POLITECNICO DI TORINO

Department of Mechanical and Aerospace Engineering

Master of Science in Mechanical Engineering

State and Parameter Estimation of Nonlinear Dynamics Systems: Kalman Filter, Unscented Kalman Filter, and Particle Filter (Monte Carlo Scheme)

Supervisors:

Dario Anastasio (Politecnico di Torino)

Ludovic Renson (Imperial College London)

Alessandro Fasana (Politecnico di Torino)

Student:

Benyamin Bahmani Ghoje Biglu (S289953)

October, 2023

Abstract:

The study of nonlinear dynamic systems presents complex challenges across various scientific domains, notably in engineering and physics. This research addresses the critical need for state and parameter estimation through the development of innovative Bayesian inference techniques, with a specific focus on Kalman filters, Unscented Kalman filters, and Particle filters. The primary objective is to bridge existing gaps in parameter estimation methods, creating techniques that not only fit data but also unveil the fundamental relationships governing system behaviors. This research establishes a robust foundation in the realm of nonlinear dynamic systems, introducing the theoretical underpinnings of Bayesian inference techniques. It provides a clear framework for numerical implementation and demonstrates practical utility in real-world scenarios. To this end, numerical examples of nonlinear mechanical systems are considered first, considering both process and measurement uncertainties. Then, experimental data of a geometrically nonlinear vibrating system is collected and used to validate the proposed approaches. The developed Bayesian inference techniques display promising results in estimating parameters, offering insights into complex system behaviors. This work contributes to the field of nonlinear dynamics, enhancing our ability to understand and manipulate complex behaviors while addressing the critical demand for model parameter estimation.

Acknowledgments

I would like to express my deepest gratitude to my supervisors, Professor Ludovic Renson from Imperial College London and Professor Dario Anastasio and Alessandro Fasana from Politecnico di Torino, for their invaluable guidance, support, and unwavering commitment to this thesis. Professor Renson, your initial idea and inspiration for this research project laid the foundation for this work, and your mentorship throughout has been instrumental in shaping its direction. Professor Anastasio, your dedication, expertise, and the countless hours you invested in advising and mentoring me are deeply appreciated. Benjamin Smart, a doctoral student of Professor Renson at Imperial College London, also deserves special mention for his assistance, guidance, and contributions to this research.

I am also thankful to the entire faculty and staff of both institutions for providing me with the resources and academic environment necessary for the successful completion of this research. The opportunities and experiences I gained through the collaboration between Politecnico di Torino and Imperial College London have been enriching and enlightening.

I must extend my gratitude to my dear friend Fatemeh Vosoughi who is more like a family and provided support, and encouragement, and insightful discussions throughout this journey.

To my family, your unwavering belief in me and your constant encouragement were the pillars of strength that sustained me during the challenges of this research. I am deeply grateful for your love and support.

This thesis would not have been possible without the collective support and collaboration of all those mentioned above. Thank you for being a part of this journey.

Benyamin Bahmani G.B.

Contents

General Introduction

In today's world, we encounter nonlinear dynamic systems in various scientific domains, from engineering to physics. These systems often present complex challenges, particularly when it comes to estimating their states and parameters accurately. To address this critical need, this research focuses on the development of Kalman filters, Unscented Kalman filters, and Particle filters which belong to the class of Bayesian Filters (Ho and Lee, 1964; Lee, 1964; Jazwinski, 1966, 1970).

Our primary goal is to bridge existing gaps in parameter estimation methods. Rather than merely fitting data, we aim to uncover the fundamental relationships governing the behaviors of these systems. This work lays a robust foundation in the realm of nonlinear dynamic systems and introduces the theoretical underpinnings of Bayesian inference techniques (Bernardo and Smith, 1994; Gelman et al., 2004).

Our research provides a clear framework for numerical implementation and demonstrates practical utility in real-world scenarios. We begin by considering numerical examples of nonlinear mechanical systems, taking into account both process and measurement uncertainties. Furthermore, we collect experimental data from a geometrically nonlinear vibrating system to validate the proposed approaches.

The Bayesian inference techniques we develop offer promising results in estimating parameters, providing valuable insights into the complex behaviors of these systems (Sarkka, 2023; Sarkka 2013). This work contributes significantly to the field of nonlinear dynamics by enhancing our ability to understand and manipulate these complex behaviors while addressing the critical demand for accurate model parameter estimation.

In a world where nonlinear dynamic systems play a central role in various fields, from swaying bridges to intricate molecular interactions, our research is not just an academic endeavor; it has the potential to reshape industries and drive scientific exploration. Our contributions lie in the innovations of Bayesian techniques and in enabling researchers and professionals to understand and manipulate nonlinear behaviors more effectively.

This introduction sets the stage for our journey into the world of nonlinear dynamic systems, where complexities and discoveries intertwine, and where we aim to shed light on their intricate behaviors and relationships.

Chapter 1. Nonlinear Dynamics Systems

1.1 Nonlinear Dynamics Systems Definition and Contrast with Linear Systems

A nonlinear dynamics system refers to a mathematical model that describes the behavior of a system where the relationship between the system's variables and their rates of change is not described by linear equations. In other words, the change in the system's state over time is not directly proportional to the system's current state. Instead, the dynamics of nonlinear systems are influenced by the interaction and interdependency of variables in complex ways, often leading to fascinating and unpredictable behavior.

To better understand nonlinear dynamics systems, it is essential to contrast them with their linear counterparts (Fuchs, 2014). In linear systems, the principle of proportionality and superposition holds, making their behavior predictable and straightforward. When the input to a linear system is doubled, the output also doubles, and the principle of superposition allows us to combine individual solutions to obtain the overall response.

In contrast, nonlinear systems exhibit non-proportional and non-additive behaviors. This means that a doubling of the input may not necessarily result in a doubling of the output. Additionally, the principle of superposition does not apply to nonlinear systems, making it challenging to decompose complex behavior into simpler components.

One classic example of a nonlinear dynamics system is the double pendulum. A double pendulum consists of two connected pendulums, where the motion of the lower pendulum is affected by both gravity and the motion of the upper pendulum. The behavior of the double pendulum is highly sensitive to its initial conditions, meaning that even slight differences in the starting positions or velocities of the pendulums can lead to dramatically different trajectories.

In a linear system, the motion of each pendulum would be independent of the other, and their behavior would be predictable and stable. However, the interdependency between the two pendulums in the double pendulum system results in chaotic and complex motion, making it an excellent example of a nonlinear dynamics system.

Nonlinear dynamics systems are of significant interest to scientists, engineers, and researchers across various disciplines. They play a critical role in modeling and understanding complex natural phenomena, such as weather patterns, fluid dynamics, and population dynamics. Additionally, nonlinear systems are prevalent in engineering applications, including robotics, control systems, and communication networks.

The study of nonlinear dynamics systems has important implications in fields such as chaos theory, dynamical systems theory, and complex systems research (Schön, 2006). By investigating nonlinear dynamics, researchers can gain insights into the emergence of patterns, bifurcations, and other complex behaviors, that are not present in linear systems.

Modeling nonlinear dynamics systems typically involves expressing the relationships between the system's variables and their rates of change using nonlinear differential equations. Solving these equations analytically is often infeasible due to their complexity, and numerical methods are used for simulation and analysis.

To analyze the behavior of nonlinear systems, researchers often employ techniques from dynamical systems theory. This theoretical framework enables the study of stability properties of equilibrium points, the identification of periodic orbits or limit cycles, and the detection of bifurcations, where the system's behavior changes qualitatively as parameters are varied.

Overall, nonlinear dynamics systems are an essential area of study with broad applications in understanding the complexities of the natural world and engineering systems. Their unique behaviors challenge researchers to develop sophisticated mathematical and computational tools, making them an exciting and dynamic field of research.

1.2 Difference between Deterministic and Stochastic Systems

Deterministic nonlinear systems are characterized by a predictable and deterministic relationship between the system's variables and their rates of change (Fuchs, 2014 ; Schön, 2006). In other words, given the initial conditions of the system, its future behavior is entirely determined and repeatable. The dynamics of deterministic systems are governed by deterministic equations, meaning that there is no inherent randomness or uncertainty in the system's evolution over time.

Consider a simple harmonic oscillator, a classic example of a deterministic nonlinear system. The motion of the oscillator is governed by a second-order nonlinear differential equation that relates the position and velocity of the oscillator to an external force. Given the initial position and velocity of the oscillator, its future trajectory can be precisely predicted using the laws of physics and calculus.

Stochastic systems (Maybeck, 1982a; Stengel, 1994), on the other hand, introduce an element of randomness or probabilistic behavior into the system's dynamics. In these systems, the future evolution of the system is uncertain, even when the initial conditions are known with certainty. The dynamics of stochastic systems are described by stochastic differential equations (SDEs) or stochastic difference equations, which incorporate random or probabilistic elements.

A well-known example of a stochastic nonlinear system is the Brownian motion or random walk. Brownian motion describes the random movement of particles suspended in a fluid due to collisions with molecules of the fluid. In this system, the motion of the particle is influenced by random and unpredictable collisions, leading to a continuous and erratic movement over time.

In some cases, systems may exhibit a combination of deterministic and stochastic behavior. These are referred to as hybrid systems. For example, in certain biological processes, deterministic processes may govern the overall behavior of the system, while random events, such as mutations or environmental fluctuations, introduce stochasticity.

Understanding the distinction between deterministic and stochastic nonlinear systems is crucial for selecting appropriate modeling techniques and analysis methods depending on the specific characteristics and uncertainties present in the system under investigation. Stochastic models are particularly useful when dealing with real-world systems subject to random events, providing a more realistic representation of the inherent uncertainty in these complex systems.

1.3 State-of-the-Art in Nonlinear Dynamics Systems

Before delving further into the specifics of this thesis, it is essential to review the state-of-the-art in the field of nonlinear dynamics systems. A comprehensive understanding of prior research and existing methodologies is critical for identifying gaps and opportunities for innovation.

1.3.1 Prior Research and Key Methodologies Used in Modeling Nonlinear Systems

A substantial body of research has been dedicated to nonlinear dynamics systems, encompassing both theoretical developments and practical applications. Researchers have explored various mathematical and computational techniques for modeling and analyzing these systems. Some of the prominent areas of study include:

1. Differential Equations: Differential equations are at the core of modeling nonlinear dynamics systems. They establish the relationship between the system's state variables and their rates of change over time. In general, differential equations can be classified into ordinary differential equations (ODEs) for continuous-time systems or partial differential equations (PDEs) for systems that depend on both time and space variables (Øksendal, 2003, L Renson, 2015, L Renson, 2018, D Anastasio, 2020).

   Nonlinear differential equations are expressed as functions of the state variables and their derivatives, and they may involve products, powers, trigonometric functions, exponential functions, or other nonlinear terms. Solving these equations analytically is often challenging or even impossible, requiring numerical methods for simulation and analysis.

2. State Space Representation: State-space representation is a mathematical technique used to model both linear and nonlinear systems in a more systematic and compact form (V Verdult, 2002). It involves converting a higher-order system of differential equations into a set of first-order equations in a state vector format.

The state vector contains all the system's state variables, and the dynamics of the system are expressed as a set of first-order ordinary differential equations. This transformation simplifies the analysis of complex systems, facilitates the use of linear algebra tools, and enables the application of control theory and system identification methods.

3. Dynamical Systems Theory: Dynamical systems theory provides a theoretical framework for analyzing the behavior of nonlinear systems. It focuses on understanding the stability and behavior of equilibrium points, periodic orbits, limit cycles, and chaotic attractors.

   Stability analysis is critical for determining the long-term behavior of the system. Stable equilibrium points act as attractors that draw the system towards a particular state, while unstable equilibrium points lead to divergent trajectories. Periodic orbits and limit cycles represent regular repeating patterns in the system's state, while chaotic attractors exhibit sensitive dependence on initial conditions and aperiodic behavior.

4. Numerical Methods: Given the complexity of most nonlinear differential equations, analytical solutions are often impractical. Numerical methods play a fundamental role in solving these equations and simulating the behavior of nonlinear systems over time. Common numerical methods include Euler's method, Runge-Kutta methods (E Hairer, 2006), and finite difference methods for ODEs, as well as finite element methods and finite volume methods for PDEs.

   Numerical simulations enable the investigation of system behavior under different conditions, the exploration of bifurcations, and the study of parameter sensitivity. These simulations provide valuable insights into the dynamics of nonlinear systems when analytical solutions are not attainable.

5. Nonlinear Control Techniques: Controlling nonlinear systems presents unique challenges, as the relationship between inputs and outputs is not necessarily linear. Nonlinear control techniques aim to stabilize and control the behavior of nonlinear systems effectively. Techniques like feedback linearization, sliding mode control, and adaptive control are utilized to handle the complexities of these systems.

6. Chaos Theory and Bifurcation Analysis: Chaos theory is a branch of mathematics that deals with the study of chaotic behavior in deterministic nonlinear systems. Chaos arises from sensitive dependence on initial conditions, which means that infinitesimal differences in initial conditions can lead to vastly different trajectories over time. Bifurcation analysis focuses on the study of qualitative changes in the system's behavior as a parameter is varied. Bifurcations can lead to the emergence of new behaviors, such as the appearance of periodic orbits or chaotic attractors.

By employing these key mathematical concepts, researchers and engineers can develop accurate and insightful models of nonlinear systems. These models facilitate the understanding of complex behaviors, provide predictive capabilities, and enable the design

of efficient control strategies for a wide range of applications across various fields of science and engineering.

1.3.2 Identified Research Gaps

Despite the progress made in modeling and analyzing nonlinear dynamics systems, several challenges and gaps persist in the field:

Parameter Estimation: Accurate parameter estimation remains a significant challenge due to the inherent nonlinearity of these systems. Current methodologies often struggle to provide reliable parameter estimates, especially when dealing with complex and high-dimensional systems.

Incorporating Uncertainty: Many real-world systems are subject to uncertainties and noise. Current approaches may not adequately address the incorporation of these uncertainties into models, leading to suboptimal predictions.

Computational Complexity: The computational demands of modeling and analyzing nonlinear systems, particularly in high-dimensional spaces, necessitate the development of efficient algorithms and numerical techniques.

1.4 Challenges in Parameter Estimation for Nonlinear Systems

Parameter estimation is a critical process in modeling nonlinear systems, as it involves determining the values of unknown parameters that best fit the model to observed data. However, parameter estimation in nonlinear systems poses several challenges due to their complex and dynamic nature. Understanding and addressing these challenges are essential to obtain accurate and reliable parameter estimates. Below are the key challenges in parameter estimation for nonlinear systems:

1. Nonlinearity: The primary challenge in parameter estimation for nonlinear systems is the inherent nonlinearity of the system dynamics. Nonlinear relationships between variables can lead to multiple solutions or ambiguities in parameter estimation. Unlike linear systems where parameter estimation is often straightforward, nonlinear systems require more sophisticated and iterative techniques to identify the optimal set of parameters that best represent the observed data.

2. Uncertainty and Noise: Real-world data collected from measurements are often affected by uncertainties and noise. These uncertainties can arise from various sources, such as sensor inaccuracies, measurement errors, and environmental variability. Incorporating these uncertainties into the parameter estimation process is crucial for obtaining reliable parameter estimates that account for the inherent noise in the data.

3. High-Dimensional State Spaces: Many nonlinear systems have high-dimensional state spaces, meaning they involve a large number of state variables. Estimating parameters

in high-dimensional spaces can be computationally intensive and may require specialized algorithms to efficiently explore the parameter space. High-dimensional systems also suffer from the "curse of dimensionality," where the volume of the parameter space grows exponentially with the number of dimensions, making exhaustive search methods impractical.

4. Identifiability and Sensitivity Analysis: Identifiability refers to the uniqueness of parameter estimation. In some cases, nonlinear systems may exhibit parameter redundancy, meaning different combinations of parameters can produce similar system behavior. Identifiability analysis helps determine which parameters can be uniquely estimated from the available data and which are partially or fully unidentifiable.

   Sensitivity analysis is crucial in assessing the impact of uncertainties in parameter estimates on the system's behavior. Small errors in parameter estimation can lead to significant deviations in system predictions, especially in chaotic or sensitive systems.

5. Model Structure Uncertainty: Choosing an appropriate model structure for a nonlinear system is often a challenging task. In some cases, the exact form of the nonlinear model may not be known a priori, and different model structures could fit the data equally well. Model structure uncertainty adds another layer of complexity to parameter estimation, as researchers must consider various plausible model formulations.

6. Global Optimization: Nonlinear parameter estimation often involves finding the global optimum in the parameter space, which can be challenging due to the presence of multiple local minima or maxima. Traditional optimization methods might get stuck in local optima, leading to suboptimal parameter estimates. Global optimization techniques, such as genetic algorithms, particle swarm optimization, or simulated annealing, are employed to address this challenge.

7. Data Sparsity and Observability: In some cases, data sparsity or limited observability of the system can hinder parameter estimation. Insufficient or sparse data may not fully capture the system's dynamics, leading to biased or inaccurate parameter estimates. Observability analysis helps determine which parts of the system can be inferred from available data and which parts remain unobservable.

8. Computational Complexity: Parameter estimation in nonlinear systems can be computationally demanding, especially for large-scale or high-dimensional models. Efficient algorithms and numerical techniques are required to handle the computational burden and provide reasonable estimation times.

To overcome these challenges, researchers often leverage Bayesian inference techniques, such as the Kalman filter, Unscented Kalman filter, and particle filter. Bayesian approaches provide a principled way to incorporate prior knowledge, handle uncertainty, and iteratively update parameter estimates as new data becomes available. These techniques allow for more robust and accurate parameter estimation, enhancing the understanding and predictive capabilities of complex nonlinear systems.

1.5 Overview of Thesis

This thesis aims to contribute to the field of nonlinear dynamics by addressing the challenges outlined above, particularly in the context of parameter estimation. We will explore innovative Bayesian inference techniques, such as the Kalman filter, Unscented Kalman filter, and particle filter, to enhance our ability to accurately estimate parameters in nonlinear systems. The subsequent chapters will delve into the theoretical foundations and practical applications of these techniques, providing a comprehensive framework for researchers and practitioners working with nonlinear dynamics systems.

we will delve into the theoretical basics of Bayesian inference techniques, laying the groundwork for our subsequent discussions. Then we will focus on numerical implementation procedures and will showcase the application of these techniques in real-world scenarios. Finally, we will summarize our findings, discuss their broader implications, and highlight the contributions made in the realm of nonlinear dynamics.

As we embark on this journey, we aim to provide valuable insights, tools, and methodologies to researchers and professionals seeking to better understand and manipulate the intricate behaviors of nonlinear dynamics systems.

Chapter 2. Bayesian Inference

2.1 Overview of Bayesian Inference

Bayesian Inference is a sophisticated statistical framework that serves as a powerful tool for reasoning about uncertainty in a systematic and coherent manner (Tarantola, 2004; Kaipio and Somersalo, 2005; Rasmussen and Williams, 2006; Sarkka, 2011, L Renson, 2022). At its core, Bayesian Inference offers a structured methodology to update our beliefs, expressed in the form of probability distributions, as new data becomes available. This updating process enables us to refine our understanding of uncertain quantities, often referred to as parameters, within a given model.

The origin of Bayesian Inference can be traced back to the 18th century when Reverend Thomas Bayes pioneered the concept of integrating prior knowledge with observed evidence to arrive at updated probabilities (Wiener, 1950; Proakis, 2001; Hayes, 1996). Unlike classical frequentist statistics, which treat parameters as fixed values and data as random outcomes, Bayesian Inference treats both parameters and data as probabilistic entities. This fundamentally aligns with the intuitive notion that our knowledge about a parameter's true value is subject to uncertainty, often stemming from factors such as measurement errors, variability, or incomplete information.

Central to Bayesian Inference is the idea of the "prior" and the "likelihood." The prior captures our initial beliefs about a parameter's distribution before considering any new data. The likelihood quantifies the probability of observing the data given specific parameter values (Milton and Arnold, 1995). As new data is obtained, the prior is updated using Bayes' theorem, which mathematically combines the prior and the likelihood to yield the "posterior" distribution. This posterior distribution represents our revised beliefs about the parameter after incorporating the observed evidence.

The iterative nature of Bayesian Inference is particularly advantageous. As more data accumulates, the posterior distribution becomes increasingly refined, providing a comprehensive picture of the parameter's true distribution. This iterative updating aligns with the scientific method's principles—continuously refining hypotheses based on new evidence.

Bayesian Inference offers several compelling advantages over classical approaches. Firstly, it provides a coherent framework to incorporate prior information or expert knowledge into the analysis, effectively utilizing existing understanding to enhance data-driven conclusions. Secondly, it naturally handles uncertainty by representing it as a probability distribution, allowing for a nuanced exploration of potential outcomes and associated probabilities. Thirdly, Bayesian methods enable the propagation of uncertainty through complex models, supporting robust decision-making even in intricate systems.

This approach finds application in a diverse range of fields. From medical diagnoses and climate modeling to financial forecasting and artificial intelligence, Bayesian Inference

equips researchers and practitioners with a versatile tool to extract meaningful insights from data while acknowledging and quantifying the inherent uncertainty in the process. By offering a principled way to reason about uncertainty, Bayesian Inference is a cornerstone of modern statistical analysis and plays a pivotal role in advancing our understanding of complex phenomena.

2.2 Bayesian Underlying Principles:

2.2.1 Probabilistic Framework: Central to the essence of Bayesian inference is its foundation in a probabilistic framework (Stratonovich, 1968; Jazwinski, 1970). This framework employs probability as a precise language to quantify and express our understanding of uncertainty in the context of parameters and observed data. In the Bayesian perspective, parameters are not regarded as fixed but as random variables, characterized by probability distributions. By doing so, Bayesian analysis seamlessly integrates the quantification of uncertainty into the entire inference process, leading to coherent and intuitive reasoning.

2.2.2 Incorporating Prior Knowledge: A distinctive feature of Bayesian inference is its capability to incorporate existing knowledge or beliefs regarding parameters before any data is collected. This is achieved through the introduction of a prior probability distribution over the parameter space. The prior distribution encapsulates what is known or assumed about the parameters based on prior information, historical data, or expert opinions. This integration of prior knowledge empowers Bayesian inference to leverage external information, thereby enhancing the precision and efficiency of parameter estimation.

2.2.3 Likelihood Function: Within the framework of Bayesian inference, the likelihood function stands as a cornerstone, embodying the essence of how observed data interacts with model parameters (Milton and Arnold, 1995). Essentially, it quantifies the plausibility of the collected data, given particular values of the model parameters. This function serves as a bridge between the theoretical predictions of the model and the actual data points observed.

Mathematically, the likelihood function $P(y \mid \mu)$ is expressed as:

$$P(y \mid \mu) = f(y \mid \mu)$$

Where:

- $P(y \mid \mu)$ is the likelihood of observing the data y given a set of parameter values μ.

- $f(y \mid \mu)$ represents the probability density function or probability mass function, depending on whether the data is continuous or discrete.

Interpretation and Significance: The crux of the likelihood function lies in its interpretation as a measure of how well the model, characterized by the parameter values θ, aligns with the actual data observations y. Larger values of $P(y \mid \mu)$ signify that the model's predictions, with the specified parameter values, closely correspond to the observed data points. Conversely, smaller values indicate a weaker match between the model's predictions and the observed data.

2.2.4 Bayes' Theorem: Bayes' theorem, a pivotal equation in Bayesian inference, governs the process of updating our beliefs about unknown parameters in response to new data. This theorem establishes a profound connection between the prior distribution, the likelihood function, and the posterior distribution. Named after Reverend Thomas Bayes (Wiener, 1950; Proakis, 2001; Hayes, 1996), it represents the crux of how probabilities are refined as new evidence is encountered.

Mathematically, Bayes' theorem is presented as:

$$P(\mu \mid y) = \frac{P(y \mid \mu) \cdot P(\mu)}{P(y)}$$

Where:

- $P(\mu \mid y)$ denotes the posterior distribution of parameters μ after observing data y.
- $P(y \mid \mu)$ is the likelihood function, which quantifies the probability of observing data y given specific values of the parameters μ.
- $P(\mu)$ signifies the prior distribution, which encapsulates the initial beliefs or knowledge about the parameter values before observing any data.
- $P(y)$ acts as the marginal likelihood or evidence, ensuring that the posterior distribution is a valid probability distribution.

Updating Beliefs using Bayes' Theorem: The journey of updating beliefs about parameters through Bayes' theorem unfolds in the following sequence:

1. Prior Beliefs: Before any data is collected, the prior distribution $P(\mu)$ captures the initial beliefs about parameter values. This distribution incorporates any prior knowledge, past studies, or external data.

2. Observing Data: As data is gathered, the likelihood function $P(y \mid \mu)$ steps in to assess how well the model, with specific parameter values, aligns with the observed data.
3. Updating Beliefs: By invoking Bayes' theorem, the prior distribution and likelihood function join forces to compute the posterior distribution $P(\mu \mid y)$. This distribution embodies our updated beliefs about the parameters after factoring in the observed data.

Mathematically, the theorem and its three-step application provide a structured and principled mechanism to refine our understanding of the parameters in light of new evidence. It captures the essence of Bayesian inference—adapting and improving our beliefs in a rigorous and iterative manner.

The updating process can be summarized as follows: As we observe new data, our beliefs about the parameters are revised in light of this new evidence. The prior distribution is combined with the likelihood function to obtain the posterior distribution, which reflects our updated understanding of the parameters. The posterior distribution is used to make probabilistic inferences, estimate credible intervals, perform model selection, and draw conclusions about unknown quantities.

2.2.5 Posterior Distribution: The culmination of Bayesian inference rests upon the posterior distribution, an embodiment of our refined beliefs regarding the model parameters following the assimilation of observed data. It serves as a harmonious synthesis between the prior knowledge and the new information gathered from the data. Unlike conventional frequentist methodologies that offer single-point estimations, Bayesian inference unveils a comprehensive probability distribution spanning the parameter space. This comprehensive distribution empowers us to not only quantify uncertainty but also draw probabilistic conclusions, elevating our grasp of the inherent variability in the system.

2.2.6 Iterative Updating: One of the hallmark attributes of Bayesian inference is its potential for iterative updating of parameter estimates as fresh data streams in. This facet is especially advantageous when data is collected sequentially or in batches. As each new data segment surfaces, the prior distribution transforms into the posterior distribution through the process of Bayesian updating. The iterative nature of this procedure permits adaptive decision-making, enabling the model to dynamically refine parameter estimates and learn from evolving data streams.

Here we can see the schematic aspect of Bayesian Inference for prior, likelihood, and posterior steps:

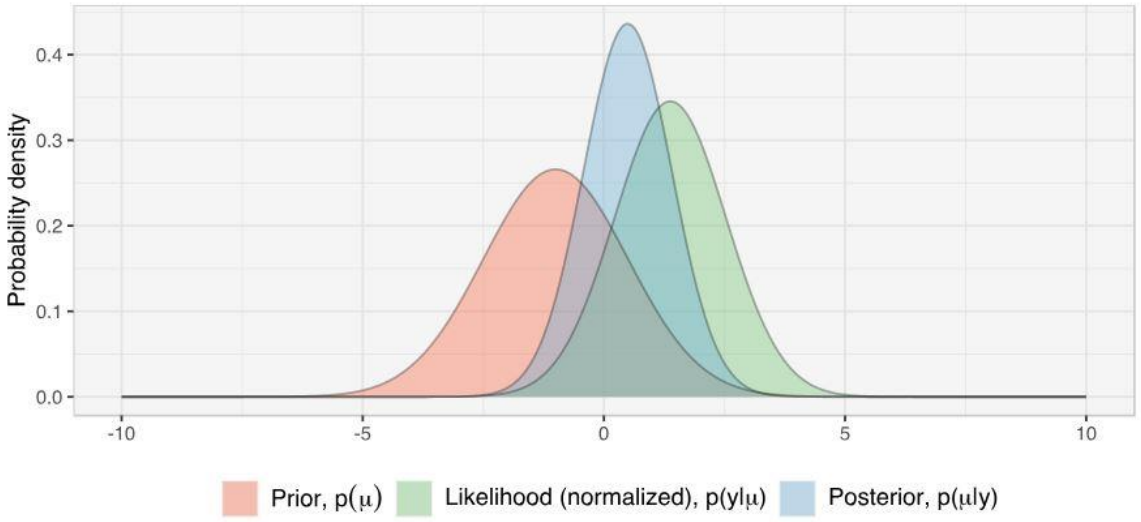


Figure 2. Prior, Likelihood, and Posterior aspects of Bayesian Inference

This iterative approach underscores the dynamic nature of Bayesian inference (E, Chatzi, 2010; E, Chatzi, 2009), where the model adapts to new insights, fosters ongoing learning, and facilitates informed decision-making in the face of evolving information. It embodies the essence of Bayesian reasoning by embracing the cyclical process of belief refinement.

Chapter 3. Kalman Filter

3.1 Introduction to the Kalman Filter

The Kalman filter stands as a cornerstone in the realm of recursive Bayesian estimation techniques, offering a powerful solution for estimating the state of linear dynamic systems in the presence of noise. Pioneered by Rudolf E. Kalman in the early 1960s (Ho and Lee, 1964; Lee, 1964; Jazwinski, 1966, 1970), this methodology has since emerged as a ubiquitous tool with applications spanning engineering, control systems, signal processing, robotics, and navigation. At its core, the Kalman filter is geared toward enabling accurate state estimation in real-time scenarios, even when systems are subject to various sources of uncertainty.

1. **Linear Dynamic Systems:** The Kalman filter is fundamentally designed to cater to linear dynamic systems. Such systems possess two integral attributes:
    - **Linearity:** Linearity denotes that the relationships governing the interaction between the system's state variables and inputs are described through linear equations. This implies that the system's behavior is encapsulated within linear combinations of its state variables and inputs.
    - **Dynamics:** Dynamics pertain to the temporal evolution of the system. Linear systems manifest their dynamics through linear differential or difference equations, representing the time-dependent transition of state variables.
2. **State-Space Representation:** Central to the Kalman filter's functioning is its operation on linear systems depicted using a state-space formulation. This representation dissects the model into two vital components: the state equation and the observation equation.
    - **State Equation:** The state equation encapsulates the evolution of the system's state variables across time. It typically assumes the form of a linear differential or difference equation, connecting the present state to the preceding state and the system inputs.
    - **Observation Equation:** The observation equation establishes the link between the actual measurements or observations and the underlying system state. This equation defines the mapping from the state variables to the observed measurements and takes into account the inherent measurement noise that often accompanies real-world data.
3. **Recursive Bayesian Estimation:** The Kalman filter's hallmark feature is its recursive Bayesian estimation paradigm, characterized by its ability to iteratively furnish state estimates at each time step. This recursive approach renders the Kalman filter exceptionally efficient and well-suited for real-time applications.
4. **Bayesian Filtering:** The Kalman filter is grounded in Bayesian filtering principles, framing the state estimation challenge as a probabilistic inference endeavor. Guided by

Bayes' theorem, the Kalman filter dynamically updates the state estimate as new measurements are assimilated.

5. **Prediction Step:** In each iteration, the Kalman filter enacts two core stages, namely prediction and update. In the prediction phase, the filter prognosticates the system's forthcoming state by leveraging the previous state estimate and the system's transition model. This step encompasses the system's dynamics and any available control inputs to propel the state estimate forward temporally.

6. **Update Step:** The subsequent update phase of the Kalman filter incorporates fresh measurements to refine and rectify the state estimate that emerged from the prediction phase. This adjustment process draws upon the observation model and measurement data to revise the state estimate, contingent upon the likelihood of the observed measurements given the forecasted state.

7. **State Estimation:** At each juncture, the Kalman filter furnishes two primary estimates:

   - **State Estimate (a posteriori):** This estimate encapsulates the optimal approximation of the genuine system state, incorporating the observed measurements. It arises as the outcome of the update step, serving as the present-day best estimate of the system's state.
   - **State Prediction (a priori):** This estimate arises from the prediction step prior to the assimilation of new measurements. It offers the subsequent state estimate, grounded solely in the system's dynamics and the antecedent state approximation.

8. **Covariance Matrix:** The Kalman filter supplements its estimates with the calculation of the covariance matrix at each time instance. This matrix quantifies the inherent uncertainty or inaccuracy intrinsic to the state estimate. It conveys the extent of dispersion or correlation existing among state variables.

9. **Adaptive Nature:** A distinctive attribute of the Kalman filter is its inherent adaptability. It capably navigates the intricacies of systems undergoing temporal alterations, varying noise attributes, and uncertainties present in real-time measurements. The Kalman filter persists in updating both its state approximation and covariance matrix, consistently refining its estimates in dynamically shifting environments.

10. **Extensions:** The Kalman filter's applicability extends through several variations tailored to specific scenarios. The **Extended Kalman Filter (EKF)** (Jazwinski, 1970; Maybeck, 1982b; Bar-Shalom et al., 2001; Grewal and Andrews, 2001) extends the basic Kalman filter to tackle non-linear dynamic systems by approximating the non-linear state and observation models via linearization. The **Unscented Kalman Filter (UKF)** (Julier et al., 1995; Julier and Uhlmann, 2004; Wan and Van der Merwe, 2001) offers an alternative, approximating non-linear transformations without necessitating explicit linearization. These extensions enhance the filter's applicability across a broader spectrum of real-world scenarios.

Here below we can summarize the steps in this figure.

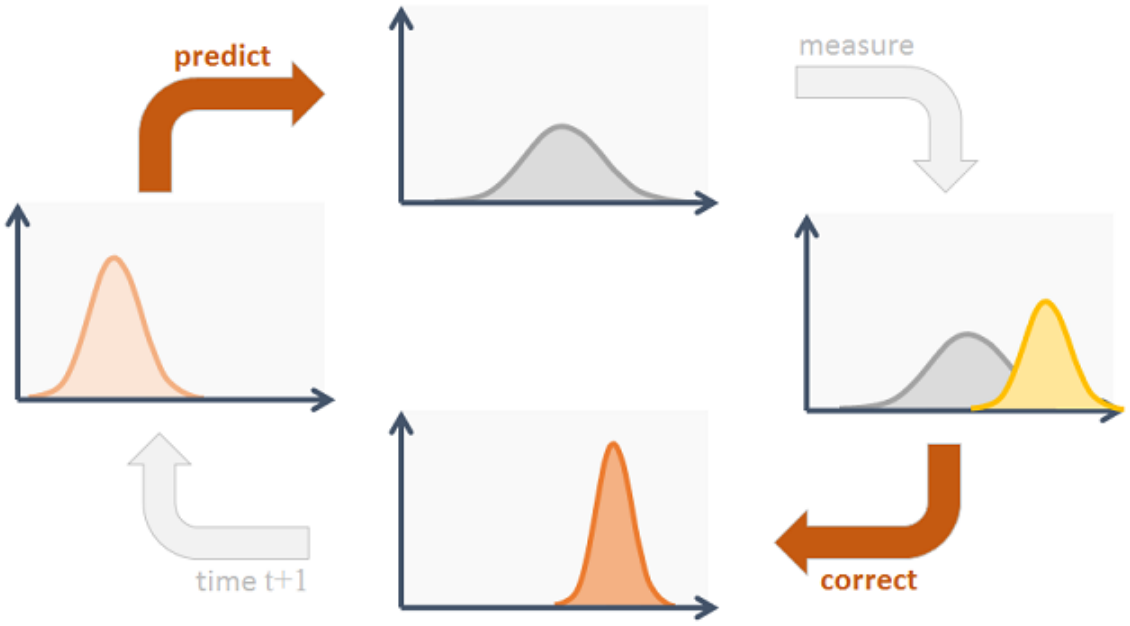


Figure 3. Kalman Filter Process

In conclusion, the Kalman filter is a versatile recursive Bayesian estimation technique designed specifically for linear dynamic systems. It provides an efficient and adaptive way to estimate the state of a system in the presence of noise and has found widespread applications in fields that involve state estimation, tracking, and sensor fusion in real-time systems (Bar-Shalom et al., 2001; Crassidis and Junkins, 2004; Challa et al., 2011). Its ability to incorporate both prior knowledge and real-time measurements makes it a valuable tool in various engineering and control applications.

3.2 Step-by-Step of the Kalman Filter Algorithm:

The Kalman filter is a recursive algorithm used to estimate the state of a linear dynamic system in the presence of noise. It operates in two main phases: prediction and update. Here is a detailed step-by-step explanation of the Kalman filter algorithm:

1. Initialization:

- Initialize the state vector $\boldsymbol{x}_0$ and the state covariance matrix $\boldsymbol{P}_0$ with their initial estimates or values.

- $\boldsymbol{x}_0 \in R^N$ represents the initial state estimate, while $\boldsymbol{P}_0 \in R^{NxN}$ that represents the initial uncertainty or error in the state estimate.

2. **Prediction Phase:** At each time step k, the Kalman Filter performs the prediction phase to predict the state estimate for the next time step (Kalman, 1960b; Kalman, 1961).

3. State Prediction (a priori):

- Use the state transition model to predict the state estimate at the next time step $\boldsymbol{x}_{k|k-1}$ based on the current state estimate $\boldsymbol{x}_{k-1|k-1}$ and the system dynamics.
- The state transition model is typically represented by a linear equation:

$$\boldsymbol{x}_{k|k-1} = \mathbf{F}_k * \mathbf{x}_{k-1|k-1} + \mathbf{B}_k * \mathbf{u}_k$$

Where:

$\boldsymbol{F}_k \in R^{NxN}$ is the state transition matrix.

$\boldsymbol{B}_k \in R^{Nx1}$ is the control input matrix (if any)

$\boldsymbol{u}_k \in R^{1x1}$ is the control input (if any).

4. State Covariance Prediction (a priori):

- Use the state transition model and the previous state covariance $\boldsymbol{P}_{k-1|k-1}$ (N $*$ N size) to predict the state covariance (Julier, 2000) at the next time step $\boldsymbol{P}_{k|k-1}$ (N $*$ N size) .
- The state covariance prediction is given by:

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k * \boldsymbol{P}_{k-1|k-1} * \boldsymbol{F}_k^{\mathrm{T}} + \boldsymbol{Q}_k$$

Where:

$\boldsymbol{Q}_k \in R^{NxN}$ is the process noise covariance matrix, representing the uncertainty introduced by the system dynamics and any unmodeled processes.

5. **Update Phase:** At each time step k, the Kalman Filter performs the update phase to incorporate new measurements and refine the state estimate.

6. Measurement Residual (Innovation):

- Calculate the measurement residual (also known as the innovation) $\boldsymbol{y_k}$ as the difference between the observed measurement $\boldsymbol{z_k}$ and the predicted measurement based on the state estimate $\boldsymbol{x_{k|k-1}}$.

$$\boldsymbol{y_k = z_k - H_k * x_{k|k-1}}$$

where $\boldsymbol{H_k} \in R^{1xN}$ is the observation matrix, which maps the state space to the measurement space.

7. Measurement Residual Covariance:

- Calculate the measurement residual covariance $\boldsymbol{S_k}$ as the sum of the measurement noise covariance $\boldsymbol{R_k}$ and the covariance of the predicted measurement.

$$\boldsymbol{S_k = H_k * P_{k|k-1} * H_k^{\intercal} + R_k}$$

where $\boldsymbol{R_k} \in R^{1X1}$ is the measurement noise covariance matrix, representing the uncertainty in the measurements.

8. Kalman Gain:

- Calculate the Kalman gain $\boldsymbol{K_k}$, which determines the contribution of the measurements to the state update.
- The Kalman gain is given by:

$$\boldsymbol{K_k = P_{k|k-1} * H_k^{\intercal} * S_k^{-1}}$$

9. State Update (a posteriori):

- Update the state estimate based on the measurements and the Kalman gain to obtain the a posteriori state estimate $\boldsymbol{x_{k|k}}$.

$$\boldsymbol{x_{k|k} = x_{k|k-1} + K_k * y_k}$$

10. State Covariance Update (a posteriori):

- Update the state covariance based on the Kalman gain to obtain the a posteriori state covariance $\boldsymbol{P_{k|k}}$.

$$\boldsymbol{P_{k|k} = (I - K_k * H_k) * P_{k|k-1}}$$

11. Repeat:

- Repeat the prediction and update phases for the next time step $k+1$ by using the updated state estimate $\boldsymbol{x}_{k|k}$ as the new initial state estimate $\boldsymbol{x}_{k+1|k}$ and the updated state covariance $\boldsymbol{P}_{k|k}$ as the new initial state covariance $\boldsymbol{P}_{k+1|k}$.

As good representation of Kalman filter we can refer to this graph below:

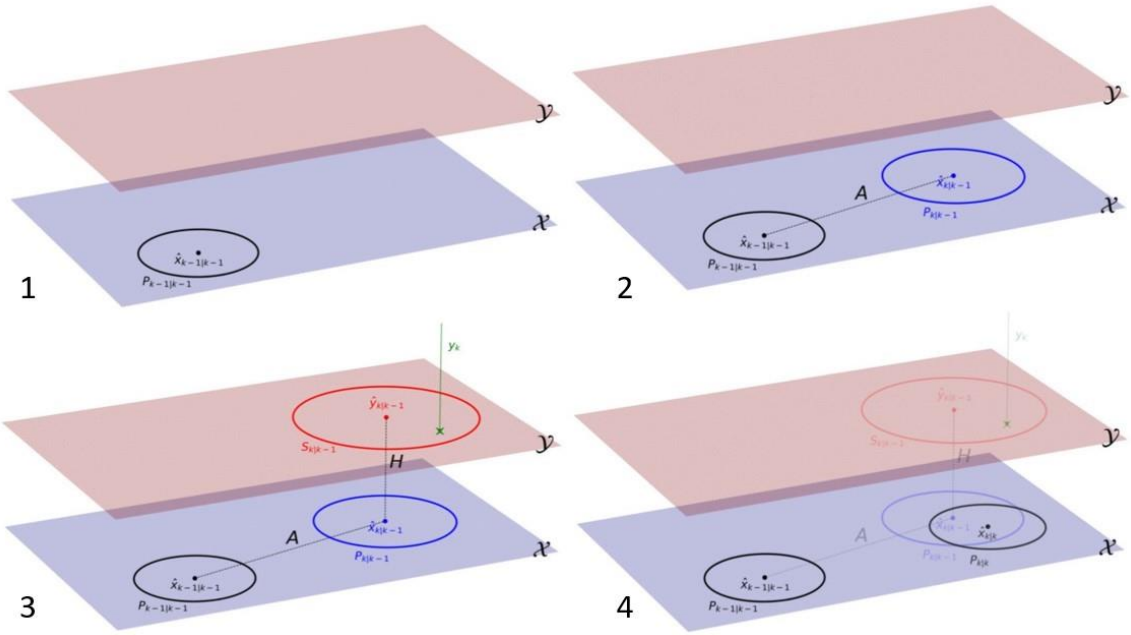


Figure 4. Kalman Filter prediction and update flow

The Kalman filter algorithm continually updates the state estimate and the state covariance at each time step as new measurements are received, providing an efficient and accurate estimate of the underlying system state even in the presence of noise and uncertainties. It is a widely used tool for state estimation, tracking, and sensor fusion in real-time systems.

3.3 Assumptions and Limitations of the Kalman Filter in Nonlinear Systems

The Kalman filter is a powerful tool for estimating the state of linear dynamic systems in the presence of noise. However, its effectiveness diminishes when applied to nonlinear systems. Here are the assumptions and limitations of the Kalman filter in handling nonlinear systems:

Assumptions:

1. **Linearity:** The Kalman filter assumes that the underlying dynamic system is linear. This assumption is crucial because the filter relies on linear algebra and statistical properties that are valid only for linear systems.
2. **Gaussian Noise:** The Kalman filter assumes that all the noise components (process noise and measurement noise) are Gaussian with zero mean and known covariance matrices. This assumption is essential for the optimal performance of the Kalman filter, as it relies on the properties of Gaussian distributions in its estimation process.
3. **Model Knowledge:** The Kalman filter assumes that the system dynamics and the observation model are known with certainty. This knowledge is necessary for setting up the state transition matrix ($\boldsymbol{F}_k$), the observation matrix ($\boldsymbol{H}_k$), and the covariance matrices ($\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$) accurately.
4. **Initial State:** The Kalman filter assumes that the initial state estimate and covariance matrix are known or can be estimated precisely. Accurate initialization is crucial, as errors in the initial state can propagate throughout the filtering process.

   The effect of good initialization and accurate initial value is important in the estimation process. If we use a not accurate or bad initial value it will take several sample until Kalman Filter find its flow. Here below we see a photo representing the case where we have a bad initial value.

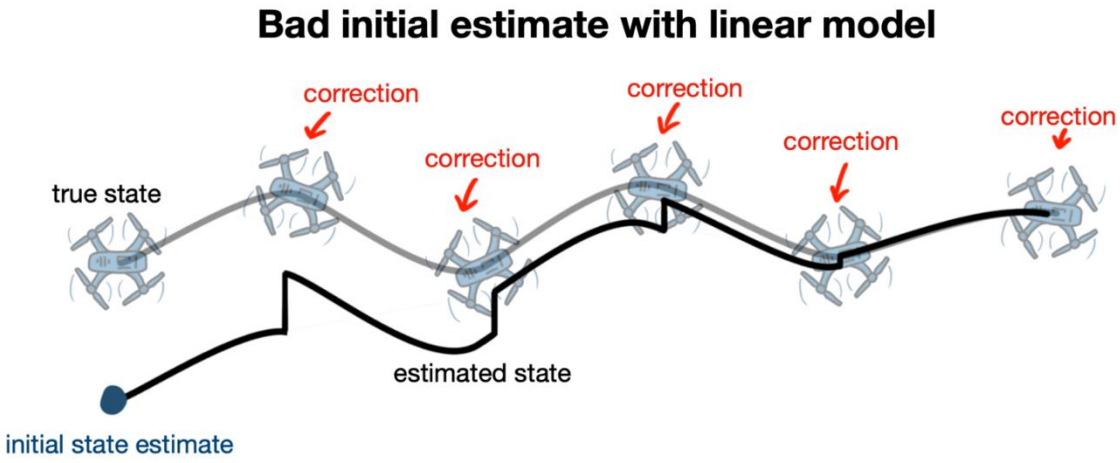


Figure 5. Bad initial value for Linear system Kalman Filter

Limitations:

1. **Nonlinear Systems:** The primary limitation of the Kalman filter is its inability to handle nonlinear dynamic systems. In nonlinear systems, the relationship between

the state variables and the system inputs becomes nonlinear, and the system dynamics cannot be represented by linear equations. As a result, the state transition model ($\boldsymbol{F}_k$) in the Kalman filter becomes invalid.

2. **Degeneracy:** In the presence of nonlinearity, the Kalman filter may suffer from degeneracy (sarkka, 2012a), where the updated state estimate and covariance matrix become unreliable due to the mismatch between the linear filter assumptions and the nonlinear system behavior.
3. **Loss of Optimality:** When applied to nonlinear systems, the Kalman filter loses its optimality property, meaning that it may no longer provide the best possible state estimate. The filter's optimality is derived from its ability to exploit the linear Gaussian structure of the system, which does not hold in nonlinear scenarios.
4. **Linearization Techniques:** Some attempts have been made to extend the Kalman filter to handle nonlinear systems by using linearization techniques such as the Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF). These methods approximate the nonlinear functions by their linear or unscented equivalents. However, these approximations may introduce errors, especially for highly nonlinear systems.
5. **Sensitivity to Initialization:** In nonlinear systems, the Kalman filter's performance can be highly sensitive to the quality of the initial state estimate and covariance. Poor initialization can lead to divergence or poor convergence, making it challenging to obtain accurate estimates.

3.4 Alternative Approaches for Nonlinear Systems

To address the limitations of the Kalman filter in handling nonlinear systems, alternative filtering techniques have been developed:

1. **Extended Kalman Filter (EKF):** The EKF linearizes the system dynamics and observation functions around the current state estimate. It uses a first-order Taylor series approximation to estimate the Jacobian matrices, enabling the Kalman filter to be applied to certain types of nonlinear systems. However, the EKF may suffer from approximation errors for highly nonlinear systems.
2. **Unscented Kalman Filter (UKF):** The UKF approximates the nonlinear functions using a set of carefully chosen sample points (sigma points) that capture the true mean and covariance of the system state. It propagates these sigma points through the nonlinear functions to compute the updated state estimate and covariance. The UKF often provides more accurate estimates than the EKF for moderately nonlinear systems.
3. **Particle Filter (Monte Carlo Filter):** The Particle Filter (Gordon et al., 1993; Kitagawa, 1996; Doucet et al., 2001; Ristic et al., 2004) represents the probability density function of the state using a set of weighted particles. It allows for more flexibility in handling

highly nonlinear systems and non-Gaussian noise (Kitagawa, 1987, 1996; Gordon, 1993). However, the Particle Filter can be computationally intensive and may suffer from the "curse of dimensionality" for high-dimensional state spaces.

In conclusion, while the Kalman filter is a powerful and widely used estimation technique for linear dynamic systems, it has important assumptions and limitations that restrict its applicability to nonlinear systems. For nonlinear systems, specialized filtering techniques such as the EKF, UKF, or Particle Filter are typically used to provide more accurate and robust state estimates (Agamennoni, 2011; Deisenroth, 2012). The choice of the appropriate filtering technique depends on the specific characteristics of the nonlinear system and the desired level of accuracy and computational complexity.

3.5 Kalman Filter Application

The Kalman filter has found successful applications in various domains due to its ability to provide accurate state estimation in the presence of noise and uncertainties. Here are some notable examples of successful applications of the Kalman filter in engineering and other fields:

1. **Navigation and GPS:** The Kalman filter is widely used in navigation systems and GPS receivers to estimate the position (Kaplan, 1996), velocity, and orientation of vehicles and mobile devices. It fuses data from multiple sensors, such as accelerometers, gyroscopes, and GPS receivers, to provide accurate and reliable real-time position tracking even in challenging environments with signal obstructions or noisy measurements.
2. **Aerospace and Aviation:** In aerospace engineering, the Kalman filter is used for various applications, including aircraft navigation, attitude determination, and flight control (Sarkka et al., 2007b). It helps stabilize aircraft during flight, estimate aircraft states for control systems, and facilitate autonomous landing and drone operations. Below we can see a photo representing the Aerospace application of Kalman Filter.

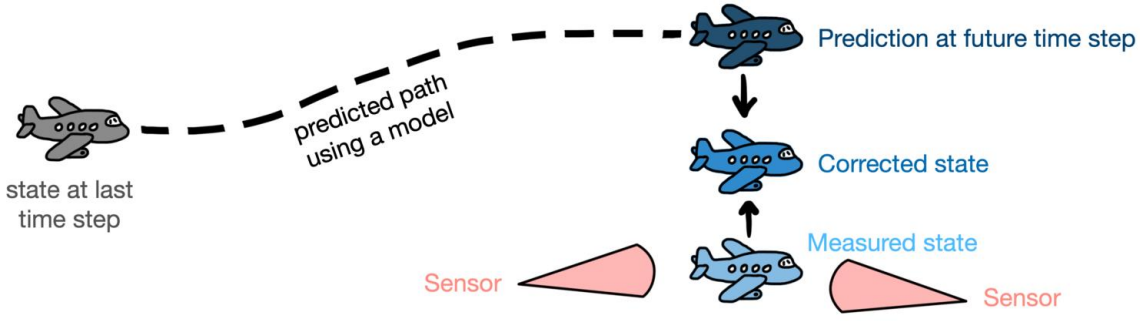


Figure 6. Kalman Filter in Aerospace application for navigation

3. **Robotics and Autonomous Systems:** In robotics, the Kalman filter plays a critical role in state estimation for robotic localization, mapping, and path planning (Bar-Shalom and Li, 1995). It enables robots to accurately estimate their position and orientation in real-time, allowing them to navigate complex environments and perform tasks autonomously.

4. **Target Tracking:** The Kalman filter is extensively used in target tracking systems, such as radar and sonar tracking (Blackman and Popoli, 1999). It provides optimal estimates of the target's position and velocity, enhancing the accuracy and efficiency of target tracking in various defense and surveillance applications.

5. **Control Systems:** The Kalman filter is widely used in control engineering for state estimation and control system design (Stone et al., 1999). It is employed in applications such as temperature control, motor control, and industrial process control, where accurate estimation of the system's state is crucial for optimal performance.

6. **Financial Markets:** In finance, the Kalman filter is used for time-series analysis, stock market forecasting, and volatility estimation. It provides a robust framework for modeling financial time series and extracting hidden patterns from noisy data.

7. **Speech and Audio Signal Processing:** The Kalman filter is applied in speech and audio processing for noise reduction, echo cancellation, and speech enhancement (Godsill and Rayner, 1998). It helps improve the quality of audio signals in noisy environments and hands-free communication systems.

8. **Image and Video Processing:** In computer vision, the Kalman filter is used for object tracking, camera motion estimation, and image stabilization (Lin et al., 2006). It helps track moving objects in video sequences and compensate for camera movements during image or video capture.

9. **Medical Imaging:** In medical imaging, the Kalman filter is used for motion compensation, cellular model predictive control, and image reconstruction (Hauk, 2004; Tarantola, 2004; Kaipio and Somersalo, 2005; Lin et al., 2006; B Smart, 2022). It helps reduce artifacts and motion-related distortions in medical images, improving the accuracy of diagnosis and treatment planning.

10. **Environmental Monitoring:** The Kalman filter is applied in environmental monitoring for estimating various environmental parameters (Murray, 1993; Keeling and Rohani, 2007), such as pollutant concentrations, temperature, and humidity. It enables accurate and timely monitoring of environmental changes and facilitates data assimilation in weather forecasting models.

11. **Power Systems:** In power systems engineering, the Kalman filter is used for state estimation in power grids. It helps estimate the state variables, such as voltage and

current, in real-time, ensuring the stability and reliability of power transmission and distribution networks.

12. **Wireless Communications:** In wireless communication systems, the Kalman filter is used for channel estimation, signal equalization, and interference cancellation (Van Trees, 1968, 1971; Proakis, 2001). It helps improve the quality of wireless communication signals and enhance the capacity of communication networks.

These successful applications highlight the versatility and effectiveness of the Kalman filter in a wide range of domains. Its ability to provide accurate and real-time state estimation in the presence of noise and uncertainties makes it a valuable tool in engineering and various other fields where accurate estimation of dynamic systems is critical for optimal performance and decision-making.

Chapter 4. Unscented Kalman Filter

4.1 Introduction to Unscented Kalman Filter

The Unscented Kalman Filter (UKF) (Julier et al., 1995; Julier and Uhlmann, 2004; Wan and Van der Merwe, 2001) stands as a sophisticated extension of the classic Kalman filter, meticulously crafted to surmount the challenge of handling nonlinear dynamic systems. This novel methodology represents a pivotal advancement that significantly broadens the applicability of the Kalman filter framework, which is intrinsically built for systems exhibiting linear characteristics. The UKF comes into play when the constraints of linearity are relinquished, ushering in a realm where complex, nonlinear systems can be effectively navigated and estimated.

Traditionally, the Kalman filter thrives in scenarios where the dynamics of a system and its observations adhere to linear relationships. However, reality often unfolds through nonlinear dynamics that elude the constraints of linearity. This is where the UKF emerges as a beacon of innovation, transcending the limitations of its predecessor. It accomplishes this by ingeniously approximating the intricate nonlinear functions inherent in the system's evolution and observations.

The UKF's approach is underpinned by the concept of sigma points, which embody a clever strategy to capture the essence of the system's true mean and covariance. These sigma points, carefully selected in a manner that holistically represents the system's statistical characteristics, serve as the linchpin of the UKF's nonlinear approximation. With these sigma points in hand, the UKF orchestrates a systematic exploration of the nonlinear functions that dictate state transitions and observations. This exploration culminates in the derivation of an updated state estimate and covariance, thereby ensuring that nonlinear dynamics are accounted for with precision and rigor.

In essence, the UKF brings a transformative capability to the Kalman filter landscape (E Chatzi, 2010; Sarkka, 2013). It equips the filter to gracefully adapt to the complexities of nonlinear systems, making it an indispensable tool for an array of real-world applications. As we delve deeper into the mechanics of the Unscented Kalman Filter, its distinct mechanisms and steps will unravel, illuminating how it breathes life into nonlinear state estimation with remarkable efficacy and versatility.

4.2 detailed outline of the Unscented Kalman Filter

4.2.1 Nonlinear System Model:

In the concept of the Unscented Kalman Filter (UKF), the "Nonlinear System Model" refers to the mathematical representation of a dynamic system that exhibits nonlinearity in its state transition and/or observation functions. The UKF is designed to handle such nonlinear systems and provide accurate state estimation in the presence of noise and uncertainties. Here we see the process in figure 7.

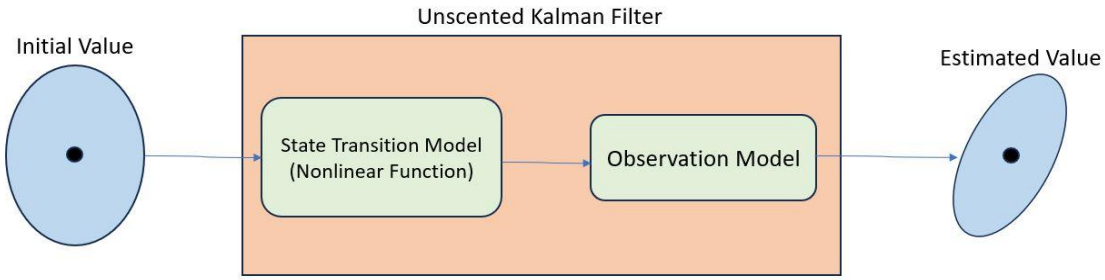


Figure 7. Unscented Kalman Filter flow summary

Let's delve into more details about the Nonlinear System Model:

1. **State Transition Model:** The state transition model represents the evolution of the system's state over time. In a discrete-time setting (commonly used in the UKF), it is defined by a nonlinear function $f(\cdot)$ that relates the current state vector $\boldsymbol{x}_k$ to the previous state vector $\boldsymbol{x}_{k-1}$ and any control input $\boldsymbol{u}_k$ at time step k. The state transition model can be expressed as:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{w}_k$$

Where:

$\boldsymbol{x}_k$ is the state vector at time step k.

$\boldsymbol{u}_k$ is the control input (if any) at time step k.

$\boldsymbol{w}_k$ represents the process noise at time step k. It is assumed to be a zero-mean Gaussian random variable with covariance matrix $\boldsymbol{Q}_k$. The process noise accounts for uncertainties and unmodeled dynamics in the system.

2. **Observation Model:** The observation model represents the relationship between the system's state $\boldsymbol{x}_k$ and the measurements or observations $\boldsymbol{z}_k$ obtained from sensors. In a discrete-time setting, the observation model is represented by a nonlinear function $h(\cdot)$ that maps the state vector to the observation space. The observation model can be expressed as:

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k$$

Where:

$\boldsymbol{z}_k$ is the measurement vector at time step k.

$\boldsymbol{v}_k$ represents the measurement noise at time step k. It is assumed to be a zero-mean Gaussian random variable with covariance matrix $\boldsymbol{R}_k$. The measurement noise accounts for inaccuracies and uncertainties in the sensor measurements.

3. **Nonlinearity in the System Model:** The key characteristic of the Nonlinear System Model is that the state transition function $f(\cdot)$ and/or the observation function $h(\cdot)$ are nonlinear. This nonlinearity can arise due to various reasons, such as physical phenomena, complex interactions, or non-linear sensor responses. Nonlinearity implies that the relationship between the system's state and its inputs or measurements cannot be represented using simple linear equations.
4. **UKF Approximation of Nonlinearity:** To address the nonlinearity in the system model, the UKF approximates the statistical properties of the state distribution after passing through the nonlinear functions without explicitly linearizing the functions. It achieves this by using a set of carefully chosen sigma points that represent the mean and covariance of the state distribution. These sigma points are propagated through the nonlinear functions to compute the predicted state estimate and covariance, which are then used in the prediction and update steps of the UKF.
5. **Handling Process and Measurement Noise:** The UKF extends the state vector and covariance matrix to incorporate the process noise $\boldsymbol{w}_k$ and measurement noise $\boldsymbol{v}_k$. This augmentation allows the inclusion of uncertainty in the propagated sigma points and enables the UKF to provide robust state estimation in the presence of noise and uncertainties.

4.2.2 Sigma Point Generation

The key idea of the UKF is to generate a set of sigma points that capture the true mean and covariance of the state. The sigma points are selected in a way that they represent the mean, and their spread approximates the covariance of the true state distribution. The number of sigma points is typically $2N + 1$, where N is the dimension of the state vector.

We can see a simple presentation of sigma points in the figure 8. in this figure as shown we have 5 sigma points; we have one in the center which is our mean point and we put the posterior point of the previous iteration as our new mean point in sigma generation and we calculate the other 4 points according to the center points and we have our Gaussian approximation circle which is formed by generating the sigma points.

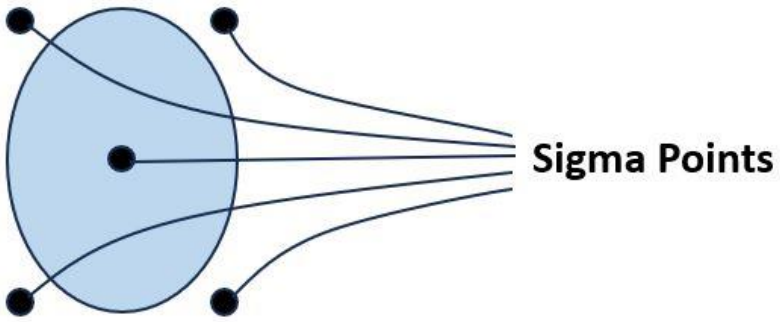


Figure 8. Sigma Points for Unscented Kalman Filter

The sigma point generation process can be outlined as follows:

1. **Determining the Number of Sigma Points:** The number of sigma points used in the UKF is typically chosen to be $2N + 1$, where N is the dimension of the state vector. Each sigma point includes the mean state vector and captures the spread of the state distribution.

2. **Calculation of Sigma Point Weights:** The weights associated with the sigma points are determined to ensure that the Unscented Kalman Filter provides unbiased and accurate state estimates. The weights are dependent on the dimension of the state vector N and a scaling factor, often denoted by λ. The choice of λ affects the spread of the sigma points.

3. **Sigma Point Computation:** To generate the sigma points, we start with the mean state vector $\boldsymbol{m}$ and the covariance matrix $\boldsymbol{P}$ of the current state distribution. For a state vector of dimension N, we compute $2N + 1$ sigma points as follows:

   a. Mean Sigma Point: The first sigma point is simply the mean state vector itself:

   $$\boldsymbol{x}_0 = \boldsymbol{m}$$

   b. Covariance Sigma Points: Next, we compute N sigma points that are dispersed along the principal directions of the state distribution, determined by the covariance matrix $\boldsymbol{P}$. These sigma points are calculated as:

   $$\boldsymbol{x}_i = \boldsymbol{m} + \left(\sqrt{(N+\lambda)\boldsymbol{P}}\right)_i \qquad for\ i = 1,2,\dots,N$$

   $$\boldsymbol{x}_{i+N} = \boldsymbol{m} - \left(\sqrt{(N+\lambda)\boldsymbol{P}}\right)_i \qquad for\ i = 1,2,\dots,N$$

Where:

$\left(\sqrt{(N+\lambda)\boldsymbol{P}}\right)_i$ denotes the i_{th} column of the square root of $(N+\lambda)\boldsymbol{P}$.

4. **Forming the Sigma Point Set:** Finally, we combine the mean sigma point and the covariance sigma points along with their adjusted weights to form the sigma point set. The set is represented as:

$$X = \{(\boldsymbol{x}_0, W_0), (\boldsymbol{x}_1, W_1), \dots, (\boldsymbol{x}_{2N}, W_{2N})\}$$

Where each element $(\boldsymbol{x}_i, W_i)$ in the set consists of a sigma point $\boldsymbol{x}_i$ and its corresponding weight W_i.

4.2.3 Weight Computation

Weight computation in the Unscented Kalman Filter (UKF) determines the significance of each sigma point and its associated measurement in the state estimation process. These weights play a crucial role in calculating the predicted state estimate, covariance, Kalman gain, and other key quantities. The weight computation step ensures that the contributions of sigma points are appropriately weighted based on their reliability and relevance.

Here's a detailed outline of the weight computation process:

1. **Define Scaling Factors:** Before computing the weights, the UKF requires two scaling factors: α and κ determines the spread of the sigma points around the mean.

$$\lambda = \alpha^2(N+\kappa) - N$$

2. **Weight for the Mean Sigma Point:** The weight associated with the mean sigma point W_0 is determined by the scaling factors α, as well as the dimensionality of the state N:

$$W_0^m = \frac{\lambda}{N+\lambda}$$

$$W_i^m = \frac{1}{2(N+\lambda)}$$

3. **Weights for Covariance Sigma Points:** The weights for the covariance sigma points Wi are calculated using the scaling factor α and ensure that the UKF provides accurate and optimal state estimates. $For\ i = 1,2, \dots, 2N$, the weights are given by:

$$W_0^c = \frac{\lambda}{N+\lambda} + (1 - \alpha^2 + \beta)$$

$$W_i^c = \frac{1}{2(N+\lambda)}$$

β is an additional algorithm parameter that can be used for incorporating prior information on the (non-Gaussian) distribution of mean.

4. **Importance of Proper Weights:** Properly computed weights ensure that the sigma points are appropriately emphasized during various steps of the UKF, such as computing the predicted state estimate, covariance, Kalman gain, and updating the state estimate based on measurements. Accurate weight computation contributes to the accuracy, stability, and convergence of the UKF algorithm.
5. **Handling Non-Gaussian Distributions:** For non-Gaussian distributions, modifications to the weight computation may be necessary to achieve optimal performance. Techniques like the Unscented Transform can be applied to adjust the weights and ensure accurate estimation in cases of non-Gaussianity.

The weights determine the influence of each sigma point in the state estimation process, and their calculation takes into account scaling factors, dimensionality, and the presence of process and measurement noise. Proper weight computation ensures that the UKF adapts effectively to the characteristics of the state distribution and provides reliable and robust state estimates even in the presence of nonlinearities and uncertainties.

4.2.4 Augmentation

Augmentation is a key step in the Unscented Kalman Filter (UKF) that involves extending the state vector and covariance matrix to incorporate the process noise and measurement noise. This step allows the UKF to account for uncertainty and noise in the propagated sigma points while handling both nonlinear state transition and observation functions.

The augmentation process can be outlined as follows:

1. **Augmentation of the State Vector:** In the standard Kalman filter, the state vector $\boldsymbol{x}_k$ only includes the primary state variables. However, in the UKF, the state vector is augmented to include the process noise $\boldsymbol{w}_k$, which represents the uncertainty in the state transition. The augmented state vector $\boldsymbol{x}_a$ is defined as:

$$\boldsymbol{x}_a = \begin{bmatrix} \boldsymbol{x_k} \\ \boldsymbol{w_k} \end{bmatrix}$$

Where:

$\boldsymbol{x}_a$ is the augmented state vector at time step *k*.

$\boldsymbol{x_k}$ is the primary state vector.

$\boldsymbol{w_k}$ is the process noise vector, assumed to be zero-mean Gaussian with covariance matrix $\boldsymbol{Q_k}$.

2. **Augmentation of the Covariance Matrix:** Similarly, the covariance matrix $\boldsymbol{P}_k$ is augmented to incorporate the covariance of the process noise. The augmented covariance matrix $\boldsymbol{P}_a$ is defined as:

$$\boldsymbol{P}_a = \begin{bmatrix} \boldsymbol{P}_k & 0 \\ 0 & \boldsymbol{Q}_k \end{bmatrix}$$

Where:

$\boldsymbol{P}_a$ is the augmented covariance matrix at time step *k*.

$\boldsymbol{P}_k$ is the covariance matrix of the primary state vector.

$\boldsymbol{Q}_k$ is the covariance matrix of the process noise.

3. **Augmentation for Measurement Noise:** In the update step of the UKF, the augmented state vector is further extended to incorporate the measurement noise $\boldsymbol{v}_k$, which represents the uncertainty in the sensor measurements. The augmented state vector including both process and measurement noise is denoted as $\boldsymbol{x}_a^m$.

4. **Propagation of Augmented Sigma Points:** With the augmented state vector $\boldsymbol{x}_a$ and augmented covariance matrix $\boldsymbol{P}_a$, the augmented sigma points are generated using the same procedure outlined in the "Sigma Point Generation" step. These augmented sigma points are then propagated through the nonlinear functions to obtain the predicted augmented state estimate and covariance.

5. **Combining Augmented Sigma Points:** The predicted augmented state estimate and covariance are used in the prediction and update steps of the UKF. During these steps, the UKF computes the mean and covariance of the predicted augmented state using the propagated augmented sigma points. The augmented sigma points and their corresponding weights are used to estimate the mean and covariance in the same way as the primary state in the traditional Kalman filter.

By including process and measurement noise in the augmented state vector and covariance matrix, the UKF provides a robust framework for accurate state estimation in the presence of noise and uncertainties, making it suitable for a wide range of real-world applications.

4.2.5 Sigma Point Propagation

Sigma point propagation in the Unscented Kalman Filter (UKF) involves passing the generated sigma points through the nonlinear state transition function to predict the new state estimates. This step is crucial for updating the state distribution while accounting for nonlinearity and uncertainty in the dynamic system. Here is a representation of sigma points propagation through state transition model in Figure 9.

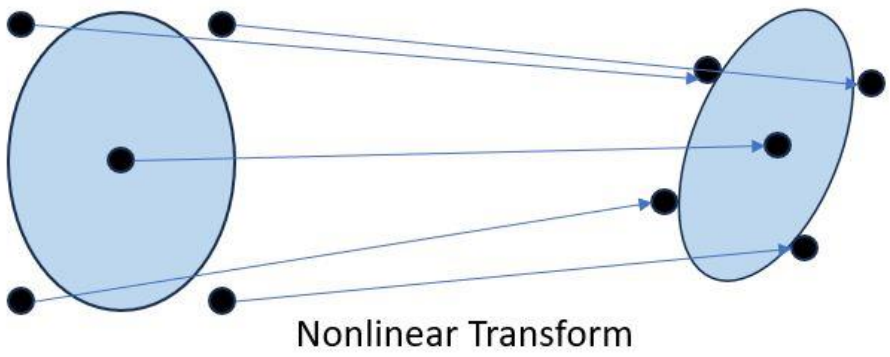


Figure 9. Sigma Points propagation through the nonlinear transform (state transition)

Here's a detailed outline of the Sigma point propagation process:

1. **Use Generated Sigma Points:** Sigma points are generated based on the current state estimate $\boldsymbol{x}_k$ and covariance matrix $\boldsymbol{P}_k$ using the procedure outlined in the "Sigma Point Generation" step. These sigma points, along with their corresponding weights, form the sigma point set X.
2. **Propagate Through Nonlinear Function:** Each sigma point in the set X is propagated through the nonlinear state transition function $f(\cdot)$ to predict the new state estimate at the next time step $k+1$. The propagation process involves applying the nonlinear function to each sigma point.
3. **Compute Predicted State Estimate and Covariance:** After propagating the sigma points through the nonlinear function, the predicted state estimates are obtained. The mean and covariance of the predicted state distribution are computed using the propagated sigma points and their associated weights. These predicted state estimates form the basis for the prediction step of the UKF.
4. **Use Propagated Sigma Points for Further Steps:** The predicted state estimate and covariance, along with the propagated sigma points and their weights, are used in

subsequent steps of the Unscented Kalman Filter, including the prediction and update steps. In the prediction step, the state estimate and covariance are used to compute the predicted mean and covariance of the state distribution.

5. **Incorporating Process Noise:** In the UKF, the process noise $\boldsymbol{w}_k$ is included in the augmented state vector. During sigma point propagation, the process noise is also propagated through the nonlinear state transition function along with the sigma points. This accounts for the uncertainty and variability introduced by the process noise during the state transition.
6. **Augmented Sigma Point Propagation:** When the augmented state vector is used (as explained in the "Augmentation" step), both the primary state and process noise components are propagated through the nonlinear function. This allows the UKF to handle both the state transition and the associated process noise in a consistent manner.
7. **Importance of Proper Nonlinear Propagation:** The accuracy of the sigma point propagation relies on the accuracy of the nonlinear state transition function $f(\cdot)$. It's crucial to ensure that the nonlinear function accurately represents the behavior of the system to obtain reliable predicted state estimates.

This step enables the UKF to capture the impact of nonlinearity and uncertainty in the dynamic system, leading to improved state estimation accuracy. By incorporating process noise and propagating augmented sigma points, the UKF provides a robust and effective framework for handling both nonlinear state transitions and noisy measurements in a wide range of real-world applications.

4.2.6 Prediction Step:

The prediction step in the Unscented Kalman Filter (UKF) involves using the propagated sigma points to compute the predicted state estimate and covariance for the next time step. This step allows the filter to forecast the evolution of the system state while accounting for nonlinearity and uncertainty.

Here's a detailed outline of the prediction step process:

1. **Propagated Sigma Points:** After the sigma point propagation through the nonlinear state transition function, we have a set of propagated sigma points X' that represent the predicted state distribution at the next time step.

2. **Computing the Predicted Mean:** The predicted mean state X_{k+1}^{-} is calculated as the weighted sum of the propagated sigma points X':

$$X_{k+1}^{-} = \sum_{i=0}^{2N} W_i * \boldsymbol{x}_i{}'$$

Where:

X_{k+1}^{-} is the predicted mean state estimate at time step $k+1$.

W_i are the weights associated with the propagated sigma points $\boldsymbol{x}_i{}'$.

3. **Computing the Predicted Covariance:** The predicted covariance matrix $\boldsymbol{P}_{k+1}^{-}$ is computed using the propagated sigma points X', the predicted mean state $\boldsymbol{x}_{k+1}^{-}$, and the process noise covariance $\boldsymbol{Q}_{k+1}$:

$$\boldsymbol{P}_{k+1}^{-} = \sum_{i=0}^{2N} W_i(\boldsymbol{x}_i' - \boldsymbol{x}_{k+1}^{-})(\boldsymbol{x}_i' - \boldsymbol{x}_{k+1}^{-})^T + \boldsymbol{Q}_{k+1}$$

Where:

$\boldsymbol{P}_{k+1}^{-}$ is the predicted covariance matrix at time step k+1.

$(\boldsymbol{x}_i' - \boldsymbol{x}_{k+1}^{-})$ represents the difference between the propagated sigma point and the predicted mean.

4. **Using Predicted Estimates:** The predicted mean state $\boldsymbol{x}_{k+1}^{-}$ and predicted covariance matrix $\boldsymbol{P}_{k+1}^{-}$ are used in the subsequent steps of the UKF. In particular, they are used in the update step to compute the Kalman gain and update the state estimate based on the measurement information.

5. **Importance of Accuracy:** The accuracy of the prediction step relies on the quality of the propagated sigma points and the proper representation of the nonlinear state transition function. If the propagated sigma points are representative of the true state distribution and the nonlinear function is accurately modeled, the predicted estimates will be reliable.

By using the propagated sigma points, the UKF computes the predicted mean state and covariance, which serve as the foundation for subsequent state updates. The prediction step ensures that the filter maintains accurate and up-to-date state estimates as it iterates through the dynamic process and assimilates new measurements.

4.2.7 Observation Sigma Points

The concept of observation sigma points is a key feature of the Unscented Kalman Filter (UKF) that enables the filter to handle nonlinear observation models. These sigma points are generated to approximate the expected measurements based on the predicted state estimate. The observation sigma points allow the UKF to incorporate nonlinearity in the measurement function and improve the accuracy of the state estimation process.

Here's a detailed outline of the observation sigma points process:

1. **Propagated Sigma Points:** After the prediction step, we have a set of propagated sigma points X' that represent the predicted state distribution at the next time step.

2. **Propagation Through Nonlinear Observation Function:** Each propagated sigma point in the set X' is passed through the nonlinear observation function $h(\cdot)$ to predict the expected measurements at the next time step:

$$\boldsymbol{z}_i' = h(\boldsymbol{x}_i') \qquad for\ i = 0,1,\dots,2N$$

Where:

$\boldsymbol{z}_i'$ is the expected measurement corresponding to the propagated state $\boldsymbol{x}_i'$.

$h(\cdot)$ is the nonlinear observation function.

3. **Computing the Predicted Measurement Mean:** The predicted measurement mean $\boldsymbol{z}_{k+1}^-$ is calculated as the weighted sum of the propagated expected measurements

$$\boldsymbol{z}_{k+1}^- = \sum_{i=0}^{2N} W_i * \boldsymbol{z}_i{}'$$

Where:

$\boldsymbol{z}_{k+1}^-$ is the predicted measurement mean at time step $k+1$.

W_i are the weights associated with the propagated sigma points $\boldsymbol{z}_i{}'$.

4. **Computing the Predicted Measurement Covariance:** The predicted measurement covariance matrix $\boldsymbol{S}_{k+1}^-$ is computed using the propagated expected measurements $\boldsymbol{z}_i{}'$, the predicted measurement mean $\boldsymbol{z}_{k+1}^-$, and the measurement noise covariance $\boldsymbol{R}_{k+1}$:

$$\boldsymbol{S}_{k+1}^- = \sum_{i=0}^{2N} W_i(\boldsymbol{z}_i' - \boldsymbol{z}_{k+1}^-)(\boldsymbol{z}_i' - \boldsymbol{z}_{k+1}^-)^T + \boldsymbol{R}_{k+1}$$

Where:

$\boldsymbol{S}_{k+1}^{-}$ is the predicted measurement covariance matrix at time step $k+1$.

$(\boldsymbol{z}_i' - \boldsymbol{z}_{k+1}^{-})$ represents the difference between the propagated expected measurement and the predicted measurement mean.

5. **Using Predicted Measurements:** The predicted measurement mean $\boldsymbol{z}_{k+1}^{-}$ and predicted measurement covariance matrix $\boldsymbol{S}_{k+1}^{-}$ are used in the update step of the UKF. They are crucial for computing the Kalman gain and updating the state estimate based on the actual measurements.

6. **Importance of Accurate Nonlinear Propagation:** The accuracy of the observation sigma point propagation relies on the quality of the nonlinear observation function $h(.)$. Ensuring that the nonlinear function accurately represents the mapping between the state space and the measurement space is essential for obtaining reliable predicted measurements.

By propagating the expected measurements through the nonlinear observation function, the UKF is able to incorporate nonlinearity in the measurement process and provide accurate predicted measurements. The predicted measurement mean and covariance are crucial components in the subsequent update step, enabling the filter to assimilate real measurements and refine the state estimate accordingly.

4.2.8 Update Step

The update step in the Unscented Kalman Filter (UKF) is where the filter assimilates real measurements to refine the state estimate. In this step, the UKF calculates the Kalman gain, updates the state estimate and covariance based on the actual measurements, and ensures that the filter adapts to new information.

Here's a detailed outline of the update step process:

1. **Propagated Sigma Points and Predicted Measurements:** After the prediction step, we have the set of propagated sigma points X' and the predicted measurement mean $\boldsymbol{z}_{k+1}^{-}$, as well as the predicted measurement covariance $\boldsymbol{S}_{k+1}^{-}$ based on the observation sigma points.

2. **Calculate Cross-Covariance Matrix:** The cross-covariance matrix $\boldsymbol{C}_{xz}$ measures the correlation between the state deviations from the predicted state estimate and the measurement deviations from the predicted measurement mean. It is computed as:

$$\boldsymbol{C}_{xz} = \sum_{i=0}^{2N} W_i (\boldsymbol{x}_i' - \boldsymbol{x}_{k+1}^{-})(\boldsymbol{z}_i' - \boldsymbol{z}_{k+1}^{-})^T$$

Where:

$\boldsymbol{C}_{xz}$ is the cross-covariance matrix.

$(\boldsymbol{x}_i' - \boldsymbol{x}_{k+1}^-)$ represents the difference between the propagated sigma point and the predicted state estimate.

$(\boldsymbol{z}_i' - \boldsymbol{z}_{k+1}^-)$ represents the difference between the propagated expected measurement and the predicted measurement mean.

3. **Calculate Kalman Gain:** The Kalman gain $\boldsymbol{K}$ determines how much weight to give to the predicted state estimate and the measurements when updating the state estimate. It is computed as:

$$\boldsymbol{K} = \boldsymbol{C}_{xz} * \boldsymbol{S}_{k+1}^{-1}$$

Where:

$\boldsymbol{K}$ is the Kalman gain.

$\boldsymbol{S}_{k+1}^{-1}$ is the inverse of the predicted measurement covariance matrix.

4. **Update State Estimate:** The state estimate is updated using the Kalman gain and the measurement deviation from the actual measurement $\boldsymbol{z}_{k+1}$:

$$\boldsymbol{x}_{k+1}^{+} = \boldsymbol{x}_{k+1}^{-} + \boldsymbol{K} * (\boldsymbol{z}_{k+1} - \boldsymbol{z}_{k+1}^{-})$$

Where:

$\boldsymbol{x}_{k+1}^{+}$ is the updated state estimate.

$\boldsymbol{z}_{k+1}$ is the actual measurement at time step $k+1$.

5. **Update Covariance Matrix:** The updated covariance matrix $\boldsymbol{P}_{k+1}^{+}$ is computed to reflect the new state estimate considering the measurement information:

$$\boldsymbol{P}_{k+1}^{+} = \boldsymbol{P}_{k+1}^{-} - \boldsymbol{K} * \boldsymbol{S}_{k+1} * \boldsymbol{K}^T$$

Where:

$\boldsymbol{P}_{k+1}^{+}$ is the updated covariance matrix.

$\boldsymbol{K}$ is the Kalman gain.

$\boldsymbol{S}_{k+1}$ is the predicted measurement covariance matrix.

6. **Finalizing the Update Step:** The updated state estimate $\boldsymbol{x}_{k+1}^{+}$ and covariance matrix $\boldsymbol{P}_{k+1}^{+}$ are used as the new estimates for the current state at time step $k+1$. The filter iterates through this process for each new measurement, adapting and refining the state estimates as more information becomes available.

By calculating the Kalman gain, updating the state estimate, and adjusting the covariance matrix, the UKF ensures that the state estimate aligns with the observed measurements while considering the uncertainties associated with both the state and the measurements. This iterative process enables the UKF to provide accurate and robust state estimates even in the presence of noise, nonlinearity, and uncertainty.

4.2.10 Repeat

The "Repeat" step in the Unscented Kalman Filter (UKF) refers to the iterative process that the filter follows to estimate the state of a dynamic system. The UKF iterates through prediction, update, and weight computation steps to continuously refine the state estimate as new measurements become available. This iterative process ensures that the filter adapts to changing conditions and provides accurate state estimates even in the presence of nonlinearity, uncertainty, and noise.

Here's a detailed outline of the iterative process:

1. **Initialization:** The UKF begins with an initial estimate of the state vector $\boldsymbol{x}_0$ and its associated covariance matrix $\boldsymbol{P}_0$. These initial estimates can come from prior knowledge or initial measurements.
2. **Prediction Step:** In this step, the UKF propagates the current state estimate $\boldsymbol{x}_k$ through the state transition function using sigma points. This results in the predicted state estimate $\boldsymbol{x}_{k+1}^-$ and the predicted covariance matrix $\boldsymbol{P}_{k+1}^-$.
3. **Weight Computation:** The weights for the sigma points are computed based on the scaling factors α and β, the dimensionality of the state N, and additional factors such as κ for augmented sigma points. These weights determine the significance of each sigma point in the subsequent steps.
4. **Observation Sigma Points:** For nonlinear observation models, the UKF generates observation sigma points based on the predicted state estimate. These observation sigma points approximate the expected measurements and facilitate handling nonlinearities in the measurement function.
5. **Update Step:** In this step, the UKF assimilates real measurements by calculating the Kalman gain and updating the state estimate and covariance based on the actual measurements. The updated state estimate $\boldsymbol{x}_{k+1}^+$ and covariance matrix $\boldsymbol{P}_{k+1}^+$ reflect the integration of measurement information.
6. **Iteration and Convergence:** The updated state estimate $\boldsymbol{x}_{k+1}^+$ serves as the current state estimate for the next iteration. The UKF iterates through the prediction, weight computation, observation sigma points, and update steps for each subsequent time step. The iterative process continues as new measurements are acquired, and the state estimate refines with each iteration.

7. **Handling Multiple Measurements:** The UKF is capable of handling multiple measurements at each time step. For each measurement, the filter performs the observation sigma points, update, and weight computation steps before proceeding to the next time step's prediction.
8. **Convergence and Stability:** The iterative process aims to converge to an accurate state estimate as the filter iterates through prediction and update steps. Proper initialization, accurate modeling of nonlinear functions, appropriate weight computation, and suitable tuning of parameters contribute to the stability and convergence of the UKF.

By iteratively refining the state estimate through prediction, observation, update, and weight computation steps, the UKF adapts to changing conditions, handles nonlinearity and uncertainty, and provides accurate state estimates even in complex and noisy environments. This iterative process is the core mechanism that makes the UKF a powerful tool for state estimation in a wide range of applications.

4.3 Unscented Kalman Filter (UKF) vs. Standard Kalman Filter

Strengths of the Unscented Kalman Filter (UKF) Compared to the Standard Kalman Filter:

1. **Nonlinearity Handling**: The UKF excels in handling nonlinear state transition and observation functions, which the standard Kalman filter struggles with. The UKF uses sigma point approximation to capture the nonlinearities, making it suitable for a wide range of real-world applications.
2. **Accurate Approximations**: By using a set of carefully chosen sigma points, the UKF provides more accurate approximations of the mean and covariance of the state distribution compared to linearization methods used in the standard Kalman filter.
3. **No Derivative Computation**: The UKF does not require the computation of derivatives (Jacobian matrices) of nonlinear functions, as required in the standard Kalman filter. This saves computational resources and simplifies implementation, especially for complex models.
4. **Flexibility**: The UKF is versatile and can be applied to systems with complex dynamics and observations without the need for explicit linearization. This flexibility is particularly useful when dealing with systems that exhibit rapidly changing nonlinear behavior.
5. **Handling of Gaussian and Non-Gaussian Distributions**: The UKF performs well for both Gaussian and moderately non-Gaussian distributions, whereas the standard

Kalman filter assumes Gaussian distributions. This allows the UKF to provide accurate estimates even in cases of mild non-Gaussianity.

Weaknesses of the Unscented Kalman Filter Compared to the Standard Kalman Filter:

1. **Computational Complexity**: The UKF requires more computational resources compared to the standard Kalman filter due to the need for sigma point generation, propagation, and weight computations. This can be a concern for real-time applications or systems with high-dimensional state spaces.
2. **Limited Handling of Highly Non-Gaussian Distributions:** While the UKF can handle moderately non-Gaussian distributions, it may struggle with highly non-Gaussian distributions. In such cases, more advanced techniques like particle filters might be more appropriate.
3. **Parameter Tuning:** The UKF involves tuning parameters such as scaling factors (α, β, κ) to achieve optimal performance. Selecting appropriate values for these parameters can be challenging and may require some trial and error.
4. **Dependency on Sampling Techniques**: The quality of the UKF estimates relies on the accuracy of the sigma point sampling. Poorly chosen sigma points can lead to inaccurate estimates and potential divergence of the filter.
5. **Limited Understanding of Uncertainty Propagation:** While the UKF provides a good approximation of the mean and covariance of the state distribution, it may not fully capture the propagation of uncertainty through complex nonlinear functions as accurately as particle filters, especially in scenarios with large uncertainties.

In summary, the Unscented Kalman Filter addresses many of the limitations of the standard Kalman filter by efficiently handling nonlinearity without requiring the computation of derivatives. However, it comes with its own set of challenges, including computational complexity and parameter tuning. The choice between the two filters depends on the specific characteristics of the system, the available computational resources, and the desired level of accuracy and robustness in state estimation.

Chapter 5. Particle Filter

5.1 Introduction to Particle Filter (Sequential Monte Carlo Method):

In the realm of state estimation and filtering, the Particle Filter, also known as the Sequential Monte Carlo (SMC) (Liu, 2001) method, stands as a remarkable breakthrough that transcends the constraints of traditional linear and Gaussian assumptions. This innovative technique has emerged as a potent solution for tackling the challenges posed by nonlinear dynamics, non-Gaussian distributions, and high-dimensional state spaces in various fields, including robotics, signal processing, finance, and more. Its roots can be traced back to the 1990s when researchers sought to extend the capabilities of traditional filters, like the Kalman filter, which are limited by their assumptions of linearity and Gaussianity (Gilks et al., 1996; Liu, 2001; Brooks et al., 2011).

The Essence of Particle Filtering:

At its core, the Particle Filter embraces the notion of non-parametric Bayesian filtering. This implies that rather than relying on closed-form equations or moment-based representations of the state distribution, the Particle Filter takes an empirical stance (Daum and Huang, 2003; Snyder et al., 2008). It harnesses the power of particles—discrete samples that embody hypothetical states—to approximate the elusive posterior distribution of the true system state. By sequentially updating and propagating these particles, the Particle Filter endeavors to capture the dynamic evolution of the state as new measurements become available.

Unraveling Complex Dynamics and Non-Gaussianity

The Particle Filter, a cornerstone of state estimation, introduces a groundbreaking approach that defies the limitations imposed by linear and Gaussian assumptions. This method, also known as the Sequential Monte Carlo (SMC) technique, has revolutionized state estimation in diverse fields by embracing the intrinsic complexity and non-Gaussian nature of real-world systems.

Conceptual Essence:

At the heart of the Particle Filter method lies a profound shift in perspective. Instead of confining the state distribution to conform to a specific functional form, the Particle Filter ventures into the realm of empirical representation. It navigates through the intricacies of nonlinear dynamics and multifaceted uncertainties by harnessing particles—discrete hypothetical states—as its currency.

Dynamic Hypotheses:

Particles are not just numerical entities; they embody dynamic hypotheses about the true state of a system. These particles span the state space, embodying a spectrum of potential states at a given time step. Unlike parametric filters that demand specific distributional assumptions, Particle Filters capitalize on the richness of particles to accommodate the diversity, nonlinearity, and even multimodality present in real-world systems.

Flexibility and Adaptability:

A hallmark of the Particle Filter is its adaptability. While traditional filters might falter in the face of nonlinearities, the Particle Filter thrives. It elegantly sidesteps the need for derivatives or Jacobian matrices, gracefully accommodating systems where mathematical tractability is elusive. This adaptability extends to a wide array of applications—robotics, finance, ecology—where complex behaviors and diverse distributions reign supreme.

Capturing Uncertainty:

The Particle Filter method doesn't shy away from uncertainty; it thrives on it. In situations where Gaussianity is but a distant dream, this technique emerges as a guiding light. Particles, equipped with weights that reflect their fidelity to observations, collectively create an ensemble that spans the spectrum of uncertainty, from low-probability outliers to high-probability modes.

Sequential Evolution:

The Particle Filter method is not a one-time wonder. It operates sequentially, mirroring the real-world progression of dynamic systems (Doucet et al., 2001). It ingeniously blends prediction, data assimilation, and state estimation in an iterative dance. With each step, particles evolve, predictions refine, and estimations converge towards the elusive truth of the state.

Beyond Linearity and Gaussianity:

Perhaps the most transformative aspect of the Particle Filter method is its liberation from linear and Gaussian assumptions. It embraces the inherent complexity of reality, acknowledging that systems rarely adhere to the simplicity of linear behavior or Gaussian distributions. This adaptability equips the Particle Filter to unveil the mysteries of chaotic systems, understand intricate biological processes, and navigate the tumultuous waters of financial markets.

Ongoing Frontiers:

While the Particle Filter method has undoubtedly elevated state estimation, it's not devoid of challenges. Sample degeneracy, computational load, and the quest for efficient resampling mechanisms remain active areas of research. As technology advances, hybrid methods that blend Particle Filters with other estimation techniques are on the horizon, promising enhanced accuracy and efficiency.

In essence, the Particle Filter method is a beacon of innovation in the realm of state estimation. It challenges the status quo, redefining how we unravel the mysteries of dynamic systems. By embracing complexity, nonlinearity, and non-Gaussianity, the Particle Filter has ushered in a new era where state estimation is not just about equations, but about capturing the essence of reality itself.

5.2 The Fundamental Steps:

5.2.1 Particle Initialization:

The Particle Filter commences its journey with the selection of an initial set of particles. These particles span the state space and are endowed with corresponding weights to reflect their significance in the estimation process.

At the heart of the Particle Filter's initiation lies the critical step of "Particle Initialization." This pivotal phase serves as the starting point for the entire estimation process, where an ensemble of particles is meticulously crafted to encapsulate the system's possible states. Let's delve into the intricacies of this step and the principles that guide it.

Conceptual Basis:

Particle initialization is akin to casting a wide net into the sea of possible states. It involves generating an ensemble of particles, each representing a speculative state of the system at the given time step. These particles collectively form an initial approximation of the true state distribution, setting the stage for subsequent steps in the Particle Filter process.

Key Elements:

1. Particle Generation: The initial ensemble of particles is drawn from an initial distribution $p(x_0)$, where x_0 represents the system's state at the initial time step (Punskaya et al, 2002). The distribution should be chosen to encompass the range of potential states and capture the inherent uncertainties.

2. Number of Particles: The size of the particle ensemble, denoted as N, is a critical parameter. It defines how finely the state space is sampled. Larger N provides a more accurate representation but escalates computational demands.
3. Weight Assignment: Alongside generating particles, each particle is endowed with an initial weight w_{0_i}, reflecting its relative importance in the ensemble (Liu and Chen, 1995). Initially, these weights are often uniform, signifying equal belief in each particle's validity.

Strategies and Considerations:

1. Prior Knowledge: Incorporating any available prior information about the system's state distribution can influence the selection of $p(x_0)$. This might involve utilizing historical data, sensor measurements, or expert insights.
2. Coverage and Spread: The choice of $p(x_0)$ should ensure that particles span the entirety of the state space, adequately capturing regions where the true state is likely to reside.

Here we can see a representation of particles and comparison with gaussian distribution we had previously in UKF (sarkka, 2013).

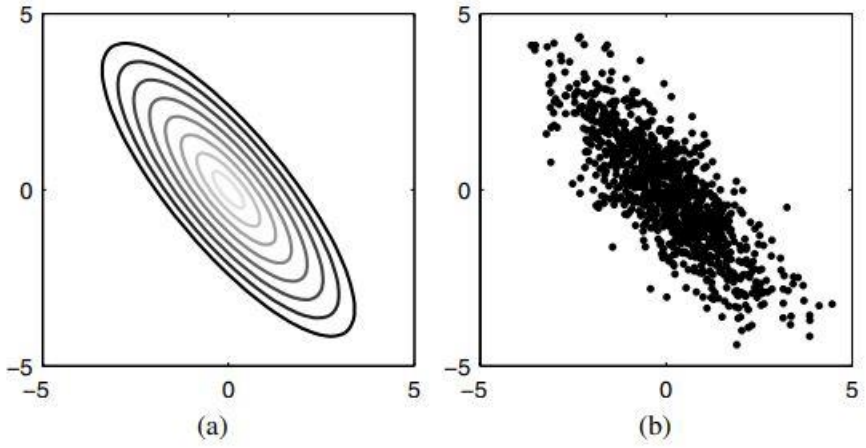


Figure 10. (a) Two-dimensional Gaussian distribution. (b) Monte Carlo representation of the same Gaussian distribution

Mathematical Representation:

Mathematically, the particle initialization can be described as follows:

1. Generate N particles:

$$x_{0_i}, \quad i = 1, \ldots, N \qquad \text{from the initial distribution } p(x_0)$$

2. Assign equal initial weights:

$$w_{0_i} = N_1 \quad for\ all\ i = 1, \ldots, N$$

The quality of the initial particle ensemble profoundly influences the Particle Filter's effectiveness. Well-chosen particles and their corresponding weights lay the groundwork for accurate state estimation. An adept initialization facilitates smoother transitions through subsequent steps like prediction, weighting, and resampling.

The "Particle Initialization" step of the Particle Filter method is not just the opening act; it's the canvas on which the intricate masterpiece of state estimation takes form. By generating particles, assigning initial weights, and framing the initial distribution, this step lays the cornerstone for a journey that traverses' uncertainties, nonlinearities, and the complex dynamics of real-world systems. It transforms the abstract idea of estimation into a tangible process, setting the stage for the Particle Filter's dynamic evolution.

5.2.2 Prediction:

The particles are then subjected to the system's nonlinear dynamics, propelled forward by the state transition model while accommodating the introduction of process noise. This prediction step paves the way for projecting the particles into the future.

Prediction in the Particle Filter: Navigating Nonlinear Dynamics and Uncertainties

The second step of the Particle Filter, known as "Prediction," ushers the ensemble of particles into the future by subjecting them to the system's nonlinear dynamics. This pivotal phase sets the stage for estimating how the system's state evolves over time, incorporating both the inherent behavior of the system and the uncertainties introduced by process noise.

Crucial Conceptual Foundation:

In the prediction step, each particle ventures forth as a potential trajectory of the system's evolution. The nonlinear dynamics of the system guide the particles as they project forward in time, while process noise introduces the vital element of uncertainty. This process lays the groundwork for assessing how the particles' speculative states evolve over the prediction interval.

Key Elements:

1. Nonlinear Dynamics: The system's behavior is often nonlinear, meaning its evolution cannot be adequately captured by linear equations. In the prediction step, particles are

propelled forward according to the nonlinear state transition model, capturing the intricate interplay of the system's variables.

2. Process Noise: The real world is rife with uncertainties and perturbations that can influence a system's dynamics. Process noise accounts for these uncertainties, ensuring that the predicted states of the particles encompass a range of possible outcomes.

Mathematical Representation:

The prediction step can be formalized as follows:

1. For each particle $i = 1, \ldots, N$:

- Propagate the particle using the nonlinear state transition model:

$$x_{t+1}^{i} = f(x_t^i) + \epsilon_t^i$$

Here:

$f(\cdot)$ represents the nonlinear state transition function

x_t^i is the particle's state at time *t*, and

ϵ_t^i is the process noise.

Process Noise Incorporation:

Process noise is essential for capturing the system's inherent uncertainty and deviations from the model's deterministic predictions. It can be drawn from a known distribution or modeled as a Gaussian process with zero mean and covariance matrix Q_t^i.

The prediction step not only advances the particles in time but also introduces the vital element of uncertainty. By integrating process noise, the Particle Filter method acknowledges that real-world dynamics are subject to deviations and perturbations that need to be accounted for in the estimation process.

The "Prediction" step in the Particle Filter is a crucial endeavor in estimating how a system evolves over time, while accommodating its nonlinear behavior and inherent uncertainties (sarkka, 2013). By propelling particles into the future and embracing process noise, this step allows the Particle Filter to navigate the intricate dynamics of the real world. It sets the particles on a trajectory that mirrors the complex evolution of the system, providing the foundation for subsequent steps like weighting, resampling, and estimation.

5.2.3 Weighting and Measurement Incorporation

As measurements are acquired, the particles' compatibility with these observations is assessed. Weights are assigned to each particle based on their likelihood with respect to the measurements, capturing the consistency between particle-generated predictions and real-world data.

Weighting and Measurement Incorporation in the Particle Filter: The Marriage of Model and Observation

In the dynamic dance of state estimation through the Particle Filter, the "Weighting and Measurement Incorporation" step holds a pivotal role, forging a profound connection between theoretical predictions and real-world measurements. This phase is where the particles' virtual trajectories intersect with the tangible reality of observed data, and the tapestry of estimation takes its first concrete shape.

Imagine a convergence point where each particle's envisioned path crosses paths with the empirical data gathered from the system. This is the core of "Weighting and Measurement Incorporation." Each particle is assessed for its alignment with the observed measurements, and its significance in the ensemble is dynamically adjusted based on its ability to harmonize with the empirical truth.

Crucial Elements:

1. Measurement Likelihood (Liu, 2001): This is the cornerstone of the step. It quantifies the probability of a particle generating a specific outcome given the measurements acquired. This likelihood is a measure of how well the particle's projected state resonates with the real-world observations.
2. Weight Assignment: With the measurement likelihood as the lodestar, each particle is assigned a weight proportional to its compatibility with the observed data. Particles generating outcomes that closely match the measurements receive higher weights, making their contribution more significant in the estimation process.

Mathematical Details:

1. Measurement Likelihood (Conditional Probability): For each particle $i = 1, \dots, N$, the measurement likelihood is calculated as:

$$p(y_t \mid x_t^i)$$

Here:

y_t represents the actual measurement at time t, and

x_t^i is the predicted state of particle i at the same time step.

2. Weight Assignment: The weight assigned to each particle i is proportional to its measurement likelihood: $w_t^i \propto p(y_t \mid x_t^i)$ to ensure that the weights sum up to unity across all particles, normalization is performed:

$$w_t^i = \frac{p(y_t \mid x_t^i)}{\sum_{j=1}^{N} p(y_t \mid x_t^j)}$$

This step signifies a balance between model and observation. Particles that conform closely to the measurements gain more weight, symbolizing their harmony with real-world evidence. Conversely, particles that diverge significantly receive lower weights, representing their incompatibility with empirical data. Consequently, this process propels the ensemble towards the true state distribution, aligning the particles' trajectories with observed reality.

"Weighting and Measurement Incorporation" represents a pivotal crossroads where hypotheses meet evidence, predictions intersect with measurements, and theoretical trajectories find resonance with empirical truth. It is where Bayesian inference manifests in its purest form, molding particle significance based on their ability to reflect the real world. This phase is the bridge that unites the abstract concept of state estimation with the tangible realm of data, illuminating the way forward in the Particle Filter's journey to uncover the hidden state of the system.

5.2.4 Sequential Importance Resampling

Importance Sampling:

In practical Bayesian models, it's often hard to directly get samples from the probability distribution $p(x \mid y_{1:T})$ because it's a complicated function (sarkka, 2013; Liu, 2001). Instead, we use a simpler distribution called the "importance distribution" $\pi(x \mid y_{1:T})$ to get samples easily. This method is called importance sampling.

Importance sampling works by breaking down the expectation of the posterior probability density into smaller parts:

$$p(x \mid y_{1:T}) : \int g(x)\, p(x \mid y_{1:T})\, dx = \int \left[g(x) \frac{p(x \mid y_{1:T})}{\pi(x \mid y_{1:T})} \right] \pi(x \mid y_{1:T})\, dx$$

Where $g(x)$ is an arbitrary function and the importance distribution $\pi(x \mid y_{1:T})$ should cover at least the same area as the actual distribution $p(x \mid y_{1:T})$. We approximate this expectation using a technique called Monte Carlo (sarkka, 2013; Liu, 2001), where we draw N samples from the importance distribution and calculate an approximation as shown below:

$$E[g(x) \mid y_{1:T}] \approx \frac{1}{N}\sum_{i=1}^{N}\frac{p\left(x^{(i)}|y_{1:T}\right)}{\pi(x^{(i)}|y_{1:T})}\, g\left(x^{(i)}\right) = \sum_{i=1}^{N}\tilde{\omega}^{(i)}\, g(x^{(i)})$$

The weights for these samples are defined in as below:

$$\tilde{\omega}^{(i)} = \frac{1}{N}\,\frac{p\left(x^{(i)}|y_{1:T}\right)}{\pi(x^{(i)}|y_{1:T})}$$

However, a drawback of direct importance sampling is that we need to evaluate $p\left(x^{(i)}|y_{1:T}\right)$ directly, which can be tricky. To solve this issue, we approximate the expectation integral by also approximating the normalization constant, which is often hard to compute.

Sequential Importance Sampling:

Sequential Importance Sampling (SIS) (Doucet et al., 2001) is like a Sequential version of importance sampling. It's a method we use to estimate the filtering distributions in state space models.

$$x_k \sim p(x_k \mid x_{k-1})$$

$$y_k \sim p(y_k \mid x_k)$$

These models involve states $x_k \in R^n$ and measurements $y_k \in R^m$, which can have both discrete and continuous parts.

In SIS, we work with a weighted set of particles, $\left\{\left(w_k^{(i)}, x_k^{(i)}\right) : i = 1, \ldots\ldots, N\right\}$ which are essentially samples from a distribution, along with their associated weights. This set helps us represent the filtering distribution $p(x_k \mid y_{1:k})$ at each time step k. We can use it to estimate the expected value of any function $g(.)$ as an average of these weighted samples, as shown here:

$$E[g(x_k) \mid y_{1:k}] \approx \sum_{i=1}^{N} w_k^{(i)}\, g(x_k^{(i)})$$

Alternatively, you can think of SIS as creating an approximation to the filtering distribution, as:

$$p(x_k \mid y_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta\left(x_k - x_k^{(i)}\right)$$

We can show the process of sequential importance sampling as figure below:

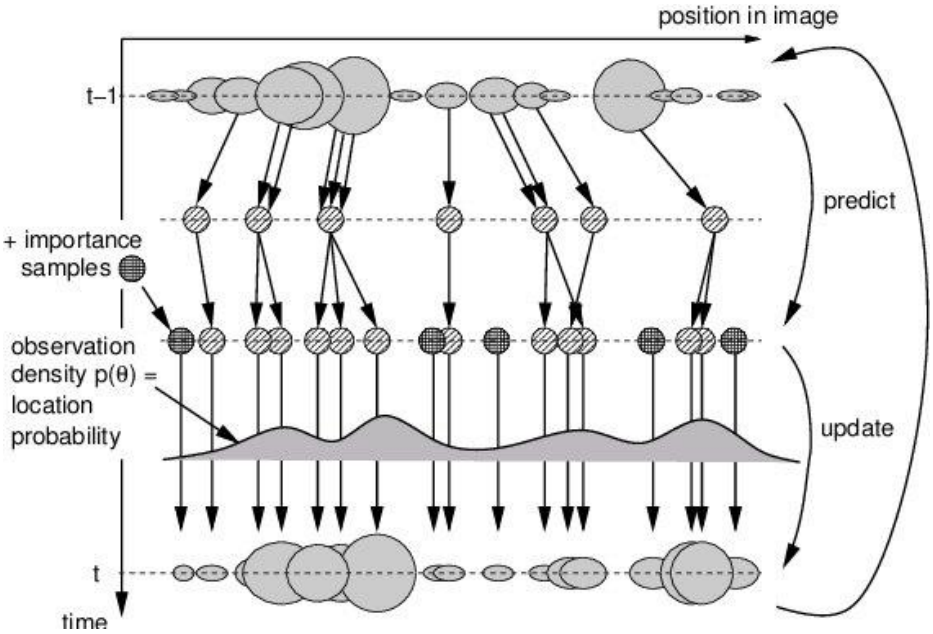


Figure 11. Importance Sampling process

Sequential Importance Resampling:

The issue with the SIS algorithm mentioned earlier is that it often leads to a situation where nearly all the particles end up with extremely low or zero weights. This problem is known as "degeneracy" in particle filtering, and it used to make particle filters not very practical for real-world applications for a long time.

To fix this degeneracy problem, we use a resampling process (Gordon et al., 1993; Kitagawa, 1996; Doucet et al., 2001; Ristic et al., 2004). This means we draw N new samples based on the weights we have and replace the old set of N samples with this fresh set.

The Particle Filter strives to maintain diversity among particles, ensuring representation across different regions of the state space. Resampling serves this purpose, granting particles with higher weights a higher chance of reproduction in the next iteration. This mechanism counteracts particle degeneracy and guarantees a balanced representation.

Resampling in the Particle Filter: Fostering Diversity and Enabling Balance

The "Resampling" step in the Particle Filter is a strategic maneuver executed to nurture diversity within the particle ensemble and to achieve a balanced representation of potential states. This step becomes increasingly crucial as the Particle Filter iterates,

preventing the ensemble from becoming dominated by a select few particles and rejuvenating underrepresented yet plausible hypotheses.

The primary goal of resampling is twofold: to breathe life into the ensemble by amplifying the voices of particles with higher weights (indicating their alignment with measurements) and to reintroduce particles embodying diverse predictions about the true state. This step mitigates issues arising from the degeneration of weights, ensuring that each particle has a meaningful impact on the evolving estimation process.

Key Elements of Resampling:

1. Importance of Diversity: As the estimation process unfolds, some particles accumulate higher weights due to their alignment with the observed measurements. While these particles contribute significantly, they can dominate the ensemble, potentially sidelining valid but underrepresented hypotheses. Resampling steps in to rekindle diversity.
2. Weighted Selection: Resampling isn't a random lottery; it's a weighted process. Particles with higher weights are more likely to be selected multiple times, while particles with lower weights have a reduced chance of continuation. This approach ensures that particles generating more plausible outcomes retain their influence.

Mathematical Mechanics:

The process of resampling can be outlined as follows:

1. Compute the cumulative weights:

$$W_t^i = \sum_{j=1}^{i} w_t^j$$

2. Generate N random samples u_i from a uniform distribution over the interval $[0, \frac{1}{N}]$.
3. Initialize indices i and j to 1.
4. For each of the N particles $k = 1, \ldots, N$:

- While u_k is less than W_t^i, increment i by 1.
- Assign particle k the properties (state, weight) of particle i.

The process of Sequential Importance Resampling is shown as figure below:

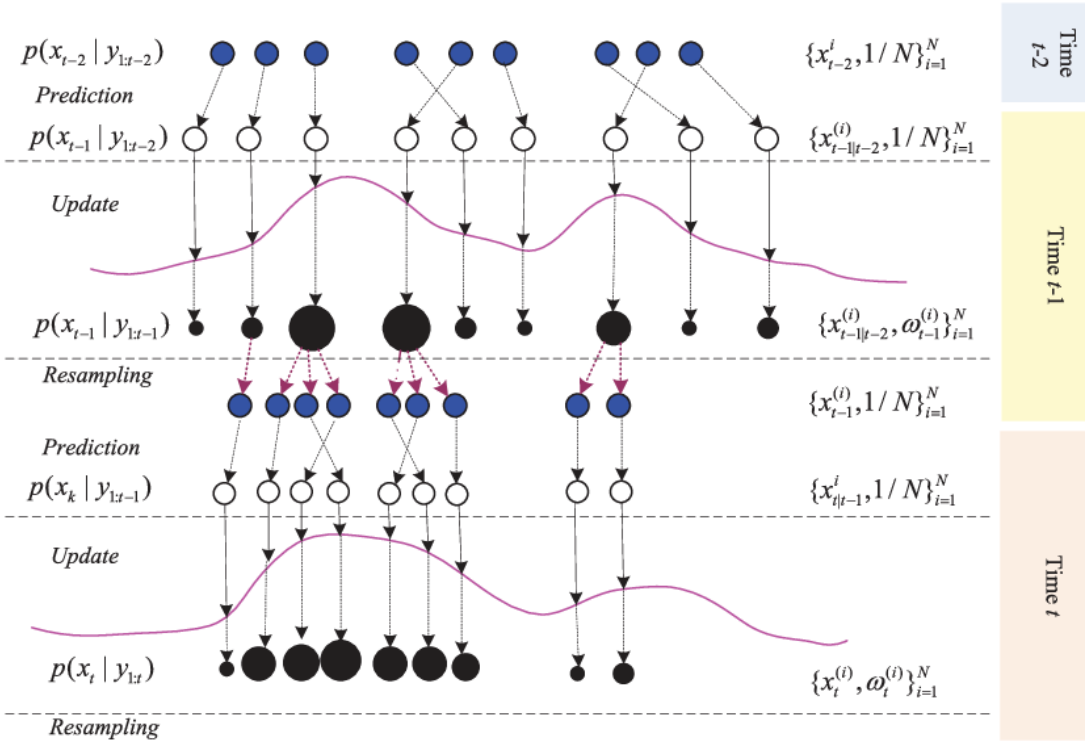


Figure 12. Sequential Importance Resampling flow process

Effective Number:

The concept of the "effective number of particles" or "effective sample size" is often used in resampling. It reflects the ensemble's ability to represent the underlying distribution. Calculating the effective number helps gauge whether resampling is necessary. It is computed as:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} \left(w_t^i\right)^2}$$

When N_{eff} is lower than a certain threshold, it signals the need for resampling to rejuvenate diversity.

Types of Resampling:

1. Systematic Resampling: Particles are organized into segments based on their cumulative weights. A single random draw determines the starting point, followed by equidistant sampling within each segment.
2. Residual Resampling: After determining whole particles to be resampled, the residual fraction decides which particles within a segment are replicated.

3. Stratified Resampling: Particles are partitioned into segments, and within each segment, random draws are performed. This approach ensures a more uniform distribution of resampled particles.

Significance and Challenges:

1. Diversity and Balance: Resampling safeguards diversity and balances the ensemble, reinvigorating underrepresented hypotheses.
2. Mitigating Degeneracy: It addresses the phenomenon of weight degeneracy, where only a few particles contribute significantly to the estimation.
3. Randomness vs. Determinism: Resampling introduces randomness, which might be challenging to reconcile in deterministic algorithms.

"Resampling" in the Particle Filter acts as a compass that steers the ensemble away from dominance, leading it towards a comprehensive representation of possible states. It's the Particle Filter's mechanism for infusing the ensemble with fresh energy, nurturing diversity, and ensuring that all valid regions of the state space receive fair consideration.

5.2.5 Estimation

The culmination of the Particle Filter process entails combining the resampled particles to formulate the estimated state. The final estimate is a weighted average of these particles, accounting for both their contribution and their significance, as inferred from their assigned weights.

Estimation in the Particle Filter: Uniting Particles into a Coherent Whole

The fifth step of the Particle Filter, aptly named "Estimation" marks the grand finale of the estimation journey. It's the moment where the diverse and resampled particles harmoniously converge to yield a weighted collective that paints an accurate portrait of the system's true state. This is the step where speculation meets evidence, and the ensemble's insights crystalize into a coherent estimation.

The essence of the "Estimation" step lies in synthesizing the knowledge encapsulated within the particle ensemble into a single, representative estimate of the system's state. Each particle contributes to this synthesis based on its assigned weight, reflecting its alignment with observed measurements. The result is a weighted aggregation that aptly captures the nuances of the true state.

Key Elements of Estimation:

1. Resampled Ensemble: After resampling, the ensemble becomes a rejuvenated assembly of particles, each carrying a unique interpretation of the true state. These particles span the range of plausible states, from most likely to less probable, and thus represent the full spectrum of potential outcomes.
2. Weighted Average: The estimation is formed as a weighted average of the particles' states. The weight assigned to each particle signifies its reliability in the estimation process, reflecting its proximity to observed measurements.

Mathematical Mechanics:

The "Estimation" step can be described mathematically as follows:

1. For each particle $i = 1, \ldots, N$:

- Multiply the particle's state x_t^i by its weight w_t^i.

2. Compute the estimated state x^t as the sum of the weighted states:

$$x^t = \sum_{i=1}^{N} w_t^i * x_t^i$$

Implications and Significance:

1. Weighted Aggregation: The estimation reflects a collective wisdom, where each particle's perspective contributes based on its alignment with measurements. This way, the ensemble's insights are weighted by credibility.
2. Adaptive Response: The Particle Filter dynamically adjusts the estimation based on the ensemble's evolving understanding, as particles gain or lose significance through the estimation journey.
3. Incorporation of Uncertainty: The ensemble's diversity and the incorporation of measurement noise within particles acknowledge the inherent uncertainty in estimating the true state.

"Estimation" in the Particle Filter is akin to weaving together a tapestry of particles, each thread contributing to the larger fabric of knowledge. As the particles unite, their individual strengths are harnessed, leading to an estimate that resonates with the observed data. This phase transforms the Particle Filter's collective speculation into a tangible conclusion, where the inherent uncertainties of the system are accommodated, and the true state emerges from a symphony of weighted insights.

5.2.6 Iterative Evolution

Armed with the newly refined state estimate, the Particle Filter embarks on a recurring journey, iterating through the prediction, weighting, resampling, and estimation steps as new measurements arrive, progressively refining its understanding of the system's underlying state.

iterative Evolution in the Particle Filter: Unveiling the True State through Iteration

The sixth step of the Particle Filter, "Iterative Evolution," marks the inception of a dynamic journey. With each iteration, the Particle Filter unveils a progressively refined understanding of the system's true state, embodying a cyclical process that adapts and updates its estimation as new measurements pour in. This iterative dance paves the way for an ever-improving approximation of reality.

Imagine the Particle Filter as a relentless seeker of truth, continuously recalibrating its estimation based on the evolving interplay between model predictions and empirical observations. The essence of "Iterative Evolution" lies in its recursive nature—a perpetual sequence of prediction, weighting, resampling, and estimation steps that mirrors the system's actual evolution.

Key Elements of Iterative Evolution:

1. Refinement of Estimation: Each iteration hones the Particle Filter's estimation. As new measurements arrive, the ensemble adapts, embracing particles that resonate with the observed data while shedding those that diverge.
2. Dynamic Response: The Particle Filter's response is dynamic and adaptive. As the ensemble evolves, the estimation converges towards a more accurate depiction of the true state.

Mathematical Mechanics:

The iterative evolution of the Particle Filter encapsulates the following sequence of steps:

1. Prediction: Propagate particles through the nonlinear state transition model, accounting for process noise:

$$x_{t+1}^{i} = f(x_t^i) + \epsilon_t^i$$

2. Weighting: Evaluate particle-measurement compatibility using measurement likelihood:

$$p(y_t \mid x_t^i)$$

Assign weights:

$$w_t^i \propto p(y_t \mid x_t^i)$$

Normalize weights:

$$w_t^i = \frac{p(y_t \mid x_t^i)}{\sum_{j=1}^{N} p(y_t \mid x_t^j)}$$

3. Resampling: Refresh the ensemble through resampling, guided by the weights. Higher weights indicate a higher likelihood of being selected.
4. Estimation: Synthesize the resampled particles to form an updated state estimate:

$$x^t = \sum_{i=1}^{N} w_t^i * x_t^i$$

Implications and Significance:

1. Adaptive Learning: "Iterative Evolution" ensures that the Particle Filter adapts to the evolving dynamics and measurements, refining its estimation over time.
2. Accommodating Change: As the system's state changes, the Particle Filter's estimation dynamically shifts to maintain relevance.
3. Balancing Exploration and Exploitation: The iterative nature allows the Particle Filter to balance between exploring new hypotheses and exploiting existing knowledge.

The essence of "Iterative Evolution" is the Particle Filter's persistent quest to unveil the hidden truth. Through the cyclical process of prediction, weighting, resampling, and estimation, the Particle Filter navigates the delicate dance between model and reality. It harnesses the power of adaptability, continuously refining its understanding of the system's true state with every step, every iteration, and every new measurement.

5.3 Advantages and Impact

Particle Filters: Unleashing the Power of Dynamic Estimation

Particle Filters are a dynamic and versatile family of algorithms that wield immense power in tackling the challenges of nonlinear, non-Gaussian state estimation. Their adaptability, ability to accommodate uncertainties, and capacity to track complex systems have made them a cornerstone in various fields, ranging from robotics and aerospace to finance and environmental monitoring. Let's delve profoundly into their advantages and the transformative impact they bring:

1. **Handling Nonlinearity and Non-Gaussianity:** Particle Filters excel where traditional methods struggle—capturing the intricate relationships of nonlinear systems. By

representing the state distribution with particles rather than relying on Gaussian assumptions, they can faithfully track the evolution of highly complex and non-Gaussian systems, ensuring that estimations are not limited by linearity.

2. **Accommodating Uncertainties:** Uncertainty is an inherent aspect of real-world systems. Particle Filters not only acknowledge this but thrive in such environments. Their ensemble-based nature allows them to explicitly model uncertainties, be it process noise, measurement noise, or model errors. This makes them robust and adaptable even when dealing with incomplete or noisy data.

3. **Real-Time Adaptability:** Particle Filters are designed for on-the-fly adaptation. With every new measurement, they update their estimation in real-time, allowing them to track dynamic systems that evolve over time. This adaptability is crucial in fields like robotics, where precise and real-time state estimation is paramount for successful navigation and control.

4. **Versatility Across Domains:** Particle Filters are domain-agnostic, transcending boundaries across various industries. In robotics, they enable simultaneous localization and mapping (SLAM); in finance, they underpin tracking financial instruments; in ecology, they monitor environmental changes. Their flexibility lies in their generic framework, making them a go-to choice for complex state estimation problems.

5. **Handling Multi-Modal Distributions:** In scenarios where the true state distribution is multi-modal, Particle Filters shine. Their ensemble nature enables them to capture multiple modes, ensuring that no significant mode of the distribution is left unexplored. This is crucial in scenarios where multiple hypotheses about the system's state are valid.

6. **Handling High-Dimensional Spaces:** Particle Filters remain effective even in high-dimensional state spaces, where traditional methods struggle due to the curse of dimensionality. Particle diversity prevents the ensemble from getting stuck in local optima, enabling them to explore large state spaces more effectively.

7. **Integration with Non-Parametric Approaches:** Particle Filters inherently embrace a non-parametric perspective, aligning well with modern trends in data analysis. They can be naturally integrated with techniques like kernel density estimation and Gaussian mixture models to enhance their capabilities.

8. **Navigating System Nonlinearity:** Particle Filters effortlessly navigate chaotic or unpredictable systems, enabling them to capture behaviors such as bifurcations and attractor transitions. This makes them suitable for systems where traditional linear approaches falter.

9. **Bridging Model and Reality:** Particle Filters bridge the gap between model predictions and real-world observations. Through resampling and weighted aggregation, they ensure that predictions resonate with actual measurements, striking a harmonious balance between theoretical insights and empirical evidence.

10. **Research and Innovation Catalyst:** Particle Filters continually push the boundaries of what's possible in state estimation. They have inspired research into advanced variants like the Unscented Kalman Filter and Gaussian Processes for Particle Filtering, driving the innovation engine in the field.

In essence, Particle Filters stand as a testament to the power of computational approaches in unraveling complex systems. Their ability to embrace uncertainty, adapt in real-time, and provide reliable estimates for nonlinear and non-Gaussian systems positions them as a pivotal tool in modern data analysis, enabling advancements across diverse disciplines.

5.4 Particle Filter Applications

Particle Filter and Monte Carlo method have found numerous important applications in various domains of life and industry (PM Djuric, 2003). Here are some of the most significant applications:

1. **Target Tracking and Localization:** In robotics and surveillance, particle filters are used to track moving objects like pedestrians, vehicles, or wildlife. They estimate the object's position and movement based on sensor data, which is crucial for applications like autonomous navigation and security systems. A figure showing the application of particle filter in localization is shown below (V Landa, 2018) where in the left part we have our particles and in the middle part we have our particles after Sequential importance resampling which leads us to the last photo in the right side where we accurately estimate the location of the object:

Figure 12. Particle Filter in localization application

2. **Robotics and Autonomous Navigation:** Robots, including autonomous cars and drones, rely on particle filters to estimate their position and orientation in real-time. This is essential for path planning, obstacle avoidance, and overall safe and efficient navigation.

3. **Human Activity Recognition:** In computer vision, particle filters are used to recognize human activities in video sequences. They can track and identify gestures, interactions, and behaviors, making them useful in applications like gaming, surveillance, and healthcare.

4. **Speech and Speaker Recognition:** Particle filters are applied in automatic speech recognition (ASR) systems to identify and distinguish speakers, which can be valuable in voice-controlled systems and security applications.

5. **Finance and Economics:** In finance, particle filters are used for estimating complex financial models, managing portfolio risk, and forecasting asset prices. They help in pricing derivatives and managing investment portfolios in uncertain markets.

6. **Environmental Monitoring:** Particle filters are employed in environmental studies to estimate the state of ecological systems. They can track the movement of animals, monitor pollution levels, and predict climate changes based on sensor data.

7. **Medical Imaging and Biomedical Applications:** In medical imaging, particle filters assist in tracking the motion of organs for radiation therapy, segmenting medical images for diagnosis, and estimating physiological parameters from patient data. They enhance the accuracy of medical diagnoses and treatments.

8. **Space Exploration:** Particle filters are used in spacecraft navigation to estimate the spacecraft's position, velocity, and orientation relative to celestial objects. This is crucial for autonomous space missions and rendezvous with other spacecraft or celestial bodies.

9. **Target Recognition and Defense:** Particle filters are applied in defense systems for tracking and recognizing targets like aircraft or missiles. They help guide interceptor missiles to their intended targets and assist in situational awareness.

10. **Finance and Risk Assessment:** In financial institutions, particle filters are used for risk assessment, credit scoring, fraud detection, and modeling financial market behaviors. They help in making informed decisions and managing financial risks.

11. **Industrial Process Control:** Particle filters assist in monitoring and controlling industrial processes, such as chemical manufacturing or power generation. They estimate the state of the process variables, ensuring efficient operation and early detection of anomalies.

12. **Oceanography and Weather Prediction:** Particle filters help scientists in oceanography by tracking ocean currents, modeling marine ecosystems, and studying

climate patterns. In weather forecasting, particle filters assimilate data from various sources to improve the accuracy of predictions.

5.5 Challenges and Frontiers

Particle Filters: Navigating Challenges and Pioneering Frontiers

While Particle Filters bring a formidable arsenal to the realm of state estimation, they are not immune to challenges and constraints. As technology evolves and computational capabilities advance, these challenges fuel innovation, leading to the exploration of new frontiers. Let's delve profoundly into the obstacles and the horizons that Particle Filters encounter:

Challenges:

1. **Computational Complexity:** Particle Filters demand computational resources that grow linearly with the number of particles used. As the number of particles increases, so does the computational burden. This can become a significant challenge when dealing with high-dimensional state spaces or real-time applications.
2. **Particle Degeneracy:** Over time, certain particles can accumulate much higher weights than others, leading to a situation known as particle degeneracy. This diminishes the diversity of the ensemble and can result in inaccuracies in the estimation process.
3. **Resampling Artifacts:** Resampling can inadvertently introduce artifacts, leading to non-physical particle distributions. These artifacts can distort the estimation process and need to be carefully managed through techniques like regularized resampling.
4. **Non-Gaussianity and High Dimensions:** In high-dimensional state spaces, maintaining a representative particle ensemble can be challenging due to the curse of dimensionality. Also, Particle Filters may struggle when dealing with multi-modal distributions or when Gaussian assumptions are unwarranted.
5. **Trade-off Between Exploration and Exploitation:** Choosing the right number of particles and the resampling threshold involves a delicate trade-off between exploration (exploring new hypotheses) and exploitation (exploiting known information). A wrong choice can lead to biased estimations or increased computational costs.

Frontiers:

1. **Advanced Resampling Techniques:** Innovations in resampling techniques aim to address challenges related to particle degeneracy and artifacts. Adaptive resampling,

regularized resampling, and hybrid resampling methods combine ideas from different domains to refine the process.

2. **Particle Diversity Maintenance:** Techniques to preserve particle diversity are actively researched. Methods like stratified resampling, auxiliary particle filters, and particle rejuvenation aim to counter the challenges of particle degeneracy.
3. **Gaussian Process Particle Filters:** This frontier blends the strengths of Gaussian Processes with Particle Filters. It involves using a Gaussian Process to model the posterior distribution of the state, allowing for more flexible and data-driven uncertainty modeling.
4. **Online Learning and Active Estimation:** Combining Particle Filters with online learning algorithms allows systems to adaptively learn and improve their estimation capabilities as new data becomes available. Active estimation strategies determine optimal measurements to acquire, enhancing estimation efficiency.
5. **Distributed Particle Filters:** With the rise of distributed systems, researchers are exploring ways to perform state estimation using distributed Particle Filters. This involves managing multiple ensembles across different nodes while exchanging information to obtain a global estimation.
6. **Variational Particle Filters:** Variational approaches combine Bayesian inference with optimization, providing a means to approximate complex posterior distributions. Variational Particle Filters strive to balance accuracy and computational efficiency.
7. **Approximate Bayesian Computing:** This innovative approach leverages particle-based methods to perform Bayesian computation when the likelihood function is computationally intractable or unavailable. It holds promise for solving complex problems in various domains.
8. **Hybrid Estimation Techniques:** Researchers are exploring hybrid methodologies that combine Particle Filters with other estimation techniques, such as Kalman Filters, Unscented Kalman Filters, and neural networks, to leverage the strengths of each approach.

In essence, the challenges faced by Particle Filters have spurred research and innovation, leading to the exploration of exciting frontiers in state estimation. As computational capabilities continue to evolve, Particle Filters are poised to overcome existing limitations and blaze new trails in diverse fields, contributing to our understanding of complex systems and enhancing our ability to make informed decisions in the face of uncertainty.

Chapter 6. Experimental and Numerical step

6.1 Numerical procedure

This section deals with a numerical implementation on MATLAB for the three kinds of filters previously discussed: Kalman Filter, Unscented Kalman Filter and Particle Filter. Three numerical integration schemes are adopted each time, namely Euler, Runge Kutta and Feedback Input integration. First, the procedure is validated on numerical data (from D Anastasio et al, 2020), and then on real data in the " Experimental setup".

The system studied is a single degree of freedom system with the following equation of motion:

$$m\ddot{\mathrm{x}}(t) + c\dot{\mathrm{x}}(t) + k\mathrm{x}(t) = \mathrm{f}(t)$$

Here the values that have been used in the algorithms are listed as below:

$m\ (mass) = 1.3\ kg$

$c\ (damping\ Coefficient) = 2\ \frac{Ns}{m}$

$k(stiffness) = 800\ \frac{N}{m}$

$Fs = 4000\ \ Hz\ (sampling\ frequency)$

With these values, a numerical model was constructed to represent the system. It is noteworthy that different numerical approaches entail distinct presentations. In the case of the Euler and Runge-Kutta approaches, the system is presented continuously, while the standard Feedback scheme necessitates a representation in the state space. Therefore, the determination of the subspace matrices, becomes essential. In this context, it is done as follows:

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix};\quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix};\quad C = [1 \quad 0];\quad D = [0]$$

The magnitude of the introduced noise remains constant in both the SDOF and MDOF systems, facilitating a clearer distinction in performance and improved result comparisons. Below, the noise levels are presented.

$Q = eye(2) * 1e - 5$ Process Noise

$R = eye(1) * 1e - 3$ Measurement Noise

First, the results for the SDOF system are presented. By implementing the algorithms with random input types for the Kalman Filter, following the procedures outlined in Chapter 3, the outcomes are depicted in Figure 20. The figure incorporates the results obtained using all three numerical approaches: Euler (green line), Runge-Kutta (black line), and Feedback input (blue line).

Two distinct segments of the figure, have been intentionally selected for magnification and additional presentation. This allows for a clearer observation of the performance of the various numerical approaches in estimating the system's state. The left segment is of particular interest because, as we approach peak values, filters typically encounter difficulties in accurately estimating the states at these points. However, it is evident that our Kalman filter accurately estimates these segments as well. Subsequently, in the RMS values, we will demonstrate that our Kalman filter exhibits exceptional performance in SDOF systems.

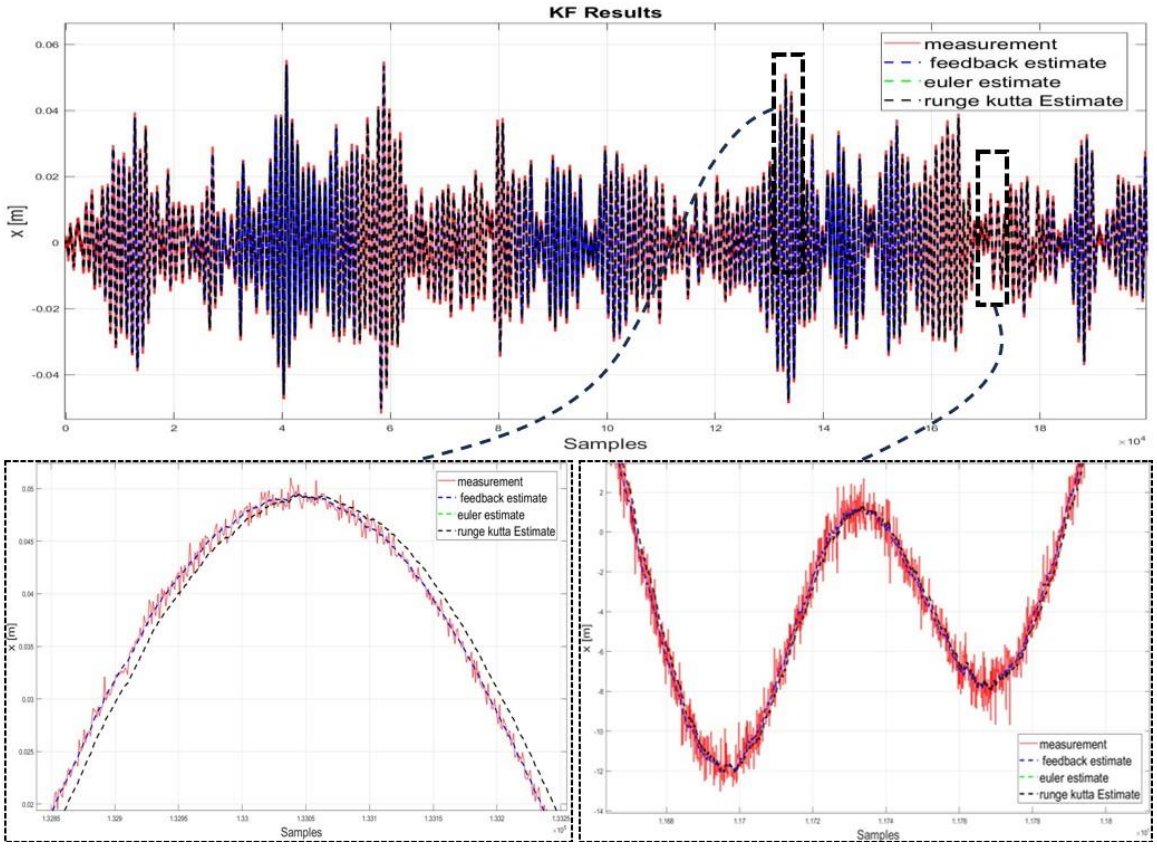


Figure 20. Kalman Filter result for SDOF in random input.

The results for the sweep input type in Kalman Filter for SDOF systems are displayed in Figure 21 below. It's important to note that the sweep input used is a linear sweep with the same frequency range as that employed for the random input.

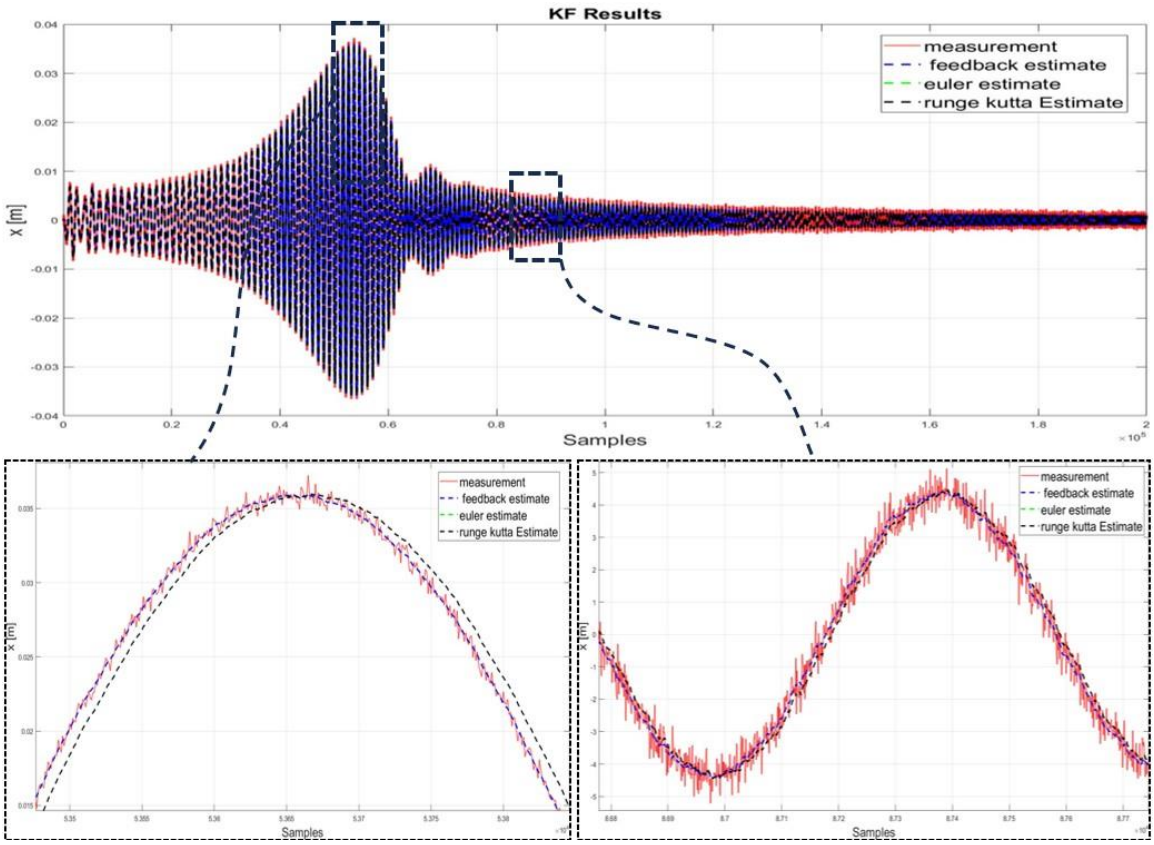


Figure 21. Kalman Filter result for SDOF in sweep input.

This pertains to a linear system using the Kalman Filter to estimate the system's states. For the nonlinear systems, two types of nonlinearities are introduced: one with a quadratic stiffness coefficient ($k_2 = 5000\ \frac{N}{m^2}$) and the other with a cubic stiffness coefficient ($k_3 = 1.5 * 10^6\ \frac{N}{m^3}$) (D Anastasio, 2020).

To account for these added nonlinear terms within our system, the equation of motion and state space model must be appropriately adjusted as follows:

$$m\ddot{x}(t) + c_v\dot{x}(t) + k_3x(t)^3 + k_2x(t)^2 + k_1x(t) = f(t)$$

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix};\quad B = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{m} & \frac{k_3}{m} & \frac{k_2}{m} \end{bmatrix};\quad C = [1 \quad 0];\quad D = [0 \quad 0 \quad 0]$$

Using the updated subspace matrices for the nonlinear system and employing both the Unscented Kalman Filter (UKF) and Particle Filter (PF) to facilitate a more comprehensive

comparison of results, various figures are presented. Firstly, Figure 22 illustrates the outcomes achieved with UKF and PF across different numerical schemes. Subsequently, the distinctions between UKF and PF within each numerical scheme are showcased individually in Figures 23.

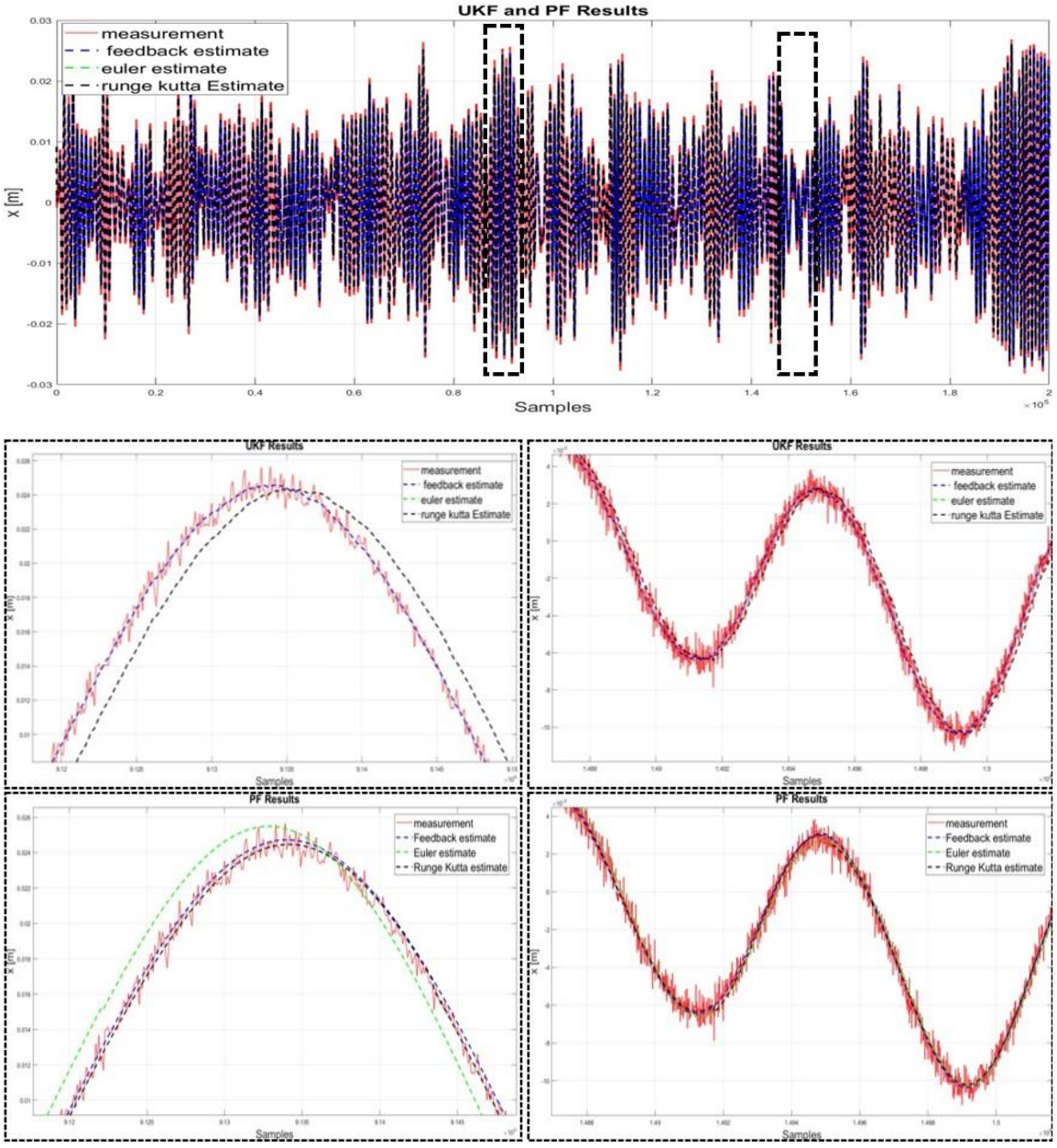


Figure 22. Unscented Kalman Filter and Particle Filter result for SDOF in random input

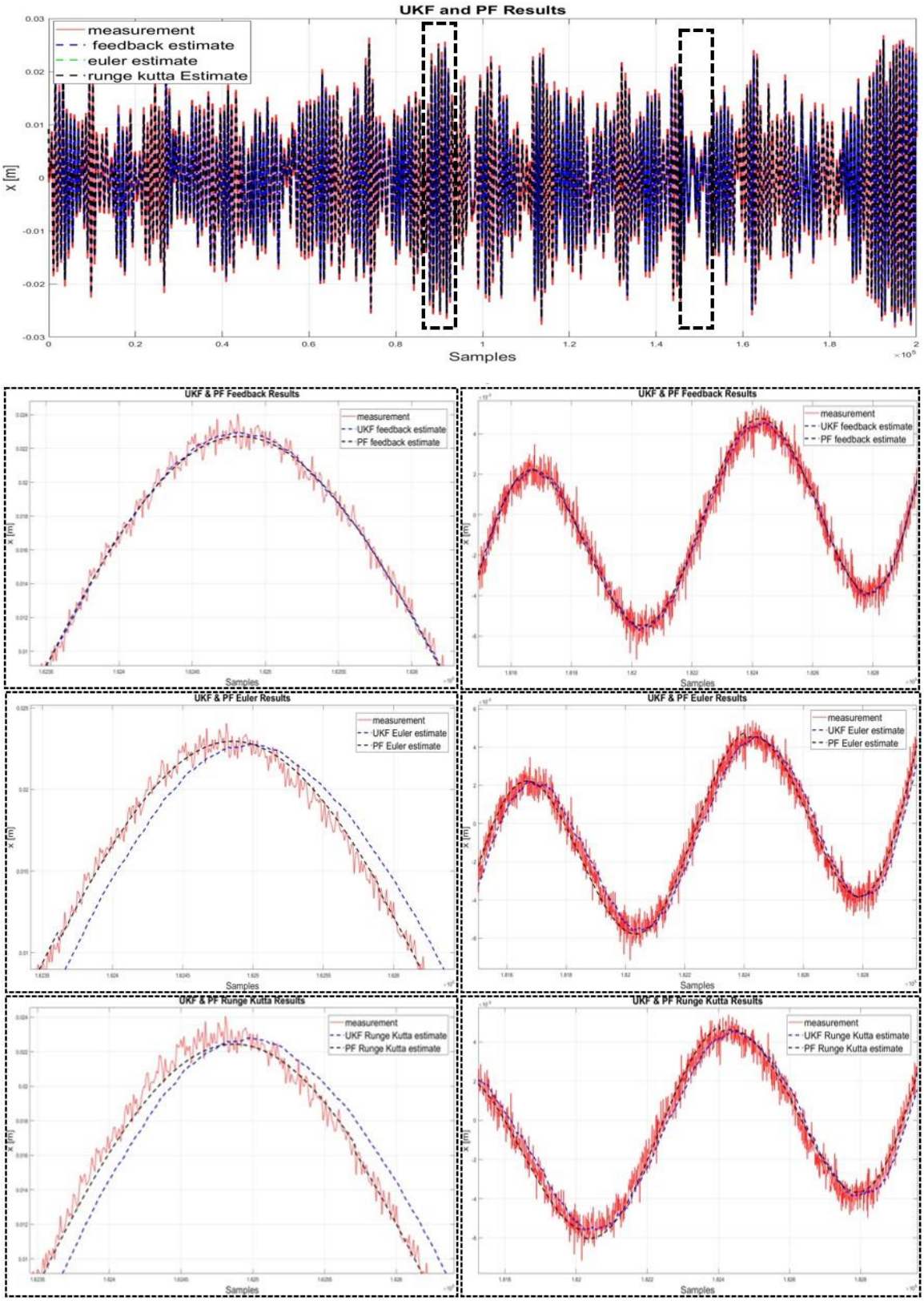


Figure 23. UKF and PF result for SDOF in random input in Feedback, Euler, and Runge kutta Scheme

The results for the sweep input type in Unscented Kalman Filter and Particle Filter for SDOF systems are displayed in Figure 24 and 25 below. It's important to note that the sweep input used is a Nonlinear sweep with the same frequency range as that employed for the random input.

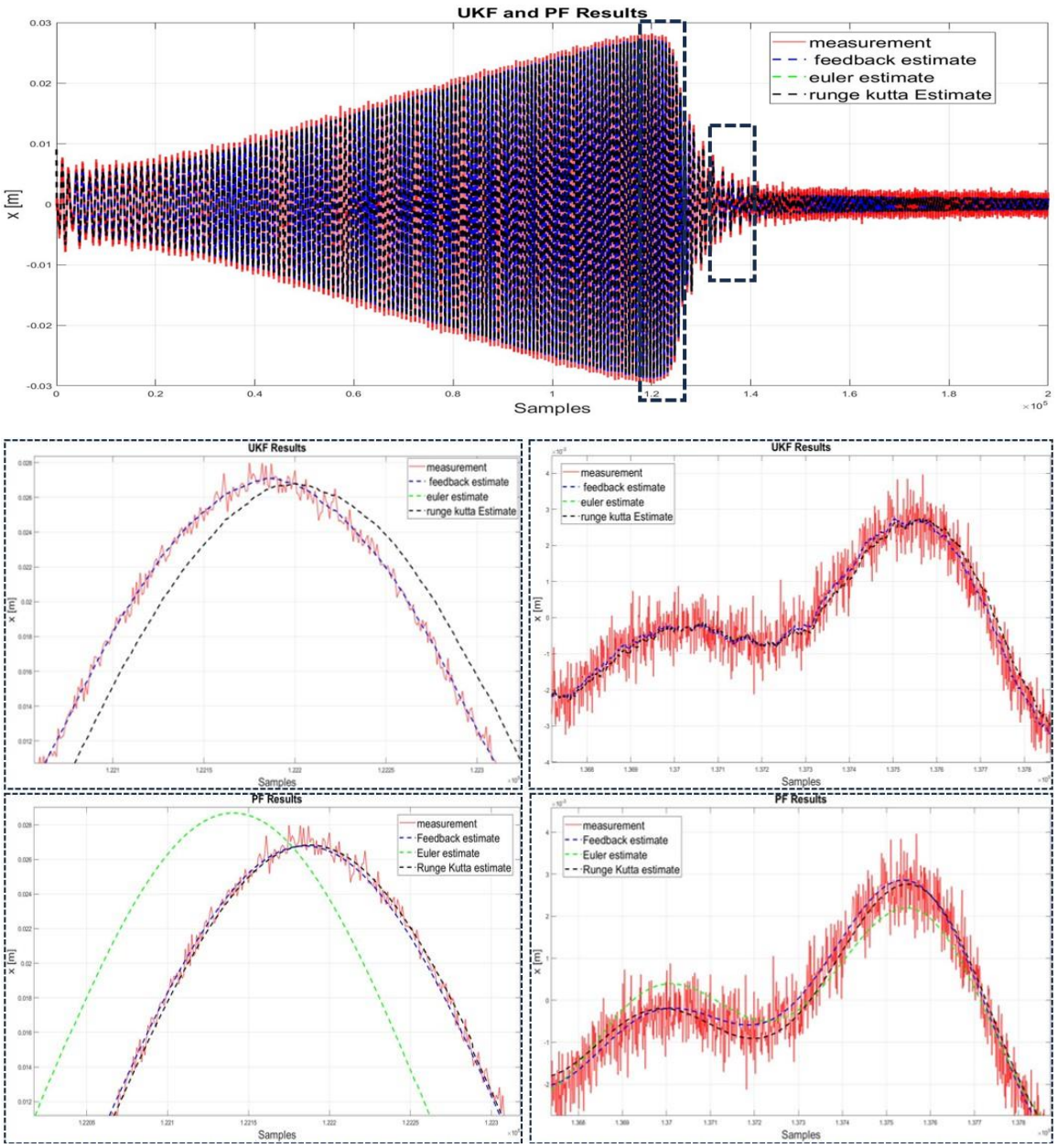


Figure 24. Unscented Kalman Filter and Particle Filter result for SDOF in sweep input

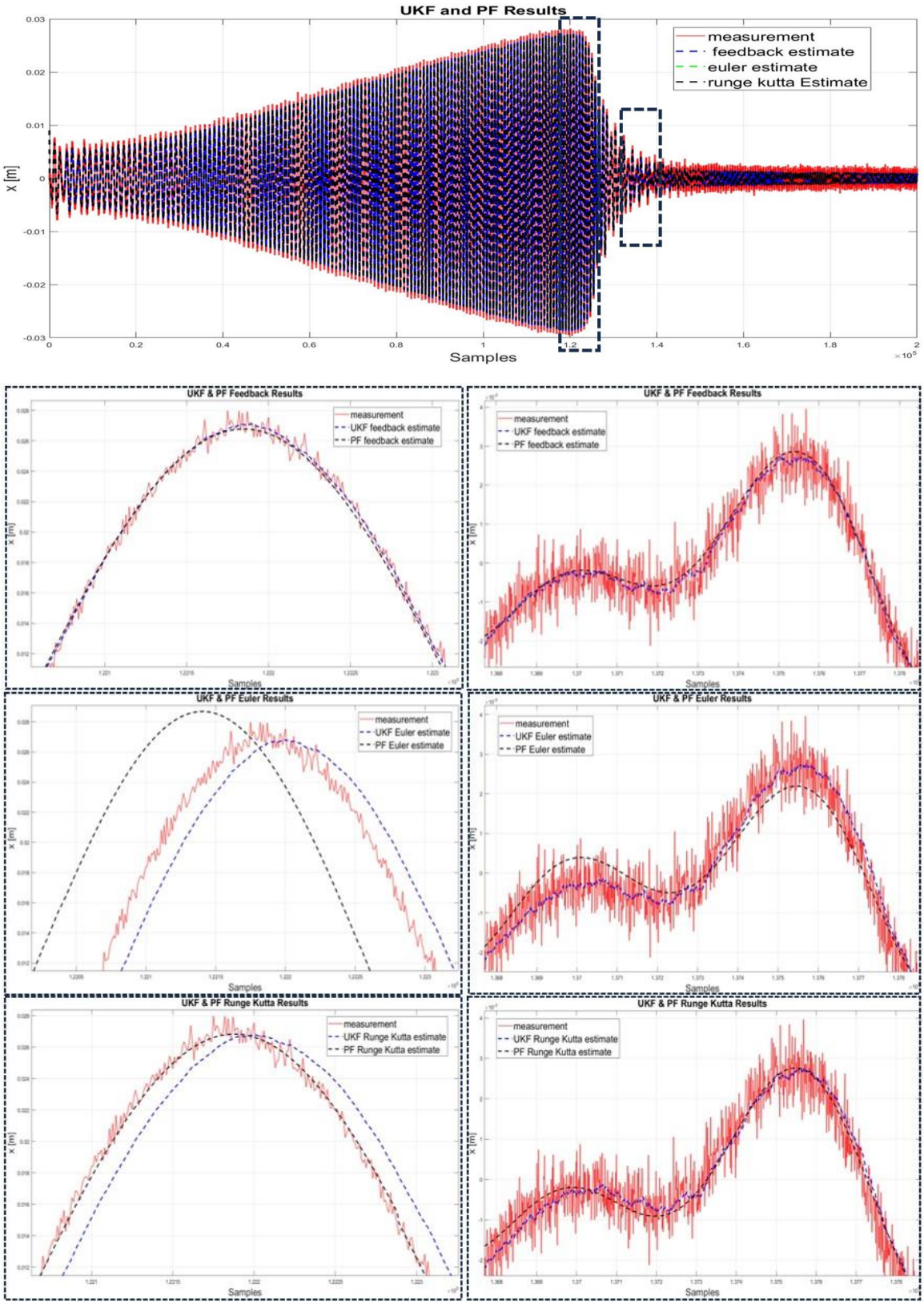


Figure 25. UKF and PF result for SDOF in sweep input in Feedback, Euler, and Runge kutta Scheme

To conduct the analysis in the 2DOF system, it is essential to introduce our 2DOF model. In the case of the linear implementation within our Kalman Filter, the following data, sourced from (S Marchesiello, 2008; D Anastasio, 2020), are employed, encompassing two masses and their corresponding system parameters:

$$m_1 = 0.5\ [kg];\ m_2 = 1\ [kg];\ c_1 = 4\ \left[\frac{Ns}{m}\right];\ c_2 = 1\ \left[\frac{Ns}{m}\right];\ k_{11} = 2e3\ \left[\frac{N}{m}\right];$$

$$k_{12} = 5e2\ \left[\frac{N}{m}\right];\ M = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix};\ c = \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix};\ k = \begin{bmatrix} k_{11} + k_{12} & -k_{12} \\ -k_{12} & k_{12} \end{bmatrix};$$

$$Fs = 4e3\ [Hz]\ (Sampling\ Frequency);\ dt = \frac{1}{Fs}\ [s]$$

Utilizing the same system parameters and maintaining an identical noise level as in the SDOF analysis, the procedure is implemented to ensure consistent conditions for comparing the filter's performance. Accordingly, the noise levels are as follows:

$Q = eye(2) * 1e - 5$ Process Noise

$R = eye(1) * 1e - 3$ Measurement Noise

Below, the results for the Kalman Filter in the 2DOF system are presented:

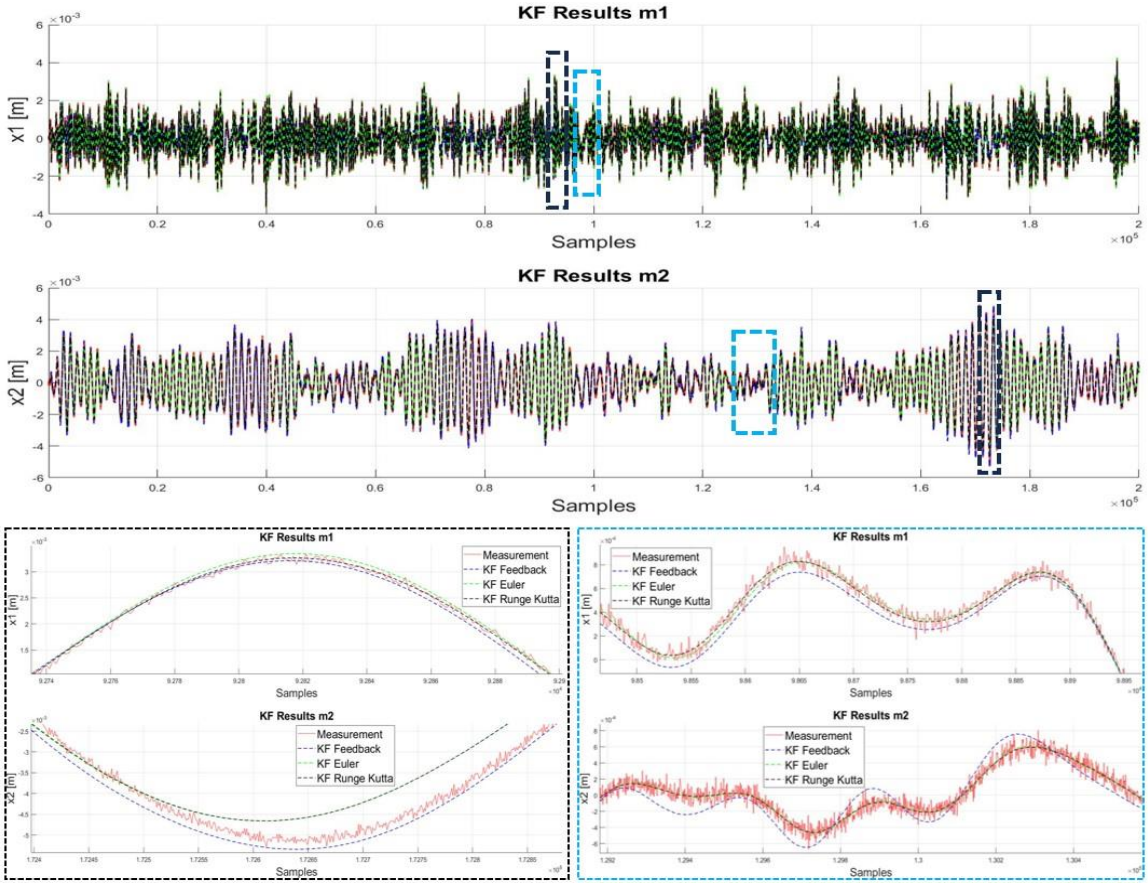


Figure 26. Kalman Filter 2DOF with Random Input.

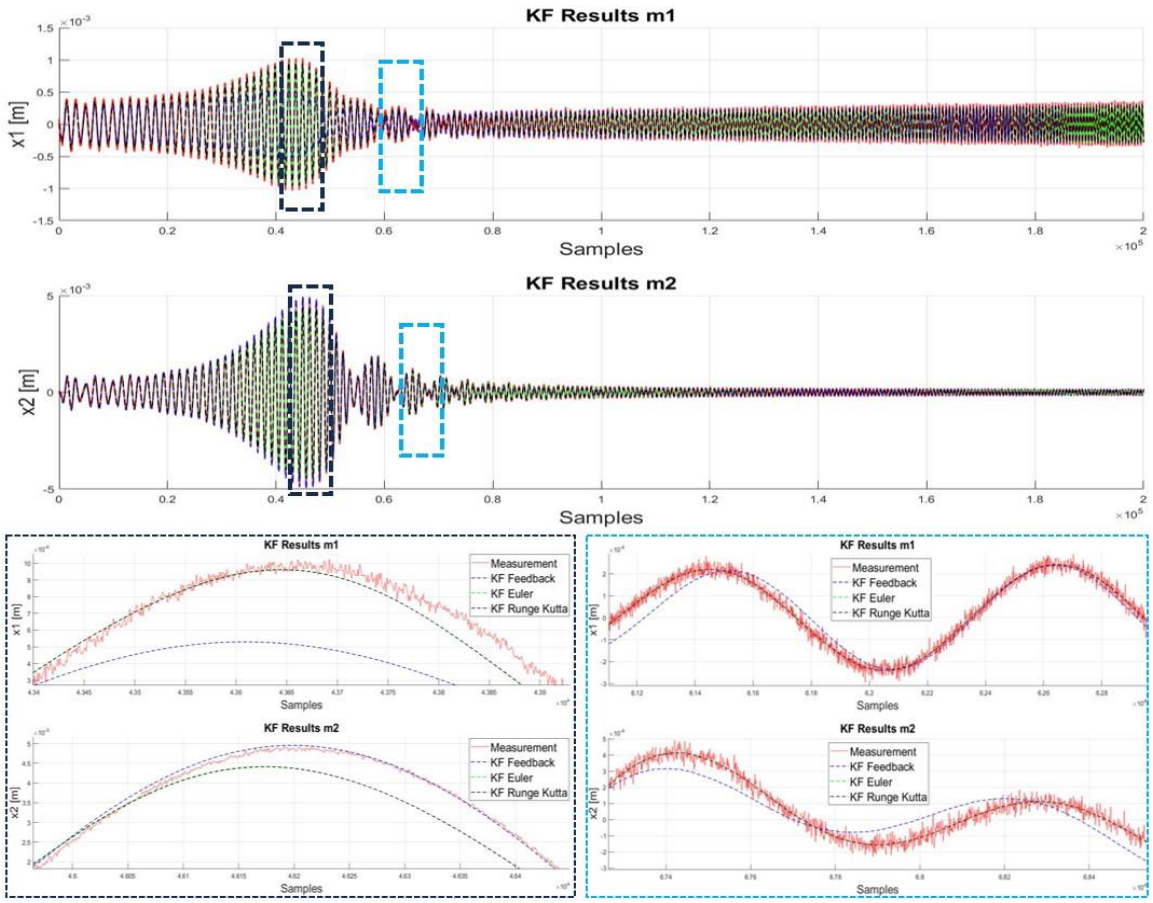


Figure 27. Kalman Filter 2DOF with Sweep Input.

This pertained to the linear system, where the Kalman Filter was employed for estimating the system's states. For the nonlinear systems, two types of nonlinearities were introduced: a quadratic stiffness coefficient ($k_2 = 5000\ \left[\frac{N}{m^2}\right]$) and a cubic stiffness coefficient ($k_3 = 1.5 * 10^6\ \left[\frac{N}{m^3}\right]$) (S Marchesiello, 2008; D Anastasio, 2020). With the incorporation of these nonlinear terms into our system, the state space model had to be accordingly adjusted to account for these nonlinearity effects. As a result, the modified values are presented below:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_{11}+k_{12}}{m_1} & \frac{k_{12}}{m_1} & -\frac{c_1+c_2}{m_1} & \frac{c_2}{m_1} \\ \frac{k_{12}}{m_2} & -\frac{k_{12}}{m_2} & \frac{c_2}{m_2} & -\frac{c_2}{m_2} \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{m_1} & 0 & \frac{k_3}{m_1} & 0 & \frac{k_2}{m_1} & 0 \\ 0 & \frac{1}{m_2} & 0 & \frac{k_3}{m_2} & 0 & \frac{k_2}{m_2} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad ; \quad D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In the previous subspace matrices, it's crucial to understand the relationships between columns and rows to correctly place parameters. Even a small mistake can significantly affect the outcome. Hence, a deep understanding of the system's model and physics informs these matrices.

These matrices apply to the Feedback input scheme in the predict step of our filters. For Euler and Runge-Kutta schemes in the predict step, we use numerical integration of the equation of motion. We also maintain the same noise level for consistent filter performance.

Below are the results for the 2DOF system with random input:

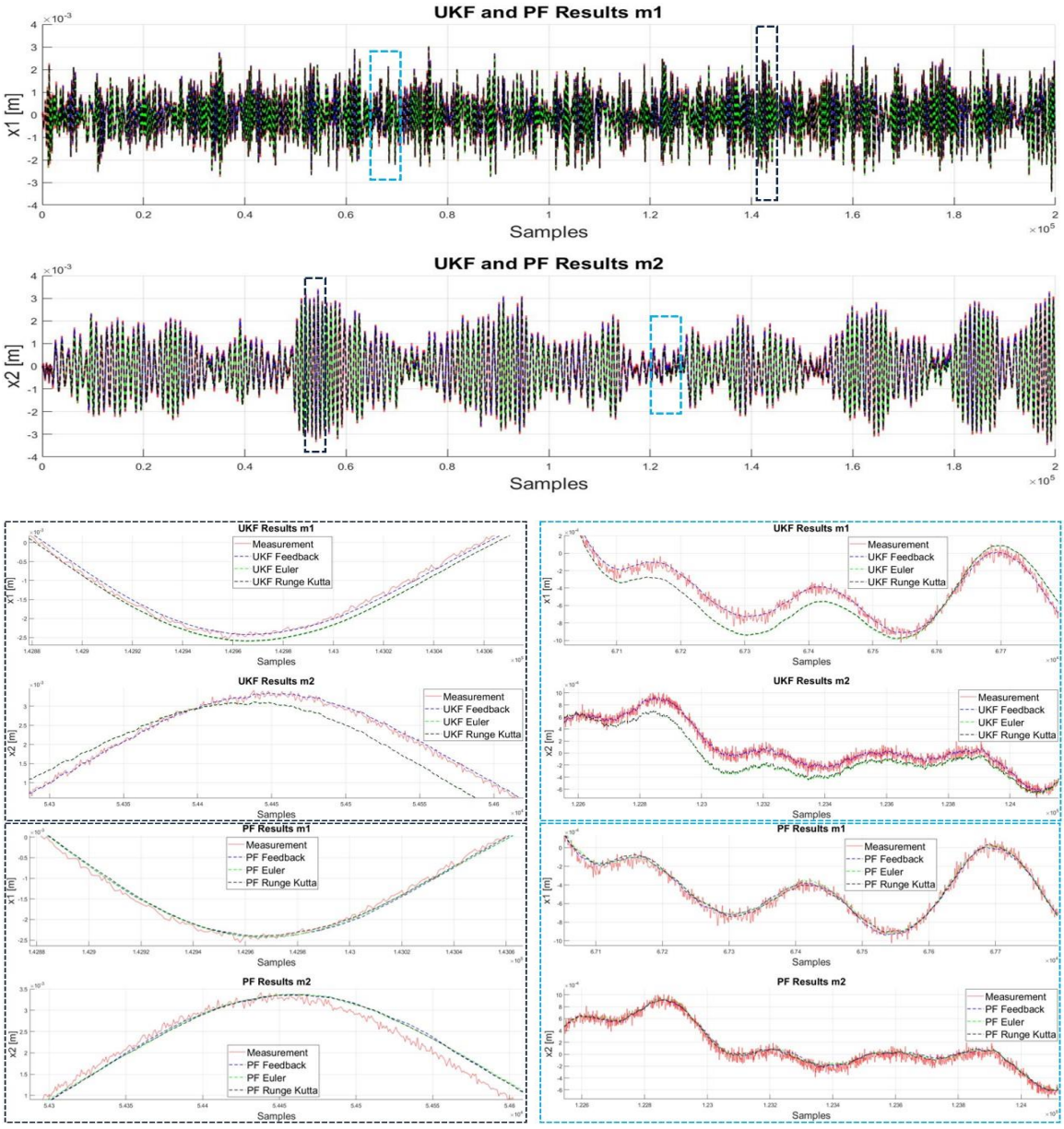


Figure 28. Unscented Kalman Filter and Particle Filter result for 2DOF in random input.

Here is the result for Feedback scheme and Runge Kutta scheme and these two scheme's performances in Random Input.

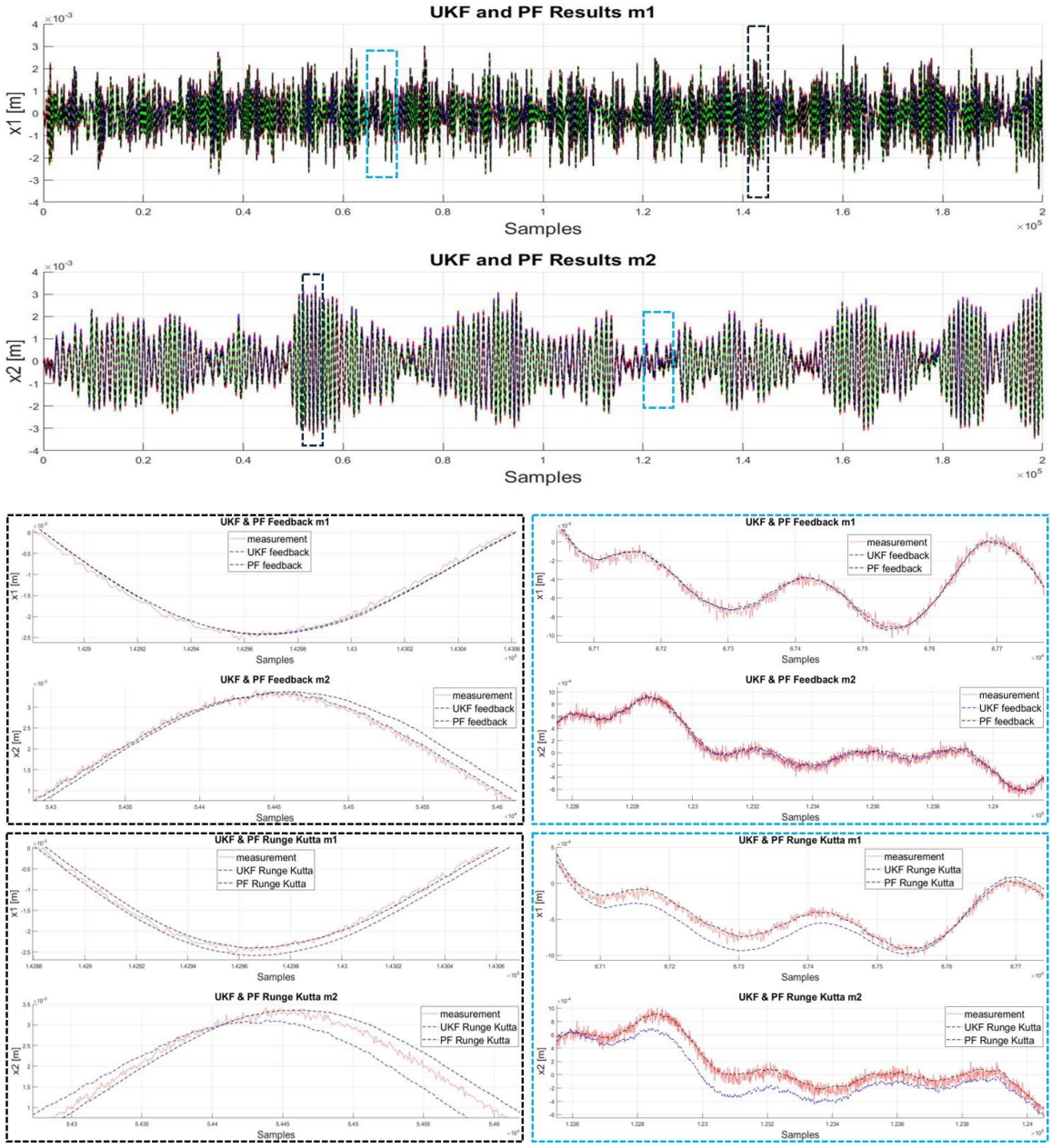


Figure 29. UKF and PF result for SDOF in random input in Feedback, and Runge kutta Scheme.

The same procedure is applied to the sweep input in 2DOF systems, yielding the results presented below:

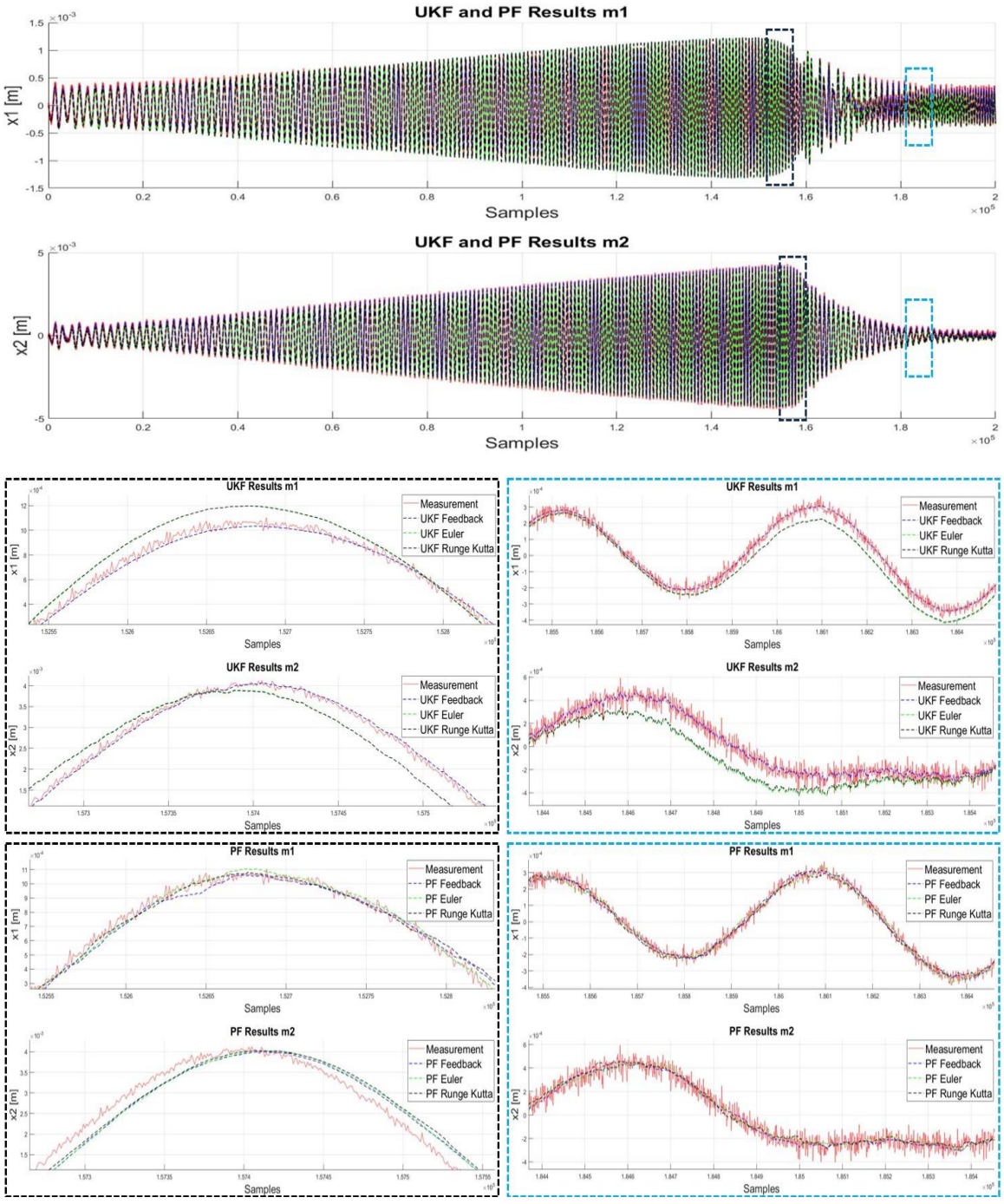


Figure 30. Unscented Kalman Filter and Particle Filter result for 2DOF in sweep input.

Below, the results for the Feedback scheme and the Runge-Kutta scheme, along with an assessment of their respective performances, are presented:

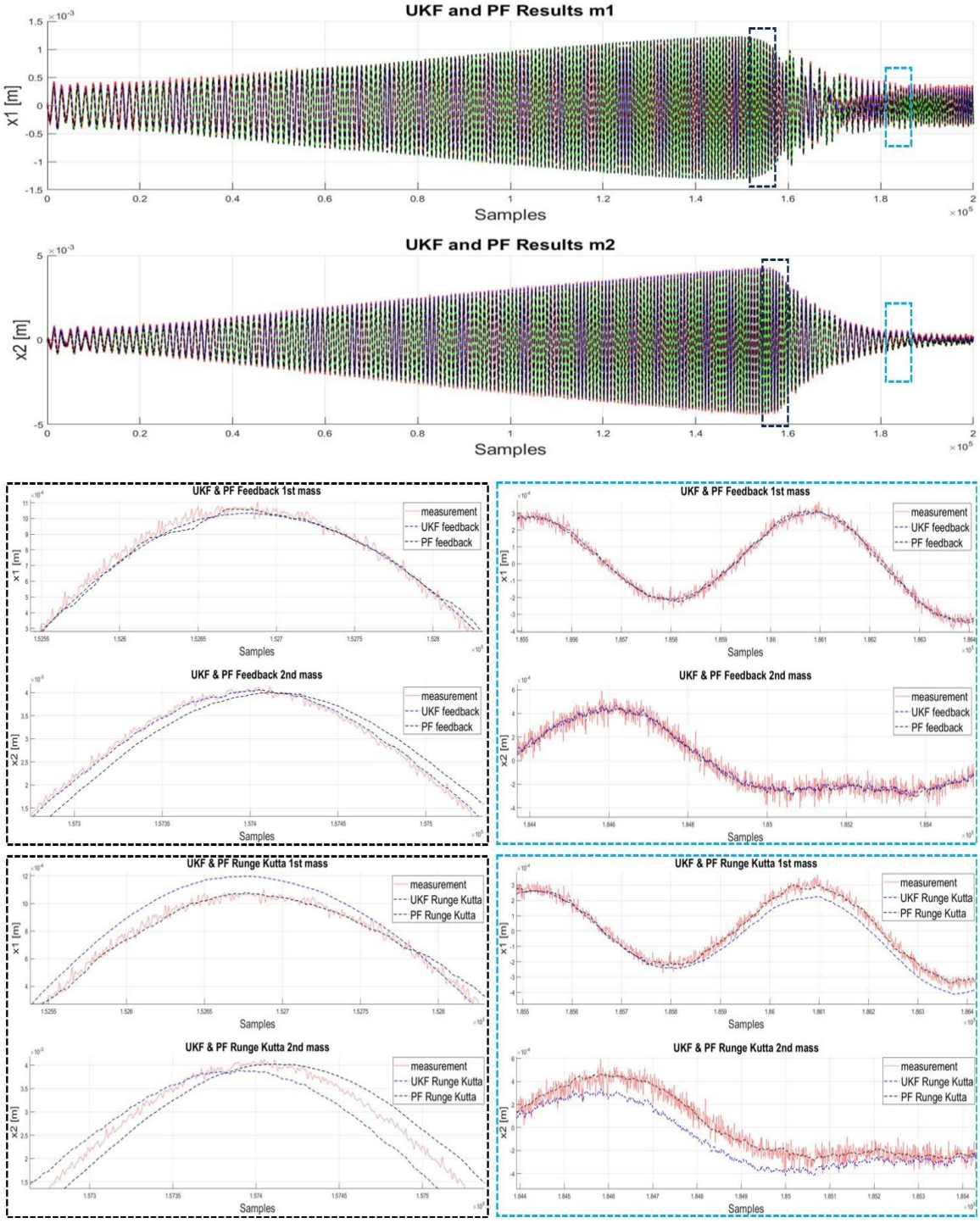


Figure 31. UKF and PF result for SDOF in random input in Feedback, and Runge kutta Scheme.

The Root Mean Square (RMS) is employed to evaluate the performance of filters in all schemes, both in SDOF and MDOF systems. The RMS provides a measure of the average magnitude of errors. The formula for RMS is:

$$RMS = \sqrt{\left[\frac{(\sum(x_i^2))}{N}\right]}$$

Where 'x_i' denotes individual data points, and 'N' represents the total number of data points.

The results are presented in both table and bar plot formats.

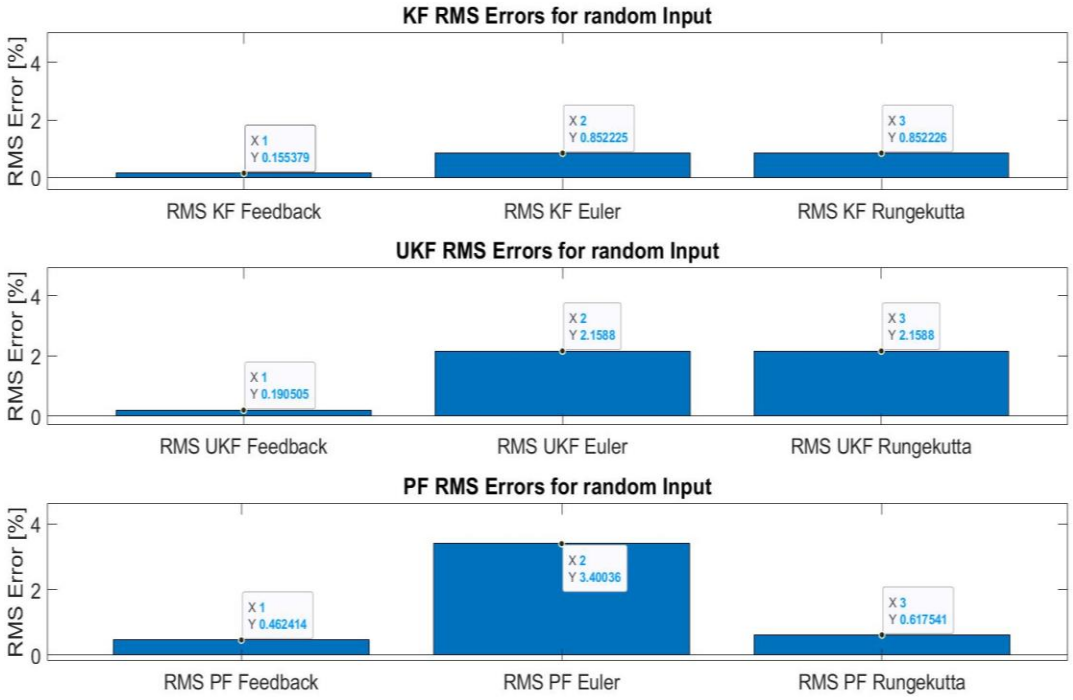
SDOF error results for both random and sweep input:

For Random Input:

RMS Table for KF, PF and UKF in random input:

KF 1DOF random	KF RMS [%]	UKF 1DOF random	UKF RMS [%]	PF 1DOF random	PF RMS [%]
{'RMS KF Feedback' }	0.15538	{'RMS UKF Feedback' }	0.19051	{'RMS PF Feedback' }	0.46241
{'RMS KF Euler' }	0.85223	{'RMS UKF Euler' }	2.1588	{'RMS PF Euler' }	3.4004
{'RMS KF Rungekutta'}	0.85223	{'RMS UKF Rungekutta'}	2.1588	{'RMS PF Rungekutta'}	0.61754

RMS Bar plot for KF, PF and UKF in random input:

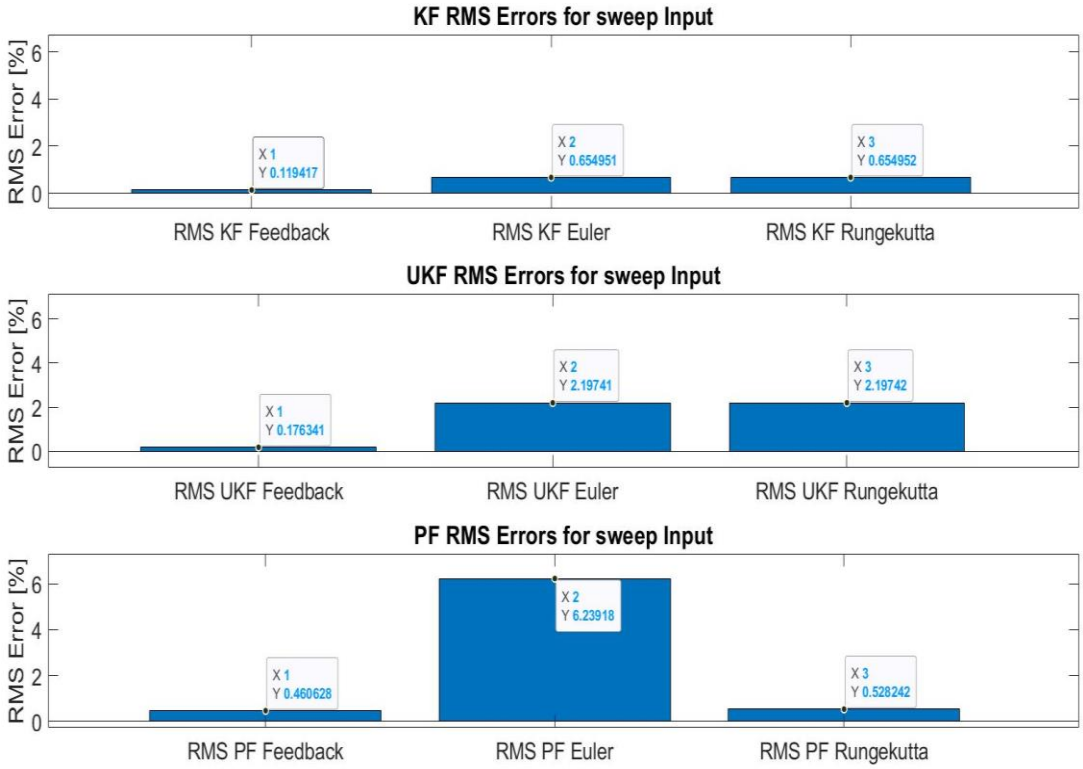


For Sweep Input:

RMS Table for KF, PF and UKF in sweep input:

KF 1DOF sweep	KF RMS [%]	UKF 1DOF sweep	UKF RMS [%]	PF 1DOF sweep	PF RMS [%]
{'RMS KF Feedback' }	0.11942	{'RMS UKF Feedback' }	0.17634	{'RMS PF Feedback' }	0.46063
{'RMS KF Euler' }	0.65495	{'RMS UKF Euler' }	2.1974	{'RMS PF Euler' }	6.2392
{'RMS KF Rungekutta'}	0.65495	{'RMS UKF Rungekutta'}	2.1974	{'RMS PF Rungekutta'}	0.52824

RMS Bar plot for KF, PF and UKF in sweep input:



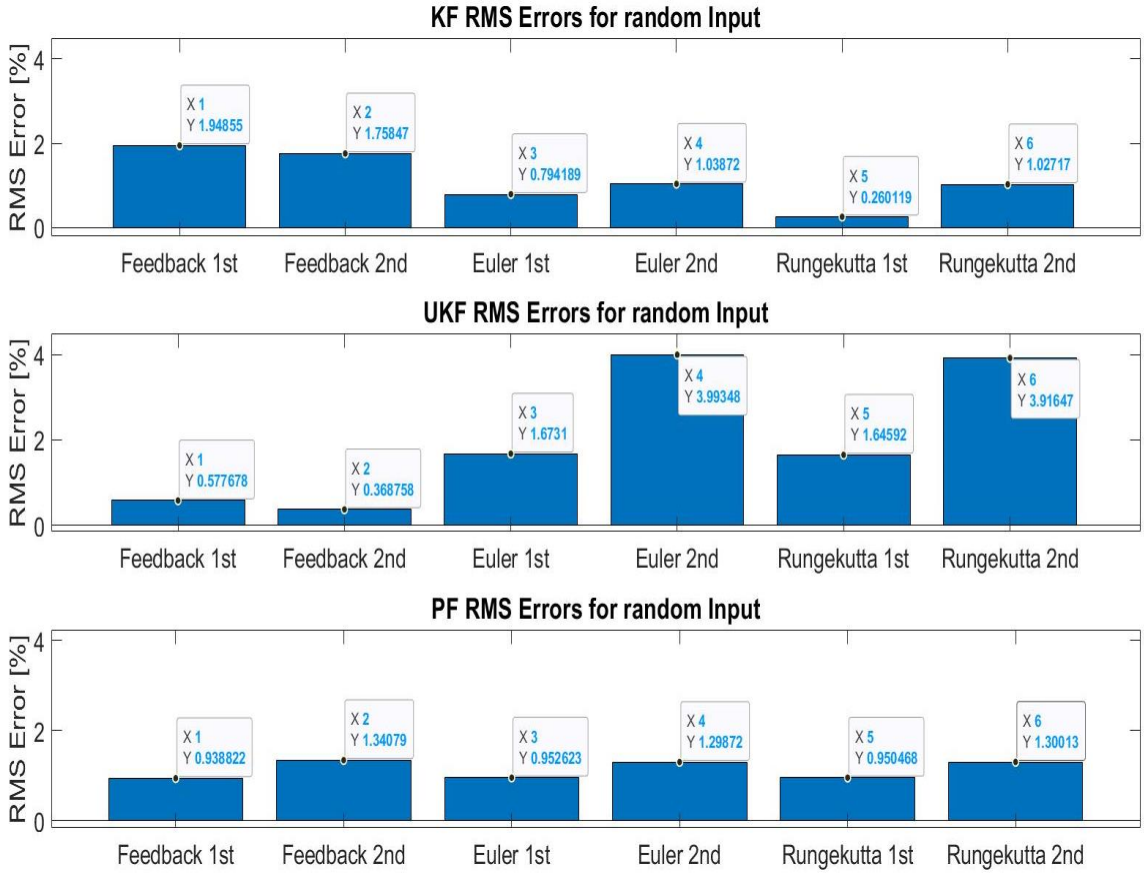
MDOF error results:

For Random Input:

RMS Table for KF, PF and UKF in random input:

KF 2DOF random	KF RMS [%]	UKF 2DOF random	UKF RMS [%]	PF 2DOF random	PF RMS [%]
{'RMS KF Feedback 1st mass' }	1.9485	{'RMS UKF Feedback 1st mass' }	0.57768	{'RMS PF Feedback 1st mass' }	0.93882
{'RMS KF Feedback 2nd mass' }	1.7585	{'RMS UKF Feedback 2nd mass' }	0.36876	{'RMS PF Feedback 2nd mass' }	1.3408
{'RMS KF Euler 1st mass' }	0.79419	{'RMS UKF Euler 1st mass' }	1.6731	{'RMS PF Euler 1st mass' }	0.95262
{'RMS KF Euler 2nd mass' }	1.0387	{'RMS UKF Euler 2nd mass' }	3.9935	{'RMS PF Euler 2nd mass' }	1.2987
{'RMS KF Rungekutta 1st mass'}	0.26012	{'RMS UKF Rungekutta 1st mass'}	1.6459	{'RMS PF Rungekutta 1st mass'}	0.95047
{'RMS KF Rungekutta 2nd mass'}	1.0272	{'RMS UKF Rungekutta 2nd mass'}	3.9165	{'RMS PF Rungekutta 2nd mass'}	1.3001

RMS Bar plot for KF, PF and UKF in random input:

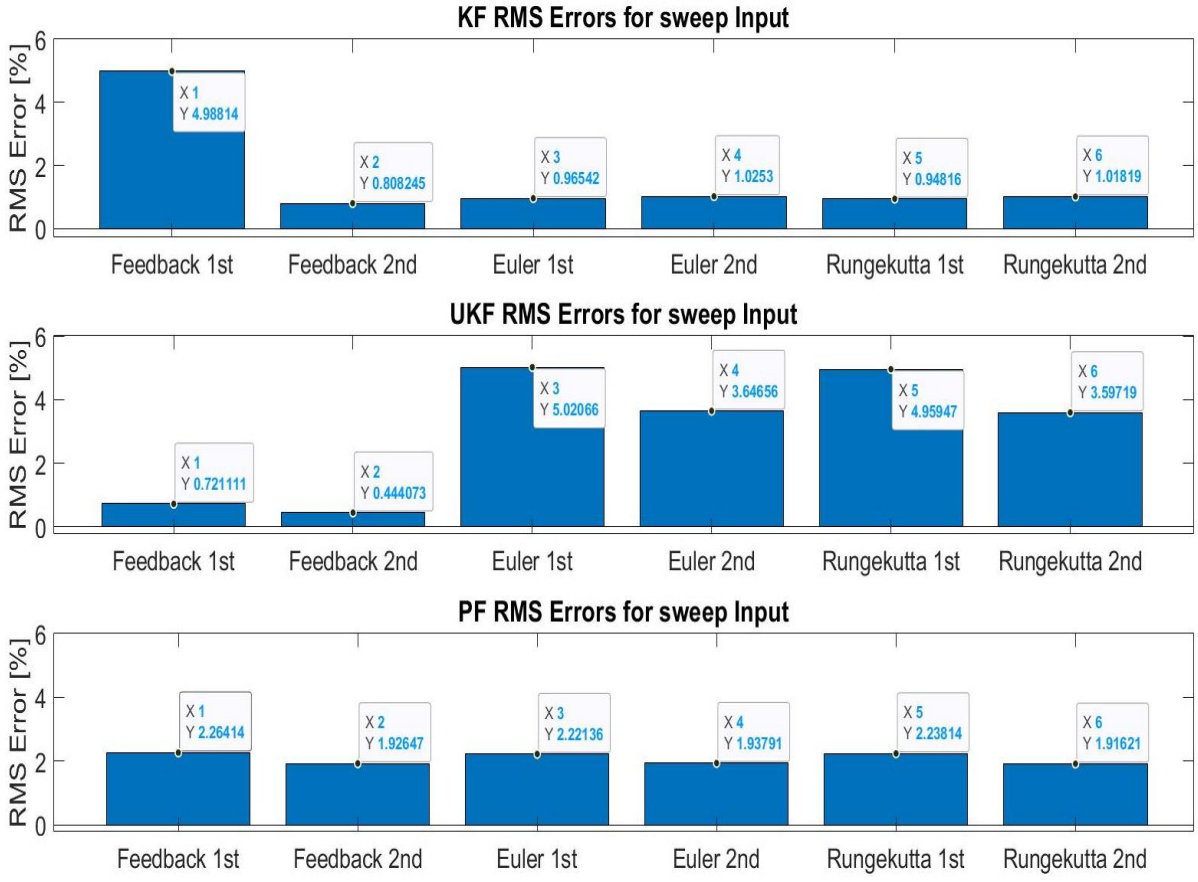


For the Sweep Input:

RMS Table for KF, PF and UKF in sweep input:

KF 2DOF sweep	KF RMS [%]	UKF 2DOF sweep	UKF RMS [%]	PF 2DOF sweep	PF RMS [%]
{'RMS KF Feedback 1st mass' }	4.9881	{'RMS UKF Feedback 1st mass' }	0.72111	{'RMS PF Feedback 1st mass' }	2.2641
{'RMS KF Feedback 2nd mass' }	0.80824	{'RMS UKF Feedback 2nd mass' }	0.44407	{'RMS PF Feedback 2nd mass' }	1.9265
{'RMS KF Euler 1st mass' }	0.96542	{'RMS UKF Euler 1st mass' }	5.0207	{'RMS PF Euler 1st mass' }	2.2214
{'RMS KF Euler 2nd mass' }	1.0253	{'RMS UKF Euler 2nd mass' }	3.6466	{'RMS PF Euler 2nd mass' }	1.9379
{'RMS KF Rungekutta 1st mass'}	0.94816	{'RMS UKF Rungekutta 1st mass'}	4.9595	{'RMS PF Rungekutta 1st mass'}	2.2381
{'RMS KF Rungekutta 2nd mass'}	1.0182	{'RMS UKF Rungekutta 2nd mass'}	3.5972	{'RMS PF Rungekutta 2nd mass'}	1.9162

RMS Bar plot for KF, PF and UKF in sweep input:



At the culmination of this results chapter, it is important to note that the overarching conclusions drawn from these findings have been expounded upon in the subsequent conclusion chapter. For a comprehensive understanding of the implications and significance of these results, readers are encouraged to refer to the conclusion chapter.

6.2 Experimental setup

In this section, a detailed description of the experimental setup utilized for parameter estimation of the double-well vibrating system (D Anastasio, 2020) is provided. This system consists of a u-shaped steel frame connected to a movable mass with vertically guided motion. Base excitation is introduced through a shaking table, resulting in the imposition of a vertical displacement, denoted as $b(t)$, onto the structure. Please refer to Figure 16 and 17 for schematic representations of the experimental device with a free body diagram and a corresponding photograph for visual reference.

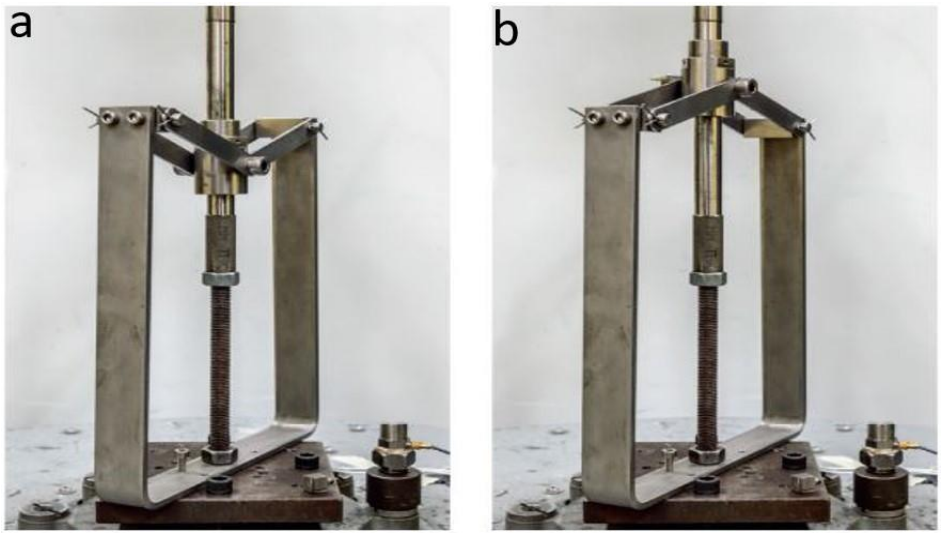


Figure 16: Photos of the experimental setup: (a) Negative Equilibrium position z_-^* , (b) Positive Equilibrium position z_+^*.

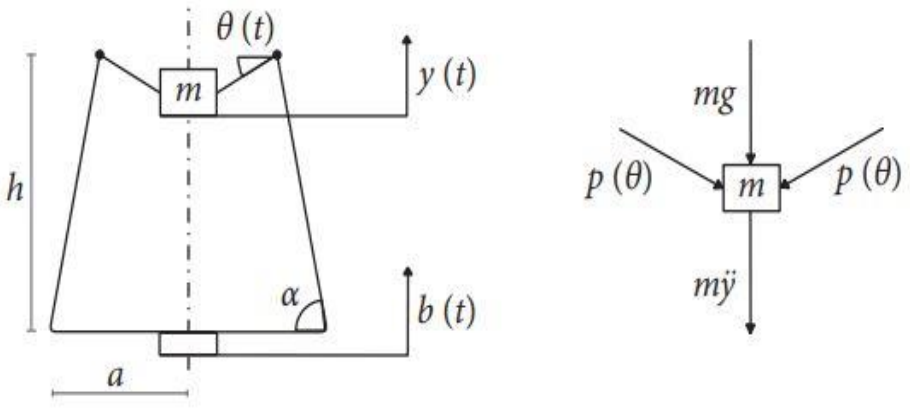


Figure 17. Model of the negative oscillator and free-body diagram of m.

The equation of motion for this system, as derived in [29], considers the vertical motion of the mass and demonstrates a double-well potential characteristic, leading to the presence of three equilibrium positions, two of which are stable. In the context of this study, exclusive attention is given to in-well oscillations centered around one of the stable equilibrium positions. The equation of motion is presented as follows:

$$m\ddot{\mathrm{x}}(t) + c_v\dot{\mathrm{x}}(t) + k_3\mathrm{x}(t)^3 + k_2\mathrm{x}(t)^2 + k_1\mathrm{x}(t) = -m\ddot{b}(t)$$

Here, 'm' denotes the mass of the moving part, 'c_v' represents the viscous damping coefficient, and 'k_1', 'k_2', and 'k_3' correspond to the linear, quadratic, and cubic stiffness coefficients, respectively. The external excitation is provided by the acceleration of the base, denoted as '$\ddot{b}(t)$'.

The damping model, extensively explored in [44], takes into account the effects of the sliding motion of the moving mass. This model is particularly relevant when studying cross-well oscillations. However, for the purposes of this study, which is focused on in-well motion, a simplified equivalent viscous damping term is employed.

To measure and record the behavior of the system, the following instrumentation has been implemented:

- **Laser Vibrometer**: The moving mass is equipped with a laser vibrometer, enabling the measurement of its absolute displacement '$y(t)$'. Data collection commences from the equilibrium position illustrated in Figure 16.
- **Accelerometer**: An accelerometer is utilized to record the acceleration of the base '$\ddot{b}(t)$'.
- **Relative Displacement**: The relative displacement '$x(t)$' is calculated by taking the difference between the laser-measured displacement '$y(t)$' and the displacement of the base '$b(t)$', which is obtained by integrating its measured acceleration twice.

For the experimental trials, random excitation is applied to the system using a sampling frequency of 512 Hz and a time span of 300 seconds. The root mean square (RMS) value of the base acceleration is $26\ \frac{m}{s^2}$ RMS and the time history of the relative displacement $x(t)$ is shown in Figure 18.

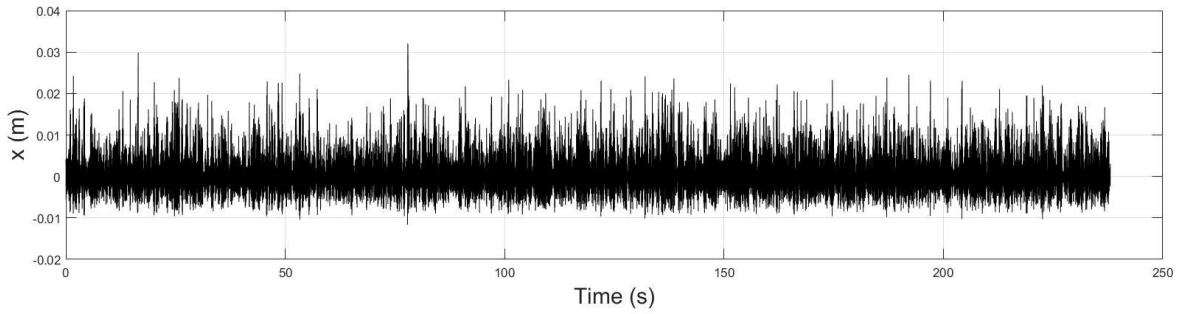


Figure 18: Time history of the relative displacement x(t).

The estimated system parameters (D Anastasio, 2020) obtained from these experiments are as follows:

- Estimated Natural Frequency: 11.22 Hz
- Damping Ratio: 6.22%
- Estimated Nonlinear Coefficients:
    - Quadratic Stiffness (k_2): $-4.5501 \times 10^5\ Nm^{-2}$
    - Cubic Stiffness (k_3): $1.8924 \times 10^5\ Nm^{-3}$

In the following sections of this thesis, Bayesian inference techniques, such as the Kalman filter, unscented Kalman filter, and particle filter, will be utilized to estimate the states of the complex nonlinear dynamic system.

Now, the results of the procedure outlined in the numerical step for the real measurement data of the previously described system are presented below:

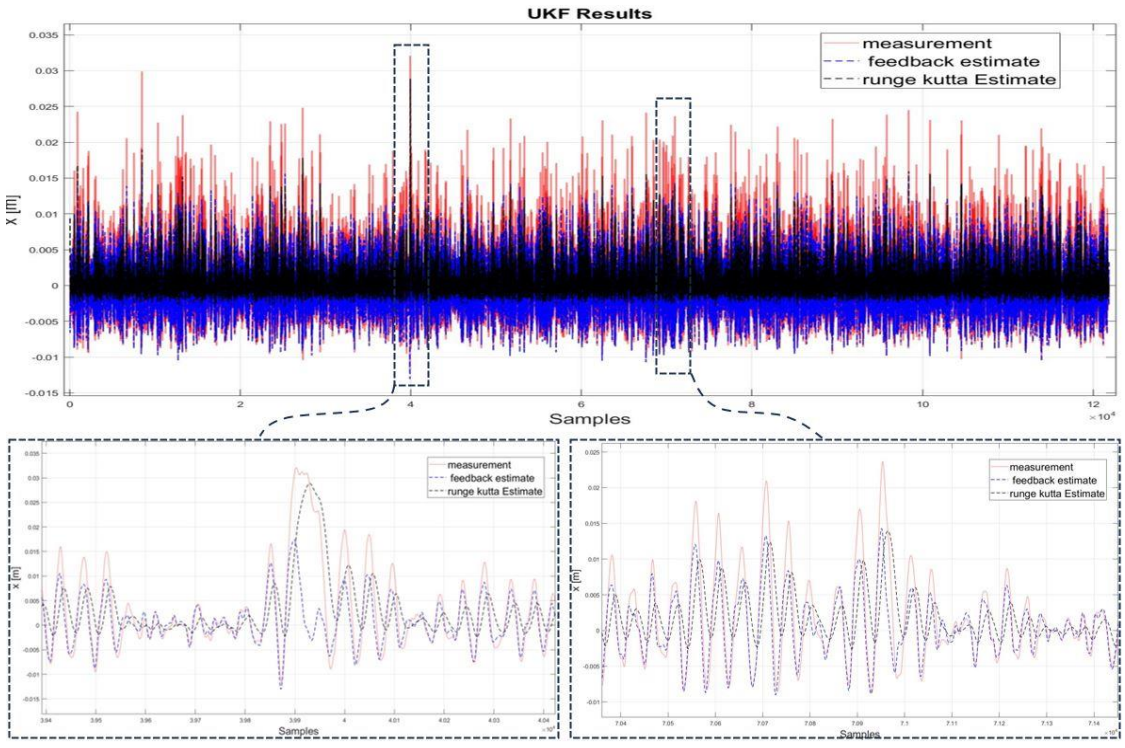


Figure. UKF State estimates with Feedback and Runge Kutta schemes.

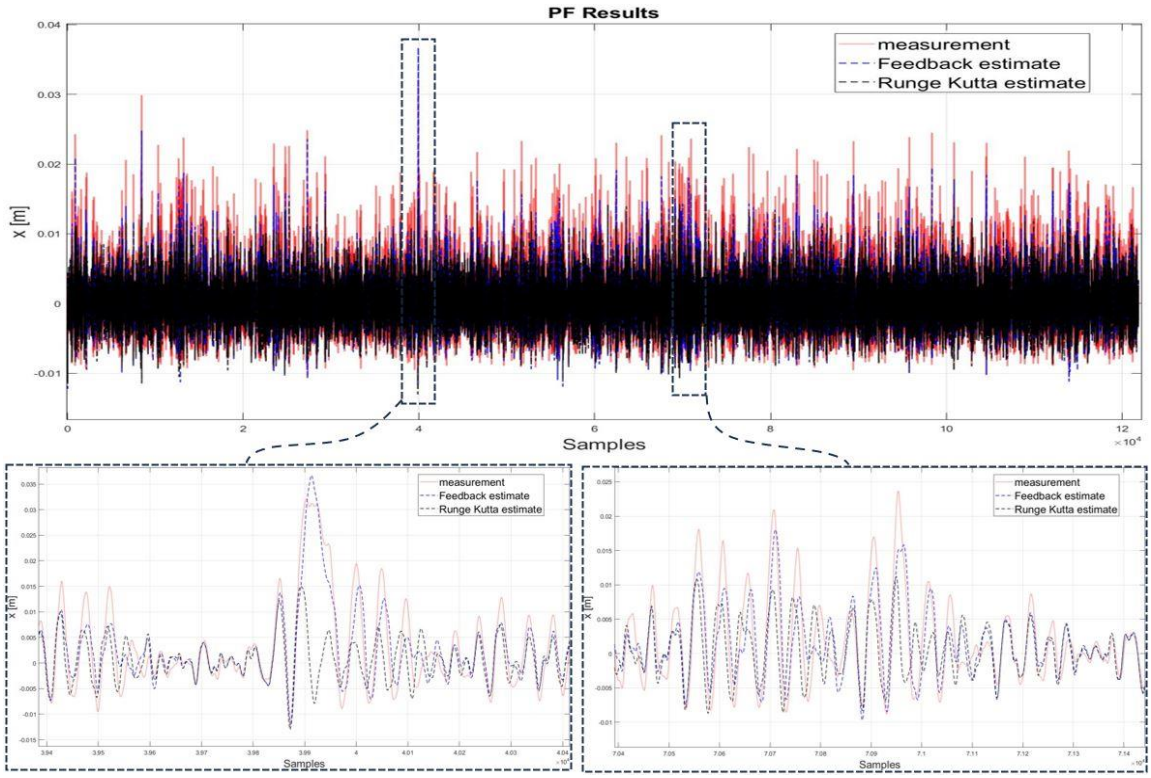


Figure. PF State estimate with Feedback and Runge Kutta schemes.

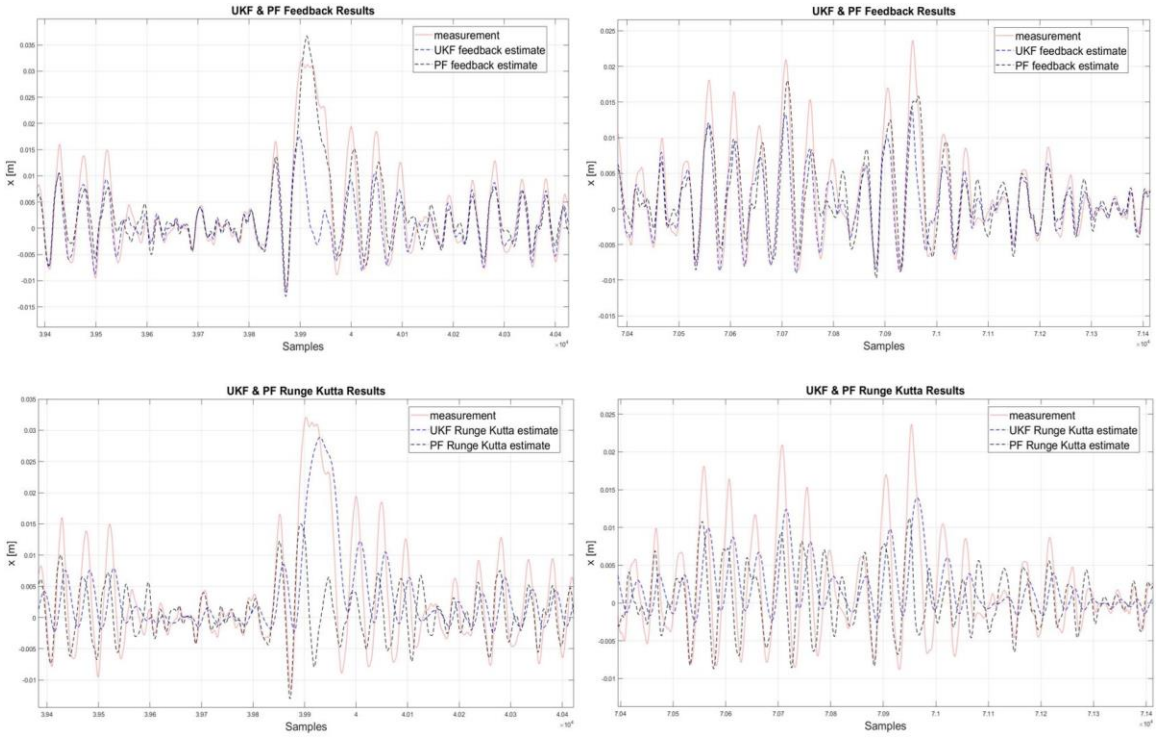


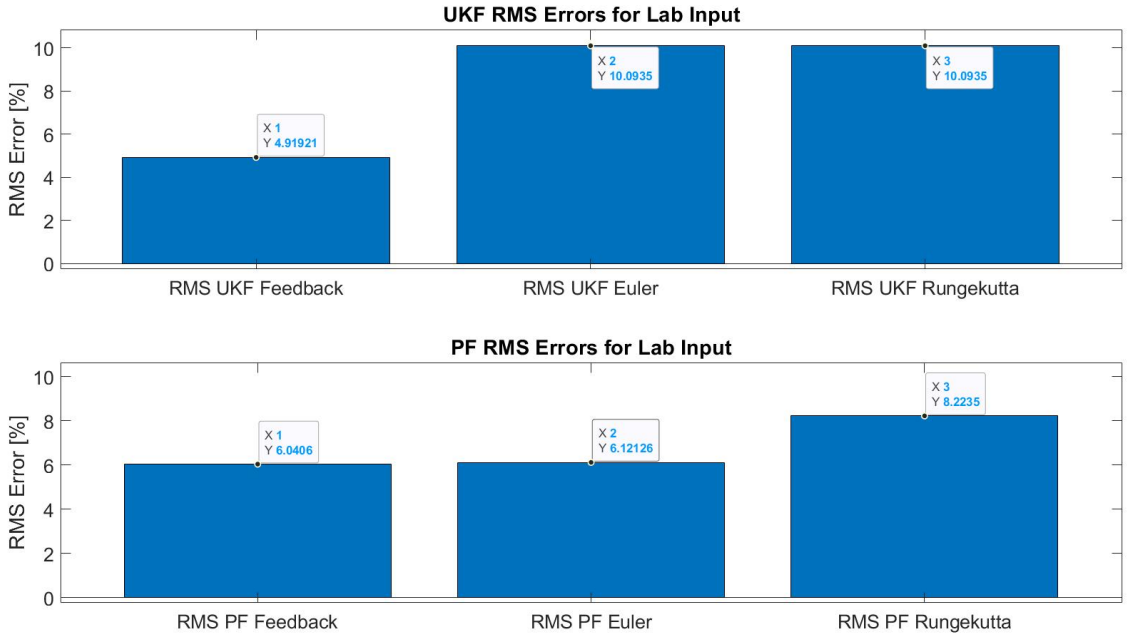
Figure. Runge Kutta and Feedback differences in UKF and PF

The Root Mean Square (RMS) error of the filters' performance in all schemes for the real measurement system has been computed based on the available data. These results are presented in both tabular and bar plot formats.

Table of RMS for real measurement data:

UKF 1DOF random	UKF RMS [%]	PF 1DOF random	PF RMS [%]
{'RMS UKF Feedback' }	4.9192	{'RMS PF Feedback' }	6.0406
{'RMS UKF Euler' }	10.093	{'RMS PF Euler' }	6.1213
{'RMS UKF Rungekutta'}	10.093	{'RMS PF Rungekutta'}	8.2235

Bar plot of RMS for real measurement data:



Chapter 7. Discuss and Conclusion

Interpreting the Results for SDOF:

1. Kalman Filter (KF) for Linear Systems: The Kalman Filter, tailored for linear systems with Gaussian noise, demonstrates superior performance within the feedback scheme. Linear systems inherently align with KF's assumptions of linearity and Gaussianity. However, in the Euler and Runge-Kutta schemes, the introduction of nonlinearity by these numerical integration methods is believed to disrupt the filter's assumptions, subsequently resulting in increased estimation errors. KF is most appropriate when applied to linear systems where Gaussian noise effectively models uncertainties.

2. Unscented Kalman Filter (UKF) for Nonlinear Systems: The Unscented Kalman Filter excels in estimating states within nonlinear systems. In the feedback scheme, it outperforms the Euler and Runge-Kutta schemes owing to its capability to approximate the state distribution through the use of meticulously selected sigma points. This enables the filter to more accurately capture the system's behavior. When the system exhibits nonlinear dynamics, UKF's feedback scheme emerges as a robust choice for achieving accurate estimations.

3. Particle Filter (PF) for Nonlinear Systems: The Particle Filter's feedback scheme exhibits the highest performance among the three schemes, reaffirming its ability to effectively manage complex nonlinear dynamics. The feedback scheme, in particular, derives benefits from its capacity to adapt to nonlinearities by representing the state distribution using a set of particles. While the Euler scheme, while less precise, provides an alternative in scenarios with constrained computational resources. PF is most valuable when addressing systems characterized by significant nonlinearities and non-Gaussian noise, scenarios where other filters may struggle to deliver precise estimates.

Filter Selection Based on SDOF Results:

- Kalman Filter: Employ the Kalman Filter in scenarios involving linear systems where Gaussian noise is an appropriate model for uncertainties. Exercise caution when employing it with Euler or Runge-Kutta integration schemes in nonlinear systems, as these solvers introduce nonlinearity, which may lead to suboptimal results.

- Unscented Kalman Filter: Opt for the Unscented Kalman Filter when working with nonlinear systems. The filter's feedback scheme, particularly within the feedback scheme, proves highly effective at accurately capturing nonlinear dynamics. It

strikes a balance between accuracy and computational efficiency, rendering it suitable for moderately nonlinear systems.

- Particle Filter: Select the Particle Filter for systems characterized by significant nonlinearities and non-Gaussian noise. The filter's resampling scheme excels in capturing complex dynamics, offering the most precise estimates.

Interpreting the Results for MDOF with Random Input:

1. Kalman Filter (KF) for Linear Systems: Higher RMS errors observed in KF's feedback scheme for both masses can be attributed to its linear assumptions and the simplicity of the prediction section. It performs reasonably well with the Runge Kutta scheme. The Euler and Runge Kutta schemes yield superior performance due to their integration process.
2. Unscented Kalman Filter (UKF) for Nonlinear Systems: UKF's feedback scheme shows lower RMS errors compared to the Euler and Runge Kutta schemes for both masses. This outcome suggests that the feedback scheme is enhanced in the 2DOF system dealing with nonlinearities, thanks to its adaptive nature.
3. Particle Filter (PF) for Nonlinear Systems: PF's feedback scheme achieves the lowest RMS errors, showcasing its ability to handle the introduced nonlinearity effectively. By employing particles to represent state distributions, the PF excels in capturing complex dynamics and exhibits very close accuracy to Runge Kutta in this scenario.

Interpreting the Results for MDOF with Sweep Input:

1. Kalman Filter (KF) for Linear Systems: KF's feedback scheme demonstrates better accuracy for the 2nd mass under sweep input conditions. However, for the 1st mass, it exhibits lower accuracy compared to Euler and Runge Kutta schemes. This discrepancy could be attributed to the nature of the Sweep input, which challenges normal filters. Additionally, differences in the integration process for force input to the 1st mass may contribute to the observed variations.
2. Unscented Kalman Filter (UKF) for Nonlinear Systems: UKF's feedback scheme maintains balanced accuracy for both masses, outperforming Euler and Runge Kutta solvers. Its ability to adapt and update itself in each iteration makes it the preferred choice for handling nonlinearities.
3. Particle Filter (PF) for Nonlinear Systems: PF's feedback scheme consistently performs well for both masses, regardless of input type. This highlights its

robustness in managing nonlinearities and its capacity to provide accurate state estimates, even under different input conditions.

Filter Selection Based on MDOF Results:

- Kalman Filter: In MDOF cases with linear systems, employing the Runge Kutta solver is advisable, as it performs reasonably well under various input types.
- Unscented Kalman Filter (UKF): Optimal for the UKF Feedback scheme when dealing with nonlinearities, irrespective of the input type. Its feedback nature excels in capturing nonlinear dynamics, making it suitable for moderately nonlinear systems.
- Particle Filter (PF): Choose the PF when confronting strong nonlinearities, especially when other filters face challenges. PF's feedback scheme offers exceptional accuracy in highly nonlinear scenarios. Additionally, the Runge Kutta scheme for PF serves as a viable option for addressing strong nonlinearities in the system.

References

Agamennoni, G., Nieto, J., and Nebot, E. 2011. An outlier-robust Kalman filter. Pages 1551–1558 of: IEEE International Conference on Robotics and Automation (ICRA).

Anastasio, D., et al. "Nonlinear dynamics of a duffing-like negative stiffness oscillator: modeling and experimental characterization." Shock and Vibration 2020 (2020).

Anastasio, Dario, and Stefano Marchesiello. "Experimental characterization of friction in a negative stiffness nonlinear oscillator." Vibration 3.2 (2020): 132-148.

Bar-Shalom, Y. and Li, X.-R. 1995. Multitarget-Multisensor Tracking: Principles and Techniques. YBS Publishing.

Bar-Shalom, Y., Li, X.-R., and Kirubarajan, T. 2001. Estimation with Applications to Tracking and Navigation. Wiley.

Bernardo, José M., and Adrian FM Smith. Bayesian theory. Vol. 405. John Wiley & Sons, 2009.

Blackman, S. and Popoli, R. 1999. Design and Analysis of Modern Tracking Systems. Artech House Radar Library.

Brooks, S., Gelman, A., Jones, G. L., and Meng, X.-L. 2011. Handbook of Markov Chain Monte Carlo. Chapman & Hall/CRC.

Challa, S., Morelande, M. R., Musicki, D., and Evans, R. J. 2011. ˇ Fundamentals of Object Tracking. Cambridge University Press.

Chatzi, Eleni N., and Andrew W. Smyth. "The unscented Kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing." Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures 16.1 (2009): 99-123.

Chatzi, Eleni N., Andrew W. Smyth, and Sami F. Masri. "Experimental application of on-line parametric identification for nonlinear hysteretic systems with model uncertainty." Structural Safety 32.5 (2010): 326-337.

Crassidis, J. L. and Junkins, J. L. 2004. Optimal Estimation of Dynamic Systems. Chapman & Hall/CRC.

Daum, F. and Huang, J. 2003. Curse of dimensionality and particle filters. Pages 1979– 1993 of: Proceedings of the IEEE Aerospace Conference, vol. 4.

Deisenroth, M., Turner, R., Huber, M., Hanebeck, U., and Rasmussen, C. 2012. Robust filtering and smoothing with Gaussian processes. IEEE Transactions on Automatic Control, 57(7), 1865–1871.

Djuric, Petar M., et al. "Particle filtering." IEEE signal processing magazine 20.5 (2003): 19-38.

Doucet, A., De Freitas, N., and Gordon, N. 2001. Sequential Monte Carlo Methods in Practice. Springer.

Fuchs, Armin. Nonlinear dynamics in complex systems. Berlin: Springer, 2014.

Gelman, Andrew. "Parameterization and Bayesian modeling." Journal of the American Statistical Association 99.466 (2004): 537-545.

Gilks, W., Richardson, S., and Spiegelhalter, D. (eds.) 1996. Markov Chain Monte Carlo in Practice. Chapman & Hall.

Godsill, S. J. and Rayner, P. J. 1998. Digital Audio Restoration: a Statistical Model Based Approach. Springer-Verlag.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. Pages 107–113 of: IEEE Proceedings on Radar and Signal Processing, vol. 140.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. Pages 107–113 of: IEEE Proceedings on Radar and Signal Processing, vol. 140.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. Pages 107–113 of: IEEE Proceedings on Radar and Signal Processing, vol. 140.

Grewal, M. S. and Andrews, A. P. 2001. Kalman Filtering, Theory and Practice Using MATLAB. Wiley.

Hairer, Ernst, Christian Lubich, and Michel Roche. The numerical solution of differential-algebraic systems by Runge-Kutta methods. Vol. 1409. Springer, 2006.

Hartikainen, Jouni, Arno Solin, and Simo Särkkä. "Optimal filtering with Kalman filters and smoothers." Department of biomedica engineering and computational sciences, Aalto University School of Science, 16th August (2011).

Hauk, O. 2004. Keep it simple: a case for using classical minimum norm estimation in the analysis of EEG and MEG data. NeuroImage, 21(4), 1612–1621.

Hayes, M. H. 1996. Statistical Digital Signal Processing and Modeling. John Wiley & Sons, Inc.

Ho, Y. C. and Lee, R. C. K. 1964. A Bayesian approach to problems in stochastic estimation and control. IEEE Transactions on Automatic Control, 9(4), 333–339.

Jazwinski, A. H. 1966. Filtering for nonlinear dynamical systems. IEEE Transactions on Automatic Control, 11(4), 765–766.

Jazwinski, A. H. 1970. Stochastic Processes and Filtering Theory. Academic Press.

Julier, S. J. and Uhlmann, J. K. 2004. Unscented filtering and nonlinear estimation. Proceedings of the IEEE, 92(3), 401–422.

Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F. 1995. A new approach for filtering nonlinear systems. Pages 1628–1632 of: Proceedings of the 1995 American Control, Conference, Seattle, Washington.

Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F. 2000. A new method for the nonlinear transformation of means and covariances in filters and estimators. IEEE Transactions on Automatic Control, 45(3), 477–482.

Kaipio, J. and Somersalo, E. 2005. Statistical and Computational Inverse Problems. Applied Mathematical Sciences, no. 160. Springer.

Kalman, R. E. 1960b. A new approach to linear filtering and prediction problems. Transactions of the ASME, Journal of Basic Engineering, 82(1), 35–45.

Kalman, R. E. and Bucy, R. S. 1961. New results in linear filtering and prediction theory. Transactions of the ASME, Journal of Basic Engineering, 83(3), 95–108.

Keeling, M. and Rohani, P. 2007. Modeling Infectious Diseases in Humans and Animals. Princeton University Press.

Kioutsioukis, Ioannis, et al. "Uncertainty and global sensitivity analysis of road transport emission estimates." Atmospheric Environment 38.38 (2004): 6609-6620.

Kitagawa, G. 1987. Non-Gaussian state-space modeling of nonstationary time series. Journal of the American Statistical Association, 82(400), 1032–1041.

Kitagawa, G. 1996. Monte Carlo filter and smoother for Non-Gaussian nonlinear state space models. Journal of Computational and Graphical Statistics, 5(1), 1–25.

Kitagawa, G. 1996. Monte Carlo filter and smoother for Non-Gaussian nonlinear state space models. Journal of Computational and Graphical Statistics, 5(1), 1–25.

Kuether, R. J., Renson, L., Detroux, T., Grappasonni, C., Kerschen, G., & Allen, M. S. (2015). Nonlinear normal modes, modal interactions and isolated resonance curves. Journal of Sound and Vibration, 351, 299-310.

Landa, Vlad, et al. "GoIn-An Accurate 3D InDoor Navigation Framework for Mobile Devices." 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN). IEEE, 2018.

Lee, R. C. K. 1964. Optimal Estimation, Identification and Control. MIT Press.

Liu, J. S. 2001. Monte Carlo Strategies in Scientific Computing. Springer.

Liu, J. S. and Chen, R. 1995. Blind deconvolution via sequential imputations. Journal of the American Statistical Association, 90(430), 567–576.

Marchesiello, Stefano, Anastasio, D. "Nonlinear system identification of a double-well Duffing oscillator with position-dependent friction." Nonlinear Dynamics 108.4 (2022): 2993-3008.

Marchesiello, Stefano, and Luigi Garibaldi. "A time domain approach for identifying nonlinear vibrating structures by subspace methods." Mechanical Systems and Signal Processing 22.1 (2008): 81-101.

Maybeck, Peter S. Stochastic models, estimation, and control. Academic press, 1982.

Milton, J. S. and Arnold, J. C. 1995. Introduction to Probability and Statistics, Principles and Applications for Engineering and the Computing Sciences. McGraw-Hill.

Murray, J. D. 1993. Mathematical Biology. Springer.

Øksendal, Bernt, and Bernt Øksendal. Stochastic differential equations. Springer Berlin Heidelberg, 2003.

Proakis, J. G. 2001. Digital Communications. Fourth edn. McGraw-Hill.

Punskaya, E., Doucet, A., and Fitzgerald, W. J. 2002. On the use and misuse of particle filtering in digital communications. In: Proceedings of EUSIPCO.

Rasmussen, Carl Edward, and Christopher KI Williams. Gaussian processes for machine learning. Vol. 1. Cambridge, MA: MIT press, 2006.

Renson, Ludovic, et al. "Force appropriation of nonlinear structures." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474.2214 (2018): 20170880.

Ristic, B., Arulampalam, S., and Gordon, N. 2004. Beyond the Kalman Filter. Artech House.

Saltelli, Andrea, et al. Sensitivity analysis in practice: a guide to assessing scientific models. Vol. 1. New York: Wiley, 2004.

Sarkk ¨ a, S., Bunch, P., and Godsill, S. J. 2012a. A backward-simulation based Rao- ¨ Blackwellized particle smoother for conditionally linear Gaussian models. Pages 506–511 of: Proceedings of SYSID 2012.

Sarkk ¨ a, S., Vehtari, A., and Lampinen, J. 2007b. Rao-Blackwellized particle filter for ¨ multiple target tracking. Information Fusion Journal, 8(1), 2–15.

Schön, Thomas B. Estimation of nonlinear dynamic systems: Theory and applications. Diss. Institutionen för systemteknik, 2006.

Smart, Benjamin, et al. "Model predictive control of cancer cellular dynamics: a new strategy for therapy design." Frontiers in Control Engineering 3 (2022): 935018.

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. 2008. Obstacles to highdimensional particle filtering. Monthly Weather Review, 136(12), 4629–4640.

Song, M., Renson, L., Moaveni, B., & Kerschen, G. (2022). Bayesian model updating and class selection of a wing-engine structure with nonlinear connections using nonlinear normal modes. Mechanical Systems and Signal Processing, 165, 108337.

Song, M., Renson, L., Noël, J. P., Moaveni, B., & Kerschen, G. (2018). Bayesian model updating of nonlinear systems using nonlinear normal modes. Structural Control and Health Monitoring, 25(12), e2258.

Stengel, Robert F. Optimal control and estimation. Courier Corporation, 1994.

Stone, L. D., Barlow, C. A., and Corwin, T. L. 1999. Bayesian Multiple Target Tracking. Artech House.

Stratonovich, R. L. 1968. Conditional Markov Processes and Their Application to the Theory of Optimal Control. Elsevier.

Tarantola, A. 2004. Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM.

Van Trees, H. L. 1968. Detection, Estimation, and Modulation Theory Part I. John Wiley & Sons.

Verdult, Vincent. "Non linear system identification: a state-space approach." (2002).

Wan, E. A. and Van der Merwe, R. 2001. The unscented Kalman filter. Chapter 7 of: Haykin, S. (ed.), Kalman Filtering and Neural Networks. Wiley.

Wiener, N. 1950. Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications. John Wiley & Sons.