

Simultaneous Estimation and Modeling of State-Space Systems Using Multi-Gaussian Belief Fusion

John Josiah Steckenrider

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State
University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Mechanical Engineering

Tomonari Furukawa
A. Lynn Abbott
Robert G. Parker
Steve C. Southward

03/23/2020
Blacksburg, Virginia

Keywords: Belief fusion, robust estimation, nonlinear systems, discrete-time state-space
models, Kalman Filters, model mismatch

Simultaneous Estimation and Modeling of State-Space Systems Using Multi-Gaussian Belief Fusion

John Josiah Steckenrider

ABSTRACT

This work describes a framework for simultaneous estimation and modeling (SEAM) of dynamic systems using non-Gaussian belief fusion by first presenting the relevant fundamental formulations, then building upon these formulations incrementally towards a more general and ubiquitous framework. Multi-Gaussian belief fusion (MBF) is introduced as a natural and effective method of fusing non-Gaussian probability distribution functions (PDFs) in arbitrary dimensions efficiently and with no loss of accuracy. Construction of some multi-Gaussian structures for potential use in MBF is addressed. Furthermore, recursive Bayesian estimation (RBE) is developed for linearized systems with uncertainty in model parameters, and a rudimentary motion model correction stage is introduced. A subsequent improvement to motion model correction for arbitrarily non-Gaussian belief is developed, followed by application to observation models. Finally, SEAM is generalized to fully nonlinear and non-Gaussian systems. Several parametric studies were performed on simulated experiments in order to assess the various dependencies of the SEAM framework and validate its effectiveness in both estimation and modeling. The results of these studies show that SEAM is capable of improving estimation when uncertainty is present in motion and observation models as compared to existing methods. Furthermore, uncertainty in model parameters is consistently reduced as these parameters are updated throughout the estimation process. SEAM and its constituents have potential uses in robotics, target tracking and localization, state estimation, and more.

Simultaneous Estimation and Modeling of State-Space Systems Using Multi-Gaussian Belief Fusion

John Josiah Steckenrider

GENERAL AUDIENCE ABSTRACT

The simultaneous estimation and modeling (SEAM) framework and its constituents described in this dissertation aim to improve estimation of signals where significant uncertainty would normally introduce error. Such signals could be electrical (e.g. voltages, currents, etc.), mechanical (e.g. accelerations, forces, etc.), or the like. Estimation is accomplished by addressing the problem probabilistically through information fusion. The proposed techniques not only improve state estimation, but also effectively “learn” about the system of interest in order to further refine estimation. Potential uses of such methods could be found in search-and-rescue robotics, robust control algorithms, and the like. The proposed framework is well-suited for any context where traditional estimation methods have difficulty handling heightened uncertainty.

To my heavenly fathers, my wife, my mom, and my family

“For am I now seeking the approval of man, or of God? Or am I trying to please man? If I were still trying to please man, I would not be a servant of Christ.”

Galatians 1:10

Special thanks to the Virginia Tech Department of Mechanical Engineering and the people of the Computational Multiphysics Systems Laboratory. Thanks also to Murata Manufacturing Co., Ltd. for supporting my second and third years of graduate study.

TABLE OF CONTENTS

CHAPTER 1. Introduction	1
1.1 Development and Motivation	1
1.2 Primary Outcome: Simultaneous Estimation and Modeling	3
CHAPTER 2. Multi-Gaussian Belief Fusion.....	5
2.1 Introduction	5
2.1.1 Motivation	5
2.1.2 Background and Related Work.....	5
2.1.3 Objectives and Outline	7
2.2 Foundational Concepts.....	8
2.2.1 Belief: Gaussian vs. Non-Gaussian PDFs	8
2.2.2 Belief Fusion.....	9
2.2.3 Product of Gaussians.....	10
2.3 Multi-Gaussian Belief Fusion	11
2.3.1 Linear Superposition of Gaussian Products	11
2.3.2 Construction of Non-Gaussian Structures.....	15
2.4.1 Belief Fusion and Decision-Making Examples	27
2.4.2 Gaussian Component Pruning	30
2.4.3 Efficiency vs. Dimensionality	32
2.4.3 Efficiency/Accuracy Tradeoff for Multiple MBF Iterations	35
2.5 Applications	37
2.5.1 Dynamics: State Variable Estimation	37
2.5.3 Automation: Classification.....	39
2.6 Conclusion.....	40
2.7 References	41
CHAPTER 3. The Gaussian Toroid as a Prediction Model	45
3.1 Introduction	45
3.1.1 Background.....	45
3.1.2 Related Work.....	46
3.2 Prediction, Belief Fusion, and Decision-Making	47
3.2.1 Prediction.....	47
3.2.2 Gaussian Toroid Model.....	48
3.2.3 Belief Fusion.....	49

3.2.4 Decision-Making	50
3.3 Recursive Bayesian Classification.....	51
3.3.1 Toroidal Prediction Model	52
3.3.2 Correction and Classification	56
3.4 Results.....	58
3.5 Conclusions and Applications	62
3.6 References	62
3.7 Appendix	64
CHAPTER 4. Introduction to Simultaneous Estimation and Modeling	67
4.1 Introduction	67
4.1.1 Background.....	67
4.1.2 Related Work.....	67
4.1.3 Objectives and Outline	69
4.2 Probabilistic Motion Tracking.....	70
4.2.1 Robotic Belief.....	70
4.2.2 Prediction.....	70
4.2.3 Correction	71
4.2.4 Adaptive Kalman Filter	72
4.3 Simultaneous Estimation and Modeling	72
4.3.1 Overview	72
4.3.2 Model Uncertainty Estimation.....	73
4.3.3 State Matrix Updating	75
4.4 Results and Validation	78
4.4.1 Simulated Experiment Description	78
4.4.2 Initial-Value (Unforced Response) Comparisons.....	80
4.4.3 Forced Response Comparisons.....	83
4.5 Conclusions and Future Work.....	85
4.6 References	86
4.7 Appendix.....	89
CHAPTER 5. Probabilistic Non-Gaussian Motion Model Correction	96
5.1 Introduction	96
5.1.1 Background.....	96
5.1.2 Related Work.....	96
5.1.3 Objectives and Outline	98

5.2 Belief, Estimation, and Model Correction	98
5.2.1 Recursive Bayesian Estimation	99
5.2.2 Kalman-type Estimation.....	100
5.2.3 Multi-Gaussian Belief Fusion.....	102
5.2.4 Simultaneous Estimation and Modeling	103
5.3 Multi-Gaussian Model Correction.....	106
5.3.1 Single State Transition Error Minimization	106
5.3.2 Multiple State Transition Mean Error Minimization	111
5.3.3 Uncertainty Propagation.....	112
5.4 Results.....	114
5.4.1 1-D Validation	114
5.4.2 2-D Validation	122
5.5 Conclusions, Applications, and Future Work	126
5.6 References	127
CHAPTER 6. Probabilistic Non-Gaussian Sensor Model Correction	131
6.1 Introduction	131
6.1.1 Background.....	131
6.1.2 Related Work.....	131
6.1.3 Objectives and Outline	133
6.2 Simultaneous Estimation and Modeling	134
6.2.1 Recursive Bayesian Estimation	134
6.2.2 Kalman Estimation.....	137
6.2.3 Multi-Gaussian Motion Model Correction.....	139
6.3 Multi-Gaussian Observation Model Correction	140
6.3.1 Observation Formulations	140
6.3.2 Single Re-observation Error Minimization	145
6.3.3 Multiple Re-observation Error Minimization.....	149
6.4 Results.....	150
6.4.1 1-D Validation	151
6.4.2 2-D Validation	158
6.5 Conclusions and Future Work.....	161
6.6 References	162
CHAPTER 7. Generalized Simultaneous Estimation and Modeling.....	166
7.1 Introduction	166

7.1.1 Background.....	166
7.1.2 Related Work.....	166
7.1.3 Objectives and Outline	169
7.2 Estimation and Motion Model Correction	169
7.2.1 Recursive Bayesian Estimation	169
7.2.2 Kalman Estimation.....	171
7.2.3 Linear Motion Model Correction.....	173
7.3 Nonlinear Estimation and Motion Model Correction	175
7.3.1 Nonlinear Estimation with Extended Uncertainty Propagation	175
7.3.2 Nonlinear Motion Model Correction	176
7.3.3 Representative Nonlinear System Derivation	181
7.4 Results.....	184
7.4.1 Description of Monte-Carlo Simulations	184
7.4.2 Qualitative Assessment	186
7.4.3 Quantitative Assessment	189
7.5 Conclusions and Future Work.....	192
7.6 References	193
CHAPTER 8. Framework Unification and Applications	197
8.1 Framework Unification.....	197
8.2 Applications	201
8.2.1 Signal Processing and State Estimation	201
8.2.2 Robotics, Localization, and Target Tracking	204
8.2.3 Conclusion.....	206

LIST OF FIGURES

Chapter 2:

Figure 1. a) 1-D Gaussian PDF, b) 1-D non-Gaussian PDF example.	9
Figure 2. Belief fusion in 1D with unimodal Gaussian PDFs.	10
Figure 3. MBF for two multi-Gaussian PDFs.	13
Figure 4. Recursive MBF for fusion of multiple belief sources.	14
Figure 5. Construction of a 2-D Gaussian toroid as the difference between two Gaussian distributions.	16
Figure 6. A 1-D Gaussian toroid.	17
Figure 7. Cross-sectional characteristics of 2-D Gaussian line.	20
Figure 8. Profiles of Gaussian sum with equal variance and constant spacing. Colors serve only to show each PDF distinctly.	20
Figure 9. Flatness as a function of the ratio between spacing and standard deviation of distributions.	21
Figure 10. Four examples of Gaussian lines with varying attributes.	24
Figure 11. MLE/decision boundary classification.	24
Figure 12. Decision-making by integration of the joint probability of training PDFs with the output of MBF.	25
Figure 13. Three training classes and their decision boundaries.	27
Figure 14. Eight examples of MBF. A: two arbitrary Gaussian mixtures (AGM), B: AGM and Gaussian toroid (GT), C: two GT, D: three GT, E: AGM and Gaussian line (GL), F: GL and GT, G: two GL, H: AGM, GT, and GL. Separated columns show result of decision-making fusion corresponding to three training classes shown in Fig. 13. Class probabilities are directly proportional to the volumes beneath the fused decision-making distributions.	28
Figure 15. Information entropy for three classifiers across eight examples of post-belief fusion decision-making.	29
Figure 16. Effect of pruning on examples A, B, E, F, G, and H with accompanying bar graphs of constituent weighting coefficients. On average, over half the components can be removed with little overall error.	30
Figure 17. Plots of output class probabilities vs. number of Gaussian components for examples A-H. Red = class 1, green = class 2, and blue = class 3.	31

Figure 18. Computation time as a function of dimensionality for uniform resolution of 10 grid cells per dimension.34

Figure 19. RMSE of grid cells in 200-by-200 belief space for example E under two methods.35

Figure 20. Eight miscellaneous multi-Gaussian PDFs fused in consecutive iterations of MBF. From upper-left to lower-right, the PDFs contain 4, 4, 3, 3, 7, 11, 5, and 4 components. The final PDF, therefore, contains 221,760 components.36

Figure 21. Timing and error as functions of the number of fusion events for various resolutions and pruning laws.36

Figure 22. Time domain plot, state-space plot, and state-space PDF of decaying sinusoid of form $C \exp(-\frac{t}{\tau}) \sin(\omega t)$ where $C = 25$, $\tau = 10$, and $\omega = 1$37

Figure 23. Joint PDF of $y(t)$ and $\dot{y}(t)$38

Figure 24. Examples of four potential state-space PDFs where the states are the associated function and its derivative. Note how the diverging first and fourth examples are unbounded in the state space in keeping with their instability.38

Figure 25. Allowable path-based non-Gaussian PDF superimposed on floorplan.39

Chapter 3:

Figure 1. Visual construction of a 2-D Gaussian toroid [15].48

Figure 2. Recursive Bayesian classification diagram. Conceptual elements specifically related to this work's contribution are shaded.52

Figure 3. Prediction from multi-Gaussian prior belief. a) Single Gaussian of the form given in (19), b) Gaussian toroid defined by (20), c) output of toroidal prediction as governed by (25)-(28).55

Figure 4. Correction as belief fusion of $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k | \mathbf{z}_k)$56

Figure 5. Prediction in the second iteration of RBC for running example.57

Figure 6. Example simulations without and with directional bias in the evolution model. Blue dots indicate training class centers, black lines correspond to quadratic decision boundaries, green arrows are ground truth trajectories, and red arrows are noisy observations of ground truth.58

Figure 7. Convergence of outcomes corresponding to parameters given by the second primary column and third row of Table 1.60

Figure A3.1. Recursive Bayesian classification using convolution in the prediction step.66

Chapter 4:

Figure 1. SEAM diagram. Shaded blocks designate original contributions detailed in section 4.3.	73
Figure 2. Mapping of 1-D MLP from step $k - 1$ to k	75
Figure 3. MSD model for validation simulation.	79
Figure 4. State-space and error plots of simulation where $\frac{\sigma}{\mu}$ and Σ_v correspond to upper-left-most combination in Table 2.	82
Figure 5. State-space and error plots of simulation where $\frac{\sigma}{\mu}$ and Σ_v correspond to lower-right-most combination in Table 2.	82
Figure 6. State matrix elements' adjustments due to model correction.	83
Figure 7. Simulation where $\frac{\sigma}{\mu} = 5\%$, $\Sigma_v = \begin{bmatrix} 0.1^2 & 0.04^2 \\ 0.04^2 & 0.08^2 \end{bmatrix}$, and $\Sigma_u = \begin{bmatrix} 5.4^2 & 2.9^2 \\ 2.9^2 & 4.6^2 \end{bmatrix}$	83

Chapter 5:

Figure 1. SEAM framework.	104
Figure 2. Proposed generalized SEAM with motion model correction employing non-Gaussian state belief.	106
Figure 3. Example 1-D simulation with single-transition updating. In this example, $A_0 = 1.47$ and $B_0 = -1.95$	115
Figure 4. State and input matrices over time, for experiment corresponding to previous figure.	116
Figure 5. Convergence of state matrix estimate for varying initial guesses; corresponding error curves for each estimator.	119
Figure 6. Example 1-D simulation with model-updating using the 25 most recent observed state transitions.	120
Figure 7. State and input parameters plotted over time.	120
Figure 8. 1-D non-Gaussian state matrix updating for normally distributed random initial estimates; a single iteration under the estimation framework.	121
Figure 9. Sample simulation for 2-D uni-Gaussian validation.	124
Figure 10. Convergence of model parameters for 2-D uni-Gaussian validation.	124

Chapter 6:

Figure 1. System diagram (SEAM with observation model correction).	140
Figure 2. Gaussian and multi-Gaussian observation models where $\bar{C} = 2$. This arbitrary multi-Gaussian model for which $c_j^{(k \hat{k})} = 1/6 \forall j$ and $\delta_j = [0.8, -0.8, 1.2, -1.2, 2, -2]$ displays bimodal characteristics.	142
Figure 3. a) Belief through multiple stages of SEAM represented as 1-D Gaussian PDFs; b) The objective function ISE_h plotted over a range of observation model parameters C	149
Figure 4. State and error of noisy observations and estimated signals over time.	152
Figure 5. Observation model correction over time.	152
Figure 6. Estimation using SEAM ⁺ improves over time, while other estimators generally yield constant error.	154
Figure 7. Ensemble curves demonstrating the convergence of observation model parameters for several initial estimates of C . Note the failure of six simulations to converge C in both plots due to particularly poor initial estimates.	155
Figure 8. PDFs of different stages of belief for one time step. The first plot models multi-Gaussian observations using $c_j^{(k \hat{k})} = [0.6, 0.1, 0.3]$ and $\delta_j = [-2.5, 0, 2.5]$, while the second plot uses using $c_j^{(k \hat{k})} = [1/3, 1/3, 1/3]$ and $\delta_j = [-1.5, 0, 1.5]$. Both use $C_{GT} \sim \mathcal{N}(1, 0.1^2)$ in this example.	155
Figure 9. Non-Gaussian observation model correction over time.	156
Figure 10. Dependency of model correction on model inaccuracy and non-Gaussian severity.	157
Figure 11. 2-D time domain signal and error plots.	159
Figure 12. 2-D observation model matrix and ISE_h over time.	160
Figure 13. Error plots representing data acquired over 100 stochastic 2-D simulations. ..	161
Chapter 7:	
Figure 1. Nonlinear SEAM framework.	175
Figure 2. Pendulum FBD.	181
Figure 3. Time domain plots.	186
Figure 4. Model correction over time.	187
Figure 5. State estimation error and ISE for sequential time steps corresponding to estimates of Φ	188

Chapter 8:

Figure 1. Unified SEAM diagram.	198
Figure 2. Insufficiency of simultaneous predictor-observer correction.	199
Figure 3. Multi-sensor fusion.	200
Figure 4. RC circuit.	202
Figure 5. MSD oscillator.	202
Figure 6. RLC circuit.	203
Figure 7. Nonlinear pendulum.	203
Figure 8. Gravitational kinematic path.	203
Figure 9. Differential drive mechanism.	204
Figure 10. A lemniscate.	205

LIST OF TABLES

Chapter 2:

Table 1. Class probabilities. Bold numbers denote winning classes; highlighted columns correspond to proposed approach.	28
--	----

Chapter 3:

Table 1. Parametric Study Results	60
Table 2. Timing Study Results (milliseconds)	61

Chapter 4:

Table 1. Constant parameters	80
Table 2. Percentage of wins in 5,000 simulations for the Kalman filter, SEAM without model-updating, and SEAM with model-updating across varied sensor and model uncertainty parameters	81
Table 3. Percentage of wins in 500 simulations for LKF, AKF, and SEAM estimators across varied input and model uncertainty parameters	84

Chapter 5:

Table 1. Constant parameters	115
Table 2. Sensitivity study in uncertainty parameters for 1-D Gaussian estimation	118
Table 3. Constant parameters	123
Table 4. Percent errors in state matrix elements for 100 trials	125
Table 5. Median root-mean-squared-errors and percent wins over 100 simulations for all estimators	125

Chapter 6:

Table 1. Constant parameters (1-D validation)	151
Table 2. Parametric sensitivity study for 1-D Gaussian estimation	153
Table 3. Constant parameters	159
Table 4. Percent errors in observation model parameters for 100 trials	160

Chapter 7:

Table 1. Ground-truth parameters	185
Table 2. Parameter uncertainties	185
Table 3. Qualitative example parameter values	186
Table 4. Root-mean-squared-errors (RMSEs) for each signal	188
Table 5. Percent of wins in 50 trials for each estimator. E = EKF, A = AEKF, S- = SEAM, and S+ = SEAM ⁺	190
Table 6. Average ISEs given by Φ_0 , Φ_G , and Φ_k	191
Table 7. Average percent of time model parameters were improved	191

LIST OF ABBREVIATIONS

AKF – Adaptive Kalman Filter
EKF – Extended Kalman Filter
EM – Expectation Maximization
ISE – Integrated-Squared-Error
KF – Kalman Filter
MBF – Multi-Gaussian Belief Fusion
MLE – Maximum Likelihood Estimation
MLP – Maximum Likelihood Point
PDF – Probability Distribution Function
PDM – Probabilistic Decision-Making
PHD – Probability Hypothesis Density
QBC – Quadratic Bayesian Classifier
RBC – Recursive Bayesian Classification
RBE – Recursive Bayesian Estimation
RMSE – Root-Mean-Squared-Error
SEAM – Simultaneous Estimation and Modeling
SLAM – Simultaneous Localization and Mapping

PREFACE

All papers contained within this manuscript-style dissertation were written by myself, Josiah Steckenrider, and reflect my original ideas, work, and formulations. My academic advisor, Dr. Tomonari Furukawa, kindly aided in the revision and reordering of these concepts as they are communicated to the reader. Information about these papers is given below in the order of their appearance as chapters within the text:

2. “Multi-Dimensional Belief Fusion of Multi-Gaussian Structures” was published in the journal *Information Fusion* in December of 2019.
3. “Recursive Bayesian Classification for Perception of Evolving Targets using a Gaussian Toroid Prediction Model” was published in the proceedings of the 2019 IEEE Conference on Robotics and Automation (ICRA).
4. “A Probabilistic Model-Adaptive Approach for Tracking of Motion with Heightened Uncertainty” is, at the time of publication of this dissertation, accepted in *The International Journal of Control, Automation, and Systems*.
5. “Probabilistic Motion Model Correction Using Non-Gaussian Belief Fusion” is, at the time of publication of this dissertation, under revision in *Automatica*.
6. “Probabilistic Observation Model Correction Using Non-Gaussian Belief Fusion” is, at the time of publication of this dissertation, under review in *Information Fusion*.
7. “Simultaneous Estimation And Modeling of Nonlinear State-Space Systems” is, at the time of publication of this dissertation, pre-submission in *Automatica*.

CHAPTER 1. Introduction

1.1 Development and Motivation

The various concepts, formulations, and ultimately overall framework discussed in this dissertation began with an examination of belief fusion of Gaussian probability distribution functions (PDFs). Because belief fusion can be accomplished at the PDF-level by multiplying and normalizing the PDFs representing belief coming from different sources, fusion of two Gaussians has a special property in that it yields another Gaussian. Though this is an elegant result, it is restricted to Gaussian belief only. However, because methods exist for fitting Gaussian sums to estimate non-Gaussian PDFs, it became evident that belief fusion of multi-Gaussian PDFs would have a similarly elegant solution by means of linear superposition. These premises laid the foundation of multi-Gaussian belief fusion (MBF).

Given the ubiquity of MBF, it became natural to investigate how to efficiently generate a set of non-(multi-)Gaussian PDFs with useful characteristics. The formulation of two of these so-called “multi-dimensional multi-Gaussian structures” was included in the work relevant to chapter 2. Upon examining potential uses of these structures in real-world applications, there arose a question of accurately accounting for uncertainty in dynamic system modeling. At this juncture, the use of MBF in the correction stage of recursive Bayesian estimation (RBE) was warranted, but formulation of multi-Gaussian prediction had not been addressed. To this end, non-Gaussian prediction for a linearized system was developed with allowance for uncertainty in not only states, but also model parameters and inputs. Because this allowance increases sensitivity to observational noise in the estimation process, a logical next step was to consider how uncertainty in a system model might be continuously reduced.

The rudimentary solution to model-updating proposed in chapter 4 makes the assumption that the mean and covariance of an arbitrary state of belief (whether Gaussian or non-Gaussian) could be readily extracted and used to update the state matrix of a linearized system. This results in an augmented least-squares solution for improving estimates of the elements of the state matrix based on previous estimates of both system parameters and states. An additional outcome of this approach to model-updating is a series of equations for updating the variances in system parameters which are then used to re-initialize another iteration of estimation. Though this approach tends to improve model estimates and therefore produce more accurate estimation, it does not fully incorporate all available probabilistic information. For this reason, ground-truth parameter values are not consistently converged upon.

To improve the model-updating step for multi-dimensional and multi-Gaussian belief, an optimization approach was taken. Rather than pursue a closed-form solution for updating state matrix elements, a state transition error minimization approach was considered. For the examples given in chapter 5, this approach gives parameter estimates which more reliably converge upon the ground-truth values. However, these formulations still make the assumption that a system is linear or linearized. Furthermore, model-updating up to this stage addressed only the system-intrinsic state matrix of a linear system without regard for the input matrix. Consequently, a more general framework was formulated to comprehensively define non-Gaussian simultaneous estimation and modeling (SEAM) for arbitrarily nonlinear systems.

In order to implement some of the tools developed here for special estimation contexts where motion models are unobtainable, chapter 3 addresses recursive Bayesian classification (RBC) using the Gaussian toroid as a prediction model. This chapter specifically addresses probabilistic classification of stochastically evolving targets in a high-dimensional feature space.

Though not addressed in this dissertation, an outcome of this approach to classification led to real-world improvement in road crack classification, a platform which fits the proposed framework well. The information introduced by even a rudimentary prediction model such as the Gaussian toroid yields better estimation and probabilistic classification with more accurate uncertainty characterization for problems where such an approach is appropriate.

1.2 Primary Outcome: Simultaneous Estimation and Modeling

Though the research presented here covers a wide range of applications and its direction branches off into various subdomains at a number of points, the primary outcome of this research is the development of the SEAM framework for estimation and modeling of dynamic systems. Chapter 2 lays the foundation for development of SEAM, chapter 4 introduces it, and chapters 5 and 6 progressively generalize and refine the framework. While the notion of overcoming system model uncertainty in estimation is not new, existing techniques generally do not address estimation and modeling as a recursively symbiotic process reminiscent of an active learning approach. Furthermore, the use of MBF in the correction stage allows belief to be continuously represented by potentially highly multi-modal non-Gaussian PDFs. This could have significant implications for several kinds of robotics tracking problems; such applications are discussed in chapter 7, where a summary of the framework can also be found.

The aim of SEAM as the outcome of this doctoral research is that properly equipped robots and autonomous systems might be able to overcome challenging scenarios where uncertainty is more complex and available information is sparser. In order to interact more naturally with humans in real-world environments, robots need to be capable of perceiving the world and drawing conclusions in a fashion that is more probabilistically similar to the way humans do. Simultaneous estimation and modeling is an active online framework for accomplishing these tasks by means of

prediction, belief fusion, uncertainty propagation, and optimization. The SEAM technique could eventually lead to higher-level characterization of the very *kinds* of targets being pursued and estimated, a valuable extension for autonomy in environments with unknown agents.

CHAPTER 2. Multi-Gaussian Belief Fusion

2.1 Introduction

2.1.1 Motivation

Recent decades have seen an increased emphasis on probabilistic methods in the areas of estimation, automation, information engineering, and the like [1]. Because humans think and act probabilistically, it is important that human-designed systems be similarly equipped. Sensor fusion has emerged as an effective tool to help overcome uncertainty in sensor measurements and belief states, though existing approaches vary widely. With regards to probabilistic information-handling, there is need for generalized representation of non-Gaussian belief and a correspondingly inclusive fusion framework. One example of a scenario which could benefit from such a framework is acoustic target tracking in complex environments, a difficult problem largely because sound signals are noisy and often aliased by surface interactions. In order to address a wide range of scenarios such as this, the work presented here investigates efficient generalized non-Gaussian belief fusion in multiple dimensions and an approach to subsequent probabilistic decision-making.

2.1.2 Background and Related Work

Although the term “sensor fusion” is more widely recognized within the appropriate scientific communities, the more general term “belief fusion” is used here to describe analogous principles, where belief is defined by probability distribution functions (PDFs). Belief fusion describes the joining of multiple estimates, each with a specified uncertainty characteristic, to synthesize an estimate with improved certainty characteristics [1]. This definition can be more broadly applied to scenarios where physical sensors are not required, making it more appropriate for the theoretical formulations presented here. In the traditional sense, fusion has applications in image processing and computer vision [2], IMU/GPS/accelerometer localization [3], [4], radar and

sonar [5], and more. However, this paper avoids direct association with any single application in the effort to clarify the ubiquity of the formulations given.

Belief fusion problems are often formulated in terms of two or more state observations, each described by a mean and uncertainty [6]. Obtaining a fused output mean and uncertainty is approached in a number of ways. The most conventional method assigns the output mean as the weighted average of contributing means, where each mean is weighted by the inverse of its variance. The output uncertainty is then given as the inverse of the sum of the inverse variances. Under the assumptions of Gaussian estimates, this is the direct result of dynamic system smoothing formulations [7], as well as an outcome of the central limit theorem [8]. Furthermore, it can be shown that the updating stage of the traditional Kalman filter also reduces to this result [6], [9]. Other variants arise under the optimal Kalman filter [10], [11], where the output belief bears resemblance to the conventional form with the inclusion of optimized elements. Some approaches to belief fusion employ Bayesian networks [12], [1] to enhance certainty about an estimated state. Furthermore, Dempster-Shafer theory includes a rule for combination of belief, though this framework generally does not employ probability distribution functions [13], [14], [15].

Recursive Bayesian estimation (RBE) is a widely implemented estimation framework which addresses belief fusion in the updating, or correction, stage [16], [17]. It is worth noting that the Kalman filter is one simplification of RBE. In general, though, belief fusion under RBE is addressed from an arbitrary standpoint where the PDFs of contributing estimates are allowed to take any form. One approach to belief fusion in RBE is the grid-based method [18], which discretizes a belief space into sampled grid cells and performs element-wise operations on all cells. Computation time for such non-parametric methods increases exponentially with dimensionality,

and the trade-off is low resolution in the belief space. This poses a problem with regards to accurate, real-time belief fusion.

An additional category of probabilistic filtering towards RBE is the well-known particle (or sequential Monte Carlo) filter [19], [20]. Developed for nonlinear/non-Gaussian applications, the particle filter is predominantly used for contexts in which decisions are made from incomplete, or “hidden”, observations. The particle filter is also used extensively in probability hypothesis density (PHD) filtering, a similar process which requires a form of belief fusion [21]. Although particle filters are capable of handling fusion of non-Gaussian PDFs, there is inherent imprecision in the required weight-changing and resampling processes, and computation demand may be high if many particles are dealt with for improved accuracy.

Sorenson, et al. [22], [23] employed Gaussian sums for belief fusion in RBE. Under this approach, there arises an issue of inflating Gaussian components which was addressed by these authors and, more recently, others [24]. This is also addressed, though to a lesser extent, in the work presented here. Because past work has largely focused on this issue in addition to proving the effectiveness of Gaussian sums in modeling non-Gaussian PDFs, a discussion of general multi-dimensional, multi-Gaussian belief fusion is lacking. More importantly, past work does not address the decision-making aspect of multi-Gaussian representation of belief, a matter which is given substantial attention here.

2.1.3 Objectives and Outline

This paper describes the mathematical formulations developed for multi-dimensional, multi-Gaussian belief fusion (MBF), in addition to original formulations of useful non-Gaussian structures and probabilistic decision-making. MBF is achieved by exploiting the properties of Gaussian multiplication and superposition in N -D. In order to make MBF a natural solution to

real-world information fusion applications, two useful non-Gaussian PDF structures are formulated here. These include toroidal and linear PDFs, where the former leverages the subtraction of two Gaussians with equal mean vectors, and the latter implements a “Gaussian train” with context-specific periodicity. These PDFs efficiently represent structured belief for natural integration into the formulations of MBF. High-level probabilistic decision-making additionally proposed in this paper exploits the properties of multi-Gaussian distributions to integrate joint PDFs for probabilistically modeled training data and give class probabilities with low computational demand. These original contributions have potential applications in a wide range of related scenarios demanding high-speed, multi-dimensional, non-Gaussian belief fusion.

The paper is organized as follows: the next section explains the fundamental concepts of belief and belief fusion as they are used in this paper, which are essential for describing the proposed MBF techniques. Section 3 presents the formulations of MBF and describes two kinds of useful non-Gaussian structures which easily integrate into the framework. Probabilistic decision-making is also addressed for multi-Gaussian belief. Numerical results are shown and analyzed in Section 4, and Section 5 introduces applications of the proposed formulations. Conclusions and ongoing work are summarized in Section 6.

2.2 Foundational Concepts

2.2.1 Belief: Gaussian vs. Non-Gaussian PDFs

While the term “belief” has various meanings and interpretations, some of which are mathematical and others philosophical, it will be defined in this paper exclusively by probability distribution functions (PDFs). A PDF is defined as a function whose integral over a specified boundary gives the probability that an associated random variable falls within that boundary [25]. From the axioms of probability, it follows that any PDF must 1) be nonnegative over the entire

domain (or belief space) and 2) integrate to 1 over the entire domain. The mathematical notation adopted here for a PDF corresponding to a random variable (or more generally, a random vector) \mathbf{X} is $p(\mathbf{x})$, where the argument \mathbf{x} is an indexing variable belonging to the belief space \mathcal{X} .

Many contexts which require a PDF-based representation of belief utilize the Gaussian distribution, shown in Fig. 1a. This particular PDF is useful for two reasons: 1) most random variables with stochastic influence coming from many sources are well-

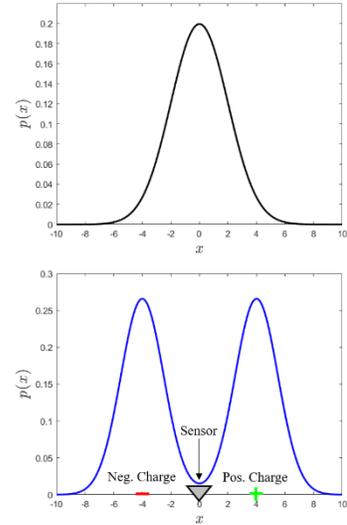


Figure 1. a) 1-D Gaussian PDF, b) 1-D non-Gaussian PDF example.

described by the Gaussian, and 2) the Gaussian can be fully characterized by only two parameters, a mean and a variance. However, some contexts require non-Gaussian representation of belief. Examples can be found across many fields, from image segmentation to acoustic localization to feature-based classification to biostatistics. Though non-Gaussian distributions tend to be fairly context-specific and unstructured, certain structured non-Gaussian PDFs can often be useful. One example is a symmetric-Gaussian distribution which may have use in electromagnetic sensing. A sensed electric field at some point in space may be caused by a positive charge at one location or a negative charge at a polar opposite location. If there is some Gaussian uncertainty in the sensor reading, belief can be represented as shown in Fig. 1b. In this example, the random variable about which belief is represented is the location of some charged particle being sensed. Although this particular example is 1-D, the concept of structured non-Gaussian PDFs scales to N -D.

2.2.2 Belief Fusion

Belief fusion combines knowledge about a particular object from multiple different sources, taking into account the parameters that describe each source's certainty. For the sake of

development, let the vector \mathbf{x} represent the state whose belief is being estimated and \mathbf{z}_i represent the i^{th} observation of \mathbf{x} . The belief that the observation \mathbf{z}_i has correctly estimated \mathbf{x} is given by a conditional PDF $p(\mathbf{x}|\mathbf{z}_i)$. The goal of belief fusion, as it is presented here, is to obtain $p(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n) = p(\mathbf{x}|\mathbf{z}_{1:n})$, the fused PDF that describes belief about \mathbf{x} , given all n available observations. Given that the observations are independent, it can be shown that this PDF is given by:

$$p(\mathbf{x}|\mathbf{z}_{1:n}) = \frac{\prod_{i=1}^n p(\mathbf{x}|\mathbf{z}_i)}{\int_{\mathcal{X}} \prod_{i=1}^n p(\mathbf{x}|\mathbf{z}_i) d\mathbf{x}}, \quad (1)$$

where each observation's PDF $p(\mathbf{x}|\mathbf{z}_i)$ can be estimated by some characterization of the uncertainty inherent in the sensor or source from which the observation came. Note that $p(\mathbf{x}|\mathbf{z}_{1:n})$ is simply the normalized product of the constituent PDFs, a fundamentally useful property that is leveraged below.

2.2.3 Product of Gaussians

In the simplest 1-D case where two observations have PDFs characterized by Gaussian distributions, it can be shown that belief fusion yields a third Gaussian PDF which is the normalized product of the two constituents. This is shown in Fig. 2. Note that the product of the distributions alone is not a PDF because it is not scaled such that the area enclosed by the

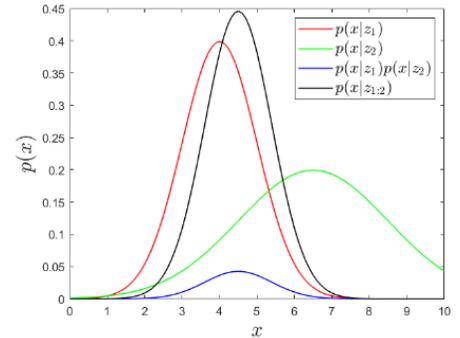


Figure 2. Belief fusion in 1D with unimodal Gaussian PDFs.

curve equals one. Once the curve has been normalized by dividing by the area, $\int_{\mathcal{X}} \prod_{i=1}^2 p(\mathbf{x}|\mathbf{z}_i) d\mathbf{x}$, the result is a proper PDF $p(\mathbf{x}|\mathbf{z}_{1:2})$.

As this example shows, the fusion of two Gaussians yields a third Gaussian with a smaller variance than either constituent PDF. Furthermore, the mean of the fused PDF is

influenced by both $p(\mathbf{x}|\mathbf{z}_1)$ and $p(\mathbf{x}|\mathbf{z}_2)$, with the latter having a weaker effect due to its higher uncertainty. This natural result lends to the intuitiveness of belief fusion as described below.

2.3 Multi-Gaussian Belief Fusion

The multi-dimensional MBF technique formulated here handles complex PDFs by making use of the concepts previewed thus far, as well as exploiting the linear superposition of Gaussian products.

2.3.1 Linear Superposition of Gaussian Products

Non-Gaussian PDFs can be estimated by normalized sums of Gaussians, described by the following formula:

$$p(\mathbf{x}) = \frac{\sum_{i=1}^I c_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\int_{\mathcal{X}} \sum_{i=1}^I c_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) d\mathbf{x}} = \sum_{i=1}^I c'_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \quad (2)$$

Here, c_i is some weighting coefficient and $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is a Gaussian distribution in the \mathbf{x} domain, with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. Note the reduction to c'_i : this is done to simplify notation, where it is understood that $c'_i = c_i (\int_{\mathcal{X}} \sum_{i=1}^I c_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) d\mathbf{x})^{-1}$. Because each Gaussian alone by definition integrates to one, it becomes clear that $c'_i = c_i (\sum_{i=1}^I c_i)^{-1}$. For unstructured PDFs, the weighting coefficients c_i are chosen to best fit a particular non-Gaussian PDF with a sum of Gaussians (this is beyond the scope of this paper). For clarity of following derivations, the general formula for a multivariate Gaussian is given below:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_i|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)\right). \quad (3)$$

It was shown by [22] that any non-Gaussian PDF can be constructed according to (2), provided a sufficient number of terms with appropriate parameters. This point is critical to propagation of belief via fusion using the proposed MBF method. Because 1) every component

PDF of belief fusion is a weighted linear superposition of Gaussians, 2) the fusion process involves the normalized product of multiple PDFs, and 3) the product of two Gaussians yields a third (unnormalized) Gaussian, the output of belief fusion can also therefore be represented as a weighted linear superposition of Gaussians. This is the essence of MBF. The mean and covariance of the product of two Gaussians can be written in terms of the constituent means and covariances [26]:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}^{(1)})\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}^{(2)}, \boldsymbol{\Sigma}^{(2)}) = c^{(3)}\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}^{(3)}, \boldsymbol{\Sigma}^{(3)}), \quad (4)$$

where

$$\begin{aligned} \boldsymbol{\Sigma}^{(3)} &= [(\boldsymbol{\Sigma}^{(1)})^{-1} + (\boldsymbol{\Sigma}^{(2)})^{-1}]^{-1}, \\ \boldsymbol{\mu}^{(3)} &= \boldsymbol{\Sigma}^{(3)}[(\boldsymbol{\Sigma}^{(1)})^{-1}\boldsymbol{\mu}^{(1)} + (\boldsymbol{\Sigma}^{(2)})^{-1}\boldsymbol{\mu}^{(2)}], \\ c^{(3)} &= \frac{1}{\sqrt{|2\pi(\boldsymbol{\Sigma}^{(1)} + \boldsymbol{\Sigma}^{(2)})|}} \exp\left(-\frac{1}{2}(\boldsymbol{\mu}^{(1)} - \boldsymbol{\mu}^{(2)})^T (\boldsymbol{\Sigma}^{(1)} + \boldsymbol{\Sigma}^{(2)})^{-1} (\boldsymbol{\mu}^{(1)} - \boldsymbol{\mu}^{(2)})\right). \end{aligned}$$

The scaling factor $c^{(3)}$ is equivalent to the area of the unnormalized curve.

The implementation of MBF for belief fusion with two multi-Gaussian PDFs is as follows.

Let the first PDF be denoted $p^{(1)}(\mathbf{x})$, and the second $p^{(2)}(\mathbf{x})$. The product of these distributions is

$$\begin{aligned} p^{(3)}(\mathbf{x}) &= [p^{(1)}(\mathbf{x})][p^{(2)}(\mathbf{x})] = \left[\sum_{i=1}^I c_i^{(1)} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^{(1)}, \boldsymbol{\Sigma}_i^{(1)}) \right] \left[\sum_{j=1}^J c_j^{(2)} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j^{(2)}, \boldsymbol{\Sigma}_j^{(2)}) \right] \\ &= \sum_{i=1}^I \sum_{j=1}^J c_i^{(1)} c_j^{(2)} c_{ij}^{(3)} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{ij}^{(3)}, \boldsymbol{\Sigma}_{ij}^{(3)}), \end{aligned} \quad (5)$$

where

$$c_{ij}^{(3)} = \frac{1}{\sqrt{\det\left(2\pi\left(\boldsymbol{\Sigma}_i^{(1)} + \boldsymbol{\Sigma}_j^{(2)}\right)\right)}} \exp\left(-\frac{1}{2}\left(\boldsymbol{\mu}_i^{(1)} - \boldsymbol{\mu}_j^{(2)}\right)^T \left(\boldsymbol{\Sigma}_i^{(1)} + \boldsymbol{\Sigma}_j^{(2)}\right)^{-1} \left(\boldsymbol{\mu}_i^{(1)} - \boldsymbol{\mu}_j^{(2)}\right)\right), \quad (6)$$

$$\mathbf{\Sigma}_{ij}^{(3)} = \left[(\mathbf{\Sigma}_i^{(1)})^{-1} + (\mathbf{\Sigma}_j^{(2)})^{-1} \right]^{-1}, \quad (7)$$

$$\boldsymbol{\mu}_{ij}^{(3)} = \mathbf{\Sigma}_{ij}^{(3)} \left[(\mathbf{\Sigma}_i^{(1)})^{-1} \boldsymbol{\mu}_i^{(1)} + (\mathbf{\Sigma}_j^{(2)})^{-1} \boldsymbol{\mu}_j^{(2)} \right]. \quad (8)$$

Therefore, the output under MBF is an array of $N \times 1$ mean vectors, $N \times N$ covariance matrices, and scalar weighting coefficients, where N is the dimensionality of \mathbf{x} ; each output channel contains $I \times J$ components. The required memory storage, then, goes as $(N^2 + N + 1)(I \times J)$. If observations are available from M sources, this becomes $(N^2 + N + 1)K$, where $K = \prod_{m=1}^M I_m$, and I_m is the number of Gaussians in the m^{th} observation's PDF.

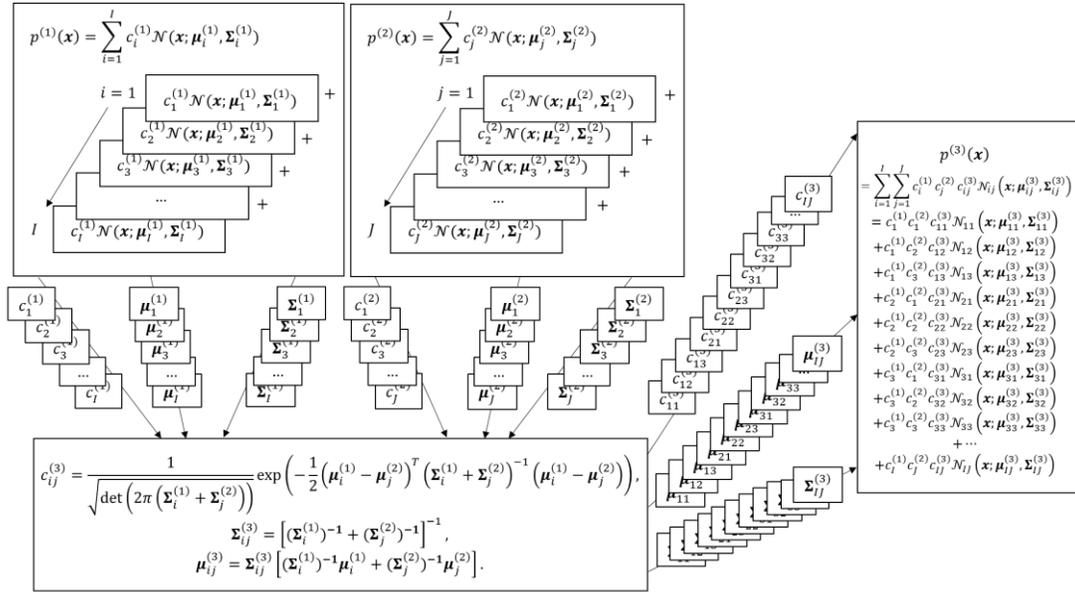


Figure 3. MBF for two multi-Gaussian PDFs.

Figure 3 provides a block diagram of MBF, with variables explicitly shown for fusion of two observations. As the figure shows, the three channels (for c , $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$) from each constituent PDF are input to the algorithm, which then outputs three new channels, each of which contains a number of elements equal to the product of the numbers of Gaussian components in all constituents. The actual output PDF can then be assembled according to (5). For fusion of multiple PDFs, the same algorithm is executed recursively, where the output of a previous iteration is fused

with the next PDF. Figure 4 illustrates this concept with a more abstracted diagram. Note that this diagram returns to the more formal notation developed in Section 2.2.

Once the three output channels have been computed, evaluating the fused PDF at any specific belief space location $\mathbf{x} = \tilde{\mathbf{x}}$ is as simple as making the following substitution:

$$p^{(3)}(\tilde{\mathbf{x}}|\mathbf{z}_{1:2}) = \sum_{i=1}^I \sum_{j=1}^J c_i^{(1)} c_j^{(2)} c_{ij}^{(3)} \mathcal{N}(\tilde{\mathbf{x}}; \boldsymbol{\mu}_{ij}^{(3)}, \boldsymbol{\Sigma}_{ij}^{(3)})$$

In general, however, for a total number K of elements in each output channel from fusion of M observations, a more appropriate algorithmic form is:

$$p(\tilde{\mathbf{x}}|\mathbf{z}_{1:n}) = \sum_{k=1}^K c_k \mathcal{N}(\tilde{\mathbf{x}}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \quad (9)$$

In this form, all three weighting coefficients are lumped into a single c_k . The MBF algorithm does this internally, so that the outputted array of weights in (9) is readily available.

Algorithm 1 provides pseudo-code for the MBF algorithm. To visualize the output, a separate function must be implemented that evaluates $p(\mathbf{x}|\mathbf{z}_{1:n})$ in a sufficiently densely sampled multi-dimensional space. It should then be evident that MBF reduces computation because, rather

than store values of the fused PDF $p(\mathbf{x}|\mathbf{z}_{1:n})$ at regular grid points in multi-dimensional space, only the parameters necessary to reconstruct this PDF are stored. The former is $O(R^N)$ complex,

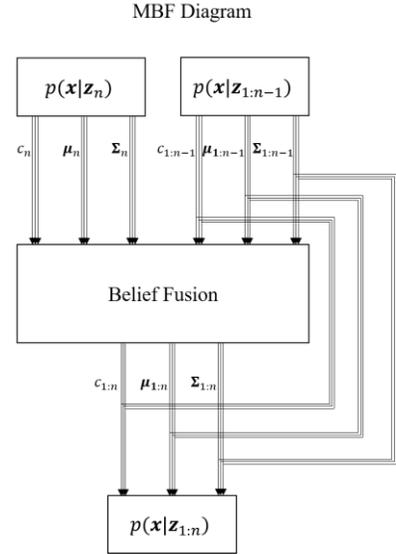


Figure 4. Recursive MBF for fusion of multiple belief sources.

```

Algorithm 1: MBF of Two Mixed Gaussians
inputs: C1,M1,S1,C2,M2,S2 // Three channel arrays per PDF
output: C,M,S // Three output channel arrays

for i along length of first channels
  for j along length of second channels
    extract ith element of C1, M1, and S1
    extract jth element of C2, M2, and S2
    compute new C, M, and S
    C = C*(ith element of C1)*(jth element of C2)
    insert new C, M, and S into output arrays
  end
end
end

```

while the latter is $O(N^2)$ complex, where R is the resolution of the sampled space and N is the dimensionality of that space.

2.3.2 Construction of Non-Gaussian Structures

The usefulness of multi-Gaussian belief fusion is directly tied to how well belief is actually represented by multi-Gaussian PDFs. Determining the composition of arbitrary unstructured PDFs input to MBF is largely left to such methods as the Expectation-Maximization (EM) algorithm [27] or other clustering algorithms [28]. However, certain types of structured non-Gaussian PDFs have particular practicality in specific contexts, such as that given in Section 2.1. When belief in some scenario is restricted by some fully or partially determined set of laws, the representative PDF ought to be structured accordingly. Such circumstances mandate the intelligent construction of multi-Gaussian PDFs for the sake of accuracy and efficiency, bypassing the need for the fitting algorithms described above. This section describes the motivations and derivations of two types of Gaussian mixtures which form structures that are of particular mathematical value. These structures are proposed in order to supplement MBF as a platform for efficient and accurate belief fusion in structured contexts.

2.3.2.1 *Gaussian Toroid*

In a multi-dimensional polar coordinate system where belief is only quantified in the radial dimension, representation of belief in tangential directions ought to be unbiased. In two dimensions, this can be visually likened to a torus, but will be more appropriately termed “toroidal” here. The toroidal distribution is, needless to say, heavily non-Gaussian in the Cartesian coordinate system. This structured PDF could, for instance, model prediction for an N -D random walk where step size is approximately normally distributed and step direction is uniformly distributed. As Fig. 5 demonstrates, a Gaussian toroid can be achieved by simply taking the difference of two Gaussian

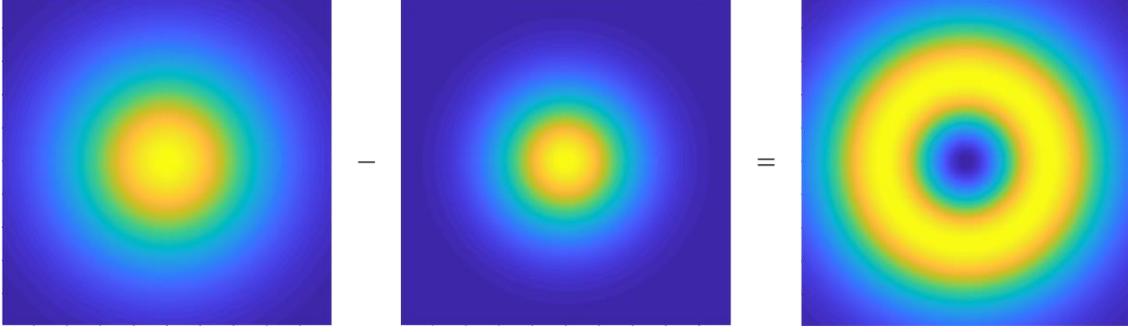


Figure 5. Construction of a 2-D Gaussian toroid as the difference between two Gaussian distributions.

distributions with the same mean and different variances. An additional constraint is that each constituent has no co-variance. Furthermore, for strictly circular toroidal distributions (excluding elliptical toroids), the variance in all dimensions of each component must be equal. Therefore, the form of a covariance matrix used to compose a Gaussian toroid is $\Sigma = \sigma^2 \mathbf{I}_{N \times N}$. Although elliptical toroids may be considered, their practical use is neither apparent nor ubiquitous, so the following developments will be reserved specifically for circular toroids.

So that the center of the toroidal PDF has a value of zero, the scaling constant that must multiply the “inner” Gaussian is the ratio of the maximum of the outer Gaussian to the maximum of the inner Gaussian. Since the maximum occurs at the mean, evaluating (3) at $\mathbf{x} = \boldsymbol{\mu}$ gives the maximum value of the function, $\frac{1}{\sqrt{|2\pi\Sigma|}}$. Let Σ_1 denote the outer distribution, while Σ_2 describes the inner distribution. The scaling ratio that must then be used is simply:

$$\frac{\frac{1}{\sqrt{|2\pi\Sigma_1|}}}{\frac{1}{\sqrt{|2\pi\Sigma_2|}}} = \sqrt{\frac{|2\pi\Sigma_2|}{|2\pi\Sigma_1|}} = \sqrt{\frac{\sigma_2^{2N}}{\sigma_1^{2N}}}$$

Because the covariance matrices are restricted by the above stipulations, this simplifies to the ratio of the scalar standard deviations raised to the power of N , the dimensionality of \mathbf{x} . In total, then, the formula for a Gaussian toroid is given as:

$$\tau(\mathbf{x}; \boldsymbol{\mu}, r) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \sigma_1^2 \mathbf{I}) - \left(\frac{\sigma_2}{\sigma_1}\right)^N \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \sigma_2^2 \mathbf{I}). \quad (10)$$

Note the introduction of a new implicit variable, r . This is the radius of the toroid, which will be shown to depend on the combination of σ_1^2 and σ_2^2 . Consider the 1-D case shown in Fig. 6. The radius is the distance from the center of the 1-D toroid to each maximum. This

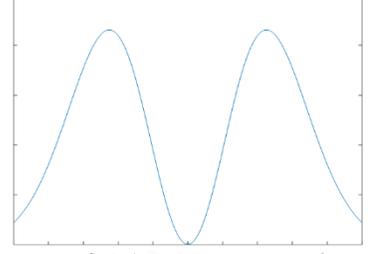


Figure 6. A 1-D Gaussian toroid.

can be found symbolically, and then extrapolated to the N -D case, by taking the derivative of $\tau(x; \mu, r)$ to find local maxima. Explicitly, this can be stated as:

$$\frac{d}{dx}(\tau(x; \mu, r)) = \frac{d}{dx} \left(\frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma_1^2}\right) - \frac{\sigma_2}{\sigma_1} \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma_2^2}\right) \right).$$

After differentiation, the following is obtained:

$$\frac{1}{\sqrt{2\pi\sigma_1^2}}(x-\mu) \left[\frac{1}{\sigma_2^2} \exp\left(-\frac{(x-\mu)^2}{2\sigma_2^2}\right) - \frac{1}{\sigma_1^2} \exp\left(-\frac{(x-\mu)^2}{2\sigma_1^2}\right) \right].$$

This expression has one clear zero at $x = \mu$, corresponding to the minimum at the mean. Other zeros occur when:

$$\frac{1}{\sigma_2^2} \exp\left(-\frac{(x-\mu)^2}{2\sigma_2^2}\right) = \frac{1}{\sigma_1^2} \exp\left(-\frac{(x-\mu)^2}{2\sigma_1^2}\right).$$

Solving the above expression for the value of x that is of interest gives the following result,

$$x = \mu \pm \sqrt{\frac{2(\ln(\sigma_2^2) - \ln(\sigma_1^2))}{\sigma_1^{-2} - \sigma_2^{-2}}},$$

where the radius is clearly then:

$$r = \sqrt{\frac{2(\ln(\sigma_2^2) - \ln(\sigma_1^2))}{\sigma_1^{-2} - \sigma_2^{-2}}}. \quad (11)$$

The above result is useful for predicting the radius of a toroid with specified variances. However, creating a toroid of a particular desired radius means solving instead for σ_1^2 when given r and σ_2^2 . Therefore, rearranging (11) gives:

$$\sigma_2^2 \exp\left(\frac{r^2}{2\sigma_2^2}\right) = \sigma_1^2 \exp\left(\frac{r^2}{2\sigma_1^2}\right),$$

a difficult expression to solve explicitly for either σ_1^2 or σ_2^2 . At this juncture, the Lambert W function (also known as the product logarithm function) [29] is introduced. It is defined as follows:

$$W(x)e^{W(x)} = x, \quad W(xe^x) = x.$$

The Lambert W function is double-valued on the interval $-\frac{1}{e} < x < 0$, and has two principle real, single-valued branches denoted by $W_0(x)$ and $W_{-1}(x)$, where $W_0(x)$ is defined over the interval $x \geq -\frac{1}{e}$ and $W_{-1}(x)$ is defined over the interval $-\frac{1}{e} \leq x < 0$. It can be shown that the solution for σ_1^2 is given by:

$$\sigma_1^2 = -\frac{r^2}{2W_0\left(-\frac{r^2}{2\sigma_2^2} \exp\left(-\frac{r^2}{2\sigma_2^2}\right)\right)}. \quad (12)$$

There are a few noteworthy observations. First, the upper bounds on the argument to the Lambert W function is zero in the trivial case when $r = 0$. Furthermore, the particular branch chosen is the W_0 branch because the W_{-1} branch yields the meaningless result that $\sigma_1^2 = \sigma_2^2$ (this is left to the reader to prove without much effort). Due to the nature of the 0^{th} branch of the function, the lower bound of the argument which yields real values is $-\frac{1}{e}$. In order to satisfy this lower bound and achieve an acceptable result, σ_2^2 must be chosen relative to r so that the following holds true:

$$2\sigma_2^2 < r^2 \rightarrow \sigma_2^2 < \frac{r^2}{2}. \quad (13)$$

Therefore, given a desired radius, the necessary variances of the inner and outer Gaussian distributions forming a Gaussian toroid can be determined from equation (13) and then (12). Algorithm 2 develops the function for creating a Gaussian toroid in more detail.

In keeping with this paper's emphasis on efficiency, it is worth noting that the function for implementing the product logarithm uses Halley's iterative

```

Algorithm 2: Gaussian Toroid
input: r // Desired radius of toroid
outputs: s1, s2 // Required outer and inner variances

s2 = (decimal between 0 and 1)(r^2/2)
s1 = -r^2/(2*W(-r^2/(2*s2)exp(-r^2/(2*s2))))

function W(x)
    implement Halley's method for the Lambert W function
end

```

approximation approach which was formulated specifically for the Lambert W function as described in [29]. The solution converges quickly for as few as five iterations, even when seeded poorly. Because the dimensionality of the covariance matrices does not appear in the final expressions (12) and (13), and because of the restrictions placed on these matrices, Algorithm 2 can be used to construct the covariance matrices in any arbitrary number of dimensions according to $\Sigma_i = \sigma_i^2 \mathbf{I}$. The Gaussian toroid fits the form of a PDF given by (2) which allows it to readily integrate into MBF. To satisfy that the function integrates over all dimensions to 1, the weighting constants must sum to 1 while their ratio is held constant. Therefore, equation (10) can be reduced to (2), where $c'_1 = \frac{\sigma_1^N}{\sigma_1^N - \sigma_2^N}$, $c'_2 = \frac{\sigma_2^N}{\sigma_2^N - \sigma_1^N}$, $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{\mu}$, $\Sigma_1 = \sigma_1^2 \mathbf{I}_{N \times N}$, and $\Sigma_2 = \sigma_2^2 \mathbf{I}_{N \times N}$, with σ_1^2 and σ_2^2 being determined by Algorithm 2. The primary advantage of this mathematical formulation for the toroid is that it allows for a highly structured and yet non-Gaussian representation of belief using only two Gaussian components. After several cycles of MBF, it becomes increasingly important for the sake of computational efficiency that the number of component PDFs be kept to a minimum, as verified in the results section.

2.3.2.2 Gaussian Line

In a Cartesian coordinate system where the PDF of belief in one dimension or direction is unquantified, an appropriate representation of belief may be what is termed here a Gaussian line. Sampling the probability distribution exclusively along a direction in which belief is unquantified

ought to yield a constant value. This is visualized below in Fig. 7. Such a structured non-Gaussian PDF could be used to probabilistically model a linear-predictive relationship between two stochastic variables. Construction the Gaussian line is derived and explained here. In order to make use of linear superposition of normal distributions, the essential exploitation of MBF, the Gaussian line must fit the form of (2). This means that, although it is a valid way to extend a Gaussian along a line, causing μ to be a function of x is not allowable here.

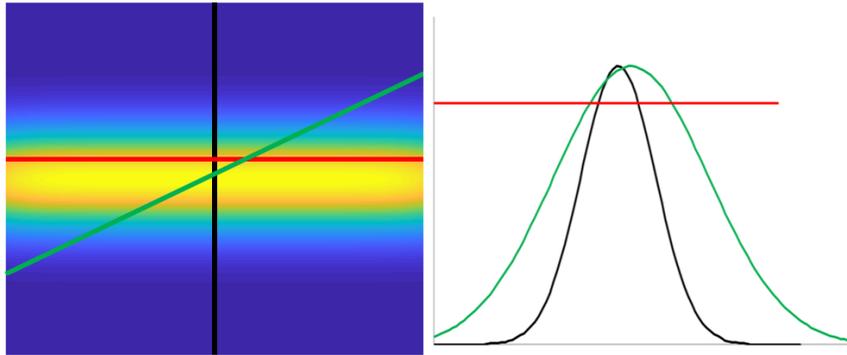


Figure 7. Cross-sectional characteristics of 2-D Gaussian line.

A finite sum of equally-spaced Gaussian PDFs with equal variances behaves differently depending on the ratio of the spacing to the variance. If this ratio is sufficiently large, the Gaussians are isolated. As the ratio decreases, they begin to interfere with one another, creating a nearly-sinusoidal profile. Over a certain range, then, this profile flattens out before beginning to resemble

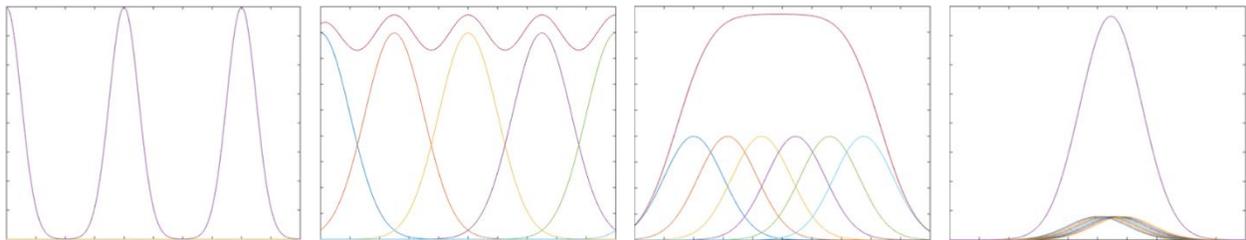


Figure 8. Profiles of Gaussian sum with equal variance and constant spacing. Colors serve only to show each PDF distinctly.

a single large Gaussian. These four cases are shown in Fig. 8. The third case is the one of interest. The two-dimensional equivalent of this case is a Gaussian line, reminiscent of the convolution of a finite linear segment and a Gaussian distribution. This holds true in higher dimensions as well.

The problem, then, is to determine the critical spacing-to-variance ratio such that a flat profile is just achieved in the center. Again, the 1-D case will be used to answer this question.

For the infinite case, a sum of Gaussians can be represented by the following:

$$y(x) = \sum_{m=-\infty}^{\infty} \mathcal{N}(x; \mu_m, \sigma^2) = \mathcal{N}(x; 0, \sigma^2) + \sum_{m=1}^{\infty} \mathcal{N}(x; \mu + m\delta, \sigma^2) + \mathcal{N}(x; \mu - m\delta, \sigma^2)$$

where δ is the desired spacing. By substitution of the formula for a 1-D Gaussian, this becomes:

$$y(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \left[\exp\left(-\frac{x^2}{2\sigma^2}\right) + \sum_{m=1}^{\infty} \exp\left(-\frac{(x+m\delta)^2}{2\sigma^2}\right) + \exp\left(-\frac{(x-m\delta)^2}{2\sigma^2}\right) \right]$$

To find the optimal spacing such that flatness is achieved, the convergence of $y(x)|_{x=\frac{\delta}{2}}$ towards $y(x)|_{x=0}$ is observed. By substitution, this requires the following:

$$1 - \exp\left(-\frac{\alpha^2}{8}\right) - \lim_{M \rightarrow \infty} \left[\sum_{m=1}^M \left(2 \exp\left(-\frac{m^2\alpha^2}{2}\right) - \exp\left(-\frac{(2m+1)^2\alpha^2}{8}\right) - \exp\left(-\frac{(2m-1)^2\alpha^2}{8}\right) \right) \right] = 0,$$

where $\alpha = \frac{\delta}{\sigma}$, the aforementioned ratio of the spacing to the standard deviation. Because this expression is intractable, it was plotted for increasingly large values of M , yielding the graphs shown in Fig. 9. As one can see, there is a range of values between $\alpha \cong 1.5$ and $\alpha \cong 0.5$ that yields consistent convergence as M increases, with the lower bound approaching zero as M approaches infinity. This means that, even for a relatively small number of Gaussian components, near-perfect

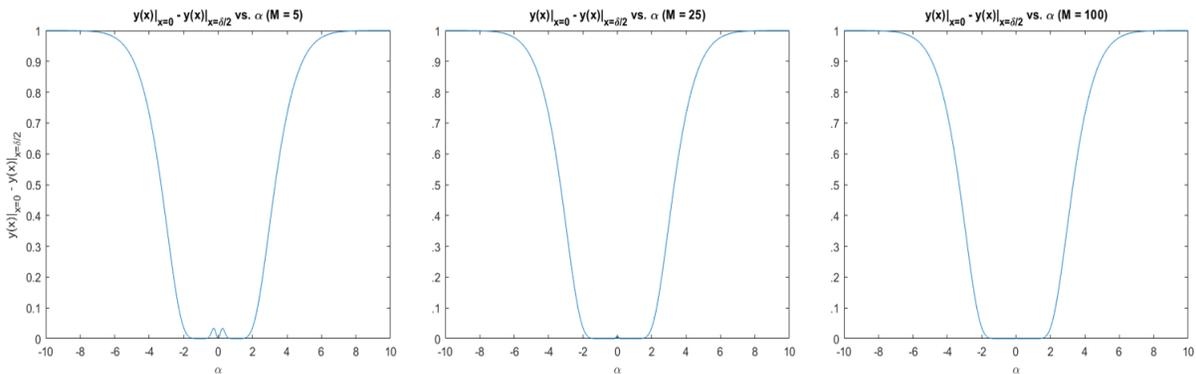


Figure 9. Flatness as a function of the ratio between spacing and standard deviation of distributions.

flatness in the center of the summed distribution can be achieved when $0.5 < \alpha < 1.5$. Choosing a value for α in algorithmic implementation is then an efficiency-accuracy tradeoff.

The value of 1.15 was heuristically chosen to satisfy the desired condition of flatness. This balances the assurance of flatness with the efficiency with which flatness can be achieved by using as few Gaussians as possible. Because $\alpha = 1.15$, the spacing must be chosen such that $\delta \leq 1.15\sigma$. Let the vectors \mathbf{e}_1 and \mathbf{e}_2 denote two endpoints of a Gaussian line in N -D. The distance D that the line spans, then, is given by $D = \|\mathbf{e}_1 - \mathbf{e}_2\|$. To construct this line, the number of Gaussian components required is $M = \frac{D}{\delta} \geq \frac{D}{1.15\sigma}$: this value is rounded to $\lceil \frac{D}{1.15\sigma} \rceil$. The required spacing between each Gaussian in the n^{th} dimension, $\delta^{(n)}$, is then given by:

$$\delta^{(n)} = \delta \left(\frac{e_1^{(n)} - e_2^{(n)}}{D} \right).$$

Finally, then, the Gaussian line is defined by the following expression:

$$\lambda(\mathbf{x}; \mathbf{e}_1, \mathbf{e}_2, \sigma) = \sum_{m=1}^M \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}) \quad (14)$$

where

$$\boldsymbol{\mu}_m = \begin{bmatrix} \mu_{m-1}^{(0)} + \delta^{(0)} \\ \mu_{m-1}^{(1)} + \delta^{(1)} \\ \vdots \\ \mu_{m-1}^{(N)} + \delta^{(N)} \end{bmatrix}$$

and

$$\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}_{N \times N}.$$

It should also be apparent that $\boldsymbol{\mu}_1 = \mathbf{e}_1$ and $\boldsymbol{\mu}_M = \mathbf{e}_2$ so that the line is closed. The above expressions inform a function which automatically produces the channels of mean vectors and covariance matrices that compose a Gaussian line between two points, with a desired cross-

sectional standard deviation σ .

The channel of weighting coefficients is simply an array of values which all equal M^{-1} .

Algorithm 3 describes the proposed solution for creating an N -D Gaussian line according to the above formulas. Note the slight

Algorithm 3: Gaussian Line

```
input: e1, e2, s // Endpoints and desired std. dev.
outputs: M, S, C // Output array of means and covariances

Compute D
Compute K (number of components, M in the text)
Compute delta

for i along dimensionality of endpoints
  Compute delta of ith dimension
  Insert delta into vector ds at ith position
End

m = e1
for i along K
  m = m + ds
  Insert m into array of means M at ith position
end
S = array of covariances, each of form s^2*I
C = array of weighting coefficients, each equal to 1/K
```

notation change of the number of components, M , to K : this is done to retain the variable name M for the output array of means in keeping with other algorithms in the text. The outputs of this algorithm can be used as inputs fed directly to the MBF algorithm.

It is worth noting that the described function creates a Gaussian line connecting two finite points, while the case where belief in one dimension or direction is unquantified implies an infinite line along this direction. Such a non-converging distribution is not a true PDF and it cannot be represented in the form of (3), a finite summation. In order to make use of the advantages of MBF, it is admittedly necessary to truncate belief in the unknown direction based on the boundaries of a certain space. However, as nearly any conceivable problem will have some expected range beyond which the true state of an estimated variable cannot possibly fall, the endpoints of the Gaussian line can simply be chosen at the edges of this belief space.

Figure 10 gives a few examples of 2-D Gaussian lines of various thicknesses (σ 's), lengths, and slopes. As the figure shows, the Gaussian line is a reasonable representation of belief that is constant in one direction and Gaussian in all others. Because the covariance matrices used to construct the Gaussian line are of the form $\sigma^2 \mathbf{I}_{N \times N}$, the level curves of each component PDF are

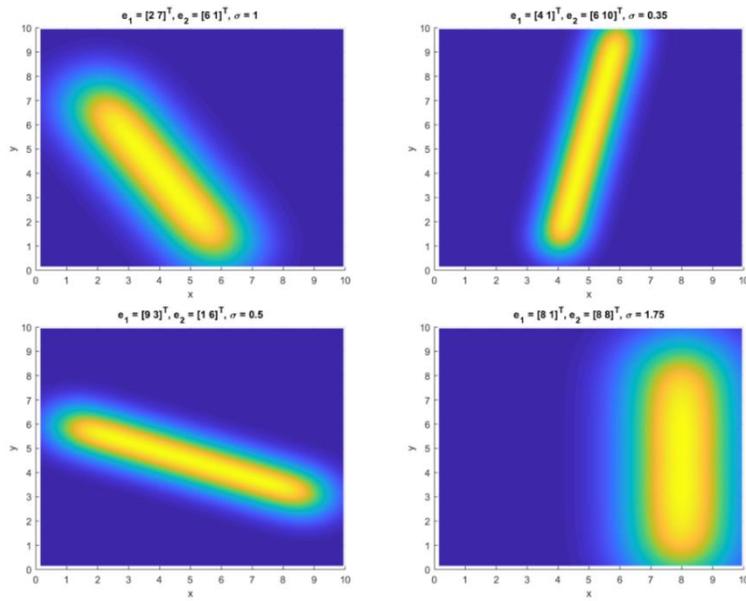


Figure 10. Four examples of Gaussian lines with varying attributes.

method, so long as the line is discretized such that the Euclidean distance between each mean and the next is approximately 1.15σ . This fact can be useful when there is some known nonlinear relationship between or amongst the dimensions of a belief space. Section 5.1 gives examples of such a scenario.

2.3.3 Decision-Making

There are multiple potential approaches to decision-making. One proposition is to simply locate the modes (local maxima) of a distribution [30]. However, this approach, known as maximum likelihood estimation (MLE), neglects most of the information-rich PDF, and multi-modal distributions can yield contradictory results. Figure

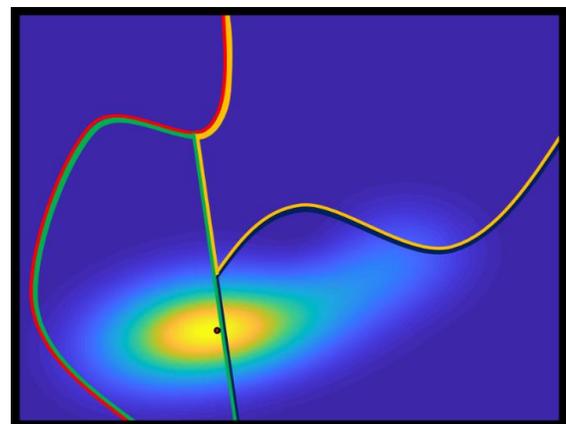


Figure 11. MLE/decision boundary classification.

11 illustrates this concept in 2-D. Another possible approach is to integrate the output PDF over decision boundaries in the multi-dimensional space to obtain the probability that belief

circular. This means that there is no inherent bias towards a “straight” line, which has significant implications with regards to the versatility of Gaussian line-making. In fact, any arbitrarily curved line in multiple dimensions can be transformed into a PDF via this

corresponds to each region. These decision boundaries can be generated by various means, one of which is the quadratic Bayesian classifier (QBC) [31]. However, efficiently performing symbolic multi-dimensional integration of a multi-Gaussian PDF over arbitrary decision boundaries is effectively intractable.

The approach to decision-making proposed in this paper is one which fully exploits all contributing components of the multi-Gaussian PDF without sampling the belief space. This can be done by an operation which bears resemblance to the total-probability theorem of continuous distributions. Let $p(w_a|\mathbf{x})$ represent the PDF corresponding to a class w_a to which a target of estimation may belong, given the estimate of \mathbf{x} that is given by training observations. Furthermore, recall that $p(\mathbf{x}|\mathbf{z}_{1:n})$ is the PDF coming from MBF. Because the actual location being estimated in the belief space \mathbf{x} is independent of the observations made (although the converse is not true), this becomes just $p(\mathbf{x})$. Therefore, the probability that a location \mathbf{x} being estimated belongs to class w_a is estimated by the marginalization of the joint probability function $p(w_a, \mathbf{x})$ as shown below:

$$P(w_a) = \int_{-\infty}^{\infty} p(w_a, \mathbf{x}) d\mathbf{x} = \int_{-\infty}^{\infty} p(w_a|\mathbf{x})p(\mathbf{x})d\mathbf{x}. \quad (15)$$

Figure 12 offers a valuable visualization of (15). The area under the joint probability curve is largest for class 2, leading to the intuitive conclusion that the probability of the output PDF belonging to this class is greatest. Because the product of the two PDFs $p(w_a|\mathbf{x})$ and $p(\mathbf{x})$ is not

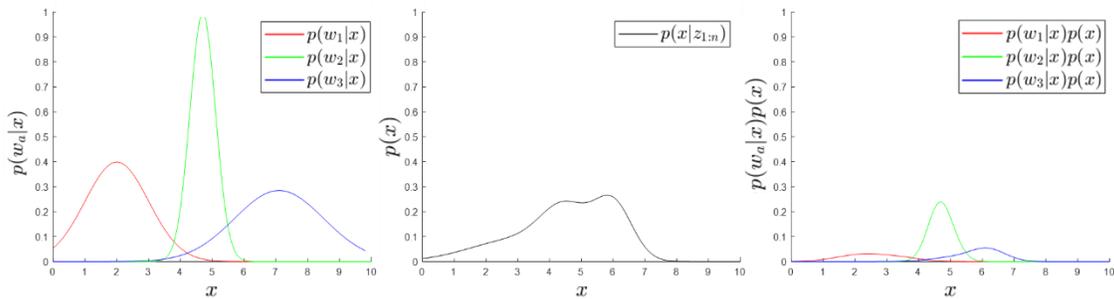


Figure 12. Decision-making by integration of the joint probability of training PDFs with the output of MBF.

normalized, the values of $P(w_a)$ will not sum to one, though their relative proportions are accurate. Therefore, a discrete normalization of these probabilities can be performed to provide probabilities corresponding to each class in the standard sense.

Equation (15) involves an integration that can be achieved non-numerically since it is not bounded by decision boundaries. According to theory developed in Section 3.1, $p(w_a, \mathbf{x})$ can be found for the general case where both $p(w_a|\mathbf{x})$ and $p(\mathbf{x})$ are multi-Gaussian PDFs,

$$p(w_a, \mathbf{x}) = \left[\sum_{i=1}^I c_i^{(p(w_a|\mathbf{x}))} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^{(p(w_a|\mathbf{x}))}, \boldsymbol{\Sigma}_i^{(p(w_a|\mathbf{x}))}) \right] \left[\sum_{j=1}^J c_j^{(p(\mathbf{x}))} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j^{(p(\mathbf{x}))}, \boldsymbol{\Sigma}_j^{(p(\mathbf{x}))}) \right]$$

$$= \sum_{i=1}^I \sum_{j=1}^J c_i^{(p(w_a|\mathbf{x}))} c_j^{(p(\mathbf{x}))} c_{ij}^{(p(w_a, \mathbf{x}))} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{ij}^{(p(w_a, \mathbf{x}))}, \boldsymbol{\Sigma}_{ij}^{(p(w_a, \mathbf{x}))})$$

where

$$c_{ij}^{(p(w_a, \mathbf{x}))} = \frac{1}{\sqrt{\det(2\pi(\boldsymbol{\Sigma}_i^{(p(w_a|\mathbf{x}))} + \boldsymbol{\Sigma}_j^{(p(\mathbf{x}))}))}} \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_i^{(p(w_a|\mathbf{x}))} - \boldsymbol{\mu}_j^{(p(\mathbf{x}))})^T (\boldsymbol{\Sigma}_i^{(p(w_a|\mathbf{x}))} + \boldsymbol{\Sigma}_j^{(p(\mathbf{x}))})^{-1} (\boldsymbol{\mu}_i^{(p(w_a|\mathbf{x}))} - \boldsymbol{\mu}_j^{(p(\mathbf{x}))}))\right).$$

This is essentially a restatement of (5). Because each Gaussian distribution integrates to one, it becomes apparent that the area under the PDF-product for each class w_a is given by:

$$P(w_a) = \int_{-\infty}^{\infty} p(w_a, \mathbf{x}) d\mathbf{x} = \sum_{i=1}^I \sum_{j=1}^J c_i^{(p(w_a|\mathbf{x}))} c_j^{(p(\mathbf{x}))} c_{ij}^{(p(w_a, \mathbf{x}))}. \quad (16)$$

The above formulations are implemented in Algorithm 4. The subsequent discrete normalization

that is necessary to yield appropriate probabilities for each class a of all A classes is then simply:

$$\tilde{P}(w_a) = \frac{P(w_a)}{\sum_{a=1}^A P(w_a)}.$$

```

Algorithm 4: Classification-Based Decision-Making
input: M1, S1, C1, M2, S2, C2 // Parameters of testing PDF
and training PDF (class wa only)
outputs: Pwa // Unnormalized probability for class wa

Pwa = 0
for i along length of first channels
  for j along length of second channels
    extract ith element of C1, M1, and S1
    extract jth element of C2, M2, and S2
    compute C
    Pwa = Pwa + C*(ith element of C1)*(jth element of C2)
  end
end
end

```

The final result is an array of probabilities, each of which corresponds to one of the classes established in training, which sum to one. These probabilities can then be used in various ways to inform action.

2.4 Results

It is insightful to include some practical examples of MBF working, and show how it compares to discretization methods. This section will provide examples of multi-Gaussian belief fusion for miscellaneous non-Gaussian PDFs (both structured and unstructured), including a comparison between the proposed probabilistic decision-making scheme and other potential approaches. In addition, several precision-efficiency trade-off studies will be presented here.

2.4.1 Belief Fusion and Decision-Making Examples

To confirm the correctness of the algorithms for constructing Gaussian structures, executing MBF, and carrying out probabilistic decision-making, eight arbitrary 2-D examples of fusion for two and three combinations of different distributions were studied. A training set consisting of three classes with uni-Gaussian PDFs was arbitrarily constructed, as shown in Fig. 13. The decision boundaries shown were established by a quadratic Bayesian classifier, though this only has direct value in terms of visualization. The results of MBF were compared to

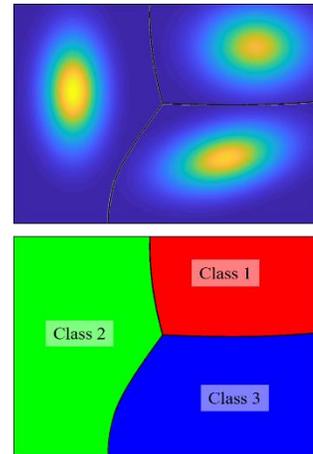


Figure 13. Three training classes and their decision boundaries.

grid-based belief fusion, where a discretized 2-D grid containing values of the PDFs evaluated at regular sample points was stored in memory and element-wise belief fusion operations were executed. Since the discretization was conducted at high resolution, the result of the grid-based belief fusion was considered ground truth. It should be noted that, although higher-dimensional

examples could be used for validation, what could be an already expansive parametric study needed to be restricted to the easily visualized 2-D examples.

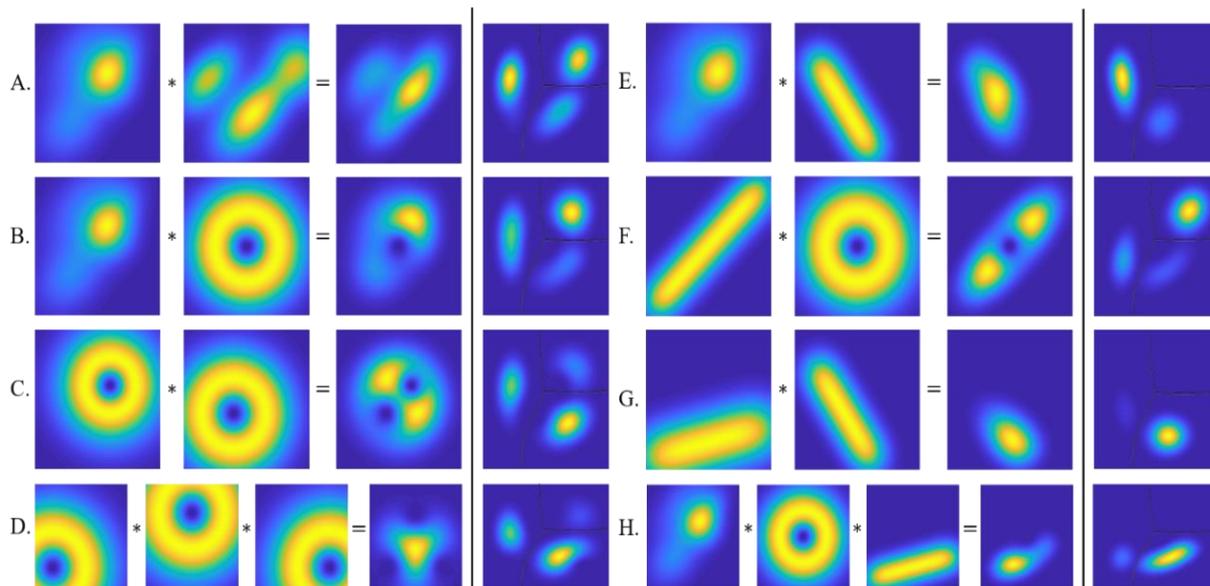


Figure 14. Eight examples of MBF. A: two arbitrary Gaussian mixtures (AGM), B: AGM and Gaussian toroid (GT), C: two GT, D: three GT, E: AGM and Gaussian line (GL), F: GL and GT, G: two GL, H: AGM, GT, and GL. Separated columns show result of decision-making fusion corresponding to three training classes shown in Fig. 13. Class probabilities are directly proportional to the volumes beneath the fused decision-making distributions.

As Fig. 14 shows, belief fusion of multiple non-Gaussian PDFs yields results which are also highly non-Gaussian and can be oftentimes multi-modal. The results of the MBF algorithm are effectively indistinguishable from ground truth. Furthermore, since MBF is a lossless method of belief fusion, any difference between it and the grid-based approach is in fact error in the latter. Table I gives class probabilities obtained in each of the eight examples under the two simpler decision-making schemes described in Section 3.3 (MLE and QBC) and the proposed technique.

Table 1. Class probabilities. Bold numbers denote winning classes; highlighted columns correspond to proposed approach.

	A			B			C			D			E			F			G			H		
	MLE	QBC	PDM																					
Class 1	0.367	0.371	0.385	0.999	0.509	0.464	0.002	0.282	0.161	0.003	0.079	0.004	0.999	0.436	0.648	0.000	0.002	0.000	0.000	0.006	0.004	0.001	0.106	0.086
Class 2	0.000	0.276	0.384	0.000	0.325	0.380	0.000	0.301	0.315	0.931	0.562	0.792	0.000	0.326	0.213	0.000	0.106	0.061	0.142	0.326	0.113	0.000	0.260	0.305
Class 3	0.633	0.353	0.231	0.001	0.166	0.156	0.998	0.417	0.524	0.066	0.359	0.204	0.001	0.238	0.139	1.000	0.892	0.939	0.858	0.668	0.883	0.999	0.634	0.609

Figure 15 shows the information entropy corresponding to each classifier for each example. This measure is employed to quantify the uncertainty present in a probabilistic decision that has been made. The information entropy $H(X)$ is calculated here as:

$$H(X) = - \sum_{i=1}^n P_i(X) \log_2(P_i(X)),$$

where $P_i(X)$ is the probability that the random variable X belongs to class i . Because the PDFs of Fig. 14 were artificially generated, ground-truth class probability information was nonexistent. Although the proposed method cannot explicitly be shown to be more accurate for this reason, some

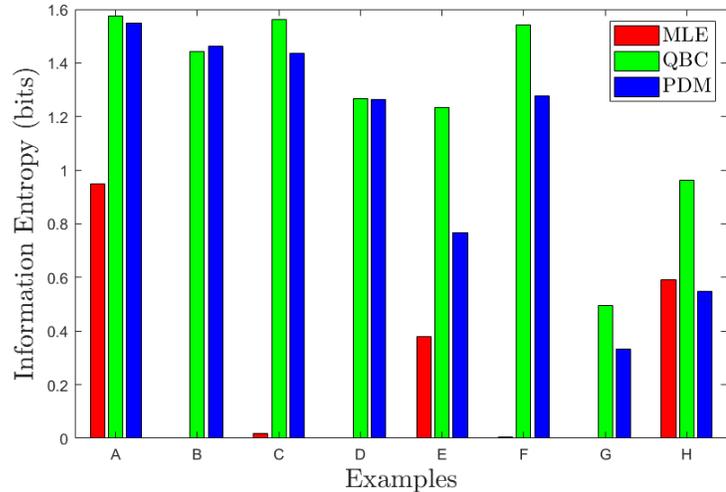


Figure 15. Information entropy for three classifiers across eight examples of post-belief fusion decision-making.

important conclusions can be made. First, though all three methods yield the same class assignment for all but one example, the MLE approach often yields vastly overconfident probabilities. This is shown by the low MLE values in Fig. 15. Furthermore, the probabilities assigned by the proposed probabilistic decision-making (PDM) and QBC methods correspond more naturally to a human's innate ability to probabilistically classify a target with significant uncertainty given sufficient information. As Fig. 15 shows, information entropy is higher for both these classifiers to reflect the uncertainty in decision-making under these approaches. Because these classification methods avoid asserting an overconfident decision, it is more meaningful in cases where $H(X)$ is low (for example, in case G). In comparing the QBC and PDM methods, however, the PDM approach is

much more efficient since the boundary-integration approach labeled by QBC requires belief-space discretization which is to be avoided for reasons already given.

2.4.2 Gaussian Component Pruning

One drawback of MBF is that, for continuous belief fusion where the algorithm is recursively implemented or when each constituent PDF has many Gaussian components, the length of the output channels becomes quite large. However, in the vast majority of instances where this situation occurs, most of the Gaussian components contribute very little. Generally, in fact, the more Gaussian components are present, the less most of them contribute. This makes it possible to simply “prune” out minimally-contributing components. This can be done by removing those whose weighting coefficients’ absolute value is below a certain threshold. Alternatively, the number of top contributors chosen can be specified. The effects of pruning on examples A, B, E, F, G, and H are shown in Fig. 16 (examples C and D are excluded because they already contain few components, and all contribute approximately equally).

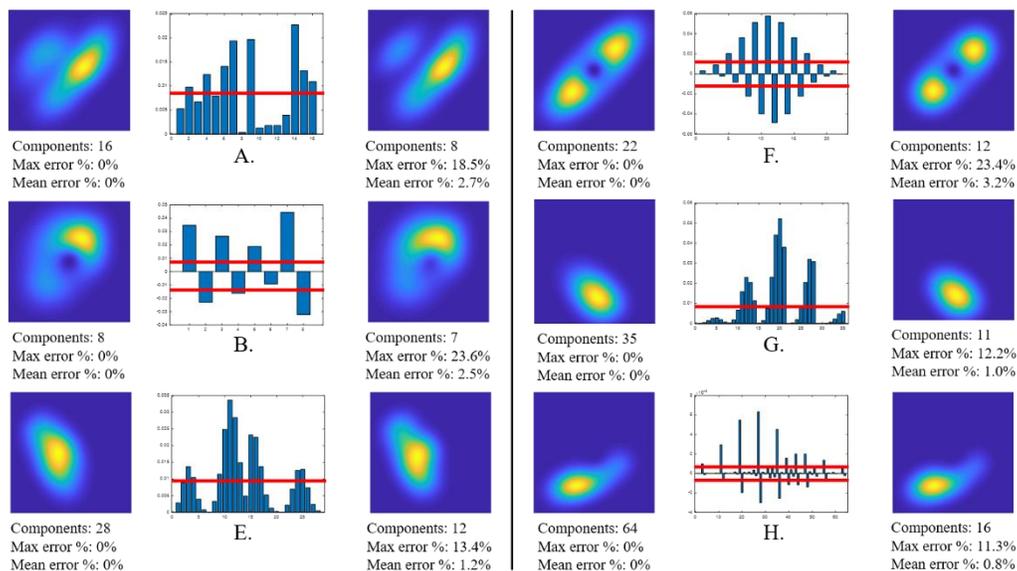


Figure 16. Effect of pruning on examples A, B, E, F, G, and H with accompanying bar graphs of constituent weighting coefficients. On average, over half the components can be removed with little overall

The accuracy of the pruned PDFs is quantified by two metrics: max error percent and mean

error percent. These are simply defined as the maximum absolute-value-error and mean absolute-

value-error in each PDF (compared to the unpruned PDF) divided by the maximum value of the pruned PDF. They are meant to show the max and mean error in proportion to the scale of the PDF overall. As the figure details, the mean error percent is often only a few percent, and the max error percent at times rises above 20%. However, considering the substantial computational benefit of pruning in some cases (particularly example H), these small errors are often worth inducing.

For a more rigorous quantitative analysis of the effect of component pruning on the class probabilities assigned in a probabilistic decision-making stage, the class probabilities were plotted over the number of components M preserved in pruning for each example. Components were added in descending order according to the absolute value of their weighting coefficients. Figure 17 shows these plots. If classes are assigned based on maximum probability, the highlighted

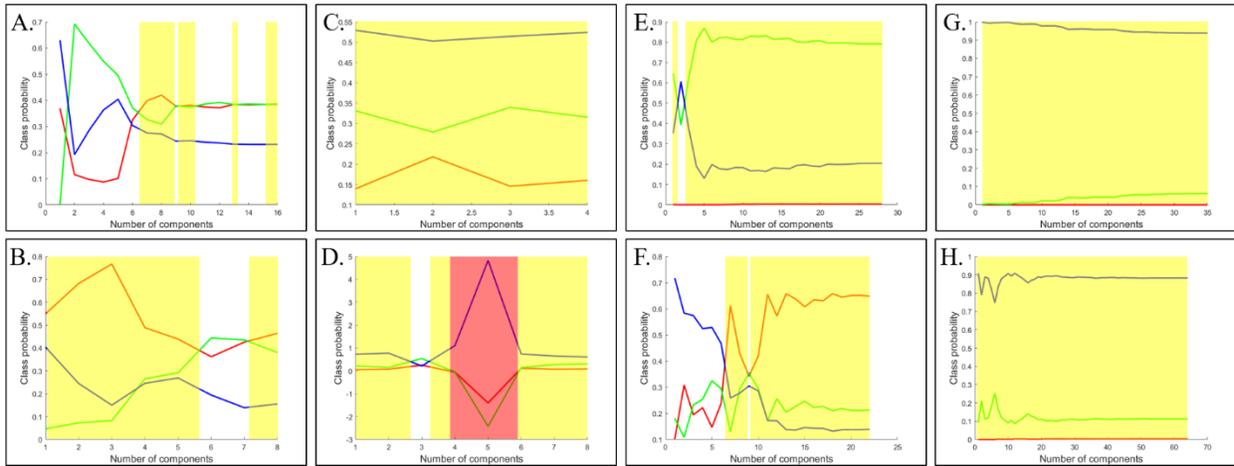


Figure 17. Plots of output class probabilities vs. number of Gaussian components for examples A-H. Red = class 1, green = class 2, and blue = class 3.

regions correspond to values of M for which pruning to the M most contributing components yields correct classification. The red region in example D indicates a condition where the particular components retained causes instability in reconstructing the output of MBF. This is due to the contribution of negative terms by the Gaussian toroid components. A more in-depth study of the

causes of unstable partial-reconstructions due to negative terms is warranted, but extends beyond the scope of this paper.

As the figure shows, removing the least-contributing half of the Gaussian components coming from MBF results in the correct maximum-probability classification for examples B, C, E, F, G, and H. In other cases, special selections of M yield correct classification. In example A, although the highlighted regions cover less area, it is clear that classes 1 and 2 compete closely, so misclassification in this case costs relatively little. It is important to note here that other methods of component selection may be utilized to yield even better results with fewer components. Again, while this warrants further scrutiny, it is beyond the scope of this paper.

2.4.3 Efficiency vs. Dimensionality

For the grid-based method, computation includes not only element-wise multiplication, division, and summation involved in belief fusion, but also the operations involved in populating a discrete space with the appropriate values corresponding to the distributions of each input PDF. Though the latter procedure is not formally a part of the belief fusion operation, it must take place for each observation contributing to belief fusion and therefore must be considered. Populating an N -D belief space of resolution R with a single Gaussian requires the following operations (among other smaller ones):

- 1) Evaluating the inverse of an $N \times N$ matrix, which involves:
 - a) Evaluating the determinant of an $N \times N$ matrix
 - b) Evaluating the adjoint of an $N \times N$ matrix
- 2) Subtracting two $N \times 1$ vectors
- 3) Evaluating the transpose of an $N \times 1$ vector
- 4) Looking up the value of an exponential function or approximating it numerically

The minimum computation, then, involved in this procedure is one iteration of the steps involved in 1) and R^N iterations of steps 2) - 4) (assuming each dimension has equal resolution). Assuming

steps 2) - 4) require a constant amount of computation time γ , the dominant characteristic of this operation in high dimensions, assuming efficient execution of 1), is at best $(R\gamma)^N$. Carrying out actual belief fusion, then, involves:

- 1) Multiplying the values of two input PDFs at each sampled cell through dimensions m_1, m_2, \dots, m_N :

$$p_{m_1, m_2, \dots, m_N}^{(3)} = p_{m_1, m_2, \dots, m_N}^{(1)} \times p_{m_1, m_2, \dots, m_N}^{(2)}$$

- 2) Summing over all locations of $p_{m_1, m_2, \dots, m_N}^{(3)}$ in N -D space:

$$\alpha = \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \dots \sum_{m_N=1}^{M_N} p_{m_1, m_2, \dots, m_N}^{(3)}$$

- 3) Dividing each cell by α :

$$p_{m_1, m_2, \dots, m_N}^{(4)} = \frac{p_{m_1, m_2, \dots, m_N}^{(3)}}{\alpha}$$

Though operations 1 and 3 are simple, they are done R^N times. Operation 2 is done once, but this operation itself requires the addition of R^N elements. Therefore, this procedure is also dominated by a power law, as was described at the end of Section 3.1.

In order to rigorously compare MBF with the existing grid-based approach, belief fusion was carried out for two arbitrary two-component PDFs in 1-, 2-, 3-, 4-, 5-, and 6-D. In each successive dimension, the means and corresponding covariances were retained from the lower dimension so as to maintain consistency across the trials. To cover the computationally worst-case scenario, each covariance matrix was non-sparse. The algorithms were run in Matlab 2018a on a non-dedicated Intel i7 processor with 12GB RAM running at 2.20 GHz. Figure 18 shows the computation time plotted as a function of dimensionality, both in linear and logarithmic scales. In

the study shown, the resolution of the sample spaces was extremely low, at only 10 grid cells per

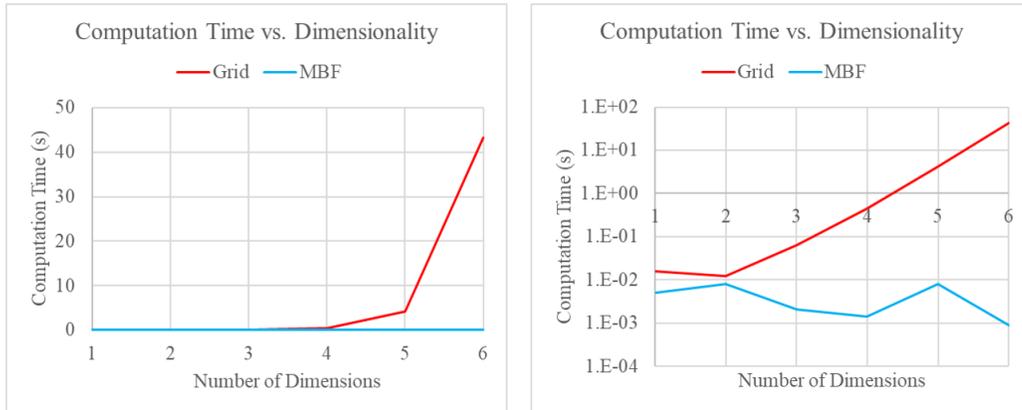


Figure 18. Computation time as a function of dimensionality for uniform resolution of 10 grid cells per dimension.

dimension.

As the figure shows, at low dimensionality (1- and 2-D), computation is fast enough that processing speeds are negligible and therefore highly variable because other processor operations on the non-dedicated machine fluctuate. However, as expected for the grid-based method, the relationship becomes dominated by a power law at higher dimensionality. This is especially clear in the near-linear form of the logarithmic plot past 2-D. The computation involved with MBF remains so small that it is dominated by other processor operations. This expected benefit makes MBF far more favorable than grid-based belief fusion of non-Gaussian PDFs in many dimensions.

Because the efficiency of the two methods is comparable in 2-D, an accuracy trade-off study was conducted to compare the two at this dimensionality. Inaccuracy in the grid-based approach comes at the discretization level, whereas it is introduced in MBF by component pruning. As a representative case, example E from Sections 4.1 and 4.2 was chosen to carry out this study. For ground-truth, a densely sampled 200-by-200 belief space was selected to evaluate the two methods. Representation of the output from grid-based fusion was examined over twenty resolutions from 5 to 100, and MBF output belief was reconstructed in the 200-by-200 belief space

after incorporating each of the 28 components, ordered from most- to least- contributing. Root-mean-squared-error (RMSE) was computed over all grid cells. The results are shown in Figure 19.

In order to make use of the figure, note that each approach's independent variable has its own axis label. To compare the two, the RMSE at a particular resolution or number of components can be found and then related to the other approach. For example, at a resolution of 30-by-30, the grid-

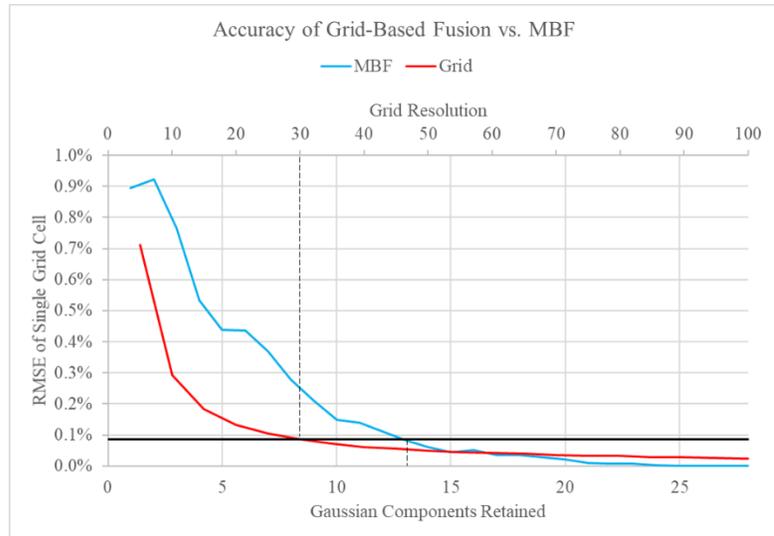


Figure 19. RMSE of grid cells in 200-by-200 belief space for example E under two methods.

based representation of belief results in error comparable to that induced by pruning approximately 13 components after MBF. In other words, to achieve results with MBF that are more accurate than the grid-based method at a resolution of 30 grid cells, one would be able to eliminate up to half the Gaussian components representing output belief. For two further iterations of belief fusion with arbitrary two-component PDFs, the average computation time required for MBF was 5.2 ms (maintaining only 13 components) while grid-based fusion took an average of 23.4 ms (for 30-by-30 resolution). This means that for the 2-D example E, MBF equals the accuracy of grid-based fusion while quartering the required computation time.

2.4.3 Efficiency/Accuracy Tradeoff for Multiple MBF Iterations

To more comprehensively study the effect of Gaussian component pruning on efficiency and accuracy, seven consecutive fusions were executed using both MBF and grid-based fusion for

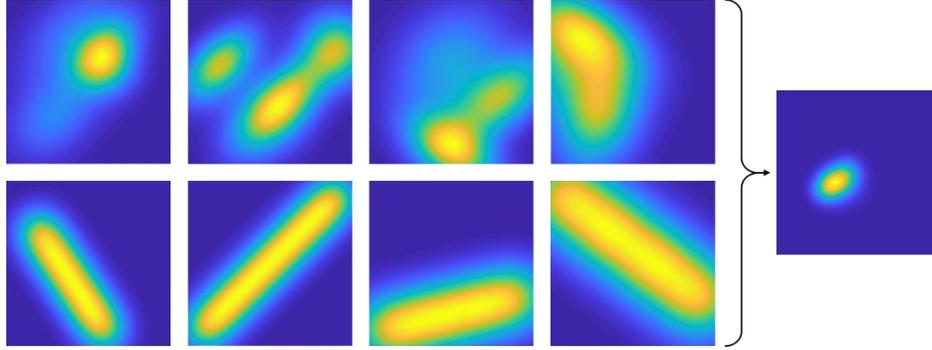


Figure 20. Eight miscellaneous multi-Gaussian PDFs fused in consecutive iterations of MBF. From upper-left to lower-right, the PDFs contain 4, 4, 3, 3, 7, 11, 5, and 4 components. The final PDF, therefore, contains 221,760 components.

the eight miscellaneous 2-D PDFs shown in Fig. 20. Between each fusion event, the cumulative execution time and the RMSE were recorded, where ground-truth was established by a densely sampled 500-by-500 cell space. These two metrics were assessed for varying resolutions under the grid-based method and for varying component-pruning laws under MBF. The results of this study are shown in Fig. 21.

As expected, there is a roughly proportional trend between time and fusion events for grid-based fusion since the memory needed to store PDF information is constant. Alternatively, because the number of components needed to represent belief under MBF follows a product law, many MBF

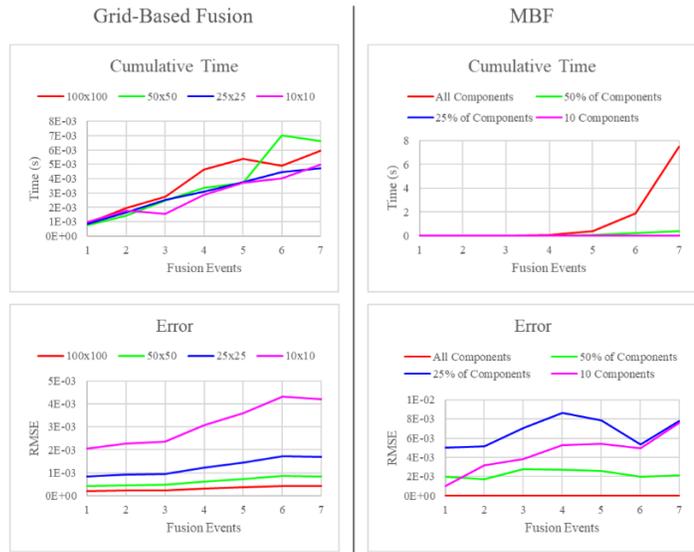


Figure 21. Timing and error as functions of the number of fusion events for various resolutions and pruning laws.

iterations become expensive under this approach. However, by simply eliminating half the Gaussian components after each fusion event, computation time is drastically reduced. Furthermore, restricting belief to be represented by a constant number of components (in this example, 10) causes MBF computation time to be governed by a proportional law. At this level, MBF is not much more expensive than grid-based fusion in 2-D, and the accuracy is also

comparable. As the previous section shows, however, the advantage of MBF becomes much more pronounced in higher dimensions. For any applications requiring the fusion of only a few PDFs, MBF is both efficient and perfectly accurate without any pruning.

2.5 Applications

For completeness, this work requires some elaboration about the various potential applications of MBF. Ultimately, in most contexts where human-like estimation and probabilistic problem solving is desired, these methods could likely have some use. This section will focus on three areas of particular interest: state variable estimation, target tracking, and classification.

2.5.1 Dynamics: State Variable Estimation

In the fields of dynamics and controls, state-space representations of complex systems are often sought after in order to solve stability problems, provide appropriate actuation, and the like. A state space is analogous to the concept of a belief space that was used to develop MBF. State spaces are often many-dimensional, and state vectors are often accompanied with uncertainty. In many cases, the Kalman filter and its variants are capable of state estimation in the time domain under the assumption of Gaussian uncertainty. A proposed use of MBF in dynamics is instead in the state domain where non-Gaussian models are appropriate.

Consider a simple system with state vector $\mathbf{x} = [y(t) \quad \dot{y}(t)]^T$. There is a deterministic curve that could be drawn in the 2-D state space for different functions $y(t)$ and their

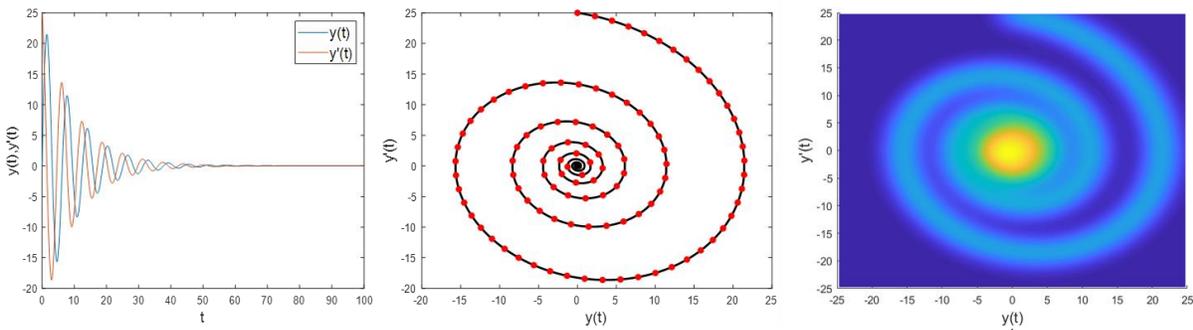


Figure 22. Time domain plot, state-space plot, and state-space PDF of decaying sinusoid of form $C \exp(-\frac{t}{\tau}) \sin(\omega t)$ where $C = 25$, $\tau = 10$, and $\omega = 1$.

corresponding derivatives $\dot{y}(t)$. For the free-response of a spring-mass-damper system, these two states are plotted in the time domain in Fig. 22. If the states are instead plotted in the state-space, the resulting shape can be described as a spiral, as shown in the second part of the figure. An observation at any time ought to fall somewhere along this path. Furthermore, if the uncertainty with which this path is known can be estimated, the deterministic trajectory can be transformed into a PDF by placing equally-spaced non-covariant Gaussians along the path according to the theory developed in Section 3.2.2 (shown in the third part of Fig. 22). Because the curve describes the dependency of $\dot{y}(t)$ on $y(t)$, this PDF corresponds to the conditional distribution of $\dot{y}(t)$ given $y(t)$ at any time. A measurement of $y(t)$ with Gaussian uncertainty would then be represented by a vertical Gaussian line with a specified mean and variance. This structure is normally distributed in the $y(t)$ direction and uniformly distributed in the $\dot{y}(t)$ direction to reflect 1-D belief in 2-D space. The fusion of these two non-Gaussian 2-D PDFs would yield the joint PDF of $\dot{y}(t)$ and $y(t)$, which could then be used to extract the marginal distribution describing $\dot{y}(t)$. This fusion is demonstrated in Fig. 23.

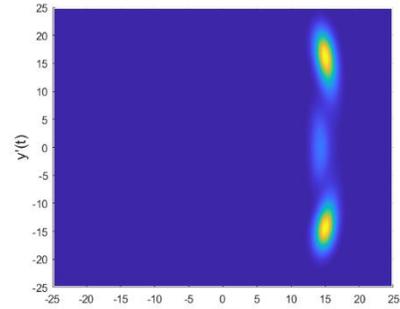


Figure 23. Joint PDF of $y(t)$ and $\dot{y}(t)$.

A few examples of other potentially relevant or interesting state-space trajectories are given in Fig. 24 as non-Gaussian PDFs. Although the primary relationship explored here is that of a

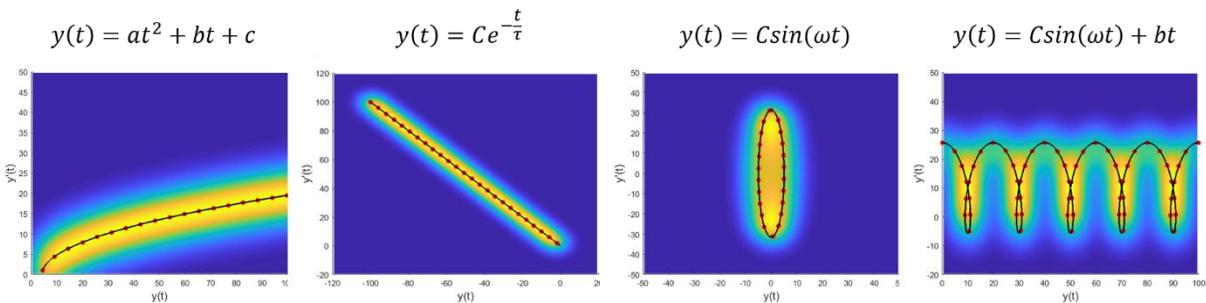


Figure 24. Examples of four potential state-space PDFs where the states are the associated function and its derivative. Note how the diverging first and fourth examples are unbounded in the state space in keeping with their instability.

are represented as PDFs. The uncertainty characteristics of potentially multi-Gaussian distributions would be determined by the reliability of the sensing system, and MBF across multiple sensors or observations would increase certainty in belief. Probabilistic classification could then be achieved by the decision-making methods described in Section 3.3 and validated in Section 4.1, where training data consists of manually assessed ground-truth instances.

2.6 Conclusion

In conclusion, the MBF technique described in this paper offers a fast and precise solution to probabilistic belief fusion in a broad range of conceivable applications. The mathematical formulations of MBF are capable of handling high-dimensional non-Gaussian belief when mandated by the problem at hand. Because there is no need to store values in a discretely sampled space or generate random particles to handle non-Gaussian belief, MBF is fast and accurate in many dimensions as compared to existing methods. While a grid-based approach requires belief space discretization and therefore reflects a power law in computational efficiency, MBF computation is dominated by a square law, making it far more efficient in many dimensions.

This work also developed two useful Gaussian structures as tools for implementing MBF in potentially common instances. In addition, a probabilistically robust decision-making scheme was presented, enhancing the practicality of MBF as shown in the working examples. For eight arbitrary 2-D examples, non-Gaussian belief fusion was executed with full precision, and it was found that the computational advantage afforded by appropriate Gaussian component pruning had little negative effect on classification outcomes. This was evidenced by the fact that, in most cases, removing the least-contributing half of the Gaussian components resulted in the same maximum-probability classification according to the classification scheme given. Considering the substantial computational benefit of pruning, small errors are often worth inducing.

Additional research is needed to investigate applications of this work to problems mandating real-time agility and real-world complexity. Ongoing work includes implementation of the developed algorithms into a Graphics Processor Unit (GPU) and the development of more useful Gaussian structures, including the effect of pruning on such structures. The developed algorithms have high compatibility with parallel computing and thus can be performed at accelerated speeds on a GPU. Techniques for creating non-Gaussian PDFs by sums of Gaussian with positive and negative weighting coefficients may substantially ease computational burden. For example, the N -D Gaussian toroid could in theory be approximated by a different Gaussian sum, but is instead much more efficiently modeled by the subtraction of only two Gaussians. Finally, for contexts where decision-making requires a single location vector rather than a PDF, multi-Gaussian-specific maximum likelihood estimation is being developed. SDG

2.7 References

- [1] M. Liggins II, D. Hall, J. Llinas, Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition, Boca Raton, FL, USA: CRC Press, 2017.
- [2] H. Lee, H. Kwon, R. M. Robinson, W. Nothwang, A. Marathe, “Dynamic Belief Fusion for Object Detection,” arXiv, pp. 1-8, 2015.
- [3] Y. Liu, X. Fan, C. Lv, J. Wu, L. Li, D. Ding, “An innovative information fusion method with adaptive Kalman filter for integrated INS/GPS navigation of autonomous vehicles,” Mechanical System and Signal Processing, vol. 100, pp. 605-616, 2018.
- [4] M. M. Atia, A. R. Hilal, C. Stelling, E. Hartwell, J. Toonstra, W. B. Miners, O. A. Basir, “A Low-Cost Lane-Determination System Using GNSS/IMU Fusion and HMM-Based Multistage Map Matching,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 11, pp. 3027 - 3037, 2017.

- [5] S. Z. Gurbuz, C. Clemente, A. Balleri, J. J. Soraghan, "Micro-Doppler-based in-home aided and unaided walking recognition with multiple radar and sonar systems," *IET Radar, Sonar & Navigation*, vol. 11, no. 1, pp. 107 - 115, 2017.
- [6] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. D, pp. 35-45, 1960.
- [7] J. E. Wall, A. S. Willsky, N. R. Sandell, "On the Fixed-Interval Smoothing Problem," *Stochastics*, vol. 5, pp. 1-41, 1981.
- [8] Salina B., P. Malathi, R. Arokia Priya, "An Efficient Sensor Fusion Technique for Obstacle Detection," *International Journal of Advance Research in Engineering, Science & Technology*, pp. 11-14, 2014.
- [9] W. Q. Liu, X. M. Wang, Z. L. Deng, "Robust centralized and weighted measurement fusion Kalman estimators for uncertain multisensor systems with linearly correlated white noises," *Information Fusion*, vol. 35, pp. 11-25, 2017.
- [10] G. Einicke, *Smoothing, Filtering and Prediction: Estimating the Past, Present and Future*, Rijeka, Croatia: IntechOpen, 2012.
- [11] V. Tzoumas, A. Jadbabaie, G. J. Pappas, "Sensor Placement for Optimal Kalman Filtering: Fundamental Limits, Submodularity, and Algorithms," in *American Control Conference (ACC)*, Boston, MA, USA, 2016.
- [12] B. Cai, Y. Liu, Q. Fan, Y. Zhang, Z. Liu, S. Yu, R. Ji, "Multi-source information fusion based fault diagnosis of ground-source heat pump using Bayesian network," *Applied Energy*, vol. 114, pp. 1-9, 2014.
- [13] G. Shafer, *A Mathematical Theory of Evidence*, vol. 1, Princeton, NJ, USA: Princeton university press Princeton, 1976.

- [14] G. Shafer, "Dempster's rule of combination," *International Journal of Approximate Reasoning*, vol. 79, pp. 26-40, 2016.
- [15] A. Jøsang, J. Diaz, M. Rifqi, "Cumulative and averaging fusion of beliefs," *Information Fusion*, vol. 11, no. 2, pp. 192-200, 2010.
- [16] N. Bergman, *Recursive Bayesian Estimation: Navigation and Tracking Applications*, Linköping, Sweden: Department of Electrical Engineering, Linköping University, 1999.
- [17] M. Kumon, D. Kimoto, K. Takami, T. Furukawa, "Bayesian Non-Field-of-View Target Estimation Incorporating An Acoustic Sensor," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3425-3432, 2013.
- [18] M. Silbert, T. Mazzuchi, S. Sarkani, "Comparison of a grid-based filter to a Kalman filter for the state estimation of a maneuvering target," *Proc. of SPIE, Signal and Data Processing of Small Targets*, vol. 8137, 2011.
- [19] Y. Kim, K. Hong, H. Bang, "Utilizing Out-of-Sequence Measurement for Ambiguous Update in Particle Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 493-501, 2018.
- [20] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.
- [21] X. Wang, T. Li, S. Sun, J. M. Corchado, "A Survey of Recent Advances in Particle Filters and Remaining Challenges for Multitarget Tracking," *Sensors*, vol. 17, no. 12, 2017.
- [22] H. W. Sorenson, D. L. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica*, vol. 7, pp. 465-479, 1971.

- [23] D. L. Alspach, H. W. Sorenson, "Nonlinear Bayesian Estimation Using Gaussian Sum Approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439-448, 1972.
- [24] T. Li, J. M. Corchado, S. Sun, "Partial Consensus and Conservative Fusion of Gaussian Mixtures for Distributed PHD Fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. Early Access, 2018.
- [25] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*, Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2008.
- [26] K. B. Petersen, M. S. Pedersen, *The Matrix Cookbook*, Technical University of Denmark, 2012.
- [27] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1-38, 1977.
- [28] K. Fukunaga, L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32 - 40, 1975.
- [29] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, D. E. Knuth, "On the Lambert W Function," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329-359, 1996.
- [30] M. Timouyas, A. Hammouch, S. Eddarouich, "A new approach of classification for non-Gaussian distribution upon competitive training," in *IEEE International Conference on Complex Systems (ICCS)*, Agadir, Morocco, 2012.
- [31] F. van der Heijden, R. P. Duin, D. de Ridder, D. M. J. Tax, *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*, West Sussex, England: John Wiley & Sons, 2004.

CHAPTER 3. The Gaussian Toroid as a Prediction Model

3.1 Introduction

3.1.1 Background

The fields of robotics and automation have become increasingly reliant on probabilistic methods in recent decades. From state estimation in the feedback control of actuated systems to the problem of target estimation, the importance of creating robots and automated systems that can overcome uncertainty and noise is well recognized. Recursive Bayesian estimation (RBE) is a tool that has been particularly influential in probabilistic robotics; it is often applied in the context of spatial target estimation, though many other applications are within the realm of possibility. One area in which the probabilistic benefits achieved by recursive Bayesian techniques have not been fully realized is in automated classification and perception. This paper presents an augmentation to RBE that is formulated specifically for classification problems. The proposed recursive Bayesian classification (RBC) technique has potential for use in a wide variety of robotic perception and machine learning problems.

In many conceivable scenarios, the class of a target of perception may evolve stochastically over time or space. Such evolution must be addressed probabilistically in order to preserve inherent uncertainties and their implications. This non-deterministic nature of class evolution is compensated for in this work by the introduction of a Gaussian toroid prediction model. Furthermore, with regards to automated perception contexts, computational efficiency and real-time processing capabilities are often key objectives. As a result, the techniques proposed here aim to specifically address the propagation of non-Gaussian belief in high-dimensional feature spaces without computational burden that would compromise real-time analysis.

3.1.2 Related Work

In traditional machine learning contexts, classification is foundational. Methods for classification vary from neural networks to statistical clustering analysis [1] to Bayesian methods [2] and the like. Various intelligent classification systems go beyond the simplest approaches to make use of rich probabilistic information and reexamine the assumptions underlying traditional methods [3], [4], [5]. However, well-established approaches do not implement the robust prediction-observation-correction scheme of recursive Bayesian estimation, usually found in target tracking or dynamics estimation problems [6], in the context of classification and robotic perception. Although the machine learning community has seen advances in adaptive classification where the classifier or underlying training classes are continuously evolving [7], [8], evaluation of evolving targets is generally not addressed in existing literature. A partial form of RBE oriented towards classification problems was investigated in [9], but this work was limited to correction only, in a low-dimensional feature space.

Existing means of carrying out RBE are well known, and range from the Kalman filter and its variants [10], [11] to the particle filter [12], to expensive discretization methods [13]. Most of these approaches work well and are efficient for low-dimensional belief spaces, but they present issues when high dimensionality and high accuracy are required. A multi-Gaussian approach to estimating non-Gaussian belief was addressed in [14], and has since been expanded by [15] for belief fusion in high-dimensional spaces. Due to the efficiency of the multi-Gaussian solution to RBE, and its ability to handle high dimensionality without approximation, this approach is leveraged in the work proposed here.

This paper is partitioned as follows: first the required fundamental formulations for RBC will be introduced, followed by a more detailed development of the technique itself. Next, the

results of recursive Bayesian classification as applied to simulated experiments will be compared to the results obtained by naïve classification. Finally, the relevant conclusions about this method will be made and applications of RBC will be briefly discussed.

3.2 Prediction, Belief Fusion, and Decision-Making

This section is devoted to an overview of the concepts on which RBC is built. These include prediction, belief fusion, and decision-making. Additionally, the formulation of the Gaussian toroid is presented here.

3.2.1 Prediction

Under the RBE framework, prediction is governed by the continuous Chapman-Kolmogorov equation given by:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}. \quad (1)$$

The probability distribution function (PDF) $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ corresponds to belief of the state \mathbf{x}_k based on a prediction model, and $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})$ corresponds to belief based on previous observations. Various simplifications arise from this formulation, one of the best-known of which is the Kalman filter (KF). This method assumes that belief of a state can be well-represented by a Gaussian distribution, so that only a mean vector and covariance matrix must be monitored through prediction and correction stages. For linear dynamic systems with state-space motion models given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{w}, \quad \mathbf{z} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{v}, \quad (2)$$

the KF dictates that the mean in prediction at discrete time step k given the previous time step $k - 1$ is

$$\bar{\mathbf{x}}_{k|k-1} = \mathbf{A}\bar{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_{k-1}. \quad (3)$$

Furthermore, the covariance $\Sigma_{x_k|k-1}$ of the Gaussian representing belief at step k is given by

$$\Sigma_{x_k|k-1} = A\Sigma_{x_{k-1}|k-1}A + \Sigma_{w_{k-1}}. \quad (4)$$

3.2.2 Gaussian Toroid Model

In recent work, formulation for a multi-dimensional Gaussian toroid was developed as a general tool to be used in representing useful non-Gaussian probability distribution functions [15]. Because many rudimentary prediction schemes fit this particular model well, it is important to introduce the Gaussian toroid in this section. A multi-dimensional Gaussian toroid is defined here as an n-shell with Gaussian cross-sectional properties. Figure 1 gives an example of a Gaussian toroid in 2-D belief space. As the figure illustrates, this shape is obtained as the difference between two non-covariant Gaussian distributions with equal means and unequal variances, offering a compact and efficient representation of the desired toroidal shape.

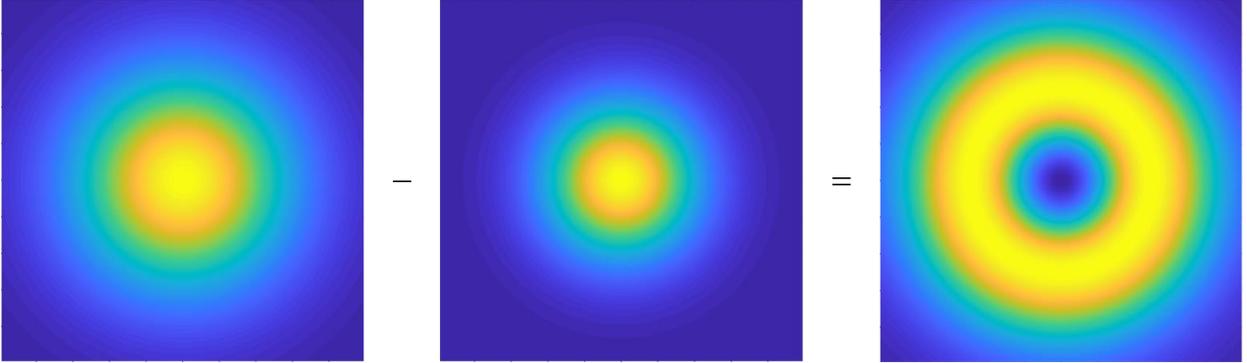


Figure 1. Visual construction of a 2-D Gaussian toroid [15].

The mathematical formulation of the Gaussian toroid is given as follows:

$$\tau(\boldsymbol{\mu}, r) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma_1) - \left(\frac{\sigma_1}{\sigma_2}\right)^N \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma_2) \quad (5)$$

where r is the radius of the toroid (an implicit function of σ_1 and σ_2), $\boldsymbol{\mu}$ is the location of the center of the toroid, and N is the dimensionality of the belief space. Furthermore, it is required that $\Sigma_1 = \sigma_1^2 \mathbf{I}$ and $\Sigma_2 = \sigma_2^2 \mathbf{I}$. The relationship between the toroidal radius and the two variances σ_1^2 and σ_2^2 can be established so that a radius is specified in the creation of a toroid. It can be shown that

$$\sigma_2^2 < \frac{r^2}{\sqrt{2}} \quad (6)$$

and

$$\sigma_1^2 = -\frac{r^2}{2W_0\left(-\frac{r^2}{2\sigma_2^2} \exp\left(-\frac{r^2}{2\sigma_2^2}\right)\right)} \quad (7)$$

where $W_0(x)$ denotes the zeroth branch of the Lambert W function. This branch is defined for $x \geq -e^{-1}$, which gives rise to the constraint of (6).

3.2.3 Belief Fusion

The second stage of recursive Bayesian classification, namely correction, is the fusion of belief from prediction with belief from an observation. The traditional formula for correction in traditional RBE is

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{l(\mathbf{x}_k | \mathbf{z}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{\int l(\mathbf{x}_k | \mathbf{z}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) d\mathbf{x}_k}. \quad (8)$$

With some generalization and reduction, this can lead to the fusion of several state estimates, given by:

$$p(\mathbf{x} | \mathbf{z}_{1:n}) = \frac{\prod_{i=1}^n p(\mathbf{x} | \mathbf{z}_i)}{\int \prod_{i=1}^n p(\mathbf{x} | \mathbf{z}_i) d\mathbf{x}}. \quad (9)$$

Here n describes the number of estimates at a given time step. This means that the belief fusion of n estimates of \mathbf{x} is simply the normalized product of the PDFs corresponding to each estimate. If the constituent PDFs of belief fusion are Gaussian or sums-of-Gaussians, the output will also be Gaussian-like or sum-of-Gaussian-like. The multi-Gaussian belief fusion (MBF) technique of [15] takes advantage of the linear superposition of Gaussians in executing belief fusion fast and in many dimensions. For the case where $n = 2$, each PDF $p(\mathbf{x} | \mathbf{z}_i)$ can be represented as a sum of Gaussians according to:

$$p(\mathbf{x}|\mathbf{z}_1) = \sum_{i=1}^I c_i^{(1)} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^{(1)}, \boldsymbol{\Sigma}_i^{(1)}) \quad (10)$$

and

$$p(\mathbf{x}|\mathbf{z}_2) = \sum_{j=1}^J c_j^{(2)} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j^{(2)}, \boldsymbol{\Sigma}_j^{(2)}). \quad (11)$$

Belief fusion of these two PDFs, then, is given by:

$$p(\mathbf{x}|\mathbf{z}_{1:2}) = \sum_{i=1}^I \sum_{j=1}^J c_i^{(1)} c_j^{(2)} c_{ij}^{(1:2)} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{ij}^{(1:2)}, \boldsymbol{\Sigma}_{ij}^{(1:2)}) \quad (12)$$

where

$$\boldsymbol{\Sigma}_{ij}^{(1:2)} = \left[(\boldsymbol{\Sigma}_i^{(1)})^{-1} + (\boldsymbol{\Sigma}_j^{(2)})^{-1} \right]^{-1}, \quad (13)$$

$$\boldsymbol{\mu}_{ij}^{(1:2)} = \boldsymbol{\Sigma}_{ij}^{(1:2)} \left[(\boldsymbol{\Sigma}_i^{(1)})^{-1} \boldsymbol{\mu}_i^{(1)} + (\boldsymbol{\Sigma}_j^{(2)})^{-1} \boldsymbol{\mu}_j^{(2)} \right], \quad (14)$$

and

$$c_{ij}^{(1:2)} = \frac{1}{\sqrt{|2\pi(\boldsymbol{S}_{ij})|}} \exp\left(-\frac{1}{2}(\mathbf{m}_{ij})^T (\boldsymbol{S}_{ij})^{-1} (\mathbf{m}_{ij})\right) \quad (15)$$

with $\mathbf{m}_{ij} = \boldsymbol{\mu}_i^{(1)} - \boldsymbol{\mu}_j^{(2)}$ and $\boldsymbol{S}_{ij} = \boldsymbol{\Sigma}_i^{(1)} + \boldsymbol{\Sigma}_j^{(2)}$. Correction using MBF is well-suited for a multi-Gaussian predictive model such as the Gaussian toroid.

3.2.4 Decision-Making

The final critical stage of the work presented here is classification. In general, this stage may be referred to as decision-making, but with regards to machine learning or robotic perception it is synonymous with classification. The goal of classification under the RBC scheme is to make full use of the probabilistic information available from prediction and correction. Consequently, a decision-making scheme is chosen such that class probabilities can be assigned to each test case

based on the output PDF of correction. These probabilities are simply the marginalized joint probability of the test PDF (denoted here by $p(\mathbf{x})$ which is really $p(\mathbf{x}|\mathbf{z}_{1:n})$, the output of MBF) with a training PDF $p(w_a|\mathbf{x})$ corresponding to each class w_a :

$$P(w_a) = \int_{\mathbf{x}} p(w_a|\mathbf{x})p(\mathbf{x})d\mathbf{x}. \quad (16)$$

Because this involves the integration of the product of two multi-Gaussian PDFs, the theory developed for MBF gives insight into computing the class probabilities quickly and easily. The scaling constants associated with each Gaussian in (12) correspond to the area beneath their PDFs. This means that the class probabilities are simply the weighting coefficients coming from belief fusion of a test PDF and the training PDFs for each class. Stated mathematically,

$$P(w_a) = \sum_{i=1}^I \sum_{j=1}^J c_i^{(p(w_a|\mathbf{x}))} c_j^{(p(\mathbf{x}))} c_{ij}^{(p(w_a|\mathbf{x})p(\mathbf{x}))}. \quad (17)$$

To enforce that these are proper probabilities such that they sum to one, they are normalized as follows:

$$\tilde{P}(w_a) = \frac{P(w_a)}{\sum_{a=1}^A P(w_a)} \quad (18)$$

so that the final probability that a test item belongs to class w_a is $\tilde{P}(w_a)$. The vector of class probabilities can then be used to inform further action through use of a cost function matrix or the like.

3.3 Recursive Bayesian Classification

This section describes the specifics of recursive Bayesian classification. Figure 2 provides a system diagram of RBC using a Gaussian toroid prediction model and generic multi-Gaussian observation model. The RBC framework follows the structure of recursive Bayesian estimation, but with some critical contributions specific to the perception of evolving targets. First, a multi-

dimensional Gaussian toroid is used to model prediction of abstract features not governed by physical equations of motion. Second, RBC allows for non-Gaussian representation of belief in the high-dimensional feature spaces oftentimes needed for classification, while mitigating computational complexity and allowing for potentially real-time analysis. Finally, the classification technique given here yields probabilistic class information useful for high-level decision-making.

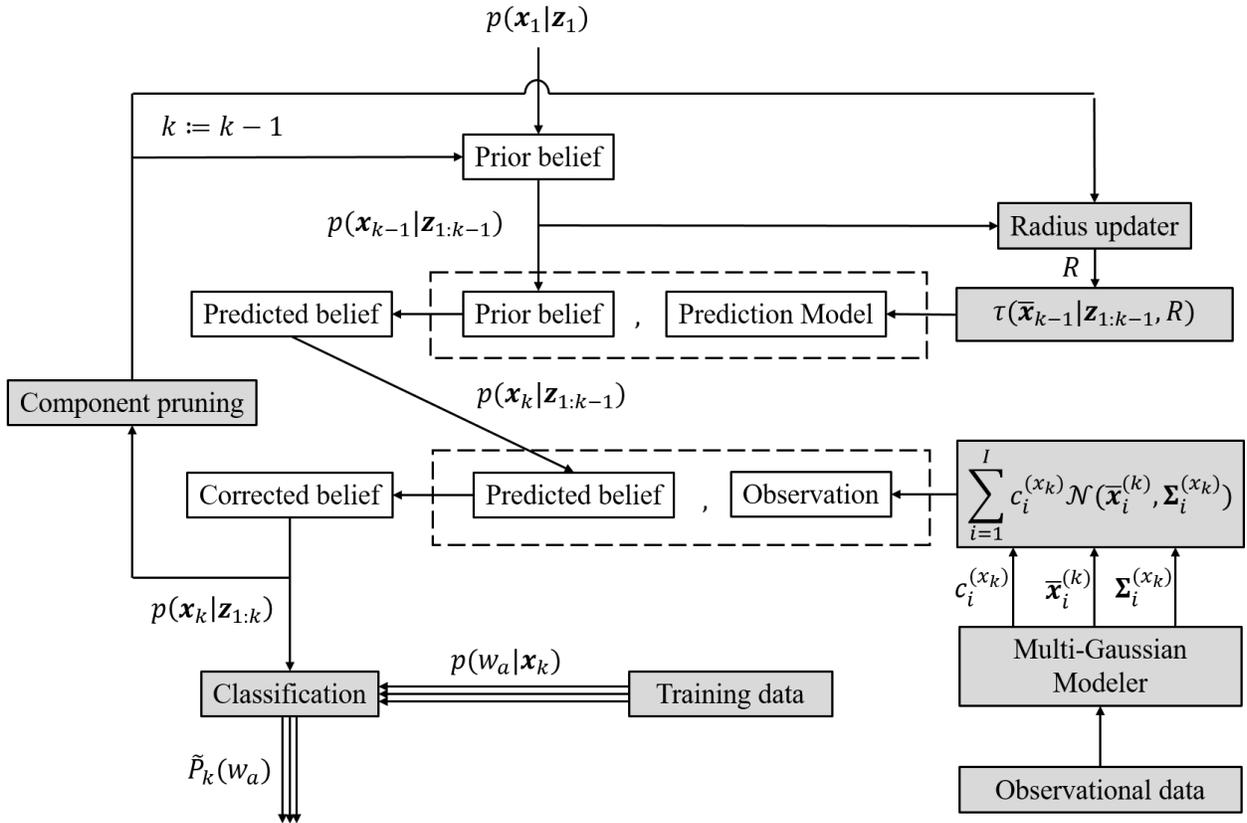


Figure 2. Recursive Bayesian classification diagram. Conceptual elements specifically related to this work's contribution are shaded.

3.3.1 Toroidal Prediction Model

RBC is formulated for any type of belief space which here is an abstract multi-dimensional feature space. Features can be extracted from classification targets by various means not addressed within the scope of this paper. For ease of visualization without loss of appreciation for the

potential complexity of PDFs handled by the approach, arbitrary two-dimensional feature spaces will be used in qualitative figures here.

A single observation of a target can be probabilistically represented in the feature space by a multi-Gaussian PDF. Oftentimes a single Gaussian is sufficient, but in general it may not be. How an observation is modeled as a mixture of Gaussians is beyond the scope of this work; methods exist for fitting a Gaussian mixture to a discrete data array or modeling a sensor's probability characteristics. The aim of observation modeling is simply that the Gaussian mixture representing an observation likelihood accurately represents belief of the target in the feature space.

To create a model of prediction for evolving targets, consider first the case where the target's state at $k - 1$ has no uncertainty and is located at position $\bar{\mathbf{x}}_{k-1}$ in the feature space. The linear Kalman filter would dictate that the mean at step k would be simply $\bar{\mathbf{x}}_k = \mathbf{A}\bar{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$. However, for non-physical classification targets, it is generally not possible to formulate matrices \mathbf{A} and \mathbf{B} . Nevertheless, it stands to reason that the feature space location at step k might be a radial distance R away from the location at step $k - 1$. Since R is only an estimate, it ought to be represented probabilistically. For this reason, a Gaussian toroid is well-suited for predictively modeling evolving targets in the feature space. As such, predicted belief is represented by a Gaussian toroid $\tau(\bar{\mathbf{x}}_{k-1}, R)$ centered at $\bar{\mathbf{x}}_{k-1}$ with radius R .

Let $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ represent the PDF estimating the state at step $k - 1$ given the observations $\mathbf{z}_{1:k-1}$ up to that step. This can be estimated by a multi-Gaussian distribution according to:

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \sum_{i=1}^I c_i^{(x_{k-1})} \mathcal{N}\left(\mathbf{x}; \bar{\mathbf{x}}_i^{(k-1)}, \boldsymbol{\Sigma}_i^{(x_{k-1})}\right). \quad (19)$$

Considering the Kalman solution once more, (4) dictates that $\Sigma_{x_k} = \mathbf{A}\Sigma_{x_{k-1}}\mathbf{A}^T + \Sigma_{w_{k-1}}$ where $\Sigma_{w_{k-1}}$ characterizes process noise. The Gaussian toroid prediction model $\tau(\bar{\mathbf{x}}_{k-1}, R)$ is simply the scaled linear superposition of two Gaussians with covariances given by $\Sigma_1^{(w_{k-1})} = (\sigma_1^{(w_{k-1})})^2 \mathbf{I}$ and $\Sigma_2^{(w_{k-1})} = (\sigma_2^{(w_{k-1})})^2 \mathbf{I}$. This property of superposition is exploited in determining $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$, the PDF expressing predicted belief at step k . The Kalman rules are applied to each of the constituent Gaussians, and the output PDF is a reconstruction of the Gaussian toroid but with Gaussians whose means and covariances have been augmented. In other words, if $\tau(\bar{\mathbf{x}}_{k-1}, R)$ is given by

$$\mathcal{N}(\bar{\mathbf{x}}_{k-1}, \Sigma_1^{(w_{k-1})}) - \left(\frac{\sigma_2^{(w_{k-1})}}{\sigma_1^{(w_{k-1})}} \right)^N \mathcal{N}(\bar{\mathbf{x}}_{k-1}, \Sigma_2^{(w_{k-1})}), \quad (20)$$

then the predicted toroid is given by

$$\mathcal{N}(\bar{\mathbf{x}}_k, \Sigma_1^{(x_k)}) - \sqrt{\frac{|\Sigma_2^{(x_k)}|}{|\Sigma_1^{(x_k)}|}} \mathcal{N}(\bar{\mathbf{x}}_k, \Sigma_2^{(x_k)}) \quad (21)$$

where

$$\bar{\mathbf{x}}_k = \mathbf{A}\bar{\mathbf{x}}_{k-1} + \mathbf{B}\bar{\mathbf{u}}_{k-1} = \bar{\mathbf{x}}_{k-1}, \quad (22)$$

$$\Sigma_1^{(x_k)} = \mathbf{A}\Sigma_{x_{k-1}}\mathbf{A}^T + \Sigma_1^{(w_{k-1})} = \Sigma_{x_{k-1}} + \Sigma_1^{(w_{k-1})}, \quad (23)$$

and

$$\Sigma_2^{(x_k)} = \mathbf{A}\Sigma_{x_{k-1}}\mathbf{A}^T + \Sigma_2^{(w_{k-1})} = \Sigma_{x_{k-1}} + \Sigma_2^{(w_{k-1})}. \quad (24)$$

It is implicit that $\mathbf{A} = \mathbf{I}$ and $\mathbf{B} = \mathbf{0}$ in keeping with the Gaussian toroid prediction model. The new scaling coefficient accompanying the second term of (21) accounts for the introduction of covariance terms by $\Sigma_{x_{k-1}}$ in (23) and (24).

Because $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ is generally multi-Gaussian, the above steps must be applied to each component. The PDF coming from the prediction stage is then:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \sum_{i=1}^I \sum_{j=1}^2 c_{ij}^{(x_k)} \mathcal{N}(\bar{\mathbf{x}}_{ij}^{(k)}, \boldsymbol{\Sigma}_{ij}^{(x_k)}), \quad (25)$$

where

$$c_{ij}^{(x_k)} = c_i^{(x_{k-1})} \begin{cases} 1, & j = 1 \\ -\frac{\sqrt{|\boldsymbol{\Sigma}_{i2}^{(x_k)}|}}{\sqrt{|\boldsymbol{\Sigma}_{i1}^{(x_k)}|}}, & j = 2' \end{cases} \quad (26)$$

$$\bar{\mathbf{x}}_{ij}^{(k)} = \bar{\mathbf{x}}_i^{(k-1)}, \quad (27)$$

and

$$\boldsymbol{\Sigma}_{ij}^{(x_k)} = \boldsymbol{\Sigma}_i^{(x_{k-1})} + \boldsymbol{\Sigma}_j^{(w_{k-1})}. \quad (28)$$

Due to the potential introduction of covariance by $\boldsymbol{\Sigma}_i^{(x_{k-1})}$ terms, this distribution is no longer in general a true toroid; Fig. 3 illustrates this fact for a fictional example. The effect of prediction from a multi-Gaussian PDF is akin to the convolution of the PDF with a Gaussian toroid, as the figure shows. It can be seen in the figure that prediction increases uncertainty, consistent with Bayesian theory.

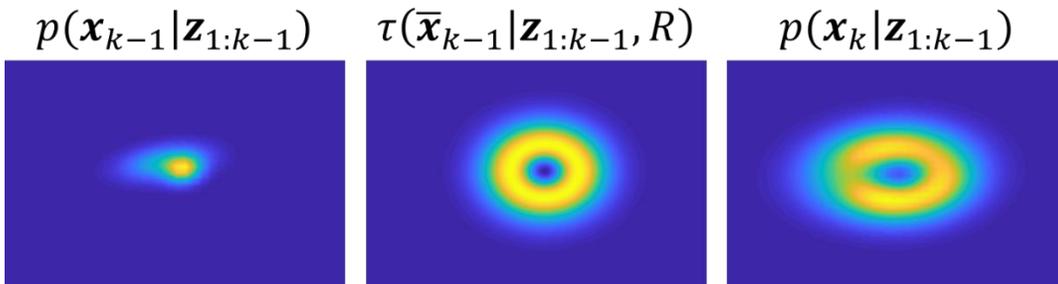


Figure 3. Prediction from multi-Gaussian prior belief. a) Single Gaussian of the form given in (19), b) Gaussian toroid defined by (20), c) output of toroidal prediction as governed by (25)-(28).

3.3.2 Correction and Classification

In order to accurately estimate the location of a target's state the feature space, predicted belief must be fused with observed belief via correction. As section 3.2.3 describes, correction can be executed quickly and in high dimensionality with non-Gaussian PDFs using multi-Gaussian belief fusion (MBF). The goal of this stage is to obtain $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, the PDF corresponding to belief of the target state \mathbf{x}_k given all prior and current observations. Belief corresponding to the k^{th} observation is given by the observation likelihood $l(\mathbf{x}_k|\mathbf{z}_k)$, which is an acceptable PDF and so is treated as $p(\mathbf{x}_k|\mathbf{z}_k)$. It is given by the following:

$$p(\mathbf{x}_k|\mathbf{z}_k) = \sum_{i=1}^I c_i^{(x_k)} \mathcal{N}(\bar{\mathbf{x}}_i^{(k)}, \boldsymbol{\Sigma}_i^{(x_k)}). \quad (29)$$

Fusing this belief with $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ gives the desired $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. This probability distribution may be highly non-Gaussian but is composed of a mixture of Gaussians whose means, covariances, and weights are tracked by the MBF algorithm. Correction is illustrated for a fictitious case in Fig. 4. Just as the prediction stage increased uncertainty in belief, correction decreases uncertainty.

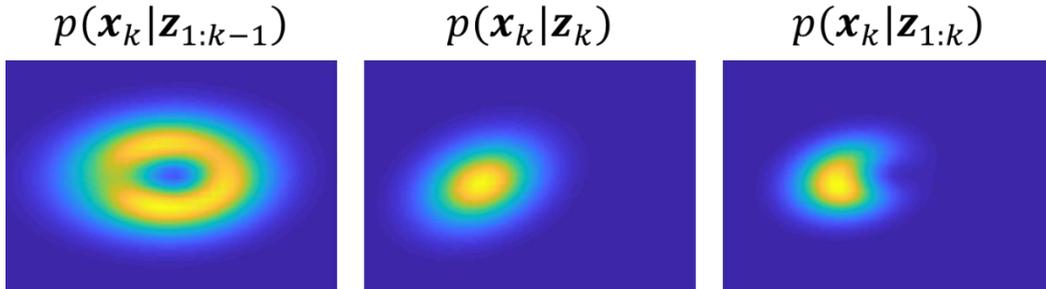


Figure 4. Correction as belief fusion of $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k|\mathbf{z}_k)$.

In order to classify the target, the decision-making algorithm of section 3.2.4 is implemented for fast probabilistic class assignment. The training set may consist of manually classified samples according to relevant categories and features; this data is used to establish the

PDFs $p(w_a|\mathbf{x}_k)$ corresponding to each class w_a . $p(\mathbf{x}_k)$ is taken to simply be the output of correction, and $P(w_a)$ is given by (16) though implemented according to (17) and (18).

After each RBC iteration, the updated belief of the target $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ is fed back to re-initialize belief, and the process is repeated. Because belief at each stage is represented by a multi-Gaussian distribution and each step is formulated for such PDFs, the loop is closed without loss of precision. Figure 5 demonstrates the second prediction step for the running example given in this paper. A disadvantage of the multi-Gaussian approach is that an exponentially increasing number of components is required to completely represent belief at each stage. This computational burden is mitigated by enforcing a pruning stage after each iteration, in which the most minimally-contributing Gaussian components are removed. This can result in small approximation errors, but such issues are normally negligible.

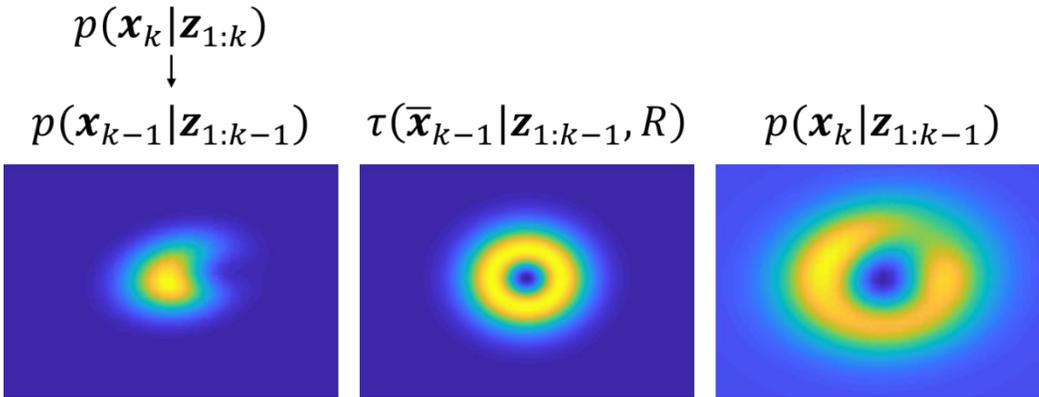


Figure 5. Prediction in the second iteration of RBC for running example.

After each RBC iteration, the toroidal radius's characteristics can be adjusted based on prior knowledge. For example, a running average or regression of past steps' change in location may be used to predict future radii. Furthermore, adjusting the relative covariances of the two toroidal Gaussians within the constraints of eqs. (6) and (7) changes the cross-sectional “thickness” of the resulting toroid without changing its radius. This attribute could be adjusted to better reflect the radial variance in observed data.

3.4 Results

The proposed recursive Bayesian classification framework was verified by means of simulated experiments in a 2-D feature space. These experiments were designed to be representative of the kinds of scenarios for which a Gaussian toroid prediction model would be most useful. For each experiment, $N_{class} = 10$ training classes were generated in a feature space as uni-Gaussian PDFs with randomized means and covariances. Initial belief $p(\mathbf{x}_1|\mathbf{z}_1)$ was given arbitrarily as a two-component multi-Gaussian distribution. For 100 iterations, ground truth was simulated as a sequence of feature space locations. Each ground truth location $\bar{\mathbf{x}}_k^{gt}$ was determined by randomly generating a radius R_k with mean $R = 3.2$ and standard deviation σ_R , and then randomly choosing a location on the circle a distance R_k from $\bar{\mathbf{x}}_k^{gt}$. Two simulation examples are given in Fig. 6.

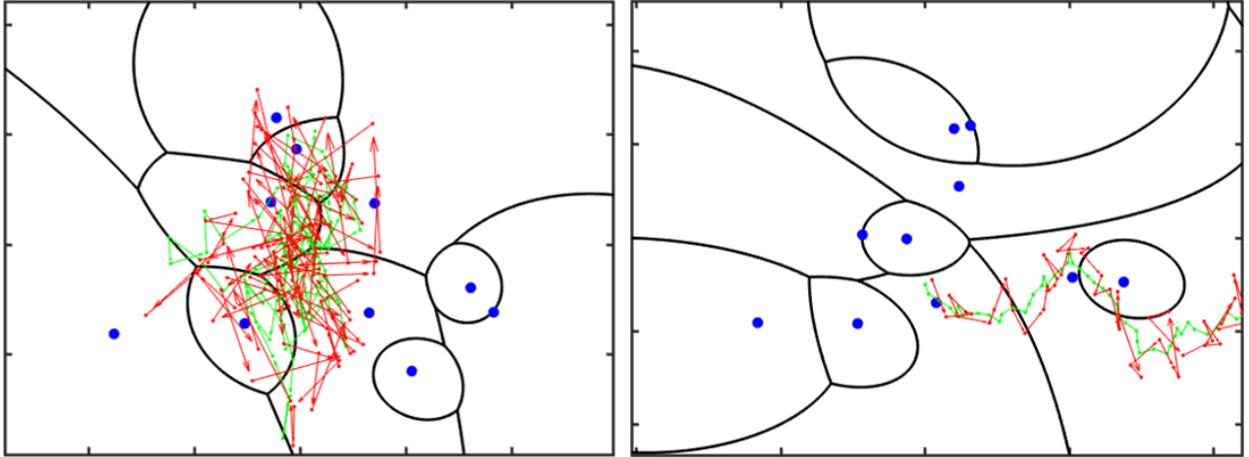


Figure 6. Example simulations without and with directional bias in the evolution model. Blue dots indicate training class centers, black lines correspond to quadratic decision boundaries, green arrows are ground truth trajectories, and red arrows are noisy observations of ground truth.

Ground truth classification was carried out by eqs. (17) and (18) to make full use of all probabilistic information available. Because ground truth uncertainty is zero, the unnormalized probabilities $P(w_a)$ are obtained by evaluating each training class PDF at the ground truth location. Classes were assigned as

$$w = \underset{w_a}{\operatorname{argmax}} \left(\tilde{P}_k(w_a) \right), \quad (30)$$

and these assignments were used to evaluate the performance of the proposed technique.

Observations were simulated as zero-mean Gaussian white noise corruption on the ground truth sequence, though in general non-Gaussian noise is allowable by eq. (29). The covariance of this noise is denoted Σ_v . For each experiment, the performance of RBC in correctly classifying each observed target location was compared to two different naïve approaches. The first of these approaches, a nearest-neighbor method, simply evaluates the Euclidean distance between each observation mean and all the class means, assigning the class with the smallest distance. The second, more intelligent, approach probabilistically classifies observation likelihoods according to eqs. (17) and (18), without implementing prediction and correction. Classes are then assigned according to (30).

Each of the above three classification approaches, two being conventional methods and one being the proposed RBC technique, were compared to ground truth in each experiment by means of a confusion matrix. In order to synthesize these matrices into single comprehensive scores for each experiment, they were multiplied element-wise by a penalty matrix that penalizes off-diagonal elements proportionally to their distance from the diagonal. The resulting matrix was normalized by the number of observations, and the final score given as the sum of the elements in this matrix. One hundred experiments were executed for each combination of Σ_v and σ_R , and the percentage of wins for each method was recorded. This validation metric is necessary due to the randomness in generation of exact simulation variables reflective of inherent uncertainty. Figure 7 shows the convergence of outcomes, justifying reporting the outcomes of experiments repeated 100 times each.

Table 1 gives the results of the parametric study for each of the three processes being compared. NN corresponds to the nearest-neighbor classification approach, while PC represents the basic probabilistic classifier. Four arbitrary parameter

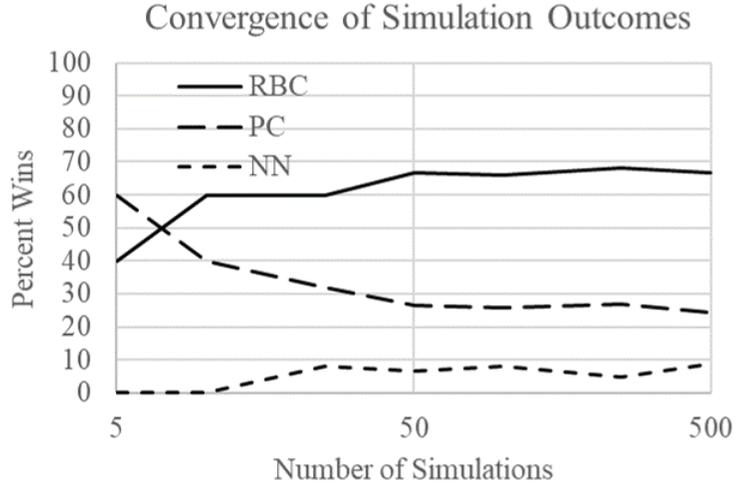


Figure 7. Convergence of outcomes corresponding to parameters given by the second primary column and third row of Table 1.

values were chosen for both Σ_v and σ_R , as the intent is simply to demonstrate general uncertainty-dependent trends for the success of the proposed approach over conventional methods.

Table 1. Parametric Study Results

		Σ_v					
		$\begin{bmatrix} 1^2 & 0.3^2 \\ 0.3^2 & 0.75^2 \end{bmatrix}$			$\begin{bmatrix} 2.6^2 & 1.5^2 \\ 1.5^2 & 3.1^2 \end{bmatrix}$		
		NN	PC	RBC	NN	PC	RBC
σ_R	0.1	0%	63%	37%	2%	24%	74%
	0.25	3%	53%	44%	8%	24%	68%
	0.5	2%	54%	44%	7%	29%	64%
	0.8	1%	55%	44%	4%	22%	74%
		Σ_v					
		$\begin{bmatrix} 3.9^2 & 2.5^2 \\ 2.5^2 & 3.4^2 \end{bmatrix}$			$\begin{bmatrix} 5.1^2 & 3.8^2 \\ 3.8^2 & 5.8^2 \end{bmatrix}$		
		NN	PC	RBC	NN	PC	RBC
σ_R	0.1	2%	22%	76%	9%	12%	79%
	0.25	5%	19%	76%	4%	14%	82%
	0.5	7%	22%	71%	11%	17%	72%
	0.8	6%	19%	75%	7%	22%	71%

As the table indicates, there is no clear trend in the variation of performance as the parameter σ_R is varied. This shows that the uncertainty in the toroidal model does not predictably

affect the proficiency of the proposed method, so long as the general toroidal shape adequately represents true target motion. However, as observational uncertainty increases, RBC outperforms conventional methods. In addition, for sufficiently high Σ_v , the nearest-neighbor classification approach begins to win more often, while the PC method wins less often due to the fact that the latter approach is heavily dependent on observational belief uncertainty. RBC continues to outperform even for high Σ_v because it both adjusts the location of the belief of the state towards ground truth and decreases the uncertainty in this belief.

To assess the computational efficiency of the proposed approach, the previously described simulations were timed both with the core algorithm alone (CA), and with the pruning and decision-making steps (CA+P+DM). To show the multi-dimensional computational advantage, similar simulated experiments were executed in 3-D and 4-D. In Matlab 2018, on a non-dedicated Intel i7 processor running at 2.2 GHz with 12 GB of RAM, the results shown in Table 2 were obtained. Means and standard deviations are given, and each time corresponds to a single iteration of RBC.

Table 2. Timing Study Results (milliseconds)

	2-D	3-D	4-D
CA	2.3±1.2	2.5±1.0	2.7±1.1
CA+P+DM	20.4±3.0	23.2±2.4	25.7±3.0

As the table suggests, the computational requirements of many real-time processes are achievable for 4-D and lower feature spaces, and similar conclusions can be drawn about much higher-dimensional spaces by extrapolation. It is important to note that the RBC algorithm outperforms both of the described naïve classification techniques even more dominantly in higher dimensions, with similar parameter dependencies as were addressed for the 2-D case. Though

higher-dimensional parametric studies took place, they are not reported here in detail for lack of space.

3.5 Conclusions and Applications

The implementation of a recursive Bayesian framework in classification contexts which do not traditionally exploit sequential probabilistic estimation via prediction and correction shows improved performance over conventional techniques, especially for scenarios with substantial uncertainty. Furthermore, the RBC technique introduced here shows strong potential for real-time processing in high-dimensional feature spaces without loss of resolution or inaccuracies introduced by other non-Gaussian estimation techniques. The toroidal prediction model upon which this work is formulated offers the ubiquity mandated by perception problems where only a rudimentary feature-space “motion model” is known.

Recursive Bayesian classification, as it has been formulated here, may have use in a variety of evolving-target perception contexts. For example, in continuous sequential acquisition of road pavement images from a moving vehicle, pavement condition can be estimated by evaluation of features consistent with disrepair. In another vehicular context, these techniques could be implemented in continuous classification, or sub-classification, of roadside objects upon approach. A similar application could be seen in the evaluation of similar but non-identical items moving along an assembly line past a sensor system capable of extracting useful features. The fundamental mathematical formulations for RBE using a Gaussian toroid prediction model could also be used in robotic target tracking, when a target has a poorly defined motion model. SDG

3.6 References

[1] M. Y. Kiang, “A comparative assessment of classification methods”, *Decision Support Systems*, vol. 35, pp. 441-454, 2003.

- [2] F. van der Heijden, R. P. W. Duin, D. de Ridder, and D. M. J. Tax, *Classification, Parameter Estimation, and State Estimation*, Wiley, 2004.
- [3] P. A. Flach and N. Lachiche, “Naïve Bayesian Classification of Structured Data”, *Machine Learning*, vol. 57, pp. 233-269, 2004.
- [4] P. Langley, “Induction of recursive Bayesian classifiers”, *European Conference on Machine Learning*, pp. 153-164, 1993.
- [5] P. Langley and S. Sage, “Induction of Selective Bayesian Classifiers”, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 399-406, 1994.
- [6] S. Särkkä, *Bayesian Filtering and Smoothing*, Cambridge University Press, pp. 51-62, 2013.
- [7] E. Lughofer and O. Buchtala, “Reliable All-Pairs Evolving Fuzzy Classifiers”, *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 4, pp. 625-641, 2013.
- [8] J. Hoffman, T. Darrell, and K. Saenko, “Continuous Manifold Based Adaptation for Evolving Visual Domains”, *Conference on Computer Vision and Pattern Recognition*, pp. 867-874, 2014.
- [9] J. J. Steckenrider, T. Furukawa, *Detection and Classification of Stochastic Features using a Multi-Bayesian Approach*, *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2017.
- [10] R. E. Kalman, *A New Approach to Linear Filtering and Prediction Problems*, *Journal of Basic Engineering*, vol. 82, series D, pp. 35-45, 1960.
- [11] C. Ko and C. Lee, “Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter”, *Energy*, vol. 49, pp. 413-422, 2013.

- [12] O. Hlinka, F. Hlawatsch, and P. M. Djurić, “Distributed Particle Filtering in Agent Networks”, IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 61-81, 2013.
- [13] M. Kumon, D. Kimoto, K. Takami, and T. Furukawa, “Bayesian Non-Field-of-View Target Estimation Incorporating An Acoustic Sensor”, International Conference on Intelligent Robots and Systems, pp. 3425-3432, 2013.
- [14] H. W. Sorenson and D. L. Alspach, “Recursive Bayesian Estimation Using Gaussian Sums,” Automatica, vol. 7, pp. 465-479, 1971.
- [15] J. J. Steckenrider and T. Furukawa, “Fast Multi-Dimensional, Multi-Gaussian Belief Fusion”, Information Fusion, vol. 57, pp. 71-88, 2020.

3.7 Appendix

This appendix did not appear in the original publication of this paper, but is included here to provide supplemental material regarding a more mathematically appropriate use of the Gaussian toroid in the prediction step of a recursive Bayesian process. The following material was adapted from a later publication entitled “Continuum Detection and Predictive-Corrective Classification of Crack Networks”, which was published in the proceedings of the 22nd International Conference on Information Fusion in July, 2019. This publication applied the concepts of RBC towards the real-world problem of road crack classification, with an augmentation to the formulation of prediction via the Gaussian toroid.

In traditional RBE, general prediction is governed by the continuous Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int_{\mathbf{x}} p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}.$$

For a Markovian process with independent increments, the transition PDF $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ can be simplified as follows:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \frac{p(\mathbf{x}_k, \mathbf{x}_{k-1})}{p(\mathbf{x}_{k-1})} = \frac{p(\mathbf{x}_{k-1})p(\mathbf{x}_k - \mathbf{x}_{k-1})}{p(\mathbf{x}_{k-1})} = p(\mathbf{x}_k - \mathbf{x}_{k-1}).$$

By substitution in the Chapman-Kolmogorov equation, it becomes clear that prediction under these constraints is simply the convolution of the PDF describing belief of the current state and the transition PDF $p(\mathbf{x}_k - \mathbf{x}_{k-1})$:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int_{\mathbf{x}} p(\mathbf{x}_k - \mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}.$$

It can be shown that the convolution of two multivariate Gaussian PDFs is a third Gaussian with a mean vector equal to the sum of the constituent mean vectors and covariance matrix equal to the sum of the covariance matrices of the constituents. Therefore, by linear superposition, the convolution of a weighted sum of Gaussians can also be represented by a weighted sum of Gaussians. This is particularly advantageous with regards to prediction using a Gaussian toroid model as is appropriate for stochastic perception of evolving targets.

When belief is represented by multivariate PDFs in a belief space and sequential observations have some correlation that can modeled by an approximate Euclidean distance in that space, the Gaussian toroid becomes a good model for the transition PDF $p(\mathbf{x}_k - \mathbf{x}_{k-1})$. Consequently, if the belief space location of a target at step $k - 1$ is represented by the PDF $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$, predicted belief about the location of the next image k can be given by $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ which is the convolution of $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ and a zero-mean Gaussian toroid, $\tau(\mathbf{0}, R)$. This PDF has the following multi-Gaussian form:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \sum_{i=1}^I \sum_{j=1}^2 c_i^{(x_{k-1})} c_j^{(\tau)} \mathcal{N}(\bar{\mathbf{x}}_{ij}^{(k)}, \boldsymbol{\Sigma}_{ij}^{(x_k)}),$$

where

$$\begin{aligned}\bar{\mathbf{x}}_{ij}^{(k)} &= \bar{\mathbf{x}}_i^{(k-1)}, \\ \boldsymbol{\Sigma}_{ij}^{(x_k)} &= \boldsymbol{\Sigma}_i^{(x_{k-1})} + \boldsymbol{\Sigma}_j^{(\tau)}, \\ c_j^{(\tau)} &= \begin{cases} 1, & j = 1 \\ -\left(\frac{\sigma_1}{\sigma_2}\right)^N, & j = 2' \end{cases}\end{aligned}$$

and σ_2 and σ_1 are given in terms of R by (6) and (7) of this chapter, respectively. This formulation is similar to that given for RBC in section 3.3.1, but with the notable distinction that the weighting coefficient is instead the product of two constituent weighting coefficients, one corresponding to the transition PDF and the other to the prior belief PDF. It is also noteworthy that, in order for this to be a true, normalized multi-Gaussian PDF, the array of weighting coefficients needed to reconstruct $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ must be discretely normalized. RBC using convolution in the prediction step is illustrated in the figure below.

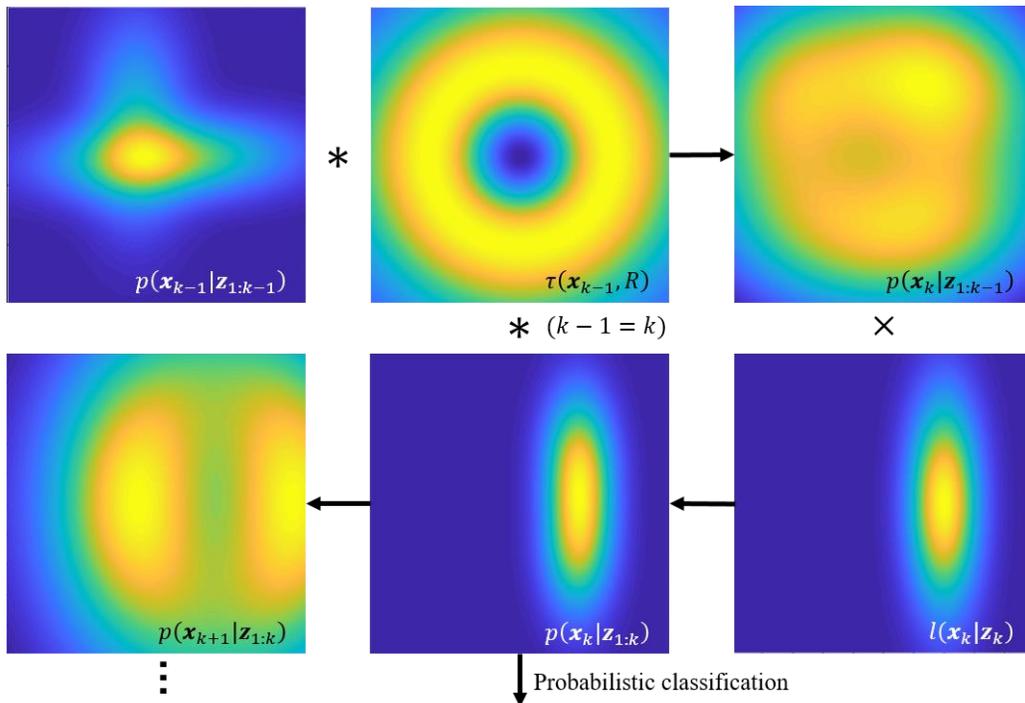


Figure A3.1. Recursive Bayesian classification using convolution in the prediction step.

CHAPTER 4. Introduction to Simultaneous Estimation and Modeling

4.1 Introduction

4.1.1 Background

In probabilistic state estimation by an autonomous robot, from drone target tracking and pursuit to pose estimation of a mechatronic manipulator, existing filtration methods are well-suited to handle uncertainty. Some techniques are even capable of overcoming non-Gaussian and nonlinear processes. Probabilistically equipped autonomous robots often employ a form of recursive Bayesian estimation (RBE) to construct belief and estimate states with inherent uncertainties. State estimation falls apart when parameters informing a predictive model are inaccurately estimated [1]. The work presented here seeks to remedy this issue by incorporating model uncertainty in RBE prediction, as well as introducing a specially formulated model-updating step. This framework, termed Simultaneous Estimation and Modeling (SEAM), is developed in this paper for systems experiencing Gaussian or near-Gaussian state uncertainty.

4.1.2 Related Work

RBE consists of the recursive iteration of three stages [2] which, while known by various terms in different communities, are referred to here as prediction, observation, and correction. Various simplifying approximations lead to different versions of RBE, of which the Kalman filter (KF) [3] and its variants [4,5] are a few. While these techniques handle exclusively Gaussian representation of belief, other approaches allow for non-Gaussian belief by sampling a probability distribution [6,7] or approximating it using Gaussian sums [2]. Furthermore, in order to more accurately enact prediction and correction, various adaptive approaches exist which estimate motion model variance [1] or attempt to handle non-constant system uncertainty [8,9]. Each method has its advantages and disadvantages, which will be discussed here.

Under the linear Kalman filter (LKF) famously proposed by Kalman in 1960 [3], belief about a target or signal is represented by a Gaussian distribution with a mean vector and covariance matrix that are continuously updated. For nonlinear systems, the extended Kalman filter (EKF) linearizes a motion model before then propagating the mean and covariance [4]. The nonlinear unscented Kalman filter (UKF) relies on random sampling and propagation of sigma points around the mean [5], while highly nonlinear motion is addressed by techniques such as the cubature Kalman Filter (CKF) [10]. For data assimilation problems, the ensemble Kalman filter (EnKF) provides a Monte Carlo-type solution to efficient filtering [11]. Although the above methods each demonstrate their efficacy for a specific class of estimation problems, they generally do not address the heightened uncertainty coming from ill-defined model parameters.

Methods for estimation in which motion models do not match the physical system have been in existence for a number of years. One such approach, the joint EKF, assumes the presence of zero-mean Gaussian noise on each of the parameters constituting the motion model and derives the corresponding Kalman gain required to account for this uncertainty [1]. The adaptive Kalman filter (AKF) updates the covariance of assumed additive Gaussian process and/or sensor noise by rescaling it so as to account for uncertainty in the motion model [8,12]. A version of this filter was proposed in recent years which updates both the observation (sensor) and process uncertainty characteristics throughout UKF estimation [13]. Other advanced approaches incorporate error minimization over state transitions to augment traditional Kalman filtering [9,14]. Some frameworks have been proposed to address estimation of specific physical system attributes in real-time [15]. Each of these methods sufficiently minimizes estimation error as compared to naïve Kalman filtering, but because this is generally their sole objective, system parameters and their uncertainties are not improved over time.

In an effort to address model uncertainty, most modern estimation and control techniques employ optimization or Monte Carlo solutions. One such modern technique known as model-predictive control (MPC) is a framework which uses optimization to control dynamic systems with variable model parameters [16]. Robust control and estimation seeks to employ uncertainty characteristics to improve system prediction through MPC. This can be done by various system identification methods [17-19]. However, simple non-optimization model-uncertainty-handling methods are in relatively short supply when it comes to state-space dynamic systems. Some model-estimating frameworks exist, including joint state-parameter estimators [20] and ensemble smoothers for inverse problems [21], though such approaches are designed to apply to more complex problems, using non-Gaussian and optimization methods respectively. Monte Carlo-based approaches [22], sensitivity methods [23], and Bayesian methods [24] are handily capable of parametric model updating, though such processes often address finite-element models and therefore make use of summary statistics like natural frequencies and mode shapes. Composite adaptation approaches [25,26] have been shown to work well for nonlinear state-space model parameter updating, but these high-quality methods have correspondingly high computational demand.

4.1.3 Objectives and Outline

The objectives of this work are twofold: 1) to introduce a more accurate formulation for covariance propagation in state prediction under RBE by incorporating uncertainty in model parameters, and 2) to leverage this uncertainty formulation by presenting a subsequent model correction step which adapts motion model parameters to improve estimation. These contributions are shown to improve estimation performance in comparison with conventional methods when uncertainty in model parameters and system inputs is high. To clarify these points, this paper is

organized as follows: first, the theoretical formulations relevant to the proposed approach are addressed. Following this, the original contributions are described in detail. Then, a results and validation section is given to provide support for these formulations in the context of a simulated parametric study. Finally, a conclusions and future work section summarizes the original contributions and potential future directions of the research.

4.2 Probabilistic Motion Tracking

4.2.1 Robotic Belief

Mathematically, robotic state belief can be defined by probability distribution functions (PDFs). While a PDF can generally take any form, the most common and useful form of a PDF is the Gaussian which is defined, in the general multivariate case, as:

$$\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{\Sigma}_x) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}_x|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{\Sigma}_x^{-1}(\mathbf{x} - \bar{\mathbf{x}})\right). \quad (1)$$

As Eq. (1) shows, the Gaussian (or normal) distribution is completely defined by its first and second moments, $\bar{\mathbf{x}}$ and $\mathbf{\Sigma}_x$. This is a useful property that is leveraged by most estimation frameworks: rather than propagating belief by fully characterizing a PDF throughout estimation, only a mean vector and covariance matrix must be stored and operated upon.

4.2.2 Prediction

The first stage of RBE, prediction, casts belief from step $k - 1$ to step k according to a known motion model of the target or object of interest. Let $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})$ be the PDF representing belief of the estimated state, \mathbf{x} , at step $k - 1$, given all observations \mathbf{z} from step 1 to $k - 1$. Furthermore, let $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ denote the PDF mapping an estimate of the state at step $k - 1$ to the estimate at step k . Prediction is governed by the continuous Chapman-Kolmogorov equation,

which reduces to the following form under the assumption that the process of interest is Markovian:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}_{k-1})d\mathbf{x}_{k-1}. \quad (2)$$

Under the LKF, in order to predict the mean and covariance at step k , a discrete linearized motion model is used:

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad (3a)$$

$$\mathbf{z}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{D}_k\mathbf{u}_k + \mathbf{v}_k, \quad (3b)$$

where \mathbf{w}_{k-1} is some Gaussian process noise with covariance $\Sigma_{\mathbf{w}_{k-1}}$ corrupting the system, and \mathbf{v}_k represents Gaussian sensor noise or observational uncertainty with covariance $\Sigma_{\mathbf{v}_k}$. To obtain the state mean $\bar{\mathbf{x}}_{k|1:k-1}$ and covariance $\Sigma_{\mathbf{x}_{k|1:k-1}}$ in prediction, the Kalman filter dictates the following:

$$\bar{\mathbf{x}}_{k|1:k-1} = \mathbf{A}_{k-1}\bar{\mathbf{x}}_{k-1|1:k-1} + \mathbf{B}_{k-1}\bar{\mathbf{u}}_{k-1}, \quad (4a)$$

$$\Sigma_{\mathbf{x}_{k|1:k-1}} = \mathbf{A}_{k-1}\Sigma_{\mathbf{x}_{k-1|1:k-1}}\mathbf{A}_{k-1}^T + \Sigma_{\mathbf{w}_{k-1}}. \quad (4b)$$

4.2.3 Correction

The correction, or updating, stage of RBE fuses predicted belief, $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$, with the PDF coming from an observation, or measurement, $p(\mathbf{x}_k|\mathbf{z}_k)$. This yields the output of a single RBE iteration, $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, which is obtained by belief fusion, or Bayesian inference:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{l(\mathbf{x}_k|\mathbf{z}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int l(\mathbf{x}_k|\mathbf{z}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k}. \quad (5)$$

In Eq. (5), $l(\mathbf{x}_k|\mathbf{z}_k)$ is the observation likelihood used in place of $p(\mathbf{x}_k|\mathbf{z}_k)$. The Kalman solution yields the following state mean $\bar{\mathbf{x}}_{k|1:k}$ and covariance $\Sigma_{\mathbf{x}_{k|1:k}}$ in the updating step:

$$\bar{\mathbf{x}}_{k|1:k} = \bar{\mathbf{x}}_{k|1:k-1} + \mathbf{K}_k(\bar{\mathbf{z}}_k - \mathbf{C}_k\bar{\mathbf{x}}_{k|1:k-1}), \quad (6a)$$

$$\Sigma_{\mathbf{x}_{k|1:k}} = (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\Sigma_{\mathbf{x}_{k|1:k-1}}, \quad (6b)$$

where the Kalman gain \mathbf{K}_k is given by:

$$\mathbf{K}_k = \boldsymbol{\Sigma}_{x_k|1:k-1} \mathbf{C}_k^T \left(\mathbf{C}_k \boldsymbol{\Sigma}_{x_k|1:k-1} \mathbf{C}_k^T + \boldsymbol{\Sigma}_{v_k} \right)^{-1}. \quad (7)$$

4.2.4 Adaptive Kalman Filter

An AKF is broadly defined as a Kalman filter which updates the covariance of process and sensor noise, $\boldsymbol{\Sigma}_{w_k}$ and $\boldsymbol{\Sigma}_{v_k}$, at each time step in order to better capture uncertainty in the motion and sensor models. One of the most popular approaches [12] is summarized here for comparison in section 4.4. At each time step, following the correction stage, $\boldsymbol{\Sigma}_{v_k}$ and $\boldsymbol{\Sigma}_{w_k}$ are updated as follows:

$$\boldsymbol{\Sigma}_{v_k} = \mathbf{G}_v - \mathbf{C}_k \boldsymbol{\Sigma}_{x_k|1:k-1} \mathbf{C}_k^T, \quad (8a)$$

$$\mathbf{G}_v = \frac{1}{M} \sum_{m=1}^M \left\| \bar{\mathbf{z}}_{k-m} - \mathbf{C}_{k-m} \bar{\mathbf{x}}_{k-m|1:k-m-1} \right\|^2, \quad (8b)$$

$$\boldsymbol{\Sigma}_{w_k} = \sqrt{\alpha} \boldsymbol{\Sigma}_{w_{k-1}}, \quad (8c)$$

$$\alpha = \frac{\text{tr}(\mathbf{G}_v - \boldsymbol{\Sigma}_{v_{k-1}})}{\text{tr}(\mathbf{C}_k \boldsymbol{\Sigma}_{x_k|1:k-1} \mathbf{C}_k^T)}. \quad (8d)$$

As these equations show, the AKF only attempts to correct for uncertainties by adapting noise covariances; model uncertainty is not specifically addressed, and model parameters are not revised throughout estimation. The resulting overconfidence potentially degrades estimation performance when model uncertainty is considerable.

4.3 Simultaneous Estimation and Modeling

4.3.1 Overview

An overview of the SEAM framework proposed in this paper is given by the block diagram in Fig. 1. The system is designed to accommodate two distinct parts: RBE (composed of model-

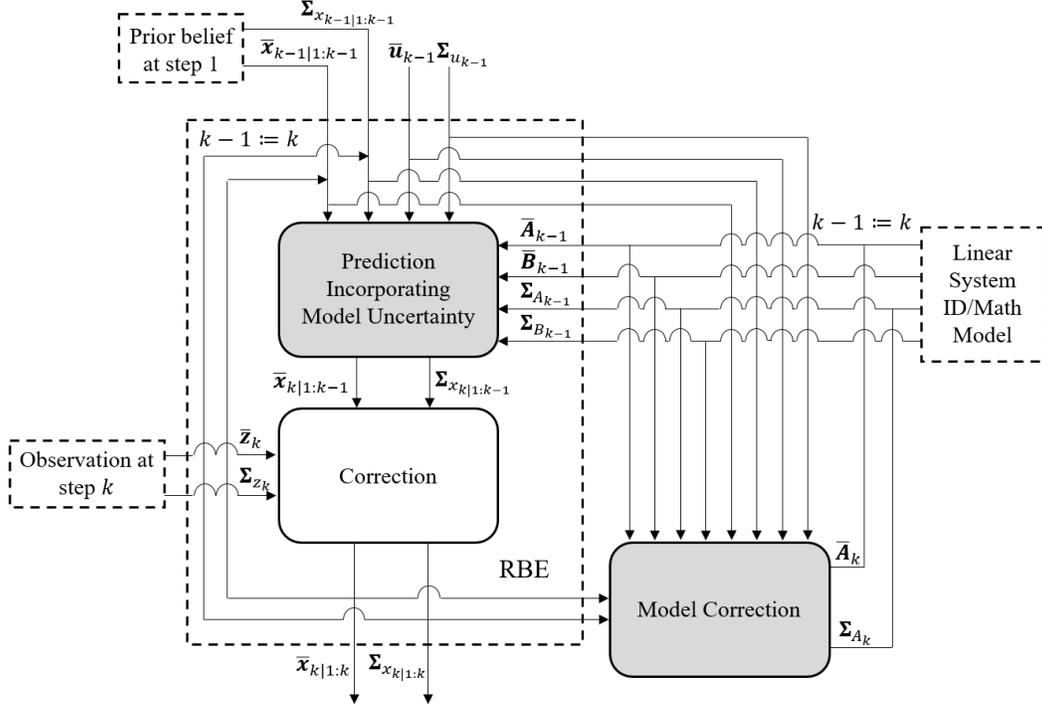


Figure 1. SEAM diagram. Shaded blocks designate original contributions detailed in section 4.3.

estimating prediction and correction) and model correction. The proposed SEAM framework incorporates model uncertainty in prediction, as well as a model correction stage to compensate for increased sensitivity to observational noise. The following sections describe and formulate each of these original contributions in detail.

4.3.2 Model Uncertainty Estimation

A target's discretized motion can generally be described as a function of its previous state, any applied inputs, and various intrinsic parameters as follows:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \Phi_{k-1}), \quad (9)$$

where Φ_{k-1} represents all relevant system model parameters. Assuming such a system can be linearized, the state mean can be predicted by Eq. (4a). Because there may be non-negligible uncertainty in model parameters, estimation based on an assumption of complete confidence in these variables yields poor results. For this reason, a special formulation of covariance propagation

through prediction is proposed. By the rules of error propagation, the mean and covariance of a function containing linear combinations of variables \mathbf{x}_i with uncertainties $\Sigma_{\mathbf{x}_i}$ are given by:

$$f = \sum_{i=1}^I \mathbf{A}_i \mathbf{x}_i \rightarrow \bar{f} = \sum_{i=1}^I \mathbf{A}_i \bar{\mathbf{x}}_i, \Sigma_f = \sum_{i=1}^I \mathbf{A}_i \Sigma_{\mathbf{x}_i} \mathbf{A}_i^T. \quad (10)$$

By expanding a 2-D case (see Appendix A), an improved equation for uncertainty in prediction is given by:

$$\begin{aligned} \Sigma_{\mathbf{x}_{k|1:k-1}} &= \Sigma_{\mathbf{w}_{k-1}} + \bar{\mathbf{A}}_{k-1} \Sigma_{\mathbf{x}_{k-1|1:k-1}} \bar{\mathbf{A}}_{k-1}^T + \bar{\mathbf{B}}_{k-1} \Sigma_{\mathbf{u}_{k-1}} \bar{\mathbf{B}}_{k-1}^T \\ &+ \Sigma_{\mathbf{A}_{k-1}} \mathcal{D}(\bar{\mathbf{x}}_{k-1|1:k-1}^{\circ 2}) \Sigma_{\mathbf{A}_{k-1}}^T + \Sigma_{\mathbf{B}_{k-1}} \mathcal{D}(\bar{\mathbf{u}}_{k-1}^{\circ 2}) \Sigma_{\mathbf{B}_{k-1}}^T. \end{aligned} \quad (11)$$

In the above expression, $\Sigma_{\mathbf{A}_{k-1}}$ and $\Sigma_{\mathbf{B}_{k-1}}$ refer to the matrices containing the standard deviations of each element in \mathbf{A}_{k-1} and \mathbf{B}_{k-1} , and $\bar{\mathbf{A}}_{k-1}$ and $\bar{\mathbf{B}}_{k-1}$ are matrices containing the means. The notation $\bar{\mathbf{x}}_{k-1|1:k-1}^{\circ 2}$ represents the element-wise square. Lastly, the operation $\mathcal{D}(\bullet)$ diagonalizes its vector argument. As can be seen in Eq. (11), the proposed approach includes three terms which do not appear in traditional Kalman filters, an inclusion which can potentially reduce state error in the prediction step. While KFs provide an optimal solution for Gaussian filtering accounting for uncertainty in the state and for process and sensor noise, this framework also accounts for uncertainty in the state-space model parameters and the input vector. The proposed formulation coincides with the KF when the last three terms of Eq. (11) are zero (i.e. model parameters and inputs are known with complete certainty).

A consequence of this method of covariance propagation, since it avoids overconfidence with regards to estimating predicted state uncertainty, is that observation noise can more strongly affect state tracking. This motivates an additional model-updating stage which can improve both the motion model and its uncertainty, thereby enhancing certainty in prediction without the risk of overconfidence.

4.3.3 State Matrix Updating

Although a linearized system can be generally described by the combination of the state matrix \mathbf{A} and the input matrix \mathbf{B} , the intrinsic dynamics of the physical target are governed by the state matrix. Because uncertainty in \mathbf{A} can have the greatest effect on accurate system identification, this will be the focus of motion model correction.

The method proposed to better map a state from time step $k - 1$ to step k seeks to match the predicted maximum likelihood point (MLP) to the corrected MLP. This incorporates belief reinforcement coming from an observation of the state at the new time step. The principle behind state matrix updating

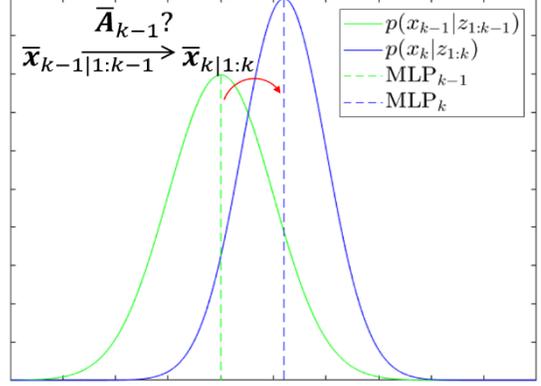


Figure 2. Mapping of 1-D MLP from step $k - 1$ to k .

is demonstrated in Fig. 2. For strictly Gaussian PDFs, such as those dealt with here, the MLP is equivalent to the mean of the distribution. In order to update the estimate $\bar{\mathbf{A}}_{k-1}$ to $\bar{\mathbf{A}}_k$ such that $\bar{\mathbf{x}}_{k|1:k-1}$ and $\bar{\mathbf{x}}_{k|1:k}$ differ minimally, the prediction mean given in Eq. (4a) is augmented by substituting $\bar{\mathbf{x}}_{k|1:k-1}$ with $\bar{\mathbf{x}}_{k|1:k}$ since the latter represents updated belief. The estimated state matrix to be updated, $\bar{\mathbf{A}}_k$, is also substituted in place of $\bar{\mathbf{A}}_{k-1}$, and the following rearrangement is made (the vector $\bar{\mathbf{q}}_k$ is introduced for conciseness in following derivations):

$$\bar{\mathbf{A}}_k \bar{\mathbf{x}}_{k-1|1:k-1} = \bar{\mathbf{x}}_{k|1:k} - \bar{\mathbf{B}}_{k-1} \bar{\mathbf{u}}_{k-1} \equiv \bar{\mathbf{q}}_k. \quad (12)$$

Because the state matrix is $N \times N$ and both $\bar{\mathbf{x}}_{k-1|1:k-1}$ and $\bar{\mathbf{q}}_k$ are $N \times 1$, where N is the dimensionality of the state space, $\bar{\mathbf{A}}_k$ is under-determined. To offer additional constraints required to solve for $\bar{\mathbf{A}}_k$, it is imposed that $\bar{\mathbf{A}}_k$ must differ minimally from its predecessor, $\bar{\mathbf{A}}_{k-1}$. This criterion is governed by the mean-squared-error of all elements in $\bar{\mathbf{A}}$ as compared between steps $k - 1$ and k :

$$MSE = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N (\bar{a}_{ij}^k - \bar{a}_{ij}^{k-1})^2. \quad (13)$$

Equations (12) and (13) can be solved in representative 2-D and 3-D cases, and from these results the updating of an N -D state matrix is given by:

$$\bar{\mathbf{A}}_k = \frac{\bar{\mathbf{A}}_{k-1}(\bar{\mathbf{X}}_{k-1|k-1} - \bar{\mathbf{X}}_{k-1|k-1}(\mathbf{1} - \mathbf{I})\bar{\mathbf{X}}_{k-1|k-1}) + \bar{\mathbf{Q}}_k \bar{\mathbf{X}}_{k-1|k-1}}{\bar{\mathbf{x}}_{k-1|1:k-1}^T \bar{\mathbf{x}}_{k-1|1:k-1}}. \quad (14)$$

(See Appendix for this derivation in the 3-D case.) The variables introduced in Eq. (14) are given as follows:

$$\bar{\mathbf{X}}_{k-1|k-1} = \begin{bmatrix} \bar{\mathbf{x}}^{(1)T} \bar{\mathbf{x}}^{(1)} & 0 & \cdots & 0 \\ 0 & \bar{\mathbf{x}}^{(2)T} \bar{\mathbf{x}}^{(2)} & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{\mathbf{x}}^{(j)T} \bar{\mathbf{x}}^{(j)} \end{bmatrix}, \quad (15a)$$

$$\bar{\mathbf{X}}_{k-1|k-1} = \mathcal{D}(\bar{\mathbf{x}}_{k-1|1:k-1}), \quad (15b)$$

$$\bar{\mathbf{Q}}_k = [\bar{q}_k \quad \bar{q}_k \quad \cdots \quad \bar{q}_k], \quad (15c)$$

$$\mathbf{1} = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}, \quad (15d)$$

and $\bar{\mathbf{x}}^{(j)}$ is $\bar{\mathbf{x}}_{k-1|1:k-1}$ with element j removed. Each of the matrices in Eqs. (15) has dimensionality $N \times N$. Note that the solution given in Eq. (14) yields the traditional least-squares solution for the case where $\bar{\mathbf{A}}_{k-1} = \mathbf{0}$.

In addition to updating the state matrix, it is also necessary to update its standard deviation matrix $\boldsymbol{\Sigma}_A$ according to Eq. (11). $\boldsymbol{\Sigma}_{A_k}$ is derived by applying error propagation to Eq. (14):

$$\boldsymbol{\Sigma}_{A_k} = (\boldsymbol{\Sigma}_{C_1} \mathbf{C}_2^2 + C_1^2 \boldsymbol{\Sigma}_{C_2})^{1/2}, \quad (16a)$$

$$C_1 = \frac{1}{\bar{\mathbf{x}}_{k-1|1:k-1}^T \bar{\mathbf{x}}_{k-1|1:k-1}}, \quad (16b)$$

$$\mathbf{C}_2 = [\bar{\mathbf{A}}_{k-1}(\bar{\mathbf{X}}_{k-1|k-1} - \bar{\mathbf{X}}_{k-1|k-1}(\mathbf{1} - \mathbf{I})\bar{\mathbf{X}}_{k-1|k-1}) + \bar{\mathbf{Q}}_k \bar{\mathbf{X}}_{k-1|k-1}], \quad (16c)$$

$$\mathbf{\Sigma}_{C_1} = \frac{\bar{\mathbf{x}}_{k-1|1:k-1}^T \mathbf{\Sigma}_{x_{k-1|1:k-1}} \bar{\mathbf{x}}_{k-1|1:k-1}}{(\bar{\mathbf{x}}_{k-1|1:k-1}^T \bar{\mathbf{x}}_{k-1|1:k-1})^4}, \quad (16d)$$

$$\mathbf{\Sigma}_{C_2} = \sum_{i=3}^{10} \mathbf{C}_i, \quad (16e)$$

$$\mathbf{C}_3 = \mathbf{\Sigma}_{A_{k-1}}^2 \bar{\mathbf{x}}_{k-1|k-1}^2, \quad (16f)$$

$$\mathbf{C}_4 = \bar{A}_{k-1}^2 \mathbf{\Sigma}_{x_{k-1|k-1}}, \quad (16g)$$

$$\mathbf{C}_5 = \mathbf{\Sigma}_{A_{k-1}}^2 (\bar{\mathbf{X}}_{k-1|k-1} \mathbf{1} \bar{\mathbf{X}}_{k-1|k-1})^2, \quad (16h)$$

$$\mathbf{C}_6 = \bar{A}_{k-1}^2 \left(\mathbf{\Sigma}_{x_{k-1|k-1}} (\mathbf{1} \bar{\mathbf{X}}_{k-1|k-1})^2 + \bar{\mathbf{X}}_{k-1|k-1}^2 \mathbf{1} \mathbf{\Sigma}_{x_{k-1|k-1}} \right). \quad (16i)$$

$$\mathbf{C}_7 = \mathbf{\Sigma}_{A_{k-1}}^2 (\bar{\mathbf{X}}_{k-1|k-1} \bar{\mathbf{X}}_{k-1|k-1})^2, \quad (16j)$$

$$\mathbf{C}_8 = \bar{A}_{k-1}^2 \left(\mathbf{\Sigma}_{x_{k-1|k-1}} \bar{\mathbf{X}}_{k-1|k-1}^2 + \bar{\mathbf{X}}_{k-1|k-1}^2 \mathbf{\Sigma}_{x_{k-1|k-1}} \right). \quad (16k)$$

$$\mathbf{C}_9 = \mathbf{\Sigma}_{Q_k} \bar{\mathbf{X}}_{k-1|k-1}^2, \quad (16l)$$

$$\mathbf{C}_{10} = \bar{Q}_k^2 \mathbf{\Sigma}_{x_{k-1|k-1}}. \quad (16m)$$

With the exception of $\mathbf{\Sigma}_{A_{k-1}}$ and $\mathbf{\Sigma}_{x_{k-1|1:k-1}}$, the elements in each $\mathbf{\Sigma}$ matrix are the variances of the corresponding elements in the mean matrix. Furthermore, each instance of an exponent implies element-wise operation. The uncertainty in $\bar{\mathbf{q}}_k$ is determined by error propagation of $\bar{\mathbf{x}}_{k|1:k} - \bar{\mathbf{B}}_{k-1} \bar{\mathbf{u}}_{k-1}$ in a manner similar to the derivation of Eq. (11):

$$\mathbf{\Sigma}_{q_k} = \mathbf{\Sigma}_{x_{k|1:k}} + \bar{\mathbf{B}}_{k-1} \mathbf{\Sigma}_{u_{k-1}} \bar{\mathbf{B}}_{k-1}^T + \mathbf{\Sigma}_B \mathcal{D}(\bar{\mathbf{u}}_{k-1}^{\circ 2}) \mathbf{\Sigma}_B^T, \quad (17)$$

and the matrix $\mathbf{\Sigma}_{Q_k}$ is populated by only the diagonal variance terms in $\mathbf{\Sigma}_{q_k}$ at their appropriate positions according to the assembly of \bar{Q}_k .

To both prevent diverging values of the elements in $\bar{\mathbf{A}}$ and ensure that $\mathbf{\Sigma}_A$ continually decreases, the state matrix estimates at steps $k-1$ and k are fused by taking their weighted

averages. The final state matrix and its uncertainty characteristic are denoted $\bar{\mathbf{A}}_k'$ and Σ_{A_k}' , whose elements are given below:

$$\bar{a}_{ij}^{k'} = \frac{\bar{a}_{ij}^k \sigma_{a_{ij}^{k-1}}^2 + \bar{a}_{ij}^{k-1} \sigma_{a_{ij}^k}^2}{\sigma_{a_{ij}^{k-1}}^2 + \sigma_{a_{ij}^k}^2}, \quad (18a)$$

$$\sigma_{a_{ij}^k}' = \sqrt{\frac{\sigma_{a_{ij}^{k-1}}^2 \sigma_{a_{ij}^k}^2}{\sigma_{a_{ij}^{k-1}}^2 + \sigma_{a_{ij}^k}^2}}. \quad (18b)$$

After prediction, observation, correction, and model correction, each of which occurs at every time step, the output parameters are fed back to initialize the next RBE step.

As several of the presented equations contain state vector norms in their denominators, it is important to note the behavior of the proposed estimator as a system's state approaches zero. Because Eqs. (18) cause \bar{a}_{ij}^k to be weighted less heavily than \bar{a}_{ij}^{k-1} , the contribution of the model correction stage goes to zero as k tends to infinity. Therefore, any numerically ill-posed computation is avoided. If a state is motionless (i.e. $\bar{\mathbf{x}}_{k-1|1:k-1} = \mathbf{0}$) at any time before k becomes sufficiently large, the problem is trivial since obtaining improved model parameters would be impossible by any method.

4.4 Results and Validation

4.4.1 Simulated Experiment Description

In many dynamic systems composed of multiple moving parts, especially robotic devices, motion often exhibits damped oscillatory characteristics. For this reason, a simple mass-spring-damper (MSD) model was used for simulation in evaluating the performance of SEAM. Fig. 3

illustrates an MSD with mass m , spring constant k , damping constant b , and an arbitrary input displacement $u(t)$. A 2-D continuous state equation describing this system is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k}{m} & \frac{b}{m} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (19)$$

where $x_1 = x(t)$, $x_2 = \dot{x}(t)$, $u_1 = u(t)$, and $u_2 = \dot{u}(t)$. From inspection of Eq. (19), the motion model parameters are given

by $\mathbf{A}_t = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}$ and $\mathbf{B}_t = \begin{bmatrix} 0 & 0 \\ \frac{k}{m} & \frac{b}{m} \end{bmatrix}$. The subscript t

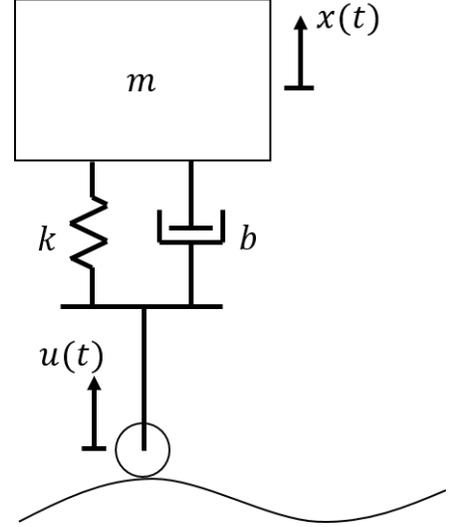


Figure 3. MSD model for validation simulation.

denotes that these are the continuous-time state space matrices, in this example constant for all time t . It is implicitly assumed that $\mathbf{C} = \mathbf{I}$ and $\mathbf{D} = \mathbf{0}$. The mean state matrix \mathbf{A}_t and output matrix \mathbf{B}_t are given by the equations above, using the means of the parameters that compose them. By error propagation of the formulas for \mathbf{A}_t and \mathbf{B}_t , the standard deviation matrices $\mathbf{\Sigma}_{A_t}$ and $\mathbf{\Sigma}_{B_t}$ are found to be:

$$\mathbf{\Sigma}_{A_t} = \mathbf{\Sigma}_{B_t} = \begin{bmatrix} 0 & 0 \\ \frac{\bar{k}}{\bar{m}} \sqrt{\left(\frac{\sigma_k}{\bar{k}}\right)^2 + \left(\frac{\sigma_m}{\bar{m}}\right)^2} & \frac{\bar{b}}{\bar{m}} \sqrt{\left(\frac{\sigma_b}{\bar{b}}\right)^2 + \left(\frac{\sigma_m}{\bar{m}}\right)^2} \end{bmatrix}. \quad (20)$$

The continuous-time state space matrices are transformed into discrete-time for RBE using the relationships $\bar{\mathbf{A}}_k = e^{\Delta t \bar{\mathbf{A}}_t}$ and $\bar{\mathbf{B}}_k = (\bar{\mathbf{A}}_k - \mathbf{I})\bar{\mathbf{A}}_t^{-1}\bar{\mathbf{B}}_t$. To determine the corresponding standard deviation matrices $\mathbf{\Sigma}_A$ and $\mathbf{\Sigma}_B$, the approximations $\bar{\mathbf{A}}_k \cong \mathbf{I} + \Delta t \bar{\mathbf{A}}_t$ and $\bar{\mathbf{B}}_k \cong \Delta t \bar{\mathbf{B}}_t$ are exploited. Uncertainty propagation of these formulas yields, for sufficiently small Δt ,

$$\mathbf{\Sigma}_{A_k} \cong \Delta t \mathbf{\Sigma}_{A_t}, \quad (21a)$$

$$\mathbf{\Sigma}_{B_k} \cong \Delta t \mathbf{\Sigma}_{B_t}. \quad (21b)$$

The elements of the ground-truth motion model matrices were chosen by a Gaussian random variable generator with means and standard deviations given by $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, $\mathbf{\Sigma}_A$ and $\mathbf{\Sigma}_B$. The

same was done for the initial conditions \mathbf{x}_0 . The system was simulated with the parameter values given in Table 1.

Table 1. Constant parameters

\bar{k}	\bar{b}	\bar{m}	Δt	$\bar{\mathbf{x}}_0$	$\Sigma_{\mathbf{x}_0}$	Σ_{w_k}
10	2	5	0.05	$\begin{bmatrix} 10 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.7^2 & 0.11^2 \\ 0.11^2 & 0.9^2 \end{bmatrix}$	$\begin{bmatrix} 0.011^2 & 0.0007^2 \\ 0.0007^2 & 0.016^2 \end{bmatrix}$

Noisy observations were simulated by adding zero-mean Gaussian noise to the ground truth signals, with constant covariance given by Σ_v . State motion $\mathbf{x} = [x(t) \quad \dot{x}(t)]^T$ was tracked from $t = 0$ to 10 seconds ($k = 0$ to 200) with an LKF, AKF, and SEAM both without and with model correction. Parameters were held constant across the three processes in each experiment.

4.4.2 Initial-Value (Unforced Response) Comparisons

To first observe estimation performance on the system's initial value response, the input vector \mathbf{u}_k and its uncertainty Σ_{u_k} were set to zero for all time. To concisely vary the model uncertainty metric, the ratio of the standard deviations σ of the three physical parameters k , b , and m to their means μ was varied from 5% to 25%. The sensor covariance matrix Σ_v was varied over four arbitrarily chosen increasing values. State tracking was simulated for each pair of values 500 times, and the RMSEs of the resulting state trajectories were used to determine a winner in each trial. Table 2 shows the percentage of wins for each estimator and each pair of parameter values. The four processes examined are labeled "LKF", "AKF", "S-" for SEAM without model correction (Eq. (11) only), and "S+" for SEAM with model updating according to Eqs. (12)-(18).

Table 2. Percentage of wins in 5,000 simulations for the Kalman filter, SEAM without model-updating, and SEAM with model-updating across varied sensor and model uncertainty parameters

		Sensor Uncertainty, Σ_v															
		$\begin{bmatrix} 0.1^2 & 0.04^2 \\ 0.04^2 & 0.08^2 \end{bmatrix}$				$\begin{bmatrix} 0.5^2 & 0.15^2 \\ 0.15^2 & 0.3^2 \end{bmatrix}$				$\begin{bmatrix} 0.7^2 & 0.4^2 \\ 0.4^2 & 0.63^2 \end{bmatrix}$				$\begin{bmatrix} 1^2 & 0.6^2 \\ 0.6^2 & 1.2^2 \end{bmatrix}$			
Model Uncertainty, $\frac{\sigma}{\mu}$		LKF	AKF	S-	S+	LKF	AKF	S-	S+	LKF	AKF	S-	S+	LKF	AKF	S-	S+
	5%	65%	6%	1%	28%	61%	0%	7%	32%	60%	1%	9%	30%	54%	1%	17%	28%
	10%	37%	5%	3%	55%	32%	1%	12%	55%	25%	1%	28%	46%	29%	1%	34%	36%
	15%	23%	4%	6%	67%	14%	1%	16%	69%	15%	1%	32%	53%	13%	0%	45%	42%
	20%	14%	2%	8%	76%	8%	0%	17%	75%	9%	1%	30%	60%	9%	1%	49%	41%
	25%	8%	3%	18%	71%	5%	1%	16%	78%	6%	0%	32%	62%	5%	1%	44%	50%

As the table shows, increasing model uncertainty such that the $\frac{\sigma}{\mu}$ ratio exceeds approximately 5% results in SEAM outperforming the LKF and AKF in the majority of cases. Furthermore, even removing the model-updating component of SEAM resulted in better performance than the LKF and AKF in each case where the $\frac{\sigma}{\mu}$ ratio was above 15% and sensor noise was appreciable. It is important to note that, as $\frac{\sigma}{\mu}$ increased, the performance of all estimators decreased; however, the decrease in performance of the SEAM estimators was less than that of the Kalman filters, as expected. The AKF often under-performed, likely due to poor conditioning of the covariance matrices. However, as expected, it did tend to outperform the LKF when sensor and process covariance converged in a reasonable fashion. The LKF tends to outperform when the state-space motion model happens to be accurately estimated at $k = 0$, due to the overconfidence reflected in fast convergence of uncertainty under this technique. Representative plots from two simulations are given in figures 4 and 5.

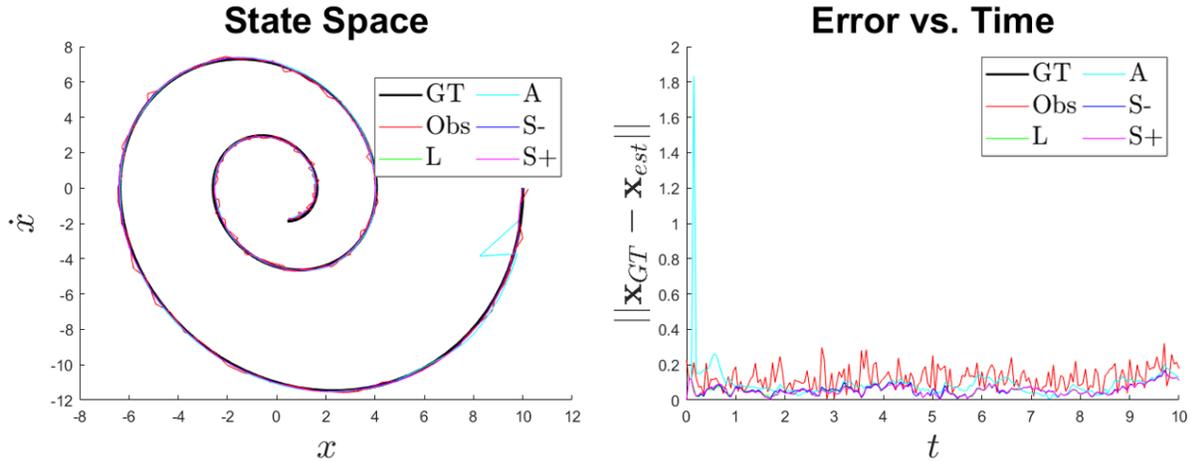


Figure 4. State-space and error plots of simulation where $\frac{\sigma}{\mu}$ and Σ_v correspond to upper-left-most combination in Table 2.

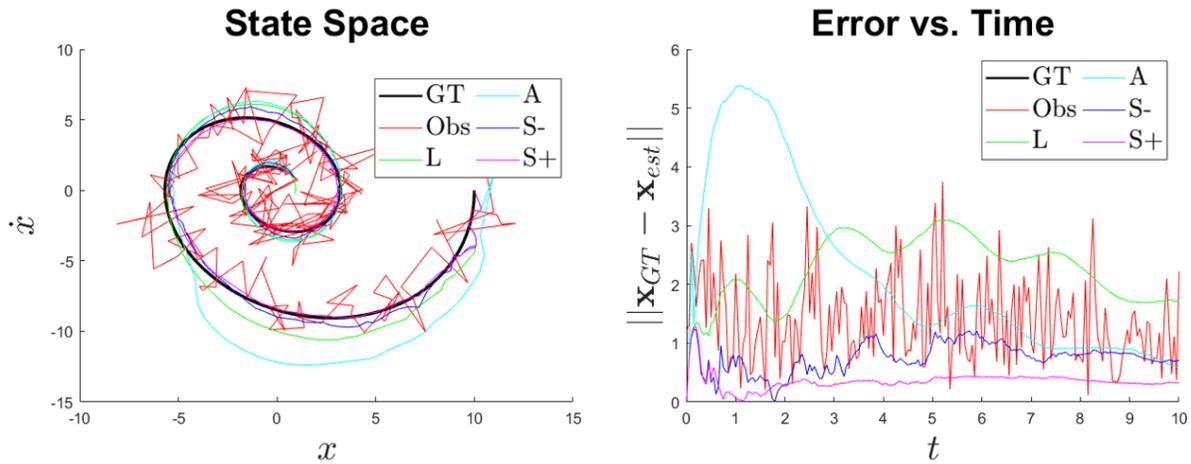


Figure 5. State-space and error plots of simulation where $\frac{\sigma}{\mu}$ and Σ_v correspond to lower-right-most combination in Table 2.

Another revealing characteristic of SEAM can be seen when the elements of the state matrix are plotted over time. Representative plots are shown in Fig. 6. Because the first two elements of the state matrix are constant properties coming from the choice of states and therefore have no uncertainty, only a_{21} and a_{22} are plotted. As the figure shows, the initial estimates of the state matrix elements stochastically improve towards the ground-truth value as a consequence of model updating. These corrected parameters yield better estimation when compared to the LKF and AKF which use a poor estimate of A .

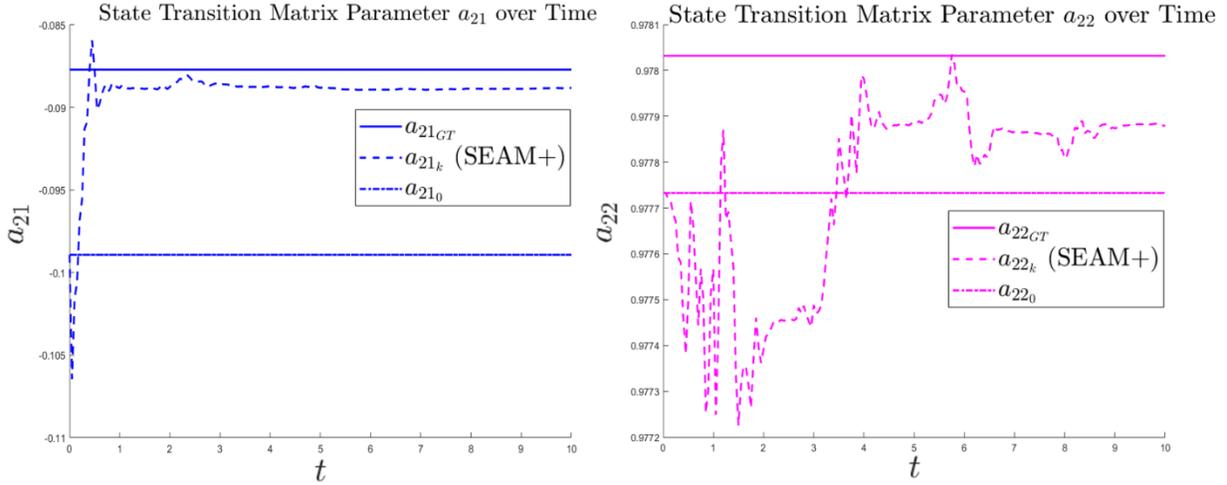


Figure 6. State matrix elements' adjustments due to model correction.

4.4.3 Forced Response Comparisons

Because SEAM is designed to account for uncertainty not only in the state-space matrices \mathbf{A} and \mathbf{B} but also the input vector \mathbf{u} , validation would be incomplete without some forced response comparisons. The input displacement applied to the MSD system was a unity-amplitude sine wave with frequency of 1 Hz.

The above simulations were repeated with the same parameter values as those given in Table 1. This time, rather than varying sensor noise, input noise was arbitrarily varied for the same

$\frac{\sigma}{\mu}$ ratios. Σ_v was held constant at $\begin{bmatrix} 0.5^2 & 0.15^2 \\ 0.15^2 & 0.3^2 \end{bmatrix}$. Table 3 shows the results of these experiments,

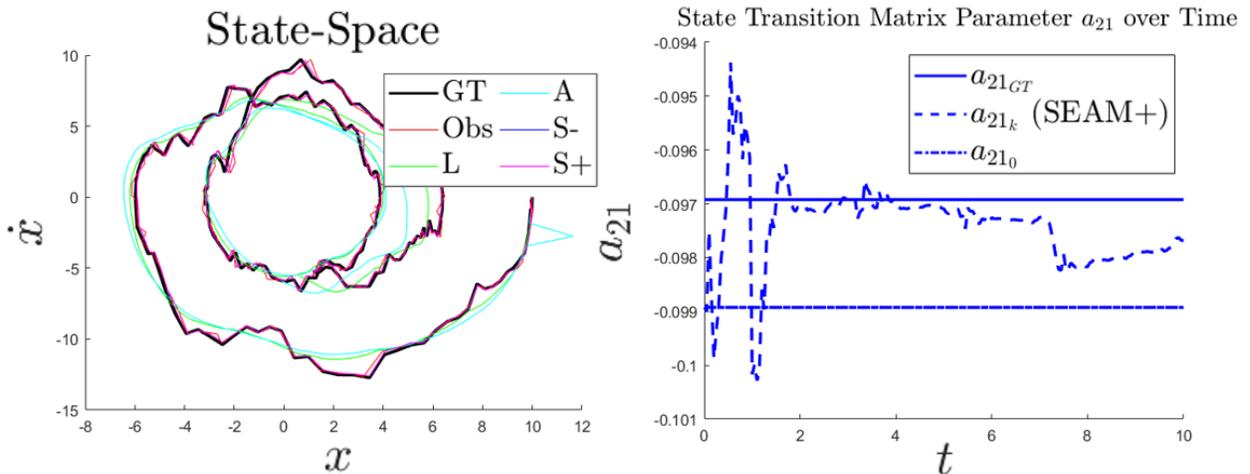


Figure 7. Simulation where $\frac{\sigma}{\mu} = 5\%$, $\Sigma_v = \begin{bmatrix} 0.1^2 & 0.04^2 \\ 0.04^2 & 0.08^2 \end{bmatrix}$, and $\Sigma_u = \begin{bmatrix} 5.4^2 & 2.9^2 \\ 2.9^2 & 4.6^2 \end{bmatrix}$.

where again 500 simulations were executed per combination of parameters. Fig. 7 is included for visualization of a sample simulation.

Table 3. Percentage of wins in 500 simulations for LKF, AKF, and SEAM estimators across varied input and model uncertainty parameters

		Input Uncertainty, Σ_u															
		$\begin{bmatrix} 0.1^2 & 0.04^2 \\ 0.04^2 & 0.08^2 \end{bmatrix}$				$\begin{bmatrix} 0.5^2 & 0.15^2 \\ 0.15^2 & 0.3^2 \end{bmatrix}$				$\begin{bmatrix} 0.7^2 & 0.4^2 \\ 0.4^2 & 0.63^2 \end{bmatrix}$				$\begin{bmatrix} 1^2 & 0.6^2 \\ 0.6^2 & 1.2^2 \end{bmatrix}$			
Model Uncertainty, $\sigma \mu$		LKF	AKF	S-	S+	LKF	AKF	S-	S+	LKF	AKF	S-	S+	LKF	AKF	S-	S+
	5%	31%	1%	33%	35%	1%	1%	48%	50%	0%	0%	61%	39%	0%	0%	57%	43%
	10%	16%	1%	26%	57%	1%	1%	41%	57%	0%	0%	53%	47%	0%	0%	57%	43%
	15%	14%	1%	12%	73%	1%	1%	26%	72%	0%	0%	43%	57%	0%	0%	55%	45%
	20%	5%	0%	12%	83%	1%	0%	19%	80%	0%	0%	35%	65%	0%	0%	59%	41%
	25%	3%	1%	9%	87%	1%	1%	14%	84%	0%	0%	31%	69%	0%	0%	59%	41%

As the table shows, SEAM always outperforms the AKF and LKF when subject to substantial input uncertainty, as the latter two methods over-smooth the estimated signal. A noteworthy observation is that for high Σ_u , the average RMSEs of S- and S+ become identical since convergence of Σ_A has a minimal effect in comparison with the much larger Σ_u . Therefore, adding the model correction stage does not compromise the quality of state estimation as compared to SEAM without model correction.

For completeness in assessing this framework, mention must be made of the computational demand associated with the proposed contributions. The computational burden of the first contribution of Eq. (11) relative to a traditional Kalman filter is negligible. The proposed model correction step of section 4.3.3 adds an average of only 98 microseconds per time step, as the closed-form matrix operations involved are done efficiently and without need for iteration or optimization. An average 500-simulation trial takes only 70 seconds to process on a non-dedicated

Intel i7 processor; considering that each trial covers 10 simulated seconds and a real-time simulation would break even at 5,000 seconds, SEAM can be easily applied online.

A limitation of the proposed framework as demonstrated by these experiments is the inefficiency of model parameter correction. Oftentimes, while model parameters can be *improved*, convergence to ground-truth does not happen quickly. Furthermore, any inaccuracies in the model parameters of the input matrix \mathbf{B} will not be rectified since the formulations of section 4.3.3 apply only to the state matrix. Finally, while SEAM improves estimation relative to conventional methods for high model parameter uncertainty, Table 2 makes it clear that KFs perform better when uncertainty is low, as these filters are more effective at signal smoothing.

4.5 Conclusions and Future Work

This paper has presented a state tracking framework for handling Gaussian belief that properly accounts for uncertainty in the underlying motion model. Because this approach avoids overconfidence in prediction, there is greater sensitivity to sensor noise. As a result, this work proposed a second stage for state matrix correction, which both improves the estimate of motion model parameters and reduces their uncertainty. These two stages symbiotically integrate to comprise the SEAM framework presented and validated in this paper.

As expected in simulation, state tracking performance was considerably improved in circumstances with high system uncertainty. This is because the proposed approach effectively shifts dependence from prediction to observation in a way that accurately reflects a lack of confidence in the predictive model. Furthermore, incorporating a model correction stage resulted in even further improvement in estimation, as this allowed the estimated means of model parameters to be continuously adapted and the associated uncertainties reduced.

The following future research topics arising from this work are currently being investigated. SEAM in the presented form requires Gaussian belief, which may not always be capable of adequately representing state belief [7]. As such, non-Gaussian methods are being developed. Additionally, state tracking may be improved to some extent by the development of an input-matrix correction stage. Further investigation of the capability of SEAM is warranted for highly nonlinear systems [25] where linearization is ineffective. In addition, we postulate that applying SEAM to the observation stage of RBE could improve both state estimation and sensor model parameter estimates. Advances in these areas could ultimately remove the need for initial models altogether, which would be a significant achievement towards a robotic system-learning platform. SDG

4.6 References

- [1] V. A. Bavdekar, R. B. Gopaluni, and S. L. Shah, "Evaluation of Adaptive Extended Kalman Filter Algorithms for State Estimation in Presence of Model-Plant Mismatch," *10th IFAC International Symposium on Dynamics and Control of Process Systems*, Mumbai, India, 2013.
- [2] H. W. Sorenson, D. L. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica*, vol. 7, pp. 465-479, 1971.
- [3] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. D, pp. 35-45, 1960.
- [4] T. Moore, D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," *Advances in Intelligent Systems and Computing: Intelligent Autonomous Systems 13*, vol. 302, pp. 335-348, 2016.

- [5] B. Allotta, A. Caiti, L. Chisci, R. Costanzi, F. Di Corato, C. Fantacci, D. Fenucci, E. Meli, A. Ridolfi, "An unscented Kalman filter based navigation algorithm for autonomous underwater vehicles," *Mechatronics*, vol. 39, pp. 185-195, November 2016.
- [6] Y. Li, M. Coates, "Particle Filtering With Invertible Particle Flow," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4102-4116, August 2017.
- [7] D. S. Kalogerias, A. P. Petropulu, "Grid Based Nonlinear Filtering Revisited: Recursive Estimation & Asymptotic Optimality," *IEEE Transactions on Signal Processing*, vol. 64, no. 16, pp. 4244-4259, August 2016.
- [8] A. Katriniok, D. Abel, "Adaptive EKF-Based Vehicle State Estimation With Online Assessment of Local Observability," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1368-1381, 2016.
- [9] F. Bakhshande, D. Söffker, "Proportional-Integral-Observer: A brief survey with special attention to the actual methods using ACC Benchmark," *IFAC-PapersOnLine*, vol. 48, no. 1, pp. 532-537, 2015.
- [10] A. Sharma, S. C. Srivastava, S. Chakrabarti, "A Cubature Kalman Filter Based Power System Dynamic State Estimator," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 8, pp. 2036-2045, August 2017.
- [11] M. Katzfuss, J. R. Stroud, C. K. Wikle, "Understanding the Ensemble Kalman Filter," *The American Statistician*, vol. 70, no. 4, pp. 350-357, 2016.
- [12] A. Almagbile, J. Wang, W. Ding, "Evaluating the Performances of Adaptive Kalman Filter Methods in GPS/INS Integration," *Journal of Global Positioning Systems*, vol. 9, no. 1, pp. 33-40, 2010.

- [13] W. Li, S. Sun, Y. Jia, J. Du, "Robust unscented Kalman filter with adaptation of process and measurement noise covariances," *Digital Signal Processing*, vol. 48, pp. 93-103, January 2016.
- [14] K. D. Sebesta, N. Boizot, "A Real-Time Adaptive High-Gain EKF, Applied to a Quadcopter Inertial Navigation System," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 495-503, 2014.
- [15] K. Oh, J. Seo, "Inertial Parameter Estimation of an Excavator with Adaptive Updating Rule Using Performance Analysis of Kalman Filter," *International Journal of Control, Automation and Systems*, vol. 16, no. 3, pp. 1226-1238, 2018.
- [16] L. Hewing, J. Kabzan, M. N. Zeilinger, "Cautious Model Predictive Control Using Gaussian Process Regression," *IEEE Transactions on Control Systems Technology*, Early access, 2019.
- [17] B. Shafai, *System Identification and Adaptive Control*, New York, NY, USA: Springer-Verlag, 2019.
- [18] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657-682, 2014.
- [19] K. Yuen, H. Mu, "Real-Time System Identification: An Algorithm for Simultaneous Model Class Selection and Parametric Identification," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, pp. 785-801, 2015.
- [20] V. Stojanovic, N. Nedic, "Joint state and parameter robust estimation of stochastic nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 14, pp. 3058-3074, 2016.
- [21] G. Evensen, "Analysis of iterative ensemble smoothers for solving inverse problems," *Computational Geosciences*, vol. 22, pp. 885-908, 2018.

[22] G. Naevdal, L. M. Johnson, S. I. Aanonsen, E. H. Vefring, “Reservoir Monitoring and Continuous Model Updating Using Ensemble Kalman Filter,” *SPE Annual Technical Conference and Exhibition*, Denver, CO, USA, 2003.

[23] J. E. Mottershead, M. Link, M. I. Friswell, “The sensitivity method in finite element model updating: A tutorial,” *Mechanical Systems and Signal Processing*, vol. 25, no. 7, pp. 2275-2296, 2011.

[24] E. Patelli, Y. Govers, M. Broggi, H. M. Gomes, M. Link, J. E. Mottershead, “Sensitivity or Bayesian model updating: a comparison of techniques using the DLR AIRMOD test data,” *Archive of Applied Mechanics*, vol. 87, no. 5, pp. 905-925, 2017.

[25] Y. Pan, T. Sun, H. Yu, “Composite adaptive dynamic surface control using online recorded data,” *International Journal of Robust and Nonlinear Control*, vol. 26, no. 18, pp. 3921-3936, 2016.

[26] H. Liu, Y. Pan, J. Cao, “Composite Learning Adaptive Dynamic Surface Control of Fractional-Order Nonlinear Systems,” *IEEE Transactions On Cybernetics*, Early access, 2019.

4.7 Appendix

[Section 4.3.2]

Consider the simplest multi-dimensional case where the state, input, and noise vectors are 2-D:

$$\begin{bmatrix} x_1^{k|k-1} \\ x_2^{k|k-1} \end{bmatrix} = \begin{bmatrix} a_{11}^{k-1} & a_{12}^{k-1} \\ a_{21}^{k-1} & a_{22}^{k-1} \end{bmatrix} \begin{bmatrix} x_1^{k-1|k-1} \\ x_2^{k-1|k-1} \end{bmatrix} + \begin{bmatrix} b_{11}^{k-1} & b_{12}^{k-1} \\ b_{21}^{k-1} & b_{22}^{k-1} \end{bmatrix} \begin{bmatrix} u_1^{k-1} \\ u_2^{k-1} \end{bmatrix} + \begin{bmatrix} w_1^{k-1} \\ w_2^{k-1} \end{bmatrix}.$$

By expanding the above equation, the following result is obtained:

$$\begin{bmatrix} x_1^{k|k-1} \\ x_2^{k|k-1} \end{bmatrix} = \begin{bmatrix} a_{11}^{k-1} x_1^{k-1|k-1} + a_{12}^{k-1} x_2^{k-1|k-1} + b_{11}^{k-1} u_1^{k-1} + b_{12}^{k-1} u_2^{k-1} + w_1^{k-1} \\ a_{21}^{k-1} x_1^{k-1|k-1} + a_{22}^{k-1} x_2^{k-1|k-1} + b_{21}^{k-1} u_1^{k-1} + b_{22}^{k-1} u_2^{k-1} + w_2^{k-1} \end{bmatrix}.$$

Following error propagation in 1-D,

$$\begin{bmatrix} \sigma_{x_1^{k|k-1}}^2 \\ \sigma_{x_2^{k|k-1}}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{w_1^{k-1}}^2 + \sum_{i=1}^2 \left((\bar{a}_{1i}^{k-1})^2 \sigma_{x_i^{k-1|k-1}}^2 + (\bar{b}_{1i}^{k-1})^2 \sigma_{u_i^{k-1}}^2 \right) + \sum_{i=1}^2 \left((\bar{x}_i^{k-1|k-1})^2 \sigma_{a_{1i}^{k-1}}^2 + (\bar{u}_i^{k-1})^2 \sigma_{b_{1i}^{k-1}}^2 \right) \\ \sigma_{w_2^{k-1}}^2 + \sum_{j=1}^2 \left((\bar{a}_{2j}^{k-1})^2 \sigma_{x_j^{k-1|k-1}}^2 + (\bar{b}_{2j}^{k-1})^2 \sigma_{u_j^{k-1}}^2 \right) + \sum_{j=1}^2 \left((\bar{x}_j^{k-1|k-1})^2 \sigma_{a_{2j}^{k-1}}^2 + (\bar{u}_j^{k-1})^2 \sigma_{b_{2j}^{k-1}}^2 \right) \end{bmatrix}.$$

With simplification to the matrix form, the above expression becomes:

$$\begin{aligned} \begin{bmatrix} \sigma_{x_1^{k|k-1}}^2 \\ \sigma_{x_2^{k|k-1}}^2 \end{bmatrix} &= \begin{bmatrix} \sigma_{w_1^{k-1}}^2 \\ \sigma_{w_2^{k-1}}^2 \end{bmatrix} + \begin{bmatrix} \bar{a}_{11}^{k-1 2} & \bar{a}_{12}^{k-1 2} \\ \bar{a}_{21}^{k-1 2} & \bar{a}_{22}^{k-1 2} \end{bmatrix} \begin{bmatrix} \sigma_{x_1^{k-1|k-1}}^2 \\ \sigma_{x_2^{k-1|k-1}}^2 \end{bmatrix} + \begin{bmatrix} \bar{b}_{11}^{k-1 2} & \bar{b}_{12}^{k-1 2} \\ \bar{b}_{21}^{k-1 2} & \bar{b}_{22}^{k-1 2} \end{bmatrix} \begin{bmatrix} \sigma_{u_1^{k-1}}^2 \\ \sigma_{u_2^{k-1}}^2 \end{bmatrix} \\ &+ \begin{bmatrix} \sigma_{a_{11}^{k-1}}^2 & \sigma_{a_{12}^{k-1}}^2 \\ \sigma_{a_{21}^{k-1}}^2 & \sigma_{a_{22}^{k-1}}^2 \end{bmatrix} \begin{bmatrix} \bar{x}_1^{k-1|k-1 2} \\ \bar{x}_2^{k-1|k-1 2} \end{bmatrix} + \begin{bmatrix} \sigma_{b_{11}^{k-1}}^2 & \sigma_{b_{12}^{k-1}}^2 \\ \sigma_{b_{21}^{k-1}}^2 & \sigma_{b_{22}^{k-1}}^2 \end{bmatrix} \begin{bmatrix} \bar{u}_1^{k-1 2} \\ \bar{u}_2^{k-1 2} \end{bmatrix}. \end{aligned}$$

The consequence of this approach is that the output variance characteristic $\sigma_{x^{k|k-1}}^2 = \begin{bmatrix} \sigma_{x_1^{k|k-1}}^2 \\ \sigma_{x_2^{k|k-1}}^2 \end{bmatrix}$ is

a 2-by-1 variance vector rather than a full 2-by-2 covariance matrix. Nevertheless, some insight is gained. In a more concise form, this expression can be written as:

$$\begin{aligned} \text{diag}(\Sigma_{x_{k|k-1}}) &= \bar{\mathbf{A}}_{k-1} \circ \bar{\mathbf{A}}_{k-1} \text{diag}(\Sigma_{x_{k-1|k-1}}) + \bar{\mathbf{B}}_{k-1} \circ \bar{\mathbf{B}}_{k-1} \text{diag}(\Sigma_{u_{k-1}}) \\ &+ (\Sigma_A \circ \Sigma_A)(\bar{\mathbf{x}}_{k-1|k-1} \circ \bar{\mathbf{x}}_{k-1|k-1}) + (\Sigma_B \circ \Sigma_B)(\bar{\mathbf{u}}_{k-1} \circ \bar{\mathbf{u}}_{k-1}) + \text{diag}(\Sigma_{w_{k-1}}) \end{aligned}$$

where the notation $\text{diag}(\cdot)$ denotes the extraction of the diagonal of a matrix and \circ indicates the Hadamard element-wise matrix product. A critical note is that the matrices Σ_A and Σ_B now refer to the matrices containing the standard deviations of each element in \mathbf{A} and \mathbf{B} . It was previously shown that the first two terms of this equation have a full-covariance form given by $\bar{\mathbf{A}}_{k-1} \Sigma_{x_{k-1|k-1}} \bar{\mathbf{A}}_{k-1}^T$ and $\bar{\mathbf{B}}_{k-1} \Sigma_{u_{k-1}} \bar{\mathbf{B}}_{k-1}^T$, respectively. The diagonal elements of the 2-by-2 covariance matrices equal the corresponding values of the 2-by-1 variance vector above. With regards to the third and fourth term of the expression, a full-covariance form can be estimated by replacing $\bar{\mathbf{x}}_{k-1|k-1} \circ \bar{\mathbf{x}}_{k-1|k-1}$ and $\bar{\mathbf{u}}_{k-1} \circ \bar{\mathbf{u}}_{k-1}$ with diagonal matrices containing the squared

scalar values of the corresponding vectors at the appropriate diagonal locations. Stated mathematically,

$$\mathcal{D}(\bar{\mathbf{x}}_{k-1|k-1} \circ \bar{\mathbf{x}}_{k-1|k-1}) = \begin{bmatrix} (\bar{x}_{k-1|k-1}^1)^2 & 0 & 0 \\ 0 & (\bar{x}_{k-1|k-1}^2)^2 & 0 \\ 0 & 0 & \ddots \end{bmatrix}$$

where the notation $\mathcal{D}(\cdot)$ casts the vector argument into a diagonal matrix. By doing this, covariance (off-diagonal) elements are introduced while the variance (diagonal) elements remain the same as in the 2-by-1 case.

[Section 4.3.3]

Consider the 3-D representation of equations (15) and (16). Explicitly written, these give:

$$\begin{bmatrix} \bar{a}_{11}^k & \bar{a}_{12}^k & \bar{a}_{13}^k \\ \bar{a}_{21}^k & \bar{a}_{22}^k & \bar{a}_{23}^k \\ \bar{a}_{31}^k & \bar{a}_{32}^k & \bar{a}_{33}^k \end{bmatrix} \begin{bmatrix} \bar{x}_1^{k-1|k-1} \\ \bar{x}_2^{k-1|k-1} \\ \bar{x}_3^{k-1|k-1} \end{bmatrix} = \begin{bmatrix} \bar{q}_1^k \\ \bar{q}_2^k \\ \bar{q}_3^k \end{bmatrix}, \begin{aligned} \bar{a}_{11}^k &= \frac{\bar{q}_1^k}{\bar{x}_1^{k-1|k-1}} - \bar{a}_{12}^k \frac{\bar{x}_2^{k-1|k-1}}{\bar{x}_1^{k-1|k-1}} - \bar{a}_{13}^k \frac{\bar{x}_3^{k-1|k-1}}{\bar{x}_1^{k-1|k-1}} \\ \bar{a}_{22}^k &= \frac{\bar{q}_2^k}{\bar{x}_2^{k-1|k-1}} - \bar{a}_{21}^k \frac{\bar{x}_1^{k-1|k-1}}{\bar{x}_2^{k-1|k-1}} - \bar{a}_{23}^k \frac{\bar{x}_3^{k-1|k-1}}{\bar{x}_2^{k-1|k-1}}, \\ \bar{a}_{33}^k &= \frac{\bar{q}_3^k}{\bar{x}_3^{k-1|k-1}} - \bar{a}_{31}^k \frac{\bar{x}_1^{k-1|k-1}}{\bar{x}_3^{k-1|k-1}} - \bar{a}_{32}^k \frac{\bar{x}_2^{k-1|k-1}}{\bar{x}_3^{k-1|k-1}} \end{aligned}$$

$$\begin{aligned} MSE &= \frac{1}{9} \left[(\bar{a}_{11}^k - \bar{a}_{11}^{k-1})^2 + (\bar{a}_{12}^k - \bar{a}_{12}^{k-1})^2 + (\bar{a}_{13}^k - \bar{a}_{13}^{k-1})^2 + (\bar{a}_{21}^k - \bar{a}_{21}^{k-1})^2 + (\bar{a}_{22}^k - \bar{a}_{22}^{k-1})^2 \right. \\ &\quad \left. + (\bar{a}_{23}^k - \bar{a}_{23}^{k-1})^2 + (\bar{a}_{31}^k - \bar{a}_{31}^{k-1})^2 + (\bar{a}_{32}^k - \bar{a}_{32}^{k-1})^2 + (\bar{a}_{33}^k - \bar{a}_{33}^{k-1})^2 \right]. \end{aligned}$$

Substitution of the three first equations into the MSE equation yields the following expression after grouping by powers of \bar{a}_{ij}^k :

$$\begin{aligned} MSE &= \frac{1}{9} \left[C_1 \bar{a}_{12}^k{}^2 + C_2 \bar{a}_{12}^k + C_3 \bar{a}_{12}^k \bar{a}_{13}^k + C_4 \bar{a}_{13}^k + C_5 \bar{a}_{13}^k{}^2 + C_6 \bar{a}_{21}^k{}^2 + C_7 \bar{a}_{21}^k + C_8 \bar{a}_{21}^k \bar{a}_{23}^k \right. \\ &\quad \left. + C_9 \bar{a}_{23}^k + C_{10} \bar{a}_{23}^k{}^2 + C_{11} \bar{a}_{31}^k{}^2 + C_{12} \bar{a}_{31}^k + C_{13} \bar{a}_{31}^k \bar{a}_{32}^k + C_{14} \bar{a}_{32}^k + C_{15} \bar{a}_{32}^k{}^2 \right. \\ &\quad \left. + C_{16} \right], \end{aligned}$$

$$\begin{aligned}
C_1 &= \left(\left(\frac{\bar{x}_2^{k-1|k-1}}{\bar{x}_1^{k-1|k-1}} \right)^2 + 1 \right), C_2 = 2 \left(\bar{a}_{11}^{k-1} \frac{\bar{x}_2^{k-1|k-1}}{\bar{x}_1^{k-1|k-1}} - \bar{q}_1^k \frac{\bar{x}_2^{k-1|k-1}}{\bar{x}_1^{k-1|k-1}^2} - \bar{a}_{12}^{k-1} \right), \\
C_5 &= \left(\left(\frac{\bar{x}_3^{k-1|k-1}}{\bar{x}_1^{k-1|k-1}} \right)^2 + 1 \right), C_4 = 2 \left(\bar{a}_{11}^{k-1} \frac{\bar{x}_3^{k-1|k-1}}{\bar{x}_1^{k-1|k-1}} - \bar{q}_1^k \frac{\bar{x}_3^{k-1|k-1}}{\bar{x}_1^{k-1|k-1}^2} - \bar{a}_{13}^{k-1} \right), \\
C_6 &= \left(\left(\frac{\bar{x}_1^{k-1|k-1}}{\bar{x}_2^{k-1|k-1}} \right)^2 + 1 \right), C_7 = 2 \left(\bar{a}_{22}^{k-1} \frac{\bar{x}_1^{k-1|k-1}}{\bar{x}_2^{k-1|k-1}} - \bar{q}_2^k \frac{\bar{x}_1^{k-1|k-1}}{\bar{x}_2^{k-1|k-1}^2} - \bar{a}_{21}^{k-1} \right), \\
C_{10} &= \left(\left(\frac{\bar{x}_3^{k-1|k-1}}{\bar{x}_2^{k-1|k-1}} \right)^2 + 1 \right), C_9 = 2 \left(\bar{a}_{22}^{k-1} \frac{\bar{x}_3^{k-1|k-1}}{\bar{x}_2^{k-1|k-1}} - \bar{q}_2^k \frac{\bar{x}_3^{k-1|k-1}}{\bar{x}_2^{k-1|k-1}^2} - \bar{a}_{23}^{k-1} \right), \\
C_{11} &= \left(\left(\frac{\bar{x}_1^{k-1|k-1}}{\bar{x}_3^{k-1|k-1}} \right)^2 + 1 \right), C_{12} = 2 \left(\bar{a}_{33}^{k-1} \frac{\bar{x}_1^{k-1|k-1}}{\bar{x}_3^{k-1|k-1}} - \bar{q}_3^k \frac{\bar{x}_1^{k-1|k-1}}{\bar{x}_3^{k-1|k-1}^2} - \bar{a}_{31}^{k-1} \right), \\
C_{15} &= \left(\left(\frac{\bar{x}_2^{k-1|k-1}}{\bar{x}_3^{k-1|k-1}} \right)^2 + 1 \right), C_{14} = 2 \left(\bar{a}_{33}^{k-1} \frac{\bar{x}_2^{k-1|k-1}}{\bar{x}_3^{k-1|k-1}} - \bar{q}_3^k \frac{\bar{x}_2^{k-1|k-1}}{\bar{x}_3^{k-1|k-1}^2} - \bar{a}_{32}^{k-1} \right), \\
C_3 &= 2 \frac{\bar{x}_2^{k-1|k-1} \bar{x}_3^{k-1|k-1}}{\bar{x}_1^{k-1|k-1}^2}, C_8 = 2 \frac{\bar{x}_1^{k-1|k-1} \bar{x}_3^{k-1|k-1}}{\bar{x}_2^{k-1|k-1}^2}, C_{13} = 2 \frac{\bar{x}_1^{k-1|k-1} \bar{x}_2^{k-1|k-1}}{\bar{x}_3^{k-1|k-1}^2}, \\
C_{16} &= \left(\frac{\bar{q}_1^k}{\bar{x}_1^{k-1|k-1}} - \bar{a}_{11}^{k-1} \right)^2 + \left(\frac{\bar{q}_2^k}{\bar{x}_2^{k-1|k-1}} - \bar{a}_{22}^{k-1} \right)^2 + \left(\frac{\bar{q}_3^k}{\bar{x}_3^{k-1|k-1}} - \bar{a}_{33}^{k-1} \right)^2 \\
&\quad + \bar{a}_{12}^{k-1}{}^2 + \bar{a}_{13}^{k-1}{}^2 + \bar{a}_{21}^{k-1}{}^2 + \bar{a}_{23}^{k-1}{}^2 + \bar{a}_{31}^{k-1}{}^2 + \bar{a}_{32}^{k-1}{}^2.
\end{aligned}$$

Setting these to zero yields the following sets of linear equations:

$$\begin{bmatrix} 2C_1 & C_3 \\ C_3 & 2C_5 \end{bmatrix} \begin{bmatrix} \bar{a}_{12}^k \\ \bar{a}_{13}^k \end{bmatrix} = \begin{bmatrix} -C_2 \\ -C_4 \end{bmatrix},$$

$$\begin{bmatrix} 2C_6 & C_8 \\ C_8 & 2C_{10} \end{bmatrix} \begin{bmatrix} \bar{a}_{21}^k \\ \bar{a}_{23}^k \end{bmatrix} = \begin{bmatrix} -C_7 \\ -C_9 \end{bmatrix},$$

$$\begin{bmatrix} 2C_{11} & C_{13} \\ C_{13} & 2C_{15} \end{bmatrix} \begin{bmatrix} \bar{a}_{31}^k \\ \bar{a}_{32}^k \end{bmatrix} = \begin{bmatrix} -C_{12} \\ -C_{14} \end{bmatrix}.$$

Solving for the variables of interest,

$$\begin{aligned} \bar{a}_{12}^k &= \frac{-2C_2C_5 + C_3C_4}{4C_1C_5 - C_3^2}, & \bar{a}_{13}^k &= \frac{-2C_1C_4 + C_3C_2}{4C_1C_5 - C_3^2}, & \bar{a}_{21}^k &= \frac{-2C_{10}C_7 + C_8C_9}{4C_6C_{10} - C_8^2}, \\ \bar{a}_{23}^k &= \frac{-2C_6C_9 + C_8C_7}{4C_6C_{10} - C_8^2}, & \bar{a}_{31}^k &= \frac{-2C_{15}C_{12} + C_{13}C_{14}}{4C_{11}C_{15} - C_{13}^2}, & \bar{a}_{32}^k &= \frac{-2C_{11}C_{14} + C_{13}C_{12}}{4C_{11}C_{15} - C_{13}^2}. \end{aligned}$$

By substitution, cancellation, and grouping, these become:

$$\begin{aligned} \bar{a}_{12}^k &= \frac{(\bar{x}_1^{k-1|k-1^2} + \bar{x}_3^{k-1|k-1^2})\bar{a}_{12}^{k-1} - \bar{x}_1^{k-1|k-1}\bar{x}_2^{k-1|k-1}\bar{a}_{11}^{k-1} - \bar{x}_3^{k-1|k-1}\bar{x}_2^{k-1|k-1}\bar{a}_{13}^{k-1} + \bar{x}_2^{k-1|k-1}\bar{q}_1^k}{\bar{x}_1^{k-1|k-1^2} + \bar{x}_2^{k-1|k-1^2} + \bar{x}_3^{k-1|k-1^2}}, \\ \bar{a}_{13}^k &= \frac{(\bar{x}_1^{k-1|k-1^2} + \bar{x}_2^{k-1|k-1^2})\bar{a}_{13}^{k-1} - \bar{x}_1^{k-1|k-1}\bar{x}_3^{k-1|k-1}\bar{a}_{11}^{k-1} - \bar{x}_2^{k-1|k-1}\bar{x}_3^{k-1|k-1}\bar{a}_{12}^{k-1} + \bar{x}_3^{k-1|k-1}\bar{q}_1^k}{\bar{x}_1^{k-1|k-1^2} + \bar{x}_2^{k-1|k-1^2} + \bar{x}_3^{k-1|k-1^2}}, \\ \bar{a}_{21}^k &= \frac{(\bar{x}_2^{k-1|k-1^2} + \bar{x}_3^{k-1|k-1^2})\bar{a}_{21}^{k-1} - \bar{x}_2^{k-1|k-1}\bar{x}_1^{k-1|k-1}\bar{a}_{22}^{k-1} - \bar{x}_3^{k-1|k-1}\bar{x}_1^{k-1|k-1}\bar{a}_{23}^{k-1} + \bar{x}_1^{k-1|k-1}\bar{q}_2^k}{\bar{x}_1^{k-1|k-1^2} + \bar{x}_2^{k-1|k-1^2} + \bar{x}_3^{k-1|k-1^2}}, \\ \bar{a}_{23}^k &= \frac{(\bar{x}_1^{k-1|k-1^2} + \bar{x}_2^{k-1|k-1^2})\bar{a}_{23}^{k-1} - \bar{x}_1^{k-1|k-1}\bar{x}_3^{k-1|k-1}\bar{a}_{21}^{k-1} - \bar{x}_2^{k-1|k-1}\bar{x}_3^{k-1|k-1}\bar{a}_{22}^{k-1} + \bar{x}_3^{k-1|k-1}\bar{q}_2^k}{\bar{x}_1^{k-1|k-1^2} + \bar{x}_2^{k-1|k-1^2} + \bar{x}_3^{k-1|k-1^2}}, \\ \bar{a}_{31}^k &= \frac{(\bar{x}_2^{k-1|k-1^2} + \bar{x}_3^{k-1|k-1^2})\bar{a}_{31}^{k-1} - \bar{x}_2^{k-1|k-1}\bar{x}_1^{k-1|k-1}\bar{a}_{32}^{k-1} - \bar{x}_3^{k-1|k-1}\bar{x}_1^{k-1|k-1}\bar{a}_{33}^{k-1} + \bar{x}_1^{k-1|k-1}\bar{q}_3^k}{\bar{x}_1^{k-1|k-1^2} + \bar{x}_2^{k-1|k-1^2} + \bar{x}_3^{k-1|k-1^2}}, \\ \bar{a}_{32}^k &= \frac{(\bar{x}_1^{k-1|k-1^2} + \bar{x}_3^{k-1|k-1^2})\bar{a}_{32}^{k-1} - \bar{x}_1^{k-1|k-1}\bar{x}_2^{k-1|k-1}\bar{a}_{31}^{k-1} - \bar{x}_3^{k-1|k-1}\bar{x}_2^{k-1|k-1}\bar{a}_{33}^{k-1} + \bar{x}_2^{k-1|k-1}\bar{q}_3^k}{\bar{x}_1^{k-1|k-1^2} + \bar{x}_2^{k-1|k-1^2} + \bar{x}_3^{k-1|k-1^2}}. \end{aligned}$$

In order to prove that the above solutions minimize the mean-squared-error, the multi-variable second derivative test is carried out. The Hessian matrix of the MSE is given as:

$$\mathbf{H}_{MSE} = \begin{bmatrix} \frac{\partial^2 MSE}{\partial \bar{a}_{12}^k{}^2} & \frac{\partial^2 MSE}{\partial \bar{a}_{13}^k \partial \bar{a}_{12}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{21}^k \partial \bar{a}_{12}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{23}^k \partial \bar{a}_{12}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{31}^k \partial \bar{a}_{12}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{32}^k \partial \bar{a}_{12}^k} \\ \frac{\partial^2 MSE}{\partial \bar{a}_{12}^k \partial \bar{a}_{13}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{13}^k{}^2} & \frac{\partial^2 MSE}{\partial \bar{a}_{21}^k \partial \bar{a}_{13}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{23}^k \partial \bar{a}_{13}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{31}^k \partial \bar{a}_{13}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{32}^k \partial \bar{a}_{13}^k} \\ \frac{\partial^2 MSE}{\partial \bar{a}_{12}^k \partial \bar{a}_{21}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{13}^k \partial \bar{a}_{21}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{21}^k{}^2} & \frac{\partial^2 MSE}{\partial \bar{a}_{23}^k \partial \bar{a}_{21}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{31}^k \partial \bar{a}_{21}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{32}^k \partial \bar{a}_{21}^k} \\ \frac{\partial^2 MSE}{\partial \bar{a}_{12}^k \partial \bar{a}_{23}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{13}^k \partial \bar{a}_{23}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{21}^k \partial \bar{a}_{23}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{23}^k{}^2} & \frac{\partial^2 MSE}{\partial \bar{a}_{31}^k \partial \bar{a}_{23}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{32}^k \partial \bar{a}_{23}^k} \\ \frac{\partial^2 MSE}{\partial \bar{a}_{12}^k \partial \bar{a}_{31}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{13}^k \partial \bar{a}_{31}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{21}^k \partial \bar{a}_{31}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{23}^k \partial \bar{a}_{31}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{31}^k{}^2} & \frac{\partial^2 MSE}{\partial \bar{a}_{32}^k \partial \bar{a}_{31}^k} \\ \frac{\partial^2 MSE}{\partial \bar{a}_{12}^k \partial \bar{a}_{32}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{13}^k \partial \bar{a}_{32}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{21}^k \partial \bar{a}_{32}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{23}^k \partial \bar{a}_{32}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{31}^k \partial \bar{a}_{32}^k} & \frac{\partial^2 MSE}{\partial \bar{a}_{32}^k{}^2} \end{bmatrix},$$

evaluated at the values determined for \bar{a}_{12}^k , \bar{a}_{13}^k , \bar{a}_{21}^k , \bar{a}_{23}^k , \bar{a}_{31}^k , and \bar{a}_{32}^k . This reduces to:

$$\mathbf{H}_{MSE} = \begin{bmatrix} 2C_1 & C_3 & 0 & 0 & 0 & 0 \\ C_3 & 2C_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2C_6 & C_8 & 0 & 0 \\ 0 & 0 & C_8 & 2C_{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2C_{11} & C_{13} \\ 0 & 0 & 0 & 0 & C_{13} & 2C_{15} \end{bmatrix}.$$

The eigenvalues of \mathbf{H}_{MSE} are found to be positive for all real $\bar{\mathbf{x}}_{k-1|k-1}$:

$$\lambda_{H_{MSE}} = 2 \begin{bmatrix} 1 \\ 1 \\ 1 \\ \left(\frac{\|\bar{\mathbf{x}}_{k-1|k-1}\|_2}{\bar{x}_1^{k-1|k-1}} \right)^2 \\ \left(\frac{\|\bar{\mathbf{x}}_{k-1|k-1}\|_2}{\bar{x}_2^{k-1|k-1}} \right)^2 \\ \left(\frac{\|\bar{\mathbf{x}}_{k-1|k-1}\|_2}{\bar{x}_3^{k-1|k-1}} \right)^2 \end{bmatrix},$$

from which it is apparent that the Hessian matrix is positive definite and the values given for each of the elements of interest minimize the mean-square-error. The final three elements of $\bar{\mathbf{A}}_k$ are then found. From observation of the 3-D trends, the general N -D formula is then given as:

$$\bar{a}_{ij}^k = \frac{\left(\sum_{n=1, n \neq j}^N \bar{x}_n^{k-1|k-1^2}\right) \bar{a}_{ij}^{k-1} - \left(\sum_{n=1, n \neq j}^N \bar{x}_n^{k-1|k-1} \bar{x}_j^{k-1|k-1} \bar{a}_{in}^{k-1}\right) + \bar{x}_j^{k-1|k-1} \bar{q}_i^k}{\|\bar{x}_{k-1|k-1}\|_2^2},$$

and the matrix-form can be readily extracted as given in the text.

CHAPTER 5. Probabilistic Non-Gaussian Motion Model Correction

5.1 Introduction

5.1.1 Background

As technology has developed, the widening range of capabilities of autonomous systems has continued to impress and astonish. These developments are largely afforded by recent emphases on probabilistic estimation methods in engineering communities. Increasingly difficult problems facing the fields of robotics, automation, information fusion, and the like mandate accurate estimation in circumstances where sensor noise is complex and potentially non-Gaussian. Furthermore, because motion prediction is a valuable tool in many automation contexts, accurate probabilistic motion modeling has become progressively important for numerous estimation applications.

5.1.2 Related Work

Traditional recursive Bayesian estimation (RBE) consists of a prediction stage, an observation stage, and a correction (or updating) stage [1], [2]. Prediction requires an adequate motion model of some target which could be as stochastic as a weather system [3] or a human being [4], or as deterministic as an automobile [5]. Observations require a sensor or equivalent data-acquirer with an associated uncertainty characteristic [6]. Correction, then, is essentially the fusion of a prediction and an observation [7] which results in an improved estimate of the target.

Several different RBE methods are available, each of which makes certain assumptions in order to handle a particular kind of problem. The most famous family of estimators consists of what are known as Kalman filters (KF) which assume state belief is Gaussian throughout estimation. The original linear KF (LKF) introduced in the 1960s assumes linear motion and observation models [8]. Other common variants of the Kalman filter include the extended Kalman

filter (EKF) [9] and the unscented Kalman filter (UKF), [10] which both deal with nonlinear systems. Other versions exist, including the ensemble Kalman filter (EnKF) which is a Monte-Carlo method that can be used for non-Gaussian estimation [11], [12], the square-root cubature Kalman filter for highly nonlinear problems [13], the Schmidt-Kalman filter with polynomial chaos expansion for problems with increased uncertainty [14], and many more.

When the probability distribution functions (PDFs) describing state belief are non-Gaussian, Kalman filters are generally not sufficient. The sequential Monte-Carlo (or particle) filter relies on random generation of thousands or even millions of particles to adequately represent non-Gaussian belief [15], [16], while the grid-based filter requires that a belief space be regularly sampled into cells and RBE operations carried out for each cell [17]. Though these methods have worked well within their required contexts, they begin to fall short for fast estimation in many dimensions. This drawback is largely remedied by the Gaussian sum approach, in which non-Gaussian belief throughout RBE is represented by weighted Gaussian sums [18], [19]. It is this approach upon which the methods proposed here build.

When motion model parameters are not known to a sufficient degree of accuracy, the prediction stage of RBE will suffer, and so will estimation at large. A number of modern approaches have been put forward to address this issue. The adaptive Kalman filter (AKF) adjusts system and sensor noise covariance estimates to better reflect uncertainties in acquired state data [20]-[22]. Joint state-parameter estimation [23], [24] is a common solution for updating model parameters themselves; such methods have been shown to apply to even nonlinear and lightly non-Gaussian problems [19], but they generally do not extract model parameters implicitly from only a sequence of observations. Sensitivity methods have also been used to build more accurate system models throughout estimation, though such approaches are usually aimed towards finite-element

models [25]-[27]. Other advanced frameworks prove capable of controlling systems with uncertainty in the motion model [28], [29], but such work does not account for system or sensor noise and is therefore not estimation-oriented. Though established work is well-suited for specific kinds of advanced estimation problems, existing estimation techniques generally do not address parameter uncertainty and model correction when state belief is fully non-Gaussian.

5.1.3 Objectives and Outline

This paper describes a technique for probabilistic motion model correction using non-Gaussian belief fusion within the framework of simultaneous estimation and modeling (SEAM), which was previously proposed by the authors in [30]. In order to correct motion model parameters, the proposed method implements a gradient-descent approach, which solves the otherwise intractable minimization of a probabilistic error function. A means of subsequently estimating the uncertainty in these parameters is also presented for consistency within the SEAM framework.

Towards completing these objectives, this paper is organized as follows: first, a review of the relevant fundamental theory is given. Next, the original contributions of this work are addressed for both single and multiple state transition error minimization. In the fourth section, the results of these formulations as applied to several simulated scenarios are given and validative comparisons are made with conventional methods. Finally, a conclusions section summarizes this work's findings and briefly discusses future work in this area.

5.2 Belief, Estimation, and Model Correction

Before addressing the original contributions of this work, it is important to establish the foundation on which the following formulations are built. This includes a mathematical summary

of RBE, a description of Kalman estimation, the derivations of multi-Gaussian belief fusion, and the theory and formulations behind the SEAM framework.

5.2.1 Recursive Bayesian Estimation

In this paper, the term “belief” is used to characterize an agent's probabilistic estimate of a target or state. From this point forward, belief is defined mathematically by PDFs. Recursive Bayesian estimation propagates belief through three stages at every time step, making no assumptions about the specific forms of the PDFs at each stage. These stages are described below.

Prediction propagates belief from step $k - 1$ to step k by making use of a motion model of the system at hand. Let $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ be the PDF describing belief about a system's state $\mathbf{x} \in \mathcal{X}$ at step $k - 1$ given observations of that state $\mathbf{z} \in \mathcal{Z}$ from time step 1 to $k - 1$. Furthermore, let $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ be a *transition* PDF describing belief about the state at k given its previous state at $k - 1$. The continuous Chapman-Kolmogorov equation then gives the *predicted* PDF $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ as follows:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int_{\mathcal{X}} p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}_{k-1})d\mathbf{x}_{k-1}. \quad (1)$$

While the observation stage has no prescribed probabilistic formulations, its objective is to obtain an *observation* PDF $p(\mathbf{x}_k|\mathbf{z}_k)$ coming from a measurement of the state at a new time step k . Such a measurement, or observation, often comes from a sensor which itself may have a deterministic model. Obtaining $p(\mathbf{x}_k|\mathbf{z}_k)$ is therefore context-specific.

State correction is accomplished by Bayesian inference. The *predicted* PDF and the *observation* PDF are fused to generate the *corrected* PDF $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ given by:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_k|\mathbf{z}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int_{\mathcal{X}} p(\mathbf{x}_k|\mathbf{z}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k}. \quad (2)$$

To satisfy the recursive nature of RBE, the Chapman-Kolmogorov equation is seeded with the corrected PDF by resetting k to $k - 1$.

5.2.2 Kalman-type Estimation

Kalman filters are recursive Bayesian estimators, consisting of prediction, observation, and correction stages, which rely on assumed Gaussian belief. A multivariate Gaussian PDF is given by the following formula:

$$\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \boldsymbol{\Sigma}_x) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_x|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \boldsymbol{\Sigma}_x^{-1}(\mathbf{x} - \bar{\mathbf{x}})\right). \quad (3)$$

As Eq. (3) shows, a multivariate Gaussian is fully defined by its mean vector and covariance matrix. As a result, Kalman filters are able to efficiently propagate belief by propagating only these parameters.

5.2.2.1 The Linear Kalman Filter

Given a linear system motion model $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$ and sensor model $\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$, the LKF propagates the mean and covariance of Gaussian state belief from step $k - 1$ to step k according to the following prediction formulas:

$$\bar{\mathbf{x}}_{k|k-1} = \mathbf{A}\bar{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_{k-1}, \quad (4a)$$

$$\boldsymbol{\Sigma}_{\mathbf{x}_{k|k-1}} = \mathbf{A}\boldsymbol{\Sigma}_{\mathbf{x}_{k-1|k-1}}\mathbf{A}^T + \boldsymbol{\Sigma}_w. \quad (4b)$$

where $\boldsymbol{\Sigma}_w$ is the covariance of assumed additive zero-mean Gaussian noise in the system. State correction, then, is given in terms of an observation \mathbf{z}_k by:

$$\bar{\mathbf{x}}_{k|k} = \bar{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{C}\bar{\mathbf{x}}_{k|k-1}), \quad (5a)$$

$$\boldsymbol{\Sigma}_{\mathbf{x}_{k|k}} = (\mathbf{I} - \mathbf{K}\mathbf{C})\boldsymbol{\Sigma}_{\mathbf{x}_{k|k-1}}, \quad (5b)$$

where

$$\mathbf{K} = \boldsymbol{\Sigma}_{\mathbf{x}_{k|k-1}}\mathbf{C}^T \left(\mathbf{C}\boldsymbol{\Sigma}_{\mathbf{x}_{k|k-1}}\mathbf{C}^T + \boldsymbol{\Sigma}_v \right)^{-1}, \quad (6)$$

and Σ_v is the covariance of additive zero-mean Gaussian sensor noise. In addition to the assumption of Gaussian belief, the LKF makes the following two assumptions: 1) uncertainty in prediction and observation is caused only by additive noise, and 2) the model parameters \mathbf{A} , \mathbf{B} , and \mathbf{C} are known with complete confidence.

5.2.2.2 The Adaptive Kalman Filter

The AKF seeks to improve traditional Kalman filtering, or estimation, by adapting Σ_w and Σ_v to better reflect uncertainty in the underlying models. Because no effort is made to augment the model equations themselves, the AKF approach can apply to either linear or nonlinear systems. However, being a Kalman-type filter, it also assumes Gaussian belief. One of the most popular AKF variants [31] provides the following formulas:

$$\Sigma_{v_k} = \mathbf{G}_v - \mathbf{C}\Sigma_{x_k|k-1}\mathbf{C}^T, \quad (7a)$$

$$\mathbf{G}_v = \frac{1}{M} \sum_{m=1}^M \|\bar{\mathbf{z}}_{k-m} - \mathbf{C}_{k-m}\bar{\mathbf{x}}_{k-m|k-m-1}\|^2, \quad (7b)$$

$$\Sigma_{w_k} = \sqrt{\alpha}\Sigma_{w_{k-1}}, \quad (7c)$$

$$\alpha = \frac{\text{tr}(\mathbf{G}_v - \Sigma_{v_{k-1}})}{\text{tr}(\mathbf{C}\Sigma_{x_k|k-1}\mathbf{C}^T)}. \quad (7d)$$

Linear prediction, observation, and correction under the AKF follow Eqs. (4a)-(6). Though the AKF generally improves on the LKF by addressing additional potential sources of uncertainty, it does so observationally rather than analytically. Furthermore, model parameters are still assumed to be known with complete confidence. Consequently, it is generally not possible for an AKF estimator to achieve zero-mean error when model parameters are not correctly known.

5.2.3 Multi-Gaussian Belief Fusion

Where Gaussian assumptions fall short, non-Gaussian methods are critical. A PDF describing non-Gaussian state belief can be represented as a weighted sum of Gaussians with unique means, covariances, and weighting coefficients [1]. Such a PDF is henceforth described as a “multi-Gaussian” distribution. Multi-Gaussian belief fusion (MBF) addresses the correction stage of RBE for such multi-Gaussian belief. Let $p(\mathbf{x}|\mathbf{z}_i)$ denote the PDF representing belief about a state given the i^{th} observation of that state. Following some development of Eq. (2), the probabilistic fusion of n observations gives the PDF $p(\mathbf{x}|\mathbf{z}_{1:n})$, which is defined by the following formula:

$$p(\mathbf{x}|\mathbf{z}_{1:n}) = \frac{\prod_{i=1}^n p(\mathbf{x}|\mathbf{z}_i)}{\int_{\mathcal{X}} \prod_{i=1}^n p(\mathbf{x}|\mathbf{z}_i) d\mathbf{x}} \quad (8)$$

Because this is the normalized product of the PDFs corresponding to each observation, MBF makes use of the linear superposition of Gaussians to efficiently carry out belief fusion [7]. Let the belief corresponding to two observations, $p(\mathbf{x}|\mathbf{z}_1)$ and $p(\mathbf{x}|\mathbf{z}_2)$, be given by multi-Gaussian distributions containing I and J components, respectively, as follows:

$$p(\mathbf{x}|\mathbf{z}_1) = \sum_{i=1}^I c_i^{(1)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_i^{(1)}, \boldsymbol{\Sigma}_{x_i}^{(1)}), \quad (9a)$$

$$p(\mathbf{x}|\mathbf{z}_2) = \sum_{j=1}^J c_j^{(2)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_j^{(2)}, \boldsymbol{\Sigma}_{x_j}^{(2)}), \quad (9b)$$

Then the PDF representing fused belief, $p(\mathbf{x}|\mathbf{z}_{1:2})$, is given by the normalized product of the two constituents:

$$p(\mathbf{x}|\mathbf{z}_{1:2}) = \sum_{i=1}^I \sum_{j=1}^J c_i^{(1)} c_j^{(2)} c_{ij}^{(1:2)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_{ij}^{(1:2)}, \boldsymbol{\Sigma}_{x_{ij}}^{(1:2)}), \quad (10)$$

where the new weighting coefficient $c_{ij}^{(1:2)}$ accounts for the normalization in the denominator of Eq. (8). The new variables introduced are given in terms of the constituent parameters by the following definitions:

$$\boldsymbol{\Sigma}_{x_{ij}}^{(1:2)} = \left[\left(\boldsymbol{\Sigma}_{x_i}^{(1)} \right)^{-1} + \left(\boldsymbol{\Sigma}_{x_j}^{(2)} \right)^{-1} \right]^{-1}, \quad (11a)$$

$$\bar{\mathbf{x}}_{ij}^{(1:2)} = \boldsymbol{\Sigma}_{x_{ij}}^{(1:2)} \left[\left(\boldsymbol{\Sigma}_{x_i}^{(1)} \right)^{-1} \bar{\mathbf{x}}_i^{(1)} + \left(\boldsymbol{\Sigma}_{x_j}^{(2)} \right)^{-1} \bar{\mathbf{x}}_j^{(2)} \right], \quad (11b)$$

$$c_{ij}^{(1:2)} = \mathcal{N} \left(\bar{\mathbf{x}}_i^{(1)}; \bar{\mathbf{x}}_j^{(2)}, \left(\boldsymbol{\Sigma}_{x_i}^{(1)} + \boldsymbol{\Sigma}_{x_j}^{(2)} \right) \right). \quad (11c)$$

In summary, the fusion of two multi-Gaussian PDFs yields a third multi-Gaussian PDF with a number of constituents equal to the product of the numbers of constituents in each original distribution. Belief fusion in this manner is associative, so that fusion of n observations can be done sequentially, two-at-a-time. The advantage of MBF is that non-Gaussian belief can be represented in high dimensions efficiently and with full accuracy, without need for discretely sampling the belief space or randomly generating particles. The output of MBF is three parallel channels: one for mean vectors, one for covariance matrices, and one for weighting coefficients.

5.2.4 Simultaneous Estimation and Modeling

The recently proposed SEAM framework [30] includes formulations which address uncertainty in linearized motion model parameters in the context of prediction within RBE. Figure 1 summarizes this framework.

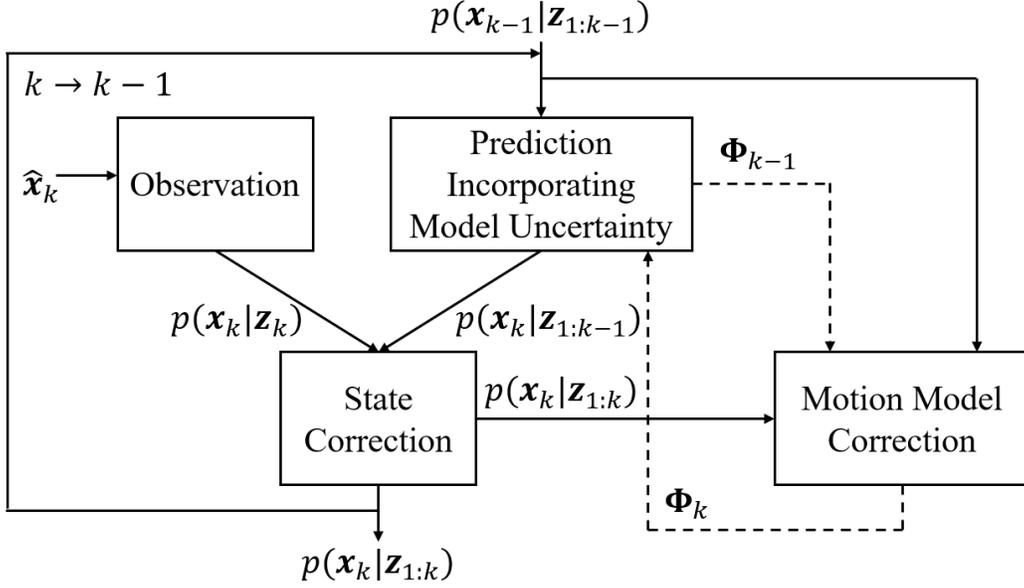


Figure 1. SEAM framework.

Let a system of interest have motion in the belief space which can be described by the following discrete-time state-space formula, with stationary, zero-mean, white process noise:

$$\mathbf{x}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{x}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad (12)$$

Here, \mathbf{w}_{k-1} is a zero-mean Gaussian random vector representing process noise, \mathbf{A}_{k-1} is the system state matrix, \mathbf{B}_{k-1} is the input matrix, and \mathbf{u}_{k-1} is an applied control effort. For Gaussian belief propagation, the mean at step k given the state at $k-1$ is determined by the means of each variable:

$$\bar{\mathbf{x}}_{k|k-1} = \bar{\mathbf{A}}_{k-1}\bar{\mathbf{x}}_{k-1|k-1} + \bar{\mathbf{B}}_{k-1}\bar{\mathbf{u}}_{k-1}, \quad (13)$$

By propagation of uncertainty, the covariance corresponding to belief at step k given step $k-1$ is estimated as

$$\begin{aligned} \Sigma_{\mathbf{x}_{k|k-1}} &= \Sigma_{\mathbf{w}_{k-1}} + \bar{\mathbf{A}}_{k-1}\Sigma_{\mathbf{x}_{k-1|k-1}}\bar{\mathbf{A}}_{k-1}^T + \bar{\mathbf{B}}_{k-1}\Sigma_{\mathbf{u}_{k-1}}\bar{\mathbf{B}}_{k-1}^T \\ &+ \Sigma_{\mathbf{A}_{k-1}}\mathcal{D}(\bar{\mathbf{x}}_{k-1|k-1}^{\circ 2})\Sigma_{\mathbf{A}_{k-1}}^T + \Sigma_{\mathbf{B}_{k-1}}\mathcal{D}(\bar{\mathbf{u}}_{k-1}^{\circ 2})\Sigma_{\mathbf{B}_{k-1}}^T. \end{aligned} \quad (14)$$

where the operation $\mathcal{D}(\bullet)$ diagonalizes its vector argument, $(\bullet)^{\circ 2}$ is the element-wise multiplication of a vector with itself, and both $\Sigma_{\mathbf{A}_{k-1}}$ and $\Sigma_{\mathbf{B}_{k-1}}$ are matrices whose elements are

the standard deviation of the corresponding elements in $\bar{\mathbf{A}}_{k-1}$ and $\bar{\mathbf{B}}_{k-1}$. Because these parameters are not random processes but rather fixed constants which are not perfectly known, these “standard deviations” act more as error bounds which quantify the uncertainty of belief in their corresponding parameters. Equation (14) simplifies to the result of the LKF's prediction step under the assumption that there is no uncertainty in the input vector or either of the state-space matrices.

A model-correction step is then implemented to increase the accuracy of the state matrix estimate for the next step $\bar{\mathbf{A}}_k$ and therefore decrease its estimated uncertainty Σ_{A_k} . This is done by minimizing the error between the maximum likelihood point (MLP), or mean, of the predicted PDF and the corrected PDF. The corrected $\bar{\mathbf{A}}_k$ matrix is given in terms of the corrected state mean $\bar{\mathbf{x}}_{k|k}$, the prior state mean $\bar{\mathbf{x}}_{k-1|k-1}$, the prior state and input matrices $\bar{\mathbf{A}}_{k-1}$ and $\bar{\mathbf{B}}_{k-1}$, and the prior input vector $\bar{\mathbf{u}}_{k-1}$ by

$$\bar{\mathbf{A}}_k = \frac{\bar{\mathbf{A}}_{k-1}(\bar{\mathbf{X}}_{k-1} - \bar{\mathbf{X}}_{k-1}(\mathbf{1} - \mathbf{I})\bar{\mathbf{X}}_{k-1}) + \bar{\mathbf{Q}}_k\bar{\mathbf{X}}_{k-1}}{\bar{\mathbf{x}}_{k-1|k-1}^T \bar{\mathbf{x}}_{k-1|k-1}}. \quad (15)$$

where $\bar{\mathbf{X}}_{k-1}$ is a diagonal matrix with $\bar{\mathbf{x}}_{k-1|k-1}^{jT} \bar{\mathbf{x}}_{k-1|k-1}^j$ at the $i = j$ positions and $\bar{\mathbf{x}}_{k-1|k-1}^j$ is defined as $\bar{\mathbf{x}}_{k-1|k-1}$ with the j^{th} element removed. Furthermore, $\bar{\mathbf{X}}_{k-1} = \mathcal{D}(\bar{\mathbf{x}}_{k-1|k-1})$ and $\bar{\mathbf{Q}}_k$ is a row matrix where each element is the column vector $\bar{\mathbf{q}}_k = \bar{\mathbf{x}}_{k|k} - \bar{\mathbf{B}}_{k-1}\bar{\mathbf{u}}_{k-1}$. The standard deviation of the corrected state matrix, Σ_{A_k} , is computed by error propagation on Eq. (15). For the sake of brevity, this formulation is not included here.

Because the above equations assume belief is Gaussian, only the means and covariances of the PDFs representing belief are used to correct \mathbf{A} . Applying the same simplistic error minimization principle to non-Gaussian belief will result in the loss of much valuable probabilistic information. Furthermore, these existing formulations do not address correction of the input matrix

\mathbf{B} , a sometimes important component of signal estimation. The original contributions introduced in the following section aim to remedy these shortcomings.

5.3 Multi-Gaussian Model Correction

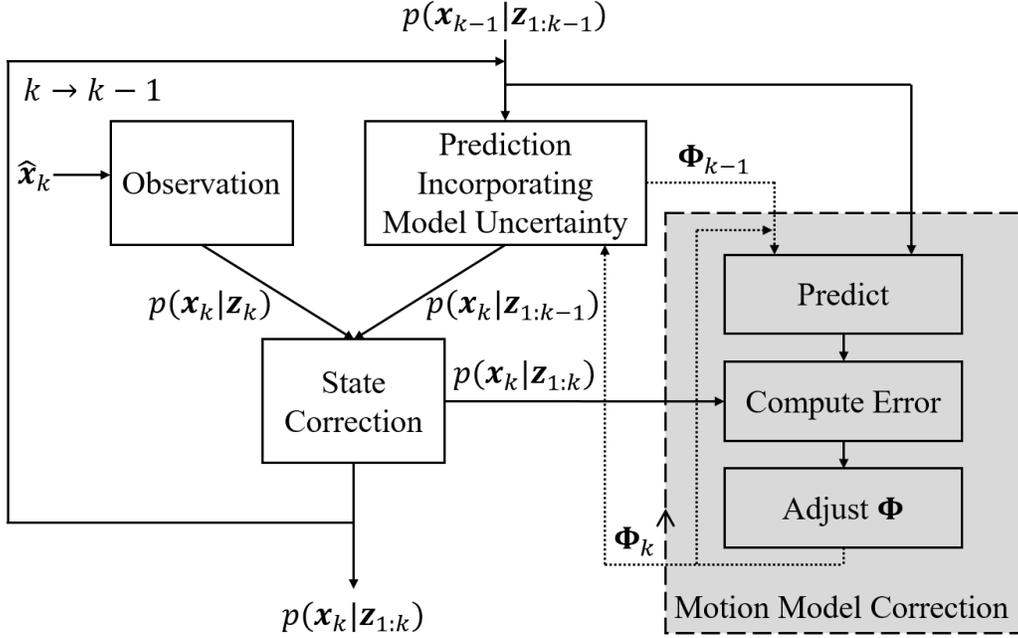


Figure 2. Proposed generalized SEAM with motion model correction employing non-Gaussian state belief.

Figure 2 shows the proposed probabilistic technique for generalizing SEAM to accommodate non-Gaussian motion model correction. The shaded block represents the proposed gradient-descent-based motion model correction technique, where belief is represented by multi-Gaussian PDFs such as those in Eqs. (9). Multi-Gaussian prediction is accomplished by applying Eqs. (13) and (14) to each Gaussian component to arrive at a new set of mean vectors and covariance matrices, with weighting coefficients remaining the same. State belief correction then takes place by fusing the predicted PDF and an observation PDF via MBF.

5.3.1 Single State Transition Error Minimization

The objective of the proposed approach to non-Gaussian motion model correction is to find the appropriate parameters in the state matrix $\bar{\mathbf{A}}_k$ and input matrix $\bar{\mathbf{B}}_k$ which yield a predicted PDF

$p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ that differs minimally from the corrected PDF $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. Stated mathematically, the desired model parameters are those which minimize an objective function. The objective function chosen here is termed the “integrated-squared-error (ISE)” and is defined as:

$$ISE = \int_{\mathbf{x}} (p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) - p(\mathbf{x}_k|\mathbf{z}_{1:k}))^2 d\mathbf{x}. \quad (16)$$

Another potential intuitive choice of objective function may be the Kullback-Leibler (K-L) Divergence of $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ from $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. However, the ISE is instead chosen because the only required PDF operations are addition, subtraction, multiplication, and integration. A K-L Divergence approach would also necessitate dividing PDFs and taking their logarithms, increasing the complexity of later steps. Furthermore, the ISE is a symmetrical measure of the similarity between the two PDFs, unlike the K-L Divergence.

Let $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ be written as multi-Gaussian PDFs, adopting a shorthand notation where the argument of the parenthetical superscript on each parameter corresponds to the time step of the most recent observation affecting that parameter. The ISE then can be written as:

$$ISE = \int_{\mathbf{x}} \left(\sum_{i=1}^I c_i^{(k-1)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_i^{(k-1)}, \boldsymbol{\Sigma}_{x_i}^{(k-1)}) - \sum_{j=1}^J c_j^{(k)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_j^{(k)}, \boldsymbol{\Sigma}_{x_j}^{(k)}) \right)^2 d\mathbf{x}. \quad (17)$$

Expanding the square gives:

$$ISE = \int_{\mathbf{x}} \left(\left(\sum_{i=1}^I c_i^{(k-1)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_i^{(k-1)}, \boldsymbol{\Sigma}_{x_i}^{(k-1)}) \right)^2 + \left(\sum_{j=1}^J c_j^{(k)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_j^{(k)}, \boldsymbol{\Sigma}_{x_j}^{(k)}) \right)^2 - 2 \sum_{i=1}^I \sum_{j=1}^J c_i^{(k-1)} c_j^{(k)} c_{ij}^{(k-1)(k)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_{ij}^{(k-1)(k)}, \boldsymbol{\Sigma}_{x_{ij}}^{(k-1)(k)}) \right) d\mathbf{x}. \quad (18)$$

Because $\int_{-\infty}^{\infty} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \boldsymbol{\Sigma}) d\mathbf{x} = 1$, this reduces to:

$$\begin{aligned}
ISE = & \sum_{i=1}^I \sum_{j=1}^I c_i^{(k-1)} c_j^{(k-1)} c_{ij}^{(k-1)(k-1)} + \sum_{i=1}^J \sum_{j=1}^J c_i^{(k)} c_j^{(k)} c_{ij}^{(k)(k)} \\
& - 2 \sum_{i=1}^I \sum_{j=1}^J c_i^{(k-1)} c_j^{(k)} c_{ij}^{(k-1)(k)} \tag{19}
\end{aligned}$$

Both $c_{ij}^{(k-1)(k-1)}$ and $c_{ij}^{(k-1)(k)}$ are implicit functions of the model parameters in $\bar{\mathbf{A}}_{k-1}$ and $\bar{\mathbf{B}}_{k-1}$.

Since this relationship is complex, solving for the optimal parameters in closed form is not viable.

The proposed technique solves this complex optimization problem using a gradient-descent method. The gradient vector is defined by assembling the derivatives of the ISE with respect to each model parameter as follows (with ‘‘parameters’’ referring to the elements in the \mathbf{A} and \mathbf{B} matrices):

$$\nabla_{\Phi}(ISE) = \left[\frac{\partial(ISE)}{\partial(a_{11})} \quad \dots \quad \frac{\partial(ISE)}{\partial(a_{pq})} \quad \dots \quad \frac{\partial(ISE)}{\partial(b_{11})} \quad \dots \quad \frac{\partial(ISE)}{\partial(b_{pq})} \quad \dots \right]^T \tag{20}$$

where Φ represents the vector of model parameters. Because the second term of Eq. (19) does not have direct functional dependence on $\bar{\mathbf{A}}_{k-1}$ or $\bar{\mathbf{B}}_{k-1}$, the gradient becomes:

$$\nabla_{\Phi}(ISE) = \sum_{i=1}^I \sum_{j=1}^I c_i^{(k-1)} c_j^{(k-1)} \nabla_{\Phi}(c_{ij}^{(k-1)(k-1)}) - 2 \sum_{i=1}^I \sum_{j=1}^J c_i^{(k-1)} c_j^{(k)} \nabla_{\Phi}(c_{ij}^{(k-1)(k)}), \tag{21}$$

where $c_{ij}^{(k-1)(k-1)}$ and $c_{ij}^{(k-1)(k)}$ are expressed as Gaussians according to Eq. (11c). Following the

chain rule on the derivative of a Gaussian, these gradients take the form:

$$\nabla_{\Phi}(c_{ij}^{(k-1)(k-1)}) = \exp(\beta_{ij}) \left(\nabla_{\Phi}(\alpha_{ij}) + \alpha_{ij} \nabla_{\Phi}(\beta_{ij}) \right), \tag{22a}$$

$$\nabla_{\Phi}(c_{ij}^{(k-1)(k)}) = \exp(\zeta_{ij}) \left(\nabla_{\Phi}(\epsilon_{ij}) + \epsilon_{ij} \nabla_{\Phi}(\zeta_{ij}) \right). \tag{22b}$$

The four placeholder variables are given by:

$$\alpha_{ij} = (|2\pi\boldsymbol{\gamma}_{ij}|)^{-\frac{1}{2}}, \quad \beta_{ij} = -\frac{1}{2} \boldsymbol{\delta}_{ij}^T \boldsymbol{\gamma}_{ij}^{-1} \boldsymbol{\delta}_{ij} \tag{23a}$$

$$\epsilon_{ij} = (|2\pi\boldsymbol{\eta}_{ij}|)^{-\frac{1}{2}}, \quad \zeta_{ij} = -\frac{1}{2} \boldsymbol{\theta}_{ij}^T \boldsymbol{\eta}_{ij}^{-1} \boldsymbol{\theta}_{ij} \quad (23b)$$

where $\boldsymbol{\gamma}_{ij}$, $\boldsymbol{\delta}_{ij}$, $\boldsymbol{\eta}_{ij}$, and $\boldsymbol{\theta}_{ij}$ follow from the formulas for mean and covariance propagation in prediction given by Eqs. (13) and (14)

$$\begin{aligned} \boldsymbol{\gamma}_{ij} = & \mathbf{A} \left(\boldsymbol{\Sigma}_{x_i}^{(k-1)} + \boldsymbol{\Sigma}_{x_j}^{(k-1)} \right) \mathbf{A}^T + \boldsymbol{\Sigma}_A \mathcal{D} \left(\bar{\mathbf{x}}_i^{(k-1)\circ 2} + \bar{\mathbf{x}}_j^{(k-1)\circ 2} \right) \boldsymbol{\Sigma}_A^T \\ & + 2 \left(\boldsymbol{\Sigma}_B \mathcal{D} \left(\bar{\mathbf{u}}^{(k-1)\circ 2} \right) \boldsymbol{\Sigma}_B^T + \mathbf{B} \left(\boldsymbol{\Sigma}_u^{(k-1)} \right) \mathbf{B}^T + \boldsymbol{\Sigma}_w \right), \end{aligned} \quad (24a)$$

$$\boldsymbol{\delta}_{ij} = \mathbf{A} \left(\bar{\mathbf{x}}_i^{(k-1)} - \bar{\mathbf{x}}_j^{(k-1)} \right), \quad (24b)$$

$$\boldsymbol{\eta}_{ij} = \mathbf{A} \boldsymbol{\Sigma}_{x_i}^{(k-1)} \mathbf{A}^T + \mathbf{B} \boldsymbol{\Sigma}_u \mathbf{B}^T + \boldsymbol{\Sigma}_A \mathcal{D} \left(\bar{\mathbf{x}}_i^{(k-1)\circ 2} \right) \boldsymbol{\Sigma}_A^T + \boldsymbol{\Sigma}_B \mathcal{D} \left(\bar{\mathbf{u}}^{(k-1)\circ 2} \right) \boldsymbol{\Sigma}_B^T + \boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_{x_j}^{(k)}, \quad (24c)$$

$$\boldsymbol{\theta}_{ij} = \mathbf{A} \bar{\mathbf{x}}_i^{(k-1)} + \mathbf{B} \bar{\mathbf{u}}^{(k-1)} - \bar{\mathbf{x}}_j^{(k)}. \quad (24d)$$

$\nabla_{\boldsymbol{\phi}}(\alpha_{ij})$, $\nabla_{\boldsymbol{\phi}}(\beta_{ij})$, $\nabla_{\boldsymbol{\phi}}(\epsilon_{ij})$, and $\nabla_{\boldsymbol{\phi}}(\zeta_{ij})$ are each assembled from the gradients with respect to the state and input matrix elements as in Eq. (20). The elements of $\nabla_{\boldsymbol{\phi}}(\alpha_{ij})$ and $\nabla_{\boldsymbol{\phi}}(\beta_{ij})$ are given as follows:

$$\frac{\partial \alpha_{ij}}{\partial a_{pq}} = -\frac{1}{2} (|2\pi\boldsymbol{\gamma}_{ij}|)^{-\frac{1}{2}} \text{tr} \left(\boldsymbol{\gamma}_{ij}^{-1} \frac{\partial \boldsymbol{\gamma}_{ij}}{\partial a_{pq}} \right), \quad (25a)$$

$$\frac{\partial \beta_{ij}}{\partial a_{pq}} = -\frac{1}{2} \left(\boldsymbol{\delta}_{ij}^T \boldsymbol{\gamma}_{ij}^{-1} \left(\frac{\partial \boldsymbol{\delta}_{ij}}{\partial a_{pq}} - \frac{\partial \boldsymbol{\gamma}_{ij}}{\partial a_{pq}} \boldsymbol{\gamma}_{ij}^{-1} \boldsymbol{\delta}_{ij} \right) + \left(\frac{\partial \boldsymbol{\delta}_{ij}}{\partial a_{pq}} \right)^T \boldsymbol{\gamma}_{ij}^{-1} \boldsymbol{\delta}_{ij} \right), \quad (25b)$$

where

$$\frac{\partial \boldsymbol{\gamma}_{ij}}{\partial a_{pq}} = \left(\mathbf{A} \left(\boldsymbol{\Sigma}_{x_i}^{(k-1)} + \boldsymbol{\Sigma}_{x_j}^{(k-1)} \right) \mathbf{O}_{pq}^T + \mathbf{O}_{pq} \left(\boldsymbol{\Sigma}_{x_i}^{(k-1)} + \boldsymbol{\Sigma}_{x_j}^{(k-1)} \right) \mathbf{A}^T \right), \quad (26)$$

and

$$\frac{\partial \boldsymbol{\delta}_{ij}}{\partial a_{pq}} = \mathbf{O}_{pq} \left(\bar{\mathbf{x}}_i^{(k-1)} - \bar{\mathbf{x}}_j^{(k-1)} \right). \quad (27)$$

Resulting from the chain and product rules, the derivatives of α_{ij} and β_{ij} with respect to the input matrix elements b_{pq} have the same form as given by Eqs. (25a) and (25b), but with each a_{pq} replaced by b_{pq} . For these,

$$\frac{\partial \gamma_{ij}}{\partial b_{pq}} = 2(\mathbf{B}\boldsymbol{\Sigma}_u \mathbf{O}_{pq}^T + \mathbf{O}_{pq} \boldsymbol{\Sigma}_u \mathbf{B}^T) \quad (28)$$

and

$$\frac{\partial \delta_{ij}}{\partial b_{pq}} = \mathbf{0}. \quad (29)$$

In the expressions above where the partial derivative with respect to a_{pq} is taken, \mathbf{O}_{pq} is a matrix the size of \mathbf{A} . For the derivatives with respect to b_{pq} , it is a matrix the size of \mathbf{B} . This matrix is defined as follows:

$$\mathbf{O}_{pq} = \begin{cases} 1 & \text{element } p, q \\ 0 & \text{elsewhere} \end{cases} \quad (30)$$

The elements of $\nabla_{\boldsymbol{\Phi}}(\epsilon_{ij})$, and $\nabla_{\boldsymbol{\Phi}}(\zeta_{ij})$ are given by the following partial derivatives:

$$\frac{\partial \epsilon_{ij}}{\partial a_{pq}} = -\frac{1}{2} (|2\pi\boldsymbol{\eta}_{ij}|)^{-\frac{1}{2}} \text{tr} \left(\boldsymbol{\eta}_{ij}^{-1} \frac{\partial \boldsymbol{\eta}_{ij}}{\partial a_{pq}} \right), \quad (31a)$$

$$\frac{\partial \zeta_{ij}}{\partial a_{pq}} = -\frac{1}{2} \left(\boldsymbol{\theta}_{ij}^T \boldsymbol{\eta}_{ij}^{-1} \left(\frac{\partial \boldsymbol{\theta}_{ij}}{\partial a_{pq}} - \frac{\partial \boldsymbol{\eta}_{ij}}{\partial a_{pq}} \boldsymbol{\eta}_{ij}^{-1} \boldsymbol{\theta}_{ij} \right) + \left(\frac{\partial \boldsymbol{\theta}_{ij}}{\partial a_{pq}} \right)^T \boldsymbol{\eta}_{ij}^{-1} \boldsymbol{\theta}_{ij} \right), \quad (31b)$$

where

$$\frac{\partial \boldsymbol{\eta}_{ij}}{\partial a_{pq}} = \mathbf{A} \boldsymbol{\Sigma}_{x_i}^{(k-1)} \mathbf{O}_{pq}^T + \mathbf{O}_{pq} \boldsymbol{\Sigma}_{x_i}^{(k-1)} \mathbf{A}^T, \quad (32)$$

and

$$\frac{\partial \boldsymbol{\theta}_{ij}}{\partial a_{pq}} = \mathbf{O}_{pq} \bar{\mathbf{x}}_i^{(k-1)}. \quad (33)$$

Again, the derivatives of ϵ_{ij} and ζ_{ij} with respect to the input matrix elements b_{pq} are given by Eqs. (31a) and (31b), but with each a_{pq} replaced by b_{pq} . For these,

$$\frac{\partial \boldsymbol{\eta}_{ij}}{\partial b_{pq}} = \mathbf{B} \boldsymbol{\Sigma}_u \mathbf{O}_{pq}^T + \mathbf{O}_{pq} \boldsymbol{\Sigma}_u \mathbf{B}^T \quad (34)$$

and

$$\frac{\partial \boldsymbol{\theta}_{ij}}{\partial b_{pq}} = \mathbf{O}_{pq} \bar{\mathbf{u}}^{(k-1)}. \quad (35)$$

The definition of \mathbf{O}_{pq} given by (30) and the accompanying description holds for Eqs. (32)-(35).

The above equations are used to inform a function which returns the gradient of the ISE evaluated for a particular set of model parameters, taking the following inputs: 1) the parameters defining the multi-Gaussian PDFs $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k | \mathbf{z}_{1:k})$, 2) the motion model parameters from step $k - 1$, 3) system inputs at step k , and 4) all parameter uncertainties. A gradient-descent algorithm is then implemented which locally minimizes the ISE by finding the optimal state and input matrix parameters assembled into the vector $\boldsymbol{\Phi}$. In this optimization process, the following expression is recursed over i until the change in $\bar{\boldsymbol{\Phi}}_k$ is sufficiently small:

$$(\bar{\boldsymbol{\Phi}}_k)_{i+1} = (\bar{\boldsymbol{\Phi}}_k)_i - d_i [\nabla_{\bar{\boldsymbol{\Phi}}}(\text{ISE})] |_{\boldsymbol{\Phi}=(\bar{\boldsymbol{\Phi}}_k)_i}. \quad (36)$$

Since this is generally a non-convex optimization problem, it is imperative that the local minimum found at step k is nearest to that at step $k - 1$ to achieve convergence of $\boldsymbol{\Phi}$ to a vector of constant values. An obvious seed for the gradient descent algorithm is $\bar{\boldsymbol{\Phi}}_{k-1}$.

5.3.2 Multiple State Transition Mean Error Minimization

There is a unique set of parameters $\boldsymbol{\Phi}$ which describe a single state transition in one dimension if the motion model is linear and subject to no noise. However, in higher dimensions, the motion model correction may require ISE minimization over multiple state transitions. In order to address such multi-dimensional problems, the ISE is augmented as follows:

$$ISE = \int_{\mathbf{x}} \left[\frac{1}{M+1} \sum_{i=k-M}^k (p(\mathbf{x}_i | \mathbf{z}_{1:i-1}) - p(\mathbf{x}_i | \mathbf{z}_{1:i}))^2 \right] d\mathbf{x}, \quad (37)$$

where $M \leq k$ is the number of recent observations used to update the parameter vector. Because of the linearity of integration, the gradient of the average ISE becomes the average of the individual gradients:

$$\nabla_{\Phi}(ISE) = \frac{1}{M+1} \sum_{i=k-M}^k \nabla_{\Phi}(ISE_i), \quad (38)$$

where $\nabla_{\Phi}(ISE_i)$ is the gradient of the i^{th} transition. The factor $\frac{1}{M+1}$ in Eq. (38) can be lumped with the step size d_i of Eq. (36), proving that the “direction” of the multi-transition ISE gradient comes from the summation of the ISE gradients of all transitions.

5.3.3 Uncertainty Propagation

In keeping with Eq. (14) of the SEAM framework, the uncertainty in the model parameter vector must be updated in addition to its mean. However, because the value of $\bar{\Phi}_k$ which minimizes the ISE cannot be found in closed form, Σ_{Φ_k} cannot be derived. Two approximation methods are given here: first, a more mathematically rigorous approach to estimating uncertainty through gradient descent propagation is presented, followed by a more implementation-oriented approach.

Consider a general error function $E = f(A)$ that is sought to be minimized by adjustment of the variable A . The gradient, in a 1-D sense, is given by $\frac{\partial E}{\partial A} \equiv g(A)$. Assume the implementation of some “smart” gradient descent algorithm where the step size d_i continually decreases as the local minimum is approached. Consider, then, the first three iterations:

$$A_0 = A_0, \quad (39a)$$

$$A_1 = A_0 - d_1 g(A_0), \quad (39b)$$

$$A_2 = A_1 - d_2 g(A_1) = (A_0 - d_1 g(A_0)) - d_2 g(A_0 - d_1 g(A_0)). \quad (39c)$$

By propagation of uncertainty, the variance at each step is given by:

$$\sigma_{A_0}^2 = \sigma_{A_0}^2, \quad (40a)$$

$$\sigma_{A_1}^2 = \sigma_{A_0}^2 + d_1^2 \sigma_{g(A_0)}^2, \quad (40b)$$

$$\sigma_{A_2}^2 = \sigma_{A_0}^2 + d_1^2 \sigma_{g(A_0)}^2 + d_1^2 \sigma_{g(A_0 - d_1 g(A_0))}^2. \quad (40c)$$

and a pattern emerges. For the “smart” algorithm assumed here, as i increases, d_i^2 decreases quickly. Furthermore, the gradient evaluated at each successive value of A constantly decreases, as does the uncertainty $\sigma_{g(A)}^2$. For these reasons, when convergence happens quickly (i.e. A_0 is near the value which locally minimizes E), additional terms contribute negligibly so that Eqs. (40a), (40b), and (40c) all provide reasonable approximations with increasing accuracy for the uncertainty in A . Therefore, with regards to the applied context, one option for estimating Σ_{Φ_k} is to ignore all terms but the first, so that:

$$\Sigma_{\Phi_k} \approx \Sigma_{\Phi_{k-1}}. \quad (41)$$

The more continuous the estimation process, the closer Σ_{Φ_k} will be to $\Sigma_{\Phi_{k-1}}$, and the better the above approximation holds.

A more heuristic and implementation-oriented approach can be considered. If the model parameters are improved with each iteration of estimation, their estimated uncertainty should continually decrease. Let a parameter λ be defined such that $0 < \lambda < 1$. The variance in Φ at step k can be adjusted according to $\Sigma_{\Phi_k} = \lambda \Sigma_{\Phi_{k-1}}$ in order to enforce that convergence of uncertainty follow a power law governed by λ^k . λ can be tuned to best reflect the increasing accuracy of model parameters over time.

5.4 Results

Several randomly generated simulations were carried out under varying conditions for the purpose of validating the proposed techniques. The presented framework was assessed by examining the convergence of model parameters to ground truth, in addition to comparing estimation performance with three other estimators. The LKF serves as a baseline technique representing the simplest of estimation methods, while the improved estimation of the AKF handles the more complex uncertainties addressed by the proposed framework. The AKF is the most advanced established estimator which can be directly compared to the proposed approach with regards to improving prediction when substantial model parameter uncertainties are present. Finally, a SEAM estimator with the recently proposed MLP-based model correction [30] is implemented for comparative purposes.

5.4.1 1-D Validation

While the formulations presented in section 5.3 apply to arbitrarily high-dimensional state vectors, this section and the next validate the proposed framework for only 1- and 2-D examples, for the sake of brevity.

5.4.1.1 Gaussian Belief

First, single-transition model correction was assessed for 5-second 1-D Gaussian simulations. Table 1 gives all constant, unitless parameter values used. These values were chosen arbitrarily, since the SEAM framework has no context-specific bias. Σ_{A_0} and Σ_{B_0} were chosen to be a percentage ε of A_{GT} and B_{GT} . To simulate an erroneous initial estimate of the motion model, A_0 and B_0 were randomly generated from normal distributions with means A_{GT} and B_{GT} , and standard deviations Σ_{A_0} and Σ_{B_0} . The control effort u used was a unit step function. The 1-D state-

space dynamics under these conditions simulate the step response of a system governed by a first-order differential equation.

Table 1. Constant parameters

A_{GT}	Σ_{A_0}	B_{GT}	Σ_{B_0}	Σ_w	x_0	Σ_{x_0}	Δt	Σ_u
0.98	εA_{GT}	1.3	εA_{GT}	0.01	2	0.09	0.025	0.01

Ground truth was simulated using A_{GT} and B_{GT} , with zero-mean Gaussian process noise w_k . Sensor noise was simulated as zero-mean additive Gaussian noise v_k on the ground-truth signal. A linear Kalman filter and more advanced adaptive Kalman filter were deployed based on the formulations given in section 5.2.2. Additionally, a SEAM estimator was implemented with MLP-based model correction according to Eq. (15). SEAM without model correction was also evaluated in order to isolate the effect of the formulations given here. Gradient descent was carried out according to Eq. (36) by using a constant step size of $d_i = 0.005$ and threshold $T = 0.00001$. Small values were chosen since the objective of these tests was proof-of-concept rather than fine-tuned efficiency. Σ_A and Σ_B were updated according to the heuristic approach described in section 5.3.3 with $\lambda = 0.99$.

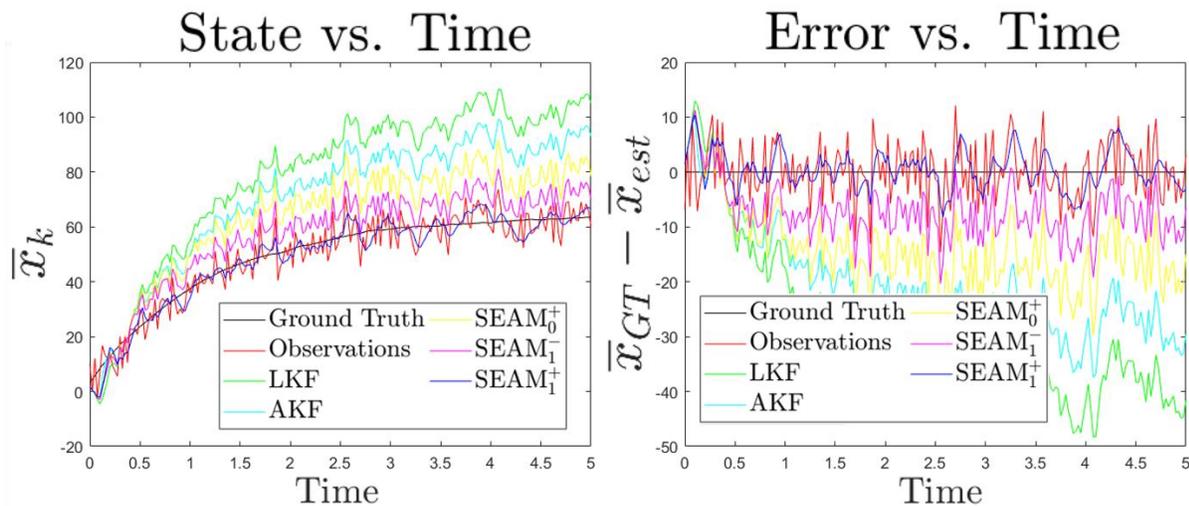


Figure 3. Example 1-D simulation with single-transition updating. In this example, $A_0 = 1.47$ and $B_0 = -1.95$.

Figure 3 shows an example trial under these simulation conditions, with $\varepsilon = 0.1$ and $\Sigma_v = 25$. Because the LKF errs toward overconfidence with respect to the motion model, even a small deviation from ground truth tends to have a dramatic impact as time progresses. The AKF improves on this by incorporating additional uncertainty, effectively relying more heavily on observations to improve estimation. Basic MLP model-correcting SEAM (labeled SEAM_0^+) further improves estimation by reducing the certainty of predictions even more, though SEAM without model correction (labeled SEAM_1^-) performs even better due to poor convergence of A by the MLP technique. SEAM with model correction according to ISE minimization (labeled SEAM_1^+) outperforms all other methods because it reliably adjusts model parameters towards ground-truth. Figure 4 plots the state and input matrices (scalars, in this case) over time.

In Fig. 4, the dashed lines represent the margin around ground-truth that represents 95% improvement from the initial parameter estimate. As the figure shows, A approaches ground-truth much more readily than B under ISE minimizing model correction. This is likely due to a number of factors, the chief of which is that the magnitude of the gradient $\nabla_{\Phi}(\text{ISE})$ is larger with respect to A than to B (i.e. the system is more sensitive to change in the state matrix than the input matrix).

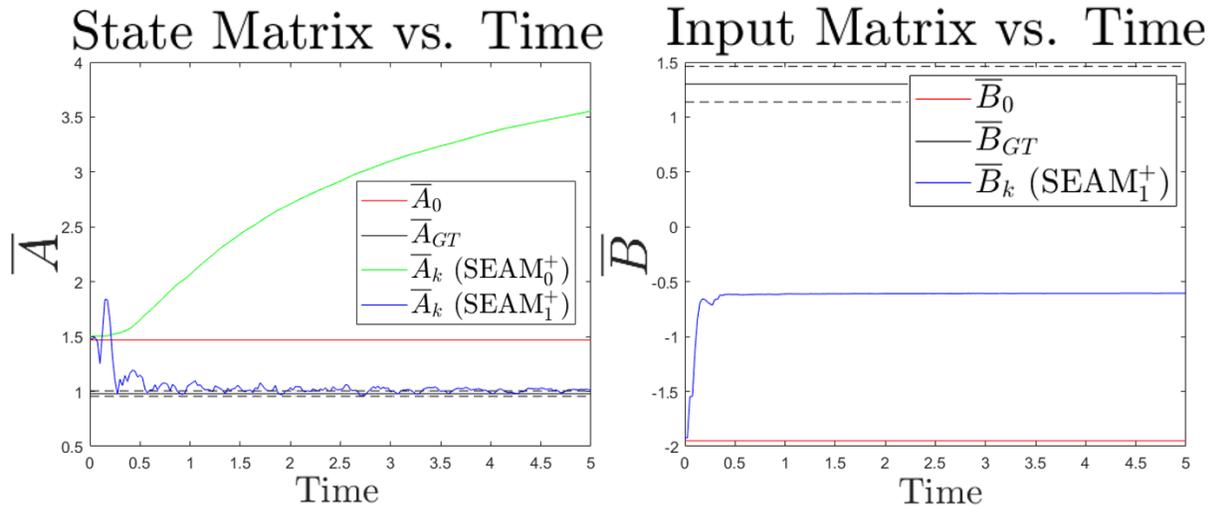


Figure 4. State and input matrices over time, for experiment corresponding to previous figure.

MLP-based model correction yields erroneous state matrix convergence, likely due to the forsaking of richer probabilistic information consistent with this approach.

The raw observation signal (O), LKF estimator (L), AKF estimator (A), MLP SEAM estimator (S_0^+), and ISE SEAM estimator without and with model correction (S_1^- and S_1^+) were each evaluated by computing their root-mean-squared-error (RMSE) scores for each of 25 trials. This was repeated for different combinations of ε and Σ_v , and the percentages of trials for which each signal had the lowest RMSE are given in Table 2.

As the table shows, the SEAM estimator with model correction as presented in this paper outperforms all other estimators the majority of the time for all combinations of noise variance and model parameter uncertainty. The AKF generally performed second-best overall. This is likely because this estimator has a greater smoothing effect, reducing RMSE in the occasional instances where the motion model is well estimated to begin with. Examining the proposed approach (S_1^+), there is a slight trend of decreasing relative performance as noise variance increases and a slight trend of increasing relative performance with increasing model uncertainty. The former result can be explained by the very poor signal-to-noise ratio in observations which negatively affects model correction. The latter result validates the hypothesized effectiveness of model correction in estimation to handle motion model mismatch.

Table 2. Sensitivity study in uncertainty parameters for 1-D Gaussian estimation

		ε							
		0.1		0.15		0.2		0.25	
Σ_v	1^2	O	0%	O	0%	O	0%	O	0%
		L	0%	L	0%	L	0%	L	0%
		A	0%	A	0%	A	4%	A	8%
		S0+	0%	S0+	0%	S0+	0%	S0+	0%
		S1-	4%	S1-	0%	S1-	0%	S1-	0%
		S1+	96%	S1+	100%	S1+	96%	S1+	92%
	5^2	O	0%	O	0%	O	0%	O	4%
		L	0%	L	0%	L	0%	L	0%
		A	12%	A	8%	A	8%	A	8%
		S0+	0%	S0+	0%	S0+	0%	S0+	0%
		S1-	0%	S1-	0%	S1-	0%	S1-	0%
		S1+	88%	S1+	92%	S1+	92%	S1+	88%
	10^2	O	0%	O	0%	O	0%	O	0%
		L	0%	L	0%	L	0%	L	0%
		A	28%	A	8%	A	8%	A	0%
		S0+	0%	S0+	0%	S0+	0%	S0+	0%
		S1-	8%	S1-	0%	S1-	0%	S1-	0%
		S1+	64%	S1+	92%	S1+	92%	S1+	100%
	15^2	O	0%	O	0%	O	0%	O	0%
		L	0%	L	0%	L	0%	L	0%
A		16%	A	28%	A	20%	A	20%	
S0+		0%	S0+	0%	S0+	0%	S0+	0%	
S1-		8%	S1-	4%	S1-	4%	S1-	4%	
S1+		76%	S1+	68%	S1+	76%	S1+	76%	
20^2	O	8%	O	4%	O	0%	O	4%	
	L	0%	L	4%	L	4%	L	0%	
	A	4%	A	12%	A	28%	A	12%	
	S0+	0%	S0+	0%	S0+	0%	S0+	0%	
	S1-	4%	S1-	8%	S1-	4%	S1-	8%	
	S1+	84%	S1+	72%	S1+	64%	S1+	76%	

In order to gain a better understanding of the convergence of model parameters given an inadequate initial estimate, model parameters and estimation error were plotted for a series of simulations in which the error in A_0 varied. Σ_v was held constant at 25, and all other parameters

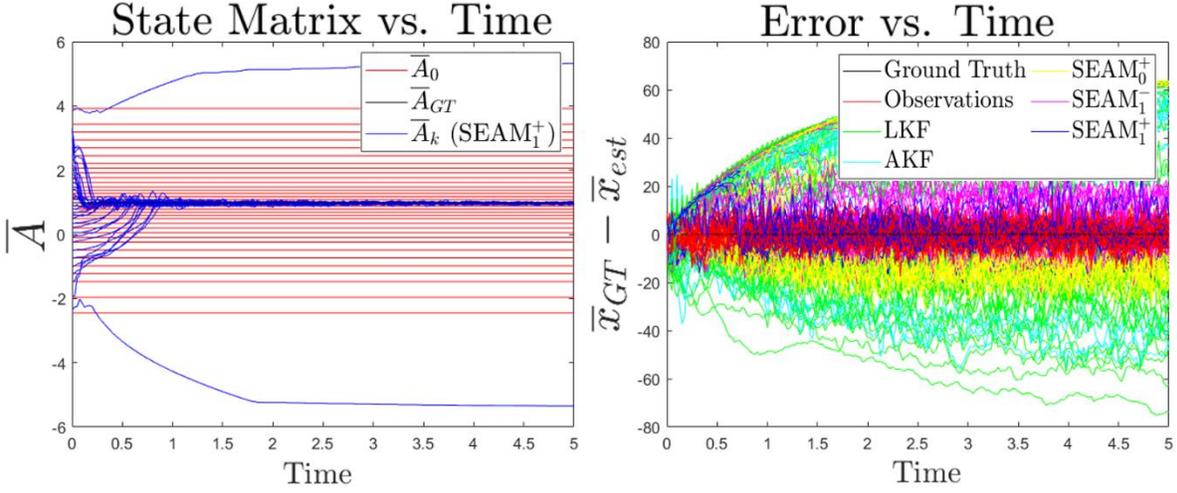


Figure 5. Convergence of state matrix estimate for varying initial guesses; corresponding error curves for each estimator.

remained unchanged. The starting values A_0 were plotted along with the updating values A_k , as shown in Fig. 5. Also included is the family of error curves coming from these experiments.

As the figure shows, model parameter convergence breaks down due to the non-convexity of the ISE with respect to A when the initial model parameter estimate is around $+4.5$ to -2.5 times the ground-truth value. It takes more time for the model parameter estimate to settle around \bar{A}_{GT} when A is significantly underestimated than when it is overestimated. The error accumulated early in simulation is another cause of high RMSE values which leads to occasional under-performance of SEAM as shown in Table 2.

A second uni-Gaussian 1-D validation was done to assess the performance of model-updating using multiple state transitions according to section 5.3.2. A sample simulation is plotted in Fig. 6 for $\varepsilon = 0.1$ and $\Sigma_v = 25$. The model parameters are plotted over time in Fig. 7. For this example, the 25 most recent state transitions were retained for adjusting the state matrix at each step. As the figure shows, there is less oscillation in the SEAM signal due to the reinforcement made possible by additional information from past observations. Again, while the input matrix

estimate does not converge to ground truth, it is nonetheless improved by the model correction step.

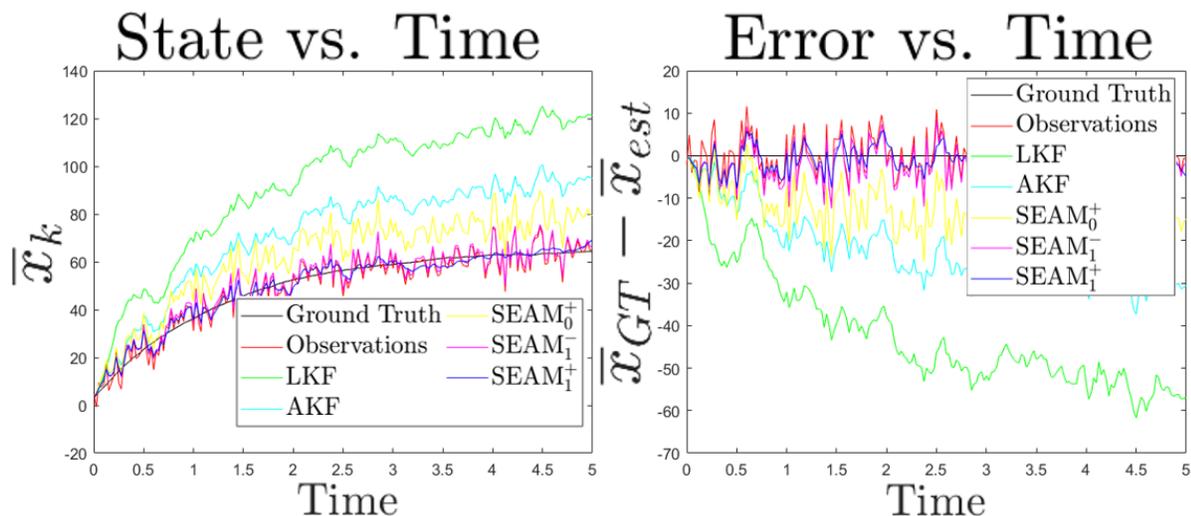


Figure 6. Example 1-D simulation with model-updating using the 25 most recent observed state transitions.

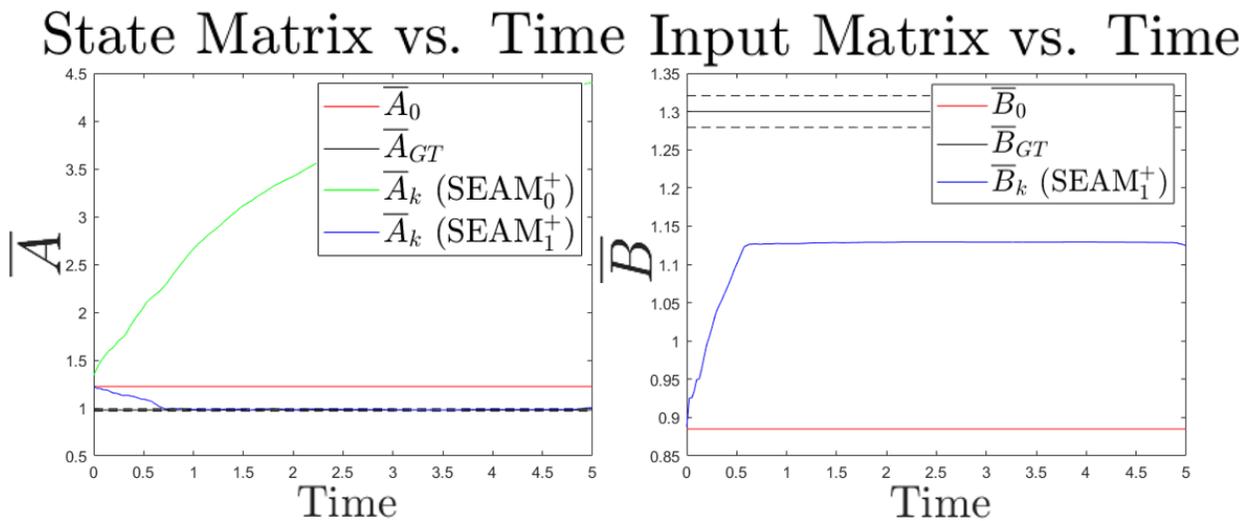


Figure 7. State and input parameters plotted over time.

5.4.1.2 Multi-Gaussian Belief

To validate model correction for multi-Gaussian belief, the above multi-transition model updating experiments were repeated where observations at each step were represented by the superposition of two Gaussians randomly placed in the vicinity of ground-truth at that step. Because no existing methods incorporate non-Gaussian model updating and the RMSE score

cannot be readily obtained since a single Gaussian mean is not available for comparison to ground truth, performance is quantified by the quality of motion model convergence.

Twenty-five experiments were conducted under the same conditions as in the previous section. The input matrix parameter B was held constant at ground-truth and not addressed because of its relatively low impact on estimation for these experiments. Furthermore, because MBF at the correction stage continually increases the number of Gaussian components, belief at each stage was pruned so that only the four strongest-contributing components were retained. Figure 8 plots \bar{A}_k for each of these experiments over time with the corresponding seed values \bar{A}_0 and the ground-truth value \bar{A}_{GT} . Also included in the figure is a visual demonstration of a single iteration of estimation using multi-Gaussian belief.

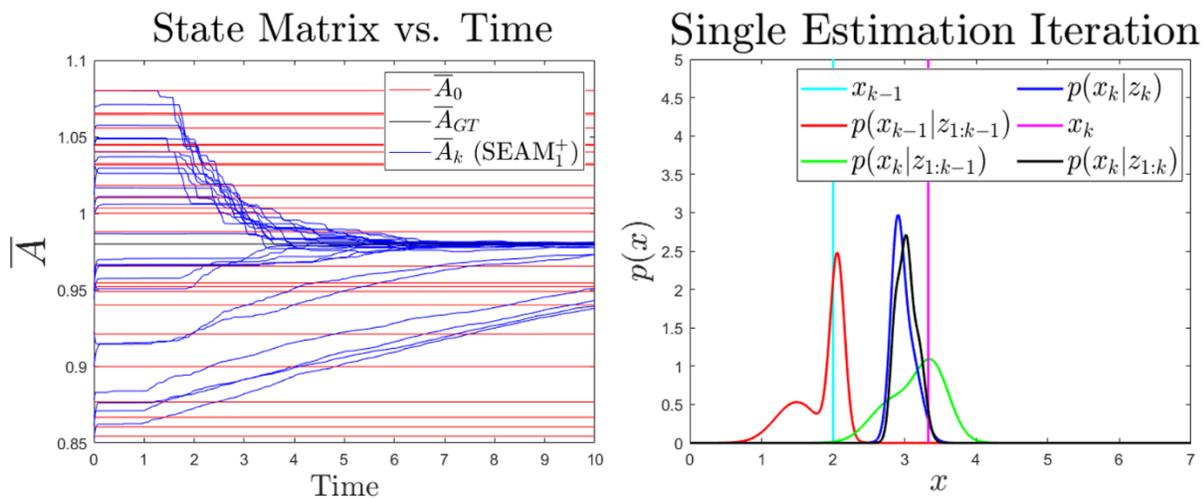


Figure 8. 1-D non-Gaussian state matrix updating for normally distributed random initial estimates; a single iteration under the estimation framework.

As the figure shows, the range of initial values A_0 that converge to ground truth over a given observation period is smaller than for the uni-Gaussian experiments presented in the previous section. It is also important to note that the total observation time was doubled, showing that under these conditions, convergence occurs more slowly. It is also clear from the plots that the true parameter A_{GT} is approached faster from a higher initial estimate than a lower one. Regarding the second part of the figure, ISE minimization as defined in Eq. (16) seeks to find

model parameters Φ which minimize the squared difference between the green and black curves (corresponding to the predicted and corrected PDFs, respectively).

5.4.2 2-D Validation

Next, a series of 2-D simulated experiments were carried out which address estimation of a second-order linear homogeneous ordinary differential equation (ODE). Many physical systems are described by decaying oscillatory motion, so simulation of such a scenario has real-world implications. N^{th} -order ODEs can be modeled by first-order linear state-space equations in N -D whose time-domain solutions also contain periodicity and decay.

In order to generate a discrete-time state-space equation describing a decaying sinusoid, three physical parameters of a mass-spring-damper (MSD) system were defined in both mean and standard deviation: mass (m), spring constant (k), and damping coefficient (b). For a displacement input $u(t)$ to the MSD system, the continuous-time state-space matrices are given by:

$$\bar{\mathbf{A}}_C = \begin{bmatrix} 0 & 1 \\ -\frac{\bar{k}}{\bar{m}} & -\frac{\bar{b}}{\bar{m}} \end{bmatrix}, \quad (42a)$$

$$\bar{\mathbf{B}}_C = \begin{bmatrix} 0 \\ \frac{\bar{k}}{\bar{m}} \\ \frac{\bar{b}}{\bar{m}} \end{bmatrix}, \quad (42b)$$

where the state and input vectors are defined as:

$$\bar{\mathbf{x}} = [\bar{x} \quad \dot{\bar{x}}]^T, \quad (43a)$$

$$\bar{\mathbf{u}} = [\bar{u} \quad \dot{\bar{u}}]^T, \quad (43b)$$

and x and u are the positions of the output and input in meters, respectively. The standard deviation matrices Σ_{A_C} and Σ_{B_C} can be determined in terms of their constituent parameters by appropriate propagation of uncertainty. The discrete-time matrices $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ can then be computed for a given

sampling time Δt by standard discretization formulas. The corresponding discrete-time standard deviation matrices can also then be determined.

Ground-truth values $\bar{\mathbf{A}}_{GT}$ and $\bar{\mathbf{B}}_{GT}$ were randomly generated from normal distributions with means and standard deviations coming from the above steps. The initial value of the state matrix was chosen to be $\bar{\mathbf{A}}_0 = \bar{\mathbf{A}} + 0.15\xi$, where ξ is a matrix of random variables uniformly distributed between 0 and 1. Ground-truth was simulated with zero-mean Gaussian process noise having covariance Σ_w , and Gaussian observation noise was simulated with covariance Σ_v on the ground-truth signal. Parameter values are given in Table 3. Standard deviations in k , m , and b were 35% of their means.

Table 3. Constant parameters

Δt	\bar{k}	\bar{m}	\bar{b}	$\bar{\mathbf{x}}_0$	Σ_{x_0}
0.05s	10 N/m	2 kg	5 Ns/m	$\begin{bmatrix} 10 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1^2 & 0 \\ 0 & 0.3^2 \end{bmatrix}$
Σ_u		Σ_v		Σ_w	
$\begin{bmatrix} 0.05^2 & 0.007^2 \\ 0.007^2 & 0.003^2 \end{bmatrix}$		$\begin{bmatrix} 0.6^2 & 0.23^2 \\ 0.23^2 & 0.57^2 \end{bmatrix}$		$\begin{bmatrix} 0.001 & 0 \\ 0 & 0.002 \end{bmatrix}$	

A 50-second sample simulation for which the control effort was a unit-amplitude sinusoidal displacement with frequency 0.125 Hz and covariance Σ_u is shown in Fig. 9 below. The figure includes plots of the output position and velocity and the corresponding RMSE errors for the same estimators as were described in section 5.4. Figure 10 shows the four elements of the state and input matrices $\bar{\mathbf{A}}_k$ and $\bar{\mathbf{B}}_k$ over time. Because SEAM with MLP-based model correction offers updated state matrix values, these are also plotted in Fig. 10a.

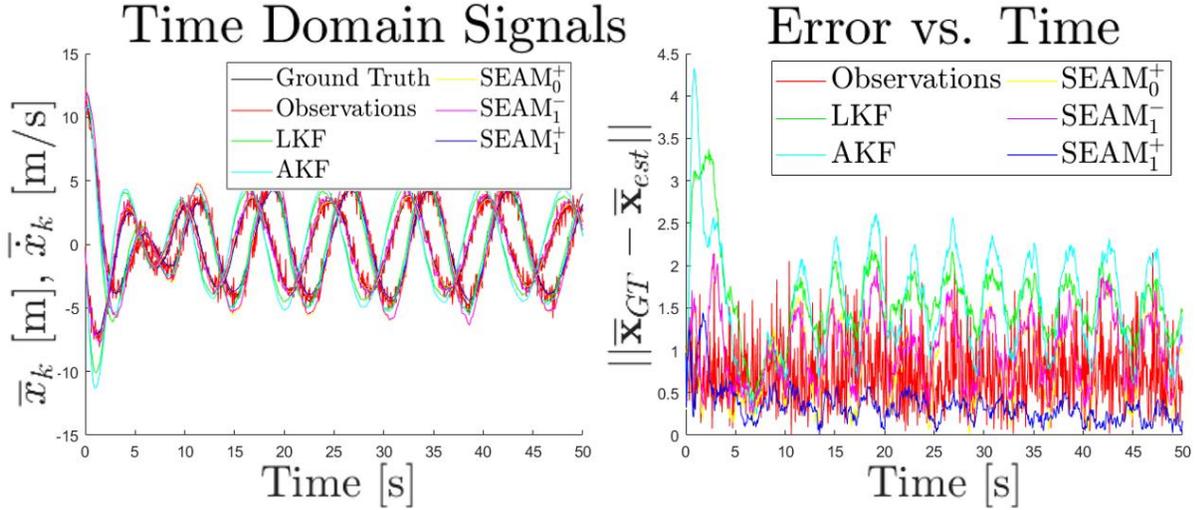


Figure 9. Sample simulation for 2-D uni-Gaussian validation.

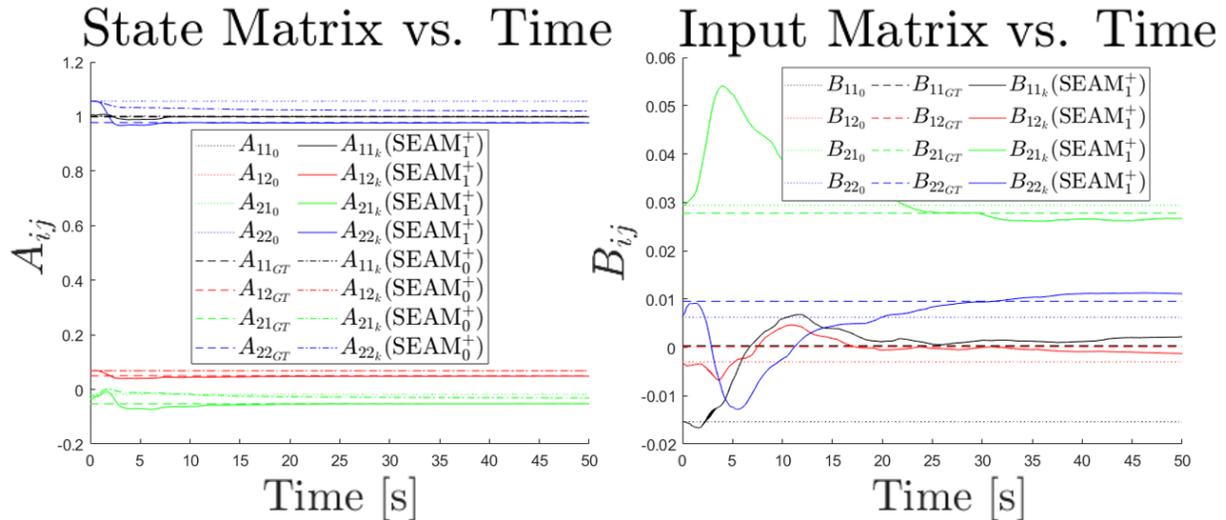


Figure 10. Convergence of model parameters for 2-D uni-Gaussian validation.

For this simulation, $M = k$. While the four elements of the state matrix converge early and accurately under the proposed ISE-based model correction approach, only A_{21} and A_{22} begin to approach ground-truth under MLP-based model correction. Even then, convergence does not occur in the allotted 50 seconds of simulation. As the figure also shows, input matrix parameters take longer to approach ground-truth under the proposed approach. Admittedly, elements of \mathbf{B} sometimes take even longer to converge or do not converge to ground-truth at all; this is again

likely due to the fact that the system dynamics are largely governed by the state matrix and the ISE is relatively insensitive to change in \mathbf{B} .

For a more quantitative assessment, simulation under the previously specified conditions was carried out 100 times. Each simulation spanned ten seconds, and the unforced response was observed. In order to assess the efficacy of the proposed ISE-based model correction, the means of the percent errors in state matrix elements were recorded both at the first time step ($k = 0$) and the last time step ($k = 200$), and the percent of the trials for which $A_{ij_{200}}$ gave lower percent error than A_{ij_0} was recorded. To assess the quality of estimation, the median RMSEs of the various estimated signals were computed, in addition to the percent of trials for which each method outperformed the others. The results are given in Tables 4 and 5 below:

Table 4. Percent errors in state matrix elements for 100 trials

	% A_{11}	% A_{12}	% A_{21}	% A_{22}
$k = 0$	7.80	170.92	95.08	7.97
$k = 200$	2.83	19.34	15.75	2.51
% Improvement	64	89	83	69

Table 5. Median root-mean-squared-errors and percent wins over 100 simulations for all estimators

	LKF	AKF	SEAM $_0^+$	SEAM $_1^-$	SEAM $_1^+$
RMSE	2.899	3.596	2.065	1.683	1.202
% Wins	0%	1%	2%	14%	83%

As Table 4 shows, under the proposed approach, state matrix parameters are improved 60%-90% of the time, resulting in a clear overall advantage with regards to estimation. The proposed framework results in an average of over 500% improvement in estimated state matrix

element error. According to Table 5, each estimator successively improves upon the last, with the unexpected exception of the AKF. Upon closer examination, the likely cause for occasional higher estimation error coming from the AKF was poor conditioning of the covariance matrices coming from the randomness in parameter setting required for this Monte Carlo type validation study. In specific simulations where this was not an issue, the AKF generally outperformed the LKF as expected.

5.5 Conclusions, Applications, and Future Work

The probabilistic model correction approach formulated here has proven capable of handling uncertainty in state-space motion model parameters by continuously updating these parameters and their uncertainties. The fundamental formulations allow for arbitrary non-Gaussian representation of belief in N -D with the only assumption being that belief can be reasonably modeled by a sum of Gaussians. Even amidst appreciable sensor noise, the proposed approach builds an improved estimate of motion model parameters which tend to converge efficiently given sufficient time. This leads to superior state estimation in comparison with traditional methods which do not account for model uncertainty or correct model parameter estimates. The proposed technique accomplishes its intended purpose in multiple dimensions, for both uni- and multi-Gaussian belief.

There is a wide range of future work that could stem from the research presented here. A study on the effect of active control on system identification via model correction would lend insight into how these techniques could be used in various control contexts. Furthermore, the application of these formulations to heavily nonlinear systems would greatly enhance their ubiquity. A more comprehensive framework would address observation model correction in addition to motion model correction, since the two play dual roles in estimation. The parameter

space for fully investigating the capabilities of probabilistic model correction is unwieldy, so only a limited study of the effect of certain parameters was undertaken here. It would be informative to further study the effect of dimensionality on the required number of state transitions in error minimization to robustly update the motion model. Finally, an investigation into efficient gradient-descent algorithms would further improve the real-time capabilities of these methods. SDG

5.6 References

- [1] H. W. Sorenson, D. L. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica*, vol. 7, pp. 465-479, 1971.
- [2] M. Kumon, D. Kimoto, K. Takami, T. Furukawa, "Bayesian Non-Field-of-View Target Estimation Incorporating An Acoustic Sensor," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3425-3432, 2013.
- [3] J. Zhang, C. Draxl, T. Hopson, L. D. Monache, E. Vanvyve, B. Hodge, "Comparison of numerical weather prediction based deterministic and probabilistic wind resource assessment methods," *Applied Energy*, pp. 528-541, 2015.
- [4] T. M. T. Do, O. Dousse, M. Miettinen, D. Gatica-Perez, "A probabilistic kernel method for human mobility prediction with smartphones," *Pervasive and Mobile Computing*, vol. 20, pp. 13-28, 2015.
- [5] N. Wan, C. Zhang, A. Vahidi, "Probabilistic Anticipation and Control in Autonomous Car Following," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 30-38, 2019.
- [6] M. Liggins II, D. Hall, J. Llinas, *Handbook of Multisensor Data Fusion: Theory and Practice*, Second Edition, Boca Raton, FL, USA: CRC Press, 2017.
- [7] J. J. Steckenrider, T. Furukawa, "Multi-Dimensional Belief Fusion of Multi-Gaussian Structures," *Information Fusion*, vol. 57, pp. 71-88, 2020.

- [8] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. D, pp. 35-45, 1960.
- [9] G.A. Einicke, L.B. White, "Robust extended Kalman filtering," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2596-2599, 1999.
- [10] E. A. Wan, R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000*, 2000.
- [11] J. Mandel, "A Brief Tutorial on the Ensemble Kalman Filter," *arXiv*, pp. 1-7, 2009.
- [12] M. Katzfuss, J. R. Stroud, C. K. Wikle, "Understanding the Ensemble Kalman Filter," *The American Statistician*, vol. 70, no. 4, pp. 350-357, 2016.
- [13] D. Potnuru, K. P. B. Chandra, I. Arasaratnam, D. Gu, K. A. Mary, S. B. Ch, "Derivative-free square-root cubature Kalman filter for non-linear brushless DC motors," *IET Electric Power Applications*, vol. 10, no. 5, pp. 419-429, 2016.
- [14] Y. Yang, H. Cai, B. Gong, R. Norman, "Schmidt-Kalman Filter with Polynomial Chaos Expansion for State Estimation," *22nd International Conference on Information Fusion*, Ottawa, Ontario, CA, 2019.
- [15] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.
- [16] D. Raihan, S. Chakravorty, "Particle Gaussian Mixture Filters: Application and Performance Evaluation," *22nd International Conference on Information Fusion*, Ottawa, Ontario, CA, 2019.
- [17] C. Kalender, A. Schöttl, "Sparse Grid-Based Nonlinear Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2386-2396, 2013.

- [18] D. L. Alspach, H. W. Sorenson, "Nonlinear Bayesian Estimation Using Gaussian Sum Approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439-448, 1972.
- [19] V. Stojanovic, N. Nedic, "Robust Kalman filtering for nonlinear multivariable stochastic systems in the presence of non-Gaussian noise," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 3, pp. 445-460, 2016.
- [20] P. Yu, M. Wu, J. She, K. Liu, Y. Nakanishi, "Robust Tracking and Disturbance Rejection for Linear Uncertain System With Unknown State Delay and Disturbance," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1445-1455, 2018.
- [21] E. Zerdali, "Adaptive Extended Kalman Filter for Speed-Sensorless Control of Induction Motors", *IEEE Transactions on Energy Conversion*, vol. 34, no. 2, pp. 789-800, 2019.
- [22] A. Katriniok, D. Abel, "Adaptive EKF-Based Vehicle State Estimation With Online Assessment of Local Observability," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1368-1381, 2016.
- [23] V. Stojanovic, N. Nedic, "Joint state and parameter robust estimation of stochastic nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 14, pp. 3058-3074, 2016.
- [24] S. Ibrir, "Joint state and parameter estimation of non-linearly parameterized discrete-time nonlinear systems," *Automatica*, vol. 97, pp. 226-233, 2018.
- [25] J. E. Mottershead, M. Link, M. I. Friswell, "The sensitivity method in finite element model updating: A tutorial," *Mechanical Systems and Signal Processing*, vol. 25, no. 7, pp. 2275-2296, 2011.
- [26] A. Lund, S. J. Dyke, W. Song, I. Billionis, "Global sensitivity analysis for the design of nonlinear identification experiments," *Nonlinear Dynamics*, vol. 98, pp. 375-394, 2019.

- [27] E. Patelli, Y. Govers, M. Broggi, H. M. Gomes, M. Link, J. E. Mottershead, “Sensitivity or Bayesian model updating: a comparison of techniques using the DLR AIRMOD test data,” *Archive of Applied Mechanics*, vol. 87, no. 5, pp. 905-925, 2017.
- [28] H. Liu, Y. Pan, J. Cao, “Composite Learning Adaptive Dynamic Surface Control of Fractional-Order Nonlinear Systems,” *IEEE Transactions On Cybernetics*, Early access, 2019.
- [29] Y. Pan, T. Sun, H. Yu, “Composite adaptive dynamic surface control using online recorded data,” *International Journal of Robust and Nonlinear Control*, vol. 26, no. 18, pp. 3921-3936, 2016.
- [30] J. J. Steckenrider, T. Furukawa, “A Probabilistic Model-Adaptive Approach for Tracking of Motion with Heightened Uncertainty,” *International Journal of Control, Automation, and Systems*, *Pre-print*, 2020.
- [31] A. Almagbile, J. Wang, W. Ding, “Evaluating the Performances of Adaptive Kalman Filter Methods in GPS/INS Integration,” *Journal of Global Positioning Systems*, vol. 9, no. 1, pp. 33-40, 2010.

CHAPTER 6. Probabilistic Non-Gaussian Sensor Model Correction

6.1 Introduction

6.1.1 Background

Today's technologically advanced world is rich with sensors and sensing agents. The advanced engineering systems which provide the services, entertainment, and conveniences of modern-day life rely heavily on their ability to acquire observations about their surroundings or their own state [1]. Because sensor imperfections are inevitable, uncertainty is often introduced in the measurement of a target signal or state. This mandates the use of probabilistic methods to overcome noise and bias in more robustly estimating an underlying target [2], [3]. Several estimation frameworks have been put forward in recent years which account for observation model uncertainties and sensor noise, the most prominent of which will be summarized here.

6.1.2 Related Work

State estimation has most famously been achieved in recent decades by an ever-expanding family of Kalman filters (KFs). Some of the more common KFs are the linear Kalman filter (LKF) for linear problems [4], extended and unscented Kalman filters (EKFs and UKFs) for standard nonlinear problems [5], [6], the square-root cubature KF (SRCKF) for highly nonlinear problems [7], the ensemble KF (EnKF) for certain non-Gaussian problems [8], and the adaptive KF (AKF) for problems with ill-defined uncertainty [9]. Each of these estimators generally tracks the first and second moments of the Gaussian probability distribution functions (PDFs) which probabilistically represent the state of interest, accounting for additive noise in both motion models and observation (or sensor) models. Because of the ubiquity of these estimators, they can be used in a wide range of contexts, varying from robotics applications [5] to battery life estimation [7].

A more general approach to both nonlinear and non-Gaussian problems is achieved by recursive Bayesian estimation (RBE) [10], [11]. RBE makes no assumptions about the form of a PDF used to probabilistically represent a state being estimated. (For generality, the estimate of a state is referred to here as “state belief”.) The formulas of RBE can be implemented in a number of ways to accommodate various assumptions, computational demands, or accuracy requirements. While the basic KFs described above assume that the estimated state is Gaussian, the more difficult problem of non-Gaussian RBE is achieved in three primary ways.

The first approach to non-Gaussian RBE, known as the grid-based filter, regularly samples the state belief space with a resolution determined by an accuracy/efficiency trade-off and predicts and updates state belief by performing operations on each grid cell [12]. A similar but more well-known approach is known as the particle, or sequential Monte Carlo, filter. Under this framework, particles representing state belief are randomly generated, weighted, and propagated according to the formulas of RBE [13], [14]. The final RBE method reviewed here approximates non-Gaussian state belief as a weighted sum of Gaussian PDFs and leverages the properties of Gaussians to efficiently carry out RBE [15]. The Simultaneous Estimation and Modeling (SEAM) framework proposed by Steckenrider, et al. in [11] and further developed in [16] uses the weighted-Gaussian-sum approach and further introduces a motion model correction stage. However, this has not yet been accomplished for observation models.

In order to effectively capture and remove the biasing characteristics of a sensor in state observation, having an accurate mathematical sensor or observation model is critical. Prior characterization experiments or calibration steps are often needed to estimate various sensor-intrinsic parameters [17]. While all the aforementioned KF and RBE variants are effective estimators of systems subject to additive observational noise, none actually address uncertainty in

observation model parameters themselves. Work by Liu, et al. seeks to estimate sensor faults in estimation and control problems [18], [19], and similar work by Youssef, et al. addresses the simultaneous estimation of system states and sensor faults [20]. These faults take the form of additive time-varying signals in a linear observation model. Though poor estimation of the output matrix may be compensated for by this inclusion, uncertainty in the output matrix is not specifically addressed. Furthermore, states are not handled probabilistically under these frameworks. While output matrix uncertainty is addressed by Sani, et al. [21], the predictive advantage of the RBE framework is not leveraged. There is a shortage of techniques for addressing uncertainty in observation model parameters within generally non-Gaussian RBE, a topic which this paper addresses.

6.1.3 Objectives and Outline

This paper proposes a framework which 1) formulates observational uncertainty by incorporating both sensor noise and observation model¹ parameter uncertainty, and 2) corrects the estimates of these model parameters by using an enhanced state estimate coming from belief fusion. These original contributions are clarified by presenting novel probabilistic formulations for observations, observation inversions, and artificial re-observations. Non-Gaussian state belief is approximated by multi-Gaussian PDFs in order to preserve both accuracy and efficiency. Complex optimization of a multi-Gaussian objective function is handled by a gradient-descent approach to observation model correction. Furthermore, in keeping with the presented observation formulations, an approach to updating the variance in observation model parameters is also given.

To clarify these objectives, the paper is organized as follows: first, section 2 establishes essential concepts and formulations related to the original contributions of this paper. Next, section

¹ Henceforth, the term “observation model” can be taken synonymously with the term “sensor model”.

3 details the proposed approach, including observation formulations, gradient-descent model correction, and uncertainty updating. Some results of the presented framework are given in section 4, followed by conclusions and future work in section 5.

6.2 Simultaneous Estimation and Modeling

6.2.1 Recursive Bayesian Estimation

Let \mathbf{x} represent a state being estimated. A PDF describing that state's belief is generally given the notation $p(\mathbf{x})$. RBE consists of three stages: prediction, observation, and correction. At each stage, state belief PDFs are propagated in such a way as to properly account for the state's probabilistic characteristics.

6.2.1.1 Prediction

Prediction is given by the continuous Chapman-Kolmogorov equation governing a Markovian process:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int_{\mathbf{x}} p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) p(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}. \quad (1)$$

The resulting *predicted* PDF describes belief about the state \mathbf{x} at step k resulting from: 1) *a priori* belief about the state at $k - 1$ (given by $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})$), and 2) a *transition* PDF $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ which can be obtained from a deterministic motion model describing the system. Such a motion model can be generally described as a function of the previous state, external inputs \mathbf{u} , certain intrinsic parameters (represented by a vector Φ), and process noise \mathbf{w} :

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \Phi) + \mathbf{w}_{k-1}. \quad (2)$$

Sometimes, these PDFs can be assumed to be Gaussian. A multivariate Gaussian PDF representing belief about \mathbf{x} has the following definition:

$$\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{\Sigma}_x) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}_x|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{\Sigma}_x^{-1}(\mathbf{x} - \bar{\mathbf{x}})\right), \quad (3)$$

where $\bar{\mathbf{x}}$ is the mean vector and $\mathbf{\Sigma}_x$ is the covariance matrix. When *a priori* state belief is generally non-Gaussian, it can often be reasonably approximated by a finite sum of weighted Gaussian distributions [10]:

$$p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) = \sum_{i=1}^I c_i^{(k-1|1:k-1)} \mathcal{N}\left(\mathbf{x}; \bar{\mathbf{x}}_i^{(k-1|1:k-1)}, \mathbf{\Sigma}_{x_i}^{(k-1|1:k-1)}\right), \quad (4)$$

where the superscripts in the means, covariances, and weighting coefficients signify the PDF to which they belong and the subscripts index the Gaussian components within the PDF. Having *a priori* state belief as a multi-Gaussian PDF, predicted belief is also given by a multi-Gaussian distribution:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \sum_{i=1}^I c_i^{(k|1:k-1)} \mathcal{N}\left(\mathbf{x}; \bar{\mathbf{x}}_i^{(k|1:k-1)}, \mathbf{\Sigma}_{x_i}^{(k|1:k-1)}\right), \quad (5)$$

where the parameters describing this PDF are given in terms of the parameters describing the *a priori* PDF and the motion model by

$$c_i^{(k|1:k-1)} = c_i^{(k-1|1:k-1)}, \quad (6a)$$

$$\bar{\mathbf{x}}_i^{(k|1:k-1)} = f\left(\bar{\mathbf{x}}_i^{(k-1|1:k-1)}, \mathbf{u}_{k-1}, \mathbf{\Phi}\right), \quad (6b)$$

$$\mathbf{\Sigma}_{x_i}^{(k|1:k-1)} \approx \left(\frac{\partial f(\bullet)}{\partial \mathbf{x}_{k-1}}\right)^2 \mathbf{\Sigma}_{x_i}^{(k-1|1:k-1)} + \left(\frac{\partial f(\bullet)}{\partial \mathbf{u}_{k-1}}\right)^2 \mathbf{\Sigma}_{u_{k-1}} + \left(\frac{\partial f(\bullet)}{\partial \mathbf{\Phi}}\right)^2 \mathbf{\Sigma}_{\Phi} + \mathbf{\Sigma}_w. \quad (6c)$$

Equation (6c) arises from error propagation of Eq. (2).

6.2.1.2 Observation

The second stage of RBE, observation, supplements the prediction stage with additional information about the state at step k . A deterministic observation model can be generally defined as a function of the state, various intrinsic parameters Ψ , and sensor noise \mathbf{v} by:

$$\mathbf{z}_k = h(\mathbf{x}_k, \Psi) + \mathbf{v}_k, \quad (7)$$

where \mathbf{z}_k is the observation at step k . The manner in which a deterministic observation can be cast into a probabilistic belief space to obtain an *observation* PDF $p(\mathbf{z}_k|\mathbf{x}_k)$ is largely context-dependent.

6.2.1.3 Correction

If prediction was not accompanied by some uncertainty, the system of interest would be purely deterministic and observation would be unnecessary. However, since system uncertainty is inevitable, the correction stage of RBE reinforces belief by fusing the predicted and observation PDFs:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_k|\mathbf{z}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int_{\mathcal{X}} p(\mathbf{x}_k|\mathbf{z}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k}. \quad (8)$$

Here, $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ may be referred to as the *corrected* PDF. To compute Eq. (8), $p(\mathbf{x}_k|\mathbf{z}_k)$ must be derived from the observation PDF $p(\mathbf{z}_k|\mathbf{x}_k)$. In the multi-Gaussian case, this distribution can be given by:

$$p(\mathbf{x}_k|\mathbf{z}_k) = \sum_{j=1}^J c_j^{(k|k)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_j^{(k|k)}, \Sigma_{x_j}^{(k|k)}). \quad (9)$$

For such a PDF, correction is accomplished by multi-Gaussian Belief Fusion (MBF) [23]. This results in the following corrected PDF:

$$\begin{aligned}
p(\mathbf{x}_k | \mathbf{z}_{1:k}) &= \sum_{i=1}^I \sum_{j=1}^J c_i^{(k|1:k-1)} c_j^{(k|k)} c_{ij}^{(k|1:k-1,k)} \mathcal{N} \left(\mathbf{x}; \bar{\mathbf{x}}_{ij}^{(k|1:k-1,k)}, \boldsymbol{\Sigma}_{x_{ij}}^{(k|1:k-1,k)} \right) \\
&= \sum_{n=1}^{IJ} c_n^{(k|1:k)} \mathcal{N} \left(\mathbf{x}; \bar{\mathbf{x}}_n^{(k|1:k)}, \boldsymbol{\Sigma}_{x_n}^{(k|1:k)} \right), \tag{10}
\end{aligned}$$

where

$$c_{ij}^{(k|1:k-1,k)} = \mathcal{N} \left(\bar{\mathbf{x}}_i^{(k|1:k-1)}; \bar{\mathbf{x}}_j^{(k|k)}, \boldsymbol{\Sigma}_{x_i}^{(k|1:k-1)} + \boldsymbol{\Sigma}_{x_j}^{(k|k)} \right), \tag{11a}$$

$$\boldsymbol{\Sigma}_{x_{ij}}^{(k|1:k-1,k)} = \left[\left(\boldsymbol{\Sigma}_{x_i}^{(k|1:k-1)} \right)^{-1} + \left(\boldsymbol{\Sigma}_{x_j}^{(k|k)} \right)^{-1} \right]^{-1}, \tag{11b}$$

$$\bar{\mathbf{x}}_{ij}^{(k|1:k-1,k)} = \boldsymbol{\Sigma}_{x_{ij}}^{(k|1:k-1,k)} \left[\left(\boldsymbol{\Sigma}_{x_i}^{(k|1:k-1)} \right)^{-1} \bar{\mathbf{x}}_i^{(k|1:k-1)} + \left(\boldsymbol{\Sigma}_{x_j}^{(k|k)} \right)^{-1} \bar{\mathbf{x}}_j^{(k|k)} \right]. \tag{11c}$$

6.2.2 Kalman Estimation

Most KFs assume belief is strictly Gaussian and therefore propagate only the mean vectors and covariance matrices describing Gaussian belief. The linear and adaptive Kalman filters are described here.

6.2.2.1 Linear Kalman Filter

The LKF predicts the mean and covariance describing Gaussian belief about a state according to a system's linear motion model as:

$$\bar{\mathbf{x}}_{k|1:k-1} = \mathbf{A} \bar{\mathbf{x}}_{k-1|1:k-1} + \mathbf{B} \mathbf{u}_{k-1}, \tag{12a}$$

$$\boldsymbol{\Sigma}_{x_{k|1:k-1}} = \mathbf{A} \boldsymbol{\Sigma}_{x_{k-1|1:k-1}} \mathbf{A}^T + \boldsymbol{\Sigma}_w. \tag{12b}$$

with $\boldsymbol{\Sigma}_w$ being the covariance of additive Gaussian system disturbance noise. Observation is also a linear process governed by:

$$\mathbf{z}_k = \mathbf{C} \mathbf{x}_k + \mathbf{v}_k. \tag{13}$$

Once an observation is received, belief is improved by correction, given in terms of the linear observation model \mathbf{C} and Kalman gain \mathbf{K} :

$$\bar{\mathbf{x}}_{k|1:k} = \bar{\mathbf{x}}_{k|1:k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{C}\bar{\mathbf{x}}_{k|1:k-1}), \quad (14a)$$

$$\boldsymbol{\Sigma}_{x_{k|1:k}} = (\mathbf{I} - \mathbf{K}\mathbf{C})\boldsymbol{\Sigma}_{x_{k|1:k-1}}, \quad (14b)$$

with

$$\mathbf{K} = \boldsymbol{\Sigma}_{x_{k|1:k-1}}\mathbf{C}^T \left(\mathbf{C}\boldsymbol{\Sigma}_{x_{k|1:k-1}}\mathbf{C}^T + \boldsymbol{\Sigma}_v \right)^{-1}. \quad (15)$$

The variable $\boldsymbol{\Sigma}_v$ represents the covariance of additive Gaussian sensor noise.

6.2.2.2 Adaptive Kalman Filter

The AKF predicts and corrects according to the appropriate motion and observation models, as in equations (12a)-(15) for a linear system and sensor, but accounts for increased uncertainty by updating $\boldsymbol{\Sigma}_w$ and $\boldsymbol{\Sigma}_v$. One of the more popular approaches is described in [22] by the following equations:

$$\boldsymbol{\Sigma}_{v_k} = \mathbf{G}_v - \mathbf{C}\boldsymbol{\Sigma}_{x_{k|1:k-1}}\mathbf{C}^T, \quad (16a)$$

$$\mathbf{G}_v = \frac{1}{M} \sum_{m=1}^M \left\| \bar{\mathbf{z}}_{k-m} - \mathbf{C}_{k-m}\bar{\mathbf{x}}_{k-m|1:k-m-1} \right\|^2, \quad (16b)$$

$$\boldsymbol{\Sigma}_{w_k} = \sqrt{\alpha}\boldsymbol{\Sigma}_{w_{k-1}}, \quad (16c)$$

$$\alpha = \frac{\text{tr}(\mathbf{G}_v - \boldsymbol{\Sigma}_{v_{k-1}})}{\text{tr}(\mathbf{C}\boldsymbol{\Sigma}_{x_{k|1:k-1}}\mathbf{C}^T)}. \quad (16d)$$

As the above formulas show, while the LKF and AKF account for uncertainty in the form of additive noise in both prediction and observation, there is no allowance for uncertainty in the model parameters \mathbf{A} , \mathbf{B} , and \mathbf{C} themselves. This can result in non-zero estimation error when the motion and/or observation model does not match the physical system and/or sensor.

6.2.3 Multi-Gaussian Motion Model Correction

Estimation accuracy is increased under the SEAM framework [16] because uncertainty in the motion model is incorporated in the prediction stage by introducing the $\frac{\partial f(\bullet)}{\partial \Phi}$ term of Eq. (6c). Because this also increases sensitivity to observational noise, a model correction stage then improves the estimated motion model parameters and consequently decreases their uncertainty to make prediction more reliable. This is accomplished in [16] by means of a gradient-descent approach which adjusts model parameters to minimize the integrated-squared-error (ISE_f) between $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ as shown below:

$$ISE_f = \int_{\mathbf{x}} (p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) - p(\mathbf{x}_k|\mathbf{z}_{1:k}))^2 d\mathbf{x}_k. \quad (17)$$

The subscript f is introduced here to emphasize that the past SEAM framework only handled motion model parameters. Because each of these PDFs is generally multi-Gaussian, ISE_f simplifies to an expression containing only weighting coefficients, some of which implicitly depend on motion model parameters Φ . Since a closed-form solution for the optimal motion model parameters Φ_k at step k is intractable, the gradient $\nabla_{\Phi}(ISE_f)$ is instead extracted and used in gradient-descent algorithms to solve for these optimal parameters:

$$\Phi_k = \underset{\Phi}{\operatorname{argmin}}(ISE_f). \quad (18)$$

Although previous work addresses motion model correction for non-Gaussian belief, congruous correction formulations for the observation model have not been addressed. Because the motion and observation models have duality in the context of RBE, it is a natural extension to apply similar derivations in obtaining an observation model correction stage within SEAM.

6.3 Multi-Gaussian Observation Model Correction

The major contributions of this paper are described here within the context of SEAM. First, formulations for Gaussian and multi-Gaussian observational belief are proposed for problems where model parameters are not well known. Included are equations for observation inversion and artificial re-observation. Next, model parameter correction is addressed for single and then multiple re-observation error minimization. Finally, a method of model parameter uncertainty propagation is introduced for consistency within SEAM.

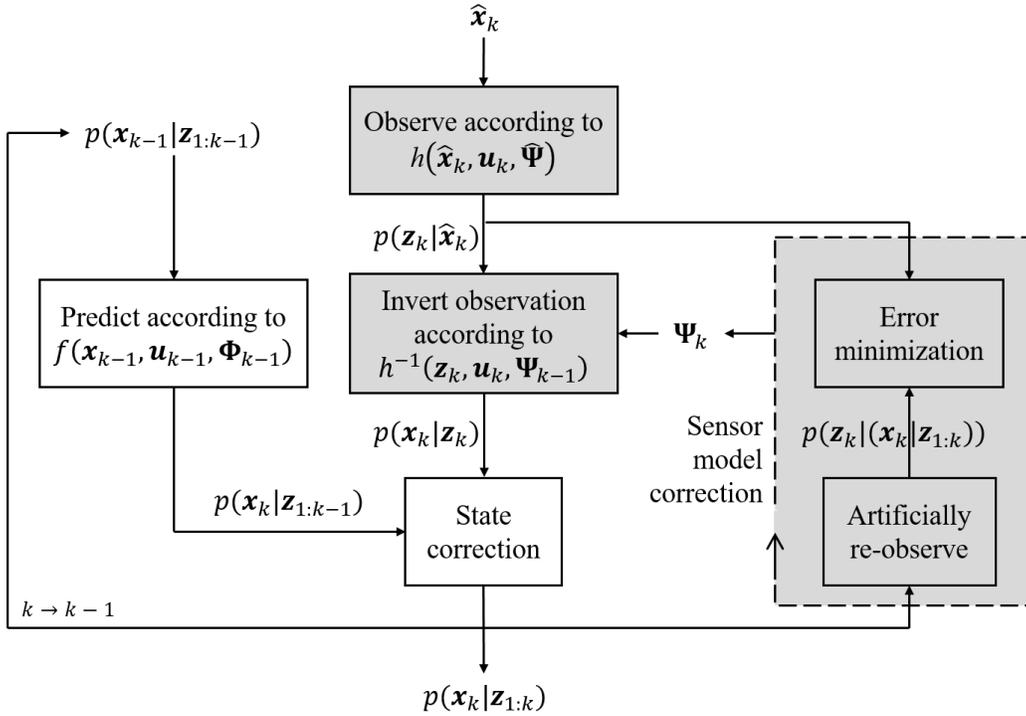


Figure 1. System diagram (SEAM with observation model correction).

The proposed framework is summarized by the diagram of Fig. 1. Shaded blocks are the particular original contributions which are described in the following subsections.

6.3.1 Observation Formulations

Consider a linearized observation model of the following form:

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k. \quad (19)$$

where the zero-mean noise process \mathbf{v}_k has covariance $\mathbf{\Sigma}_v$. Although the concepts of this section could apply to observation matrices \mathbf{C} of varying dimensionality and rank, \mathbf{z}_k is assumed here to have the same dimensionality as \mathbf{x}_k so that \mathbf{C} is square. The expected value of a reading from a sensor for which Eq. (19) holds is $\bar{\mathbf{z}}_k = \mathbf{C}\hat{\mathbf{x}}_k$, where $\hat{\mathbf{x}}_k$ represents the ground-truth state. When observational belief is Gaussian, $p(\mathbf{z}_k|\hat{\mathbf{x}}_k)$ is first proposed to be given by the following normal distribution:

$$p(\mathbf{z}_k|\hat{\mathbf{x}}_k) = \mathcal{N}(\mathbf{z}; \bar{\mathbf{z}}_k, \mathbf{\Sigma}_{z_k}), \quad (20)$$

where

$$\bar{\mathbf{z}}_k = \bar{\mathbf{C}}\hat{\mathbf{x}}_k \quad (21)$$

and

$$\mathbf{\Sigma}_{z_k} = \mathbf{\Sigma}_v + \mathbf{\Sigma}_C \mathcal{D}(\hat{\mathbf{x}}_k \circ \hat{\mathbf{x}}_k) \mathbf{\Sigma}_C^T. \quad (22)$$

In Eq. (22), $\mathbf{\Sigma}_C$ is a matrix the size of $\bar{\mathbf{C}}$ with each element equal to the standard deviation of the corresponding element in $\bar{\mathbf{C}}$, the matrix of estimated means. In addition, the notation $\mathcal{D}(\bullet)$ diagonalizes its vector argument and \circ represents the Hadamard element-wise product. This equation follows from error propagation on Eq. (19). It is important to note here the difference between “sensor noise” and “observational uncertainty”. The former is characterized by only $\mathbf{\Sigma}_v$, while the latter is characterized by $\mathbf{\Sigma}_{z_k}$, which incorporates both sensor noise and model uncertainty. When model uncertainty is assumed to be zero, as is the case with the vast majority of Kalman filters, observational uncertainty and sensor noise become equivalent and interchangeable.

In cases where observational belief is more accurately modeled by a zero-mean non-Gaussian PDF, the above formulations are insufficient. The proposed method models non-

Gaussian observational belief such that the linear observation model is still employed. Let $p(\mathbf{z}_k|\hat{\mathbf{x}}_k)$ be approximated by a Gaussian sum:

$$p(\mathbf{z}_k|\hat{\mathbf{x}}_k) = \sum_{j=1}^J c_j^{(k|\hat{k})} \mathcal{N}(\mathbf{z}; \bar{\mathbf{z}}_j^{(k|\hat{k})}, \boldsymbol{\Sigma}_{z_j}^{(k|\hat{k})}), \quad (23)$$

with overall mean $\bar{\mathbf{z}}_k$ equal to $\bar{\mathbf{C}}\hat{\mathbf{x}}_k$ and $\sum_{j=1}^J c_j^{(k|\hat{k})} = 1$. In order to incorporate the observation model in defining $\bar{\mathbf{z}}_j^{(k|\hat{k})}$ and $\boldsymbol{\Sigma}_{z_j}^{(k|\hat{k})}$, the linear mean equation is augmented as follows:

$$\bar{\mathbf{z}}_j^{(k|\hat{k})} = \bar{\mathbf{C}}(\hat{\mathbf{x}}_k + \boldsymbol{\delta}_j), \quad (24)$$

while the covariance of each Gaussian component becomes

$$\boldsymbol{\Sigma}_{z_j}^{(k|\hat{k})} = \boldsymbol{\Sigma}_v + \boldsymbol{\Sigma}_C \mathcal{D}((\hat{\mathbf{x}}_k + \boldsymbol{\delta}_j) \circ (\hat{\mathbf{x}}_k + \boldsymbol{\delta}_j)) \boldsymbol{\Sigma}_C^T. \quad (25)$$

The centroid of the delta points $\boldsymbol{\delta}_j$ is restricted to be zero so that the mean of $p(\mathbf{z}_k|\hat{\mathbf{x}}_k)$ remains equal to $\bar{\mathbf{C}}\hat{\mathbf{x}}_k$. Furthermore, the weighting coefficients $c_j^{(k|\hat{k})}$ must be balanced for the same reason. The variables $\boldsymbol{\Sigma}_v$, $\boldsymbol{\delta}_j$, and $c_j^{(k|\hat{k})}$ are sensor-intrinsic properties to be determined by off-line characterization in addition to the intrinsic parameter matrix $\bar{\mathbf{C}}$. Figure 2 demonstrates Gaussian and non-Gaussian observation modeling for an arbitrary one-dimensional (1-D) example.

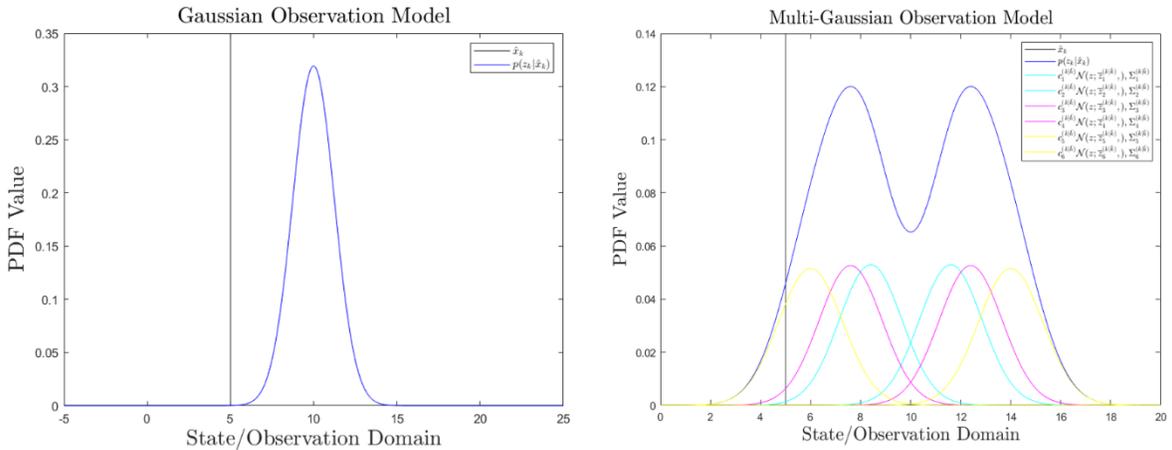


Figure 2. Gaussian and multi-Gaussian observation models where $\bar{\mathbf{C}} = 2$. This arbitrary multi-Gaussian model for which $c_j^{(k|\hat{k})} = 1/6 \forall j$ and $\boldsymbol{\delta}_j = [0.8, -0.8, 1.2, -1.2, 2, -2]$ displays bimodal characteristics.

Observations are associated with some level of uncertainty, and so are treated as random vectors drawn from the distribution $p(\mathbf{z}_k|\hat{\mathbf{x}}_k)$. Accurately characterizing the uncertainty in such observations as given by Eqs. (22) and (25) depends on knowledge of the “hidden” ground-truth state $\hat{\mathbf{x}}_k$ as these formulas show. Because this is the target of estimation and is therefore unknown, an approximation must be made. The component means $\bar{\mathbf{z}}_j^{(k|\hat{k})}$ are estimated by assuming that a received observation $\hat{\mathbf{z}}_k$ has the expected value $\bar{\mathbf{z}}_k$. This means:

$$\bar{\mathbf{z}}_j^{(k|\hat{k})} = \bar{\mathbf{C}}(\hat{\mathbf{x}}_k + \boldsymbol{\delta}_j) = \bar{\mathbf{z}}_k + \bar{\mathbf{C}}\boldsymbol{\delta}_j \cong \hat{\mathbf{z}}_k + \bar{\mathbf{C}}\boldsymbol{\delta}_j, \quad (26)$$

and so

$$\boldsymbol{\Sigma}_{\mathbf{z}_j}^{(k|\hat{k})} = \boldsymbol{\Sigma}_v + \boldsymbol{\Sigma}_c \mathcal{D} \left(\left(\bar{\mathbf{C}}^{-1} \bar{\mathbf{z}}_j^{(k|\hat{k})} \right) \circ \left(\bar{\mathbf{C}}^{-1} \bar{\mathbf{z}}_j^{(k|\hat{k})} \right) \right) \boldsymbol{\Sigma}_c^T. \quad (27)$$

6.3.1.1 Observation Inversion

SEAM is concerned with estimating the state of a system, so vectors in the observation domain cannot be directly fused with predicted states (with the exception of the trivial case where $\mathbf{C} = \mathbf{I}$). As a result, an observation must be inverted to obtain $p(\mathbf{x}_k|\mathbf{z}_k)$ from $p(\mathbf{z}_k|\mathbf{x}_k)$. This stage is termed “observation inversion” here. Inverting a multi-Gaussian observation can be thought of as applying another linear propagation to the PDF coming from an observation, $p(\mathbf{z}_k|\hat{\mathbf{x}}_k)$. This propagation is applied to each component of the distribution to yield another Gaussian sum:

$$p(\mathbf{x}_k|\mathbf{z}_k) = \sum_{j=1}^J c_j^{(k|k)} \mathcal{N} \left(\mathbf{x}; \bar{\mathbf{x}}_j^{(k|k)}, \boldsymbol{\Sigma}_{\mathbf{x}_j}^{(k|k)} \right), \quad (28)$$

where

$$c_j^{(k|k)} = c_j^{(k|\hat{k})}, \quad (29a)$$

$$\bar{\mathbf{x}}_j^{(k|k)} = \bar{\mathbf{C}}^{-1} \bar{\mathbf{z}}_j^{(k|\hat{k})}, \quad (29b)$$

$$\boldsymbol{\Sigma}_{x_j}^{(k|k)} = \boldsymbol{\Sigma}_{C^{-1}} \mathcal{D} \left(\bar{\mathbf{z}}_j^{(k|\hat{k}) \circ 2} \right) \boldsymbol{\Sigma}_{C^{-1}}^T + \bar{\mathbf{C}}^{-1} \left(\boldsymbol{\Sigma}_{z_j}^{(k|\hat{k})} + \boldsymbol{\Sigma}_v \right) \bar{\mathbf{C}}^{-1T}, \quad (29c)$$

and the power notation “ $\circ 2$ ” more concisely denotes the element-wise square of the corresponding vector. Again, the covariance formula is derived from error propagation.

Inspection of Eq. (29c) reveals the need for $\boldsymbol{\Sigma}_{C^{-1}}$, the matrix containing standard deviations of corresponding elements in the mean inverse matrix $\bar{\mathbf{C}}^{-1}$. In general, this matrix cannot be obtained; however, it can be approximated by Monte Carlo methods or other techniques [24]. In practice, however, $\boldsymbol{\Sigma}_{C^{-1}}$ is approximated using some effective heuristic, as the primary objective is to broadly capture the inflation of uncertainty consistent with the inverting of a stochastic observation to the underlying state.

6.3.1.2 Artificial Re-Observation

The aim of “artificial re-observation” is to simulate an observation of the corrected state under the assumption that it is more accurately estimated than the inverse-observed state alone. The PDF describing the re-observed state is denoted $p(\mathbf{z}_k | (\mathbf{x}_k | \mathbf{z}_{1:k}))$. Again, this is given by a multi-Gaussian distribution:

$$p(\mathbf{z}_k | (\mathbf{x}_k | \mathbf{z}_{1:k})) = \sum_{n=1}^{IJ} c_n^{(k|k|1:k)} \mathcal{N} \left(\mathbf{z}; \bar{\mathbf{z}}_n^{(k|k|1:k)}, \boldsymbol{\Sigma}_{z_n}^{(k|k|1:k)} \right), \quad (30)$$

where

$$c_n^{(k|k|1:k)} = c_n^{(k|1:k)}, \quad (31a)$$

$$\bar{\mathbf{z}}_n^{(k|k|1:k)} = \bar{\mathbf{C}} \bar{\mathbf{x}}_n^{(k|1:k)}, \quad (31b)$$

$$\boldsymbol{\Sigma}_{z_n}^{(k|k|1:k)} = \boldsymbol{\Sigma}_C \mathcal{D} \left(\bar{\mathbf{x}}_n^{(k|1:k) \circ 2} \right) \boldsymbol{\Sigma}_C^T + \bar{\mathbf{C}} \boldsymbol{\Sigma}_{x_n}^{(k|1:k)} \bar{\mathbf{C}}^T + \boldsymbol{\Sigma}_v. \quad (31c)$$

Recall that the parameters with superscripts $(k|1:k)$ define $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ as given by Eq. (10).

6.3.2 Single Re-observation Error Minimization

In order to effectively correct observation model parameter estimates, the proposed technique compares observed and re-observed belief and adjusts observation model parameters such that the two PDFs match optimally. Mathematically, this is accomplished by minimizing an objective function ISE_h . The optimal parameter vector Ψ_k at step k is found by

$$\Psi_k = \underset{\Psi}{\operatorname{argmin}}(ISE_h). \quad (32)$$

where ISE_h is defined as

$$ISE_h = \int_{\mathbf{z}} (p(\mathbf{z}_k | \hat{\mathbf{x}}_k) - p(\mathbf{z}_k | (\mathbf{x}_k | \mathbf{z}_{1:k})))^2 d\mathbf{z}. \quad (33)$$

Because each of these PDFs is generally given by a Gaussian sum, a closed-form solution for model parameters which minimize ISE_h is unattainable. However, since $p(\mathbf{z}_k | (\mathbf{x}_k | \mathbf{z}_{1:k}))$ is a function of the estimated parameter vector Ψ , a gradient descent approach can be used.

The integrated-squared-error simplifies under the Gaussian sum approximation to:

$$ISE_h = \int_{\mathbf{z}} \left(\left(\sum_{j=1}^J c_j^{(k|\hat{k})} \mathcal{N}(\mathbf{z}; \bar{\mathbf{z}}_j^{(k|\hat{k})}, \Sigma_{z_j}^{(k|\hat{k})}) \right)^2 + \left(\sum_{n=1}^{IJ} c_n^{(k|k|1:k)} \mathcal{N}(\mathbf{z}; \bar{\mathbf{z}}_n^{(k|k|1:k)}, \Sigma_{z_n}^{(k|k|1:k)}) \right)^2 - 2 \sum_{j=1}^J \sum_{n=1}^{IJ} c_j^{(k|\hat{k})} c_n^{(k|k|1:k)} c_{jn}^{(k|\hat{k},k|1:k)} \mathcal{N}(\mathbf{z}; \bar{\mathbf{z}}_{jn}^{(k|\hat{k},k|1:k)}, \Sigma_{z_{jn}}^{(k|\hat{k},k|1:k)}) \right) d\mathbf{z}. \quad (34)$$

With further expansion, and because each Gaussian integrates to one, this becomes:

$$ISE_h = \sum_{j=1}^J \sum_{l=1}^J c_j^{(k|\hat{k})} c_l^{(k|\hat{k})} c_{jl}^{(k|\hat{k},\hat{k})} + \sum_{n=1}^{IJ} \sum_{m=1}^{IJ} c_n^{(k|k|1:k)} c_m^{(k|k|1:k)} c_{nm}^{(k|k|1:k,k|1:k)} - 2 \sum_{j=1}^J \sum_{n=1}^{IJ} c_j^{(k|\hat{k})} c_n^{(k|k|1:k)} c_{jn}^{(k|\hat{k},k|1:k)}. \quad (35)$$

In this expression, only $c_{nm}^{(k|k|1:k,k|1:k)}$ and $c_{jn}^{(k|\hat{k},k|1:k)}$ depend directly on Ψ , as parameters with $(k|\hat{k})$ superscripts are taken to be constant since they define the reference PDF coming from an observation $p(\mathbf{z}_k|\hat{\mathbf{x}}_k)$. Therefore, the gradient of ISE_h with respect to Ψ is given by:

$$\begin{aligned} \nabla_{\Psi}(ISE_h) &= \sum_{n=1}^{IJ} \sum_{m=1}^{IJ} c_n^{(k|k|1:k)} c_m^{(k|k|1:k)} \nabla_{\Psi} \left(c_{nm}^{(k|k|1:k,k|1:k)} \right) \\ &\quad - 2 \sum_{j=1}^J \sum_{n=1}^{IJ} c_j^{(k|\hat{k})} c_n^{(k|k|1:k)} \nabla_{\Psi} \left(c_{jn}^{(k|\hat{k},k|1:k)} \right). \end{aligned} \quad (36)$$

Because they come from MBF of two multi-Gaussian PDFs, $c_{nm}^{(k|k|1:k,k|1:k)}$ and $c_{jn}^{(k|\hat{k},k|1:k)}$ are given by normal distributions according to Eq. (11a). By the chain rule, their gradients can be expressed as follows:

$$\nabla_{\Psi} \left(c_{nm}^{(k|k|1:k,k|1:k)} \right) = \exp(\beta_{nm}) \left(\nabla_{\Psi}(\alpha_{nm}) + \alpha_{nm} \nabla_{\Psi}(\beta_{nm}) \right), \quad (37a)$$

$$\nabla_{\Psi} \left(c_{jn}^{(k|\hat{k},k|1:k)} \right) = \exp(\zeta_{nm}) \left(\nabla_{\Psi}(\epsilon_{nm}) + \epsilon_{nm} \nabla_{\Psi}(\zeta_{nm}) \right), \quad (37b)$$

where

$$\alpha_{nm} = (|2\pi\boldsymbol{\gamma}_{nm}|)^{-\frac{1}{2}}, \quad (38a)$$

$$\beta_{nm} = -\frac{1}{2} \boldsymbol{\delta}_{nm}^T \boldsymbol{\gamma}_{nm}^{-1} \boldsymbol{\delta}_{nm}, \quad (38b)$$

$$\epsilon_{jn} = (|2\pi\boldsymbol{\eta}_{jn}|)^{-\frac{1}{2}}, \quad (38c)$$

$$\zeta_{jn} = -\frac{1}{2} \boldsymbol{\theta}_{jn}^T \boldsymbol{\eta}_{jn}^{-1} \boldsymbol{\theta}_{jn}, \quad (38d)$$

and

$$\begin{aligned} \boldsymbol{\gamma}_{nm} &= \boldsymbol{\Sigma}_{z_n}^{(k|k|1:k)} + \boldsymbol{\Sigma}_{z_m}^{(k|k|1:k)} \\ &= \boldsymbol{\Sigma}_C \mathcal{D} \left(\bar{\mathbf{x}}_n^{(k|1:k)\circ 2} + \bar{\mathbf{x}}_m^{(k|1:k)\circ 2} \right) \boldsymbol{\Sigma}_C^T + \bar{\mathbf{C}} \left(\boldsymbol{\Sigma}_{x_n}^{(k|1:k)} + \boldsymbol{\Sigma}_{x_m}^{(k|1:k)} \right) \bar{\mathbf{C}}^T + 2\boldsymbol{\Sigma}_v, \end{aligned} \quad (39a)$$

$$\boldsymbol{\delta}_{nm} = \bar{\mathbf{z}}_n^{(k|k|1:k)} - \bar{\mathbf{z}}_m^{(k|k|1:k)} = \bar{\mathbf{C}} \left(\bar{\mathbf{x}}_n^{(k|1:k)} + \bar{\mathbf{x}}_m^{(k|1:k)} \right), \quad (39b)$$

$$\boldsymbol{\eta}_{jn} = \boldsymbol{\Sigma}_{z_j}^{(k|\hat{k})} + \boldsymbol{\Sigma}_{z_n}^{(k|k|1:k)} = \boldsymbol{\Sigma}_{z_j}^{(k|\hat{k})} + \boldsymbol{\Sigma}_C \mathcal{D} \left(\bar{\mathbf{x}}_n^{(k|1:k)^2} \right) \boldsymbol{\Sigma}_C^T + \bar{\mathbf{C}} \boldsymbol{\Sigma}_{x_n}^{(k|1:k)} \bar{\mathbf{C}}^T + \boldsymbol{\Sigma}_v, \quad (39c)$$

$$\boldsymbol{\theta}_{jn} = \bar{\mathbf{z}}_j^{(k|\hat{k})} - \bar{\mathbf{z}}_n^{(k|k|1:k)} = \bar{\mathbf{z}}_j^{(k|\hat{k})} - \bar{\mathbf{C}} \bar{\mathbf{x}}_n^{(k|1:k)}. \quad (39d)$$

Because the vector $\boldsymbol{\Psi}$ is assembled from the observation model parameters (i.e. the indexed elements C_{pq} of \mathbf{C} , the four gradients of Eqs. (37a) and (37b) must be assembled from the partial derivatives of their arguments with respect to each parameter. For example,

$$\nabla_{\boldsymbol{\Psi}}(\alpha_{nm}) = \left[\frac{\partial \alpha_{nm}}{\partial C_{11}} \quad \dots \quad \frac{\partial \alpha_{nm}}{\partial C_{1q}} \quad \frac{\partial \alpha_{nm}}{\partial C_{21}} \quad \dots \quad \frac{\partial \alpha_{nm}}{\partial C_{2q}} \quad \dots \quad \frac{\partial \alpha_{nm}}{\partial C_{pq}} \right]^T. \quad (40)$$

Each placeholder variable's gradient is found accordingly. Evaluating the partial derivatives of Eqs. (37a) and (37b) gives the following:

$$\frac{\partial \alpha_{nm}}{\partial C_{pq}} = -\frac{1}{2} (|2\pi \boldsymbol{\gamma}_{nm}|)^{-\frac{1}{2}} \text{tr} \left(\boldsymbol{\gamma}_{nm}^{-1} \frac{\partial \boldsymbol{\gamma}_{nm}}{\partial C_{pq}} \right), \quad (41a)$$

$$\frac{\partial \beta_{nm}}{\partial C_{pq}} = -\frac{1}{2} \left(\boldsymbol{\delta}_{nm}^T \boldsymbol{\gamma}_{nm}^{-1} \left(\frac{\partial \boldsymbol{\delta}_{nm}}{\partial C_{pq}} - \frac{\partial \boldsymbol{\gamma}_{nm}}{\partial C_{pq}} \boldsymbol{\gamma}_{nm}^{-1} \boldsymbol{\delta}_{nm} \right) + \left(\frac{\partial \boldsymbol{\delta}_{nm}}{\partial C_{pq}} \right)^T \boldsymbol{\gamma}_{nm}^{-1} \boldsymbol{\delta}_{nm} \right), \quad (41b)$$

$$\frac{\partial \epsilon_{jn}}{\partial C_{pq}} = -\frac{1}{2} (|2\pi \boldsymbol{\eta}_{jn}|)^{-\frac{1}{2}} \text{tr} \left(\boldsymbol{\eta}_{jn}^{-1} \frac{\partial \boldsymbol{\eta}_{jn}}{\partial C_{pq}} \right), \quad (41c)$$

$$\frac{\partial \zeta_{jn}}{\partial C_{pq}} = -\frac{1}{2} \left(\boldsymbol{\theta}_{jn}^T \boldsymbol{\eta}_{jn}^{-1} \left(\frac{\partial \boldsymbol{\theta}_{jn}}{\partial C_{pq}} - \frac{\partial \boldsymbol{\eta}_{jn}}{\partial C_{pq}} \boldsymbol{\eta}_{jn}^{-1} \boldsymbol{\theta}_{jn} \right) + \left(\frac{\partial \boldsymbol{\theta}_{jn}}{\partial C_{pq}} \right)^T \boldsymbol{\eta}_{jn}^{-1} \boldsymbol{\theta}_{jn} \right), \quad (41d)$$

and the final derivatives are given by:

$$\frac{\partial \boldsymbol{\gamma}_{nm}}{\partial C_{pq}} = \bar{\mathbf{C}} \left(\boldsymbol{\Sigma}_{x_n}^{(k|1:k)} + \boldsymbol{\Sigma}_{x_m}^{(k|1:k)} \right) \mathbf{O}_{pq}^T + \mathbf{O}_{pq} \left(\boldsymbol{\Sigma}_{x_n}^{(k|1:k)} + \boldsymbol{\Sigma}_{x_m}^{(k|1:k)} \right) \bar{\mathbf{C}}^T, \quad (42a)$$

$$\frac{\partial \boldsymbol{\delta}_{nm}}{\partial C_{pq}} = \mathbf{O}_{pq} \left(\bar{\mathbf{x}}_n^{(k|1:k)} + \bar{\mathbf{x}}_m^{(k|1:k)} \right), \quad (42b)$$

$$\frac{\partial \boldsymbol{\eta}_{jn}}{\partial C_{pq}} = \bar{\mathbf{C}} \boldsymbol{\Sigma}_{x_n}^{(k|1:k)} \mathbf{O}_{pq}^T + \mathbf{O}_{pq} \boldsymbol{\Sigma}_{x_n}^{(k|1:k)} \bar{\mathbf{C}}^T, \quad (42c)$$

$$\frac{\partial \theta_{jn}}{\partial C_{pq}} = -\mathbf{O}_{pq} \bar{\mathbf{x}}_n^{(k|1:k)}. \quad (42d)$$

In the above equations, the matrix \mathbf{O}_{pq} is defined as follows:

$$\mathbf{O}_{pq} = \begin{cases} 1 & \text{element } p, q \\ 0 & \text{elsewhere} \end{cases} \quad (43)$$

These formulations are used to implement gradient descent and find the parameter vector Ψ_k which minimizes the observation model objective function ISE_h for observation k . While the focus of this paper is not on particularly effective gradient-descent algorithms, the general process is governed by the following recurrence relation:

$$(\bar{\Psi}_k)_{i+1} = (\bar{\Psi}_k)_i - d_i [\nabla_{\Psi} (ISE_h)]|_{\Psi=(\bar{\Psi}_k)_i}. \quad (44)$$

The intuitive choice for seeding the algorithm is $(\bar{\Psi}_k)_i = \bar{\Psi}_{k-1}$, as this has the best chance of avoiding any anticipated non-convexities in the objective function which may result in convergence to an erroneous local minimum.

Figure 3 demonstrates the principle of observation model correction via ISE_h minimization. The relevant Gaussian PDFs of an arbitrary 1-D example are shown and ISE_h is plotted as a function of the 1-D observation model parameter C . In this example, the ground-truth state at step 2, \hat{x}_2 , is identified by a delta function² at 5. This particular observation model with linear bias $C_{GT} = 1.93$ yields an observation whose PDF is given by the red Gaussian curve with mean $1.93 \times 5 = 9.65$. Inverting this observation yields belief about x_2 which is shown by the green Gaussian PDF. Note that in each of these steps, uncertainty has increased in keeping with Eqs. (25) and (29c). The PDF coming from prediction, $p(x_2|z_1)$, is plotted in blue, while the corrected state belief given by the fusion of $p(x_2|z_2)$ and $p(x_2|z_1)$ is given by the magenta curve, $p(x_2|z_{1:2})$. Applying the principle of artificial re-observation to this PDF yields the cyan PDF.

² The Dirac delta can also be thought of as a Gaussian with infinitesimal variance.

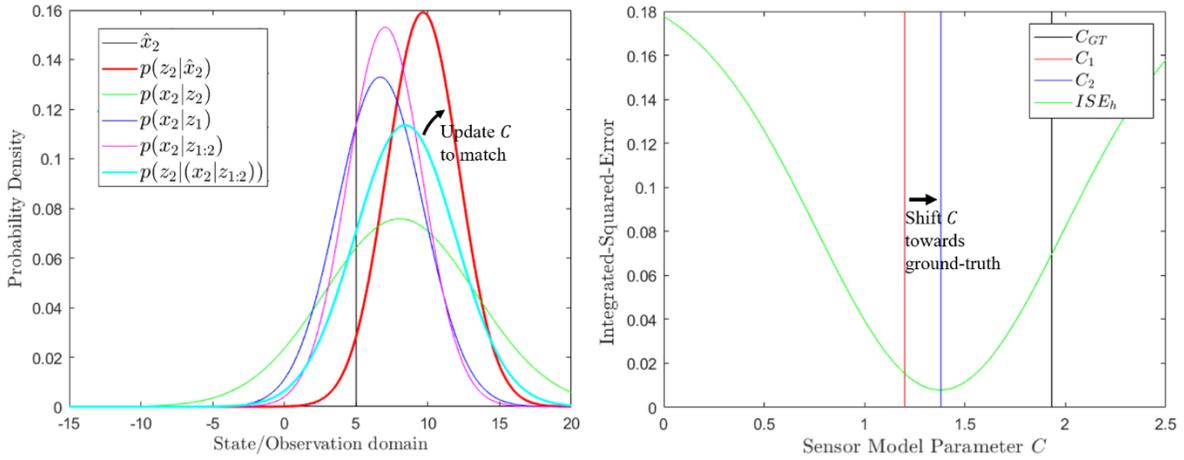


Figure 3. a) Belief through multiple stages of SEAM represented as 1-D Gaussian PDFs; b) The objective function ISE_h plotted over a range of observation model parameters C .

Model correction attempts to match the cyan PDF with the red PDF as was expressed mathematically in Eq. (33). This objective function is plotted in green in the second part of the figure, showing a minimum (blue) which improves the initial estimate of C (red) towards the ground-truth value (black). The minimum of ISE_h does not lie on C_{GT} as a result of the additive noise introduced in the observation step. However, over several time steps, the observation model parameter estimate tends to approach ground-truth because of the zero-mean-stationarity of sensor noise.

6.3.3 Multiple Re-observation Error Minimization

As the dimensionality N of \mathbf{C} increases, the length of the parameter vector Ψ increases by N^2 . Therefore, moderately high-dimensional problems mandate the optimization of much higher-dimensional parameter vectors. For cases where $N > 1$, the previously formulated objective function does not provide sufficient power to constrain unique convergence of all the parameter estimates to ground-truth. For this reason, ISE_h is redefined to average the re-observation error over multiple time steps:

$$ISE_h = \int_{\mathbf{z}} \left[\frac{1}{M+1} \sum_{i=k-M}^k (p(\mathbf{z}_i|\hat{\mathbf{x}}_i) - p(\mathbf{z}_i|(\mathbf{x}_i|\mathbf{z}_{1:i})))^2 \right] d\mathbf{z}. \quad (45)$$

The linearity of integration allows for the following simplification:

$$ISE_h = \frac{1}{M+1} \sum_{i=k-M}^k \int_{\mathbf{z}} (p(\mathbf{z}_i|\hat{\mathbf{x}}_i) - p(\mathbf{z}_i|(\mathbf{x}_i|\mathbf{z}_{1:i})))^2 d\mathbf{z} = \frac{1}{M+1} \sum_{i=k-M}^k (ISE_h)_i. \quad (46)$$

The gradient of the multiple-time-step ISE_h objective function is then given by the average of the individual gradients coming from M past time steps:

$$\nabla_{\Psi}(ISE_h) = \frac{1}{M+1} \sum_{i=k-M}^k \nabla_{\Psi}(ISE_h)_i. \quad (47)$$

6.3.4 Uncertainty Propagation

As presented in Eqs. (22), (25), and (31c) of section 3.1, an estimate of the uncertainty Σ_C in observation model parameters is required to accurately obtain observational belief. After the mean sensor matrix $\bar{\mathbf{C}}_k$ is made available via gradient descent model correction, the corresponding standard-deviation matrix is updated according to the following heuristic:

$$\Sigma_{C_k} \approx \lambda \Sigma_{C_{k-1}}, \quad 0 < \lambda < 1. \quad (48)$$

This guarantees that the estimated uncertainty in observation model parameters continually decreases following the power law λ^k to reflect the increasing accuracy with which $\bar{\mathbf{C}}_k$ is estimated.

6.4 Results

In order to assess the ubiquity of the proposed framework and avoid the selection of only special cases where it works, simulated experiments with randomly generated parameters were conducted. Under this Monte-Carlo validation study, both 1-D and 2-D experiments were simulated, including scenarios with both Gaussian and non-Gaussian belief.

6.4.1 1-D Validation

A 1-D linear system was simulated with constant parameter values assigned as shown in Table 1. C_0 is the initial estimate of the observation model parameter, with associated standard deviation Σ_{C_0} . d_i is the gradient-descent step size (held constant here), and T is the gradient-descent threshold parameter. The ground-truth parameter C_{GT} is randomly generated by sampling a normal distribution: $C_{GT} \sim \mathcal{N}(C_0, \Sigma_{C_0})$. The 1-D linear discrete-time system has an exponential solution which converges for $A < 1$. The system's step response is observed by setting $u_k = 1 \forall k > 0$.

Table 1. Constant parameters (1-D validation)

A_{GT}	B_{GT}	C_0	Σ_{C_0}	Σ_w	x_0	Σ_{x_0}	Δt	Σ_u	d_i	T
0.98	1.3	1	εC_0	0.01	2	0.09	0.025	0.01	0.001	0.0001

6.4.1.1 Uni-Gaussian Belief

For the simplest scenario where observational belief is represented by single Gaussians, the proposed framework was compared with both the LKF and the AKF. These provide examples of state estimation in which observation model uncertainty is not incorporated. In order to demonstrate the performance of SEAM with observation model correction, two estimators were deployed: the first incorporated observation model uncertainty but did not correct C (represented by SEAM⁻, while the second both accounted for model uncertainty and corrected C (represented by SEAM⁺). Model correction was implemented using all available observations ($M = k$). Figure 4 gives a sample simulation ($\Sigma_v = 1$, $\varepsilon = 0.1$, and $\lambda = 0.99$).

In these plots, the red trace corresponds to the *inverted* observations (the means of $p(x_k|z_k)$) coming from the SEAM⁺ estimator. While all the estimators have a smoothing effect on the noisy sensor signal, all but SEAM⁺ have non-zero estimation error due to their inability to invert observations based on a correct observation model. Although the sensor signal shown here

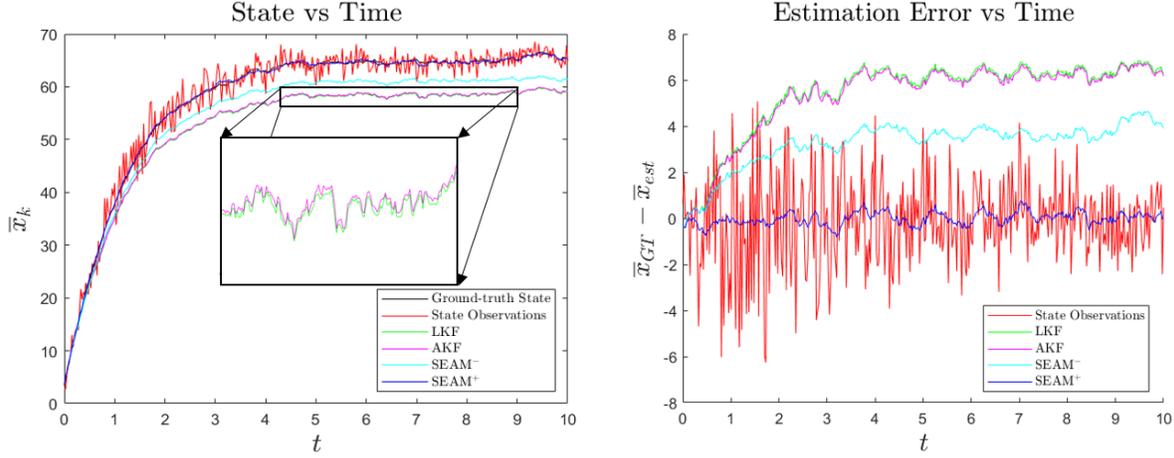


Figure 4. State and error of noisy observations and estimated signals over time.

remains centered on ground-truth, strong fluctuations result in high root-mean-squared-error (RMSE) and make any subsequent decision-making unreliable.

Figure 5 shows the trajectory of C for the same experiment of Fig. 4 as it is corrected over time. The model parameter estimate quickly improves, overshoots, and is gradually refined as more observations become available. This improvement is reflected by the performance of the SEAM⁺ estimator in the previous figure. For this

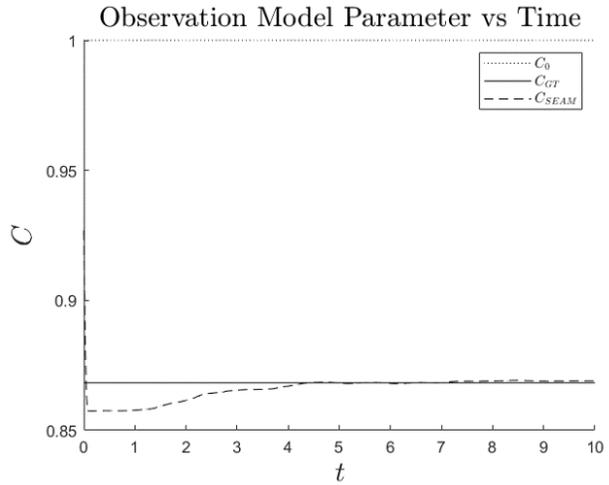


Figure 5. Observation model correction over time.

particular example, $C_{GT} = 0.8683$, and the estimated observation model parameter was improved 180% from an initial estimate of $C_0 = 1$ to a final estimate of $C_{SEAM} = 0.8690$.

For a more comprehensive and quantitative assessment, 50 simulations were executed for different combinations of model uncertainty Σ_{C_0} and noise variance Σ_v . Table 2 shows the percent of trials each estimator won (L = LKF, A = AKF, S⁻ = SEAM⁻, and S⁺ = SEAM⁺). A few noteworthy trends can be observed. First, as ε increases, the SEAM estimators tend to outperform the Kalman

filters as expected, because they account for the increased model uncertainty. This trend is especially noticeable for high sensor noise. Furthermore, as the variance in sensor noise increases, the estimators begin to perform fairly similarly, causing more spread in the distribution of winners. Because the estimators have similar performance under strong sensor noise, a “win” becomes less meaningful. For example, in a simulation where $\Sigma_v = 7.5^2$, the RMSEs are [1.07 1.32 0.79 1.22], whereas a sample simulation with $\Sigma_v = 1^2$ gives [11.09 11.19 2.70 1.11]. The average RMSE values of the KF estimators are sometimes actually lower under high sensor noise because Σ_v begins to dominate observational uncertainty, making overconfidence in the observation model parameters less penalizing.

Table 2. Parametric sensitivity study for 1-D Gaussian estimation

		ε							
		0.1	0.15	0.2	0.25				
Σ_v	0.5^2	L	0%	L	0%	L	0%	L	0%
		A	0%	A	0%	A	0%	A	0%
		S ⁻	16%	S ⁻	28%	S ⁻	26%	S ⁻	26%
		S ⁺	84%	S ⁺	72%	S ⁺	74%	S ⁺	74%
	1^2	L	2%	L	0%	L	0%	L	2%
		A	0%	A	0%	A	0%	A	0%
		S ⁻	8%	S ⁻	22%	S ⁻	16%	S ⁻	24%
		S ⁺	90%	S ⁺	78%	S ⁺	84%	S ⁺	74%
	2.5^2	L	2%	L	2%	L	4%	L	0%
		A	2%	A	0%	A	0%	A	2%
		S ⁻	12%	S ⁻	14%	S ⁻	12%	S ⁻	26%
		S ⁺	84%	S ⁺	84%	S ⁺	84%	S ⁺	72%
	5^2	L	30%	L	22%	L	10%	L	6%
		A	0%	A	2%	A	0%	A	0%
		S ⁻	6%	S ⁻	20%	S ⁻	30%	S ⁻	38%
		S ⁺	64%	S ⁺	56%	S ⁺	60%	S ⁺	56%
	7.5^2	L	74%	L	60%	L	36%	L	34%
		A	2%	A	0%	A	4%	A	2%
		S ⁻	6%	S ⁻	16%	S ⁻	44%	S ⁻	58%
		S ⁺	18%	S ⁺	24%	S ⁺	16%	S ⁺	6%

A final important comment regarding Table 2 explains the unexpected phenomenon that the performance of the proposed approach actually tends to decrease as model uncertainty increases. Because the initial model parameter estimates are further from ground-truth under this condition, it takes more time for the parameter estimate to converge. As a result, the first few seconds of simulation yield higher RMSE values which tend to overshadow the improved RMSE of the latter portion of simulation. This is demonstrated by Fig. 6 in the plot of a sample simulation for the severe scenario where $\Sigma_v = 7.5^2$ and $\varepsilon = 0.25$. (As a side-note, the AKF sometimes performs worse than the LKF due to poor conditioning of the randomly generated covariances. In cases where this is not an issue, estimation is improved as expected.)

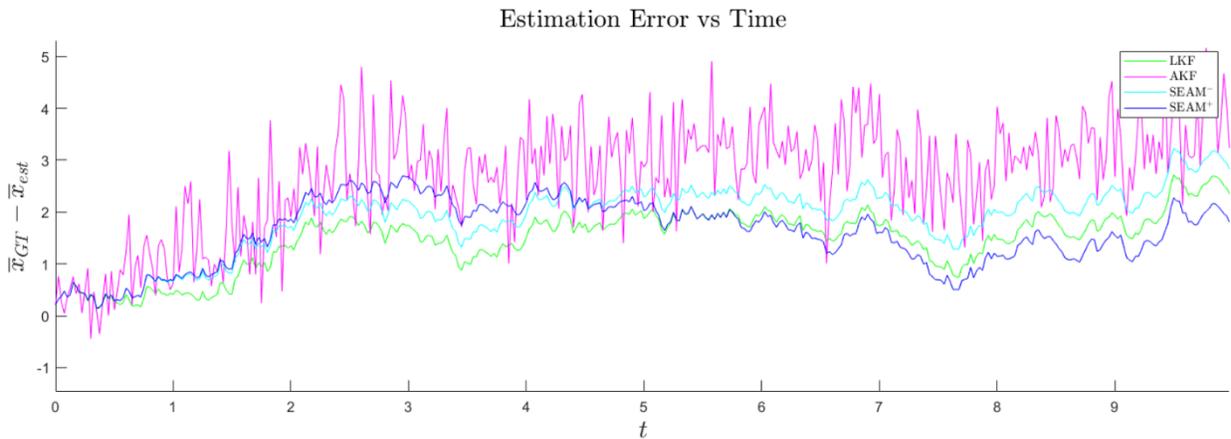


Figure 6. Estimation using $SEAM^+$ improves over time, while other estimators generally yield constant error.

To further demonstrate the ability of the proposed framework to correct observation model parameters, a series of ensemble curves were generated to show estimation subject to initial estimates of C_{GT} with varying levels of error. The error plots and model parameter plots are shown in Fig. 7. As the figure shows, initial estimates within approximately $\pm 50\%$ of ground truth converge quickly, in about 7 seconds. Larger initial errors tend to take much more time, generally longer than the 15 seconds afforded in simulation, to begin to approach an improved observation model estimate. This is likely due to a combination of sub-optimal gradient-descent parameter

tuning and the general non-convexity of even Gaussian model correction based on ISE_h minimization. Nevertheless, the efficacy of the proposed model correction approach is clearly demonstrated by the plots.

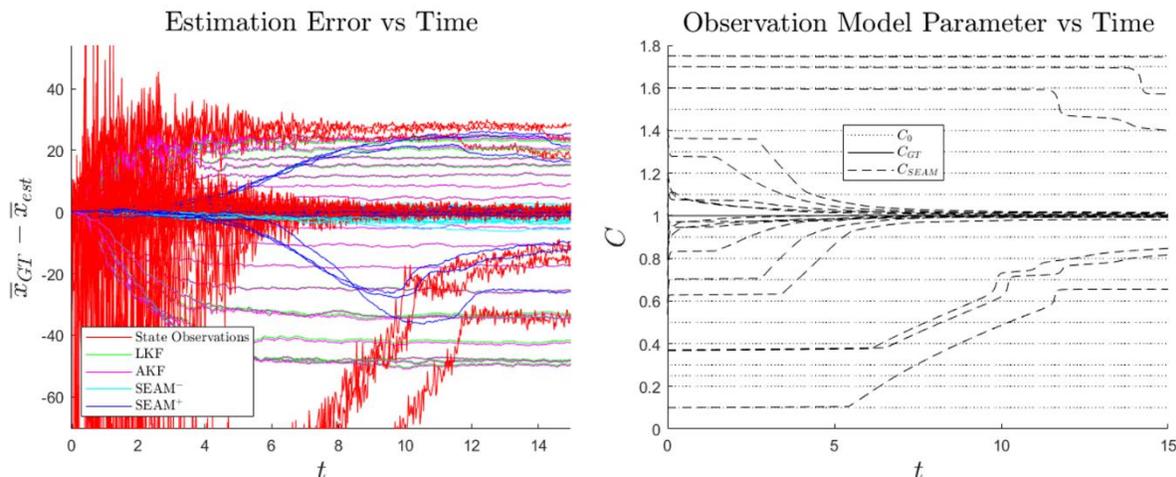


Figure 7. Ensemble curves demonstrating the convergence of observation model parameters for several initial estimates of C . Note the failure of six simulations to converge C in both plots due to particularly poor initial estimates.

6.4.1.2 Multi-Gaussian Belief

In order to avoid potential problems of severe non-convexity in ISE_h minimization for multi-Gaussian validation, δ_j and $c_j^{(k|\hat{k})}$ were chosen so that $p(z_k|\hat{x}_k)$ was near-Gaussian for the majority of these validative tests. Figure 8 shows the PDFs corresponding to the first time step of two different simulated experiments. The first employs a more strongly non-Gaussian observation

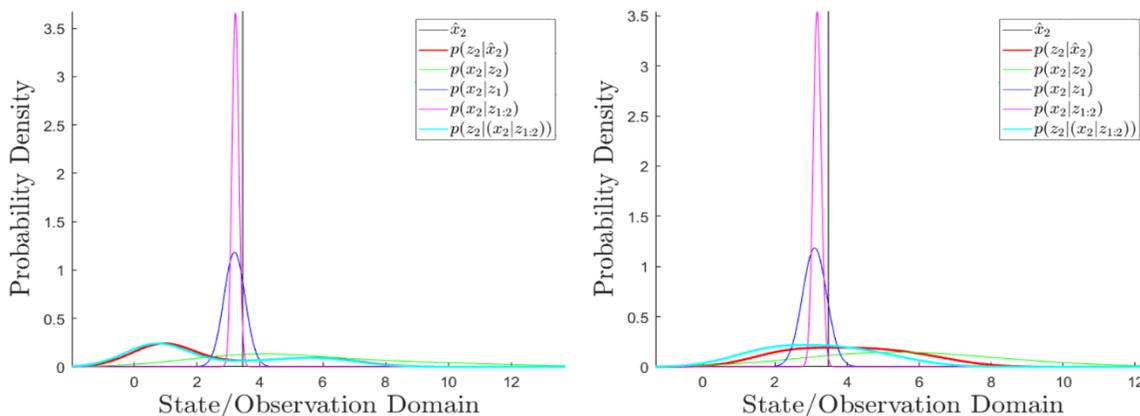


Figure 8. PDFs of different stages of belief for one time step. The first plot models multi-Gaussian observations using $c_j^{(k|\hat{k})} = [0.6, 0.1, 0.3]$ and $\delta_j = [-2.5, 0, 2.5]$, while the second plot uses using $c_j^{(k|\hat{k})} = [1/3, 1/3, 1/3]$ and $\delta_j = [-1.5, 0, 1.5]$. Both use $C_{GT} \sim \mathcal{N}(1, 0.1^2)$ in this example.

model, while model used for validation in this section. In simulation, a generated observation \hat{z}_k is drawn from its multi-Gaussian distribution by using its inverse cumulative distribution function (CDF). The estimated uncertainty in that observation is characterized by covariances $\Sigma_{z_j}^{(k|\hat{k})}$, given by Eq. (25) and implemented according to Eq. (27). The means of the Gaussian components are approximated by Eq. (26). Obtaining the weighting coefficients $c_j^{(k|\hat{k})}$ is trivial, as these come directly from off-line model characterization.

The nature of non-Gaussian belief is such that the estimated state cannot be adequately summarized by a single state vector. For this reason, state-time and error-time plots are unobtainable in such problems. Also, non-Gaussian estimation cannot be compared with KF approaches because the latter require Gaussian belief. However, the primary aim of this paper is to prove the efficacy of observation model parameter correction; comparing estimation performance is only secondary. For several Monte-Carlo simulated non-Gaussian experiments, ensemble curves were generated to show the convergence of the 1-D observation model parameter C toward ground-truth, as in the second plot of Fig. 7. Again, the constants of Table 1 were used,

with time spanning from 0 to 15 seconds, $\Sigma_v = 1$, $M = k$, and $\lambda = 0.98$. These curves are shown in Fig. 9 below.

In similar fashion to the Gaussian example of Fig. 7, convergence occurs quickly for relatively low initial parameter estimate error, but takes more

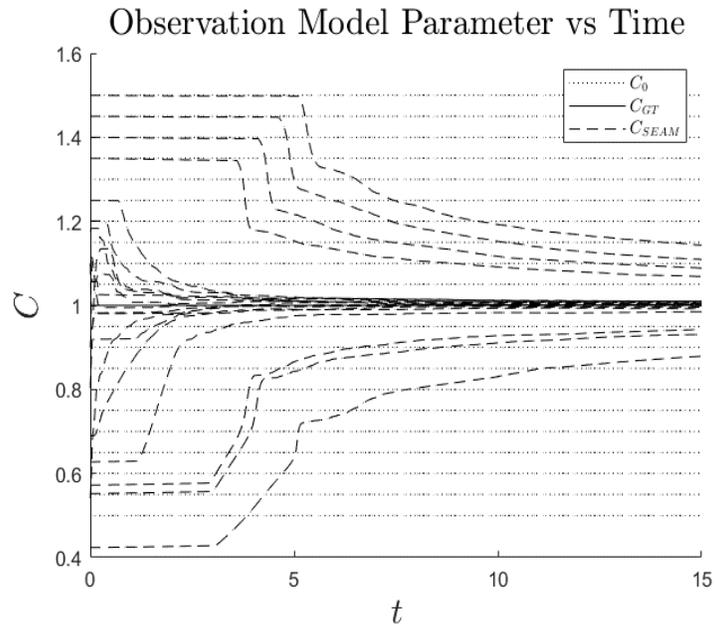


Figure 9. Non-Gaussian observation model correction over time.

time for more severe initial error. In fact, non-Gaussian convergence appears to be, on average, more efficient. This is likely due only to improved parameter tuning (i.e. λ was decreased to 0.98 which can have a substantial effect over 600 time steps). This implies that improved performance could be achieved in general with more attention to parameter tuning.

In order to test the limits of the proposed method of non-Gaussian observation model correction on a 1-D system, the ability of the framework to improve C was assessed for combinations of two extreme cases: highly non-Gaussian PDFs and highly inaccurate initial estimates C_0 . While not fully quantitative due to the vastness of the potential parameter space over which model correction might be assessed, this study gives valuable insight into the sensitivities of the framework. Figure 10 summarizes these findings. Model correction was impressively capable of handling even a tri-modal observation PDF $p(z_k|\hat{x}_k)$ when ε was low, though

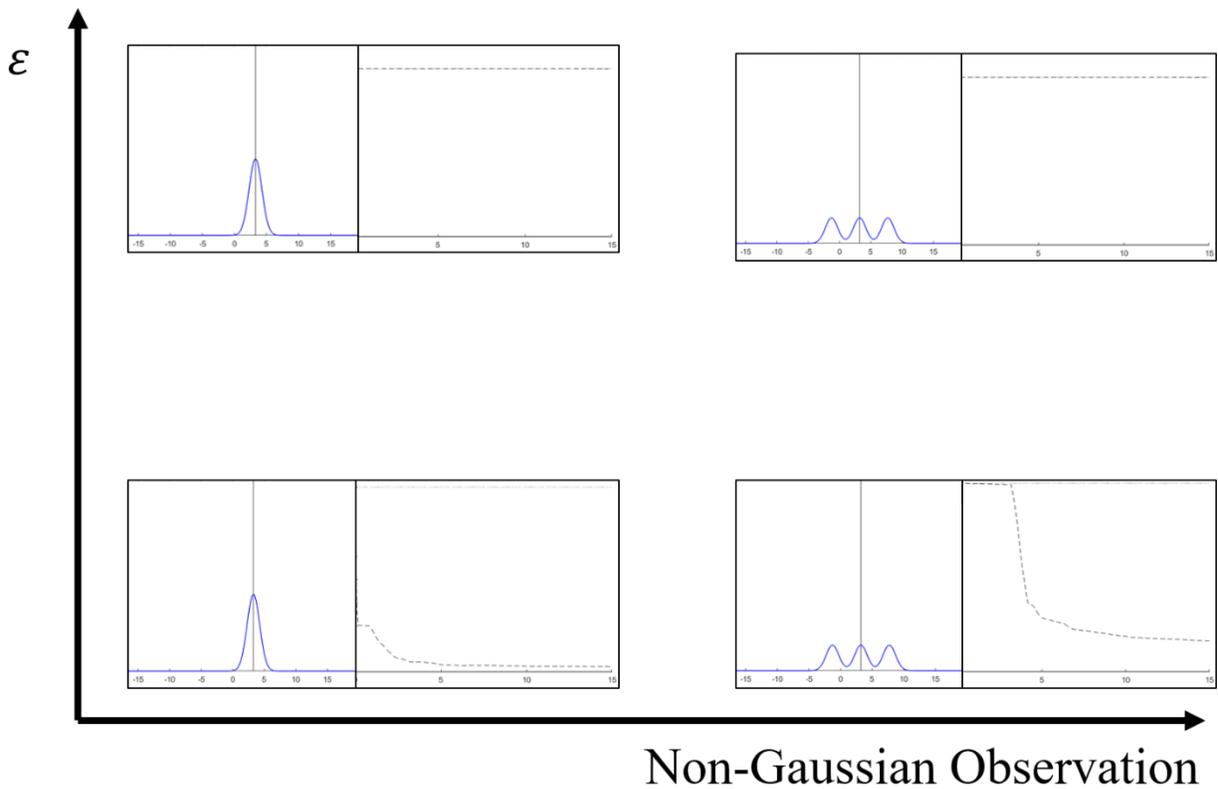


Figure 10. Dependency of model correction on model inaccuracy and non-Gaussian severity.

improvements in the estimate of C were slow and did not completely converge in the simulation time allotted. However, neither Gaussian nor non-Gaussian model correction succeeded for high ε . As is generally true with non-convex gradient-based optimization, success is highly dependent on a good initial parameter estimate.

6.4.2 2-D Validation

In order to study the multi-dimensionality of the proposed framework, the four aforementioned estimators were applied in two dimensions. The simulation consisted of a linear mass-spring-damper (MSD) system which can be represented as a first-order 2-D ordinary differential equation (ODE). For this arbitrary system, the continuous-time motion model matrices \mathbf{A} and \mathbf{B} were assigned the following parameter values:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{k}{m} & \frac{b}{m} \end{bmatrix}, \quad (49)$$

where k is a stiffness coefficient, b is a damping coefficient, and m is a mass. These linear motion model matrices can then be converted to discrete-time by use of the matrix exponential.

The unforced response of the system to initial conditions \mathbf{x}_0 was observed and estimated for 100 different trials where the ground-truth observation model matrix \mathbf{C}_{CT} was randomly generated according to $\mathcal{N}(\mathbf{C}_0, \mathbf{\Sigma}_{C_0})$. Again, \mathbf{C}_0 is the initial model estimate with element-wise standard deviations given by $\mathbf{\Sigma}_{C_0}$. For the sake of simplicity, a Gaussian observation model was used here. Table 3 gives all arbitrarily chosen parameter values used in these experiments.

Results of a sample simulation are given in Figs. 11 and 12. As Fig. 11 demonstrates, the error in the two SEAM estimator' signals is low compared to unfiltered observations and Kalman-filtered signals. Specifically, SEAM with observation model correction outperforms all other estimators in this simulation simply because model parameter estimates are improved over time.

This is also particularly evident in the first plot of Fig. 12. Though \mathbf{C}_{SEAM} approaches \mathbf{C}_{GT} more slowly and stochastically than in the 1-D examples, likely due to the fourfold increase in degrees of freedom, there is clear improvement by the end of the ten-second simulation. This is also reflected in the average reduction in ISE_h over time, as shown in the second plot.

Table 3. Constant parameters

k	b	m	λ	M	Δt	d_i	T
10 N/m	5 Ns/m	2 kg	0.99	50	0.03s	0.05	$5 \cdot 10^{-6}$
\mathbf{x}_0	$\Sigma_{\mathbf{x}_0}$	\mathbf{C}_0	$\Sigma_{\mathbf{C}_0}$				
$\begin{bmatrix} 10 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1^2 & 0^2 \\ 0^2 & 0.3^2 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$				
Σ_w		Σ_v					
$\begin{bmatrix} 0.01^2 & 0.002^2 \\ 0.002^2 & 0.015^2 \end{bmatrix}$		$0.1 \cdot \begin{bmatrix} 0.6^2 & 0.1^2 \\ 0.1^2 & 0.57^2 \end{bmatrix}$					

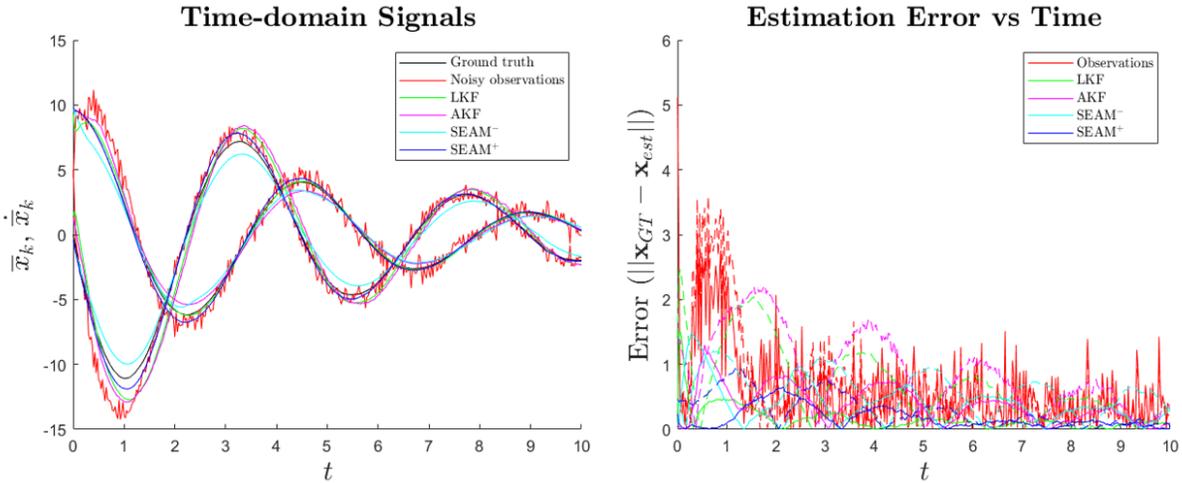


Figure 11. 2-D time domain signal and error plots.

For the 100 randomly generated simulations, the median percent error at $t = 0$ and $t = 10$ ($k = 334$) in the elements of \mathbf{C}_{SEAM} was recorded, as well as the percent of trials for which there was improvement by the end of simulation. These findings are recorded in Table 4 below. In

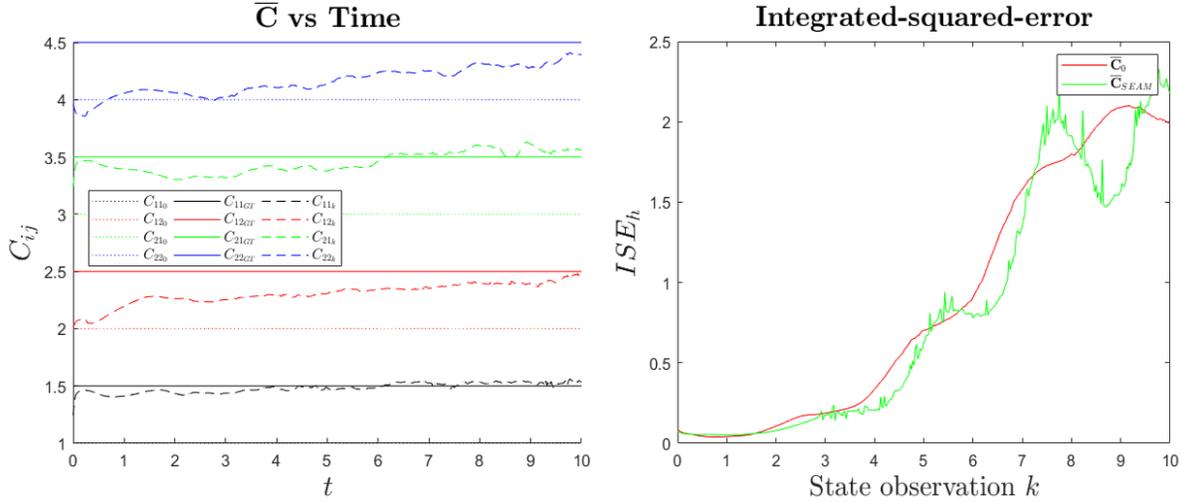


Figure 12. 2-D observation model matrix and ISE_h over time.

addition, the median RMSE scores and percentage of wins corresponding to each of the four estimators are reported in Table 5.

Table 4. Percent errors in observation model parameters for 100 trials

	$\%C_{11}$	$\%C_{12}$	$\%C_{21}$	$\%C_{22}$
$k = 0$	32.38	16.21	11.64	7.36
$k = 334$	21.10	12.38	16.30	10.88
% Improvement	61	57	38	39

As evidenced by Table 4, model correction tends to improve parameter estimates more reliably when initial error is not too small. Parameters C_{11} and C_{12} were often improved due to their substantial average initial inaccuracy, while improvement in C_{21} and C_{22} was less reliable because those parameters were not as inaccurate on average. This supports an intuitive understanding of gradient-descent error minimization, in that gradient sensitivity is lower with respect to variables with less error than those with greater error. Furthermore, in reference to Table 5, it is apparent that SEAM-plus-model-correction handily outperforms traditional estimators. Though model correction may not always improve a model estimate towards ground-truth in a

timely fashion, this evidence suggests that model correction does reliably yield a set of parameters that reduces estimation error, satisfying the overall objective of SEAM.

The above observations are further supported by the plots in Fig. 13 below. As the first plot shows, there is a clear proportional relationship between the error in \mathbf{C} at the end of each trial and the RMSE estimation error. This is the motivating factor in model correction and is, of course, to be expected. In the second plot, note that the majority of points lie above the x-axis, indicating that estimation was improved for the majority of trials by implementing model correction. Furthermore, because most of these points lie in the upper-right quadrant, it is clear that estimation improves most often when the percent-error in \mathbf{C} reduces.

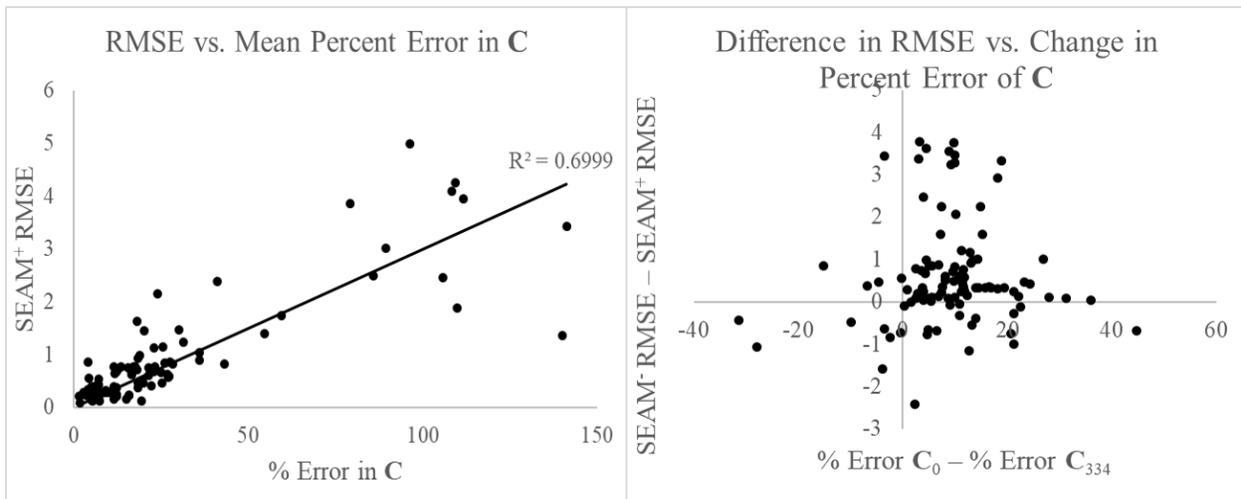


Figure 13. Error plots representing data acquired over 100 stochastic 2-D simulations.

6.5 Conclusions and Future Work

This paper has developed a framework for simultaneous estimation and observation model correction, allowing for non-Gaussian belief in multiple dimensions. The approach demonstrates effectiveness in the various proof-of-concept simulations detailed above, consisting of Gaussian and non-Gaussian trials in 1 and 2 dimensions. The proposed technique formulates observational uncertainty by incorporating uncertainties in model parameters, in addition to recursively updating those parameter estimates. According to the results obtained through Monte Carlo simulations,

even just accounting for increased uncertainty resulted in improved state estimation as compared to more traditional Kalman-based approaches. By incorporating model correction, estimation was improved even further, in keeping with the hypothesized advantages of the proposed framework.

Though this work shows promise, current limitations warrant future efforts to refine and develop the efficiency, accuracy, and ubiquity of the framework. The linear observation model considered here, though common in real-world problems, does not apply to all potentially useful contexts. As such, derivations for belief propagation under generally nonlinear observation models are warranted. Even for a linear observation model, only a square and invertible \mathbf{C} matrix was addressed here; inverse observation for non-square observation matrices has yet to be formulated under this framework. Furthermore, the implications of state observability are not yet well known with regards to this work. Finally, major improvements can be made in gradient-descent algorithms for fast ISE_h minimization as presented in this paper. The proof-of-concept simulations carried out in the results section employed the most basic form of linear gradient-descent to minimize state re-observation error. More advanced methods hold promise for improved efficiency in the model correction stage. SDG

6.6 References

- [1] J. Vetelino, A. Reghu, "Introduction to Sensors," Boca Raton, FL, USA: CRC Press, pp. 1, 2011.
- [2] V. Pathak, V. Kumar, D. N. Sandeep, R. K. Barik, "Tunable Multilevel Probabilistic Sensing Model based Intrusion Detection in Gaussian Distributed WSNs," 14th IEEE India Council International Conference (INDICON), Roorkee, India, 2017.

- [3] Y. Chen, C. Chen, "Sensor Deployment under Probabilistic Sensing Model," Proceedings of the 2nd High Performance Computing and Cluster Technologies Conference, Beijing, China, pp. 33-36, 2018.
- [4] C. K. Chui, G. Chen, "Kalman Filtering with Real-Time Applications: Fifth Edition," Cham, Switzerland: Springer International Publishing AG, pp. 19, 2017.
- [5] T. Moore, D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," Intelligent Autonomous Systems, vol. 13, pp. 335-348, 2015.
- [6] A. Olivier, A. W. Smyth, "A marginalized unscented Kalman filter for efficient parameter estimation with applications to finite element models," Computer Methods in Applied Mechanics and Engineering, vol. 339, pp. 615-643, 2018.
- [7] X. Cui, Z. He, E. Li, A. Cheng, M. Luo, Y. Guo, "State-of-charge estimation of power lithium-ion batteries based on an embedded micro control unit using a square root cubature Kalman filter at various ambient temperatures," International Journal of Energy Research, vol. 43, no. 8, pp. 3561-3577, 2019.
- [8] J. R. Stroud, M. Katzfuss, C. K. Wikle, "A Bayesian Adaptive Ensemble Kalman Filter for Sequential State and Parameter Estimation," Monthly Weather Review, vol. 146, pp. 373-386, 2018.
- [9] A. Katriniok, D. Abel, "Adaptive EKF-Based Vehicle State Estimation With Online Assessment of Local Observability," IEEE Transactions on Control Systems Technology, vol. 24, no. 4, pp. 1368-1381, 2016.
- [10] H. W. Sorenson, D. L. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," Automatica, vol. 7, pp. 465-479, 1971.

- [11] J. J. Steckenrider, T. Furukawa, "A Probabilistic Model-Adaptive Approach for Tracking of Motion with Heightened Uncertainty," *International Journal of Control, Automation, and Systems*, *Pre-print*, 2019.
- [12] C. Kalender, A. Schöttl, "Sparse Grid-Based Nonlinear Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2386-2396, 2013.
- [13] M. Pulido, P. J. Leeuwen, "Sequential Monte Carlo with kernel embedded mappings: The mapping particle filter," *Journal of Computational Physics*, vol. 396, pp. 400-415, 2019.
- [14] D. Raihan, S. Chakravorty, "Particle Gaussian Mixture Filters: Application and Performance Evaluation," *22nd International Conference on Information Fusion*, Ottawa, Ontario, CA, 2019.
- [15] D. L. Alspach, "Gaussian Sum Approximations in Nonlinear Filtering and Control," *Information Sciences*, vol. 7, pp. 271-290, 1974.
- [16] J. J. Steckenrider, T. Furukawa, "Probabilistic Motion Model Correction Using Non-Gaussian Belief Fusion," *Automatica*, *Under revision*, 2019.
- [17] K. Whitehouse, D. Culler, "Calibration as parameter estimation in sensor networks," *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, Georgia, USA, pp. 59-67, 2002.
- [18] M. Liu, P. Shi, "Sensor fault estimation and tolerant control for Itô stochastic systems with a descriptor sliding mode approach," vol. 49, no. 5, pp. 1242-1250, 2013.
- [19] M. Liu, X. Cao, P. Shi, "Fault Estimation and Tolerant Control for Fuzzy Stochastic Systems," vol. 21, no. 2, pp. 221-229, 2013.
- [20] T. Youssef, M. Chadli, H. R. Karimi, R. Wang, "Actuator and sensor faults estimation based on proportional integral observer for TS fuzzy model," vol. 354, no. 6, pp. 2524-2542, 2017.

- [21] A. Sani, A. Vosoughi, "On Distributed Linear Estimation With Observation Model Uncertainties," vol. 66, no. 12, pp. 3212-3227, 2018.
- [22] A. Almagbile, J. Wang, W. Ding, "Evaluating the Performances of Adaptive Kalman Filter Methods in GPS/INS Integration," *Journal of Global Positioning Systems*, vol. 9, no. 1, pp. 33-40, 2010.
- [23] J. J. Steckenrider, T. Furukawa, "Multi-Dimensional Belief Fusion of Multi-Gaussian Structures," *Information Fusion*, vol. 57, pp. 71-88, 2020.
- [24] M. Lefebvre, R.K. Keeler, R. Sobie, J. White, "Propagation of Errors for Matrix Inversion," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 451, no. 2, pp. 520-528, 2000.

CHAPTER 7. Generalized Simultaneous Estimation and Modeling

7.1 Introduction

7.1.1 Background

In real-world science and engineering systems, nothing can be known with complete certainty. Even for highly deterministic processes, randomness is injected by compounding factors that cannot be isolated from a system of interest. Furthermore, our ability to measure a system's states is impaired by the intrinsic limitations of our physical sensors. State estimation techniques have emerged in recent decades with the goal of overcoming these fundamental barriers. The abilities of probabilistic methods to accomplish state estimation are ever-growing, yet comprehensive frameworks for solving some of the most complex problems are still elusive.

7.1.2 Related Work

One of the most general probabilistic estimation frameworks is known as recursive Bayesian estimation (RBE), or often just Bayesian estimation [1]. RBE leverages predictions and observations of a system's state to provide a corrected probabilistic estimate of that state. State estimates are propagated through RBE by probability distribution functions (PDFs) which are used to mathematically represent what is termed here *state belief*. Predictions rely on a mathematical model of the system at hand [2], while observations come from some sensor or observer [3] which can also be modeled mathematically [4]. These models are usually derived from governing differential equations, which are commonly nonlinear [5]. Oftentimes, these models or the parameters composing them are not well known [6]. Furthermore, state belief can sometimes be highly non-Gaussian, and even multi-modal [7]. The following review of literature addresses related works which have investigated each of these complexities in RBE.

State belief is commonly represented by the Gaussian, or normal, distribution [8]. Under the Gaussian assumption, a family of popular estimators arises, known as Kalman filters (KFs). The original linear Kalman filter (LKF) was first proposed in 1960 by R. E. Kalman [9]. For nonlinear problems, other variants began to arise, such as the extended KF (EKF) [10], unscented KF (UKF) [11], and the square-root cubature Kalman filter (SRCKF) [12]. Other versions including the ensemble KF (EnKF) [13] and Schmidt-Kalman filter [14] have been proposed to handle increased uncertainty. While these filters successfully handle estimation in their designed contexts, most rely on Gaussian assumptions. Furthermore, there is a general assumption that predictive model parameters are known with complete confidence, an assumption which may not always be correct.

Non-Gaussian RBE is primarily accomplished by sampling techniques. The first of such approaches is the well-known sequential Monte-Carlo, or particle, filter [15], [16]. This non-Gaussian approximator relies on belief propagation through randomly generated weighted particles. A similar approach, often known as the grid-based filter [17], implements the equations of RBE for grid cells in a discretized belief space. Another approach employs a bootstrap filter to sample a non-Gaussian PDF and propagate nonlinear belief according to the rules of RBE [18]. Finally, the multi-Gaussian approach proposed by Alspach, et al. [8] and expanded upon by Steckenrider, et al. [7] approximates non-Gaussian PDFs with weighted Gaussian sums and applies the equations of RBE to each Gaussian. This technique both reduces computational demand and increases accuracy, allowing for lossless non-Gaussian belief propagation. A similar approach proposed for handling non-Gaussian outliers in nonlinear estimation was proposed by Stojanovic [19], but this framework is only shown to handle uni-modal belief and does not address non-Gaussian prediction.

Because RBE is highly dependent on the accuracy of the equations and parameters used to model a system, system identification is a critical component of state estimation. This is usually done offline, either theoretically from first principles [20], or experimentally using frequency-response methods [21] or the like. Some techniques are also capable of estimation when there is uncertainty in model parameters [22]; the adaptive extended KF (AEKF) [23], [24] was developed to address ill-defined uncertainties in nonlinear contexts. However, only a few methods [25], [26] actually incorporate parametric uncertainties in state estimation. Sensitivity methods have proven capable of online system identification [27], [28], [29], but these frameworks usually aim to characterize finite-element models by leveraging summary characteristics like mode shapes and natural frequencies. Adaptive parameter estimation is accomplished by Oh, et al. [30], though this work relies on linear and Gaussian assumptions. Algorithms proposed by Liu [5] and Pan [31] are capable of controlling systems with uncertainty in nonlinear models, but these frameworks are not estimation-oriented, as they do not incorporate sensor or process noise. Other recent achievements have been made in both functional and parametric real-time model estimation [32], [33], but such approaches do not incorporate non-Gaussian belief. Similar techniques have been developed for joint state and parameter estimation [6], allowing for lightly non-Gaussian sensor noise. However, these methods do not implicitly extract model parameter estimates from state observations alone. This is accomplished by a non-Gaussian variant [7] of the Simultaneous Estimation and Modeling (SEAM) framework proposed by Steckenrider, et al. [34] which accounts for parametric uncertainty but does not allow for nonlinear motion models. In general, most existing methods assume a system is linear, state belief is Gaussian, motion models are accurate, or some combination of these.

7.1.3 Objectives and Outline

In light of the limitations of established work, this paper presents a generalization of non-Gaussian SEAM for nonlinear state-space problems. The two primary original contributions of the proposed framework include: 1) formulations for nonlinear estimation which incorporate model uncertainty, and 2) a nonlinear model correction stage. The paper is organized as follows: section 7.2 reviews fundamental concepts essential to the original contributions, which are developed in section 7.3. Results are given in section 7.4, and section 7.5 presents some conclusions and future work related to these efforts.

7.2 Estimation and Motion Model Correction

7.2.1 Recursive Bayesian Estimation

RBE consists of prediction, observation, and correction stages. In the following developments, the term *belief* is used to describe the PDF of a stochastic state vector $x \in \mathcal{X}$. The formulations of RBE make no assumptions about the forms of PDFs describing state belief.

7.2.1.1 Prediction

The prediction stage propagates state belief from step $k - 1$ to step k according to a motion model describing the underlying system whose state is being estimated. Let $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ be the *a priori* PDF describing belief about the state given all prior observations $\mathbf{z}_{1:k-1}$. Furthermore, let $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ be a PDF which governs the transition of the state from one step to the next. The continuous Chapman-Kolmogorov equation gives $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$, the *predicted* PDF, by:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int_{\mathcal{X}} p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}_{k-1})d\mathbf{x}_{k-1}. \quad (1)$$

Note that, for Markovian processes with independent increments, $p(\mathbf{x}_k|\mathbf{x}_{k-1}) = p(\mathbf{x}_k - \mathbf{x}_{k-1})$, and Eq. (1) becomes a convolution integral.

7.2.1.2 Observation

The observation stage is responsible for deriving the *observed* PDF, $p(\mathbf{x}_k|\mathbf{z}_k)$, given an observation or measurement \mathbf{z}_k . Such an observation often comes from a physical sensor which may be characterized by some model with certain intrinsic parameters. Obtaining $p(\mathbf{x}_k|\mathbf{z}_k)$, then, requires some context-specific manipulation of such an observation model.

7.2.1.3 Correction

The correction stage fuses predicted and observed belief, resulting in a PDF with less uncertainty than either $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ or $p(\mathbf{x}_k|\mathbf{z}_k)$. The *corrected*, or *a posteriori*, PDF $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is given by

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_k|\mathbf{z}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int_{\mathcal{X}} p(\mathbf{x}_k|\mathbf{z}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k}. \quad (2)$$

The most common form of a PDF composing belief in RBE is the Gaussian distribution. A multivariate Gaussian PDF is defined by only two parameters, a mean vector $\bar{\mathbf{x}}$ and a covariance matrix $\Sigma_{\mathbf{x}}$:

$$\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \Sigma_{\mathbf{x}}) = \frac{1}{\sqrt{|2\pi\Sigma_{\mathbf{x}}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x} - \bar{\mathbf{x}})\right), \quad (3)$$

It is well known that the product of two Gaussians yields a third unnormalized Gaussian. Therefore, if the predicted and observed PDFs are given by weighted Gaussian sums (or *multi-Gaussian* distributions) of the following form:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \sum_{i=1}^I c_i^{(k|1:k-1)} \mathcal{N}\left(\mathbf{x}; \bar{\mathbf{x}}_i^{(k|1:k-1)}, \Sigma_{\mathbf{x}_i}^{(k|1:k-1)}\right), \quad (4a)$$

$$p(\mathbf{x}_k|\mathbf{z}_k) = \sum_{j=1}^J c_j^{(k|k)} \mathcal{N}\left(\mathbf{x}; \bar{\mathbf{x}}_j^{(k|k)}, \Sigma_{\mathbf{x}_j}^{(k|k)}\right), \quad (4b)$$

the corrected PDF will also be multi-Gaussian:

$$\begin{aligned}
p(\mathbf{x}_k | \mathbf{z}_{1:k}) &= \sum_{i=1}^I \sum_{j=1}^J c_i^{(k|1:k-1)} c_j^{(k|k)} c_{ij}^{(k|1:k-1,k)} \mathcal{N} \left(\mathbf{x}; \bar{\mathbf{x}}_{ij}^{(k|1:k-1,k)}, \boldsymbol{\Sigma}_{x_{ij}}^{(k|1:k-1,k)} \right) \\
&= \sum_{n=1}^{IJ} c_n^{(k|1:k)} \mathcal{N} \left(\mathbf{x}; \bar{\mathbf{x}}_n^{(k|1:k)}, \boldsymbol{\Sigma}_{x_n}^{(k|1:k)} \right), \tag{5}
\end{aligned}$$

where

$$c_{ij}^{(k|1:k-1,k)} = \mathcal{N} \left(\bar{\mathbf{x}}_i^{(k|1:k-1)}; \bar{\mathbf{x}}_j^{(k|k)}, \boldsymbol{\Sigma}_{x_i}^{(k|1:k-1)} + \boldsymbol{\Sigma}_{x_j}^{(k|k)} \right), \tag{6a}$$

$$\boldsymbol{\Sigma}_{x_{ij}}^{(k|1:k-1,k)} = \left[\left(\boldsymbol{\Sigma}_{x_i}^{(k|1:k-1)} \right)^{-1} + \left(\boldsymbol{\Sigma}_{x_j}^{(k|k)} \right)^{-1} \right]^{-1}, \tag{6b}$$

$$\bar{\mathbf{x}}_{ij}^{(k|1:k-1,k)} = \boldsymbol{\Sigma}_{x_{ij}}^{(k|1:k-1,k)} \left[\left(\boldsymbol{\Sigma}_{x_i}^{(k|1:k-1)} \right)^{-1} \bar{\mathbf{x}}_i^{(k|1:k-1)} + \left(\boldsymbol{\Sigma}_{x_j}^{(k|k)} \right)^{-1} \bar{\mathbf{x}}_j^{(k|k)} \right]. \tag{6c}$$

7.2.2 Kalman Estimation

Kalman filters take advantage of the properties of Gaussian PDFs by propagating only the mean vectors and covariance matrices through the stages of RBE.

7.2.2.1 The Extended Kalman Filter

Let a system be represented by a generally nonlinear state-space motion model:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}. \tag{7}$$

where \mathbf{w}_{k-1} is a zero-mean Gaussian random vector with covariance $\boldsymbol{\Sigma}_{w_{k-1}}$ and \mathbf{u}_{k-1} is an input.

The mean and covariance are given in prediction by the EKF as:

$$\bar{\mathbf{x}}_{k|1:k-1} = f(\bar{\mathbf{x}}_{k-1|1:k-1}, \mathbf{u}_{k-1}), \tag{8a}$$

$$\boldsymbol{\Sigma}_{x_{k|1:k-1}} = \mathbf{J}_f \boldsymbol{\Sigma}_{x_{k-1|1:k-1}} \mathbf{J}_f^T + \boldsymbol{\Sigma}_{w_{k-1}}, \tag{8b}$$

where \mathbf{J}_f is the following Jacobian matrix:

$$\mathbf{J}_f = \left. \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})}{\partial \mathbf{x}_{k-1}} \right|_{\bar{\mathbf{x}}_{k-1|1:k-1}, \mathbf{u}_{k-1}}. \tag{9}$$

Let the sensor or observer be modeled according to the following generally nonlinear equation:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (10)$$

where \mathbf{v}_k is a zero-mean Gaussian random vector with covariance Σ_{v_k} . Correction is given by the following formulas:

$$\bar{\mathbf{x}}_{k|1:k} = \bar{\mathbf{x}}_{k|1:k-1} + \mathbf{K}_k \left(\mathbf{z}_k - h(\bar{\mathbf{x}}_{k|1:k-1}) \right), \quad (11a)$$

$$\Sigma_{x_{k|1:k}} = (I - \mathbf{K}_k \mathbf{J}_h) \Sigma_{x_{k|1:k-1}}, \quad (11b)$$

where the Kalman gain \mathbf{K}_k is given by:

$$\mathbf{K}_k = \Sigma_{x_{k|1:k-1}} \mathbf{J}_h^T \left(\mathbf{J}_h \Sigma_{x_{k|1:k-1}} \mathbf{J}_h^T + \Sigma_{v_k} \right)^{-1}, \quad (12)$$

and

$$\mathbf{J}_h = \left. \frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{\bar{\mathbf{x}}_{k|1:k-1}}. \quad (13)$$

As these equations demonstrate, though the EKF may effectively reduce sensitivity to sensor and system (process) noise by incorporating their covariances, it does not account for uncertainties in intrinsic model parameters. If such parameters are in reality random variables but are treated as well-defined constants, state estimation will suffer.

7.2.2.2 The Adaptive Extended Kalman Filter

The adaptive extended Kalman filter (AEKF) aims to improve on the EKF in cases where uncertainty in prediction and observation is underestimated. This is accomplished by adapting the covariances Σ_{w_k} and Σ_{v_k} over the course of estimation. One common formulation for adapting these matrix parameters is given as follows:

$$\Sigma_{v_k} = \mathbf{G}_v - \mathbf{J}_h \Sigma_{x_{k|1:k-1}} \mathbf{J}_h^T, \quad (14a)$$

$$\mathbf{G}_v = \frac{1}{M} \sum_{m=1}^M \|\bar{\mathbf{z}}_{k-m} - h(\bar{\mathbf{x}}_{k-m|1:k-m-1})\|^2, \quad (14b)$$

$$\boldsymbol{\Sigma}_{w_k} = \sqrt{\alpha} \boldsymbol{\Sigma}_{w_{k-1}}, \quad (14c)$$

$$\alpha = \frac{\text{tr}(\mathbf{G}_v - \boldsymbol{\Sigma}_{v_{k-1}})}{\text{tr}(\mathbf{J}_h \boldsymbol{\Sigma}_{x_{k|1:k-1}} \mathbf{J}_h^T)}. \quad (14d)$$

Propagation of the state mean and covariance still follows Eqs. (8) and (11). Though the AEKF captures potentially increased uncertainty in estimation, it does not specifically address the uncertainty coming from a potentially inadequate estimate of model parameters. Consequently, any model mismatch will cause non-zero-mean estimation error.

7.2.3 Linear Motion Model Correction

In order to rectify the problem of model mismatch in estimation, SEAM both incorporates parameter uncertainty and corrects parameter estimates during the course of estimation. Under a generalized SEAM framework, non-Gaussian PDFs are approximated by multi-Gaussian distributions. Belief is then propagated by applying the stages of RBE to channels of mean vectors, covariance matrices, and weighting coefficients.

A linear system is generally represented by the following state-space motion and observation models:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad (15a)$$

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k. \quad (15b)$$

The mean in prediction is given by the means of all involved variables as shown in Eq. (16a) below. Uncertainties in the motion model parameters \mathbf{A} and \mathbf{B} and \mathbf{x}_{k-1} , \mathbf{u}_{k-1} , and \mathbf{w}_{k-1} are incorporated in prediction by the covariance formula of Eq. (16b) below.

$$\bar{\mathbf{x}}_{k|1:k-1} = \bar{\mathbf{A}}_{k-1} \bar{\mathbf{x}}_{k-1|1:k-1} + \bar{\mathbf{B}}_{k-1} \bar{\mathbf{u}}_{k-1}, \quad (16a)$$

$$\begin{aligned}
\boldsymbol{\Sigma}_{x_k|1:k-1} &= \boldsymbol{\Sigma}_{w_{k-1}} + \bar{\mathbf{A}}_{k-1} \boldsymbol{\Sigma}_{x_{k-1}|1:k-1} \bar{\mathbf{A}}_{k-1}^T + \bar{\mathbf{B}}_{k-1} \boldsymbol{\Sigma}_{u_{k-1}} \bar{\mathbf{B}}_{k-1}^T \\
&+ \boldsymbol{\Sigma}_{A_{k-1}} \mathcal{D}(\bar{\mathbf{x}}_{k-1|1:k-1}^{\circ 2}) \boldsymbol{\Sigma}_{A_{k-1}}^T + \boldsymbol{\Sigma}_{B_{k-1}} \mathcal{D}(\bar{\mathbf{u}}_{k-1}^{\circ 2}) \boldsymbol{\Sigma}_{B_{k-1}}^T.
\end{aligned} \tag{16b}$$

Here, $\boldsymbol{\Sigma}_{A_{k-1}}$ and $\boldsymbol{\Sigma}_{B_{k-1}}$ are matrices containing the *standard deviations* of the corresponding elements in the mean matrices $\bar{\mathbf{A}}_{k-1}$ and $\bar{\mathbf{B}}_{k-1}$, the operation $\mathcal{D}(\bullet)$ diagonalizes its vector argument, and $(\bullet)^{\circ 2}$ denotes the element-wise multiplication of a vector with itself. Under a multi-Gaussian representation of belief, Eqs. (16) are applied to each Gaussian component in the distribution.

The motion model correction approach of [7] applies to non-Gaussian belief and addresses correction of the linear motion model parameters \mathbf{A} and \mathbf{B} . This is accomplished by minimizing an objective function, defined as the integrated-squared-error (ISE):

$$ISE = \int_{\mathbf{x}} (p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) - p(\mathbf{x}_k | \mathbf{z}_{1:k}))^2 d\mathbf{x}_k. \tag{17}$$

Under this approach, model correction becomes an optimization problem which is solved using gradient-descent. The gradient of the ISE is taken with respect to all involved motion model parameters and used to iteratively improve these parameters.

Because existing SEAM formulations apply only to linear estimation problems, a comprehensive nonlinear/non-Gaussian SEAM framework has not yet been established. The following section describes the original contributions of this work in generalizing SEAM to such contexts.

7.3 Nonlinear Estimation and Motion Model Correction

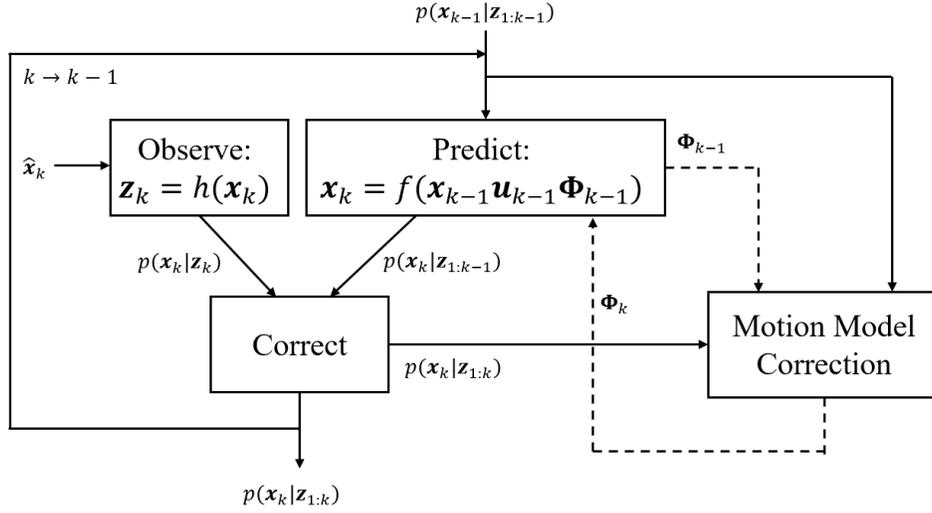


Figure 1. Nonlinear SEAM framework.

Figure 1 summarizes the SEAM framework as proposed for application to generally nonlinear problems. The original contributions detailed in this section include formulations for extended uncertainty propagation and nonlinear motion model correction.

7.3.1 Nonlinear Estimation with Extended Uncertainty Propagation

Let a nonlinear system be described by the following state-space model:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \Phi_{k-1}) + \mathbf{w}_{k-1}, \quad (18)$$

where Φ_{k-1} is a vector of motion model parameters determined by the system at hand. In contrast with Eq. (7), it is critical here to explicitly consider the functional dependence of a system on its model parameters. Because additive noise is assumed to be zero-mean, the mean in prediction can be given by:

$$\bar{\mathbf{x}}_{k|1:k-1} = f(\bar{\mathbf{x}}_{k-1|1:k-1}, \bar{\mathbf{u}}_{k-1}, \bar{\Phi}_{k-1}). \quad (19)$$

The covariance is then:

$$\Sigma_{\mathbf{x}_{k|1:k-1}} = \Sigma_f + \Sigma_{\mathbf{w}_{k-1}}. \quad (20)$$

By error propagation, the covariance matrix Σ_f corresponding to $f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \Phi_{k-1})$ requires three terms in order to address uncertainty in \mathbf{x}_{k-1} , \mathbf{u}_{k-1} , and Φ_{k-1} :

$$\Sigma_f = J_f^x \Sigma_{x_{k-1|1:k-1}} J_f^{xT} + J_f^u \Sigma_{u_{k-1}} J_f^{uT} + J_f^\Phi \Sigma_{\Phi_{k-1}} J_f^{\Phi T}, \quad (21)$$

where

$$J_f^x = \left. \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \Phi_{k-1})}{\partial \mathbf{x}_{k-1}} \right|_{\bar{\mathbf{x}}_{k-1|1:k-1}, \bar{\mathbf{u}}_{k-1}, \bar{\Phi}_{k-1}}, \quad (22a)$$

$$J_f^u = \left. \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \Phi_{k-1})}{\partial \mathbf{u}_{k-1}} \right|_{\bar{\mathbf{x}}_{k-1|1:k-1}, \bar{\mathbf{u}}_{k-1}, \bar{\Phi}_{k-1}}, \quad (22b)$$

$$J_f^\Phi = \left. \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \Phi_{k-1})}{\partial \Phi_{k-1}} \right|_{\bar{\mathbf{x}}_{k-1|1:k-1}, \bar{\mathbf{u}}_{k-1}, \bar{\Phi}_{k-1}}. \quad (22c)$$

It can be shown that Eq. (16b) is a simpler linear result of this formulation that assumes the covariance between individual model parameters is zero. Furthermore, Kalman filters assume $\Sigma_{u_{k-1}} = \mathbf{0}$ and $\Sigma_\Phi = \mathbf{0}$. In cases where there is some uncertainty in the control effort and/or model parameters, the second two terms of Eq. (21) are non-negligible and prediction under a Kalman filter becomes overconfident.

When belief is multi-Gaussian, the mean and covariance propagation of Eqs. (19)-(22) are applied to each Gaussian component in the *a priori* PDF. An observation PDF may also have a multi-Gaussian form, depending on specific sensor characteristics. Correction then consists of the fusion of the predicted and observed PDFs according to Eqs. (5) and (6) to yield a multi-Gaussian *a posteriori* PDF.

7.3.2 Nonlinear Motion Model Correction

In order to correct nonlinear motion model parameters Φ_k at step k , we seek to minimize an integrated-squared-error objective function:

$$\Phi_k = \underset{\Phi}{\operatorname{argmin}}(ISE). \quad (23)$$

Let the ISE be defined as in Eq. (17). This objective function is chosen over other potential functions (e.g. the Kullback-Leibler Divergence of $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ from $p(\mathbf{x}_k|\mathbf{z}_{1:k})$) because 1) it provides a symmetrical assessment of the similarity between the two PDFs, and 2) it requires only addition, subtraction, multiplication, and integration of PDFs. In order to handle non-Gaussian belief, let $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ be given by the general multi-Gaussian distributions of Eqs. (4a) and (5). The ISE then becomes:

$$ISE = \int_{\mathbf{x}} \left(\sum_{i=1}^I c_i^{(k|1:k-1)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_i^{(k|1:k-1)}, \Sigma_{x_i}^{(k|1:k-1)}) - \sum_{n=1}^{IJ} c_n^{(k|1:k)} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_n^{(k|1:k)}, \Sigma_{x_n}^{(k|1:k)}) \right)^2 d\mathbf{x}. \quad (24)$$

In the above expression, the variables $c_n^{(k|1:k)}$, $\bar{\mathbf{x}}_n^{(k|1:k)}$, and $\Sigma_{x_n}^{(k|1:k)}$ which define corrected belief are treated as constants obtained by the correction stage of RBE. This is because $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is a better estimate of \mathbf{x}_k which acts as a reference PDF to which $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ must be matched by adjusting Φ .

By 1) expanding the square, 2) leveraging the fact that the products of each Gaussian sum is another Gaussian sum, and 3) reducing integrals of Gaussians because $\int_{\mathcal{X}} \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \Sigma_x) d\mathbf{x} = 1$, the ISE becomes:

$$ISE = \sum_{i=1}^I \sum_{l=1}^I c_i^{(k|1:k-1)} c_l^{(k|1:k-1)} c_{il}^{(k|1:k-1,1:k-1)} - 2 \sum_{i=1}^I \sum_{n=1}^{IJ} c_i^{(k|1:k-1)} c_n^{(k|1:k)} c_{in}^{(k|1:k-1,1:k)} + \sum_{n=1}^{IJ} \sum_{m=1}^{IJ} c_n^{(k|1:k)} c_m^{(k|1:k)} c_{nm}^{(k|1:k,1:k)}. \quad (25)$$

The mixed weighting coefficients c_{il} , c_{in} , and c_{nm} are given as Gaussians by Eq. 6a). These coefficients arise from the product of $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ with itself, $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ with $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, and $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ with itself, respectively. Finding motion model parameters which minimize the ISE requires taking its gradient with respect to Φ :

$$\begin{aligned} \nabla_{\Phi}(ISE) &= \sum_{i=1}^I \sum_{l=1}^I c_i^{(k|1:k-1)} c_l^{(k|1:k-1)} \nabla_{\Phi} \left(c_{il}^{(k|1:k-1,1:k-1)} \right) \\ &\quad - 2 \sum_{i=1}^I \sum_{n=1}^{IJ} c_i^{(k|1:k-1)} c_n^{(k|1:k)} \nabla_{\Phi} \left(c_{in}^{(k|1:k-1,1:k)} \right). \end{aligned} \quad (26)$$

Because only $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ depends directly on the motion model via prediction, the coefficient c_{nm} is not functionally dependent on Φ and so the third term of the ISE does not contribute to the gradient.

As Eq. (26) shows, the gradient of the ISE contains many terms, each of which is a complex function of Φ . For this reason, a closed-form solution for the model parameters which minimize the ISE is intractable and a gradient-descent approach is instead considered. Because c_{il} and c_{in} are given by Gaussians of the form $\alpha_{il} \exp(\beta_{il})$ and $\epsilon_{in} \exp(\zeta_{in})$, respectively, their gradients are given by the chain rule as follows:

$$\nabla_{\Phi} \left(c_{il}^{(k|1:k-1,1:k-1)} \right) = \exp(\beta_{il}) \left(\nabla_{\Phi}(\alpha_{il}) + \alpha_{il} \nabla_{\Phi}(\beta_{il}) \right), \quad (27a)$$

$$\nabla_{\Phi} \left(c_{in}^{(k|1:k-1,1:k)} \right) = \exp(\zeta_{in}) \left(\nabla_{\Phi}(\epsilon_{in}) + \epsilon_{in} \nabla_{\Phi}(\zeta_{in}) \right). \quad (27b)$$

where

$$\alpha_{il} = (|2\pi\boldsymbol{\gamma}_{il}|)^{-\frac{1}{2}}, \quad \beta_{il} = -\frac{1}{2} \boldsymbol{\delta}_{il}^T \boldsymbol{\gamma}_{il}^{-1} \boldsymbol{\delta}_{il}, \quad (28a)$$

$$\epsilon_{in} = (|2\pi\boldsymbol{\eta}_{in}|)^{-\frac{1}{2}}, \quad \zeta_{in} = -\frac{1}{2} \boldsymbol{\theta}_{in}^T \boldsymbol{\eta}_{in}^{-1} \boldsymbol{\theta}_{in}, \quad (28b)$$

and

$$\boldsymbol{\gamma}_{il} = \boldsymbol{\Sigma}_{x_i}^{(k|1:k-1)} + \boldsymbol{\Sigma}_{x_l}^{(k|1:k-1)}, \quad (29a)$$

$$\boldsymbol{\delta}_{il} = \bar{\mathbf{x}}_i^{(k|1:k-1)} - \bar{\mathbf{x}}_l^{(k|1:k-1)}, \quad (29b)$$

$$\boldsymbol{\eta}_{in} = \boldsymbol{\Sigma}_{x_i}^{(k|1:k-1)} + \boldsymbol{\Sigma}_{x_n}^{(k|1:k)}, \quad (29c)$$

$$\boldsymbol{\theta}_{in} = \bar{\mathbf{x}}_i^{(k|1:k-1)} - \bar{\mathbf{x}}_n^{(k|1:k)}. \quad (29d)$$

In Eqs. (29), only variables with i and j subscripts are implicit functions of the motion model.

These functional relationships are given by prediction according to Eqs. (19)-(22).

The four gradients of Eqs. (27) are assembled from the partial derivatives with respect to each model parameter. For example,

$$\nabla_{\boldsymbol{\Phi}}(\alpha_{il}) = \left[\frac{\partial \alpha_{il}}{\partial \boldsymbol{\Phi}_{(1)}} \quad \frac{\partial \alpha_{il}}{\partial \boldsymbol{\Phi}_{(2)}} \quad \cdots \quad \frac{\partial \alpha_{il}}{\partial \boldsymbol{\Phi}_{(q)}} \quad \cdots \quad \frac{\partial \alpha_{il}}{\partial \boldsymbol{\Phi}_{(Q)}} \right]^T, \quad (30)$$

where $\boldsymbol{\Phi}$ is a $Q \times 1$ vector and $\boldsymbol{\Phi}_{(q)}$ is the q^{th} element. These partial derivatives are given by the chain and product rules as follows:

$$\frac{\partial \alpha_{il}}{\partial \boldsymbol{\Phi}_{(q)}} = -\frac{1}{2} (|2\pi\boldsymbol{\gamma}_{il}|)^{-\frac{1}{2}} \text{tr} \left(\boldsymbol{\gamma}_{il}^{-1} \frac{\partial \boldsymbol{\gamma}_{il}}{\partial \boldsymbol{\Phi}_{(q)}} \right), \quad (31a)$$

$$\frac{\partial \beta_{il}}{\partial \boldsymbol{\Phi}_{(q)}} = -\frac{1}{2} \left(\boldsymbol{\delta}_{il}^T \boldsymbol{\gamma}_{il}^{-1} \left(\frac{\partial \boldsymbol{\delta}_{il}}{\partial \boldsymbol{\Phi}_{(q)}} - \frac{\partial \boldsymbol{\gamma}_{il}}{\partial \boldsymbol{\Phi}_{(q)}} \boldsymbol{\gamma}_{il}^{-1} \boldsymbol{\delta}_{il} \right) + \left(\frac{\partial \boldsymbol{\delta}_{il}}{\partial \boldsymbol{\Phi}_{(q)}} \right)^T \boldsymbol{\gamma}_{il}^{-1} \boldsymbol{\delta}_{il} \right), \quad (31b)$$

$$\frac{\partial \epsilon_{in}}{\partial \boldsymbol{\Phi}_{(q)}} = -\frac{1}{2} (|2\pi\boldsymbol{\eta}_{in}|)^{-\frac{1}{2}} \text{tr} \left(\boldsymbol{\eta}_{in}^{-1} \frac{\partial \boldsymbol{\eta}_{in}}{\partial \boldsymbol{\Phi}_{(q)}} \right), \quad (31c)$$

$$\frac{\partial \zeta_{in}}{\partial \boldsymbol{\Phi}_{(q)}} = -\frac{1}{2} \left(\boldsymbol{\theta}_{in}^T \boldsymbol{\eta}_{in}^{-1} \left(\frac{\partial \boldsymbol{\theta}_{in}}{\partial \boldsymbol{\Phi}_{(q)}} - \frac{\partial \boldsymbol{\eta}_{in}}{\partial \boldsymbol{\Phi}_{(q)}} \boldsymbol{\eta}_{in}^{-1} \boldsymbol{\theta}_{in} \right) + \left(\frac{\partial \boldsymbol{\theta}_{in}}{\partial \boldsymbol{\Phi}_{(q)}} \right)^T \boldsymbol{\eta}_{in}^{-1} \boldsymbol{\theta}_{in} \right). \quad (31d)$$

The remaining partial derivatives depend on the particular motion model at hand.

The gradient of the ISE informs a gradient-descent algorithm which is governed by the following recurrence relation:

$$(\bar{\boldsymbol{\Phi}}_k)_{i+1} = (\bar{\boldsymbol{\Phi}}_k)_i - d_i [\nabla_{\bar{\boldsymbol{\Phi}}}(\text{ISE})]_{\boldsymbol{\Phi}=(\bar{\boldsymbol{\Phi}}_k)_i}. \quad (32)$$

This recurrence is repeated until the gradient vector becomes sufficiently small, as specified by some threshold. The step size d_i can be determined by any number of existing gradient-descent algorithms. Because the objective function is generally non-convex, it is critical to initialize gradient-descent with a good starting estimate. In practice, letting $(\bar{\Phi}_k)_0 = \bar{\Phi}_{k-1}$ often works well.

In multi-dimensional problems, minimization of the ISE corresponding to the single most recent state transition is generally not sufficient to guarantee convergence of estimated motion model parameters to ground-truth. For this reason, the objective function is redefined as the average of the M most recent state transitions:

$$ISE = \int_{\mathcal{X}} \left[\frac{1}{M+1} \sum_{i=k-M}^k (p(\mathbf{x}_i | \mathbf{z}_{1:i-1}) - p(\mathbf{x}_i | \mathbf{z}_{1:i}))^2 \right] d\mathbf{x}. \quad (33)$$

After some simplification, it can be shown that the gradient of the average ISE is the average of the ISEs of the M most recent state transitions:

$$\nabla_{\Phi}(ISE) = \frac{1}{M+1} \sum_{i=k-M}^k \nabla_{\Phi}(ISE_i). \quad (34)$$

The value of M used in motion model correction can be determined for a given context, but it is generally necessary to increase M as the dimensionality of the problem increases.

As the third term of Eq. (21) shows, having an estimate of the uncertainty in motion model parameters (Σ_{Φ}) is a critical part of the SEAM framework. Because these parameters are constantly being improved in the course of estimation, Σ_{Φ} ought to continuously reflect this increasing certainty. While a number of approximations may be considered, a heuristic one is described here for implementation. By letting $\Sigma_{\Phi_k} = \lambda \Sigma_{\Phi_{k-1}}$ with $0 < \lambda < 1$, motion model parameter uncertainty is guaranteed to decrease over time, with a rate determined by the power law λ^k .

7.3.3 Representative Nonlinear System Derivation

Thus far, enhanced estimation and motion model correction of nonlinear state-space systems have been formulated. The equations presented until this point apply to any general nonlinear system subject to non-Gaussian belief. Without entering into a specific context, derivations cannot develop further. For the purpose of validating this framework in the results section, this sub-section derives the common instructive nonlinear system of a

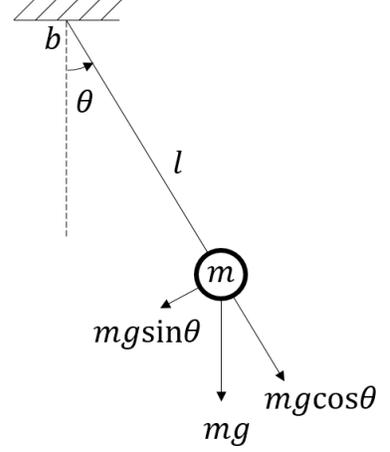


Figure 2. Pendulum FBD.

pendulum in the large-angle regime. Figure 2 shows the free-body diagram (FBD) of a pendulum, where m is the mass of the hanging object, l is the length of the string, and b is a coefficient accounting for frictional losses at the point of attachment and due to drag. By summing the moments about the point of attachment, the following continuous-time unforced equation of motion is obtained:

$$ml^2\ddot{\theta}(t) + b\dot{\theta}(t) + mgl\sin\theta(t) = 0. \quad (35)$$

In order to apply the SEAM framework to such a system, this differential equation must be discretized.

Let the continuous-time variables $\theta(t)$, $\dot{\theta}(t)$, and $\ddot{\theta}(t)$ at a time step k be approximated by the following:

$$\theta(k) = \theta_k, \quad \dot{\theta}(k) = \frac{\theta_{k+1} - \theta_k}{\Delta t}, \quad \ddot{\theta}(k) = \frac{\theta_{k+2} - 2\theta_{k+1} + \theta_k}{\Delta t^2}. \quad (36)$$

By substitution and rearrangement, Eq. (35) becomes:

$$\left(\frac{ml^2}{\Delta t^2}\right)\theta_{k+2} - \left(2\frac{ml^2}{\Delta t^2} - \frac{b}{\Delta t}\right)\theta_{k+1} + \left(\frac{ml^2}{\Delta t^2} - \frac{b}{\Delta t}\right)\theta_k + mgl\sin\theta_k = 0. \quad (37)$$

Let $x_{3,k} \equiv \theta_{k+2}$, $x_{2,k} \equiv \theta_{k+1} = x_{3,k-1}$, and $x_{1,k} \equiv \theta_k = x_{2,k-1}$. After solving Eq. (37) for θ_{k+2} and substituting, this system of equations can be written in matrix form as:

$$\begin{bmatrix} x_{1,k} \\ x_{2,k} \\ x_{3,k} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & \left(\frac{b\Delta t}{ml^2} - 1\right) & \left(2 - \frac{b\Delta t}{ml^2}\right) \end{bmatrix} \begin{bmatrix} x_{1,k-1} \\ x_{2,k-1} \\ x_{3,k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{g\Delta t}{l} \sin(x_{2,k-1}) \end{bmatrix}. \quad (38)$$

In order to eliminate redundancies in this equation, the first state is eliminated and the states are renamed:

$$\begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \left(\frac{b\Delta t}{ml^2} - 1\right) & \left(2 - \frac{b\Delta t}{ml^2}\right) \end{bmatrix} \begin{bmatrix} x_{1,k-1} \\ x_{2,k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{g\Delta t}{l} \sin(x_{2,k-1}) \end{bmatrix}. \quad (39)$$

This state-space equation governing the discretized nonlinear pendulum system can be represented more efficiently as:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{b}(x_{k-1}). \quad (40)$$

Let the angular position and velocity be the outputs of interest. From the first two equations of (36), the output equation is given by:

$$\begin{bmatrix} z_{1,k} \\ z_{2,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{\Delta t} & \frac{1}{\Delta t} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix}, \quad (41)$$

or more succinctly,

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k. \quad (42)$$

Because $x_{1,k}$ is just a time-shifted version of $x_{2,k}$, estimation of these signals is ill-posed. In order to directly estimate the angular position and velocity, Eqs. (40) and (42) can be combined:

$$\mathbf{z}_k = \mathbf{C}(\mathbf{A}\mathbf{x}_{k-1} + \mathbf{b}(x_{k-1})) = \mathbf{C}\mathbf{A}\mathbf{x}_{k-1} + \mathbf{C}\mathbf{b}(x_{k-1}). \quad (43)$$

Next, substitute $\mathbf{x}_{k-1} = \mathbf{C}^{-1}\mathbf{z}_{k-1}$ into Eq. (43):

$$\mathbf{z}_k = \mathbf{C}\mathbf{A}\mathbf{C}^{-1}\mathbf{z}_{k-1} + \mathbf{C}\mathbf{b}(\mathbf{C}^{-1}\mathbf{z}_{k-1}). \quad (44)$$

Finally, in order to give this a more compact state equation appearance as in Eq. (40), variables are redefined as follows: $\mathbf{x}_k \equiv \mathbf{z}_k$, $\mathbf{x}_{k-1} \equiv \mathbf{z}_{k-1}$, $\mathbf{A} \equiv \mathbf{CAC}^{-1}$, and $\mathbf{b}(\mathbf{x}_{k-1}) \equiv \mathbf{Cb}(\mathbf{C}^{-1}\mathbf{z}_{k-1})$. In full matrix form, this nonlinear state-space equation is now:

$$\begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 - \frac{b\Delta t}{ml^2} \end{bmatrix} \begin{bmatrix} x_{1,k-1} \\ x_{2,k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{g}{l} \sin(x_{1,k-1} + \Delta tx_{2,k-1}) \end{bmatrix}. \quad (45)$$

In terms of placeholder variables,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad (46a)$$

$$\mathbf{b}(\mathbf{x}_{k-1}) = \begin{bmatrix} b_1 \\ b_2 \sin(x_{1,k-1} + \Delta tx_{2,k-1}) \end{bmatrix}, \quad (46b)$$

where $a_{11} = 1$, $a_{12} = \Delta t$, $a_{21} = 0$, $a_{22} = 1 - \frac{b\Delta t}{ml^2}$, $b_1 = 0$, and $b_2 = -\frac{g}{l}$.

For this nonlinear system, the Jacobians of Eqs. (22) are given by:

$$\mathbf{J}_f^x = \mathbf{A} + \mathbf{J}_{\mathbf{b}(\mathbf{x}_{k-1})}^x, \quad (47a)$$

$$\mathbf{J}_f^u = \mathbf{0}, \quad (47b)$$

$$\mathbf{J}_f^\Phi = \begin{bmatrix} \mathbf{x}_{k-1}^T & \mathbf{0} & 1 & 0 \\ \mathbf{0} & \mathbf{x}_{k-1}^T & 0 & \sin(x_{1,k-1} + \Delta tx_{2,k-1}) \end{bmatrix}, \quad (47c)$$

where

$$\mathbf{J}_{\mathbf{b}(\mathbf{x}_{k-1})}^x = b_2 \cos(x_{1,k-1} + \Delta tx_{2,k-1}) \begin{bmatrix} 0 & 0 \\ 1 & \Delta t \end{bmatrix}. \quad (48)$$

In order to compute the gradient of the ISE, the partial derivatives of $\boldsymbol{\gamma}_{il}$, $\boldsymbol{\delta}_{il}$, $\boldsymbol{\eta}_{in}$, and $\boldsymbol{\theta}_{in}$ must be obtained. By combination of Eqs. (19)-(21) and (29), and following the product rule, these four derivatives are obtained as:

$$\frac{\partial \boldsymbol{\gamma}_{il}}{\partial \boldsymbol{\Phi}_{(q)}} = \mathbf{J}_f^{x_i} \boldsymbol{\Sigma}_{x_i}^{(k-1|1:k-1)} \frac{\partial \mathbf{J}_f^{x_i^T}}{\partial \boldsymbol{\Phi}_{(q)}} + \frac{\partial \mathbf{J}_f^{x_i}}{\partial \boldsymbol{\Phi}_{(q)}} \boldsymbol{\Sigma}_{x_i}^{(k-1|1:k-1)} \mathbf{J}_f^{x_i^T}$$

$$+ \mathbf{J}_f^{x_l} \boldsymbol{\Sigma}_{x_l}^{(k-1|1:k-1)} \frac{\partial \mathbf{J}_f^{x_l T}}{\partial \boldsymbol{\Phi}_{(q)}} + \frac{\partial \mathbf{J}_f^{x_l}}{\partial \boldsymbol{\Phi}_{(q)}} \boldsymbol{\Sigma}_{x_l}^{(k-1|1:k-1)} \mathbf{J}_f^{x_l T}, \quad (49a)$$

$$\frac{\partial \boldsymbol{\delta}_{il}}{\partial \boldsymbol{\Phi}_{(q)}} = \frac{\partial \mathbf{A}}{\partial \boldsymbol{\Phi}_{(q)}} \left(\bar{\mathbf{x}}_i^{(k|1:k-1)} - \bar{\mathbf{x}}_l^{(k|1:k-1)} \right) + \frac{\partial}{\partial \boldsymbol{\Phi}_{(q)}} \left(\mathbf{b} \left(\bar{\mathbf{x}}_i^{(k|1:k-1)} \right) - \mathbf{b} \left(\bar{\mathbf{x}}_l^{(k|1:k-1)} \right) \right), \quad (49b)$$

$$\frac{\partial \boldsymbol{\eta}_{in}}{\partial \boldsymbol{\Phi}_{(q)}} = \mathbf{J}_f^{x_i} \boldsymbol{\Sigma}_{x_i}^{(k-1|1:k-1)} \frac{\partial \mathbf{J}_f^{x_i T}}{\partial \boldsymbol{\Phi}_{(q)}} + \frac{\partial \mathbf{J}_f^{x_i}}{\partial \boldsymbol{\Phi}_{(q)}} \boldsymbol{\Sigma}_{x_i}^{(k-1|1:k-1)} \mathbf{J}_f^{x_i T}, \quad (49c)$$

$$\frac{\partial \boldsymbol{\theta}_{in}}{\partial \boldsymbol{\Phi}_{(q)}} = \frac{\partial \mathbf{A}}{\partial \boldsymbol{\Phi}_{(q)}} \left(\bar{\mathbf{x}}_i^{(k|1:k-1)} \right) + \frac{\partial}{\partial \boldsymbol{\Phi}_{(q)}} \left(\mathbf{b} \left(\bar{\mathbf{x}}_i^{(k|1:k-1)} \right) \right). \quad (49d)$$

Several of the terms which would arise from Eq. (21) do not appear in Eqs. (49) because they are not functionally dependent on model parameters $\boldsymbol{\Phi}_{(q)}$.

For the pendulum example, the parameter vector is given by $\boldsymbol{\Phi} = [a_{11} \ a_{12} \ a_{21} \ a_{22} \ b_1 \ b_2]^T$. Obtaining the six corresponding partial derivatives for each of the four variables of Eqs. (49) is straightforward. For instance,

$$\frac{\partial \mathbf{A}}{\partial \boldsymbol{\Phi}_{(a_{11})}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \quad (50)$$

The overall gradient of the ISE can then be computed by cascading the formulas presented in section 7.3.2.

7.4 Results

7.4.1 Description of Monte-Carlo Simulations

In order to validate the proposed nonlinear SEAM framework, a Monte-Carlo simulation approach was taken. By conducting simulated experiments with randomly generated parameters, three things are accomplished: 1) the ubiquity of the framework can be demonstrated since it can be shown to work for more than just specially-chosen combinations of parameters, 2) performance evaluations can be made since ground-truth information is available for comparison, and 3) more

meaningful statistical summaries of the proposed methods can be acquired since data is aggregated from many experiments.

The ground-truth state of the pendulum's free-response was simulated for six seconds with the parameter values given in Table 1. The initial angle was chosen to be slightly less than π (vertical) to emphasize the nonlinearity of motion in large-angle simulation. Gaussian process noise with covariance Σ_w was added at each step in keeping with Eq. (18).

Table 1. Ground-truth parameters

l	m	b	Δt	θ_0	$\dot{\theta}_0$	Σ_w
100	0.015	15	0.02	3.141	0	$10^{-7}[1, 0; 0, 1.7]$

In order to simulate sensor noise, a zero-mean Gaussian random signal with covariance Σ_v was added to the ground-truth signal. Furthermore, the uncertainty in each physical model parameter was given as a proportion ε of its mean. From this, the uncertainties in the model parameters were derived. The initial parameter estimates used for signal estimation were then randomly generated from a normal distribution with means given by the ground-truth value and standard deviations given by the values in Table 2. This table also contains the standard deviations in initial conditions used to seed the estimators.

Table 2. Parameter uncertainties

σ_l	σ_m	σ_b	σ_{θ_0}	$\sigma_{\dot{\theta}_0}$
εl	εm	εb	0.2	0.08

For comparison in validation, the EKF and AEKF were implemented in addition to nonlinear SEAM. In order to isolate the effect of model parameter correction, SEAM was implemented both without (SEAM⁻) and with (SEAM⁺) model correction. The following two subsections demonstrate the results of these simulations both qualitatively and quantitatively.

7.4.2 Qualitative Assessment

Figure 3 shows plots of angular position and velocity for the ground-truth, observed, and estimated signals of a sample simulation. For this experiment, baseline parameters are given by Tables 1 and 2 and other simulation-specific parameters are given in Table 3 below.

Table 3. Qualitative example parameter values

M	λ	d_i	ε	Σ_v
50	0.97	0.0005	0.8	[0.075, -0.008; -0.008, 0.06]

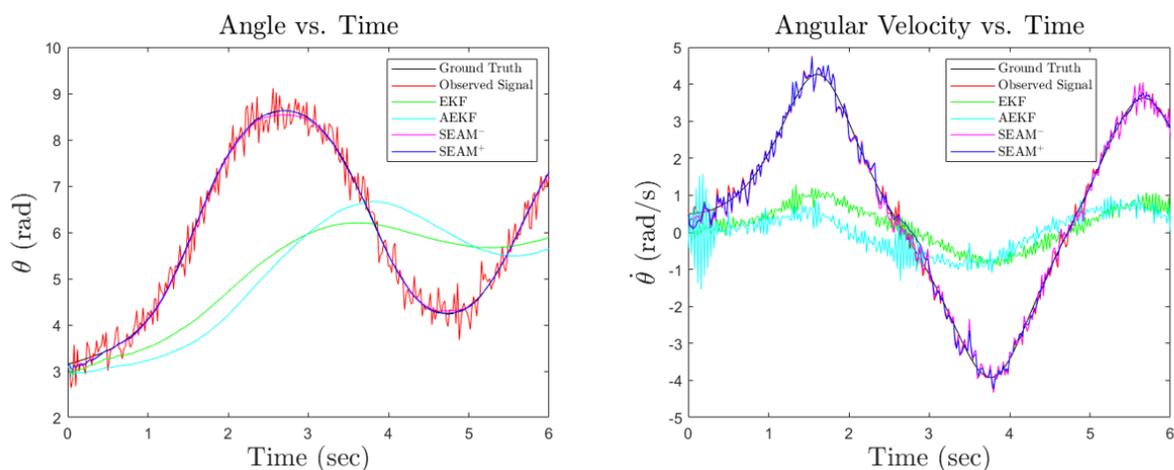


Figure 3. Time domain plots.

As the figure shows, the KF-based approaches fail to adequately estimate the state of the pendulum due to high uncertainty in model parameters. Without incorporating this uncertainty in estimation, the prediction stage of RBE becomes overconfident and the more accurate sensor observations are essentially ignored. This problem is overcome by the inclusion of model parameter uncertainty (i.e. the third term of Eq. (21)). Even without model correction, this addition allows the estimator to rely more heavily on observations and thereby weight erroneous predictions less. However, without the reinforcing information provided by an accurate prediction model, the SEAM estimator without model correction is heavily influenced by sensor noise. This is particularly apparent in the fluctuations of the magenta $\dot{\theta}$ vs. t plot of Fig. 3.

Model parameter correction is introduced in order to increase the robustness of SEAM estimation to sensor noise. By correcting model parameters and their uncertainties, the smoothing effect of a Kalman filter is achieved simultaneously with the zero-mean-error estimation of SEAM. The blue plots in Fig. 3 show how, as model parameters are corrected and their uncertainties reduced, the SEAM⁺ signal deviates from the SEAM⁻ signal towards a smoother estimate of the ground-truth state. This smoothing is largely a result of the convergence of Σ_{Φ_k} by the power law λ^k as described at the end of section 7.3.2. However, it is only because $\bar{\Phi}$ becomes more accurate that the error in this signal is zero-mean. Figure 4 shows the convergence of model parameters towards ground-truth.

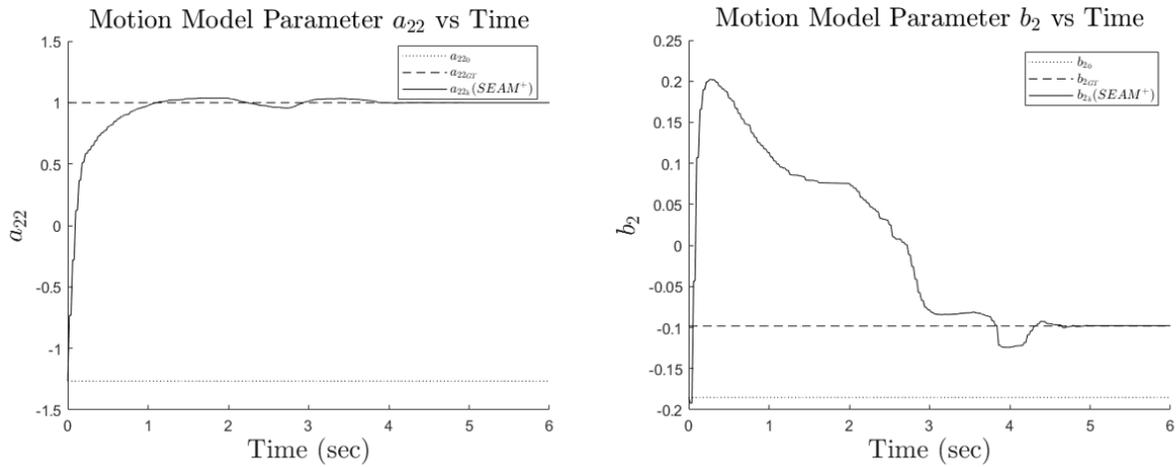


Figure 4. Model correction over time.

Of the six presented model parameters, only a_{22} and b_2 can be corrected for the nonlinear pendulum because they depend on the physical parameters of the system according to Eq. (45). As Fig. 4 demonstrates, these two parameters converge to ground-truth over the course of the six-second simulation. The speed and constancy of convergence are largely dependent on the fine-tuning of gradient-descent parameters. A more thorough performance optimization study is warranted, but extends beyond the scope of this work. Furthermore, a rigorous mathematical

justification of convergence and stability is not possible because the presented solution can only be implemented in open-form via gradient-descent.

Finally, Fig. 5 contains error plots for the sample simulation. The first part of the figure shows the time-domain errors of all the signals, while the second part of the figure plots the ISE for all time steps using the initial model parameter vector estimate Φ_0 , the ground-truth parameter vector Φ_G , and the continuously corrected Φ_k . As is apparent from the figure, SEAM⁻ and SEAM⁺ provide comparable performance, with the exception being that SEAM⁺ outperforms in the second half of simulation when the model estimate is improved. Average estimation errors are summarized for this sample simulation in Table 4.

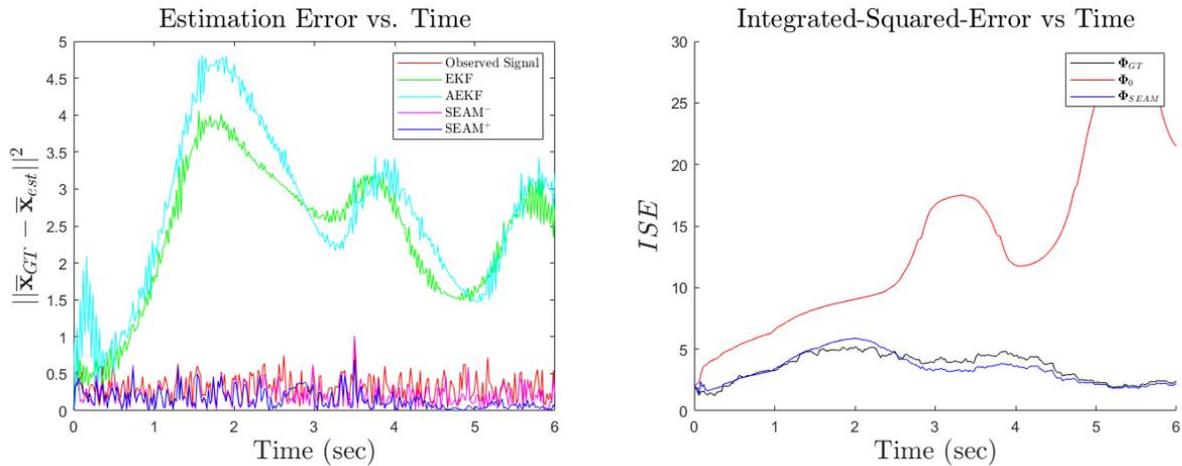


Figure 5. State estimation error and ISE for sequential time steps corresponding to estimates of Φ .

Table 4. Root-mean-squared-errors (RMSEs) for each signal

Observations	EKF	AEKF	SEAM ⁻	SEAM ⁺
0.252	1.824	2.033	0.171	0.140

According to the second part of Fig. 5, while continuing to use a poor initial estimate of model parameters Φ_0 causes the ISE to increase, updating these parameters yields an ISE that is often below even that given by the ground-truth parameters. It is this proven ability of the proposed

technique to minimize the ISE that results in such effective estimation in comparison with other methods.

7.4.3 Quantitative Assessment

In order to quantitatively assess nonlinear SEAM, 50 simulated experiments were carried out for each of 16 combinations of parameters ε and Σ_v . For each experimental configuration, various outcomes were measured and reported in Tables 5 through 7 below. For all trials, the parameters M , λ , and d_i were set to the values specified in Table 3. Other baseline parameters were given by Tables 1 and 2. Because the proposed framework accommodates non-Gaussian belief, it is important that non-Gaussian SEAM be validated. However, since this was done in previous work [7], the simulations of this section are implemented for Gaussian belief.

Table 5 summarizes the performance of each estimator in the aforementioned simulations. The percent of trials for which each estimator won is reported, where an estimator “wins” when it yields an RMSE lower than the rest. As the table shows, the SEAM estimators outperform the KFs handily for all but a select few cases. For relatively low sensor noise, motion model correction is most successful and the SEAM⁺ estimator wins in the majority of simulations. However, as Σ_v increases, corrected model estimates under SEAM⁺ do not converge sufficiently to outperform SEAM⁻ in the time allotted. Furthermore, there is a slight negative trend in SEAM⁺ performance for increasing ε ; this can be explained by the fact that initial model estimates are more likely to be further from ground-truth as ε increases. This means model convergence will take longer, and for this reason SEAM⁺ is less successful over such short time periods.

Table 5. Percent of wins in 50 trials for each estimator. $E = EKF$, $A = AEKF$, $S^- = SEAM$, and $S^+ = SEAM^+$

		ε							
		0.2	0.4	0.6	0.8				
Σ_v	$\begin{bmatrix} 0.075 & -0.008 \\ -0.008 & 0.06 \end{bmatrix}$	E	0%	E	0%	E	0%	E	0%
		A	0%	A	0%	A	0%	A	0%
		S1-	8%	S1-	2%	S1-	2%	S1-	14%
		S1+	92%	S1+	98%	S1+	98%	S1+	86%
	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.08 \end{bmatrix}$	E	0%	E	0%	E	0%	E	0%
		A	0%	A	0%	A	8%	A	0%
		S1-	12%	S1-	8%	S1-	10%	S1-	36%
		S1+	88%	S1+	92%	S1+	90%	S1+	64%
	$\begin{bmatrix} 0.2 & 0.01 \\ 0.01 & 0.32 \end{bmatrix}$	E	0%	E	0%	E	0%	E	0%
		A	2%	A	0%	A	0%	A	0%
		S1-	26%	S1-	44%	S1-	66%	S1-	70%
		S1+	72%	S1+	56%	S1+	34%	S1+	30%
	$\begin{bmatrix} 0.25 & 0.01 \\ 0.01 & 0.56 \end{bmatrix}$	E	0%	E	0%	E	2%	E	0%
		A	0%	A	2%	A	0%	A	0%
		S1-	44%	S1-	54%	S1-	38%	S1-	54%
		S1+	56%	S1+	44%	S1+	60%	S1+	46%

Table 6 summarizes the average ISE found for each of the 50-simulation trials calculated with the initial model parameters Φ_0 , the ground-truth parameters Φ_G , and the updated parameters Φ_k . In all but two of the most extreme cases where both ε and Σ_v are high, using the corrected model parameters successfully reduces the ISE even below what is obtained using Φ_G . This verifies the efficacy of the formulas presented for $\nabla_{\Phi}(ISE)$ and shows that the gradient-descent algorithm of Eq. (32) effectively minimizes the ISE. In simulations where the ISE is successfully minimized but SEAM⁺ does not win, the most likely explanation is that the randomness arising from noise processes causes the ISE-minimizing parameter vector to deviate from the ground-truth parameter vector.

Table 6. Average ISEs given by Φ_0 , Φ_G , and Φ_k

		ε				
		0.2	0.4	0.6	0.8	
Σ_v	$\begin{bmatrix} 0.075 & -0.008 \\ -0.008 & 0.06 \end{bmatrix}$	Φ_0	17.1	Φ_0 13.3	Φ_0 12.5	Φ_0 11.6
		Φ_G	3.0	Φ_G 3.2	Φ_G 3.6	Φ_G 3.9
		Φ_k	2.5	Φ_k 2.8	Φ_k 3.2	Φ_k 3.5
	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.08 \end{bmatrix}$	Φ_0	12.6	Φ_0 10.2	Φ_0 9.1	Φ_0 8.4
		Φ_G	2.0	Φ_G 2.3	Φ_G 2.5	Φ_G 2.8
		Φ_k	1.8	Φ_k 2.0	Φ_k 2.2	Φ_k 2.4
	$\begin{bmatrix} 0.2 & 0.01 \\ 0.01 & 0.32 \end{bmatrix}$	Φ_0	5.2	Φ_0 4.3	Φ_0 3.8	Φ_0 3.2
		Φ_G	0.8	Φ_G 0.8	Φ_G 0.9	Φ_G 2.7
		Φ_k	0.5	Φ_k 0.6	Φ_k 0.6	Φ_k 3.0
	$\begin{bmatrix} 0.25 & 0.01 \\ 0.01 & 0.56 \end{bmatrix}$	Φ_0	3.8	Φ_0 2.8	Φ_0 2.3	Φ_0 2.0
		Φ_G	0.7	Φ_G 1.5	Φ_G 0.7	Φ_G 1.4
		Φ_k	0.4	Φ_k 0.6	Φ_k 0.5	Φ_k 1.7

Table 7 is given for a quantitative summary of the ability of SEAM⁺ to improve model parameters a_{22} and b_2 . The table shows the average percent of the time during each simulation that model parameters were improved. The number of time steps for which a_{22k} and b_{2k} were closer to Φ_G than to Φ_0 were divided by the total number of time steps. As the table shows, there is a strong correlation between improvement in Φ and minimization of the ISE. The fact that a_{22} is more often successfully corrected suggests that the ISE objective function is more sensitive to change in a_{22} than to change in b_2 .

Table 7. Average percent of time model parameters were improved

		ε				
		0.2	0.4	0.6	0.8	
Σ_v	$\begin{bmatrix} 0.075 & -0.008 \\ -0.008 & 0.06 \end{bmatrix}$	a_{22}	96.3%	a_{22} 92.1%	a_{22} 92.4%	a_{22} 78.2%
		b_2	65.1%	b_2 66.8%	b_2 64.8%	b_2 54.9%
	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.08 \end{bmatrix}$	a_{22}	90.1%	a_{22} 88.4%	a_{22} 84.5%	a_{22} 59.5%
		b_2	62.2%	b_2 61.9%	b_2 64.7%	b_2 53.8%
	$\begin{bmatrix} 0.2 & 0.01 \\ 0.01 & 0.32 \end{bmatrix}$	a_{22}	68.5%	a_{22} 57.0%	a_{22} 42.6%	a_{22} 26.8%
		b_2	44.6%	b_2 45.2%	b_2 26.4%	b_2 29.3%
	$\begin{bmatrix} 0.25 & 0.01 \\ 0.01 & 0.56 \end{bmatrix}$	a_{22}	60.9%	a_{22} 44.4%	a_{22} 49.2%	a_{22} 32.7%
		b_2	40.9%	b_2 29.6%	b_2 30.1%	b_2 28.1%

7.5 Conclusions and Future Work

Several conclusions can be drawn about the proposed nonlinear/non-Gaussian SEAM framework. First, the inclusion of motion model uncertainty in state prediction is demonstrated to significantly improve the quality of nonlinear state estimation as compared to conventional Kalman filters when model parameters are poorly estimated. Second, because this approach results in higher sensitivity to observation noise, the addition of a model correction stage further refines the quality of estimation. This is accomplished by both improving model parameters and reducing their estimated uncertainty. For hundreds of randomly generated Monte-Carlo simulations, the proposed techniques are shown to work well under moderate sensor noise when model uncertainty is appreciable.

In providing a fair assessment of the proposed framework, a summary of its limitations is necessary. Because gradient-descent is required for model correction, updating model parameters at each time step can be computationally costly and ineffective for real-time applications. This can be rectified by choosing to employ model correction only every Y time steps, where Y would be adjusted according to an accuracy/efficiency trade-off. Furthermore, because the framework is intended to handle some of the more difficult estimation problems, it is not always a good “out-of-the-box” solution. While the general approach is ubiquitous, certain elements must be tuned according to the context at hand.

The future work related to this research is ample. First, correcting the physical pendulum parameters l , m , and b rather than the abstracted model parameters a_{22} and b_2 may improve the quality of the framework. Second, this work could benefit from a closer examination of effective gradient-descent techniques to improve the efficiency of model correction. Finally, the ubiquity of the approach would be further clarified if it was validated for a wider range of nonlinear state-

space systems with varying dimensionalities and complexities. Nevertheless, as a proof-of-concept, this paper shows that nonlinear/non-Gaussian SEAM holds potential for handling complex estimation problems in a variety of contexts. SDG

7.6 References

- [1] J. Sun, H.A.P. Blom, J. Ellerbroek, J.M. Hoekstra, "Aircraft Mass and Thrust Estimation Using Recursive Bayesian Method," Proceedings of the International Conference on Research in Air Transportation, Barcelona, Spain, 2018.
- [2] M. Kumon, D. Kimoto, K. Takami, T. Furukawa, "Bayesian Non-Field-of-View Target Estimation Incorporating An Acoustic Sensor," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3425-3432, 2013.
- [3] J. Vetelino, A. Reghu, "Introduction to Sensors," Boca Raton, FL, USA: CRC Press, pp. 1, 2011.
- [4] Y. Chen, C. Chen, "Sensor Deployment under Probabilistic Sensing Model," Proceedings of the 2nd High Performance Computing and Cluster Technologies Conference, Beijing, China, pp. 33-36, 2018.
- [5] H. Liu, Y. Pan, J. Cao, "Composite Learning Adaptive Dynamic Surface Control of Fractional-Order Nonlinear Systems," IEEE Transactions On Cybernetics, Early access, 2019.
- [6] V. Stojanovic, N. Nedic, "Joint state and parameter robust estimation of stochastic nonlinear systems," International Journal of Robust and Nonlinear Control, vol. 26, no. 14, pp. 3058-3074, 2016.
- [7] J. J. Steckenrider, T. Furukawa, "Probabilistic Motion Model Correction Using Non-Gaussian Belief Fusion," Automatica, *Under revision*, 2019.

- [8] D. L. Alspach, "Gaussian Sum Approximations in Nonlinear Filtering and Control," *Information Sciences*, vol. 7, pp. 271-290, 1974.
- [9] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. D, pp. 35-45, 1960.
- [10] T. Moore, D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," *Intelligent Autonomous Systems*, vol. 13, pp. 335-348, 2015.
- [11] A. Olivier, A. W. Smyth, "A marginalized unscented Kalman filter for efficient parameter estimation with applications to finite element models," *Computer Methods in Applied Mechanics and Engineering*, vol. 339, pp. 615-643, 2018.
- [12] D. Potnuru, K. P. B. Chandra, I. Arasaratnam, D. Gu, K. A. Mary, S. B. Ch, "Derivative-free square-root cubature Kalman filter for non-linear brushless DC motors," *IET Electric Power Applications*, vol. 10, no. 5, pp. 419-429, 2016.
- [13] J. R. Stroud, M. Katzfuss, C. K. Wikle, "A Bayesian Adaptive Ensemble Kalman Filter for Sequential State and Parameter Estimation," *Monthly Weather Review*, vol. 146, pp. 373-386, 2018.
- [14] Y. Yang, H. Cai, B. Gong, R. Norman, "Schmidt-Kalman Filter with Polynomial Chaos Expansion for State Estimation," *22nd International Conference on Information Fusion*, Ottawa, Ontario, CA, 2019.
- [15] D. Raihan, S. Chakravorty, "Particle Gaussian Mixture Filters: Application and Performance Evaluation," *22nd International Conference on Information Fusion*, Ottawa, Ontario, CA, 2019.
- [16] M. Pulido, P. J. Leeuwen, "Sequential Monte Carlo with kernel embedded mappings: The mapping particle filter," *Journal of Computational Physics*, vol. 396, pp. 400-415, 2019.

- [17] C. Kalender, A. Schöttl, “Sparse Grid-Based Nonlinear Filtering,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2386-2396, 2013.
- [18] N. J. Gordon, D. J. Salmond, A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings F- Radar and Signal Processing*, vol. 140, vol. 2, pp. 107-113, 1993.
- [19] V. Stojanovic, N. Nedic, “Robust Kalman filtering for nonlinear multivariable stochastic systems in the presence of non-Gaussian noise,” *International Journal of Robust and Nonlinear Control*, vol. 26, no. 3, pp. 445-460, 2016.
- [20] R.S. Esfandiari, B. Lu, “Modeling and Analysis of Dynamic Systems, Third Edition” Boca Raton, FL, USA: CRC Press, ch. 4, 2018.
- [21] L. Ljung, “System Identification,” in *Signal Analysis and Prediction*, Boston, MA, Birkhäuser, 1998, pp. 163-173.
- [22] P. Yu, M. Wu, J. She, K. Liu, Y. Nakanishi, “Robust Tracking and Disturbance Rejection for Linear Uncertain System With Unknown State Delay and Disturbance,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1445-1455, 2018.
- [23] E. Zerdali, “Adaptive Extended Kalman Filter for Speed-Sensorless Control of Induction Motors”, *IEEE Transactions on Energy Conversion*, vol. 34, no. 2, pp. 789-800, 2019.
- [24] A. Katriniok, D. Abel, “Adaptive EKF-Based Vehicle State Estimation With Online Assessment of Local Observability,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1368-1381, 2016.
- [25] C. F. Ansley, R. Kohn, “Estimation, Filtering, and Smoothing in State Space Models with Incompletely Specified Initial Conditions,” *The Annals of Statistics*, vol. 13, no. 4, pp. 1286-1316, 1985.

- [26] A. Sani, A. Vosoughi, "On Distributed Linear Estimation With Observation Model Uncertainties," vol. 66, no. 12, pp. 3212-3227, 2018.
- [27] J. E. Mottershead, M. Link, M. I. Friswell, "The sensitivity method in finite element model updating: A tutorial," *Mechanical Systems and Signal Processing*, vol. 25, no. 7, pp. 2275-2296, 2011.
- [28] A. Lund, S. J. Dyke, W. Song, I. Billionis, "Global sensitivity analysis for the design of nonlinear identification experiments," *Nonlinear Dynamics*, vol. 98, pp. 375-394, 2019.
- [29] E. Patelli, Y. Govers, M. Broggi, H. M. Gomes, M. Link, J. E. Mottershead, "Sensitivity or Bayesian model updating: a comparison of techniques using the DLR AIRMOD test data," *Archive of Applied Mechanics*, vol. 87, no. 5, pp. 905-925, 2017.
- [30] K. Oh, J. Seo, "Inertial Parameter Estimation of an Excavator with Adaptive Updating Rule Using Performance Analysis of Kalman Filter," *International Journal of Control, Automation and Systems*, vol. 16, no. 3, pp. 1226-1238, 2018.
- [31] Y. Pan, T. Sun, H. Yu, "Composite adaptive dynamic surface control using online recorded data," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 18, pp. 3921-3936, 2016.
- [32] K. Yuen, H. Mu, "Real-Time System Identification: An Algorithm for Simultaneous Model Class Selection and Parametric Identification," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, pp. 785-801, 2015.
- [33] K. Yuen, H. Mu, "Self-calibrating Bayesian real-time system identification," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 9, pp. 806-821, 2019.
- [34] J. J. Steckenrider, T. Furukawa, "A Probabilistic Model-Adaptive Approach for Tracking of Motion with Heightened Uncertainty," *International Journal of Control, Automation, and Systems*, *Accepted*, 2019.

CHAPTER 8. Framework Unification and Applications

8.1 Framework Unification

In order to further unify the concepts presented throughout this dissertation, a concluding summary of this doctoral work is appropriate. The chief problem which is addressed by the formulations and frameworks of this research is summarized as follows: what can be done to improve the knowledge about a set of variables when the ability to know them becomes increasingly obscured? Information fusion is the foundation upon which answers to this problem are built, and various estimation frameworks solve this problem by leveraging the concept of information fusion in different ways. This dissertation has addressed two estimation frameworks. The first, recursive Bayesian classification (RBC), implements a Gaussian toroid prediction model for contexts where motion is highly stochastic but can be modeled by an N -D random walk. The second and more thoroughly investigated framework, simultaneous estimation and modeling (SEAM), addresses estimation where model parameters are not accurately known. The applications such estimation approaches are explored in section 8.2.

The SEAM framework upon which chapters 4-7 build is shown to improve state estimation when there is uncertainty in parameters of either motion or observation models. Furthermore, the ubiquity of the framework is extended to the more general case where state belief is non-Gaussian. What is not addressed in the above chapters, however, is a unification of the framework where *both* observation and motion model parameters are improved over time. The diagram of Fig. 1 shows what such a technique would look like. As the diagram demonstrates, both Φ and Ψ would be corrected by the ISE-optimization methods discussed in the text using corrected, predicted, and observed state belief.

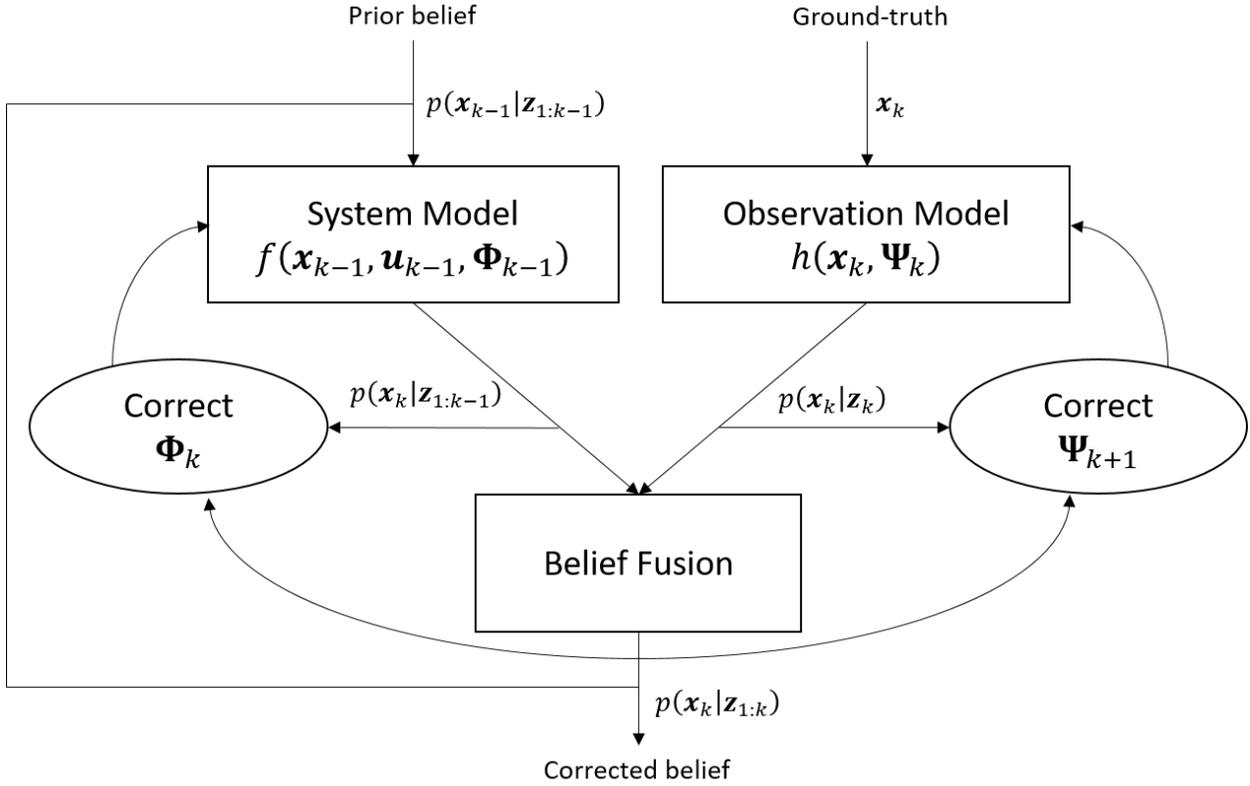


Figure 1. Unified SEAM diagram.

Though it would be an elegant and comprehensive approach to solving the kinds of difficult estimation problems posed here, a conceptual evaluation of such a unified framework casts doubt on its actual efficacy. In the contexts already studied, when a prediction is erroneous due to poor knowledge of the parameter vector Φ , an accurate observation will improve the state estimate. The converse is true when Ψ is poorly known. In each of these cases, there is an assumption that corrected belief $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ is more accurate than predicted belief $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ or observed belief $p(\mathbf{x}_k | \mathbf{z}_k)$, respectively. This assumption underlies the use of ISE minimization as an effective means of model correction. However, suppose both Φ and Ψ are poorly known. In this case, both $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k | \mathbf{z}_k)$ will be bad estimates of the true state \mathbf{x}_k . Therefore, when these PDFs are fused, it cannot be known if the resulting corrected PDF $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ is an improvement for the predictor or the observer. This is visually demonstrated in Fig. 2 below.

As the figure shows, while neither PDF is a particularly good estimate of \mathbf{x}_k since both Φ and Ψ are poorly known, state belief fusion happens to result in an improved estimate from the perspective of the predictor. Therefore, correcting Φ by minimizing ISE_f would be a valid approach. However, since $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is a worse estimate of \mathbf{x}_k than $p(\mathbf{x}_k|\mathbf{z}_k)$, correcting

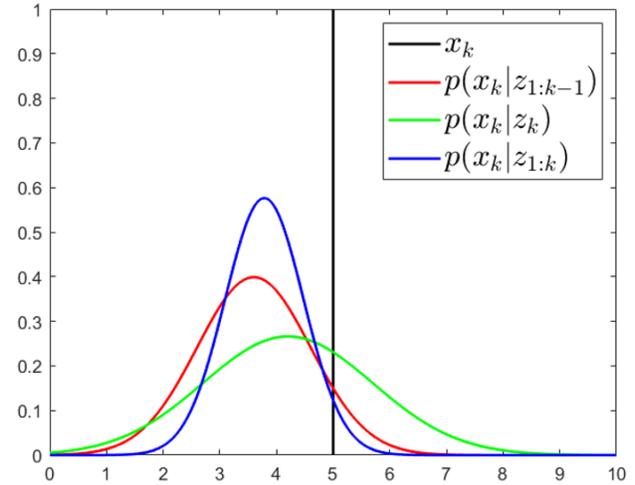


Figure 2. Insufficiency of simultaneous predictor-observer correction.

Ψ by minimizing ISE_h would cause observation model parameters to deviate from ground-truth. At any time step, the opposite may be true; in general, Φ and Ψ cannot both be corrected for this reason. However, it is possible to solve this issue by adding information to the estimator.

If a second observer (or sensor) is incorporated in estimation, for which model parameters Ψ_2 are known with sufficient accuracy, the state belief coming from the fusion of both sensors should allow for correction of Ψ_1 by minimization of ISE_h . Similarly, the fusion of belief coming from prediction with that coming from the second sensor should allow for correction of Φ by minimization of ISE_f . This is because it can be confidently asserted that the corrected state belief $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is more accurate than either $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ or $p(\mathbf{x}_k|\mathbf{z}_k)$. If, however, Ψ_2 is also poorly known, yet another more accurate source of information is required to correct model parameters. This leads to the generalization that, if motion and observation model parameters are to be reliably and simultaneously corrected under a unified SEAM framework, corrected state belief must always be more accurate than *all* of the predicted and observed PDFs. This can be accomplished, in theory, by either 1) adding an accurate observer o which gives observed state belief $p(\mathbf{x}_k|\mathbf{z}_k)_o$ that has extremely low uncertainty, or 2) deploying a suite of O observers for which each parameter vector

Ψ_o may be poorly known, but for which all observed PDFs $p(x_k|z_k)_o$ cluster around the true state x_k . This principle is illustrated in Fig. 3.

As the figure shows, while each individual sensor may provide an inaccurate estimate of the state, their fusion offers a much better estimate. Using this fused PDF to then obtain the corrected state belief $p(x_k|z_{1:k})$ will allow for motion and observation model correction using ISE

minimization, despite the fact that all model parameters may be poorly known. An analogy that may be considered is as follows: consider a jury which has been tasked with ascertaining the truth about a criminal case based only on the evidence submitted in court. It can be reasonably expected that the

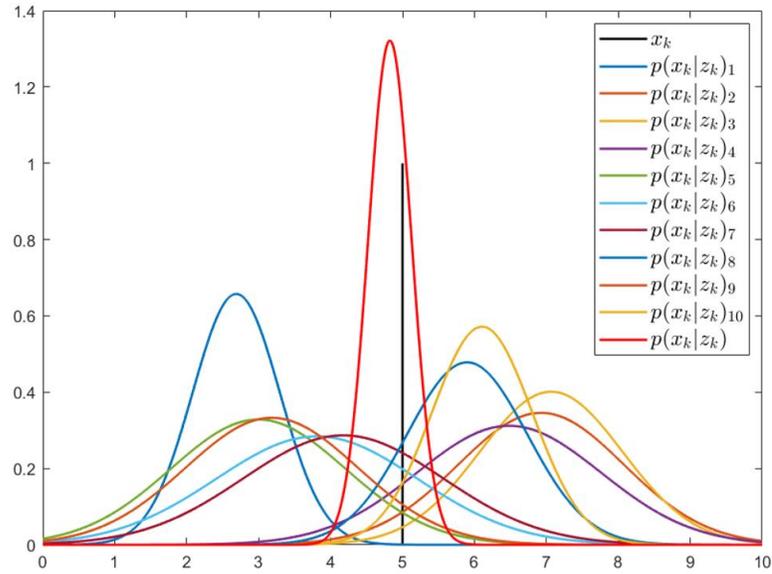


Figure 3. Multi-sensor fusion.

prosecution and defense have both presented evidence and statements that are a skewed version of the truth. The ground-truth is known only by the defendant, but it will never be made public knowledge since a confession would incriminate him. It is the responsibility of the jury to work out what most likely happened based on what they've been given, but each member will naturally bring a slightly different perspective and line of reasoning to the deliberation. Though each juror alone would likely do a poor job of working out the truth due to their own imperfect detective skills, the combined effort of the group will inevitably yield a result that is more accurate than any individuals'. In analogy to the idea of model correction, the final decision of the jury could then be used to improve each juror's reasoning and truth-finding skills.

In summary, a unified SEAM framework in which motion and observation model parameters are all simultaneously corrected is only possible if the fusion of all observed and predicted state belief yields a PDF which is a better estimate of the state as compared to each individual prediction or observation. Investigating and validating this assertion is an aim of future work, as is extending SEAM to nonlinear observation models. It is worth noting that, as this research becomes increasingly theoretical and ubiquitous, the domain of actual contexts for which extended formulations are justified becomes smaller. Nevertheless, the foundational SEAM framework presented in this dissertation has several potential applications which are addressed in the following section.

8.2 Applications

To conclude this work, it is appropriate to summarize the various potential applications of the research presented in this dissertation. The theoretical formulations developed in the first seven chapters have merit of their own, but without considering applied contexts, the relevance and usefulness of this work is unclear. The following two subsections explore applications related to state estimation and target tracking.

8.2.1 Signal Processing and State Estimation

The area of state estimation is concerned with actively filtering sensor signals coming from primarily dynamic, electric, or kinematic systems. Measuring such signals is challenging when sensors are corrupted with noise; when state feedback is employed to control the system, discontinuities such noisy signals can become problematic. This mandates the use of an active filter which leverages deterministic knowledge of the mathematics of the system at hand to improve state estimates.

Below are a few examples of deterministic physical systems across many domains for which state estimation may be necessary in order to overcome sensor noise. In some cases, this is to allow for smooth feedback control, while in other cases there may be some alternative decision-making effort employed. Some of these are directly addressed in the chapters above. For thoroughness, the governing discrete- or continuous-time state-space equations are also given for each example.

Resistance-Capacitance Circuit

$$[\dot{v}(t)] = \left[-\frac{1}{RC}\right][v(t)]$$

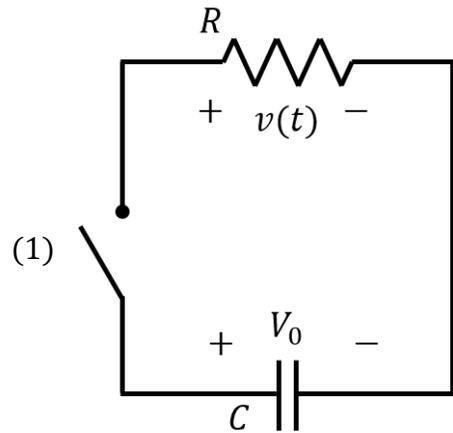


Figure 4. RC circuit.

Mass-Spring-Damper

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\dot{x}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k}{m} & \frac{b}{m} \end{bmatrix} \begin{bmatrix} u(t) \\ \dot{u}(t) \end{bmatrix}$$

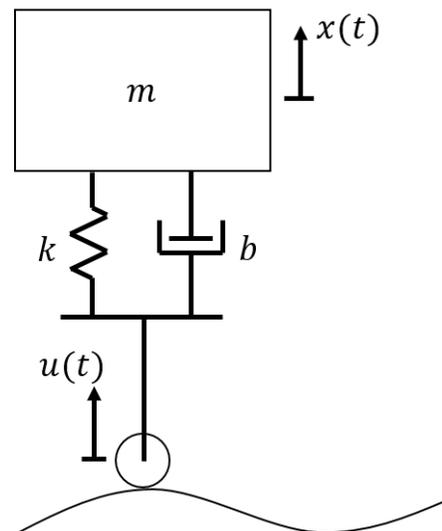


Figure 5. MSD oscillator.

Resistance-Inductance-Capacitance Circuit

$$\begin{bmatrix} \dot{i}(t) \\ \ddot{i}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} i(t) \\ \dot{i}(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [I(t)] \quad (3)$$

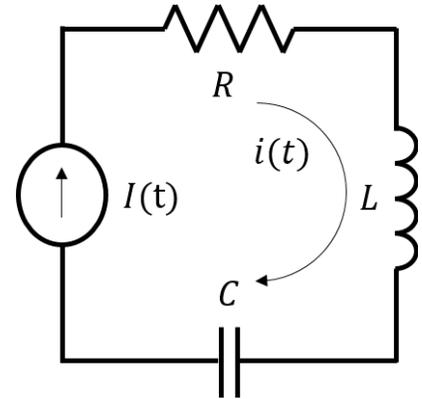


Figure 6. RLC circuit.

Large-Angle Pendulum

$$\begin{bmatrix} \theta_k \\ \dot{\theta}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 - \frac{b\Delta t}{ml^2} \end{bmatrix} \begin{bmatrix} \theta_{k-1} \\ \dot{\theta}_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{g}{l} \sin(\theta_{k-1} + \Delta t \dot{\theta}_{k-1}) \end{bmatrix} \quad (4)$$

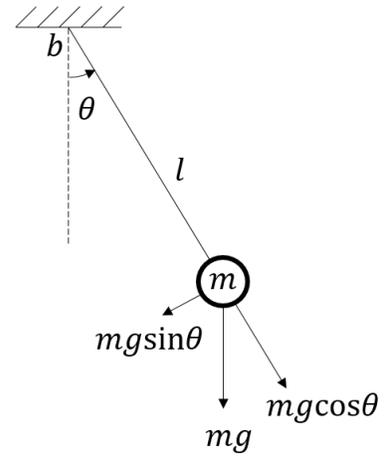


Figure 7. Nonlinear pendulum.

Gravitational Kinematics

$$\begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \\ y_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} - g\Delta t \begin{bmatrix} 0 \\ 0 \\ \Delta t \\ 1 \end{bmatrix} \quad (5)$$

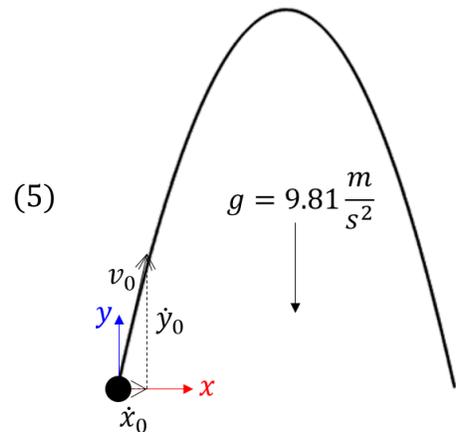


Figure 8. Gravitational kinematic path.

8.2.2 Robotics, Localization, and Target Tracking

A field analogous to that of state estimation is target localization and tracking. Within this area, the location of a physical target (whether that be a robot, vehicle, human, or the like) is tracked rather than the state of a system. While problems in state estimation consist largely of physical systems responding to natural inputs or initial conditions, target tracking and localization problems almost exclusively deal with position (whether 1-, 2-, or 3-D) and/or pose (which may also be represented by a 1-3 dimensional vector). Like state estimation, the location and pose of a target may be measured by a physical sensor (for example, a gyroscope, inertial measurement unit, or global positioning system), but other methods may also become available. For instance, a robotic field agent may track a target by implementing a computer vision algorithm, where observational uncertainty comes from any number of more abstract confounding factors beyond simple sensor noise.

For tracking and localization problems, mathematical models rely heavily on user input. While the plant dynamics of, say, an unmanned ground vehicle (UGV) may be governed by a discrete-time state-space equation, the macroscopic movement of interest is dominated by the control effort used to direct the vehicle. For example, consider the differential-drive UGV of Fig.

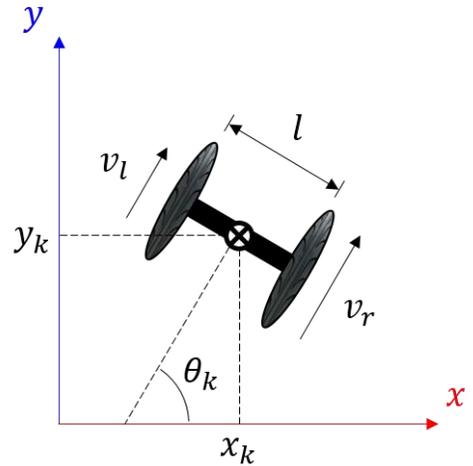


Figure 9. Differential drive mechanism.

9. This system is governed by the following 2-D kinematic state-space equation:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{(v_r - v_l)\Delta t}{l}\right) & -\sin\left(\frac{(v_r - v_l)\Delta t}{l}\right) & 0 \\ \sin\left(\frac{(v_r - v_l)\Delta t}{l}\right) & \cos\left(\frac{(v_r - v_l)\Delta t}{l}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \left(\frac{l(v_r + v_l)}{2(v_r - v_l)}\right) \sin \theta_{k-1} \\ -\left(\frac{l(v_r + v_l)}{2(v_r - v_l)}\right) \cos \theta_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} x_{k-1} - \left(\frac{l(v_r + v_l)}{2(v_r - v_l)}\right) \sin \theta_{k-1} \\ y_{k-1} + \left(\frac{l(v_r + v_l)}{2(v_r - v_l)}\right) \cos \theta_{k-1} \\ \frac{(v_r - v_l)\Delta t}{l} \end{bmatrix}. \quad (6)$$

As this equation shows, the position and heading $[x_k \ y_k \ \theta_k]^T$ at step k depend much more heavily on system inputs (the wheel speeds v_l and v_r) than on the prior “system state” (position and heading). Even if the dynamics of the vehicle and the mechatronics of the motors were included in the model, the large-scale motion of the UGV would dominate in most non-trivial circumstances. For this reason, it is important that user input and its uncertainty be well known for SEAM estimation to work well in most localization and tracking scenarios.

A special kind of motion where plant dynamics need not be modeled and user inputs can be unknown is when targets follow a well-defined path. One such example is the figure-8 pattern; such a path may be found in search-and-rescue contexts, among others (see Fig. 10). The proper mathematical term for a figure-8 is a lemniscate, which has a few strict mathematical definitions. One form of a lemniscate also belongs to a family of 2-D curves known as the Lissajous curves. Such a curve arises when the x- and y-components of a path follow simple harmonic motion. Because harmonic oscillation is easily described by a two dimensional state-space equation, the 2-D position a target following a Lissajous lemniscate can be obtained by the following state and output equations:

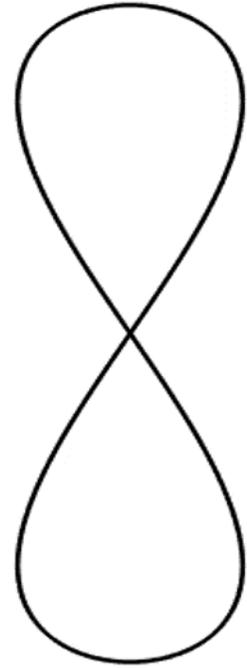


Figure 10. A lemniscate.

$$\begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_x^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_y^2 & 0 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \\ y_{k-1} \\ \dot{y}_{k-1} \end{bmatrix}, \begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \end{bmatrix}. \quad (7)$$

For the lemniscate of Fig. 10, $\omega_x = 2\omega_y$ and $[x_0 \ \dot{x}_0 \ y_0 \ \dot{y}_0]^T = [0 \ \omega_x \ 0 \ \omega_y]^T$. However, any Lissajous curve can in fact be described by the above equation, with the ratio of the x- and y-angular frequencies and initial conditions taking on different values. The dynamics and

inputs of a target following such a path can be eliminated by modeling and estimating only this deterministic linear macroscopic motion.

In the field of robotics, simultaneous localization and mapping (SLAM) is a well-known framework often used for localization problems. This framework accomplishes self-localization and environment mapping simultaneously by employing recursive Bayesian estimation (RBE) to overcome uncertainties in sensing and 3-D reconstruction. Because RBE is the basis for the estimation component of SEAM, integrating SEAM and SLAM is a natural extension of the work developed in this dissertation. Such a SLAM-SEAM framework would allow the location of a target or agent within an unfamiliar environment to be estimated with more accuracy than traditional methods if the model parameters pertaining to the target are not well known. Furthermore, because the accuracy of mapping is highly dependent on the accuracy of localization, the improvements afforded by SLAM-SEAM would also extend to mapping.

8.2.3 Conclusion

The research presented in this dissertation has a wide range of applications from state estimation and signal processing to target localization and tracking. If any discrete-time state-space model can be obtained which describes a system's state trajectory or the characteristics of a sensor used to observe it, the SEAM framework offers a robust solution for estimation when model parameters are poorly known and state belief may be non-Gaussian. Future work in this area is warranted, especially with regards to real-world validation of the framework in the field. Nevertheless, the extensive studies investigated in this dissertation offer strong support for the efficacy and promise of this research. SDG