaccent{\tilde}{X}_a$	$NMSE_X$	33	`om:dimensionOne`
span of a shrinks	$s_a(4s) > s_a(16s)$	34	
span of b shrinks	$s_b(4s) > s_b(16s)$	34	
span of σ_y shrinks	$s_\sigma(4s) > s_\sigma(16s)$	34	
u_a below 0.1	$t_a(0.1), t^*_a(0.1)$	35, 36	`om:Time`
u_b below 0.1	$t_b(0.1), t^*_b(0.1)$	35, 36	`om:Time`
u_σ below 0.1	$t_\sigma(0.1), t^*_\sigma(0.1)$	35, 36	`om:Time`

Table 4.2.4: Metrics used to evaluate the results of each co-calibration method in specific scenarios.

4.2.2 Results

The results are presented in two complementing ways. To compare the performance of individual methods in different scenarios with regard to a specific metric, suitable plots are provided in this section. The x-axis corresponds to the abbreviated scenario (e.g., `03b`) and the y-axis equals to the value of the specific metric. Different methods are distinguished by the marker symbol and the color of each marker indicates the kind of the method (blue: method-1-type, cyan: method-2-type, black: Stankovic-type). The runtime and consistency metrics are visualized in figures 4.2.1 to 4.2.10. To maintain a good presentation of the plots, values well above or below[18] the main data range are indicated with ↑! or ↓! at the upper and lower boundaries of each plot. The data underlying these figures is also provided in scenario-wise tables in appendix D. There, the detailed performance (including the convergence metrics) of each method in each scenario is given in tables D.1 to D.42. These tables also contain the estimated parameters as the main results of each simulation. A discussion of the results is provided in chapter 4.4.

[18]From all values within a plot the range from the 10%-quantile to the 90%-quantile is calculated. Values more than twice this range above the 90%-quantile or below the 10%-quantile are considered outliers in this plot. If a logarithmic scale is used, the logarithm of the values is used in these computations.

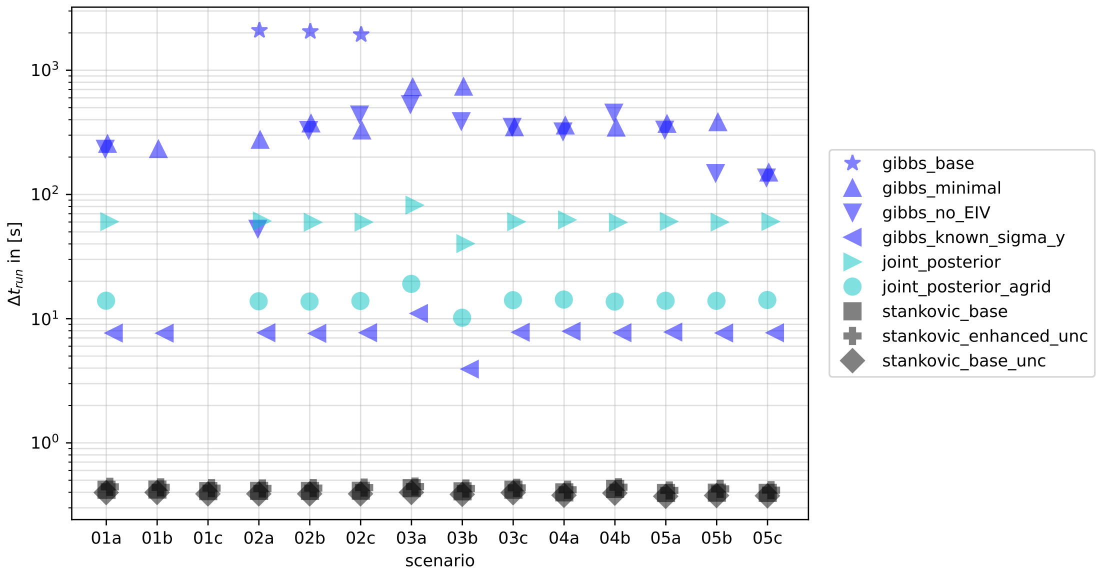

Figure 4.2.1: Overview of Δt_{run} (computation duration) for multiple methods in different scenarios.

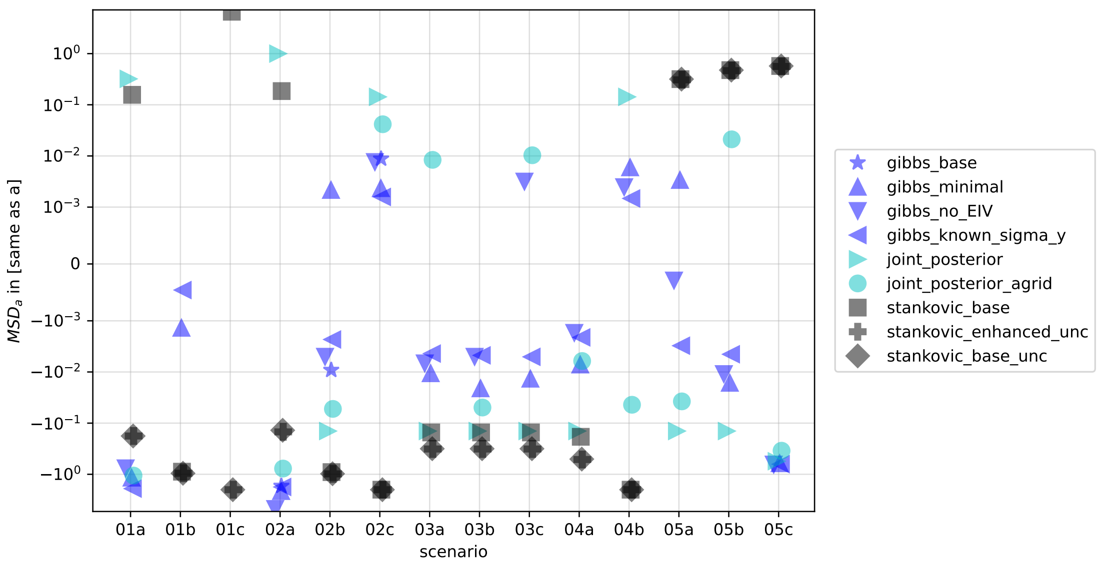

Figure 4.2.2: Overview of MSD_a (mean signed difference of a) for multiple methods in different scenarios.

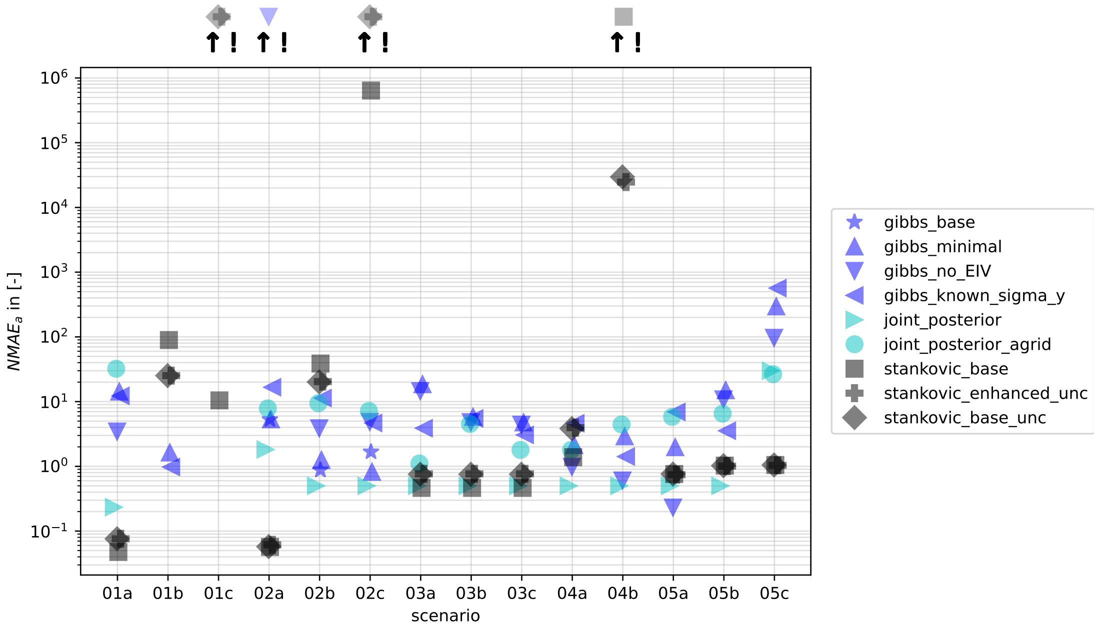

Figure 4.2.3: Overview of $NMAE_a$ (normalized mean absolute error of a) for multiple methods in different scenarios. Outliers are drawn outside above or below the plot area to maintain a good display of the majority of data points.

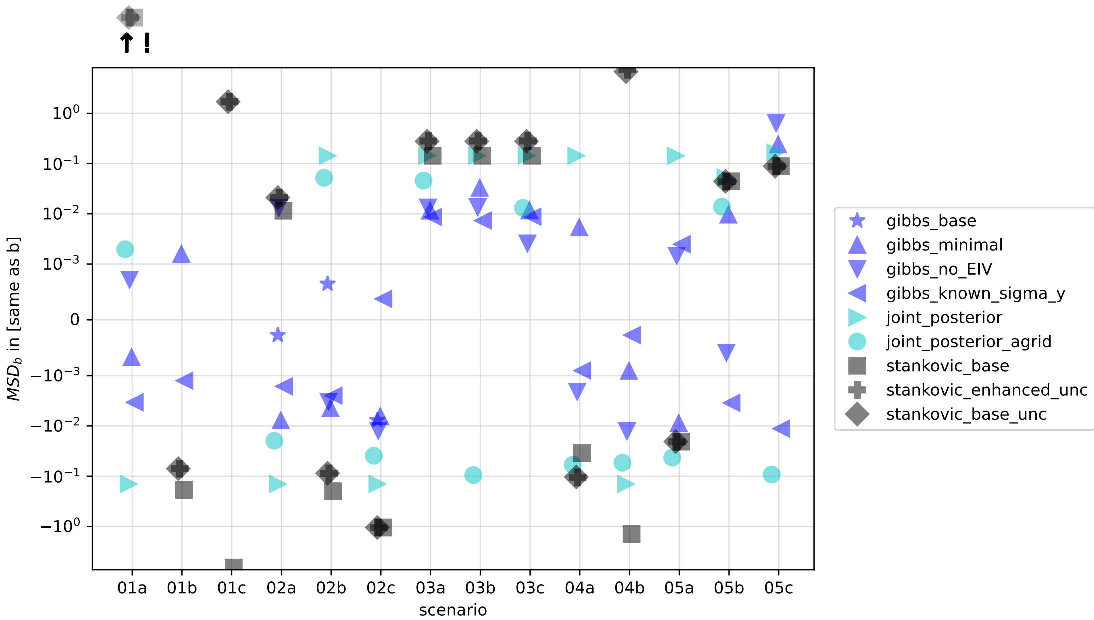

Figure 4.2.4: Overview of MSD_b (mean signed difference of b) for multiple methods in different scenarios. Outliers are drawn outside above or below the plot area to maintain a good display of the majority of data points.

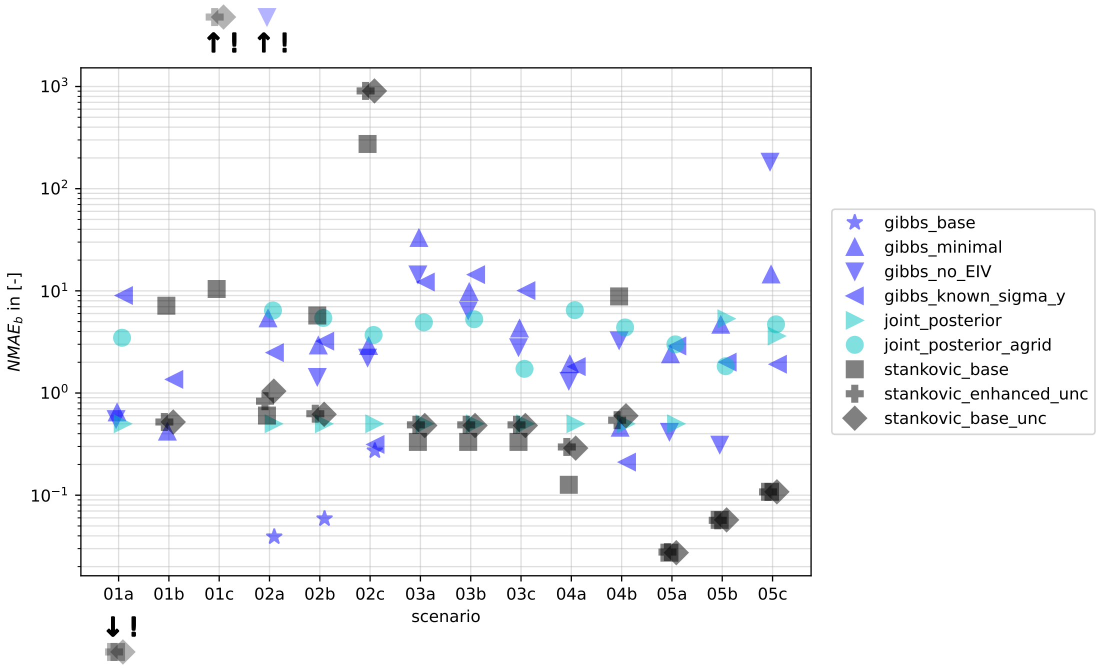

Figure 4.2.5: Overview of $NMAE_b$ (normalized mean absolute error of b) for multiple methods in different scenarios. Outliers are drawn outside above or below the plot area to maintain a good display of the majority of data points.

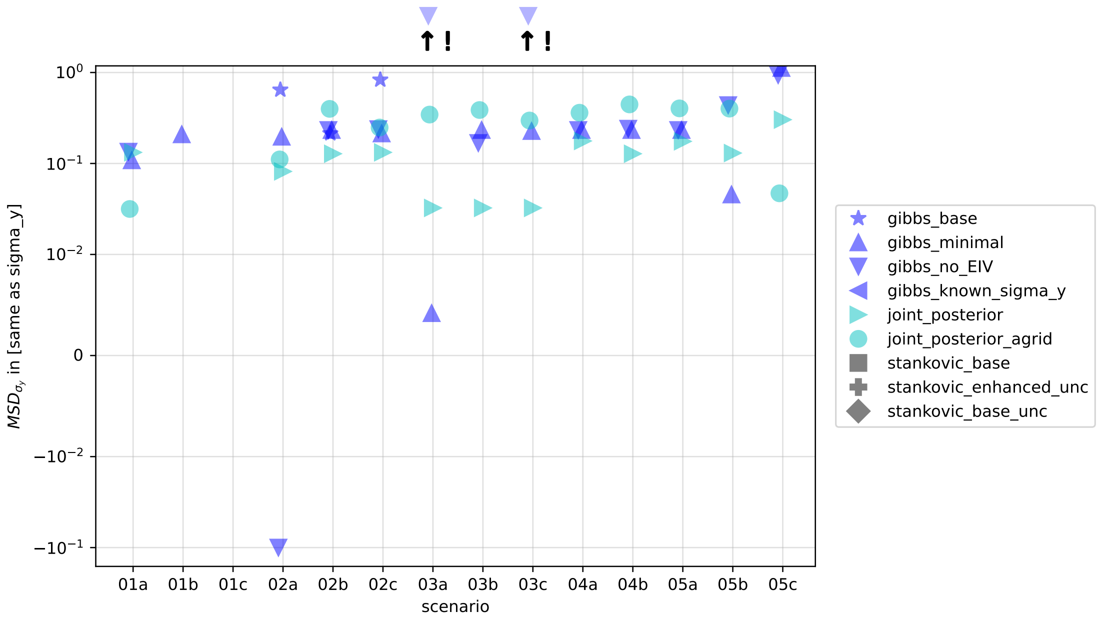

Figure 4.2.6: Overview of MSD_{σ_y} (mean signed difference of σ_y) for multiple methods in different scenarios. Outliers are drawn outside above or below the plot area to maintain a good display of the majority of data points.

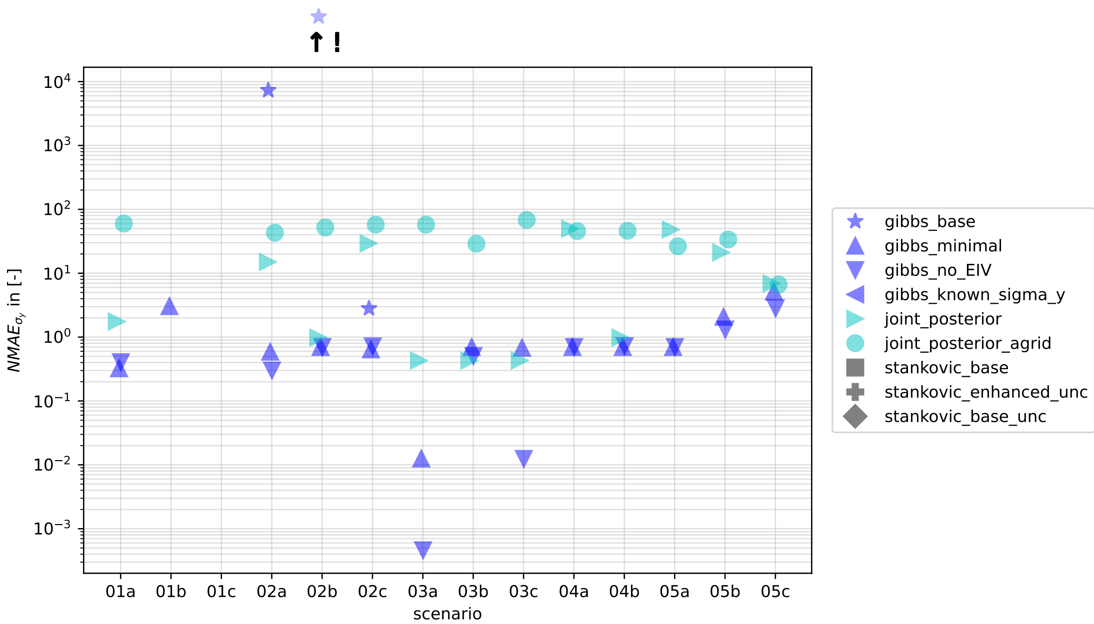

Figure 4.2.7: Overview of $NMAE_{\sigma_y}$ (normalized mean absolute error of σ_y) for multiple methods in different scenarios. Outliers are drawn outside above or below the plot area to maintain a good display of the majority of data points.

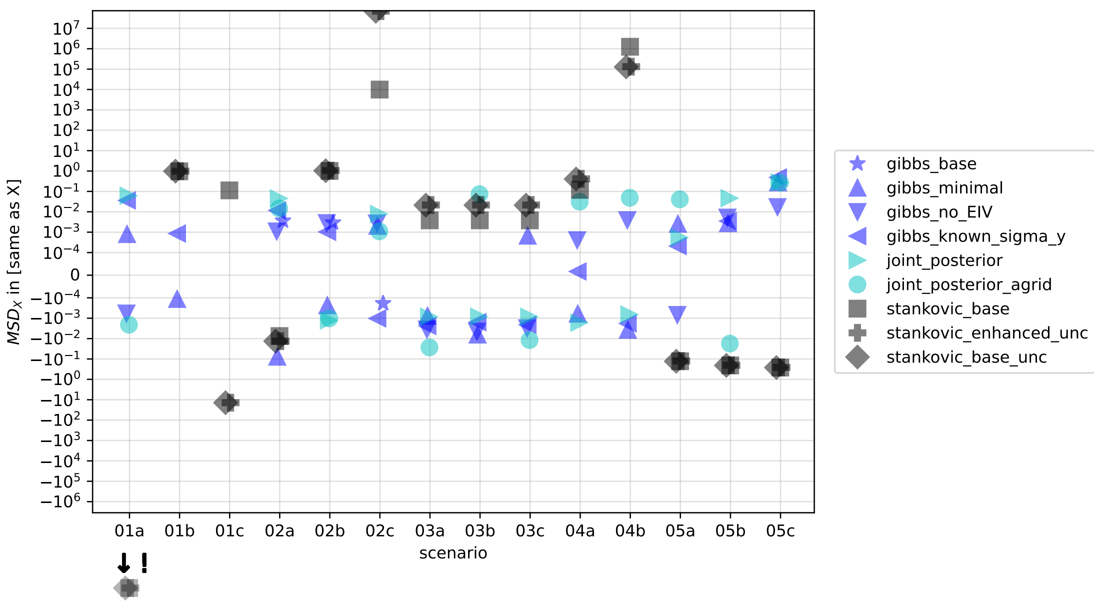

Figure 4.2.8: Overview of MSD_X (mean signed difference of X) for multiple methods in different scenarios. Outliers are drawn outside above or below the plot area to maintain a good display of the majority of data points.

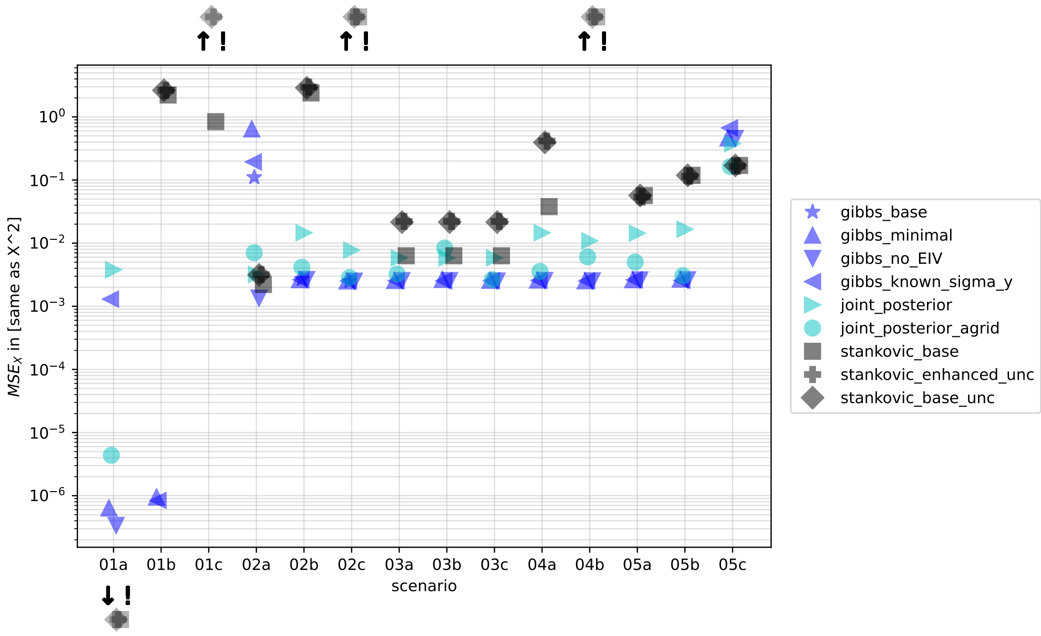

Figure 4.2.9: Overview of MSE_X (mean squared error of X) for multiple methods in different scenarios. Outliers are drawn outside above or below the plot area to maintain a good display of the majority of data points.

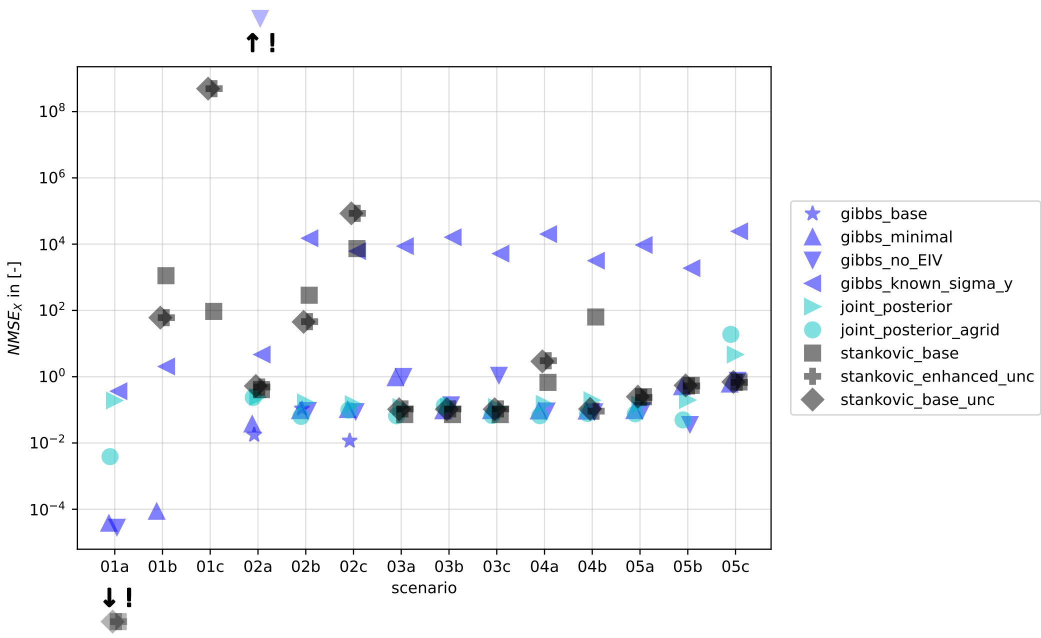

Figure 4.2.10: Overview of $NMSE_X$ (normalized mean squared error of X) for multiple methods in different scenarios. Outliers are drawn outside above or below the plot area to maintain a good display of the majority of data points.

Chapter Summary

The scenarios in which multiple methods are applied to compute a co-calibration are presented in detail. Moreover, the metrics used to evaluate the results are described. The results of these simulations are presented as metric-wise figures for comparison of methods in scenarios with regard to one specific metric. This prepares the detailed discussion in chapter 4.4.

4.3 Ontology Assessment

The two ontologies proposed in chapter 3.2 need to be evaluated whether they match good ontology design practice and are fit for the intended purpose. This is done by multiple means that cover qualitative, logical and quantitative views. Ontology evaluation should focus on seven criteria [135]:

- accuracy: compliance of the axioms with the domain knowledge
- completeness: coverage of the domain of interest
- conciseness: no irrelevant elements or redundant representations
- adaptability: performance in non-anticipated use cases
- clarity: efficient communication of the intended meaning of terms
- computational efficiency: ability and speed of reasoning tasks
- consistency: no inclusion of or potential for contradictions

As proposed by [135], established methods for ontology evaluation can be grouped into four categories. Each of these has a different profile with regard to which criteria it is able to support. *Accuracy*, *completeness* and *conciseness* will be checked by comparing the ontologies with the corpus of a domain-relevant book. By checking the coverage of the intended purpose in terms of competency questions and highlighting future extension, *adaptability* is addressed, although not thoroughly. *Clarity* and *consistency* of the backbone taxonomy will be evaluated by applying the OntoClean methodology to it. By computing metrics of the proposed ontologies and relying on the OWL2 DL, conclusions about general feasibility and *computational efficiency* of reasoning tasks can be made. Moreover, the relevance of a semantic approach is discussed.

4.3.1 Corpus Comparison

A semi-automated corpus based evaluation of the `trans` ontology is performed [135]. As corpus the book *Signals & Systems (2nd Ed.)* by Oppenheim, Willsky, and Nawab is chosen [136]. The book covers fundamental aspects of systems and transfer behaviors and with that is a good candidate for the comparison. However, the book covers more than just system transfer behavior. The top 500 "keywords" (n-grams up to a length of three) are extracted from the book's PDF (approx. 990 pages) using the the YAKE!-algorithm [137].

The string representing each base concept in the ontology (omitting the prefix `trans:`) is then compared to the extracted keywords using the Jaro-Winkler-similarity measure [138]. The most similar match is saved and manually examined. Out of 46 defined concepts, 17 are well represented, 9 fairly well and 21 are not found in the extracted top 500 keywords. After manual re-inspection of not well matching terms against the corpus, 31 concepts can be considered to be present in the corpus of the book. The detailed mapping and some explanation are provided in table 4.3.1.

Concept	Keyword	Score	++	+	-	*+	Remark
`BandPass`	bandpass	1.00	✓			✓	
`Butterworth`	butterworth filters	0.92	✓			✓	
`Continuous`	continuous-time	0.93	✓			✓	
`Discrete`	discrete time	0.92	✓			✓	
`Dynamic`	dynamics	0.97	✓			✓	
`Elliptic`	elliptic	1.00	✓			✓	
`FilterType`	filter	0.92	✓			✓	
`HighPass`	highpass	1.00	✓			✓	
`ImpulseResponseModel`	impulse response	0.94	✓			✓	
`LinearSystem`	linear system theory	0.92	✓			✓	
`LowPass`	lowpass	1.00	✓			✓	
`LTISystem`	lti system. sec	0.91	✓			✓	
`NonlinearSystem`	nonlinear system	0.99	✓			✓	
`Polynomial`	polynomials	0.98	✓			✓	
`StepResponseModel`	step response asymptotically	0.88	✓			✓	
`SystemType`	system	0.92	✓			✓	
`TimeDomain`	time domain	0.98	✓			✓	
`BandStop`	band	0.90		✓		✓	mentioned in book
`EllipticRationalFunction`	elliptic	0.87		✓		✓	
`FrequencyBehavior`	frequency	0.91		✓		✓	
`FrequencyDomain`	frequency modulation	0.94		✓		✓	
`FrequencySpectrum`	frequency	0.91		✓		✓	
`TimeSeries`	time	0.88		✓		✓	
`FiniteImpulseResponseModel`	finite	0.85			✓	✓	listed in index
`InfiniteImpulseResponseModel`	iii	0.73			✓	✓	listed in index
`LinearDifferenceEquation`	linear	0.85			✓	✓	listed in index
`LinearOrdinaryDifferentialEquation`	linear	0.84			✓	✓	"linear differential equation" listed in index
`LinearStateSpaceModel`	linear amplitude scale	0.86			✓	✓	only hint on state space models in book
`StateSpaceMatrixNotation`	statement	0.83			✓	✓	only hint on state space models in book
`StateSpaceModel`	statement	0.82			✓	✓	only hint on state space models in book
`TransferModel`	transforms	0.90			✓	✓	"transfer function" listed in index
`MathematicalObject`	mathematical	0.93		✓			computational representation not scope of book
`NonLinearStateSpaceModel`	nonlinear system	0.88		✓			only hint on state space models in book
`RationalFraction`	rational	0.90		✓			
`AnalyticalDomain`	lti	0.73			✓		not mentioned in book
`Array`	area	0.83			✓		computational representation not scope of book
`ArrayWithUncertainty`	write	0.68			✓		computational representation not scope of book
`Bessel`	essentially	0.76			✓		Bessel filter not mentioned in book
`Chebyshev`	achieve	0.69			✓		Chebyshev filter not mentioned in book
`ContinuousImpulseResponseModel`	continuous-time fourier series	0.86			✓		only parent concept mentioned
`DimensionStructure`	distinct	0.82			✓		not mentioned in book
`GainOffsetNotation`	gain	0.84			✓		not mentioned in book
`LinearAffineModel`	linear	0.87			✓		not mentioned in book
`QualitativeBehavior`	lie	0.72			✓		not mentioned in book
`Static`	statistical signal	0.87			✓		not mentioned in book
`TemporalBehavior`	temperature	0.82			✓		not mentioned in book

Table 4.3.1: Details of the corpus evaluation of the `trans` ontology. Good match (++), mediocre match (+), bad match (-), match based on manual review(*+).

4.3.2 Coverage of the Intended Purpose

The main intention of the `scal` and `trans` ontologies is to represent metrologically relevant information in sensor (network) descriptions. This intention is made explicit by competency questions which capture what a knowledge representation on the basis of this ontology should be able to answer. As noted by [121, 139], three general categories of competency questions can be distinguished: class and relation queries, decision queries and interrogative queries.

The `scal` ontology was designed using the (decision and interrogative) CQs listed in chapter 3.2. Chapters 3.3 and 4.1 show, how these can be translated into machine-actionable queries. From this, it can be concluded that it is possible to answer the competency questions. However, it should be noted, that the first question (a decision query) is rephrased into two sequentially linked questions: (1) “Has a specific sensor a property of type CalibrationModel?” and (2) “Are all conditions of this property fulfilled?”. In addition, the ontology allows to answer the following questions, which have not been part of the design constraints:

- What Parameters and Variables are used in an `EquationModel`?
- Is a Parameter defined with or without uncertainty?

Despite many answerable questions, there are questions which cannot be answered based on the `scal` ontology:

- Is a specific `EquationModel` input-ouput-equivalent to another specific `EquationModel`?
- Is a specific `EquationModel` input-ouput-equivalent to a specific `trans:TransferModel`?

Competency questions for the `trans` ontology have not been stated so far in this thesis, but are partly given in [19]. Leading competency questions in the design of `trans` are:

- What is the `TransferModel` of a specific sensor?
- Is the `TransferModel` of a specific sensor a specialization of another `TransferModel` type?
- What mathematical objects characterize a specific `TransferModel`?
- What are qualitative behaviors of a specific `TransferModel`?
- Is a specific `TransferModel` represented by mathematical objects which have an assigned uncertainty?

These questions all refer to constraints on an individual transfer model and can be directly answered by design. Moreover, the ontology can guide developers of software frameworks what transfer behaviors they may want to implement (and if how), by asking use case independent questions:

- Which subtypes of `TransferModel` exist?
- Which mathematical object can be used to express a `LinearStateSpaceModel`?
- What are the attributes of a `TransferModel`?

However, there are questions that can not be answered using just the knowledge available in the `trans` ontology.

- Is a specific `TransferModel` input-output-equivalent to another specific `TransferModel`?
- What is the discrete variant of a continuous `TransferModel`?

As exemplified, this (as for `scal`) concerns questions regarding the equivalence of two systems in terms of input-output behavior. No conversion-links are provided in the ontology, but are often available in signal-processing toolboxes.

Another intention of the proposed ontologies is to enable sensor self descriptions that are ready for machine-interpretable content according to the five levels of digitalization [140]. That means, that these descriptions are not only machine-readable, but the relationships and meaning used within the description are provided in a machine-actionable way. By relying on Semantic Web technologies (mainly OWL and RDF), the proposed sensor self description based on `scal` and `trans` fulfill the requirements of level four and prepare the fifth level.

4.3.3 Evaluation of the Backbone Taxonomy

The OntoClean methodology is based on modal logics and allows to reveal inconsistencies in the taxonmic backbone of an ontology [141, 142, 143]. The method requires assignment of general (rather) philosophical meta-properties to each concept in the ontology. These meta-properties are: [142]

- rigidity: Is a property essential to an entity in every possible world?
- identity: Can one recognize two individuals as being the same or different?
- unity: Can one recognize all the parts that form an individual entity? [19]
- dependency: Can an entity exist only on existence of another?

For all these properties, a corresponding non-property (can hold for some of its instances) and for some anti-properties (holds for none of its instances) exists. Although there is no definite right or wrong way of assigning these meta-properties to concepts within an ontology (as this depends on the intended meaning), there are rules how some of these properties are inherited along class-subclass-relations. These rules are formulated as an `ontoclean` ontology, which allows to search for inconsistencies with these rules in a programmatic way [144]. To apply the OntoClean method to `scal` and `trans`, the approach follows the essential steps of the tutorial provided in [144]. First, the ontology under evaluation needs to be adapted ("punning") to match the application requirements of the `ontoclean` ontology. To yield the backbone taxonomy (which can be written into a separate file):

- extract all mentioned `owl:Class`
- extract all mentioned `rdfs:subClassOf` relations between the classes
- convert each extracted `owl:Class` into `owl:NamedIndividual` and `ontoclean:Class`
- convert each extracted `rdfs:subClassOf` into `ontoclean:subClassOf`

Next, the meta-properties are assigned to each `ontoclean:Class` using the following classes and abbreviations:

- rigidity (`+R`) by `ontoclean:RigidityClass`
- non-rigidity (`-R`) by `ontoclean:NonRigidityClass`
- anti-rigidity (`~R`) by `ontoclean:AntiRigidityClass`

- unity (`+U`) by `ontoclean:UnityClass`

[19] Or expressed differently: If an individual can be decomposed into instances of the same class, it has a non-unity meta-property.

- non-unity (-U) by `ontoclean:NonUnityClass`
- anti-unity (~U) by `ontoclean:AntiUnityClass`

- identity (+I) by `ontoclean:SortalClass`
- non-identity (-I) by `ontoclean:NonSortalClass`

- dependence (+D) by `ontoclean:DependentClass`
- non-dependence (-D) by `ontoclean:NonDependentClass`

Moreover, the `ontoclean` ontology needs to be listed as explicit import in the adapted backbone taxonomy. Finally, a reasoner is applied to the extracted and adjusted file to detect potential inconsistencies, e.g., the HermiT reasoner can be executed via the Protégé-Frontend [145, 146].

The meta-properties assumed in the `scal` and `trans` ontologies are provided in tables 4.3.2 and 4.3.3 respectively. Because no meta-properties of the underlying external ontologies are known, they are assigned to the best of one's knowledge. No inconsistencies are found within the current state of the ontologies.

To see what would happen in case of an inconsistency, consider the following: If `scal:Location` would be assumed to carry unity `+U` (in the sense of "a single location is a whole and cannot be split into multiple locations"), then the checks would reveal a unity-issue because `scal:Location` is a subclass of `sosa:Platform` (-U) and `geo:SpatialObject` (~U). However, a subclass of a anti-unity carrying class cannot be carrying unity [142] and in the Protégé UI this inconsistency would be visualized and explained as shown in figure 4.3.1.

Class	Rigidity	Unity	Identity	Dependence
`sosa:Platform` *	+R	~U	+I	-D
`sosa:Result` *	+R	+U	+I	-D
`geo:SpatialObject` *	+R	-U	+I	-D
`om:Measure` *	+R	+U	+I	-D
`om:Quantity` *	+R	+U	+I	-D
`sosa:Sensor` *	+R	+U	+I	-D
`ssn:Property` *	+R	+U	+I	-D
`scal:CalibratedSensor`	+R	+U	+I	+D
`scal:CalibrationModel`	+R	+U	+I	-D
`scal:Location`	+R	-U	+I	-D
`scal:MeasureWithUncertainty`	+R	+U	+I	+D
`scal:EquationModel`	+R	+U	+I	+D
`scal:Equation`	+R	+U	+I	-D
`scal:Parameter`	+R	+U	+I	-D
`scal:Variable`	+R	+U	+I	-D

Table 4.3.2: Detailed assertions of meta-properties to the classes in `scal`. (Classes marked with * are assertions to external concepts.)

Class	Rigidity	Unity	Identity	Dependence
math:E34 *	+R	-U	+I	-D
om:Measure *	+R	+U	+I	-D
scal:CalibrationModel *	+R	+U	+I	-D
si:MeasureWithUncertainty *	+R	+U	+I	+D
trans:AnalyticalDomain	+R	+U	+I	-D
trans:Array	+R	+U	+I	+D
trans:ArrayWithUncertainty	+R	+U	+I	+D
trans:BandPass	-R	+U	-I	-D
trans:BandStop	-R	+U	-I	-D
trans:Bessel	+R	+U	+I	+D
trans:Butterworth	+R	+U	+I	+D
trans:Chebyshev	+R	+U	+I	+D
trans:Continuous	+R	+U	+I	-D
trans:ContinuousImpulseResponseModel	+R	+U	+I	+D
trans:DimensionStructure	+R	+U	+I	-D
trans:Discrete	+R	+U	+I	-D
trans:Dynamic	-R	+U	-I	-D
trans:Elliptic	+R	+U	+I	+D
trans:EllipticRationalFunction	+R	+U	+I	+D
trans:FilterType	+R	+U	+I	+D
trans:FiniteImpulseResponseModel	+R	+U	+I	+D
trans:FrequencyBehavior	+R	+U	-I	-D
trans:FrequencyDomain	+R	+U	+I	-D
trans:FrequencySpectrum	+R	-U	+I	+D
trans:GainOffsetNotation	+R	+U	+I	+D
trans:HighPass	-R	+U	-I	-D
trans:ImpulseResponseModel	+R	+U	+I	+D
trans:InfiniteImpulseResponseModel	+R	+U	+I	+D
trans:LinearAffineModel	+R	+U	+I	+D
trans:LinearDifferenceEquation	+R	+U	+I	+D
trans:LinearOrdinaryDifferentialEquation	+R	+U	+I	+D
trans:LinearStateSpaceModel	+R	+U	+I	+D
trans:LinearSystem	+R	+U	+I	-D
trans:LowPass	-R	+U	-I	-D
trans:LTISystem	+R	+U	+I	-D
trans:MathematicalObject	+R	-U	+I	-D
trans:NonLinearStateSpaceModel	+R	+U	+I	+D
trans:NonlinearSystem	+R	+U	+I	-D
trans:Polynomial	+R	+U	+I	+D
trans:QualitativeBehavior	+R	+U	-I	-D
trans:RationalFraction	+R	+U	+I	+D
trans:StateSpaceMatrixNotation	+R	+U	+I	+D
trans:StateSpaceModel	+R	+U	+I	+D
trans:Static	-R	+U	-I	-D
trans:StepResponseModel	+R	+U	+I	+D
trans:SystemType	+R	+U	+I	-D
trans:TemporalBehavior	-R	+U	-I	-D
trans:TimeDomain	+R	+U	+I	-D
trans:TimeSeries	+R	-U	+I	+D
trans:TransferModel	+R	+U	+I	+D

Table 4.3.3: Detailed assertions of meta-properties to the classes in `scal`. (Classes marked with * are assertions to external concepts.)

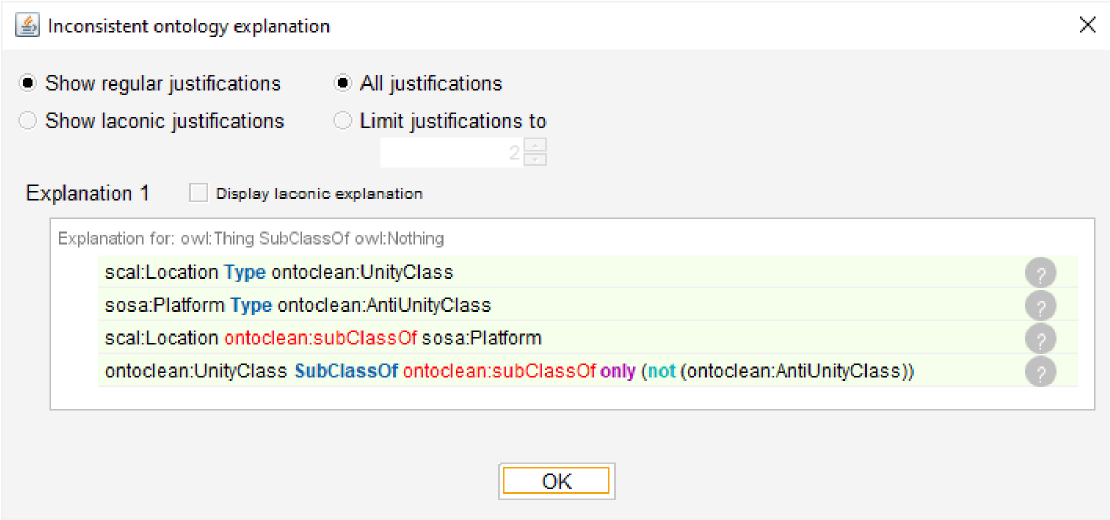

Figure 4.3.1: Protégé screenshot revealing an inconsistency in a hypothetical version of the `scal` ontology.

4.3.4 Schema Metrics

The `scal` and `trans` ontologies are evaluated according to existing schema metrics [147]. The metrics are based on counting objects in the ontology with certain properties and are computed using a combination of SPARQL-queries with subsequent calculations in Python. All queries are evaluated on the ontology after the application of a reasoner and therefore include implicit relations.

The relationship richness (RR) metric defined in definition 37 “reflects the diversity of relations [...] in the ontology” [147] by quantifying the percentage of taxonomic (class-subclass) relations in it. The attribute richness (AR) metric defined in definition 38 “indicate[s] [...] the quality of the ontology design” [147] by quantifying the amount of (human readable) labels and annotations. The inheritance richness (IR) metric defined in definition 39 “describes the distribution of information across different levels of the ontology’s inheritance tree” [147] by calculating the mean number of subclasses per class.

The numeric results of all three metrics are provided in table 4.3.4.

Definition 37 (Relationship Richness Metric)**.** *Within a selected ontology, let SC be the number of all unique classes with a `rdfs:subClassOf`-relation and ALL be the number of unique classes with at least one relation. The relationship richness metric RR is then calculated according to [147]*

$$RR = 1 - \frac{SC}{ALL} \tag{4.3.1}$$

Where SC is computed using listing C.1 and ALL is computed using listing C.2. Values of RR close to zero indicate that the majority of relations are of hierarchical nature, while values close to one indicate the opposite.

Definition 38 (Attribute Richness Metric)**.** *Within a selected ontology, let C be the number of classes and ATT be the number of number of classes that are annotated. The attribute richness metric AR is then calculated according to [147]*

$$AR = \frac{ATT}{C} \tag{4.3.2}$$

Where C is computed using listing C.3 and ATT is computed using listing C.4. Values of AR close to one indicate that most of the classes are annotated, while a value of zero means that no class is annotated.

Definition 39 (Inheritance Richness Metric)**.** *Within a selected ontology, let H_{sum} be the sum of subclasses of each class and C be the number of classes in the ontology. The inheritance richness metric IR is then calculated according to [147]*

$$IR = \frac{H_{sum}}{C} \tag{4.3.3}$$

Where H_{sum} is computed using listing C.5 and C is computed using listing C.3). Smaller values of IR indicate a shallow taxonomic graph, while higher values indicate a large number of inheritance levels.

Metric		`scal`	`trans`
relationship richness	(RR)	0.75	0.53
attribute richness	(AR)	1.00	0.20
inheritance richness	(IR)	0.125	1.59

Table 4.3.4: Results of schema metric evaluation.

Chapter Summary

The two proposed ontologies `scal` and `trans` are evaluated according to existing methods to describe the multiple aspects of these ontologies. Specifically, a corpus comparison revealed good agreement to a book, the execution of competency questions shows the agreement to initial design goals, the OntoClean methodology assesses the consistency of the backbone taxonomy and basic ontology metrics can be calculated. This prepares a discussion of the results in chapter 4.4.

4.4 Discussion of Results

In this chapter, the results presented in the previous two chapters 4.2 and 4.3 are assessed and conclusions are drawn. The discussion covers a critical comparison to the existing state-of-the-art and reflects the consequences with regard to the research questions of this thesis.

4.4.1 Mathematical Insights

Based on the provided measures in different scenarios, statements about the robustness and applicability of each method can be made. Moreover, it is possible to compare the proposed methods against state-of-the-art reference methods developed by Stanković et al. [6, 133]. The general idea is to assess the individual performance aspects based on the simulation results of a specific scenario group.

4.4.1.1 General Performance (`02a`, `02b`, `02c`)

General method performance is assessed using scenarios `02a`, `02b` and `02c`, which differ in the characteristic of the input signal (stationary input, sinusoidal input, jumping chirp input).

Parameter estimates $\hat{a}$ show a good agreement to the true value used in the simulation ($a_{true} = 2.0$) with regard to the MSD_a metric in figure 4.2.2 for the proposed methods only in scenarios `02b` and `02c`. This is confirmed with values below of $NMAE_a < 2.0$ for the `gibbs_base`, `gibbs_minimal` and `joint_posterior` methods. Parameter estimates $\hat{a}$ in scenario `02a` are weak according to MSD_a and the $NMAE_a$ metric indicates an underestimated parameter uncertainty.

Parameter estimates for $\hat{b}$ show good agreement with the true value used in the simulation ($b_{true} = 1.0$) with regard to the MSD_b metric in figure 4.2.4 in all three scenarios. However, only `gibbs_base` and `joint_posterior` are also consistent ($NMAE_b < 2.0$) within their estimated parameter uncertainty in all three scenarios, as seen in figure 4.2.5.

Parameter estimates for $\hat{\sigma}_y$ do not show a good agreement with the true value used in the simulation ($\sigma_y = 0.1$) as indicated by the MSD_{σ_y} metric in figure 4.2.6 in all three scenarios. Although there is a tendency to overestimate this parameter value by all methods, the deviation remains within an acceptable range of $NMAE_{\sigma_y} < 2.0$ for the `gibbs_minimal` and `gibbs_no_EIV` methods as shown in figure 4.2.7. The `joint_posterior_agrid` method consistently underestimates the parameter uncertainty in all three cases, yielding a high $NMAE_{\sigma_y}$ metric.

The general behavior of the parameter estimations is explainable, as the effect of $\hat{a}$ and $\hat{b}$ onto the output signal of the device under test can only be separated if a variation in the input signal exists. Moreover, because the proposed methods (in contrast to the reference methods) are also estimating $\hat{\sigma}_y$, variations similar to (white) noise contribute to the estimation thereof and do not (or only to a small extend) influence the estimate $\hat{a}$. In contrast, the reference methods use any variation (and with that also noise) in the input data to estimate $\hat{a}$, as they aren't estimating the error term related to $\hat{\sigma}_y$. Therefore, the proposed methods need to operate on time varying signals, in which the white noise and deterministic signal can be distinguished. This also shows, that the proposed methods not only *can* operate on time varying signals, but also *need* a noticeable variation over time to obtain usable parameter estimates.

Using the estimated parameters to compute an inverse model yields practically bias-free estimates of the measurand as indicated by the MSD_X metric being close to zero ($|MSD_X| \leq 10^{-1}$) in figure 4.2.8 in all scenarios for all proposed methods. Moreover, the mean squared error MSE_X is kept below 0.02 indicating a good agreement (see figure 4.2.9), but with a potentially overestimated model error σ_y indicated by the $NMSE_X$ metric being much lower than (the ideal value of) 1.0 in figure 4.2.10.

All three reference methods (`stankovic_base`, `stankovic_base_unc`, `stankovic_enhanced_unc`) are providing good parameter estimates in `02a`, however performance degrades for non-stationary input signals (scenarios `02b` and `02c`) in all consistency metrics (e.g., check MSD_a, MSD_b and $NMSE_X$ in figures 4.2.2, 4.2.4 and 4.2.10). This was to some extend unexpected, as no limitations regarding the input signal are mentioned in the related publications, e.g. [6].

Convergence behavior is supporting the performance observations already made and details are given in tables D.12, D.15 and D.18. In scenario `02a`, although most of the parameter estimate spans reduce, a target uncertainty of 0.1 is reached (and maintained) only for few methods with parameter a, but for all methods in parameter b and σ_y. In all scenarios, the proposed methods reach a target uncertainty level of 0.1 earlier than the reference methods.

Runtime metric Δt_{run} varies greatly between methods as seen in figure 4.2.1, but computation runtime of a specific method remains stable across scenarios. In general, the reference methods are one to two orders of magnitude faster than the proposed methods. This is a direct consequence of the involved mathematics behind the methods. While the reference methods use gradient-based and computationally light update formulas that need to be evaluated just once per input data point, the proposed methods use computationally more demanding expressions (PDF-calculations) that are evaluated on a block of input data and need to be evaluated either multiple times (Block-Gibbs-Sampling) or for many parameter combinations (Discrete Grid). For the proposed methods, the quality of the results is directly related to the number of Monte Carlo runs or grid resolution, setting a limit to computational performance gains.

4.4.1.2 Streaming Performance (`03a`, `03b`, `03c`)

Streaming performance is assessed using scenarios `03a`, `03b` and `03c`, which differ only in the length of the incoming input blocks (variable length $1 - 400$, blocksize 100, blocksize 200). It is of interest, whether the method results are affected by the different splits and if there are runtime changes.

Method results are affected by the input data blocksize, except `joint_posterior`. Longer blocksizes yield better parameter estimates in terms of the $NMAE_a$ and $NMAE_b$ metrics. This is expected to a certain degree and a consequence of the block-wise data processing. The reason is that the Bernstein-von Mises theorem [119] does only apply to the `joint_posterior` and `joint_posterior_agrid` methods (method 2), where the full posterior information is maintained between update cycles within the internal state. In contrast - although not a general limitation of MCMC-based methods - the method-1-based methods discard correlations between the parameters because the result distributions are fitted against marginalized posterior samples. Hence, longer blocksizes allow to keep the correlation information over more input data points, resulting in better estimates. Additionally, only the analytical posteriors of a and b are of the same type as their priors, while the posterior for σ_y does not directly yield again an inverse gamma distribution (see appendix B.1.4). The `joint_posterior` and `joint_posterior_agrid` methods are based on analytically non-approximate calculations, with potential issues only arising from the grid update and numerical treatment. Inverse model performance (figures 4.2.8 to 4.2.10) and convergence performance (tables D.21, D.24 and D.27) are good for all proposed methods.

Runtime is affected by the input data blocksize. While runtime decreases with larger blocksize for methods that need to draw samples from the σ_y-posterior distribution (`gibbs_minimal`, `gibbs_no_EIV`), it increases for methods that do not (`gibbs_known_sigma_y`, `joint_posterior`, `joint_posterior_agrid`). This is expected, as the latter methods benefit from longer, but fewer blocks in total, reducing the total amount of required (computation intensive) samples from the σ_y-posterior distribution. Conversely, for the former methods, the main runtime influence is given by the error-in-variables model which requires matrix decompositions that scale worse-than-linear with blocksize.

Runtime and estimates are practically identical in all three scenarios for the reference methods. This is expected as the reference methods internally process each data point separately and calculations are therefore not affected by the choice of input blocks.

4.4.1.3 Dropout and Outlier Performance (`04a`, `04b`)

Performance in the presence of outliers and dropouts in the reference sensor measurements is assessed using scenarios `04a` and `04b`. The robustness against such communication errors is provided by the sensor fusion ahead of the actual parameter estimation routine, as described in chapter 3.1. Hence, multiple reference sensors are required to provide a basis for the statistics-based outlier detection.

In both scenarios, the proposed methods show similar performance as in scenarios without outliers (e.g., `02b`, `02c`). Although no test regarding a breakdown point of this performance is executed, from the design of the sensor fusion method it can be inferred, that this performance can be maintained, as long as less than half of the sensors are affected by outliers at a single point in time.

While the reference methods maintain good performance in case of dropouts, their performance breaks down in the presence of outliers. This can be best observed in the MSD_a and MSD_b metrics in figures 4.2.2 and 4.2.4.

4.4.1.4 Influence of Reference Sensor Uncertainty (`05a`, `05b`, `05c`)

Performance for different levels of uncertainty of the reference sensors is assessed using scenarios `05a`, `05b` and `05c`. These scenarios differ only in the uncertainty of the reference sensor reading and the added sensor noise based on this uncertainty level.

In scenarios `05a` and `05b` performance of the proposed methods stays on a similar and good level. However, the performance degrades in scenario `05c`, especially parameter a and inverse model performance can no longer be maintained consistently (see figures 4.2.3, 4.2.8 and 4.2.10). This indicates that the proposed methods can supply usable results even if a sensor with similar characteristics as the device under test is used as reference sensor. A sensor that performs worse than the device under test should not be used as reference sensor, although the (inverse) model for measurand estimation remains consistent within the stated uncertainty, mainly due to a high (and overestimated) σ_y.

For the reference methods, the performance is affected in scenario `05b` and `05c`. Only due to high estimated uncertainties of the parameter estimates, consistency can be maintained in all three scenarios.

4.4.1.5 No-noise Performance (`01a`, `01b`, `01c`)

Performance of the methods is tested in a way that deviates from the statistical model given in definition 21 by setting the error terms of equations (3.1.9) and (3.1.10) to zero in scenarios `01a`, `01b` and `01c`. These scenarios are otherwise equivalent to `02a`, `02b` and `02c`.

The proposed methods do not perform well in scenario `01a`, especially MSD_a and $NMAE_a$ deviate far from their ideal values (figures 4.2.2 and 4.2.3). This is due to missing variations in the input signal, which does not allow to distinguish effects of the parameter a and b properly. In scenario `01b` only the `gibbs_minimal` and `gibbs_known_sigma_y` methods perform well. The proposed methods fail to provide parameter uncertainties in scenario `01c`. The reason for these unsuccessful simulations are numerical instabilities from very narrow posterior distributions. These are reached because no noise or error is present in the data, which is a deviation from the actual statistical model underpinning these methods. An application of the proposed methods should therefore check in advance, how well the actual measurement data matches the assumed statistical model.

The parameter estimates of the reference methods are comparable to the corresponding results of `02a`, `02b` and `02c`. Hence, scenario `01a` is providing good results, while performance on non-stationary signals in scenarios `01b` and `01c` is degraded.

4.4.1.6 Further Observations

The results of the `joint_posterior` method show equal uncertainty levels in most of the scenarios. Closer inspection of the simulation results shows that the posterior distribution became too narrow to estimate a variance, in which case the algorithm falls back to the grid spacing as uncertainty quantification. Hence, the reported uncertainty level is not representative. For this reason, a variant of the method with auto-adjusting grid is introduced as `joint_posterior_agrid`.

To quantify, whether the reported parameter values are consistent within the limits of their stated uncertainty, the normalized estimation error metrics ($NMAE_a$, $NMAE_b$, $NMAE_{\sigma_y}$) are taken into account. These indicate a too narrow uncertainty (if $NMAE > 1.0$) for the proposed methods. Closer inspection reveals, that this is not an issue of the Bayes-methods themselves, but of the Laplace approximation, which only fits a Gaussian around the maximum of the posterior, leading to narrower reported uncertainty. Internally (see figure 3.1.2 and section 3.1.4), these methods use distributions with larger variances.

4.4.2 Semantic Insights

The results of the evaluation in chapter 4.3 are discussed and put into context. Furthermore, the general necessity and requirement of a semantic approach is discussed.

4.4.2.1 Discussion of the Evaluation

Based on the corpus comparison, it can be stated that a majority of the classes in the ontology are found in the chosen corpus. Terms covering state space systems, linear affine behavior and the mathematical notation of transfer behavior are not very well represented in the corpus. Moreover, certain subsuming terms introduced in the ontology are not present in the corpus (e.g., `QualitativeBehavior`). Nevertheless, the `trans` ontology can be called *complete* and *accurate* according result of the corpus comparison.

Redundancy is avoided by relying on and refining existing concepts of other ontologies whenever suitable. Although not all terms can be found in the chosen corpus, these terms are not irrelevant, but only out of scope in that specific corpus. The OntoClean-methodology identified no “attributes”[20], supporting the relevance of all established concepts. Overall, the absence of redundant or irrelevant terms indicates that the ontology is *concise*.

The intended competency questions and also additional queries can be answered (or decided) based on the two proposed ontologies and with that support already a wide range of use cases. Moreover, the design of the `trans` ontology allows straightforward addition of new transfer behaviors. Questions concerning the input-output-equivalence of two models are not answerable (by design). Overall, *adaptability* to non-anticipated use cases is given but limited.

No structural inconsistencies are disclosed by the OntoClean methodology. Providing the meta-properties rigidity, unity, identity and dependence together with the ontologies allows to carry the full intended meaning of each concept in an explicit way. Hence, both ontologies are *consistent* and *clear* in the use of their terms, from which users and developers can benefit.

The result of the evaluated relationship richness confirms the need of using an ontological structure to cover the considered knowledge, by indicating that a majority of the relations are not of hierarchical/subclass-type (for both ontologies) and therefore can not be represented by a pure taxonomy. In `scal` ontology, all introduced classes are annotated, leading to an excellent AR-value. However, the `trans` ontology is not document well and leaves room for improvement. The inheritance richness shows that the `scal` ontology is a broader knowledge representation, covering multiple concepts, but stays at a rather abstract level and does not focus on providing

[20] concepts with only weak meta-properties (`-R`, `-I`, `-U`, `-D`) [142]

class-subclass details. In contrast, the `trans` ontology is more specialized and detailed in capturing specific domain knowledge by specifying (on average) between one and two subclasses per class. This fits the intentions of both knowledge representations well and confirms the decisions made during the design process.

Moreover, by extracting these metrics (semi-)automatically it is shown that both ontologies are executable in a programmatic way. Furthermore, because both ontologies are designed within the OWL2 DL standard, it can be assumed that the *computations are efficient* as state-of-the-art tools and reasoners are available and optimized for this task.

4.4.2.2 Relevance of the Semantic Approach

Another important aspect is the general relevance of choosing a semantic approach. This should answer questions like: Is a semantic approach to represent metrological information in sensor networks in general a suitable and efficient approach? What benefits would a (hypothetical) solution without semantic methods have over the implemented semantic approach? In order to discuss this, general benefits and disadvantages of semantic approaches are discussed and a connection to the domain of the IoT and sensor networks is shown.

A driving idea of the IoT and Semantic Web is the reuse of information in multiple use cases and contexts [148, 149, 150]. Neither a semantic nor non-semantic approach can guarantee the reuse of existing concepts, descriptions or knowledge. However, a non-semantic approach would define a list of relevant properties for a specific use case, which cannot enable the reuse of these properties outside its intended scope without a manual mapping. In contrast, the chosen semantic representation opens up that possibility by referring to published knowledge representations that clearly, precisely and transparently provide the descriptions of its concepts and their relations.

Another desirable property for knowledge models in general is interoperability and typically involves that exchanged information can be unambiguously interpreted. In a non-semantic knowledge representation this is typically achieved by an agreement across all participants on a syntactic structure to exchange the knowledge. A benefit of following a semantic approach is that parties (or systems) can interoperate without such an explicit agreement, as the meaning of the communicated knowledge is transferred (or at least obtainable) as well. E.g. within a sensor network, two sensors might provide their measurement results using different unit systems. Because the used unit system is communicated along the actual unit information, a translation is automatically possible, if corresponding mappings have been provided in one or the other unit system (e.g., QUDT provides mappings to multiple unit systems for many relevant units). This however points to the issue of redundantly defined concepts or relations that coexist in multiple ontologies with overlapping goals and definitions, which is a side effect of the Semantic Web idea. This redundancy can affect the interoperability of a semantic approach, if not taken care of already at the definition level.

Semantic approaches require a good understanding of the semantic concepts, tools and domain relevant ontologies and often come with a (depending on the chosen ontology potentially substantial) overhead in the implementation. With that and as of now, skilled personal is required to implement and maintain such solutions [148, 149]. [21] These increased efforts of a semantic approach over an often (computationally) lighter, simpler and faster to implement non-semantic solution need good justifications. One justification is that the semantic representation helps to

[21] I.e. "semantic" means not necessarily easier - but the complexity is made more transparent and explicit to users and applications.

carry over the meaning of a concept to algorithms, skilled users and non-experts alike. Another reason for these additional efforts can come from scaling effects that result from the agreement-free interoperability. Once established, the expansion of a semantic approach is straightforward and communication thereof uses the same mechanisms that are already in place. Moreover, lightweight ontologies can help to produce a similar overhead as comparable non-semantic approaches.

The proposed method is designed with IIoT applications in mind and therefore interoperability plays a key role. Moreover, the additional communication overhead can be covered well within communication brokers like OPC-UA [151] and could be abstracted from the user in automated tools. Finally, a highly relevant aspect in the implemented semantic approach is the ability to perform semantic filtering, e.g., even a local subclass `custom:VerySpecificSensor` of `sosa:Sensor` will still match queries that search for sensors in the sense of `sosa:Sensor`.

4.4.3 Result Summary

The proposed co-calibration methods show clear advantages for time-depending input signals, outliers in the input data and regarding the achievable uncertainty of the parameter estimate. Moreover, an additional estimated parameter allows to distinguish the parameter uncertainty and the additional (gaussian) model error. Furthermore, the proposed methods perform well even if the uncertainty of the input data is equal to the model error of the device under test, which is an important observation regarding the adaption in IIoT environments and allows to co-calibrate against sensors with similar specifications. The two proposed special cases of method 1 provide some computational runtime gains, but perform worse in many of the tested scenarios. The preferred approach is therefore to optimize method parameters of both proposed base methods to achieve better computational efficiency. This has been done for `gibbs_base` (leading to `gibbs_minimal` with reduced Monte Carlo runs) and `joint_posterior` (leading to `joint_posterior_agrid` with fewer grid points, but adaptive grid). Overall, the `gibbs_minimal` and `joint_posterior_agrid` provide a suitable performance to computational effort ratio.

The computational runtime indicates potential issues for real-time execution on IIoT edge hardware. These can be solved by either running the methods at the cloud-level with appropriate computational capacities or transferring the method to a compiled programming language (e.g., C++). On a smaller scale, a noticeable runtime improvement would already be achievable by providing native support to draw samples from the posterior distribution of σ_y.

The reference methods outperform the proposed methods in case of static input signals and the computational runtime is significantly reduced by approximately two orders of magnitude. However, in the practically relevant case of time-dependent input signals, these benefits do not persist and parameter estimation performance is degraded for the reference methods. Moreover, outlier robustness is not on the level of the proposed methods. The GUM-based uncertainty evaluation of the reference methods shows in general high parameter uncertainties, leading to mostly consistent parameter estimations (but at a less restricted level than the proposed methods).

The ontology assessment shows a good agreement of the proposed ontologies to established criteria using existing evaluation methods. The proposed knowledge representations therefore build a solid base to provide sensor self-descriptions in metrological use cases, based on which the flow charts presented in chapter 3.3 can initialize the mathematical co-calibration method. Moreover, the general need for a semantic solution is discussed and favored with regard to the intended application domain.

Part V

Conclusion and Outlook

5.1 Conclusion

This chapter concludes this thesis by referring back to the research questions, summarizing the findings and highlighting the contributions made. It will also cover the limitations of the thesis with regard to its framing, approach and methods. Potential topics and questions for future work are provided in chapter 5.2.

5.1.1 Answering the Research Questions

Based on the research presented in this thesis, it is possible to provide answers to the research questions formulated in chapter 1.1.

Regarding the semantic representation of relevant metrological knowledge (RQ1), the proposed method of relying on existing knowledge representations and only adding specific missing concepts allows to represent metrological relevant knowledge about a sensor in a semantic expressive and machine-executable way. The latter part is established through techniques and methods of the Semantic Web. One result of the research is that all required concepts can be represented within the possibilities of the OWL2 DL, allowing to profit from available and efficient reasoners. An iteratively executable Bayesian parameter estimation method suited for linear affine sensor transfer behavior is developed, directly addressing the second research question (RQ2). It uses available uncertainty information from calibrated reference devices in an error-in-variables model for the inputs, estimates parameters with uncertainty of the transfer behavior and also quantifies a model error of this transfer behavior. Robustness with regard to the number of input sensors and outliers is achieved by relying on a robust sensor fusion. The traceability of the method is discussed in terms of agreement to established definitions of calibration, matching the definition provided in the VIM. To see how the semantic knowledge can be used to initialize the mathematical method (RQ3), a concept is provided. Semantic knowledge can be used to initialize a mathematical method by following the developed decision flowcharts. As the mathematical methods always require the same level of detail for initialization, information gaps can arise and need to be filled with low-informative assumptions. Hence, the provided flowcharts fill these gaps based on heuristics. Conditions that must be fulfilled in order to perform a successful co-calibration (RQ4) can be specified by a combination of the semantic queries (chapter 3.2) and the requirements of the mathematical method (chapter 3.1), alongside the findings of the evaluation (chapter 4.2). In summary, there must be at least one reference sensor qualifying the semantic queries, this sensor needs to provide a time-varying signal that matches the assumptions of the statistical model and moreover needs to have a lower or equal uncertainty compared to the (simulated) noise level of the device under test.

5.1.2 Significance, Implications and Contribution

This work considers two aspects that are essential to maintaining trust in large scale dense sensor networks, which have not been considered in this connected way before. These are the explicit consideration of metrological traceability of in-situ calibration operations and the use of existing knowledge that is available in a semantic expressive form. It is shown, what metrological relevant information was not representable within existing ontologies. These gaps are closed in a constructive approach and a sensor self-description making use of these new concepts is shown. Furthermore, this explicit machine-interpretable knowledge is used to find suitable reference sensors in a sensor network and develop a concept to initialize a co-calibration method based on this knowledge. The research indicated that the term "calibration" is often not used in the strict metrological sense within the sensor network literature. Therefore, the term co-calibration is introduced and a mathematical method that explicitly considers metrological traceability by evaluating the uncertainty of its results in a VIM-compatible way is developed. This method is designed to be used in homogeneous sensor networks and also considers streaming applicability, outlier robustness and inclusion of pre-knowledge. The evaluation shows good co-calibration results even in the case of reference sensors with comparable uncertainty characteristics as the device under test, which is of great value for applications in IIoT-environments. Furthermore, while evaluating the proposed method against the state of the art, it was found that the proposed method is suited for time-dependent input signals, while the state of the art reference method cannot yield a usable estimation quality for these non-stationary signals.

5.1.3 Limitations

In order to obtain the above results, certain limitations of the scope are necessary. Although the developed `trans` ontology supports a large selection of transfer behaviors, only **linear affine** transfer behavior is considered in the mathematical method. This decision was made in order to limit the amount of mathematical complexity and still maintain comparability to a majority of the state-of-the-art in-situ calibrations. In general, linear affine behavior is a valid and common model choice for many applications, but far from generic, as it, e.g., cannot cover dynamic or non-linear transfer behaviors. **Interpolation** of input data is a highly relevant topic within sensor networks because of a (typically) non-synchronized data acquisition. However, as methods for uncertainty-aware interpolation are available[22], the topic is considered out of scope in this thesis and the general assumption is, that the input data is already available on a common time-base. The iterative co-calibration is meant to operate on **streaming** input data. Although the method is designed to support this, the computational efforts involved currently prevent a true online / real-time application even on capable hardware. This might be less of a problem for cloud-level applications, but is a major concern for potential edge-level implementations. Depending on the temporal behavior of the input signal, the evaluated reference methods with added uncertainty propagation can provide a less computationally intensive alternative. The semantic and mathematical aspects of this thesis have been developed to jointly interact. However, the actual implementation considers each domain **separately**. The focus of this thesis is to develop a co-calibration method and evaluate it against a ground truth. This lead to the decision to perform a **simulation-based** evaluation that allows for a direct comparison against the true values and exact repeatability of the evaluation results. Although some parts of the method (sensor fusion) have been deployed on a real-world testbed, others (Bayesian update) have not.

[22] although with limitations, see section 3.1.1

5.2 Outlook

Many exciting continuations of the work in this thesis are possible and leave room for future research and projects. Although many of these suggestions target the mathematical co-calibration method, they go hand in hand with an extension of the semantics as well. In general, it would be very interesting to apply the full proposed method in a suitable and productive sensor network over an extended period of time.

The proposed co-calibration method could be extended to: (i) distributed homogeneous sensor networks (by means of spatio-temporal interpolation), (ii) heterogeneous sensor networks (by forming virtual traceable references from laws of physics or artificial intelligence) and (iii) support of different and more complex transfer behaviors. Another aspect are efficiency enhancements of the core Bayesian update. These are necessary in order to make it suitable for edge devices in production environments and could be achieved by developing a random sample generator for the less common posterior distributions used. Furthermore, recent developments [152, 153] hint at a revision of the GUM with a shift towards the Bayesian probability notion, which could enable an extension of theorem 10 in the direction of the ISO17025 standard [4].

It could be evaluated, how the co-calibration can be used to detect failure, drift or aging of single sensors. A potential way to achieved this could be based on inferring knowledge from cross-validation style co-calibrations on a routine schedule. Development over time of the estimated parameters can then be analyzed to decide whether a re-calibration or sensor replacement is required.

Regarding the semantic contributions of this thesis, it would be interesting to provide a semantic sound framework for the representation of formulas and mathematical knowledge. Such a framework would enable translation of mathematical representations into specific executable programming languages and contribute to the interoperability. Another important aspect is to allow different systems of units in the sensor self-description and making sure that they are compatible.

Bibliography

[1] Richard J. C. Brown. „Measuring Measurement – What Is Metrology and Why Does It Matter?“ In: *Measurement* 168 (Jan. 15, 2021), p. 108408. ISSN: 0263-2241. DOI: 10.1016/j.measurement.2020.108408.

[2] Sascha Eichstädt and Björn Ludwig. „Metrologie Für Heterogene Sensornetzwerke Und Industrie 4.0“. In: *tm - Technisches Messen* 86.11 (2019). ISSN: 2196-7113. DOI: 10.1515/teme-2019-0073.

[3] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. *International Vocabulary of Metrology - Basic and General Concepts and Associated Terms (VIM).* 2012.

[4] ISO. *DIN EN ISO/IEC 17025:2018-03, Allgemeine Anforderungen an Die Kompetenz von Prüf- Und Kalibrierlaboratorien (ISO/IEC_17025:2017).* Beuth Verlag GmbH, 2018. DOI: 10.31030/2731745.

[5] J.E.J. Gravel. *Principles Behind The Requirements Of ISO/IEC 17025.* 2002. URL: http://download.caltech.se/download/validering/diverse/17025/17025comparisons/ISO_IEC_17025_PRINCIPLES_2.pdf.

[6] Miloš S. Stanković, Srdjan S. Stanković, Karl Henrik Johansson, Marko Beko, and Luis M. Camarinha-Matos. „On Consensus-Based Distributed Blind Calibration of Sensor Networks“. In: *Sensors* 18.11 (Nov. 2018), p. 4027. DOI: 10.3390/s18114027.

[7] Emiliano Miluzzo, Nicholas D. Lane, Andrew T. Campbell, and Reza Olfati-Saber. „CaliBree: A Self-calibration System for Mobile Sensor Networks“. *Distributed Computing in Sensor Systems.* Ed. by Sotiris E. Nikoletseas, Bogdan S. Chlebus, David B. Johnson, and Bhaskar Krishnamachari. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 314–331. ISBN: 978-3-540-69170-9. DOI: 10.1007/978-3-540-69170-9_21.

[8] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. *Guide to the Expression of Uncertainty in Measurement.* 2008.

[9] Tim Berners-Lee, James Hendler, and Ora Lassila. „The Semantic Web“. In: *Scientific American* 284.5 (2001), pp. 34–43. ISSN: 0036-8733. URL: https://www.jstor.org/stable/26059207 (visited on 03/30/2022). JSTOR: 26059207.

[10] D. Rod White and Peter Saunders. „The Propagation of Uncertainty with Calibration Equations“. In: *Measurement Science and Technology* 18.7 (June 2007), pp. 2157–2169. ISSN: 0957-0233. DOI: 10.1088/0957-0233/18/7/047.

[11] D. Rod White. „Propagation of Uncertainty and Comparison of Interpolation Schemes“. In: *International Journal of Thermophysics* 38.3 (Jan. 5, 2017), p. 39. ISSN: 1572-9567. DOI: 10.1007/s10765-016-2174-6.

[12] Jens Jauch, Felix Bleimund, Stephan Rhode, and Frank Gauterin. „Recursive B-spline Approximation Using the Kalman Filter“. In: *Engineering Science and Technology, an International Journal* 20.1 (Feb. 1, 2017), pp. 28–34. ISSN: 2215-0986. DOI: 10.1016/j.jestch.2016.09.015.

[13] Sascha Eichstädt, Maximilian Gruber, Anupam Prasad Vedurmudi, Benedikt Seeger, Thomas Bruns, and Gertjan Kok. „Toward Smart Traceability for Digital Sensors and the Industrial Internet of Things“. In: *Sensors* 21.6 (6 Jan. 2021), p. 2019. ISSN: 1424-8220. DOI: 10.3390/s21062019.

[14] Maximilian Gruber, Tanja Dorst, Andreas Schütze, Sascha Eichstädt, and Clemens Elster. „Discrete Wavelet Transform on Uncertain Data: Efficient Online Implementation for Practical Applications“. *Advanced Mathematical and Computational Tools in Metrology and Testing XII.* Vol. Volume 90. Series on Advances in Mathematics for Applied Sciences Volume 90. WORLD SCIENTIFIC, May 31, 2021, pp. 249–261. ISBN: 9789811242373. DOI: 10.1142/9789811242380_0014.

[15] Tanja Dorst, Maximilian Gruber, Benedikt Seeger, Anupam Prasad Vedurmudi, Tizian Schneider, Sascha Eichstädt, and Andreas Schütze. „Uncertainty-Aware Data Pipeline of Calibrated MEMS Sensors Used for Machine Learning“. In: *Measurement: Sensors* 22 (Aug. 1, 2022), p. 100376. ISSN: 2665-9174. DOI: 10.1016/j.measen.2022.100376.

[16] Maximilian Gruber, Sascha Eichstädt, Wenzel Pilar von Pilchau, Jörg Hähner, Varun Gowtham, Alexander Willner, Nikolaos-Stefanos Koutrakis, Julian Polte, and Matthias Riedl. „Uncertainty-Aware Sensor Fusion in Sensor Networks“. In: *SMSI 2021 - Measurement Science* (May 3, 2021), pp. 246–247. DOI: 10.5162/SMSI2021/D2.2.

[17] Maximilian Gruber, Wenzel Pilar von Pilchau, Varun Gowtham, Nikolaos-Stefanos Koutrakis, Nicolas Schönborn, Sascha Eichstädt, Jörg Hähner, Marius-Iulian Corici, Thomas Magedanz, Julian Polte, and Eckart Uhlmann. „Application of Uncertainty-Aware Sensor Fusion in Physical Sensor Networks“. *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC).* 2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). May 2022, pp. 1–6. DOI: 10.1109/I2MTC48687.2022.9806580.

[18] Maximilian Gruber, Sascha Eichstädt, Julia Neumann, and Adrian Paschke. „Semantic Information in Sensor Networks: How to Combine Existing Ontologies, Vocabularies and Data Schemes to Fit a Metrology Use Case“. *2020 IEEE International Workshop on Metrology for Industry 4.0 IoT.* 2020 IEEE International Workshop on Metrology for Industry 4.0 IoT. June 2020, pp. 469–473. DOI: 10.1109/MetroInd4.0IoT48571.2020.9138282.

[19] Anupam Prasad Vedurmudi, Maximilian Gruber, Sascha Eichstädt, and Adrian Paschke. „Semantics in Sensor Networks: An Ontology for Dynamic Transfer Behavior in Calibrated Sensors“. *2021 IEEE International Workshop on Metrology for Industry 4.0 IoT (MetroInd4.0 IoT).* 2021 IEEE International Workshop on Metrology for Industry 4.0 IoT (MetroInd4.0 IoT). June 2021, pp. 358–363. DOI: 10.1109/MetroInd4.0IoT51437.2021.9488554.

[20] John W. Creswell and J. David Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches.* 5th ed. SAGE Publications, Inc, 2017. ISBN: 978-1-5063-8669-0.

[21] Maximilian Gruber, Sascha Eichstädt, and Clemens Elster. „Modeling Dynamic Measurements in Metrology and Propagation of Uncertainties“. *Dynamic Measuring Systems: Fundamentals and Application of Time-Dependent Measurements.* De Gruyter Series in Measurement Sciences. De Gruyter Oldenbourg, 2023.

[22] Sascha Eichstädt, ed. *Dynamic Measuring Systems: Fundamentals and Application of Time-Dependent Measurements.* De Gruyter Series in Measurement Sciences. De Gruyter Oldenbourg, 2023. ISBN: 978-3-11-071310-7. URL: https://www.degruyter.com/document/isbn/9783110713107/html (visited on 11/04/2022).

[23] *Gesetz Über Die Einheiten Im Messwesen Und Die Zeitbestimmung.* Bundesamts für Justiz, 2008. URL: https://www.gesetze-im-internet.de/me_einhg/index.html.

[24] BIPM. *The International System of Units (SI).* 2019.

[25] Miguel A. Martin-Delgado. „The New SI and the Fundamental Constants of Nature“. In: *European Journal of Physics* 41.6 (Oct. 2020), p. 063003. ISSN: 0143-0807. DOI: 10.1088/1361-6404/abab5e.

[26] Sheila Devasahayam. „An Overview of Internationally Integrated Nanotechnology Governance“. In: *International Journal of Metrology and Quality Engineering* 8 (Jan. 1, 2017). DOI: 10.1051/ijmqe/2017002.

[27] Imminent77. *English: A Pyramid Showing the Levels of Metrology along with the Way That Traceability and Calibrations Run.* Mar. 1, 2017. URL: https://commons.wikimedia.org/wiki/File:Traceability_Pyramid.png (visited on 05/03/2022).

[28] CODATA. *Digital Representation of Units of Measurement (DRUM).* CODATA, The Committee on Data for Science and Technology. 2022. URL: https://codata.org/initiatives/task-groups/drum/ (visited on 05/03/2022).

[29] *Regulation (EC) No 765/2008 of the European Parliament and of the Council of 9 July 2008 Setting out the Requirements for Accreditation and Market Surveillance Relating to the Marketing of Products and Repealing Regulation (EEC) No 339/93 (Text with EEA Relevance).* July 9, 2008. URL: http://data.europa.eu/eli/reg/2008/765/oj/eng (visited on 09/02/2022).

[30] Siegfried Hackel, Frank Härtig, Julia Hornig, and Thomas Wiedenhöfer. „The Digital Calibration Certificate“ (2017).

[31] Daniel Hutzschenreuter, Frank Härtig, Wiebke Heeren, Thomas Wiedenhöfer, Alistair Forbes, Clifford Brown, Ian Smith, Susan Rhodes, Ivana Linkeová, Jakub Sýkora, Vít Zelený, Bojan Ačko, Rok Klobučar, Pekka Nikander, Tommi Elo, Tuukka Mustapää, Petri Kuosmanen, Olaf Maennel, Kristine Hovhannisyan, Bernd Müller, Lukas Heindorf, and Vincenzo Paciello. „SmartCom Digital System of Units (D-SI) Guide for the Use of the Metadata-Format Used in Metrology for the Easy-to-Use, Safe, Harmonised and Unambiguous Digital Transfer of Metrological Data“ (2019). DOI: 10.5281/zenodo.3522631.

[32] Tim Berners-Lee, Roy Fielding, and Larry Masinter. *Uniform Resource Identifiers (URI): Generic Syntax.* 1998. URL: https://www.ietf.org/rfc/rfc2396.txt.

[33] Ben Adida, Mark Birbeck, Shane McCarron, and Steven Pemberton. *RDFa in XHTML: Syntax and Processing.* 2008. URL: https://www.w3.org/TR/rdfa-syntax/ (visited on 05/03/2022).

[34] Hugh Boyes, Bil Hallaq, Joe Cunningham, and Tim Watson. „The Industrial Internet of Things (IIoT): An Analysis Framework“. In: *Computers in Industry* 101 (Oct. 1, 2018), pp. 1–12. ISSN: 0166-3615. DOI: 10.1016/j.compind.2018.04.015.

[35] wikipedia. *Industrial Internet of Things. Wikipedia.* Apr. 22, 2022. URL: https://en.wikipedia.org/w/index.php?title=Industrial_internet_of_things&oldid=1084074679 (visited on 05/05/2022).

[36] Veena S. Chakravarthi. *Internet of Things and M2M Communication Technologies: Architecture and Practical Design Approach to IoT in Industry 4.0.* Cham: Springer International Publishing, 2021. ISBN: 978-3-030-79272-5. DOI: 10.1007/978-3-030-79272-5.

[37] Andreas Schütze and Nikolai Helwig. „Sensorik Und Messtechnik Für Die Industrie 4.0“. In: *tm - Technisches Messen* 84.5 (2017), pp. 310–319. ISSN: 0171-8096. DOI: 10.1515/teme-2016-0047.

[38] Shiting (Justin) Lu, Russell Tessier, and Wayne Burleson. „Collaborative Calibration of On-Chip Thermal Sensors Using Performance Counters“. Proceedings of the International Conference on Computer-Aided Design. ACM, May 11, 2012, pp. 15–22. ISBN: 978-1-4503-1573-9. DOI: 10.1145/2429384.2429388.

[39] Mert Bal. „Industrial Applications of Collaborative Wireless Sensor Networks: A Survey“. IEEE International Symposium on Industrial Electronics. June 1, 2014, pp. 1463–1468. ISBN: 978-1-4799-2399-1. DOI: 10.1109/ISIE.2014.6864830.

[40] Ahmad Ali, Yu Ming, Sagnik Chakraborty, and Saima Iram. „A Comprehensive Survey on Real-Time Applications of WSN“. In: *Future Internet* 9 (Nov. 7, 2017), p. 77. DOI: 10.3390/fi9040077.

[41] Rui Tan, Guoliang Xing, Xue Liu, Jianguo Yao, and Zhaohui Yuan. „Adaptive Calibration for Fusion-Based Cyber-Physical Systems“. In: *ACM Transactions on Embedded Computing Systems (TECS)* 11.4 (Jan. 12, 2012), p. 80. ISSN: 1539-9087. DOI: 10.1145/2362336.2362347.

[42] Rui Tan, Guoliang Xing, Zhaohui Yuan, Xue Liu, and Jianguo Yao. „System-Level Calibration for Data Fusion in Wireless Sensor Networks“. In: *ACM Transactions on Sensor Networks (TOSN)* 9.3 (Jan. 5, 2013), p. 28. ISSN: 1550-4859. DOI: 10.1145/2480730.2480731.

[43] S. R. Jino Ramson and D. Jackuline Moni. „Applications of Wireless Sensor Networks — A Survey“. *2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT).* 2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT). Feb. 2017, pp. 325–329. DOI: 10.1109/ICIEEIMT.2017.8116858.

[44] Jose M. Barcelo-Ordinas, Jorge Garcia-Vidal, Messaoud Doudou, Santiago Rodrigo-Muñoz, and Albert Cerezo-Llavero. „Calibrating Low-Cost Air Quality Sensors Using Multiple Arrays of Sensors“. *2018 IEEE Wireless Communications and Networking Conference (WCNC).* 2018 IEEE Wireless Communications and Networking Conference (WCNC). Apr. 2018, pp. 1–6. DOI: 10.1109/WCNC.2018.8377051.

[45] Jose M. Barcelo-Ordinas, Messaoud Doudou, Jorge Garcia-Vidal, and Nadjib Badache. „Self-Calibration Methods for Uncontrolled Environments in Sensor Networks: A Reference Survey“. In: *Ad Hoc Networks* 88 (May 15, 2019), pp. 142–159. ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2019.01.008.

[46] Florentin Delaine, Bérengère Lebental, and Hervé Rivano. „In Situ Calibration Algorithms for Environmental Sensor Networks: A Review“. In: *IEEE Sensors Journal* 19.15 (Aug. 2019), pp. 5968–5978. ISSN: 1558-1748. DOI: 10.1109/JSEN.2019.2910317.

[47] David Hasenfratz, Olga Saukh, and Lothar Thiele. „On-the-Fly Calibration of Low-Cost Gas Sensors“. Proceedings of the 9th European Conference on Wireless Sensor Networks. Springer-Verlag, Feb. 15, 2012, pp. 228–244. ISBN: 978-3-642-28168-6. DOI: 10.1007/978-3-642-28169-3_15.

[48] Hüseyin Akcan. „On the Complexity of Energy Efficient Pairwise Calibration in Embedded Sensors". In: *Applied Soft Computing* 13.4 (Apr. 1, 2013), pp. 1766–1773. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2013.01.013.

[49] Olga Saukh, David Hasenfratz, and Lothar Thiele. „Reducing Multi-Hop Calibration Errors in Large-Scale Mobile Sensor Networks". Proceedings of the 14th International Conference on Information Processing in Sensor Networks. ACM, Apr. 13, 2015, pp. 274–285. ISBN: 978-1-4503-3475-4. DOI: 10.1145/2737095.2737113.

[50] Zhan Li, Yuzhi Wang, Anqi Yang, and Huazhong Yang. „Drift Detection and Calibration of Sensor Networks". *2015 International Conference on Wireless Communications Signal Processing (WCSP)*. 2015 International Conference on Wireless Communications Signal Processing (WCSP). Oct. 2015, pp. 1–6. DOI: 10.1109/WCSP.2015.7341138.

[51] Yuzhi Wang, Anqi Yang, Zhan Li, Pengjun Wang, and Huazhong Yang. „Blind Drift Calibration of Sensor Networks Using Signal Space Projection and Kalman Filter". *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). Apr. 2015, pp. 1–6. DOI: 10.1109/ISSNIP.2015.7106904.

[52] Yuzhi Wang, Anqi Yang, Zhan Li, Xiaoming Chen, Pengjun Wang, and Huazhong Yang. „Blind Drift Calibration of Sensor Networks Using Sparse Bayesian Learning". In: *IEEE Sensors Journal* 16.16 (Aug. 2016), pp. 6249–6260. ISSN: 1530-437X, 1558-1748, 2379-9153. DOI: 10.1109/JSEN.2016.2582539.

[53] Yuzhi Wang, Anqi Yang, Xiaoming Chen, Pengjun Wang, Yu Wang, and Huazhong Yang. „A Deep Learning Approach for Blind Drift Calibration of Sensor Networks". In: *IEEE Sensors Journal* 17.13 (July 2017), pp. 4158–4171. ISSN: 1530-437X, 1558-1748, 2379-9153. DOI: 10.1109/JSEN.2017.2703885.

[54] Shaoming He, Hyo-Sang Shin, Shuoyuan Xu, and Antonios Tsourdos. „Distributed Estimation over a Low-Cost Sensor Network: A Review of State-of-the-Art". In: *Information Fusion* 54 (Feb. 1, 2020), pp. 21–43. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2019.06.026.

[55] David E. Williams. „Low Cost Sensor Networks: How Do We Know the Data Are Reliable?" In: *ACS Sensors* (Sept. 16, 2019). DOI: 10.1021/acssensors.9b01455.

[56] Reza Olfati Saber and Richard M. Murray. „Consensus Protocols for Networks of Dynamic Agents". *Proceedings of the 2003 American Control Conference, 2003*. Vol. 2. Piscataway, NJ: IEEE, June 2003, pp. 951–956. ISBN: 978-0-7803-7896-4. URL: https://resolver.caltech.edu/CaltechAUTHORS:20170522-153018041 (visited on 07/08/2020).

[57] Qiuxi Zhu, Francoise Sailhan, Md Yusuf Sarwar Uddin, Valérie Issarny, and Nalini Venkatasubramanian. „Multi-Sensor Calibration Planning in IoT-Enabled Smart Spaces". *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). July 2019, pp. 722–731. DOI: 10.1109/ICDCS.2019.00077.

[58] R. Olfati-Saber and R.M. Murray. „Consensus Problems in Networks of Agents with Switching Topology and Time-Delays". In: *IEEE Transactions on Automatic Control* 49.9 (Sept. 2004), pp. 1520–1533. ISSN: 1558-2523. DOI: 10.1109/TAC.2004.834113.

[59] Vladimir Bychkovskiy, Seapahn Megerian, Deborah Estrin, and Miodrag Potkonjak. „A Collaborative Approach to In-Place Sensor Calibration". *Information Processing in Sensor Networks*. Ed. by Feng Zhao and Leonidas Guibas. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2003, pp. 301–316. ISBN: 978-3-540-36978-3. DOI: 10.1007/3-540-36978-3_20.

[60] Laura Balzano and Robert Nowak. „Blind Calibration of Sensor Networks". *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*. IPSN '07. Cambridge, Massachusetts, USA: Association for Computing Machinery, Apr. 25, 2007, pp. 79–88. ISBN: 978-1-59593-638-7. DOI: 10.1145/1236360.1236372.

[61] Jielong Yang, Xionghu Zhong, and Wee Peng Tay. „A Dynamic Bayesian Nonparametric Model for Blind Calibration of Sensor Networks". In: *IEEE Internet of Things Journal* 5.5 (Oct. 2018), pp. 3942–3953. ISSN: 2327-4662. DOI: 10.1109/JIOT.2018.2847697.

[62] Maja S. Stanković and Dragan S. Antić. „Distributed Non-Linear Robust Consensus-Based Sensor Calibration for Networked Control Systems". In: *IET Control Theory & Applications* 14.9 (2020), pp. 1200–1208. ISSN: 1751-8652. DOI: 10.1049/iet-cta.2019.0672.

[63] Srdjan S. Stanković, Marko Beko, and Miloš S. Stanković. „Nonlinear Robustified Stochastic Consensus Seeking". In: *Systems & Control Letters* 139 (May 1, 2020), p. 104667. ISSN: 0167-6911. DOI: 10.1016/j.sysconle.2020.104667.

[64] Yuxiang Lin, Wei Dong, and Yuan Chen. „Calibrating Low-Cost Sensors by a Two-Phase Learning Approach for Urban Air Quality Measurement". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1 (Mar. 26, 2018), 18:1–18:18. DOI: 10.1145/3191750.

[65] Balz Maag, Zimu Zhou, and Lothar Thiele. „Enhancing Multi-hop Sensor Calibration with Uncertainty Estimates". *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. Aug. 2019, pp. 618–625. DOI: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00143.

[66] Clemens Elster and Alfred Link. „Uncertainty Evaluation for Dynamic Measurements Modelled by a Linear Time-Invariant System". In: *Metrologia* 45.4 (2008), pp. 464–473. ISSN: 0026-1394. DOI: 10.1088/0026-1394/45/4/013.

[67] Sascha Eichstädt, Natalia Makarava, and Clemens Elster. „On the Evaluation of Uncertainties for State Estimation with the Kalman Filter". In: *Measurement Science and Technology* 27.12 (Oct. 2016), p. 125009. ISSN: 0957-0233. DOI: 10.1088/0957-0233/27/12/125009.

[68] Sascha Eichstädt, Clemens Elster, Ian M. Smith, and Trevor J. Esward. „Evaluation of Dynamic Measurement Uncertainty – an Open-Source Software Package to Bridge Theory and Practice". In: *Journal of Sensors and Sensor Systems* 6.1 (2017), pp. 97–105. ISSN: 2194-8771. DOI: 10.5194/jsss-6-97-2017.

[69] Yun Cheng, Xiaoxi He, Zimu Zhou, and Lothar Thiele. „ICT: In-field Calibration Transfer for Air Quality Sensor Deployments". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.1 (Mar. 29, 2019), 6:1–6:19. DOI: 10.1145/3314393.

[70] John Lipor and Laura Balzano. „Robust Blind Calibration via Total Least Squares". *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). May 2014, pp. 4244–4248. DOI: 10.1109/ICASSP.2014.6854402.

[71] Laura Balzano and Robert Nowak. „Blind Calibration of Sensor Networks“. Proceedings of the 6th International Conference on Information Processing in Sensor Networks. ACM, Apr. 25, 2007, pp. 79–88. ISBN: 978-1-59593-638-7. DOI: 10.1145/1236360.1236372.

[72] Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter. *Handbook of Knowledge Representation*. San Diego, CA, USA: Elsevier Science, 2007. 1034 pp. ISBN: 978-0-444-52211-5.

[73] W3C OWL Working Group. *OWL 2 Web Ontology Language Document Overview (Second Edition)*. 2012. URL: https://www.w3.org/TR/2012/REC-owl2-overview-20121211/#Profiles (visited on 06/19/2022).

[74] Natasha Noy and Mark Musen. „PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment“. AAAI/IAAI. July 30, 2000. URL: https://www.semanticscholar.org/paper/PROMPT%3A-Algorithm-and-Tool-for-Automated-Ontology-Noy-Musen/f663c08950b28111b7a74db8b16ecfbb035d4dfa (visited on 08/28/2023).

[75] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. *OWL 2 Web Ontology Language Primer (Second Edition)*. 2012. URL: https://www.w3.org/TR/2012/REC-owl2-primer-20121211/ (visited on 06/19/2022).

[76] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. *OWL 2 Web Ontology Language Profiles (Second Edition)*. 2012. URL: https://www.w3.org/TR/2012/REC-owl2-profiles-20121211/ (visited on 06/19/2022).

[77] Birte Glimm, Ian Horrocks, Boris Motik, and Giorgos Stoilos. „Optimising Ontology Classification“. *The Semantic Web – ISWC 2010*. Ed. by Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks, and Birte Glimm. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 225–240. ISBN: 978-3-642-17746-0. DOI: 10.1007/978-3-642-17746-0_15.

[78] Steve Harris and Andy Seaborne. *SPARQL 1.1 Query Language*. 2013. URL: https://www.w3.org/TR/sparql11-query/ (visited on 06/19/2022).

[79] Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. „Cypher: An Evolving Query Language for Property Graphs“. *SIGMOD'18 Proceedings of the 2018 International Conference on Management of Data*. Houston, United States: ACM Press, June 2018, p. 1433. DOI: 10.1145/3183713.3190657.

[80] *ISO/IEC CD 39075*. ISO. URL: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/61/76120.html (visited on 06/19/2022).

[81] *GraphQL | A Query Language for Your API*. URL: https://graphql.org/ (visited on 06/19/2022).

[82] Marko A. Rodriguez. „The Gremlin Graph Traversal Machine and Language“. In: *Proceedings of the 15th Symposium on Database Programming Languages* (Oct. 27, 2015), pp. 1–10. arXiv: 1508.03843. DOI: 10.1145/2815072.2815073.

[83] Krzysztof Janowicz, Armin Haller, Simon J. D. Cox, Danh Le Phuoc, and Maxime Lefrancois. „SOSA: A Lightweight Ontology for Sensors, Observations, Samples, and Actuators“. In: *Journal of Web Semantics* 56 (May 2019), pp. 1–10. ISSN: 15708268. arXiv: 1805.09979. DOI: 10.1016/j.websem.2018.06.003.

[84] Armin Haller, Krzysztof Janowicz, Simon Cox, Maxime Lefrançois, Kerry Taylor, Danh Le Phuoc, Josh Lieberman, Raúl García-Castro, Rob Atkinson, and Claus Stadler. „The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation“ (2018). DOI: 10.3233/SW-180320.

[85] María Bermúdez-Edo, Tarek Elsaleh, Payam Barnaghi, and Kerry Taylor. „IoT-Lite: A Lightweight Semantic Model for the Internet of Things“. July 1, 2016. DOI: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0035.

[86] Hajo Rijgersberg, Mark van Assem, and Jan Top. „Ontology of Units of Measure and Related Concepts“. In: *Semantic Web – Interoperability, Usability, Applicability* (2011).

[87] Steven Ray. „Quantities, Units, Dimensions and Types (QUDT)“ (2011). DOI: doi.org/10.25504/FAIRsharing.d3pqw7.

[88] Open Geospatial Consortium. *GeoSPARQL - A Geographic Query Language for RDF Data.* 2012.

[89] David Carlisle, Patrick Ion, and Robert Miner. *Mathematical Markup Language (MathML) Version 3.0 2nd Edition.* W3C, 2014.

[90] Thomas R. Gruber and Gregory R. Olsen. „An Ontology for Engineering Mathematics“. *Principles of Knowledge Representation and Reasoning.* Ed. by Jon Doyle, Erik Sandewall, and Pietro Torasso. The Morgan Kaufmann Series in Representation and Reasoning. Morgan Kaufmann, Jan. 1, 1994, pp. 258–269. DOI: 10.1016/B978-1-4832-1452-8.50120-2.

[91] Blair D. Hall. „Software for Calculation with Physical Quantities“. *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT.* 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT. June 2020, pp. 458–463. DOI: 10.1109/MetroInd4.0IoT48571.2020.9138281.

[92] Robert Hanisch, Stuart Chalk, Romain Coulon, Simon Cox, Steven Emmerson, Francisco Javier Flamenco Sandoval, Alistair Forbes, Jeremy Frey, Blair Hall, Richard Hartshorn, Pascal Heus, Simon Hodson, Kazumoto Hosaka, Daniel Hutzschenreuter, Chu-Shik Kang, Susanne Picard, and Ryan White. „Stop Squandering Data: Make Units of Measurement Machine-Readable“. In: *Nature* 605.7909 (7909 May 2022), pp. 222–224. DOI: 10.1038/d41586-022-01233-w.

[93] Jon Wakefield. *Bayesian and Frequentist Regression Methods.* Springer, 2013. ISBN: 978-1-4939-3862-9. URL: https://link.springer.com/book/10.1007/978-1-4419-0925-1 (visited on 07/05/2022).

[94] George E. P. Box. *Bayesian Inference in Statistical Analysis.* New York: Wiley, 1992. DOI: 10.1002/9781118033197.

[95] Günter Bärwolff. *Höhere Mathematik Für Naturwissenschaftler Und Ingenieure.* 2017. URL: https://link.springer.com/book/10.1007/978-3-662-55022-9 (visited on 05/28/2022).

[96] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. *Supplement 1 to the "Guide to the Expression of Uncertainty in Measurement" - Propagation of Distributions Using a Monte Carlo Method.* 2008.

[97] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. *Supplement 2 to the "Guide to the Expression of Uncertainty in Measurement" – Extension to Any Number of Output Quantities.* 2011.

[98] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. *Guide to the Expression of Uncertainty in Measurement - Part 6: Developing and Using Measurement Models.* 2020.

[99] Alfred Link and Clemens Elster. „Uncertainty Evaluation for IIR (Infinite Impulse Response) Filtering Using a State-Space Approach“. In: *Meas. Sci. Technol.* 20.5 (2009). ISSN: 0957-0233. DOI: 10.1088/0957-0233/20/5/055104.

[100] Sascha Eichstädt. „Analysis of Dynamic Measurements“. Berlin: Technische Universität Berlin, 2012.

[101] Maurice G. Cox. „The Evaluation of Key Comparison Data“. In: *Metrologia* 39.6 (Dec. 2002), pp. 589–595. ISSN: 0026-1394. DOI: 10.1088/0026-1394/39/6/10.

[102] Florentin Delaine. „In Situ Calibration of Low-Cost Instrumentation for the Measurement of Ambient Quantities : Evaluation Methodology of the Algorithms and Diagnosis of Drifts“. PhD thesis. Institut Polytechnique de Paris, Dec. 4, 2020. URL: https://theses.hal.science/tel-03086234 (visited on 03/31/2023).

[103] Rik Pintelon and Johan Schoukens. *System Identification: A Frequency Domain Approach.* 2nd ed. MATLAB Examples. Piscataway, NJ: IEEE Press, 2012. xliv+743. ISBN: 978-0-470-64037-1.

[104] Hedde HWJ Bosman, Giovanni Iacca, Arturo Tejada, Heinrich J. Wörtche, and Antonio Liotta. „Spatial Anomaly Detection in Sensor Networks Using Neighborhood Information“. In: *Information Fusion* 33 (Jan. 1, 2017), pp. 41–56. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2016.04.007.

[105] *DIN EN ISO 10628-1: Schemata Für Die Chemische Und Petrochemische Industrie - Teil 1: Spezifikation Der Schemata.* Beuth, 2015. URL: https://dx.doi.org/10.31030/2158506.

[106] britannica. *Linguistics. The New Ecyclopædia Britannica.* 15th ed. Vol. 23. Ecyclopædia Britannica, Inc., 1998.

[107] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, eds. *The Description Logic Handbook: Theory, Implementation, and Applications.* USA: Cambridge University Press, 2003. 578 pp. ISBN: 978-0-521-78176-3.

[108] Camila Bezerra, Fred Freitas, and Filipe Santana. „Evaluating Ontologies with Competency Questions“. *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT).* 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). Vol. 3. Nov. 2013, pp. 284–285. DOI: 10.1109/WI-IAT.2013.199.

[109] Boris Motik, Peter F. Patel-Schneider, and Bernardo Cuenca Grau. *OWL 2 Web Ontology Language Direct Semantics (Second Edition).* 2012. URL: https://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/#Direct_Model-Theoretic_Semantics_for_OWL_2 (visited on 06/19/2022).

[110] Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition).* 2012. URL: https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/#Entities.2C_Literals.2C_and_Anonymous_Individuals (visited on 06/19/2022).

[111] Jörg Martin, Guido Bartl, and Clemens Elster. „Application of Bayesian Model Averaging to the Determination of Thermal Expansion of Single-Crystal Silicon“. In: *Measurement Science and Technology* 30.4 (Mar. 2019), p. 045012. ISSN: 0957-0233. DOI: 10.1088/1361-6501/ab094b.

[112] Giorgio Battistelli, Luigi Chisci, Claudio Fantacci, Alfonso Farina, and Antonio Graziano. „Consensus-Based Multiple-Model Bayesian Filtering for Distributed Tracking“. In: *IET Radar, Sonar & Navigation* 9.4 (2015), pp. 401–410. ISSN: 1751-8792. DOI: 10.1049/iet-rsn.2014.0071.

[113] Vahid Badeli, Sascha Ranftl, Gian Marco Melito, Alice Reinbacher-Köstinger, Wolfgang Von Der Linden, Katrin Ellermann, and Oszkar Biro. „Bayesian Inference of Multi-Sensors Impedance Cardiography for Detection of Aortic Dissection". In: *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering* 41.3 (Jan. 1, 2021), pp. 824–839. ISSN: 0332-1649. DOI: 10.1108/COMPEL-03-2021-0072.

[114] Wolfgang Koch. *Tracking and Sensor Data Fusion*. Mathematical Engineering. Berlin, Heidelberg: Springer, 2014. ISBN: 978-3-642-39270-2. DOI: 10.1007/978-3-642-39271-9.

[115] Petros Dellaportas and David A. Stephens. „Bayesian Analysis of Errors-in-Variables Regression Models". In: *Biometrics* 51.3 (1995), pp. 1085–1095. ISSN: 0006-341X. JSTOR: 2533007. DOI: 10.2307/2533007.

[116] Michael I. Jordan. „Graphical Models". In: *Statistical Science* 19.1 (Feb. 2004), pp. 140–155. ISSN: 0883-4237, 2168-8745. DOI: 10.1214/088342304000000026.

[117] Benjamin T. Vincent. „A Tutorial on Bayesian Models of Perception". In: *Journal of Mathematical Psychology* 66 (June 1, 2015), pp. 103–114. ISSN: 0022-2496. DOI: 10.1016/j.jmp.2015.02.001.

[118] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. 3rd ed. New York: Chapman and Hall/CRC, July 6, 2015. 675 pp. ISBN: 978-0-429-11307-9. DOI: 10.1201/b16018.

[119] Aad W. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press, 1998. ISBN: 978-0-521-78450-4. DOI: 10.1017/CBO9780511802256.

[120] Clemens Elster. „Bayesian Uncertainty Analysis Compared with the Application of the GUM and Its Supplements". In: *Metrologia* 51.4 (July 2014), S159. ISSN: 0026-1394. DOI: 10.1088/0026-1394/51/4/S159.

[121] Camila Bezerra, Fred Freitas, and Filipe Santana. „Evaluating Ontologies with Competency Questions". *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*. 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). Vol. 3. Nov. 2013, pp. 284–285. DOI: 10.1109/WI-IAT.2013.199.

[122] Jan Martin Keil and Sirko Schindler. „Comparison and Evaluation of Ontologies for Units of Measurement". In: *Semantic Web* 10 (Aug. 13, 2018), pp. 1–19. DOI: 10.3233/SW-180310.

[123] W3C. *Spatial Data on the Web Best Practices*. 2017. URL: https://www.w3.org/TR/sdw-bp/#dfn-spatial-data (visited on 01/23/2020).

[124] NIMA. *TR8350.2*. 1984. URL: https://www.wikidata.org/wiki/Q26732204 (visited on 08/01/2022).

[125] *Python.Org*. Python.org. Apr. 13, 2023. URL: https://www.python.org/ (visited on 04/14/2023).

[126] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. „Array Programming with NumPy". In: *Nature* 585.7825 (7825 Sept. 2020), pp. 357–362. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2649-2.

[127] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. „SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17.3 (3 Mar. 2020), pp. 261–272. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0686-2.

[128] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AmiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. „SymPy: Symbolic Computing in Python". In: *PeerJ Computer Science* 3 (Jan. 2, 2017), e103. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.103.

[129] John D. Hunter. „Matplotlib: A 2D Graphics Environment". In: *Computing in Science & Engineering* 9.3 (May 2007), pp. 90–95. ISSN: 1558-366X. DOI: 10.1109/MCSE.2007.55.

[130] Maximilian Gruber and Björn Ludwig. *Time-Series-Buffer 0.1.4b0 : This Package Provides Support for Buffering Time-Series with Uncertainty.* 2022. URL: https://github.com/PTB-PSt1/time-series-buffer (visited on 04/14/2023).

[131] Jean-Baptiste Lamy. „Owlready: Ontology-oriented Programming in Python with Automatic Classification and High Level Constructs for Biomedical Ontologies". In: *Artificial Intelligence in Medicine* 80 (July 1, 2017), pp. 11–28. ISSN: 0933-3657. DOI: 10.1016/j.artmed.2017.07.002.

[132] *Rdflib: RDFLib Is a Python Library for Working with RDF, a Simple yet Powerful Language for Representing Information.* Version 6.3.2. URL: https://github.com/RDFLib/rdflib (visited on 04/14/2023).

[133] Srdjan S. Stanković, Marko Beko, and Miloš S. Stanković. „A Robust Consensus Seeking Algorithm". *IEEE EUROCON 2019 -18th International Conference on Smart Technologies.* IEEE EUROCON 2019 -18th International Conference on Smart Technologies. July 2019, pp. 1–6. DOI: 10.1109/EUROCON.2019.8861907.

[134] Peter J. Bickel and Kjell A. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics, Volumes I-II Package.* New York: Chapman and Hall/CRC, Dec. 8, 2015. 1065 pp. ISBN: 978-1-315-36926-6. DOI: 10.1201/9781315369266.

[135] Joe Raad and Christophe Cruz. „A Survey on Ontology Evaluation Methods". *Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management.* Lisbonne, Portugal, Nov. 2015. DOI: 10.5220/0005591001790186.

[136] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals & Systems (2nd Ed.)* USA: Prentice-Hall, Inc., 1996. 957 pp. ISBN: 978-0-13-814757-0.

[137] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. „YAKE! Keyword Extraction from Single Documents Using Multiple Local Features". In: *Information Sciences* 509 (Jan. 1, 2020), pp. 257–289. ISSN: 0020-0255. DOI: 10.1016/j.ins.2019.09.013.

[138] William Winkler. „The State of Record Linkage and Current Research Problems“. In: *Statist. Med.* 14 (Oct. 13, 1999).

[139] Bezerra. *CQChecker: A Tool to Check Ontologies in OWL-DL using Competency Questions written in Controlled Natural Language.* Learning and NonLinear Models. 2014. URL: `https://sbic.org.br/lnlm/publicacoes/vol12-no2/vol12-no2-art4/` (visited on 12/12/2022).

[140] DIN and DKE. *Scenarios for Digitizing Standardization and Standards.* 2022.

[141] Nicola Guarino and Christopher Welty. „Evaluating Ontological Decisions with Ontoclean“. In: *Communications of the ACM* 45 (Feb. 1, 2002), pp. 61–65.

[142] Nicola Guarino and Christopher Welty. „An Overview of OntoClean“. May 22, 2009, pp. 201–220. DOI: `10.1007/978-3-540-92673-3_9`.

[143] Christopher Welty and William Andersen. „Towards OntoClean 2.0: A Framework for Rigidity.“ In: *Applied Ontology* 1 (Jan. 1, 2005), pp. 107–116.

[144] Zola Mahlaza and C. Maria Keet. *OntoClean in OWL with a DL Reasoner – A Tutorial.* 2019. DOI: `10.1007/978-3-642-17746-0_15`.

[145] Boris Motik, Rob Shearer, and Ian Horrocks. „Hypertableau Reasoning for Description Logics“. In: *Journal of Artificial Intelligence Research* 36 (Oct. 28, 2009), pp. 165–228. ISSN: 1076-9757. DOI: `10.1613/jair.2811`.

[146] Mark A. Musen. „The Protégé Project: A Look Back and a Look Forward“. In: *AI matters* 1.4 (June 2015), pp. 4–12. ISSN: 2372-3483. pmid: 27239556. DOI: `10.1145/2757001.2757003`.

[147] Samir Tartir, Ismailcem Arpinar, Michael Moore, Amit Sheth, and Boanerges Aleman-Meza. „OntoQA: Metric-Based Ontology Quality Analysis“. Nov. 27, 2005.

[148] Martin Bauer, Hamza Baqa, Sonia Bilbao, Aitor Corchero, Laura Daniele, Iker Esnaola-Gonzalez, Izaskun Fernandez, Östen Frånberg, Raúl García Castro, Marc Girod-Genet, Patrick Guillemin, Amelie Gyrard, Charbel El Kaed, Antonio Kung, Jaeho Lee, Maxime Lefrançois, Wenbin Li, Dave Raggett, and Michelle Wetterwald. *Semantic IoT Solutions - A Developer Perspective.* Oct. 22, 2019. DOI: `10.13140/RG.2.2.16339.53286`.

[149] Komal Gilani, Jaho Kim, JaeSeung Song, Dale Seed, and Chonggang Wang. „Semantic Enablement in IoT Service Layers—Standard Progress and Challenges“. In: *IEEE Internet Computing* 22.4 (July 2018), pp. 56–63. ISSN: 1941-0131. DOI: `10.1109/MIC.2018.043051465`.

[150] Soulakshmee Devi Nagowah, Hatem Ben Sta, and Baby Ashwin Gobin-Rahimbux. „An Overview of Semantic Interoperability Ontologies and Frameworks for IoT“. *2018 Sixth International Conference on Enterprise Systems (ES).* 2018 Sixth International Conference on Enterprise Systems (ES). Oct. 2018, pp. 82–89. DOI: `10.1109/ES.2018.00020`.

[151] OPC Foundation. *UA Part 1: Overview and Concepts.* Nov. 1, 2022. URL: `https://reference.opcfoundation.org/Core/Part1/v105/docs/` (visited on 08/03/2023).

[152] Ignacio Lira. „The GUM Revision: The Bayesian View toward the Expression of Measurement Uncertainty“. In: *European Journal of Physics* 37.2 (Feb. 2016), p. 025803. ISSN: 0143-0807. DOI: `10.1088/0143-0807/37/2/025803`.

[153] Robin Willink. „On Revision of the Guide to the Expression of Uncertainty in Measurement: Proofs of Fundamental Errors in Bayesian Approaches“. In: *Measurement: Sensors* 24 (Dec. 1, 2022), p. 100416. ISSN: 2665-9174. DOI: `10.1016/j.measen.2022.100416`.

[154] Tanja Dorst, Maximilian Gruber, Anupam P. Vedurmudi, Daniel Hutzschenreuter, Sascha Eichstädt, and Andreas Schütze. „A Case Study on Providing FAIR and Metrologically Traceable Data Sets“. In: *Acta IMEKO* 12.1 (1 Mar. 28, 2023), pp. 1–6. ISSN: 2221-870X. DOI: 10.21014/actaimeko.v12i1.1401.

[155] Sascha Eichstädt, Anupam Prasad Vedurmudi, Maximilian Gruber, and Daniel Hutzschenreuter. „Fundamental Aspects in Data Analysis for Sensor Network Metrology“. *Proceedings of the First International IMEKO TC6 Conference on Metrology and Digital Transformation - M4Dconf2022*. First International IMEKO TC6 Conference on Metrology and Digital Transformation. Berlin, GERMANY: IMEKO, 2022, pp. 1–4. DOI: 10.21014/tc6-2022.030.

[156] Maximilian Gruber and Sascha Eichstädt. „Representing Semantic Information in Sensor Networks“. In: *SMSI 2021 - System of Units and Metreological Infrastructure* (May 3, 2021), pp. 316–317. DOI: 10.5162/SMSI2021/D1.2.

[157] Maximilian Gruber, Sascha Eichstädt, and Anupam P. Vedurmudi. „Co-Calibration in Distributed Homogeneous Sensor Networks“. In: *Lectures* (May 8, 2023), pp. 47–48. DOI: 10.5162/SMSI2023/A3.2.

[158] Wenzel Pilar von Pilchau, Varun Gowtham, Maximilian Gruber, Matthias Riedl, Nikolaos-Stefanos Koutrakis, Jawad Tayyub, Jörg Hähner, Sascha Eichstädt, Eckart Uhlmann, Julian Polte, Volker Frey, and Alexander Willner. „An Architectural Design for Measurement Uncertainty Evaluation in Cyber-Physical Systems“. *Annals of Computer Science and Information Systems*. Position Papers of the 2020 Federated Conference on Computer Science and Information Systems. Vol. 22. 2020, pp. 53–57. ISBN: 978-83-959183-0-8. URL: https://annals-csis.org/proceedings/2020/drp/203.html (visited on 04/28/2023).

[159] Anupam Prasad Vedurmudi, Julia Neumann, Maximilian Gruber, and Sascha Eichstädt. „Semantic Description of Quality of Data in Sensor Networks“. In: *Sensors* 21.19 (19 Jan. 2021), p. 6462. ISSN: 1424-8220. DOI: 10.3390/s21196462.

Part VI

Annex

A Uncertainty Evaluation for the Stankovic Method and Linear Affine Models

To evaluate the performance of the co-calibration methods presented in chapter 3.1, a method of the state of the art developed by Stanković et al. is presented in detail and extended with GUM-compliant uncertainty evaluation for the estimated parameters [6, 133].

Sensors with linear affine input-output behavior are considered. The real transfer model with parameters (α_j, β_j) are unknown. From a calibration the estimated compensation model is known and given by parameters (a_j, b_j) of another affine linear model. The inverse of the compensation model is the estimated transfer model.

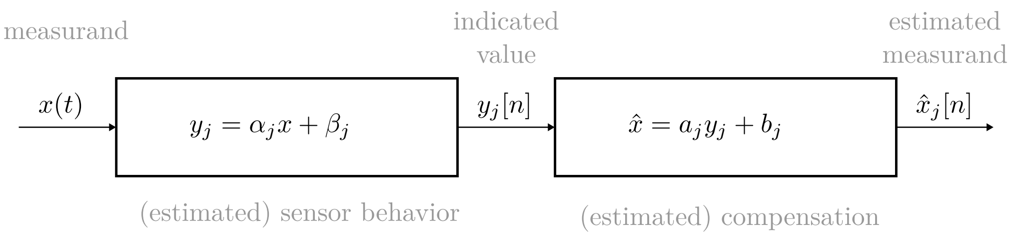

Figure A.1: Assumed system structure for estimation.

The scalar input signal $x(t)$ ((bounded) stochastic stationary process) is measured by a network $\mathcal{N}$ of sensors. The i-th sensor measures $y_j(t) = \alpha_j * x(t) + \beta_j$. To estimate the original input, the inverse model $\hat{x}_j = a_j * y_j(t) + b_j$ is required.

For new a sensor i the parameters (a_i, b_i) are unknown and can be estimated from the neighbors $\mathcal{N}_i$ of the new sensor. This can be achieved by the proposed update equation from [6], which is based on a gradient optimization scheme:

$$\underline{\theta}[n+1] = \underline{\theta}[n] + \delta(t) * \underline{\nabla} J \tag{A.1}$$

With

$$\underline{\theta} = [a_i, b_i]^T \tag{A.2}$$

$$\begin{aligned}\underline{\nabla} J &= \sum_{j\in\mathcal{N}_i} \gamma_{ij} \mathbb{E}\left\{ (\hat{x}_j[n] - \hat{x}_i[n]) * \begin{bmatrix} y_i[n-d] \\ 1 \end{bmatrix} \right\} \\ &= \sum_{j\in\mathcal{N}_i} \gamma_{ij} \begin{bmatrix} -a_i[n] * y_i[n-d]^2 + (\hat{x}_j[n] - b_i[n]) * y_i[n-d] \\ -a_i[n] * y_i[n-d] + (\hat{x}_j[n] - b_i[n]) \end{bmatrix}\end{aligned} \tag{A.3}$$

and weights $\gamma_{ij} = 1$ for the baseline method or $\gamma_{ij} \propto \frac{1}{u(\hat{x}_j[n])}$ to achieve an uncertainty-weighted gradient calculation (to which sensors with smaller uncertainty contribute more).

A.1 Uncertainty of the Parameter Update

The uncertainty of the parameter update proposed by Stankovic and reproduced in equation (A.1) is given by theorem 15.

Theorem 15 (Uncertainty of Parameter Update)**.** *The uncertainty of* $\underline{\theta}[n+1]$ *as used in equation* (A.1) *is given by*

$$U_{\underline{\theta}[n+1]} = C * U * C^T \tag{A.4}$$

with covariance of the inputs U *and sensitivities* C

$$U = \begin{bmatrix} u(a_i[n])^2 & u(a_i[n], b_i[n]) & 0 & 0 & \\ u(a_i[n], b_i[n]) & u(b_i[n])^2 & 0 & 0 & \cdots \\ 0 & 0 & u(y_i[n-d])^2 & 0 & \\ 0 & 0 & 0 & u(\hat{x}_j[n])^2 & \\ & \vdots & & & \ddots \end{bmatrix} \tag{A.5}$$

$$C = \begin{bmatrix} \frac{\partial a_i[n+1]}{\partial a_i[n]} & \frac{\partial a_i[n+1]}{\partial b_i[n]} & \frac{\partial a_i[n+1]}{\partial y_i[n]} & \frac{\partial a_i[n+1]}{\partial \hat{x}_j[n]} \\ \frac{\partial b_i[n+1]}{\partial a_i[n]} & \frac{\partial b_i[n+1]}{\partial b_i[n]} & \frac{\partial b_i[n+1]}{\partial y_i[n]} & \frac{\partial b_i[n+1]}{\partial \hat{x}_j[n]} \end{bmatrix} \tag{A.6}$$

and

$$\frac{\partial a_i[n+1]}{\partial a_i[n]} = 1 + \delta(t) * \sum_{j \in \mathcal{N}_i} \gamma_{ij} * (-y_i[n-d]^2) \tag{A.7}$$

$$\frac{\partial a_i[n+1]}{\partial b_i[n]} = 0 + \delta(t) * \sum_{j \in \mathcal{N}_i} \gamma_{ij} * (-y_i[n-d]) \tag{A.8}$$

$$\frac{\partial a_i[n+1]}{\partial y_i[n]} = 0 + \delta(t) * \sum_{j \in \mathcal{N}_i} \gamma_{ij} * (-2a_i[n] * y_i[n-d] + \hat{x}_j[n] - b_i[n]) \tag{A.9}$$

$$\frac{\partial a_i[n+1]}{\partial \hat{x}_j[n]} = 0 + \delta(t) * \gamma_{ij} * y_i[n-d] \tag{A.10}$$

$$\frac{\partial b_i[n+1]}{\partial a_i[n]} = 0 + \delta(t) * \sum_{j \in \mathcal{N}_i} \gamma_{ij} * (-y_i[n-d]) \tag{A.11}$$

$$\frac{\partial b_i[n+1]}{\partial b_i[n]} = 1 + \delta(t) * \sum_{j \in \mathcal{N}_i} \gamma_{ij} * (-1) \tag{A.12}$$

$$\frac{\partial b_i[n+1]}{\partial y_i[n]} = 0 + \delta(t) * \sum_{j \in \mathcal{N}_i} \gamma_{ij} * (-a_i[n]) \tag{A.13}$$

$$\frac{\partial b_i[n+1]}{\partial \hat{x}_j[n]} = 0 + \delta(t) * \gamma_{ij} \tag{A.14}$$

Proof. The update procedure defined by equation (A.1) can be written to fit the GUM-8 (GUM S2) [97] formalism $\underline{Y} = \underline{f}(\underline{X})$ by choosing

$$\underline{X} = \left[a_i[n], b_i[n], y_i[n-d], \underbrace{\hat{x}_j[n], \ldots}_{j \in \mathcal{N}_i} \right]^T \tag{A.15}$$

$$\underline{Y} = \underline{\theta}[n+1] \tag{A.16}$$

$$\underline{f} = \text{see equation (A.1)} \tag{A.17}$$

Then apply GUM-8 [8], i.e., calculate sensitivity (first order partial derivatives) and construct relevant matrices. Note that U contains information from $U_{\underline{\theta}[n]}$ and $\delta(t)$ is assumed to be deterministic (e.g., constant). Uncertainty of $\hat{x}_j$ is obtained by propagating an input through a linear affine model with uncertain parameters, as described in theorem 17. $\square$

A.2 The enhanced Stankovic algorithm

Stanković et al. also proposes an enhanced version that is better suited for noisy measurements y_i with known variance $\sigma_y^2 = u_y^2$ [6]. The central parameter update equation is given by

$$\underline{\theta}[n+1] = \begin{bmatrix} 1+c & 0 \\ 0 & 1 \end{bmatrix} \underline{\theta}[n] + \delta(t) * \underline{\nabla} J \tag{A.18}$$

$$c = \delta(t)\sigma_y^2 \sum_{j \in \mathcal{N}_i} \gamma_{ij} \tag{A.19}$$

This results in a change from the sensitivity stated in theorem 15 and is summarized as theorem 16.

Theorem 16 (Uncertainty of Enhanced Parameter Update)**.** *The uncertainty of $\underline{\theta}[n+1]$ as used in equation* (A.18) *is given in analogy to theorem 15, but with*

$$\frac{\partial a_i[n+1]}{\partial a_i[n]} = (1+c) + \delta(t) * \sum_{j \in \mathcal{N}_i} \gamma_{ij} * (-y_i[n-d]^2) \tag{A.20}$$

replacing equation (A.7).

Proof. Apply GUM-8 [8] in analogy to theorem 15. □

A.3 Uncertainty of Output of Linear Affine Transfer-Function

The uncertainty of the output of a linear affine model is given by theorem 17.

Theorem 17 (Uncertainty of Linear Affine Output)**.** *The output uncertainty u_y of a linear affine model with parameters (a, b) and input x is given by*

$${u_y}^2 = C * U * C^T \tag{A.21}$$

with

$$U = \begin{bmatrix} u(a)^2 & u(a,b) & 0 \\ u(a,b) & u(b)^2 & 0 \\ 0 & 0 & u(x)^2 \end{bmatrix} \tag{A.22}$$

$$C = \begin{bmatrix} x & 1 & a \end{bmatrix} \tag{A.23}$$

Proof. Fit the linear affine model $y = a * x + b$ to the GUM-formalism $\underline{Y} = \underline{f}(\underline{X})$ by choosing

$$\underline{X} = [a, b, x]^T \tag{A.24}$$

$$\underline{Y} = [y] \tag{A.25}$$

$$\underline{f} = [a * x + b] \tag{A.26}$$

Then apply GUM-8 [8], i.e., calculate sensitivity (first order partial derivatives) and construct relevant matrices. □

A.4 Uncertainty of the estimated inverse Model-Parameters

The algorithm of Stankovic [6] estimates the compensation model (see figure A.1). However, the actual sensor behavior is of interest as well. The inverse of a linear affine model is again linear affine, but the parameter uncertainties need some attention. Inverting the model given in definition 20 yields:

$$x = \frac{1}{a} * y - \frac{b}{a} \tag{A.27}$$

$$= a_{inv} * y + b_{inv} \tag{A.28}$$

The parameters of the inverse model are a function of the initial model. The parameter uncertainty of this inverse model is then given by theorem 18.

Theorem 18 (Uncertainty of Inverse Linear Affine Model Parameters). *The covariance U_{inv} of the parameters of the inverse of a linear affine model with parameters (a, b) is given by*

$$U_{inv} = C * U * C^T \tag{A.29}$$

with

$$U = \begin{bmatrix} u(a)^2 & u(a,b) \\ u(a,b) & u(b)^2 \end{bmatrix} \tag{A.30}$$

$$C = \begin{bmatrix} -\frac{1}{a^2} & 0 \\ \frac{b}{a^2} & -\frac{1}{a} \end{bmatrix} \tag{A.31}$$

Proof. The inverse model parameters (gain and offset) are given by, e.g., reorganizing the equation given in definition 20 for x:

$$[a_{inv}, b_{inv}]^T = [\frac{1}{a}, -\frac{b}{a}]^T \tag{A.32}$$

This relation fits the GUM-formalism $\underline{Y} = \underline{f}(\underline{X})$ by choosing

$$\underline{X} = [a, b]^T \tag{A.33}$$

$$\underline{Y} = [a_{inv}, b_{inv}]^T \tag{A.34}$$

$$\underline{f} = [\frac{1}{a}, -\frac{b}{a}]^T \tag{A.35}$$

Then apply GUM-8 [8], i.e., calculate sensitivity (first order partial derivatives) and construct relevant matrices. □

B Calculations of Posterior Distributions

B.1 Posteriors for Block-Gibbs-Sampling

B.1.1 Posterior for $\underline{X}_a$

Evaluate equation (3.1.20).

$$
\begin{aligned}
&p(\underline{X}_a|\underline{\theta}, \sigma_y, \sigma_x, \underline{Y}, \underline{X}_o) \\
&\propto p(\underline{Y}|\underline{X}_a, \underline{\theta}, \sigma_y)p(\underline{X}_a|\sigma_x, \underline{X}_o) && \text{(B.1)}\\
&\propto \exp\left\{-\frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(Y_i - aX_{ai} - b)^2\right\}\cdot\exp\left\{-\frac{1}{2}(\underline{X}_a - \underline{X}_o)^T\boldsymbol{U_x}^{-1}(\underline{X}_a - \underline{X}_o)\right\} && \text{(B.2)}\\
&\propto \exp\left\{-\frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(a^2X_{ai}^2 + 2abX_{ai} - 2aY_iX_{ai} + Y_i^2 + b^2 - 2bY_i)\right\}\\
&\quad\cdot\exp\left\{-\frac{1}{2}(\underline{X}_a{}^T\boldsymbol{U_x}^{-1}\underline{X}_a - \underline{X}_a{}^T\boldsymbol{U_x}^{-1}\underline{X}_o - \underline{X}_o{}^T\boldsymbol{U_x}^{-1}\underline{X}_a + \underline{X}_o{}^T\boldsymbol{U_x}^{-1}\underline{X}_o)\right\} && \text{(B.3)}\\
&\propto \exp\left\{-\frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(a^2X_{ai}^2 + 2a(Y_i - b)X_{ai})\right\}\\
&\quad\cdot\exp\left\{-\frac{1}{2}(\underline{X}_a{}^T\boldsymbol{U_x}^{-1}\underline{X}_a - \underline{X}_a{}^T\boldsymbol{U_x}^{-1}\underline{X}_o - \underline{X}_o{}^T\boldsymbol{U_x}^{-1}\underline{X}_a)\right\} && \text{(B.4)}\\
&\propto \exp\left\{-\frac{1}{2}(\underline{X}_a{}^T\boldsymbol{F_1}\underline{X}_a - \underline{X}_a{}^T\underline{F}_2 - \underline{F}_2{}^T\underline{X}_a)\right\}\\
&\quad\cdot\exp\left\{-\frac{1}{2}(\underline{X}_a{}^T\boldsymbol{U_x}^{-1}\underline{X}_a - \underline{X}_a{}^T\boldsymbol{U_x}^{-1}\underline{X}_o - \underline{X}_o{}^T\boldsymbol{U_x}^{-1}\underline{X}_a)\right\} && \text{(B.5)}\\
&\propto \exp\left\{-\frac{1}{2}(\underline{X}_a{}^T\underbrace{(\boldsymbol{F_1} + \boldsymbol{U_x}^{-1})}_{\boldsymbol{V}^{-1}}\underline{X}_a - \underline{X}_a{}^T(\underline{F}_2 + \boldsymbol{U_x}^{-1}\underline{X}_o) - (\underline{F}_2{}^T + \underline{X}_o{}^T\boldsymbol{U_x}^{-1})\underline{X}_a)\right\} && \text{(B.6)}\\
&\propto \exp\left\{-\frac{1}{2}(\underline{X}_a{}^T\boldsymbol{V}^{-1}\underline{X}_a - \underline{X}_a{}^T\boldsymbol{V}^{-1}\underbrace{\boldsymbol{V}(\boldsymbol{U_x}^{-1}\underline{X}_o + \underline{F}_2)}_{\underline{M}} - \underline{M}^T\boldsymbol{V}^{-1}\underline{X}_a)\right\} && \text{(B.7)}\\
&\propto \exp\left\{-\frac{1}{2}(\underline{X}_a - \underline{M})^T\boldsymbol{V}^{-1}(\underline{X}_a - \underline{M})\right\} && \text{(B.8)}
\end{aligned}
$$

The following matrices are introduced to transform the sum into a matrix operation:

$$\boldsymbol{F_1} = \frac{a^2}{\sigma_y^2} \cdot \boldsymbol{I_N} \tag{B.9}$$

$$\underline{F_2} = \frac{a}{\sigma_y^2} \begin{bmatrix} Y_1 - b & \dots & Y_N - b \end{bmatrix}^T \tag{B.10}$$

B.1.2 Posterior for a

Evaluate equation (3.1.21) for $\theta_i = a$.

$$p(a|b, \underline{X}_a, \sigma_y, \underline{Y})$$

$$\propto p(\underline{Y}|\underline{X}_a, \underline{\theta}, \sigma_y) \underbrace{p(a|b, \underline{X}_a, \sigma_y)}_{\mathcal{N}(\mu_a, \sigma_a^2)} \tag{B.11}$$

$$\propto \exp\left\{-\frac{1}{2\sigma_y^2} \sum_{i=1}^{N} (Y_i - aX_{ai} - b)^2\right\} \cdot \exp\left\{-\frac{1}{2\sigma_a^2}(a - \mu_a)^2\right\} \tag{B.12}$$

$$\propto \exp\left\{-\frac{1}{2\sigma_y^2} \sum_{i=1}^{N}(a^2X_{ai}^2 + 2abX_{ai} - 2aY_iX_{ai} + Y_i^2 + b^2 - 2bY_i) - \frac{1}{2\sigma_a^2}(a^2 - 2a\mu_a - \mu_a^2)\right\} \tag{B.13}$$

$$\propto \exp\left\{-\frac{1}{2\sigma_y^2} \sum_{i=1}^{N}(a^2X_{ai}^2 + 2a(b - Y_i)X_{ai}) - \frac{1}{2\sigma_a^2}(a^2 - 2a\mu_a)\right\} \tag{B.14}$$

$$\propto \exp\left\{-2a \underbrace{\left[\sum_{i=1}^{N} \frac{(b - Y_i)X_{ai}}{2\sigma_y^2} - \frac{\mu_a}{2\sigma_a^2}\right]}_{B} + a^2 \underbrace{\left[-\sum_{i=1}^{N} \frac{X_{ai}^2}{2\sigma_y^2} - \frac{1}{2\sigma_a^2}\right]}_{A}\right\} \tag{B.15}$$

$$\propto \exp\left\{A(a - \frac{B}{A})^2\right\} \tag{B.16}$$

B.1.3 Posterior for b

Evaluate equation (3.1.21) for $\theta_i = b$.

$$\begin{aligned}
&p(b|a, \underline{X}_a, \sigma_y, \underline{Y}) \\
&\propto p(\underline{Y}|\underline{X}_a, \underline{\theta}, \sigma_y)\underbrace{p(b|a, \underline{X}_a, \sigma_y)}_{\mathcal{N}(\mu_b, \sigma_b^2)} && \text{(B.17)}\\
&\propto \exp\left\{-\frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(Y_i - aX_{ai} - b)^2\right\}\cdot\exp\left\{-\frac{1}{2\sigma_b^2}(b-\mu_b)^2\right\} && \text{(B.18)}\\
&\propto \exp\left\{-\frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(a^2X_{ai}^2 + 2abX_{ai} - 2aY_iX_{ai} + Y_i^2 + b^2 - 2bY_i) - \frac{1}{2\sigma_b^2}(b^2 - 2b\mu_b - \mu_b^2)\right\} && \text{(B.19)}\\
&\propto \exp\left\{-\frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(2b(aX_{ai} - Y_i) + b^2) - \frac{1}{2\sigma_b^2}(b^2 - 2b\mu_b)\right\} && \text{(B.20)}\\
&\propto \exp\left\{-2b\underbrace{\left[\sum_{i=1}^{N}\frac{aX_{ai} - Y_i}{2\sigma_y^2} - \frac{\mu_b}{2\sigma_b^2}\right]}_{B} + b^2\underbrace{\left[-\frac{N}{2\sigma_y^2} - \frac{1}{2\sigma_b^2}\right]}_{A}\right\} && \text{(B.21)}\\
&\propto \exp\left\{A(a - \frac{B}{A})^2\right\} && \text{(B.22)}
\end{aligned}$$

B.1.4 Posterior for σ_y

Evaluate equation (3.1.22).

$$\begin{aligned}
&p(\sigma_y|\underline{\theta}, \underline{X}_a, \underline{Y}) \\
&\propto p(\underline{Y}|\underline{X}_a, \underline{\theta}, \sigma_y)\ \underbrace{p(\sigma_y|\underline{\theta}, \underline{X}_a)}_{\text{invgamma}(\alpha,\beta,\gamma)} && \text{(B.23)}\\
&\propto \frac{1}{|\sigma_y|^N}\exp\left\{-\frac{1}{2\sigma_y^2}\underbrace{\sum_{i=1}^{N}(Y_i - aX_{ai} - b)^2}_{2\tilde{A}}\right\}\cdot\frac{\beta^\alpha}{\Gamma(\alpha)}(\sigma_y - \gamma)^{-\alpha-1}\exp\left\{-\frac{\beta}{\sigma_y - \gamma}\right\} && \text{(B.24)}\\
&\propto \exp\left\{-N\ln(|\sigma_y|) - \tilde{A}\frac{1}{\sigma_y^2} - (\alpha+1)\ln(\sigma_y - \gamma) - \frac{\beta}{\sigma_y - \gamma}\right\} && \text{(B.25)}
\end{aligned}$$

B.2 Posteriors for Discrete Grid

B.2.1 Marginalization over $\underline{X}_a$

$$p(\sigma_y, \underline{\theta} | \underline{Y}, \underline{X}_o)$$

$$= \int_{\mathbb{R}^N} p(\sigma_y, \underline{\theta}, \underline{X}_a | \underline{Y}, \underline{X}_o) d\underline{X}_a \tag{B.26}$$

$$\propto \int_{\mathbb{R}^N} p(\underline{Y}, \underline{X}_o | \sigma_y, \underline{\theta}, \underline{X}_a) \cdot p(\sigma_y, \underline{\theta}, \underline{X}_a) d\underline{X}_a \tag{B.27}$$

$$= \int_{\mathbb{R}^N} p(\underline{Y} | \underline{X}_o, \sigma_y, \underline{\theta}, \underline{X}_a) \cdot \underbrace{p(\underline{X}_o | \sigma_y, \underline{\theta}, \underline{X}_a) \cdot p(\underline{X}_a | \sigma_y, \underline{\theta})} \cdot p(\sigma_y, \underline{\theta}) d\underline{X}_a \tag{B.28}$$

$$= \int_{\mathbb{R}^N} p(\underline{Y} | \underline{X}_o, \sigma_y, \underline{\theta}, \underline{X}_a) \cdot p(\underline{X}_o, \underline{X}_a | \sigma_y, \underline{\theta}) \cdot p(\sigma_y, \underline{\theta}) d\underline{X}_a \tag{B.29}$$

$$= \int_{\mathbb{R}^N} p(\underline{Y} | \underline{X}_o, \sigma_y, \underline{\theta}, \underline{X}_a) \cdot p(\underline{X}_a | \sigma_y, \underline{\theta}, \underline{X}_o) \cdot p(\underline{X}_o | \sigma_y, \underline{\theta}) \cdot p(\sigma_y, \underline{\theta}) d\underline{X}_a \tag{B.30}$$

$$= p(\underline{X}_o | \sigma_y, \underline{\theta}) \cdot p(\sigma_y, \underline{\theta}) \cdot \int_{\mathbb{R}^N} p(\underline{Y} | \underline{X}_o, \sigma_y, \underline{\theta}, \underline{X}_a) \cdot p(\underline{X}_a | \sigma_y, \underline{\theta}, \underline{X}_o) d\underline{X}_a \tag{B.31}$$

$$= p(\underline{X}_o) \cdot p(\sigma_y, \underline{\theta}) \cdot \int_{\mathbb{R}^N} p(\underline{Y} | \sigma_y, \underline{\theta}, \underline{X}_a) \cdot p(\underline{X}_a | \underline{X}_o) d\underline{X}_a \tag{B.32}$$

$$\propto p(\sigma_y, \underline{\theta}) \cdot \int_{\mathbb{R}^N} p(\underline{Y} | \sigma_y, \underline{\theta}, \underline{X}_a) \cdot p(\underline{X}_a | \underline{X}_o) d\underline{X}_a \tag{B.33}$$

$$\begin{aligned} &\propto p(\sigma_y, \underline{\theta}) \cdot \int_{\mathbb{R}^N} \frac{1}{|\sigma_y|^N} \cdot \exp\left\{ -\frac{1}{2\sigma_y^2} \sum_{i=1}^{N} (a^2 X_{ai}^2 - 2a(Y_i - b) X_{ai} + (Y_i - b)^2) \right\} \\ &\quad \cdot \frac{1}{\sqrt{|\boldsymbol{U_x}|}} \cdot \exp\left\{ -\frac{1}{2} (\underline{X}_a - \underline{X}_o)^T \boldsymbol{U_x}^{-1} (\underline{X}_a - \underline{X}_o) \right\} d\underline{X}_a \end{aligned} \tag{B.34}$$

$$\begin{aligned} &= \frac{p(\sigma_y, \underline{\theta})}{|\sigma_y|^N \sqrt{|\boldsymbol{U_x}|}} \cdot \int_{\mathbb{R}^N} \exp\left\{ -\frac{1}{2} (\underline{X}_a{}^T \boldsymbol{F_1} \underline{X}_a - \underline{X}_a{}^T \underline{F}_2 - \underline{F}_2{}^T \underline{X}_a + F_3) \right\} \\ &\quad \cdot \exp\left\{ -\frac{1}{2} (\underline{X}_a{}^T \boldsymbol{U_x}^{-1} \underline{X}_a - \underline{X}_a{}^T \boldsymbol{U_x}^{-1} \underline{X}_o - \underline{X}_o{}^T \boldsymbol{U_x}^{-1} \underline{X}_a + \underline{X}_o{}^T \boldsymbol{U_x}^{-1} \underline{X}_o) \right\} d\underline{X}_a \end{aligned} \tag{B.35}$$

$$= \frac{p(\sigma_y, \underline{\theta})}{|\sigma_y|^N \sqrt{|\boldsymbol{U_x}|}} \cdot \int_{\mathbb{R}^N} \exp\left\{ -\frac{1}{2} (\underline{X}_a{}^T \underbrace{(\boldsymbol{F_1} + \boldsymbol{U_x}^{-1})}_{\boldsymbol{V}^{-1}} \underline{X}_a - \underline{X}_a{}^T \boldsymbol{V}^{-1} \underbrace{\boldsymbol{V}(\boldsymbol{U_x}^{-1} \underline{X}_o + \underline{F}_2)}_{\underline{M}} - \underline{M}^T \boldsymbol{V}^{-1} \underline{X}_a + \underline{X}_o{}^T \boldsymbol{U_x}^{-1} \underline{X}_o + F_3) \right\} d\underline{X}_a \tag{B.36}$$

$$= \frac{p(\sigma_y, \underline{\theta})}{|\sigma_y|^N \sqrt{|\boldsymbol{U_x}|}} \cdot \int_{\mathbb{R}^N} \exp\left\{ -\frac{1}{2} \Big((\underline{X}_a - \underline{M})^T \boldsymbol{V}^{-1} (\boldsymbol{X_a} - \underline{M}) + \underline{X}_o{}^T \boldsymbol{U_x}^{-1} \underline{X}_o - \underline{M}^T \boldsymbol{V}^{-1} \underline{M} + F_3 \Big) \right\} d\underline{X}_a \tag{B.37}$$

$$= \frac{p(\sigma_y, \underline{\theta})}{|\sigma_y|^N \sqrt{|\boldsymbol{U_x}|}} \cdot \exp\left\{ -\frac{1}{2} (\underline{X}_o{}^T \boldsymbol{U_x}^{-1} \underline{X}_o - \underline{M}^T \boldsymbol{V}^{-1} \underline{M} + F_3) \right\} \cdot \int_{\mathbb{R}^N} \exp\left\{ -\frac{1}{2} (\underline{X}_a - \underline{M})^T \boldsymbol{V}^{-1} (\boldsymbol{X_a} - \underline{M})) \right\} d\underline{X}_a \tag{B.38}$$

$$\propto p(\sigma_y, \underline{\theta}) \cdot \exp\left\{ -\frac{1}{2} (\underline{X}_o{}^T \boldsymbol{U_x}^{-1} \underline{X}_o - \underline{M}^T \boldsymbol{V}^{-1} \underline{M} + F_3) \right\} \cdot \frac{\sqrt{|\boldsymbol{V}|}}{|\sigma_y|^N} \tag{B.39}$$

with (similar to appendix B.1.1)

$$\boldsymbol{F_1} = \boldsymbol{G_1}^T \boldsymbol{G_1} \tag{B.40}$$

$$\underline{F}_2 = \boldsymbol{G_1} \underline{G}_2 \tag{B.41}$$

$$F_3 = \underline{G}_2^T \underline{G}_2 \tag{B.42}$$

$$\boldsymbol{G_1} = \frac{a}{\sigma_y} \cdot \boldsymbol{I_N} \tag{B.43}$$

$$\underline{G}_2 = \frac{1}{\sigma_y} \begin{bmatrix} Y_1 - b & \dots & Y_N - b \end{bmatrix}^T \tag{B.44}$$

C SPARQL Queries

C.1 Queries to Evaluate Ontology Metric

```
SELECT (SAMPLE(?s) AS ?subject) (COUNT(?o) as ?n_props)
WHERE {
       ?s a/rdfs:subClassOf* ?o .
       FILTER(STRSTARTS(STR(?s), @@onto_ns@@))
}
GROUP BY ?s
```

Listing C.1: SPARQL template querying elements of ontology `@@onto_ns@@` with subclass relations.

```
SELECT (SAMPLE(?s) AS ?subject) (COUNT(?o) as ?n_props)
WHERE {
       ?s ?p ?o .
       FILTER(STRSTARTS(STR(?s), @@onto_ns@@))
}
GROUP BY ?s
```

Listing C.2: SPARQL template querying elements of ontology `@@onto_ns@@` with any kind of relations.

```
SELECT DISTINCT ?s
WHERE {
       ?s a owl:Class .
       FILTER(STRSTARTS(STR(?s), @@onto_ns@@))
}
```

Listing C.3: SPARQL template querying unique classes of ontology `@@onto_ns@@`.

```
SELECT (SAMPLE(?s) AS ?subject) (COUNT(?att) as ?n_att)
WHERE {
       {
              ?s a owl:Class ;
              rdfs:comment ?att
       }
       UNION
       {
              ?s a owl:Class ;
              rdfs:label ?att
       } .
       FILTER(STRSTARTS(STR(?s), @@onto_ns@@))
}
GROUP BY ?s
```

Listing C.4: SPARQL template querying unique annotated classes of ontology `@@onto_ns@@`.

```
SELECT (SAMPLE(?s1) AS ?subject) (COUNT(?s2) as ?n_subclasses)
WHERE {
       ?s1 a owl:Class .
       ?s2 rdfs:subClassOf+ ?s1 .
       FILTER(STRSTARTS(STR(?s2), @@onto_ns@@))
}
GROUP BY ?s1
```

Listing C.5: SPARQL template querying number of subclasses of each class in ontology `@@onto_ns@@`.

D Evaluation Results per Scenario

In tables tables D.1 to D.42 values coming from a method that did not run or did not run successfully are denoted by `/`, while values that are not applicable are denoted by `-`.

D.1 Scenario `01a_static_input`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
`gibbs_base`	/	/	/	/	/	/	/	/
`gibbs_minimal`	2.00e+00	1.00e+00	0.00e+00	8.36e-01	9.99e-01	1.08e-01	8.15e-02	1.03e-03
`gibbs_no_EIV`	2.00e+00	1.00e+00	0.00e+00	1.24e+00	1.00e+00	1.35e-01	2.21e-01	1.30e-03
`gibbs_known_sigma_y`	2.00e+00	1.00e+00	0.00e+00	9.40e-02	9.97e-01	-	1.55e-01	3.76e-04
`joint_posterior`	2.00e+00	1.00e+00	0.00e+00	2.32e+00	8.57e-01	1.32e-01	1.37e+00	2.86e-01
`joint_posterior_agrid`	2.00e+00	1.00e+00	0.00e+00	9.42e-01	1.00e+00	3.16e-02	3.29e-02	5.66e-04
`stankovic_base`	2.00e+00	1.00e+00	0.00e+00	2.16e+00	1.00e+00	-	3.29e+00	6.83e-03
`stankovic_enhanced_unc`	2.00e+00	1.00e+00	0.00e+00	1.82e+00	1.00e+00	-	2.35e+00	5.06e-03
`stankovic_base_unc`	2.00e+00	1.00e+00	0.00e+00	1.82e+00	1.00e+00	-	2.35e+00	5.06e-03

Table D.1: Main results of scenario `01a_static_input`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
`gibbs_base`	/	/	/	/	/	/	/	/	/	/
`gibbs_minimal`	2.56e+02	-1.16e+00	1.43e+01	-6.68e-04	6.46e-01	1.08e-01	3.25e-01	7.99e-04	6.38e-07	3.80e-05
`gibbs_no_EIV`	2.33e+02	-7.63e-01	3.45e+00	7.20e-04	5.55e-01	1.35e-01	4.04e-01	-5.82e-04	3.39e-07	2.85e-05
`gibbs_known_sigma_y`	7.66e+00	-1.91e+00	1.23e+01	-3.39e-03	9.00e+00	-	-	3.60e-02	1.30e-03	3.65e-01
`joint_posterior`	6.04e+01	3.21e-01	2.34e-01	-1.43e-01	5.00e-01	1.32e-01	1.75e+00	6.15e-02	3.79e-03	1.92e-01
`joint_posterior_agrid`	1.39e+01	-1.06e+00	3.22e+01	1.97e-03	3.47e+00	3.16e-02	5.99e+01	-2.09e-03	4.37e-06	3.87e-03
`stankovic_base`	4.19e-01	1.57e-01	4.76e-02	-9.99e-16	1.46e-13	-	-	5.00e-16	2.50e-31	2.49e-26
`stankovic_enhanced_unc`	4.43e-01	-1.78e-01	7.60e-02	-7.77e-16	1.54e-13	-	-	4.44e-16	1.97e-31	2.56e-26
`stankovic_base_unc`	3.98e-01	-1.78e-01	7.60e-02	-7.77e-16	1.54e-13	-	-	4.44e-16	1.97e-31	2.56e-26

Table D.2: Runtime and consistency metrics of scenario `01a_static_input`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
`gibbs_base`	/	/	/	/	/	/	/	/	/
`gibbs_minimal`	yes	yes	yes	5.99e+00	1.40e+01	1.99e+00	1.99e+00	never	never
`gibbs_no_EIV`	yes	yes	yes	never	never	1.99e+00	1.99e+00	3.99e+00	never
`gibbs_known_sigma_y`	yes	yes	-	never	never	1.99e+00	1.99e+00	-	-
`joint_posterior`	yes	yes	yes	never	never	1.99e+00	never	1.99e+00	1.99e+00
`joint_posterior_agrid`	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00
`stankovic_base`	yes	yes	-	never	never	5.99e+00	5.99e+00	-	-
`stankovic_enhanced_unc`	yes	yes	-	never	never	9.99e+00	9.99e+00	-	-
`stankovic_base_unc`	yes	yes	-	never	never	9.99e+00	9.99e+00	-	-

Table D.3: Convergence metrics of scenario `01a_static_input`.

D.2 Scenario `01b_sinusoidal_input`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	/	/	/	/	/	/	/	/
gibbs_minimal	2.00e+00	1.00e+00	0.00e+00	2.00e+00	1.00e+00	2.09e-01	8.37e-04	3.81e-03
gibbs_no_EIV	-	-	-	-	-	-	-	-
gibbs_known_sigma_y	2.00e+00	1.00e+00	0.00e+00	2.00e+00	9.99e-01	-	4.70e-04	9.24e-04
joint_posterior	/	/	/	/	/	/	/	/
joint_posterior_agrid	/	/	/	/	/	/	/	/
stankovic_base	2.00e+00	1.00e+00	0.00e+00	1.11e+00	8.13e-01	-	9.95e-03	2.61e-02
stankovic_enhanced_unc	2.00e+00	1.00e+00	0.00e+00	1.04e+00	9.29e-01	-	3.79e-02	1.36e-01
stankovic_base_unc	2.00e+00	1.00e+00	0.00e+00	1.04e+00	9.29e-01	-	3.79e-02	1.36e-01

Table D.4: Main results of scenario `01b_sinusoidal_input`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	/	/	/	/	/	/	/	/	/	/
gibbs_minimal	2.32e+02	-1.37e-03	1.64e+00	1.60e-03	4.19e-01	2.09e-01	3.06e+00	-1.14e-04	9.53e-07	8.70e-05
gibbs_no_EIV	0.00e+00	-	-	-	-	-	-	-	-	-
gibbs_known_sigma_y	7.64e+00	-4.59e-04	9.76e-01	-1.26e-03	1.36e+00	-	-	8.58e-04	8.41e-07	2.03e+00
joint_posterior	/	/	/	/	/	/	/	/	/	/
joint_posterior_agrid	/	/	/	/	/	/	/	/	/	/
stankovic_base	4.21e-01	-8.89e-01	8.94e+01	-1.87e-01	7.15e+00	-	-	9.69e-01	2.22e+00	1.12e+03
stankovic_enhanced_unc	4.39e-01	-9.55e-01	2.52e+01	-7.09e-02	5.20e-01	-	-	9.82e-01	2.64e+00	6.05e+01
stankovic_base_unc	3.99e-01	-9.55e-01	2.52e+01	-7.09e-02	5.20e-01	-	-	9.82e-01	2.64e+00	6.05e+01

Table D.5: Runtime and consistency metrics of scenario `01b_sinusoidal_input`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t^*_a(0.1)$	$t_b(0.1)$	$t^*_b(0.1)$	$t_\sigma(0.1)$	$t^*_\sigma(0.1)$
gibbs_base	/	/	/	/	/	/	/	/	/
gibbs_minimal	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	3.99e+00	2.00e+01
gibbs_no_EIV	-	-	-	-	-	-	-	-	-
gibbs_known_sigma_y	yes	yes	-	1.99e+00	1.99e+00	1.99e+00	1.99e+00	-	-
joint_posterior	/	/	/	/	/	/	/	/	/
joint_posterior_agrid	/	/	/	/	/	/	/	/	/
stankovic_base	yes	yes	-	7.99e+00	7.99e+00	1.40e+01	1.40e+01	-	-
stankovic_enhanced_unc	yes	yes	-	7.99e+00	7.99e+00	never	never	-	-
stankovic_base_unc	yes	yes	-	7.99e+00	7.99e+00	never	never	-	-

Table D.6: Convergence metrics of scenario `01b_sinusoidal_input`.

D.3 Scenario `01c_jumping_input`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
`gibbs_base`	/	/	/	/	/	/	/	/
`gibbs_minimal`	-	-	-	-	-	-	-	-
`gibbs_no_EIV`	/	/	/	/	/	/	/	/
`gibbs_known_sigma_y`	/	/	/	/	/	/	/	/
`joint_posterior`	/	/	/	/	/	/	/	/
`joint_posterior_agrid`	/	/	/	/	/	/	/	/
`stankovic_base`	2.00e+00	1.00e+00	0.00e+00	8.58e+00	-5.59e+00	-	6.30e-01	6.32e-01
`stankovic_enhanced_unc`	2.00e+00	1.00e+00	0.00e+00	-7.31e-04	2.70e+00	-	2.88e-08	2.22e-05
`stankovic_base_unc`	2.00e+00	1.00e+00	0.00e+00	-7.31e-04	2.70e+00	-	2.88e-08	2.22e-05

Table D.7: Main results of scenario `01c_jumping_input`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
`gibbs_base`	/	/	/	/	/	/	/	/	/	/
`gibbs_minimal`	0.00e+00	-	-	-	-	-	-	-	-	-
`gibbs_no_EIV`	/	/	/	/	/	/	/	/	/	/
`gibbs_known_sigma_y`	/	/	/	/	/	/	/	/	/	/
`joint_posterior`	/	/	/	/	/	/	/	/	/	/
`joint_posterior_agrid`	/	/	/	/	/	/	/	/	/	/
`stankovic_base`	4.09e-01	6.58e+00	1.04e+01	-6.59e+00	1.04e+01	-	-	1.12e-01	8.41e-01	9.33e+01
`stankovic_enhanced_unc`	4.32e-01	-2.00e+00	6.94e+07	1.70e+00	7.64e+04	-	-	-1.40e+01	1.06e+07	4.93e+08
`stankovic_base_unc`	3.89e-01	-2.00e+00	6.94e+07	1.70e+00	7.64e+04	-	-	-1.40e+01	1.06e+07	4.93e+08

Table D.8: Runtime and consistency metrics of scenario `01c_jumping_input`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
`gibbs_base`	/	/	/	/	/	/	/	/	/
`gibbs_minimal`	-	-	-	-	-	-	-	-	-
`gibbs_no_EIV`	/	/	/	/	/	/	/	/	/
`gibbs_known_sigma_y`	/	/	/	/	/	/	/	/	/
`joint_posterior`	/	/	/	/	/	/	/	/	/
`joint_posterior_agrid`	/	/	/	/	/	/	/	/	/
`stankovic_base`	No	yes	-	1.60e+01	never	1.60e+01	never	-	-
`stankovic_enhanced_unc`	yes	No	-	1.80e+01	1.80e+01	1.80e+01	1.80e+01	-	-
`stankovic_base_unc`	yes	No	-	1.80e+01	1.80e+01	1.80e+01	1.80e+01	-	-

Table D.9: Convergence metrics of scenario `01c_jumping_input`.

D.4 Scenario `02a_static_input_noisy`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	2.00e+00	1.00e+00	1.00e-01	3.06e-01	1.00e+00	7.45e-01	3.21e-01	6.96e-03
gibbs_minimal	2.00e+00	1.00e+00	1.00e-01	-1.27e-01	9.92e-01	2.97e-01	3.99e-01	1.44e-03
gibbs_no_EIV	2.00e+00	1.00e+00	1.00e-01	-2.76e+00	1.01e+00	-2.78e-17	4.24e-08	1.27e-10
gibbs_known_sigma_y	2.00e+00	1.00e+00	1.00e-01	2.31e-01	9.98e-01	-	1.06e-01	6.53e-04
joint_posterior	2.00e+00	1.00e+00	1.00e-01	3.00e+00	8.57e-01	1.82e-01	5.50e-01	2.86e-01
joint_posterior_agrid	2.00e+00	1.00e+00	1.00e-01	1.23e+00	9.80e-01	2.11e-01	9.82e-02	3.07e-03
stankovic_base	2.00e+00	1.00e+00	1.00e-01	2.19e+00	1.01e+00	-	3.26e+00	1.92e-02
stankovic_enhanced_unc	2.00e+00	1.00e+00	1.00e-01	1.85e+00	1.02e+00	-	2.42e+00	2.52e-02
stankovic_base_unc	2.00e+00	1.00e+00	1.00e-01	1.86e+00	1.02e+00	-	2.40e+00	2.02e-02

Table D.10: Main results of scenario `02a_static_input_noisy`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	2.09e+03	-1.69e+00	5.27e+00	-2.73e-04	3.92e-02	6.45e-01	7.32e+03	3.63e-03	1.10e-01	1.75e-02
gibbs_minimal	2.76e+02	-2.13e+00	5.33e+00	-7.74e-03	5.38e+00	1.97e-01	5.90e-01	-7.79e-02	6.46e-01	3.76e-02
gibbs_no_EIV	5.29e+01	-4.76e+00	1.12e+08	1.31e-02	1.04e+08	-1.00e-01	3.00e-01	1.10e-03	1.35e-03	4.18e+15
gibbs_known_sigma_y	7.70e+00	-1.77e+00	1.67e+01	-1.62e-03	2.49e+00	-	-	1.17e-02	1.94e-01	4.67e+00
joint_posterior	6.12e+01	1.00e+00	1.82e+00	-1.43e-01	5.00e-01	8.18e-02	1.50e+01	4.52e-02	3.19e-03	2.45e-01
joint_posterior_agrid	1.38e+01	-7.72e-01	7.86e+00	-1.98e-02	6.43e+00	1.11e-01	4.31e+01	1.45e-02	7.04e-03	2.38e-01
stankovic_base	4.11e-01	1.85e-01	5.69e-02	1.15e-02	6.02e-01	-	-	-7.49e-03	2.21e-03	4.07e-01
stankovic_enhanced_unc	4.30e-01	-1.48e-01	6.13e-02	2.09e-02	8.32e-01	-	-	-1.34e-02	3.18e-03	4.89e-01
stankovic_base_unc	3.89e-01	-1.38e-01	5.76e-02	2.11e-02	1.04e+00	-	-	-1.34e-02	3.15e-03	5.26e-01

Table D.11: Runtime and consistency metrics of scenario `02a_static_input_noisy`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
gibbs_base	yes	yes	yes	never	never	1.99e+00	1.99e+00	9.99e+00	1.80e+01
gibbs_minimal	yes	yes	yes	never	never	1.99e+00	5.99e+00	1.99e+00	never
gibbs_no_EIV	No	yes	yes	7.99e+00	7.99e+00	1.99e+00	1.99e+00	1.99e+00	never
gibbs_known_sigma_y	yes	yes	-	1.80e+01	never	1.99e+00	1.99e+00	-	-
joint_posterior	No	yes	yes	never	never	1.99e+00	never	1.99e+00	1.99e+00
joint_posterior_agrid	yes	yes	yes	3.99e+00	3.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00
stankovic_base	yes	yes	-	never	never	5.99e+00	5.99e+00	-	-
stankovic_enhanced_unc	yes	yes	-	never	never	9.99e+00	9.99e+00	-	-
stankovic_base_unc	yes	yes	-	never	never	9.99e+00	9.99e+00	-	-

Table D.12: Convergence metrics of scenario `02a_static_input_noisy`.

D.5 Scenario `02b_sinusoidal_input_noisy`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	2.00e+00	1.00e+00	1.00e-01	1.99e+00	1.00e+00	3.14e-01	1.05e-02	1.10e-02
gibbs_minimal	2.00e+00	1.00e+00	1.00e-01	2.00e+00	9.96e-01	3.33e-01	1.72e-03	1.52e-03
gibbs_no_EIV	2.00e+00	1.00e+00	1.00e-01	1.99e+00	9.97e-01	3.33e-01	1.30e-03	2.29e-03
gibbs_known_sigma_y	2.00e+00	1.00e+00	1.00e-01	2.00e+00	9.98e-01	-	2.04e-04	7.73e-04
joint_posterior	2.00e+00	1.00e+00	1.00e-01	1.86e+00	1.14e+00	2.28e-01	2.86e-01	2.86e-01
joint_posterior_agrid	2.00e+00	1.00e+00	1.00e-01	1.95e+00	1.05e+00	4.98e-01	5.57e-03	9.60e-03
stankovic_base	2.00e+00	1.00e+00	1.00e-01	1.09e+00	8.01e-01	-	2.34e-02	3.47e-02
stankovic_enhanced_unc	2.00e+00	1.00e+00	1.00e-01	1.02e+00	9.12e-01	-	4.82e-02	1.40e-01
stankovic_base_unc	2.00e+00	1.00e+00	1.00e-01	1.02e+00	9.13e-01	-	4.85e-02	1.41e-01

Table D.13: Main results of scenario `02b_sinusoidal_input_noisy`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	2.05e+03	-9.19e-03	8.75e-01	6.45e-04	5.89e-02	2.14e-01	1.13e+05	2.93e-03	2.69e-03	1.08e-01
gibbs_minimal	3.74e+02	2.16e-03	1.26e+00	-4.44e-03	2.92e+00	2.33e-01	7.00e-01	-2.34e-04	2.63e-03	9.49e-02
gibbs_no_EIV	3.30e+02	-5.04e-03	3.88e+00	-3.29e-03	1.44e+00	2.33e-01	7.00e-01	2.80e-03	2.66e-03	9.52e-02
gibbs_known_sigma_y	7.61e+00	-2.32e-03	1.13e+01	-2.49e-03	3.22e+00	-	-	1.04e-03	2.64e-03	1.51e+04
joint_posterior	5.97e+01	-1.43e-01	5.00e-01	1.43e-01	5.00e-01	1.28e-01	9.85e-01	-1.28e-03	1.46e-02	1.68e-01
joint_posterior_agrid	1.38e+01	-5.27e-02	9.46e+00	5.21e-02	5.43e+00	3.98e-01	5.21e+01	-1.02e-03	4.15e-03	6.33e-02
stankovic_base	4.10e-01	-9.07e-01	3.88e+01	-1.99e-01	5.74e+00	-	-	1.01e+00	2.40e+00	2.87e+02
stankovic_enhanced_unc	4.32e-01	-9.82e-01	2.04e+01	-8.81e-02	6.27e-01	-	-	1.05e+00	2.97e+00	4.59e+01
stankovic_base_unc	3.89e-01	-9.75e-01	2.01e+01	-8.74e-02	6.19e-01	-	-	1.04e+00	2.89e+00	4.51e+01

Table D.14: Runtime and consistency metrics of scenario `02b_sinusoidal_input_noisy`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t^*_a(0.1)$	$t_b(0.1)$	$t^*_b(0.1)$	$t_\sigma(0.1)$	$t^*_\sigma(0.1)$
gibbs_base	yes	yes	No	1.99e+00	1.99e+00	1.99e+00	1.99e+00	2.00e+01	2.00e+01
gibbs_minimal	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	never	never
gibbs_no_EIV	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	never	never
gibbs_known_sigma_y	yes	yes	-	1.99e+00	1.99e+00	1.99e+00	1.99e+00	-	-
joint_posterior	yes	yes	yes	1.99e+00	never	1.99e+00	never	1.99e+00	never
joint_posterior_agrid	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00
stankovic_base	yes	yes	-	7.99e+00	7.99e+00	1.40e+01	1.40e+01	-	-
stankovic_enhanced_unc	yes	yes	-	5.99e+00	5.99e+00	never	never	-	-
stankovic_base_unc	yes	yes	-	5.99e+00	5.99e+00	never	never	-	-

Table D.15: Convergence metrics of scenario `02b_sinusoidal_input_noisy`.

D.6 Scenario `02c_jumping_input_noisy`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	2.00e+00	1.00e+00	1.00e-01	2.01e+00	9.92e-01	9.31e-01	5.25e-03	2.81e-02
gibbs_minimal	2.00e+00	1.00e+00	1.00e-01	2.00e+00	9.94e-01	3.14e-01	2.84e-03	2.22e-03
gibbs_no_EIV	2.00e+00	1.00e+00	1.00e-01	2.01e+00	9.88e-01	3.38e-01	1.57e-03	5.59e-03
gibbs_known_sigma_y	2.00e+00	1.00e+00	1.00e-01	2.00e+00	1.00e+00	-	3.38e-04	1.20e-03
joint_posterior	2.00e+00	1.00e+00	1.00e-01	2.14e+00	8.57e-01	2.32e-01	2.86e-01	2.86e-01
joint_posterior_agrid	2.00e+00	1.00e+00	1.00e-01	2.04e+00	9.61e-01	3.48e-01	5.80e-03	1.06e-02
stankovic_base	2.00e+00	1.00e+00	1.00e-01	2.83e-04	-5.02e-02	-	3.13e-06	3.83e-03
stankovic_enhanced_unc	2.00e+00	1.00e+00	1.00e-01	4.02e-08	-5.02e-02	-	1.31e-10	1.16e-03
stankovic_base_unc	2.00e+00	1.00e+00	1.00e-01	4.09e-08	-5.02e-02	-	1.33e-10	1.16e-03

Table D.16: Main results of scenario `02c_jumping_input_noisy`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	1.94e+03	8.79e-03	1.67e+00	-7.65e-03	2.72e-01	8.31e-01	2.81e+00	-1.87e-04	2.50e-03	1.16e-02
gibbs_minimal	3.28e+02	2.34e-03	8.24e-01	-6.41e-03	2.89e+00	2.14e-01	6.42e-01	2.03e-03	2.52e-03	1.02e-01
gibbs_no_EIV	4.40e+02	7.61e-03	4.86e+00	-1.24e-02	2.22e+00	2.38e-01	7.15e-01	2.69e-03	2.51e-03	8.82e-02
gibbs_known_sigma_y	7.71e+00	1.58e-03	4.68e+00	3.77e-04	3.14e-01	-	-	-1.03e-03	2.52e-03	6.15e+03
joint_posterior	5.99e+01	1.43e-01	5.00e-01	-1.43e-01	5.00e-01	1.32e-01	2.93e+01	7.81e-03	7.74e-03	1.49e-01
joint_posterior_agrid	1.39e+01	4.16e-02	7.18e+00	-3.92e-02	3.70e+00	2.48e-01	5.76e+01	1.10e-03	2.85e-03	9.78e-02
stankovic_base	4.12e-01	-2.00e+00	6.39e+05	-1.05e+00	2.74e+02	-	-	9.94e+03	1.65e+08	7.35e+03
stankovic_enhanced_unc	4.37e-01	-2.00e+00	1.53e+10	-1.05e+00	9.05e+02	-	-	7.00e+07	8.17e+15	8.50e+04
stankovic_base_unc	3.89e-01	-2.00e+00	1.51e+10	-1.05e+00	9.05e+02	-	-	6.88e+07	7.91e+15	8.50e+04

Table D.17: Runtime and consistency metrics of scenario `02c_jumping_input_noisy`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
gibbs_base	yes	yes	No	1.99e+00	1.99e+00	1.99e+00	1.99e+00	never	never
gibbs_minimal	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	3.99e+00	never
gibbs_no_EIV	yes	No	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	3.99e+00	never
gibbs_known_sigma_y	yes	yes	-	1.99e+00	1.99e+00	1.99e+00	1.99e+00	-	-
joint_posterior	yes	yes	yes	1.99e+00	never	1.99e+00	never	1.99e+00	1.99e+00
joint_posterior_agrid	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00
stankovic_base	No	yes	-	1.60e+01	1.60e+01	1.80e+01	1.80e+01	-	-
stankovic_enhanced_unc	No	No	-	2.00e+01	2.00e+01	2.00e+01	2.00e+01	-	-
stankovic_base_unc	No	yes	-	2.00e+01	2.00e+01	2.00e+01	2.00e+01	-	-

Table D.18: Convergence metrics of scenario `02c_jumping_input_noisy`.

D.7 Scenario `03a_variable_blockwise`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
`gibbs_base`	/	/	/	/	/	/	/	/
`gibbs_minimal`	2.00e+00	1.00e+00	1.00e-01	1.99e+00	1.01e+00	1.04e-01	5.71e-04	3.51e-04
`gibbs_no_EIV`	2.00e+00	1.00e+00	1.00e-01	1.99e+00	1.01e+00	9.98e-02	4.67e-04	9.21e-04
`gibbs_known_sigma_y`	2.00e+00	1.00e+00	1.00e-01	2.00e+00	1.01e+00	-	1.13e-03	7.09e-04
`joint_posterior`	2.00e+00	1.00e+00	1.00e-01	1.86e+00	1.14e+00	1.32e-01	2.86e-01	2.86e-01
`joint_posterior_agrid`	2.00e+00	1.00e+00	1.00e-01	2.01e+00	1.05e+00	4.45e-01	7.57e-03	9.30e-03
`stankovic_base`	2.00e+00	1.00e+00	1.00e-01	1.85e+00	1.14e+00	-	3.27e-01	4.32e-01
`stankovic_enhanced_unc`	2.00e+00	1.00e+00	1.00e-01	1.68e+00	1.28e+00	-	4.18e-01	5.76e-01
`stankovic_base_unc`	2.00e+00	1.00e+00	1.00e-01	1.68e+00	1.28e+00	-	4.19e-01	5.77e-01

Table D.19: Main results of scenario `03a_variable_blockwise`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
`gibbs_base`	/	/	/	/	/	/	/	/	/	/
`gibbs_minimal`	7.31e+02	-1.06e-02	1.86e+01	1.16e-02	3.29e+01	4.19e-03	1.26e-02	-7.66e-04	2.56e-03	9.33e-01
`gibbs_no_EIV`	5.33e+02	-6.78e-03	1.45e+01	1.33e-02	1.45e+01	-1.53e-04	4.60e-04	-3.56e-03	2.55e-03	1.02e+00
`gibbs_known_sigma_y`	1.10e+01	-4.37e-03	3.88e+00	8.64e-03	1.22e+01	-	-	-2.40e-03	2.54e-03	8.71e+03
`joint_posterior`	8.20e+01	-1.43e-01	5.00e-01	1.43e-01	5.00e-01	3.25e-02	4.30e-01	-8.40e-04	5.84e-03	1.22e-01
`joint_posterior_agrid`	1.91e+01	8.39e-03	1.11e+00	4.58e-02	4.92e+00	3.45e-01	5.76e+01	-2.72e-02	3.25e-03	6.61e-02
`stankovic_base`	4.32e-01	-1.52e-01	4.65e-01	1.43e-01	3.32e-01	-	-	3.80e-03	6.31e-03	7.20e-02
`stankovic_enhanced_unc`	4.45e-01	-3.20e-01	7.66e-01	2.81e-01	4.88e-01	-	-	2.16e-02	2.21e-02	1.07e-01
`stankovic_base_unc`	3.99e-01	-3.16e-01	7.55e-01	2.78e-01	4.81e-01	-	-	2.11e-02	2.16e-02	1.04e-01

Table D.20: Runtime and consistency metrics of scenario `03a_variable_blockwise`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
`gibbs_base`	/	/	/	/	/	/	/	/	/
`gibbs_minimal`	yes	yes	yes	4.00e-02	4.00e-02	4.00e-02	4.00e-02	4.00e-02	never
`gibbs_no_EIV`	yes	yes	yes	4.00e-02	4.00e-02	1.90e-01	1.90e-01	4.00e-02	never
`gibbs_known_sigma_y`	yes	yes	-	4.00e-02	4.00e-02	4.00e-02	4.00e-02	-	-
`joint_posterior`	yes	yes	yes	1.90e-01	never	1.90e-01	never	1.90e-01	1.90e-01
`joint_posterior_agrid`	yes	yes	yes	2.33e+00	2.33e+00	2.33e+00	2.33e+00	4.00e-02	4.00e-02
`stankovic_base`	yes	yes	-	never	never	never	never	-	-
`stankovic_enhanced_unc`	yes	yes	-	never	never	never	never	-	-
`stankovic_base_unc`	yes	yes	-	never	never	never	never	-	-

Table D.21: Convergence metrics of scenario `03a_variable_blockwise`.

D.8 Scenario `03b_smaller_blocks`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	/	/	/	/	/	/	/	/
gibbs_minimal	2.00e+00	1.00e+00	1.00e-01	1.98e+00	1.03e+00	3.33e-01	3.60e-03	3.37e-03
gibbs_no_EIV	2.00e+00	1.00e+00	1.00e-01	1.99e+00	1.01e+00	2.68e-01	1.05e-03	2.11e-03
gibbs_known_sigma_y	2.00e+00	1.00e+00	1.00e-01	2.00e+00	1.01e+00	-	8.60e-04	5.09e-04
joint_posterior	2.00e+00	1.00e+00	1.00e-01	1.86e+00	1.14e+00	1.32e-01	2.86e-01	2.86e-01
joint_posterior_agrid	2.00e+00	1.00e+00	1.00e-01	1.95e+00	9.05e-01	4.87e-01	1.10e-02	1.80e-02
stankovic_base	2.00e+00	1.00e+00	1.00e-01	1.85e+00	1.14e+00	-	3.27e-01	4.32e-01
stankovic_enhanced_unc	2.00e+00	1.00e+00	1.00e-01	1.68e+00	1.28e+00	-	4.18e-01	5.76e-01
stankovic_base_unc	2.00e+00	1.00e+00	1.00e-01	1.68e+00	1.28e+00	-	4.19e-01	5.77e-01

Table D.22: Main results of scenario `03b_smaller_blocks`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	/	/	/	/	/	/	/	/	/	/
gibbs_minimal	7.38e+02	-2.09e-02	5.81e+00	3.29e-02	9.76e+00	2.33e-01	7.00e-01	-6.41e-03	2.66e-03	9.39e-02
gibbs_no_EIV	3.89e+02	-5.04e-03	4.81e+00	1.36e-02	6.43e+00	1.68e-01	5.04e-01	-4.54e-03	2.55e-03	1.42e-01
gibbs_known_sigma_y	3.93e+00	-4.73e-03	5.51e+00	7.34e-03	1.44e+01	-	-	-1.57e-03	2.54e-03	1.62e+04
joint_posterior	4.02e+01	-1.43e-01	5.00e-01	1.43e-01	5.00e-01	3.25e-02	4.30e-01	-8.40e-04	5.84e-03	1.22e-01
joint_posterior_agrid	1.02e+01	-4.95e-02	4.52e+00	-9.48e-02	5.27e+00	3.87e-01	2.90e+01	7.36e-02	8.38e-03	1.34e-01
stankovic_base	4.08e-01	-1.52e-01	4.65e-01	1.43e-01	3.32e-01	-	-	3.80e-03	6.31e-03	7.20e-02
stankovic_enhanced_unc	4.29e-01	-3.20e-01	7.66e-01	2.81e-01	4.88e-01	-	-	2.16e-02	2.21e-02	1.07e-01
stankovic_base_unc	3.85e-01	-3.16e-01	7.55e-01	2.78e-01	4.81e-01	-	-	2.11e-02	2.16e-02	1.04e-01

Table D.23: Runtime and consistency metrics of scenario `03b_smaller_blocks`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
gibbs_base	/	/	/	/	/	/	/	/	/
gibbs_minimal	yes	yes	yes	9.90e-01	9.90e-01	9.90e-01	9.90e-01	never	never
gibbs_no_EIV	yes	yes	yes	9.90e-01	9.90e-01	9.90e-01	9.90e-01	9.90e-01	never
gibbs_known_sigma_y	yes	yes	-	9.90e-01	9.90e-01	9.90e-01	9.90e-01	-	-
joint_posterior	yes	yes	yes	9.90e-01	never	9.90e-01	never	9.90e-01	9.90e-01
joint_posterior_agrid	yes	yes	yes	9.90e-01	9.90e-01	9.90e-01	9.90e-01	9.90e-01	9.90e-01
stankovic_base	yes	yes	-	never	never	never	never	-	-
stankovic_enhanced_unc	yes	yes	-	never	never	never	never	-	-
stankovic_base_unc	yes	yes	-	never	never	never	never	-	-

Table D.24: Convergence metrics of scenario `03b_smaller_blocks`.

D.9 Scenario `03c_larger_blocks`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	/	/	/	/	/	/	/	/
gibbs_minimal	2.00e+00	1.00e+00	1.00e-01	1.99e+00	1.01e+00	3.27e-01	2.84e-03	2.72e-03
gibbs_no_EIV	2.00e+00	1.00e+00	1.00e-01	2.00e+00	1.00e+00	9.59e-02	7.17e-04	9.17e-04
gibbs_known_sigma_y	2.00e+00	1.00e+00	1.00e-01	1.99e+00	1.01e+00	-	1.65e-03	8.70e-04
joint_posterior	2.00e+00	1.00e+00	1.00e-01	1.86e+00	1.14e+00	1.32e-01	2.86e-01	2.86e-01
joint_posterior_agrid	2.00e+00	1.00e+00	1.00e-01	2.01e+00	1.01e+00	3.97e-01	5.76e-03	7.60e-03
stankovic_base	2.00e+00	1.00e+00	1.00e-01	1.85e+00	1.14e+00	-	3.27e-01	4.32e-01
stankovic_enhanced_unc	2.00e+00	1.00e+00	1.00e-01	1.68e+00	1.28e+00	-	4.18e-01	5.76e-01
stankovic_base_unc	2.00e+00	1.00e+00	1.00e-01	1.68e+00	1.28e+00	-	4.19e-01	5.77e-01

Table D.25: Main results of scenario `03c_larger_blocks`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	/	/	/	/	/	/	/	/	/	/
gibbs_minimal	3.48e+02	-1.35e-02	4.77e+00	1.16e-02	4.28e+00	2.27e-01	6.82e-01	6.67e-04	2.57e-03	9.48e-02
gibbs_no_EIV	3.50e+02	3.16e-03	4.40e+00	2.57e-03	2.81e+00	-4.13e-03	1.24e-02	-3.09e-03	2.52e-03	1.10e+00
gibbs_known_sigma_y	7.77e+00	-5.06e-03	3.06e+00	8.75e-03	1.01e+01	-	-	-2.11e-03	2.54e-03	5.19e+03
joint_posterior	6.03e+01	-1.43e-01	5.00e-01	1.43e-01	5.00e-01	3.25e-02	4.30e-01	-8.40e-04	5.84e-03	1.22e-01
joint_posterior_agrid	1.41e+01	1.03e-02	1.79e+00	1.31e-02	1.73e+00	2.97e-01	6.83e+01	-1.19e-02	2.65e-03	6.80e-02
stankovic_base	4.20e-01	-1.52e-01	4.65e-01	1.43e-01	3.32e-01	-	-	3.80e-03	6.31e-03	7.20e-02
stankovic_enhanced_unc	4.35e-01	-3.20e-01	7.66e-01	2.81e-01	4.88e-01	-	-	2.16e-02	2.21e-02	1.07e-01
stankovic_base_unc	3.97e-01	-3.16e-01	7.55e-01	2.78e-01	4.81e-01	-	-	2.11e-02	2.16e-02	1.04e-01

Table D.26: Runtime and consistency metrics of scenario `03c_larger_blocks`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t^*_a(0.1)$	$t_b(0.1)$	$t^*_b(0.1)$	$t_\sigma(0.1)$	$t^*_\sigma(0.1)$
gibbs_base	/	/	/	/	/	/	/	/	/
gibbs_minimal	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	3.99e+00	never
gibbs_no_EIV	yes	yes	yes	1.99e+00	1.40e+01	1.99e+00	1.99e+00	5.99e+00	never
gibbs_known_sigma_y	yes	yes	-	1.99e+00	1.99e+00	1.99e+00	1.99e+00	-	-
joint_posterior	yes	yes	yes	1.99e+00	never	1.99e+00	never	1.99e+00	1.99e+00
joint_posterior_agrid	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00
stankovic_base	yes	yes	-	never	never	never	never	-	-
stankovic_enhanced_unc	yes	yes	-	never	never	never	never	-	-
stankovic_base_unc	yes	yes	-	never	never	never	never	-	-

Table D.27: Convergence metrics of scenario `03c_larger_blocks`.

D.10 Scenario 04a_dropouts

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	/	/	/	/	/	/	/	/
gibbs_minimal	2.00e+00	1.00e+00	1.00e-01	1.99e+00	1.01e+00	3.33e-01	3.36e-03	2.81e-03
gibbs_no_EIV	2.00e+00	1.00e+00	1.00e-01	2.00e+00	9.98e-01	3.33e-01	1.75e-03	1.58e-03
gibbs_known_sigma_y	2.00e+00	1.00e+00	1.00e-01	2.00e+00	9.99e-01	-	4.56e-04	5.00e-04
joint_posterior	2.00e+00	1.00e+00	1.00e-01	1.86e+00	1.14e+00	2.77e-01	2.86e-01	2.86e-01
joint_posterior_agrid	2.00e+00	1.00e+00	1.00e-01	1.99e+00	9.41e-01	4.61e-01	3.41e-03	9.15e-03
stankovic_base	2.00e+00	1.00e+00	1.00e-01	1.81e+00	9.65e-01	-	1.34e-01	2.75e-01
stankovic_enhanced_unc	2.00e+00	1.00e+00	1.00e-01	1.49e+00	8.94e-01	-	1.30e-01	3.57e-01
stankovic_base_unc	2.00e+00	1.00e+00	1.00e-01	1.50e+00	8.96e-01	-	1.31e-01	3.59e-01

Table D.28: Main results of scenario 04a_dropouts.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	/	/	/	/	/	/	/	/	/	/
gibbs_minimal	3.60e+02	-7.16e-03	2.13e+00	5.37e-03	1.91e+00	2.33e-01	7.00e-01	-6.01e-04	2.60e-03	9.29e-02
gibbs_no_EIV	3.22e+02	-1.72e-03	9.84e-01	-2.07e-03	1.32e+00	2.33e-01	7.00e-01	4.06e-04	2.57e-03	9.23e-02
gibbs_known_sigma_y	7.92e+00	-2.11e-03	4.62e+00	-9.07e-04	1.81e+00	-	-	1.65e-05	2.57e-03	2.01e+04
joint_posterior	6.24e+01	-1.43e-01	5.00e-01	1.43e-01	5.00e-01	1.77e-01	4.98e+01	-1.64e-03	1.46e-02	1.52e-01
joint_posterior_agrid	1.43e+01	-6.11e-03	1.79e+00	-5.92e-02	6.47e+00	3.61e-01	4.56e+01	3.13e-02	3.57e-03	6.66e-02
stankovic_base	4.00e-01	-1.85e-01	1.39e+00	-3.48e-02	1.26e-01	-	-	1.19e-01	3.79e-02	6.74e-01
stankovic_enhanced_unc	4.19e-01	-5.14e-01	3.95e+00	-1.06e-01	2.97e-01	-	-	4.15e-01	4.15e-01	3.04e+00
stankovic_base_unc	3.79e-01	-5.04e-01	3.84e+00	-1.04e-01	2.89e-01	-	-	4.04e-01	3.95e-01	2.90e+00

Table D.29: Runtime and consistency metrics of scenario 04a_dropouts.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
gibbs_base	/	/	/	/	/	/	/	/	/
gibbs_minimal	yes	yes	No	1.99e+00	9.99e+00	1.99e+00	1.80e+01	1.99e+00	never
gibbs_no_EIV	yes	yes	yes	1.99e+00	2.00e+01	1.99e+00	1.99e+00	1.99e+00	never
gibbs_known_sigma_y	yes	yes	-	1.99e+00	1.99e+00	1.99e+00	1.99e+00	-	-
joint_posterior	yes	yes	yes	1.99e+00	never	1.99e+00	never	1.99e+00	1.99e+00
joint_posterior_agrid	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00
stankovic_base	yes	yes	-	never	never	never	never	-	-
stankovic_enhanced_unc	yes	yes	-	never	never	never	never	-	-
stankovic_base_unc	yes	yes	-	never	never	never	never	-	-

Table D.30: Convergence metrics of scenario 04a_dropouts.

D.11 Scenario `04b_outliers`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	/	/	/	/	/	/	/	/
gibbs_minimal	2.00e+00	1.00e+00	1.00e-01	2.01e+00	9.99e-01	3.33e-01	2.06e-03	1.99e-03
gibbs_no_EIV	2.00e+00	1.00e+00	1.00e-01	2.00e+00	9.87e-01	3.41e-01	4.06e-03	3.86e-03
gibbs_known_sigma_y	2.00e+00	1.00e+00	1.00e-01	2.00e+00	1.00e+00	-	1.04e-03	1.29e-03
joint_posterior	2.00e+00	1.00e+00	1.00e-01	2.14e+00	8.57e-01	2.28e-01	2.86e-01	2.86e-01
joint_posterior_agrid	2.00e+00	1.00e+00	1.00e-01	1.96e+00	9.46e-01	5.46e-01	9.86e-03	1.23e-02
stankovic_base	2.00e+00	1.00e+00	1.00e-01	2.71e-06	-4.09e-01	-	2.65e-07	1.60e-01
stankovic_enhanced_unc	2.00e+00	1.00e+00	1.00e-01	-4.08e-05	8.37e+00	-	8.12e-05	1.36e+01
stankovic_base_unc	2.00e+00	1.00e+00	1.00e-01	-3.73e-05	7.77e+00	-	6.75e-05	1.13e+01

Table D.31: Main results of scenario `04b_outliers`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	/	/	/	/	/	/	/	/	/	/
gibbs_minimal	3.46e+02	5.98e-03	2.91e+00	-9.12e-04	4.58e-01	2.33e-01	6.99e-01	-3.72e-03	2.53e-03	9.18e-02
gibbs_no_EIV	4.55e+02	2.46e-03	6.05e-01	-1.27e-02	3.28e+00	2.41e-01	7.24e-01	3.88e-03	2.53e-03	8.72e-02
gibbs_known_sigma_y	7.71e+00	1.48e-03	1.42e+00	-2.73e-04	2.11e-01	-	-	-1.82e-03	2.52e-03	3.17e+03
joint_posterior	5.96e+01	1.43e-01	5.00e-01	-1.43e-01	5.00e-01	1.28e-01	9.85e-01	-6.54e-04	1.09e-02	1.95e-01
joint_posterior_agrid	1.37e+01	-4.38e-02	4.44e+00	-5.40e-02	4.38e+00	4.46e-01	4.62e+01	4.85e-02	6.06e-03	7.76e-02
stankovic_base	4.26e-01	-2.00e+00	7.56e+06	-1.41e+00	8.82e+00	-	-	1.25e+06	2.66e+12	6.35e+01
stankovic_enhanced_unc	4.39e-01	-2.00e+00	2.46e+04	7.37e+00	5.42e-01	-	-	1.32e+05	2.22e+10	9.13e-02
stankovic_base_unc	3.94e-01	-2.00e+00	2.96e+04	6.77e+00	5.99e-01	-	-	1.28e+05	2.22e+10	1.07e-01

Table D.32: Runtime and consistency metrics of scenario `04b_outliers`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
gibbs_base	/	/	/	/	/	/	/	/	/
gibbs_minimal	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	never
gibbs_no_EIV	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	3.99e+00	never
gibbs_known_sigma_y	yes	No	-	1.99e+00	1.99e+00	1.99e+00	1.99e+00	-	-
joint_posterior	yes	yes	yes	1.99e+00	never	1.99e+00	never	1.99e+00	never
joint_posterior_agrid	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00
stankovic_base	yes	yes	-	1.99e+00	1.99e+00	3.99e+00	never	-	-
stankovic_enhanced_unc	No	No	-	1.99e+00	1.99e+00	1.99e+00	never	-	-
stankovic_base_unc	No	No	-	1.99e+00	1.99e+00	1.99e+00	never	-	-

Table D.33: Convergence metrics of scenario `04b_outliers`.

D.12 Scenario `05a_better_references`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	/	/	/	/	/	/	/	/
gibbs_minimal	2.00e+00	1.00e+00	1.00e-01	2.00e+00	9.91e-01	3.33e-01	1.71e-03	3.66e-03
gibbs_no_EIV	2.00e+00	1.00e+00	1.00e-01	2.00e+00	1.00e+00	3.33e-01	1.26e-03	3.52e-03
gibbs_known_sigma_y	2.00e+00	1.00e+00	1.00e-01	2.00e+00	1.00e+00	-	4.43e-04	8.59e-04
joint_posterior	2.00e+00	1.00e+00	1.00e-01	1.86e+00	1.14e+00	2.75e-01	2.86e-01	2.86e-01
joint_posterior_agrid	2.00e+00	1.00e+00	1.00e-01	1.96e+00	9.57e-01	5.03e-01	6.46e-03	1.42e-02
stankovic_base	2.00e+00	1.00e+00	1.00e-01	2.32e+00	9.80e-01	-	4.17e-01	7.48e-01
stankovic_enhanced_unc	2.00e+00	1.00e+00	1.00e-01	2.31e+00	9.79e-01	-	4.14e-01	7.45e-01
stankovic_base_unc	2.00e+00	1.00e+00	1.00e-01	2.32e+00	9.80e-01	-	4.17e-01	7.48e-01

Table D.34: Main results of scenario `05a_better_references`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	/	/	/	/	/	/	/	/	/	/
gibbs_minimal	3.70e+02	3.39e-03	1.98e+00	-8.77e-03	2.40e+00	2.33e-01	7.00e-01	2.57e-03	2.63e-03	9.51e-02
gibbs_no_EIV	3.32e+02	-2.91e-04	2.31e-01	1.47e-03	4.17e-01	2.33e-01	7.00e-01	-6.85e-04	2.62e-03	9.43e-02
gibbs_known_sigma_y	7.80e+00	-3.07e-03	6.93e+00	2.48e-03	2.89e+00	-	-	2.08e-04	2.62e-03	9.43e+03
joint_posterior	6.07e+01	-1.43e-01	5.00e-01	1.43e-01	5.00e-01	1.75e-01	4.83e+01	5.44e-04	1.44e-02	1.46e-01
joint_posterior_agrid	1.39e+01	-3.76e-02	5.82e+00	-4.26e-02	2.99e+00	4.03e-01	2.65e+01	4.09e-02	5.01e-03	7.61e-02
stankovic_base	3.95e-01	3.17e-01	7.61e-01	-2.05e-02	2.74e-02	-	-	-1.29e-01	5.69e-02	2.50e-01
stankovic_enhanced_unc	4.14e-01	3.08e-01	7.43e-01	-2.07e-02	2.78e-02	-	-	-1.26e-01	5.39e-02	2.37e-01
stankovic_base_unc	3.72e-01	3.17e-01	7.61e-01	-2.05e-02	2.74e-02	-	-	-1.29e-01	5.69e-02	2.50e-01

Table D.35: Runtime and consistency metrics of scenario `05a_better_references`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
gibbs_base	/	/	/	/	/	/	/	/	/
gibbs_minimal	yes	yes	No	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	never
gibbs_no_EIV	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	never	never
gibbs_known_sigma_y	yes	yes	-	1.99e+00	7.99e+00	1.99e+00	1.99e+00	-	-
joint_posterior	yes	yes	yes	1.99e+00	never	1.99e+00	never	1.99e+00	1.99e+00
joint_posterior_agrid	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00
stankovic_base	yes	yes	-	never	never	never	never	-	-
stankovic_enhanced_unc	yes	yes	-	never	never	never	never	-	-
stankovic_base_unc	yes	yes	-	never	never	never	never	-	-

Table D.36: Convergence metrics of scenario `05a_better_references`.

D.13 Scenario `05b_equal_references`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	/	/	/	/	/	/	/	/
gibbs_minimal	2.00e+00	1.00e+00	1.00e-01	1.98e+00	1.01e+00	1.46e-01	1.09e-03	2.07e-03
gibbs_no_EIV	2.00e+00	1.00e+00	1.00e-01	1.99e+00	9.99e-01	5.38e-01	1.01e-03	1.88e-03
gibbs_known_sigma_y	2.00e+00	1.00e+00	1.00e-01	2.00e+00	9.97e-01	-	1.27e-03	1.73e-03
joint_posterior	2.00e+00	1.00e+00	1.00e-01	1.86e+00	1.05e+00	2.30e-01	2.86e-01	9.81e-03
joint_posterior_agrid	2.00e+00	1.00e+00	1.00e-01	2.02e+00	1.01e+00	5.01e-01	3.24e-03	7.58e-03
stankovic_base	2.00e+00	1.00e+00	1.00e-01	2.48e+00	1.04e+00	-	4.64e-01	7.73e-01
stankovic_enhanced_unc	2.00e+00	1.00e+00	1.00e-01	2.46e+00	1.04e+00	-	4.61e-01	7.69e-01
stankovic_base_unc	2.00e+00	1.00e+00	1.00e-01	2.48e+00	1.04e+00	-	4.64e-01	7.73e-01

Table D.37: Main results of scenario `05b_equal_references`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	/	/	/	/	/	/	/	/	/	/
gibbs_minimal	3.82e+02	-1.63e-02	1.50e+01	9.62e-03	4.66e+00	4.56e-02	2.08e+00	2.74e-03	2.69e-03	4.99e-01
gibbs_no_EIV	1.49e+02	-1.10e-02	1.08e+01	-5.87e-04	3.12e-01	4.38e-01	1.31e+00	5.24e-03	2.64e-03	3.60e-02
gibbs_known_sigma_y	7.67e+00	-4.51e-03	3.55e+00	-3.47e-03	2.01e+00	-	-	3.49e-03	2.57e-03	1.89e+03
joint_posterior	5.99e+01	-1.43e-01	5.00e-01	5.24e-02	5.35e+00	1.30e-01	2.12e+01	4.66e-02	1.66e-02	1.98e-01
joint_posterior_agrid	1.39e+01	2.11e-02	6.52e+00	1.39e-02	1.83e+00	4.01e-01	3.38e+01	-1.75e-02	3.05e-03	4.97e-02
stankovic_base	3.97e-01	4.76e-01	1.03e+00	4.43e-02	5.73e-02	-	-	-2.06e-01	1.19e-01	5.48e-01
stankovic_enhanced_unc	4.27e-01	4.65e-01	1.01e+00	4.37e-02	5.67e-02	-	-	-2.03e-01	1.14e-01	5.26e-01
stankovic_base_unc	3.77e-01	4.76e-01	1.03e+00	4.43e-02	5.73e-02	-	-	-2.06e-01	1.19e-01	5.48e-01

Table D.38: Runtime and consistency metrics of scenario `05b_equal_references`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
gibbs_base	/	/	/	/	/	/	/	/	/
gibbs_minimal	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	5.99e+00	5.99e+00
gibbs_no_EIV	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	never
gibbs_known_sigma_y	yes	yes	-	1.99e+00	1.99e+00	1.99e+00	1.99e+00	-	-
joint_posterior	yes	yes	yes	1.99e+00	never	1.99e+00	1.99e+00	1.99e+00	1.99e+00
joint_posterior_agrid	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00
stankovic_base	yes	yes	-	never	never	never	never	-	-
stankovic_enhanced_unc	yes	yes	-	never	never	never	never	-	-
stankovic_base_unc	yes	yes	-	never	never	never	never	-	-

Table D.39: Convergence metrics of scenario `05b_equal_references`.

D.14 Scenario `05c_worse_references`

	a	b	σ_y	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_y$	$u_{\hat{a}}$	$u_{\hat{b}}$
gibbs_base	/	/	/	/	/	/	/	/
gibbs_minimal	2.00e+00	1.00e+00	1.00e-01	1.39e+00	1.24e+00	1.22e+00	2.06e-03	1.66e-02
gibbs_no_EIV	2.00e+00	1.00e+00	1.00e-01	1.36e+00	1.64e+00	1.04e+00	6.66e-03	3.47e-03
gibbs_known_sigma_y	2.00e+00	1.00e+00	1.00e-01	1.37e+00	9.89e-01	-	1.12e-03	5.90e-03
joint_posterior	2.00e+00	1.00e+00	1.00e-01	1.44e+00	1.17e+00	4.03e-01	1.85e-02	4.58e-02
joint_posterior_agrid	2.00e+00	1.00e+00	1.00e-01	1.65e+00	9.07e-01	1.47e-01	1.30e-02	1.97e-02
stankovic_base	2.00e+00	1.00e+00	1.00e-01	2.57e+00	1.09e+00	-	5.46e-01	8.19e-01
stankovic_enhanced_unc	2.00e+00	1.00e+00	1.00e-01	2.56e+00	1.09e+00	-	5.42e-01	8.16e-01
stankovic_base_unc	2.00e+00	1.00e+00	1.00e-01	2.57e+00	1.09e+00	-	5.46e-01	8.19e-01

Table D.40: Main results of scenario `05c_worse_references`.

	Δt_{run}	MSD_a	$NMAE_a$	MSD_b	$NMAE_b$	MSD_{σ_y}	$NMAE_{\sigma_y}$	MSD_X	MSE_X	$NMSE_X$
gibbs_base	/	/	/	/	/	/	/	/	/	/
gibbs_minimal	1.50e+02	-6.09e-01	2.96e+02	2.41e-01	1.45e+01	1.12e+00	5.17e+00	2.75e-01	4.63e-01	6.00e-01
gibbs_no_EIV	1.37e+02	-6.44e-01	9.68e+01	6.37e-01	1.83e+02	9.39e-01	2.82e+00	1.71e-02	4.56e-01	7.76e-01
gibbs_known_sigma_y	7.70e+00	-6.34e-01	5.67e+02	-1.13e-02	1.91e+00	-	-	4.84e-01	6.69e-01	2.44e+04
joint_posterior	6.06e+01	-5.55e-01	3.00e+01	1.65e-01	3.61e+00	3.03e-01	6.89e+00	2.79e-01	3.77e-01	4.67e+00
joint_posterior_agrid	1.41e+01	-3.46e-01	2.65e+01	-9.25e-02	4.70e+00	4.70e-02	6.64e+00	2.70e-01	1.64e-01	1.89e+01
stankovic_base	3.96e-01	5.73e-01	1.05e+00	8.84e-02	1.08e-01	-	-	-2.63e-01	1.70e-01	6.95e-01
stankovic_enhanced_unc	4.16e-01	5.61e-01	1.04e+00	8.79e-02	1.08e-01	-	-	-2.59e-01	1.65e-01	6.73e-01
stankovic_base_unc	3.75e-01	5.73e-01	1.05e+00	8.84e-02	1.08e-01	-	-	-2.63e-01	1.70e-01	6.95e-01

Table D.41: Runtime and consistency metrics of scenario `05c_worse_references`.

	$s_a(4s) > s_a(16s)$	$s_b(4s) > s_b(16s)$	$s_\sigma(4s) > s_\sigma(16s)$	$t_a(0.1)$	$t_a^*(0.1)$	$t_b(0.1)$	$t_b^*(0.1)$	$t_\sigma(0.1)$	$t_\sigma^*(0.1)$
gibbs_base	/	/	/	/	/	/	/	/	/
gibbs_minimal	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	9.99e+00	never	never
gibbs_no_EIV	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	never	never
gibbs_known_sigma_y	yes	yes	-	1.99e+00	1.99e+00	1.99e+00	1.99e+00	-	-
joint_posterior	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	3.99e+00	7.99e+00
joint_posterior_agrid	yes	yes	yes	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00	1.99e+00
stankovic_base	No	No	-	never	never	never	never	-	-
stankovic_enhanced_unc	No	No	-	never	never	never	never	-	-
stankovic_base_unc	No	No	-	never	never	never	never	-	-

Table D.42: Convergence metrics of scenario `05c_worse_references`.

E Formal Requirements

A german translation of the abstract, a short vita of the author and a list of publications is provided.

E.1 Zusammenfassung

Große Sensornetzwerke bilden einen wichtigen Teil des industriellen Internets der Dinge. Um den Betrieb solcher Netzwerke über lange Zeiträume zu gewährleisten, muss die Qualität der Messwerte gesichert sein. In diesem Zusammenhang wird eine metrologisch rückführbare in-situ Kalibrierungsmethode entwickelt, die auf Bayesschen Prinzipien aufbaut und lokale Sensorredundanzen ausnutzt. Die Automatisierung solcher in-situ Kalibrierungen spielt ebenfalls eine wesentliche Rolle. Um dies zu ermöglichen wird eine Erweiterung bestehender Ontologien mit Sensorbezug vorgeschlagen, sodass metrologisch relevante Eigenschaften abgedeckt werden können. Aufbauend auf diesen Wissensrepräsentationen können Sensorselbstbeschreibungen erstellt werden, die es ermöglichen in-situ Kalibrierungen beim Finden von passenden Referenzsensoren und der Initialisierung der mathematischen Methode zu unterstützen. Die mathematische Methode wird in Simulationsstudien bewertet und mit einer dem Stand der Technik entsprechenden in-situ Kalibriermethode verglichen. Die Evaluierungsergebnisse zeigen eine gute Schätzleistung in Szenarien mit zeitabhängigen Eingangssignalen sowie bei Sensoren mit vergleichbarem Unsicherheitsniveau, offenbaren jedoch auch einen erhöhten Rechenaufwand. Die entwickelten Ontologien werden durch einen Korpusvergleich, Ontologiemetriken und logische Prüfungen der hinterlegten Taxonomie geprüft und zeigen eine gute Übereinstimmung zu bestehenden Qualitätsstandards von Ontologien.

E.2 Curriculum Vitae

The vita is not included in the electronic publication for privacy protection reasons.

E.3 Publications the Author was Involved In

In addition to the publications already listed in table 1.2.1 (which directly contribute to this thesis), the author also contributed to [22, 154, 155, 156, 157, 158, 159]. Hence, the full list of publications that the author was involved in is:[23]

(R,J) Tanja Dorst, Maximilian Gruber, Benedikt Seeger, Anupam Prasad Vedurmudi, Tizian Schneider, Sascha Eichstädt, and Andreas Schütze. „Uncertainty-Aware Data Pipeline of Calibrated MEMS Sensors Used for Machine Learning". In: *Measurement: Sensors* 22 (Aug. 1, 2022), p. 100376. ISSN: 2665-9174. DOI: `10.1016/j.measen.2022.100376`

(R,J) Tanja Dorst, Maximilian Gruber, Anupam P. Vedurmudi, Daniel Hutzschenreuter, Sascha Eichstädt, and Andreas Schütze. „A Case Study on Providing FAIR and Metrologically Traceable Data Sets". In: *Acta IMEKO* 12.1 (1 Mar. 28, 2023), pp. 1–6. ISSN: 2221-870X. DOI: `10.21014/actaimeko.v12i1.1401`

(R,C) Sascha Eichstädt, Anupam Prasad Vedurmudi, Maximilian Gruber, and Daniel Hutzschenreuter. „Fundamental Aspects in Data Analysis for Sensor Network Metrology". *Proceedings of the First International IMEKO TC6 Conference on Metrology and Digital Transformation - M4Dconf2022*. First International IMEKO TC6 Conference on Metrology and Digital Transformation. Berlin, GERMANY: IMEKO, 2022, pp. 1–4. DOI: `10.21014/tc6-2022.030`

(R,J) Sascha Eichstädt, Maximilian Gruber, Anupam Prasad Vedurmudi, Benedikt Seeger, Thomas Bruns, and Gertjan Kok. „Toward Smart Traceability for Digital Sensors and the Industrial Internet of Things". In: *Sensors* 21.6 (6 Jan. 2021), p. 2019. ISSN: 1424-8220. DOI: `10.3390/s21062019`

(B) Sascha Eichstädt, ed. *Dynamic Measuring Systems: Fundamentals and Application of Time-Dependent Measurements*. De Gruyter Series in Measurement Sciences. De Gruyter Oldenbourg, 2023. ISBN: 978-3-11-071310-7. URL: `https://www.degruyter.com/document/isbn/9783110713107/html` (visited on 11/04/2022)

(R,C) Maximilian Gruber, Sascha Eichstädt, Julia Neumann, and Adrian Paschke. „Semantic Information in Sensor Networks: How to Combine Existing Ontologies, Vocabularies and Data Schemes to Fit a Metrology Use Case". *2020 IEEE International Workshop on Metrology for Industry 4.0 IoT*. 2020 IEEE International Workshop on Metrology for Industry 4.0 IoT. June 2020, pp. 469–473. DOI: `10.1109/MetroInd4.0IoT48571.2020.9138282`

(R,C) Maximilian Gruber, Sascha Eichstädt, Wenzel Pilar von Pilchau, Jörg Hähner, Varun Gowtham, Alexander Willner, Nikolaos-Stefanos Koutrakis, Julian Polte, and Matthias Riedl. „Uncertainty-Aware Sensor Fusion in Sensor Networks". In: *SMSI 2021 - Measurement Science* (May 3, 2021), pp. 246–247. DOI: `10.5162/SMSI2021/D2.2`

(R,J) Maximilian Gruber, Tanja Dorst, Andreas Schütze, Sascha Eichstädt, and Clemens Elster. „Discrete Wavelet Transform on Uncertain Data: Efficient Online Implementation for Practical Applications". *Advanced Mathematical and Computational Tools in Metrology and Testing XII*. vol. Volume 90. Series on Advances in Mathematics for Applied Sciences Volume 90. WORLD SCIENTIFIC, May 31, 2021, pp. 249–261. ISBN: 9789811242373. DOI: `10.1142/9789811242380_0014`

[23] **R**: Peer Review, **C**: Conference Proceeding, **J**: Journal Paper, **B**: Book chapter

(R,C) Maximilian Gruber and Sascha Eichstädt. „Representing Semantic Information in Sensor Networks“. In: *SMSI 2021 - System of Units and Metreological Infrastructure* (May 3, 2021), pp. 316–317. DOI: 10.5162/SMSI2021/D1.2

(R,C) Maximilian Gruber, Wenzel Pilar von Pilchau, Varun Gowtham, Nikolaos-Stefanos Koutrakis, Nicolas Schönborn, Sascha Eichstädt, Jörg Hähner, Marius-Iulian Corici, Thomas Magedanz, Julian Polte, and Eckart Uhlmann. „Application of Uncertainty-Aware Sensor Fusion in Physical Sensor Networks“. *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. 2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). May 2022, pp. 1–6. DOI: 10.1109/I2MTC48687.2022.9806580

(R,C) Maximilian Gruber, Sascha Eichstädt, and Anupam P. Vedurmudi. „Co-Calibration in Distributed Homogeneous Sensor Networks“. In: *Lectures* (May 8, 2023), pp. 47–48. DOI: 10.5162/SMSI2023/A3.2

(B) Maximilian Gruber, Sascha Eichstädt, and Clemens Elster. „Modeling Dynamic Measurements in Metrology and Propagation of Uncertainties“. *Dynamic Measuring Systems: Fundamentals and Application of Time-Dependent Measurements*. De Gruyter Series in Measurement Sciences. De Gruyter Oldenbourg, 2023

(R,C) Wenzel Pilar von Pilchau, Varun Gowtham, Maximilian Gruber, Matthias Riedl, Nikolaos-Stefanos Koutrakis, Jawad Tayyub, Jörg Hähner, Sascha Eichstädt, Eckart Uhlmann, Julian Polte, Volker Frey, and Alexander Willner. „An Architectural Design for Measurement Uncertainty Evaluation in Cyber-Physical Systems“. *Annals of Computer Science and Information Systems*. Position Papers of the 2020 Federated Conference on Computer Science and Information Systems. Vol. 22. 2020, pp. 53–57. ISBN: 978-83-959183-0-8. URL: https://annals-csis.org/proceedings/2020/drp/203.html (visited on 04/28/2023)

(R,C) Anupam Prasad Vedurmudi, Maximilian Gruber, Sascha Eichstädt, and Adrian Paschke. „Semantics in Sensor Networks: An Ontology for Dynamic Transfer Behavior in Calibrated Sensors“. *2021 IEEE International Workshop on Metrology for Industry 4.0 IoT (MetroInd4.0 IoT)*. 2021 IEEE International Workshop on Metrology for Industry 4.0 IoT (MetroInd4.0 IoT). June 2021, pp. 358–363. DOI: 10.1109/MetroInd4.0IoT51437.2021.9488554

(R,J) Anupam Prasad Vedurmudi, Julia Neumann, Maximilian Gruber, and Sascha Eichstädt. „Semantic Description of Quality of Data in Sensor Networks“. In: *Sensors* 21.19 (19 Jan. 2021), p. 6462. ISSN: 1424-8220. DOI: 10.3390/s21196462