

KALMAN FILTER

from the Ground Up



Alex Becker

First Edition

Revision history for the first edition

2023-05-01 First Release

2023-05-08 Minor Typo Updates

ISBN 978-965-598-439-2

Copyright © 2023 Alex Becker

KALMANFILTER.NET

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the author, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. The author will not be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

First edition, May 2023

The road to learning by precept is long, by example short and effective.

Lucius Annaeus Seneca

A philosopher of Ancient Rome

Preface

Introduction

The Kalman Filter algorithm is a powerful tool for estimating and predicting system states in the presence of uncertainty and is widely used as a fundamental component in applications such as target tracking, navigation, and control.

Although the Kalman Filter is a straightforward concept, many resources on the subject require extensive mathematical background and fail to provide practical examples and illustrations, making it more complicated than necessary.

Back in 2017, I created an online tutorial based on numerical examples and intuitive explanations to make the topic more accessible and understandable. The online tutorial provides introductory material covering the univariate (one-dimensional) and multivariate (multidimensional) Kalman Filters.

Over time, I have received many requests to include more advanced topics, such as non-linear Kalman Filters (Extended Kalman Filter and Unscented Kalman Filter), sensors fusion, and practical implementation guidelines.

Based on the material covered in the online tutorial, I authored the “*Kalman Filter from the Ground Up*” [e-book](#).

The original online tutorial will remain available for free access on the [KALMANFILTER.NET](#) website. The e-book “Kalman Filter from the Ground Up” and the source code for the numerical examples can be [purchased online](#).

The book takes the reader from the basics to the advanced topics, covering both theoretical concepts and practical applications. The writing style is intuitive, prioritizing clarity of ideas over mathematical rigor, and it approaches the topic from a philosophical perspective before delving into quantification.

The book contains many illustrative examples, including 14 fully solved numerical examples with performance plots and tables. Examples progress in a paced, logical manner and build upon each other.

The book also includes the necessary mathematical background, providing a solid foundation to expand your knowledge and help to overcome your math fears.

This book is the solution for those facing challenges with the Kalman Filter and the underlying math.

Upon finishing this book, you will be able to design, simulate, and evaluate the performance of the Kalman Filter.

The book includes four parts:

- **Part 1** serves as an introduction to the Kalman Filter, using eight numerical examples, and doesn't require any prior mathematical knowledge. You can call it "The Kalman Filter for Dummies," as it aims to provide an intuitive understanding and develop "Kalman Filter intuition." Upon completing Part 1, readers will thoroughly understand the Kalman Filter's concept and be able to design a univariate (one-dimensional) Kalman Filter.

This part is available for free access!

- **Part 2** presents the Kalman Filter in matrix notation, covering the multivariate (multidimensional) Kalman Filter. It includes a mathematical derivation of Kalman Filter equations, dynamic systems modeling, and two numerical examples. This section is more advanced and requires basic knowledge of Linear Algebra (only matrix operations). Upon completion, readers will understand the math behind the Kalman Filter and be able to design a multivariate Kalman Filter.

Most of this part is available for free access!

- **Part 3** is dedicated to the non-linear Kalman Filter, which is essential for mastering the Kalman Filter since most real-life systems are non-linear. This part begins with a problem statement and describes the differences between linear and non-linear systems. It includes derivation and examples of the most common non-linear filters: the Extended Kalman Filter and the Unscented Kalman Filter.
- **Part 4** contains practical guidelines for Kalman Filter implementation, including sensor fusion, variable measurement uncertainty, treatment of missing measurements, treatment of outliers, and the Kalman Filter design process.

About the author

My name is Alex Becker. I am from Israel. I am an engineer with over 20 years of experience in the wireless technologies field. As a part of my work, I had to deal with Kalman Filters, mainly for tracking applications.

Constructive criticism is always welcome. I would greatly appreciate your comments and suggestions. Please drop me an email (alex@kalmanfilter.net).



R The numerical examples in this book do not exemplify any modes, methodologies, techniques, or parameters employed by any operational system known to the author.

About the Kalman Filter

Many modern systems utilize multiple sensors to estimate hidden (unknown) states through a series of measurements. For instance, a GPS receiver can estimate location and velocity, where location and velocity represent the hidden states, while the differential time of the arrival of signals from satellites serves as measurements.

One of the biggest challenges of tracking and control systems is providing an accurate and precise estimation of the hidden states in the presence of uncertainty. For example, GPS receivers are subject to measurement uncertainties influenced by external factors, such as thermal noise, atmospheric effects, slight changes in satellite positions, receiver clock precision, and more.

The Kalman Filter is a widely used estimation algorithm that plays a critical role in many fields. It is designed to estimate the hidden states of the system, even when the measurements are imprecise and uncertain. Also, the Kalman Filter predicts the future system state based on past estimations.

The filter is named after Rudolf E. Kálmán (May 19, 1930 – July 2, 2016). In 1960, Kálmán published his famous paper describing a recursive solution to the discrete-data linear filtering problem [1].



Contents

I	Introduction to Kalman Filter	
1	The Necessity of Prediction	27
2	Essential background I	29
2.1	Mean and Expected Value	29
2.2	Variance and Standard deviation	30
2.3	Normal Distribution	33
2.4	Random Variables	35
2.5	Estimate, Accuracy and Precision	36
2.6	Summary	37
3	The $\alpha - \beta - \gamma$ filter	39
3.1	Example 1 – Weighting the gold	39
3.1.1	Estimation algorithm	44
3.1.2	The numerical example	44
3.1.3	Results analysis	48
3.1.4	Example summary	48
3.2	Example 2 – Tracking the constant velocity aircraft	49
3.2.1	The $\alpha - \beta$ filter	50
3.2.2	Estimation Algorithm	53
3.2.3	The numerical example	53
3.2.4	Results analysis	58
3.2.5	Example summary	59
3.3	Example 3 – Tracking accelerating aircraft	60
3.3.1	The numerical example	61
3.3.2	Results analysis	64
3.3.3	Example summary	65
3.4	Example 4 – Tracking accelerating aircraft using the $\alpha - \beta - \gamma$ filter	66
3.4.1	The $\alpha - \beta - \gamma$ filter	66

3.4.2	The numerical example	66
3.4.3	Results analysis	72
3.5	Summary of the $\alpha - \beta - (\gamma)$ filter	73
4	Kalman Filter in one dimension	75
4.1	One-dimensional Kalman Filter without process noise	75
4.1.1	Estimate as a random variable	77
4.1.2	Measurement as a random variable	77
4.1.3	State prediction	78
4.1.4	State update	80
4.1.5	Putting all together	84
4.1.6	Kalman Gain intuition	87
4.2	Example 5 – Estimating the height of a building	90
4.2.1	The numerical example	90
4.2.2	Results analysis	95
4.2.3	Example summary	98
5	Adding process noise	99
5.1	The complete model of the one-dimensional Kalman Filter	99
5.1.1	The Process Noise	99
5.2	Example 6 – Estimating the temperature of the liquid in a tank ..	101
5.2.1	The numerical example	101
5.2.2	Results analysis	106
5.2.3	Example summary	108
5.3	Example 7 – Estimating the temperature of a heating liquid I ...	109
5.3.1	The numerical example	109
5.3.2	Results analysis	114
5.3.3	Example summary	115
5.4	Example 8 – Estimating the temperature of a heating liquid II ..	116
5.4.1	The numerical example	116
5.4.2	Results analysis	120
5.4.3	Example summary	121

7	Essential background II	129
7.1	Matrix operations	129
7.2	Expectation algebra	129
7.2.1	Basic expectation rules	130
7.2.2	Variance and Covariance expectation rules	131
7.3	Multivariate Normal Distribution	136
7.3.1	Introduction	136
7.3.2	Covariance	136
7.3.3	Covariance matrix	140
7.3.4	Multivariate normal distribution	142
7.3.5	Bivariate normal distribution	143
7.3.6	Confidence intervals	144
7.3.7	Covariance ellipse	145
7.3.8	Confidence ellipse	148
8	Kalman Filter Equations Derivation	151
8.1	State Extrapolation Equation	151
8.1.1	Example - airplane - no control input	153
8.1.2	Example - airplane - with control input	155
8.1.3	Example – falling object	157
8.1.4	State extrapolation equation dimensions	158
8.1.5	Linear time-invariant systems	158
8.2	Covariance Extrapolation Equation	160
8.2.1	The estimate covariance without process noise	160
8.2.2	Constructing the process noise matrix Q	161
8.3	Measurement equation	167
8.3.1	The observation matrix	168
8.3.2	Measurement equation dimensions	169
8.4	Interim Summary	170
8.4.1	Prediction equations	171
8.4.2	Auxiliary equations	172
8.5	State Update Equation	174
8.5.1	State Update Equation dimensions	175
8.6	Covariance Update Equation	176
8.6.1	Covariance Update Equation Derivation	176

8.7	The Kalman Gain	179
8.7.1	Kalman Gain Equation Derivation	179
8.8	Simplified Covariance Update Equation	182
8.9	Summary	183
9	Multivariate KF Examples	187
9.1	Example 9 – vehicle location estimation	187
9.1.1	Kalman Filter equations	187
9.1.2	The numerical example	196
9.1.3	Example analysis	203
9.2	Example 10 – rocket altitude estimation	205
9.2.1	Kalman Filter equations	206
9.2.2	The numerical example	211
9.2.3	Example analysis	216

III

Non-linear Kalman Filters

10	Foreword	221
11	Essential background III	223
11.1	The square root of a matrix	223
11.2	Cholesky decomposition	223
12	Non-linearity problem	227
12.1	Example – linear system	227
12.2	Example – State-to-measurement non-linear relation	230
12.3	Example – Non-linear system dynamics	232
13	Extended Kalman Filter (EKF)	237
13.1	Analytic linearization	237
13.2	First-order Taylor series expansion	239
13.3	Uncertainty projection in one dimension	239
13.3.1	Example – linearization in a single dimension	240
13.4	Uncertainty projection in two dimensions	243

13.5	Multivariate uncertainty projection	245
13.5.1	Jacobian derivation example	246
13.6	EKF equations	249
13.6.1	The EKF observation matrix	249
13.6.2	The EKF state transition matrix	250
13.6.3	EKF equations summary	250
13.7	Example 11 – vehicle location estimation using radar	252
13.7.1	Kalman Filter equations	252
13.7.2	The numerical example	257
13.7.3	Example summary	266
13.8	Example 12 - estimating the pendulum angle	268
13.8.1	Kalman Filter equations	269
13.8.2	The numerical example	273
13.8.3	Example summary	279
13.9	Limitations of EKF	280
13.9.1	Linearization error - 2D example	280
14	Unscented Kalman Filter (UKF)	283
14.1	The Unscented Transform (UT)	284
14.1.1	Step 1 – sigma points selection	284
14.1.2	Step 2 – points propagation	289
14.1.3	Step 3 – compute sigma points weights	291
14.1.4	Step 4 - approximate the mean and covariance of the output distribution	291
14.1.5	Unscented Transform summary	295
14.2	The UKF algorithm - Predict Stage	297
14.3	Statistical linear regression	298
14.4	The UKF algorithm - Update Stage	300
14.4.1	State update	300
14.4.2	Kalman gain derivation	300
14.4.3	Covariance update equation	301
14.5	UKF update summary	304
14.6	UKF algorithm summary	305
14.7	Example 13 – vehicle location estimation using radar	306
14.7.1	The numerical example	307
14.7.2	Example summary	318

14.8	Sigma Point Algorithm Modification	320
14.9	Modified UKF algorithm summary	323
14.10	Example 14 - estimating the pendulum angle	324
14.10.1	The numerical example	325
14.10.2	Example summary	332

15	Non-linear filters comparison	333
-----------	--------------------------------------	------------

16	Conclusion	337
-----------	-------------------	------------

IV

Kalman Filter in practice

17	Sensors Fusion	341
17.1	Combining measurements in one dimension	341
17.2	Combining n measurements	344
17.3	Combining measurements in k dimensions	344
17.4	Sensor data fusion using Kalman filter	346
17.4.1	Method 1 – measurements fusion	346
17.4.2	Method 2 – state fusion	347
17.5	Multirate Kalman Filter	349
18	Variable measurement error	351
19	Treating missing measurements	353
20	Treating outliers	355
20.1	Identifying outliers	355
20.1.1	Unlikely or unusual measurements	356
20.1.2	High statistical distance	356
20.2	Impact of outliers	358
20.3	Treating outliers	360
21	Kalman Filter Initialization	363
21.1	Linear KF Initialization	363
21.2	Non-linear KF initialization	366

21.3	KF initialization techniques	369
-------------	-------------------------------------	------------

22 KF Development Process **371**

22.1	Kalman Filter Design	371
-------------	-----------------------------	------------

22.2	Simulation	373
-------------	-------------------	------------

22.2.1	Scenario Module	373
--------	-----------------	-----

22.2.2	Measurements Module	374
--------	---------------------	-----

22.2.3	Kalman Filter Module	374
--------	----------------------	-----

22.2.4	Analysis	374
--------	----------	-----

22.2.5	Performance Examination	375
--------	-------------------------	-----

V

Appendices

A The expectation of variance **379**

A.1	Expectation rules	379
------------	--------------------------	------------

A.2	The expectation of the variance	380
------------	--	------------

A.3	The expectation of the body displacement variance	381
------------	--	------------

B Confidence Interval **383**

B.1	Cumulative Probability	383
------------	-------------------------------	------------

B.2	Normal inverse cumulative distribution	387
------------	---	------------

B.3	Confidence interval	389
------------	----------------------------	------------

C Modeling linear dynamic systems **391**

C.1	Derivation of the state extrapolation equation	391
------------	---	------------

C.2	The state space representation	392
------------	---------------------------------------	------------

C.2.1	Example - constant velocity moving body	392
-------	---	-----

C.2.2	Modeling high-order dynamic systems	394
-------	-------------------------------------	-----

C.2.3	Example - constant acceleration moving body	396
-------	---	-----

C.2.4	Example - mass-spring-damper system	398
-------	-------------------------------------	-----

C.2.5	More examples	401
-------	---------------	-----

C.3	Solving the differential equation	401
------------	--	------------

C.3.1	Dynamic systems without input variable	401
-------	--	-----

C.3.2	Dynamic systems with an input variable	404
-------	--	-----

D	Derivative of matrix product trace	407
D.1	Statement 1	407
D.2	Statement 2	408
E	Pendulum motion simulation	411
F	Statistical Linear Regression	415
G	The product of univariate Gaussian PDFs	421
G.1	Product of two univariate Gaussian PDFs	421
G.2	Product of n univariate Gaussian PDFs	424
H	Product of multivariate Gaussian PDFs	427
H.1	Product of n multivariate Gaussian PDFs	427
H.2	Product of 2 multivariate Gaussian PDFs	429
	Appendices	377
	Bibliography	433
	Articles	433
	Books	434
	Index	435

List of Figures

1.1	Tracking radar.	27
2.1	Coins.	29
2.2	Man on scales.	30
2.3	Normal distribution PDFs.	34
2.4	Proportions of the normal distribution.	35
2.5	Accuracy and Precision.	36
2.6	Statistical view of measurement.	37
3.1	Gold Bars.	39
3.2	Measurements vs. True value.	40
3.3	Example Notation.	41
3.4	State Update Equation.	43
3.5	State Update Equation.	44
3.6	Example 1: Measurements vs. True value vs. Estimates.	48
3.7	1D scenario.	49
3.8	$\alpha - \beta$ filter estimation algorithm.	53
3.9	Measurements vs. True value vs. Estimates - low Alpha and Beta.	58
3.10	Measurements vs. True value vs. Estimates - high Alpha and Beta.	58
3.11	Constant Velocity Movement.	60
3.12	Accelerated Movement.	61
3.13	Example 3 - range vs. time.	64
3.14	Example 3 - velocity vs. time.	65
3.15	Example 4: Range vs. Time.	72
3.16	Example 4: Velocity vs. Time.	72
3.17	Example 4: Acceleration vs. Time.	73
4.1	Schematic description of the Kalman Filter algorithm.	76
4.2	Example 1: Measurements vs. True value vs. Estimates.	76
4.3	Measurements Probability Density Function.	78
4.4	State Prediction Illustration.	79
4.5	State Update Illustration.	80
4.6	Detailed description of the Kalman Filter algorithm.	86

4.7	High Kalman Gain.	88
4.8	Low Kalman Gain.	89
4.9	Estimating the building height.	90
4.10	Example 5: the Kalman Gain.	95
4.11	Example 5: True value, measured values and estimates.	96
4.12	High uncertainty.	97
4.13	Low uncertainty.	97
4.14	Normal uncertainty.	98
5.1	Estimating the liquid temperature.	101
5.2	Example 6 : true temperature vs. measurements	102
5.3	Example 6: the Kalman Gain.	107
5.4	Example 6: true value, measured values and estimates.	107
5.5	Example 7 : true temperature vs. measurements	109
5.6	Example 7: true value, measured values and estimates.	114
5.7	Example 7: 100 measurements.	115
5.8	Example 8: true value, measured values and estimates.	120
5.9	Example 8: the Kalman Gain.	120
6.1	Airplane in 3D.	126
7.1	Object on the x-y plane.	137
7.2	Examples of different measurement sets.	137
7.3	Bivariate Gaussian.	143
7.4	Univariate Gaussian.	144
7.5	Bivariate Gaussian with projection.	145
7.6	Covariance ellipse.	146
7.7	Confidence ellipse.	148
8.1	Kalman Filter Extrapolation.	152
8.2	Spring System.	153
8.3	Falling Object.	153
8.4	Amplifier.	159
8.5	Discrete Noise.	162
8.6	Continuous Noise.	166
8.7	Predict-Update Diagram.	183
8.8	The Kalman Filter Diagram.	183
9.1	Vehicle location estimation.	187

9.2	Vehicle trajectory.	196
9.3	Example 9: true value, measured values and estimates.	203
9.4	Example 9: true value, measured values and estimates - zoom.	204
9.5	Example 10: rocket altitude estimation.	205
9.6	Example 10: true value, measured values and estimates of the rocket altitude.	216
9.7	Example 10: true value, measured values and estimates of the rocket velocity.	216
12.1	Balloon altitude measurement using radar.	228
12.2	Linear System.	229
12.3	Balloon altitude measurement using optical sensor.	230
12.4	Non-linear System.	231
12.5	Pendulum.	232
13.1	Analytic Linearization.	237
13.2	Linearization	242
13.3	The tangent plane.	243
13.4	Vehicle location estimation using radar.	247
13.5	Vehicle location estimation using radar.	252
13.6	Vehicle trajectory.	257
13.7	Example 11: true value, measured values and estimates.	266
13.8	Example 11: true value, measured values and estimates - zoom.	267
13.9	Pendulum position measurement.	268
13.10	Pendulum true position and velocity.	273
13.11	Example 12: pendulum angle - true value, measured values and estimates.	279
13.12	Example 12: pendulum angular velocity - true value, measured values and estimates.	279
13.13	Linearization Error.	280
13.14	2D example.	281
13.15	EKF linearized covariance.	281
13.16	EKF vs. UKF linearized covariance.	282
14.1	1D RV Sigma Points.	286
14.2	2D RV Sigma Points.	289
14.3	1D RV Sigma Points propagation.	290
14.4	2D RV Sigma Points propagation.	291

14.5	1D RV Unscented Transform.	292
14.6	2D RV Unscented Transform.	295
14.7	UKF algorithm diagram.	305
14.8	Example 13: true value, measured values and estimates.	319
14.9	Example 13: true value, measured values and estimates - zoom.	319
14.10	α influence on the Sigma Points.	322
14.11	Modified UKF algorithm diagram.	323
14.12	Example 14: pendulum angle - true value, measured values and estimates.	332
14.13	Example 14: pendulum velocity - true value, measured values and estimates.	332
15.1	EKF and UKF absolute error of the vehicle position.	333
15.2	EKF and UKF estimations uncertainty of the vehicle position.	334
15.3	EKF and UKF absolute error of the pendulum angle and angular velocity.	334
15.4	EKF and UKF estimations uncertainty of the pendulum angle and angular velocity.	335
17.1	Two measurements PDF.	342
17.2	Two measurements fusion.	344
17.3	Two 2D measurements fusion.	345
17.4	2 Sensors measurements fusion.	346
17.5	Track-to-track fusion.	347
17.6	Multirate Kalman Filter.	349
20.1	Outlier example.	355
20.2	Low Mahalanobis distance.	357
20.3	High Mahalanobis distance.	357
20.4	Abnormal measurement with high uncertainty.	359
20.5	Abnormal measurement with low uncertainty.	360
21.1	LKF rough initiation.	363
21.2	LKF rough initiation: True vs. estimated position.	364
21.3	LKF fine initiation: True vs. estimated position.	365
21.4	LKF uncertainty: rough vs. fine initiation.	366
21.5	Non-linear KF fine initiation.	367
21.6	Non-linear KF rough initiation.	367
21.7	Non-linear KF very rough initiation.	368
21.8	Non-linear KF very rough initiation: filter performance.	368

22.1	KF development process.	371
22.2	KF simulation diagram.	373
22.3	KF simulation diagram.	376
B.1	Cumulative Probability.	383
B.2	Standard Normal Distribution.	385
B.3	z-score table.	385
B.4	Normal Inverse Cumulative Distribution.	388
B.5	z-score table.	388
B.6	Confidence interval.	390
C.1	The process of the state extrapolation equation derivation.	392
C.2	The constant acceleration model.	396
C.3	Mass-spring-damper model.	398
C.4	Mass-spring-damper forces.	399
E.1	Pendulum motion.	411

List of Tables

2.1	Players' heights.	31
2.2	Distance from the mean.	31
2.3	Squared distance from the mean.	32
3.1	Averaging equation.	42
3.2	Example 1 summary.	48
3.3	Example 2 summary.	57
3.4	Example 3 filter iterations.	62
3.5	Example 4 filter iterations.	69
4.1	Covariance update equation derivation.	83
4.2	Kalman Filter equations in one dimension.	85
4.3	Example 5 filter iterations.	93
5.1	Kalman Filter equations in one dimension with process noise.	100
5.2	Example 6 filter iterations.	104
5.3	Example 7 filter iterations.	111
5.4	Example 5 filter iterations.	117
7.1	Expectation rules.	130
7.2	Variance and covariance expectation rules.	131
7.3	Variance expectation rule.	132
7.4	Covariance expectation rule.	133
7.5	Variance square expectation rule.	134
7.6	Variance sum expectation rule.	135
7.7	Covariance equation.	138
7.8	Sample covariance equation.	139
8.1	Matrix dimensions of the state extrapolation equation.	158
8.2	Matrix dimensions of the measurement equation variables.	169
8.3	Matrix dimensions of the state update equation variables.	175
8.4	Equations for the Covariance Update Equation derivation.	176
8.5	Covariance Update Equation derivation.	177
8.6	Covariance Update Equation rearrange.	180

8.7	Kalman Gain Equation Derivation.	181
8.8	Equations for the Covariance Update Equation derivation.	182
8.9	Kalman Filter equations.	184
8.10	Kalman Filter notation.	185
9.1	Example 9 measurements.	198
9.2	Example 10 measurements.	212
13.1	Kalman Filter equations.	251
13.2	Example 11 measurements.	259
13.3	Example 12 measurements.	274
14.1	LKF and UKF predict stage equations.	298
14.2	Definitions.	299
14.3	Covariance Update Equation derivation.	301
14.4	LKF and UKF update stage equations.	304
14.5	Example 13 measurements.	307
14.6	Example 14 measurements.	325
A.1	Expectation rules.	379
A.2	Variance expectation rule.	380
A.3	Variance square expectation rule.	381
B.1	Cumulative distribution from z-score.	386
B.2	Cumulative distribution from μ and σ	387
B.3	Normal cumulative distribution.	389
D.1	Statements.	407
F.1	Definitions.	415
F.2	Expand the error equation.	416
F.3	Finding an optimal \mathbf{b}	417
F.4	Finding an optimal \mathbf{M}	417
F.5	Finding an optimal \mathbf{M} continued.	418
G.1	Exponent term.	422
G.2	Three Gaussian multiplication variance.	424
G.3	Three Gaussian multiplication variance.	425
H.1	Change the form of a multivariate Gaussian equation.	427
H.2	Reducing the number of the matrix inversions.	431

Acronyms

CPU	Central Processing Unit
EKF	Extended Kalman Filter
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
INS	Inertial Navigation System
KF	Kalman Filter
LiDAR	Light Detection and Ranging
LKF	Linear Kalman Filter
LTI	Linear Time Invariant
NASA	National Aeronautics and Space Administration
PDF	Probability Density Function
RMS	Root Mean Square
RMSE	Root Mean Square Error
SNR	Signal to Noise Ratio
SPKF	Sigma-point Kalman Filter
UAV	Unmanned Air Vehicle
UKF	Unscented Kalman Filter
UT	Unscented Transform



Introduction to Kalman Filter

1	The Necessity of Prediction	27
2	Essential background I	29
3	The $\alpha - \beta - \gamma$ filter	39
4	Kalman Filter in one dimension	75
5	Adding process noise	99

1. The Necessity of Prediction

Before delving into the Kalman Filter explanation, let us first understand the necessity of a tracking and prediction algorithm.

To illustrate this point, let's take the example of a tracking radar.

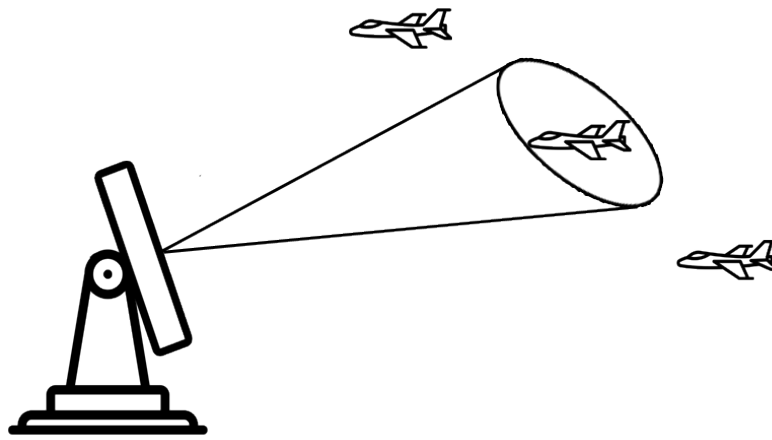


Figure 1.1: *Tracking radar.*

Suppose we have a track cycle of 5 seconds. At intervals of 5 seconds, the radar samples the target by directing a dedicated pencil beam.

Once the radar “visits” the target, it proceeds to estimate the current position and velocity of the target. The radar also estimates (or predicts) the target's position at the time of the next track beam.

The future target position can be easily calculated using Newton's motion equations:

$$x = x_0 + v_0\Delta t + \frac{1}{2}a\Delta t^2 \tag{1.1}$$

Where:

x is the target position

x_0 is the initial target position

v_0 is the initial target velocity

a is the target acceleration

Δt is the time interval (5 seconds in our example)

When dealing with three dimensions, Newton's motion equations can be expressed as a system of equations:

$$\begin{cases} x = x_0 + v_{x0}\Delta t + \frac{1}{2} a_x \Delta t^2 \\ y = y_0 + v_{y0}\Delta t + \frac{1}{2} a_y \Delta t^2 \\ z = z_0 + v_{z0}\Delta t + \frac{1}{2} a_z \Delta t^2 \end{cases} \quad (1.2)$$

The set of target parameters $[x, y, z, v_x, v_y, v_z, a_x, a_y, a_z]$ is known as the **System State**. The current state serves as the input for the prediction algorithm, while the algorithm's output is the future state, which includes the target parameters for the subsequent time interval.

The system of equations mentioned above is known as a **Dynamic Model** or **State Space Model**. The dynamic model describes the relationship between the input and output of the system.

Apparently, if the target's current state and dynamic model are known, predicting the target's subsequent state can be easily accomplished.

In reality, the radar measurement is not entirely accurate. It contains random errors or uncertainties that can affect the accuracy of the predicted target state. The magnitude of the errors depends on various factors, such as radar calibration, beam width, and signal-to-noise ratio of the returned echo. The random errors or uncertainties in the radar measurement are known as **Measurement Noise**.

In addition, the target motion is not always aligned with the motion equations due to external factors like wind, air turbulence, and pilot maneuvers. This misalignment between the motion equations and the actual target motion results in an error or uncertainty in the dynamic model, which is called **Process Noise**.

Due to the Measurement Noise and the Process Noise, the estimated target position can be far away from the actual target position. In this case, the radar might send the track beam in the wrong direction and miss the target.

In order to improve the radar's tracking accuracy, it is essential to employ a prediction algorithm that accounts for both process and measurement uncertainty.

The most common tracking and prediction algorithm is the **Kalman Filter**.

2. Essential background I

Before we start, I would like to explain several fundamental terms such as variance, standard deviation, normal distribution, estimate, accuracy, precision, mean, expected value, and random variable.

I expect that many readers of this book are familiar with introductory statistics. However, at the beginning of this book, I promised to supply the necessary background that is required to understand how the Kalman Filter works. If you are familiar with this topic, feel free to skip this chapter and jump to chapter 3.

2.1 Mean and Expected Value

Mean and **Expected Value** are closely related terms. However, there is a difference.

For example, given five coins – two 5-cent coins and three 10-cent coins, we can easily calculate the mean value by averaging the values of the coins.



Figure 2.1: *Coins.*

$$V_{mean} = \frac{1}{N} \sum_{n=1}^N V_n = \frac{1}{5} (5 + 5 + 10 + 10 + 10) = 8cent \quad (2.1)$$

The above outcome cannot be defined as the expected value because the system states (the coin values) are not hidden, and we've used the entire population (all 5 coins) for the mean value calculation.

Now assume five different weight measurements of the same person: 79.8kg, 80kg, 80.1kg, 79.8kg, and 80.2kg. The person is a system, and the person's weight is a system state.



Figure 2.2: *Man on scales.*

The measurements are different due to the random measurement error of the scales. We do not know the true value of the weight since it is a **Hidden State**. However, we can estimate the weight by averaging the scales' measurements.

$$W = \frac{1}{N} \sum_{n=1}^N W_n = \frac{1}{5} (79.8 + 80 + 80.1 + 79.8 + 80.2) = 79.98kg \quad (2.2)$$

The outcome of the estimate is the expected value of the weight.

The expected value is the value you would expect your hidden variable to have over a long time or many trials.

The mean is usually denoted by the Greek letter μ .

The letter E usually denotes the expected value.

2.2 Variance and Standard deviation

The **Variance** is a measure of the spreading of the data set from its mean.

The **Standard Deviation** is the square root of the variance.

The standard deviation is denoted by the Greek letter σ (sigma). Accordingly, the variance is denoted by σ^2 .

Suppose we want to compare the heights of two high school basketball teams. The following table provides the players' heights and the mean height of each team.

	Player 1	Player 2	Player 3	Player 4	Player 5	Mean
Team A	1.89m	2.1m	1.75m	1.98m	1.85m	1.914m
Team B	1.94m	1.9m	1.97m	1.89m	1.87m	1.914m

Table 2.1: *Players' heights.*

As we can see, the mean height of both teams is the same. Let us examine the height variance.

Since the variance measures the spreading of the data set, we would like to know the data set deviation from its mean. We can calculate the distance from the mean for each variable by subtracting the mean from each variable.

The height is denoted by x , and the heights mean by the Greek letter μ . The distance from the mean for each variable would be:

$$x_n - \mu = x_n - 1.914m \quad (2.3)$$

The following table presents the distance from the mean for each variable.

	Player 1	Player 2	Player 3	Player 4	Player 5
Team A	-0.024m	0.186m	-0.164m	0.066m	-0.064m
Team B	0.026m	-0.014m	0.056m	-0.024m	-0.044m

Table 2.2: *Distance from the mean.*

Some of the values are negative. To get rid of the negative values, let us square the distance from the mean:

$$(x_n - \mu)^2 = (x_n - 1.914m)^2 \quad (2.4)$$

The following table presents the squared distance from the mean for each variable.

	Player 1	Player 2	Player 3	Player 4	Player 5
Team A	0.000576m ²	0.034596m ²	0.026896m ²	0.004356m ²	0.004096m ²
Team B	0.000676m ²	0.000196m ²	0.003136m ²	0.000576m ²	0.001936m ²

Table 2.3: *Squared distance from the mean.*

To calculate the variance of the data set, we need to find the average value of all squared distances from the mean:

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \mu)^2 \quad (2.5)$$

For team A, the variance would be:

$$\begin{aligned} \sigma_A^2 &= \frac{1}{N} \sum_{n=1}^N (x_n - \mu)^2 \\ &= \frac{1}{5} (0.000576 + 0.034596 + 0.026896 + 0.004356 + 0.004096) = 0.014m^2 \end{aligned}$$

For team B, the variance would be:

$$\begin{aligned} \sigma_B^2 &= \frac{1}{N} \sum_{n=1}^N (x_n - \mu)^2 \\ &= \frac{1}{5} (0.000676 + 0.000196 + 0.003136 + 0.000576 + 0.001936) = 0.0013m^2 \end{aligned}$$

We can see that although the mean of both teams is the same, the measure of the height spreading of Team A is higher than the measure of the height spreading of Team B. Therefore, the Team A players are more diverse than the Team B players. There are players for different positions like ball handler, center, and guards, while the Team B players are not versatile.

The units of the variance are meters squared; it is more convenient to look at the standard deviation, which is a square root of the variance.

$$\sigma = \sqrt{\frac{1}{N} \sum_{n=1}^N (x_n - \mu)^2} \quad (2.6)$$

- The standard deviation of Team A players' heights would be 0.12m.
- The standard deviation of Team B players' heights would be 0.036m.

Now, assume that we would like to calculate the mean and variance of all basketball players in all high schools. That would be an arduous task - we would need to collect data on every player from every high school.

On the other hand, we can estimate the players' mean and variance by picking a big data set and making the calculations on this data set.

The data set of 100 randomly selected players should be sufficient for an accurate estimation.

However, when we estimate the variance, the equation for the variance calculation is slightly different. Instead of normalizing by the factor N , we shall normalize by the factor $N - 1$:

$$\sigma^2 = \frac{1}{N - 1} \sum_{n=1}^N (x_n - \mu)^2 \quad (2.7)$$

The factor of $N - 1$ is called Bessel's correction.

You can see the mathematical proof of the above equation on [visiondummy](#) or [Wikipedia](#).

2.3 Normal Distribution

It turns out that many natural phenomena follow the **Normal Distribution**. The normal distribution, also known as the **Gaussian** (named after the mathematician Carl Friedrich Gauss), is described by the following equation:

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (2.8)$$

The Gaussian curve is also called the **PDF (Probability Density Function)** for the normal distribution.

The following chart describes PDFs of the pizza delivery time in three cities: city 'A,' city 'B,' and city 'C.'

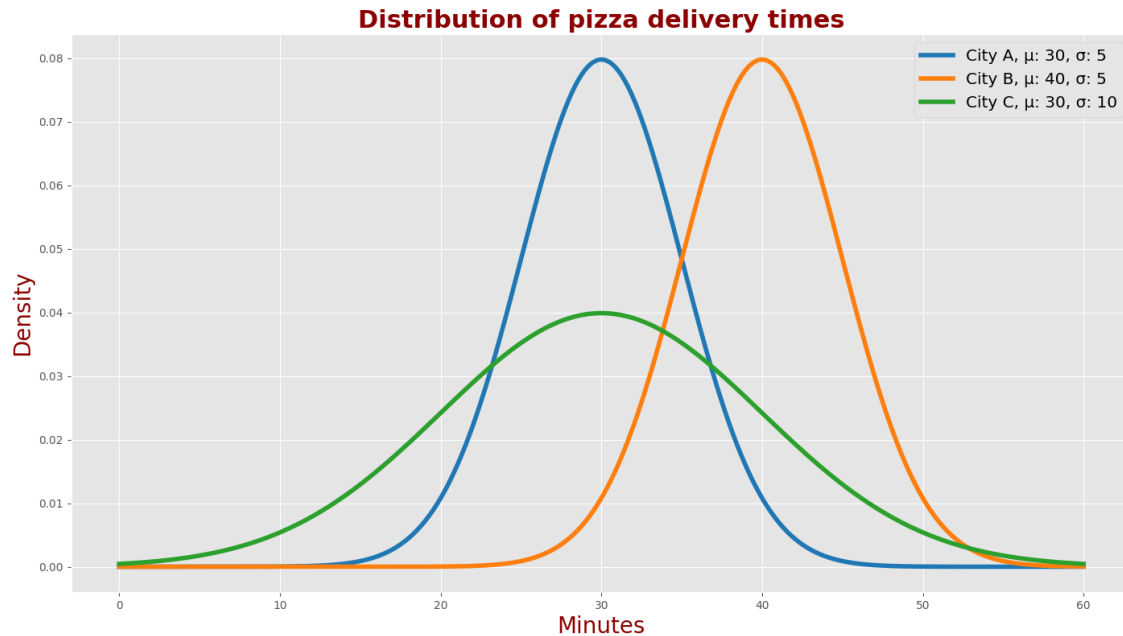


Figure 2.3: Normal distribution PDFs.

- In city 'A,' the mean delivery time is 30 minutes, and the standard deviation is 5 minutes.
- In city 'B,' the mean delivery time is 40 minutes, and the standard deviation is 5 minutes.
- In city 'C,' the mean delivery time is 30 minutes, and the standard deviation is 10 minutes.

We can see that the Gaussian shapes of the city 'A' and city 'B' pizza delivery times are identical; however, their centers are different. That means that in city 'A,' you wait for pizza for 10 minutes less on average, while the measure of spread in pizza delivery time is the same.

We can also see that the centers of Gaussians in the city 'A' and city 'C' are the same; however, their shapes are different. Therefore the average pizza delivery time in both cities is the same, but the measure of spread is different.

The following chart describes the proportions of the normal distribution.

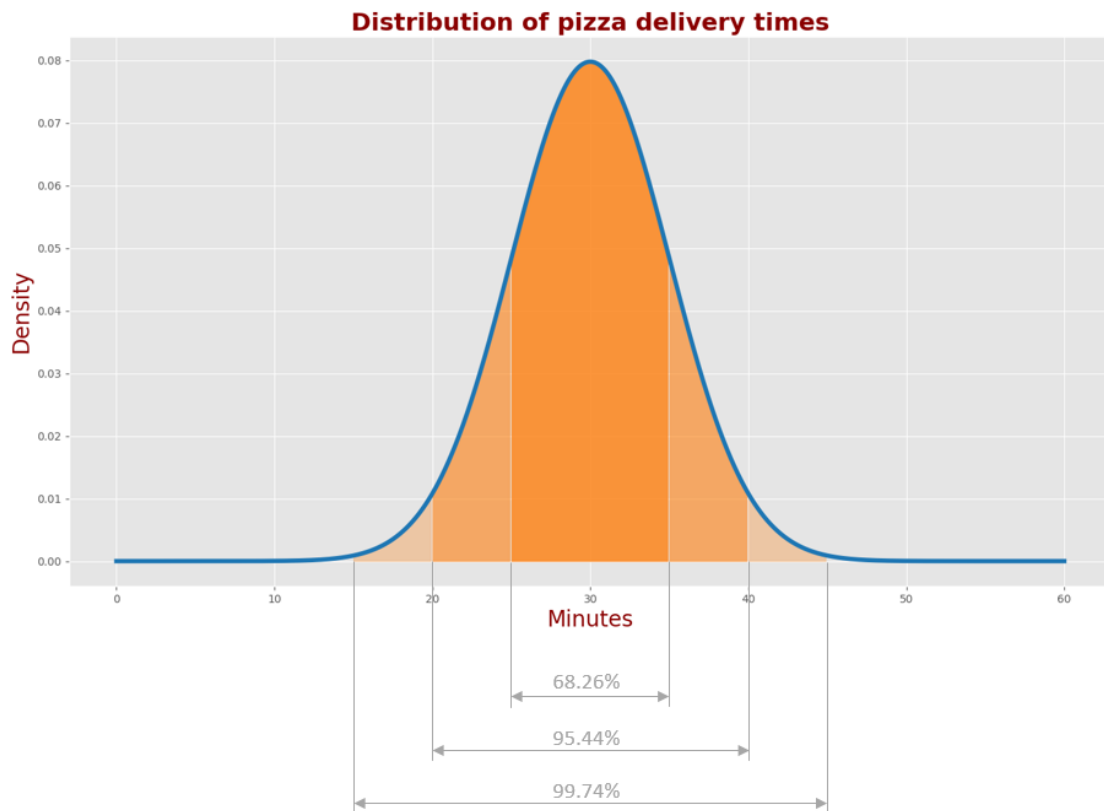


Figure 2.4: Proportions of the normal distribution.

- 68.26% of the pizza delivery times in City A lie within $\mu \pm \sigma$ range (25-35 minutes)
- 95.44% of the pizza delivery times in City A lie within $\mu \pm 2\sigma$ range (20-40 minutes)
- 99.74% of the pizza delivery times in City A lie within $\mu \pm 3\sigma$ range (15-45 minutes)

Usually, measurement errors are distributed normally. The Kalman Filter design assumes a normal distribution of the measurement errors.

2.4 Random Variables

A **random variable** describes the hidden state of the system. A random variable is a set of possible values from a random experiment.

The random variable can be continuous or discrete:

- A continuous random variable can take any value within a specific range, such as battery charge time or marathon race time.
- A discrete random variable is countable, such as the number of website visitors or the number of students in the class.

The random variable is described by the probability density function. The probability density function is characterized by **moments**.

The moments of the random value are expected values of powers of the random variable. We are interested in two types of moments:

- The k^{th} raw moment is the expected value of the k^{th} power of the random variable: $E(X^k)$.
- The k^{th} central moment is the expected value of the k^{th} power of the random variable distribution about its mean: $E((X - \mu_X)^k)$.

In this book, the random variables are characterized by the following:

- The first raw moment $E(X)$ – the mean of the sequence of measurements.
- The second central moment $E((X - \mu_X)^2)$ – the variance of the sequence of measurements.

2.5 Estimate, Accuracy and Precision

An **Estimate** is about evaluating the hidden state of the system. The true position of the aircraft is hidden from the observer. We can estimate the aircraft position using sensors, such as radar. The estimate can be significantly improved by using multiple sensors and applying advanced estimation and tracking algorithms (such as the Kalman Filter). Every measured or computed parameter is an estimate.

Accuracy indicates how close the measurement is to the true value.

Precision describes the variability in a series of measurements of the same parameter. Accuracy and precision form the basis of the estimate.

The following figure illustrates accuracy and precision.

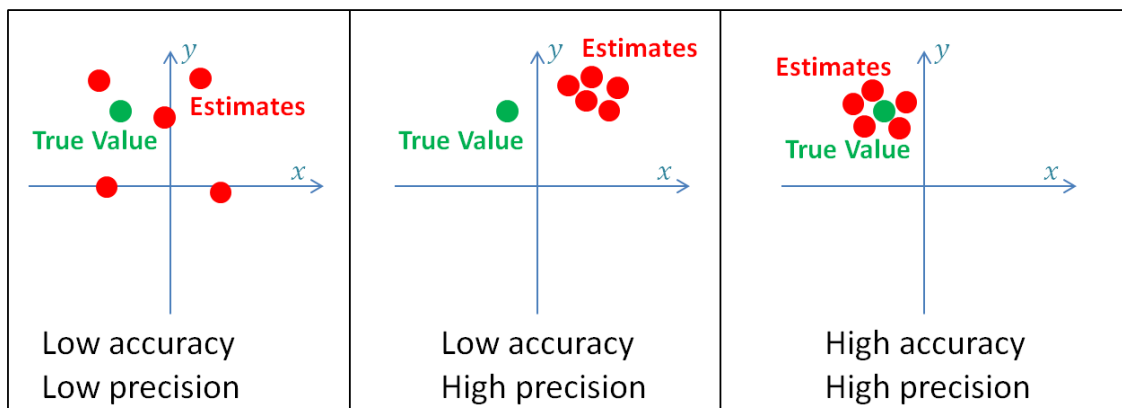


Figure 2.5: Accuracy and Precision.

High-precision systems have low variance in their measurements (i.e., low uncertainty), while low-precision systems have high variance in their measurements (i.e., high uncertainty). The random measurement error produces the variance.

Low-accuracy systems are called **biased** systems since their measurements have a built-in systematic error (bias).

The influence of the variance can be significantly reduced by averaging or smoothing measurements. For example, if we measure temperature using a thermometer with a random measurement error, we can make multiple measurements and average them. Since the error is random, some measurements would be above the true value and others below the true value. The estimate would be close to the true value. The more measurements we make, the closer the estimate will be.

On the other hand, a biased thermometer produces a constant systematic error in the estimate.

All examples in this book assume **unbiased** systems.

2.6 Summary

The following figure represents a statistical view of measurement.

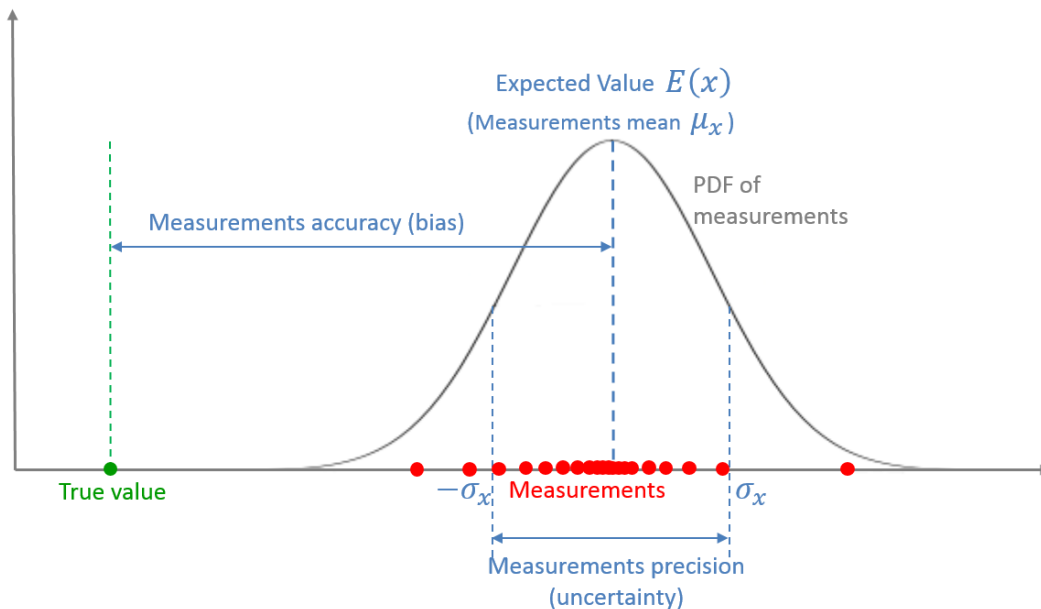


Figure 2.6: *Statistical view of measurement.*

A measurement is a **random variable** described by the **PDF**.

The mean of the measurements is the **Expected Value** of the random variable.

The offset between the mean of the measurements and the true value is the **accuracy of the measurements**, also known as **bias** or **systematic measurement error**.

The dispersion of the distribution is the measurement **precision**, also known as the **measurement noise**, **random measurement error**, or **measurement uncertainty**.

3. The $\alpha - \beta - \gamma$ filter

This chapter is introductory, and it describes the $\alpha - \beta$ and $\alpha - \beta - \gamma$ filters. These filters are frequently used for time series data smoothing. The principles of the $\alpha - \beta(-\gamma)$ filter are closely related to the Kalman Filter principles.

3.1 Example 1 – Weighting the gold

Now we are ready for the first simple example. In this example, we estimate the state of the static system. A static system is a system that doesn't change its state over a reasonable period. For instance, the static system could be a tower, and the state would be its height.

In this example, we estimate the weight of the gold bar. We have unbiased scales, i.e., the measurements don't have a systematic error, but the measurements do include random noise.



Figure 3.1: *Gold Bars.*

The system is the gold bar, and the system state is the weight of the gold bar. The dynamic model of the system is constant since we assume that the weight doesn't change over short periods.

To estimate the system state (i.e., the weight value), we can make multiple measurements and average them.

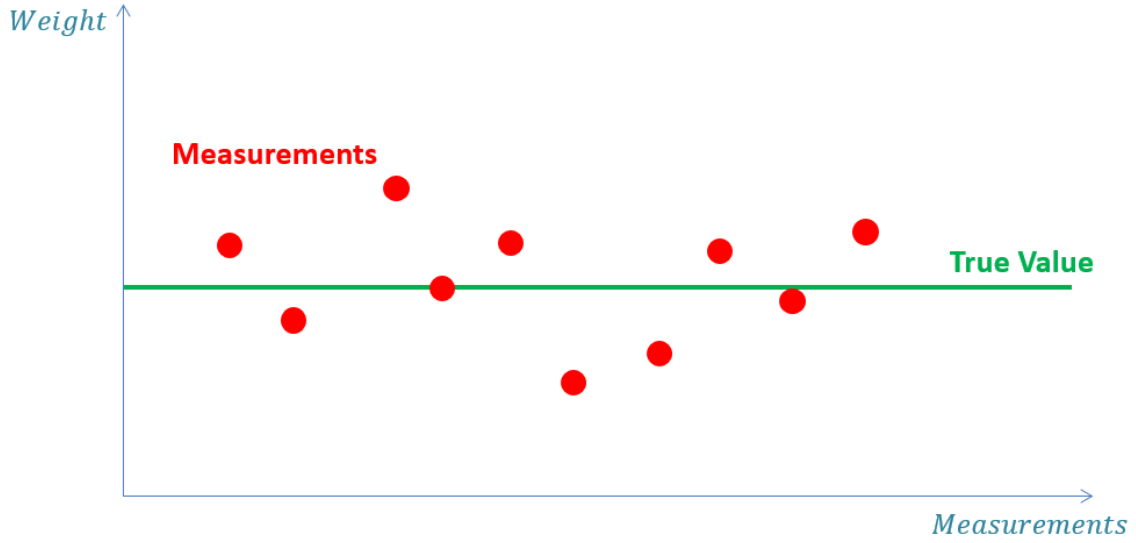


Figure 3.2: *Measurements vs. True value.*

At the time n , the estimate $\hat{x}_{n,n}$ would be the average of all previous measurements:

$$\hat{x}_{n,n} = \frac{1}{n} (z_1 + z_2 + \dots + z_{n-1} + z_n) = \frac{1}{n} \sum_{i=1}^n (z_i) \quad (3.1)$$

Example Notation:

x	is the true value of the weight
z_n	is the measured value of the weight at time n
$\hat{x}_{n,n}$	is the estimate of x at time n (the estimate is made after taking the measurement z_n)
$\hat{x}_{n+1,n}$	is the estimate of the future state $(n+1)$ of x . The estimate is made at the time n . In other words, $\hat{x}_{n+1,n}$ is a predicted state or extrapolated state
$\hat{x}_{n-1,n-1}$	is the estimate of x at time $n-1$ (the estimate is made after taking the measurement z_{n-1})
$\hat{x}_{n,n-1}$	is a prior prediction - the estimate of the state at time n . The prediction is made at the time $n-1$



In the literature, a caret (or hat) over a variable indicates an estimated value.

The dynamic model in this example is static (or constant) since the weight of gold doesn't change over time, therefore $\hat{x}_{n+1,n} = \hat{x}_{n,n}$.

Although the Equation 3.1 is mathematically correct, it is not practical for implementation. In order to estimate $\hat{x}_{n,n}$ we need to remember all historical measurements; therefore, we need a large memory. We also need to recalculate the average repeatedly

if we want to update the estimated value after every new measurement. Thus, we need a more powerful Central Processing Unit (CPU).

It would be more practical to keep the last estimate only ($\hat{x}_{n-1,n-1}$) and update it after every new measurement. The following figure exemplifies the required algorithm:

- Estimate the current state based on the measurement and prior prediction.
- Predict the next state based on the current state estimate using the Dynamic Model.

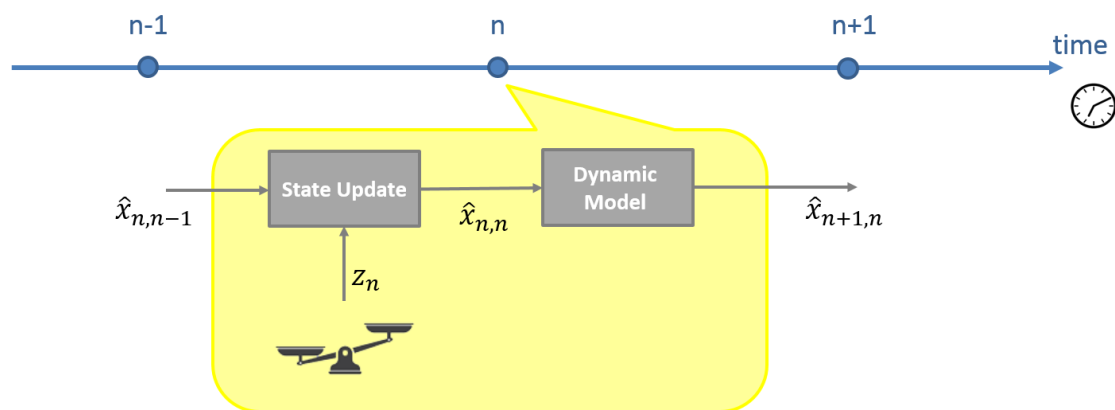


Figure 3.3: *Example Notation.*

We can modify the averaging equation for our needs using a small mathematical trick:

Equation	Notes
$\hat{x}_{n,n} = \frac{1}{n} \sum_{i=1}^n (z_i)$	Average formula: sum of n measurements divided by n
$= \frac{1}{n} \left(\sum_{i=1}^{n-1} (z_i) + z_n \right)$	Sum of the $n - 1$ measurements plus the last measurement divided by n
$= \frac{1}{n} \sum_{i=1}^{n-1} (z_i) + \frac{1}{n} z_n$	Expand
$= \frac{1}{n} \frac{n-1}{n-1} \sum_{i=1}^{n-1} (z_i) + \frac{1}{n} z_n$	Multiply and divide by term $n - 1$
$= \frac{n-1}{n} \frac{1}{n-1} \sum_{i=1}^{n-1} (z_i) + \frac{1}{n} z_n$	Reorder. The 'orange' term is the prior estimate
$= \frac{n-1}{n} \hat{x}_{n-1,n-1} + \frac{1}{n} z_n$	Rewriting the sum
$= \hat{x}_{n-1,n-1} - \frac{1}{n} \hat{x}_{n-1,n-1} + \frac{1}{n} z_n$	Distribute the term $\frac{n-1}{n}$
$= \hat{x}_{n-1,n-1} + \frac{1}{n} (z_n - \hat{x}_{n-1,n-1})$	Reorder

Table 3.1: *Averaging equation.*

$\hat{x}_{n-1,n-1}$ is the estimated state of x at the time $n - 1$, based on the measurement at the time $n - 1$.

Let's find $\hat{x}_{n,n-1}$ (the predicted state of x at the time n), based on $\hat{x}_{n-1,n-1}$ (the estimation at the time $n - 1$). In other words, we would like to extrapolate $\hat{x}_{n-1,n-1}$ to the time n .

Since the dynamic model in this example is static, the predicted state of x equals the estimated state of x : $\hat{x}_{n,n-1} = \hat{x}_{n-1,n-1}$.

Based on the above, we can write the **State Update Equation**:

State Update Equation

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \frac{1}{n} (z_n - \hat{x}_{n,n-1}) \quad (3.2)$$

The State Update Equation is one of the five Kalman filter equations. It means the following:

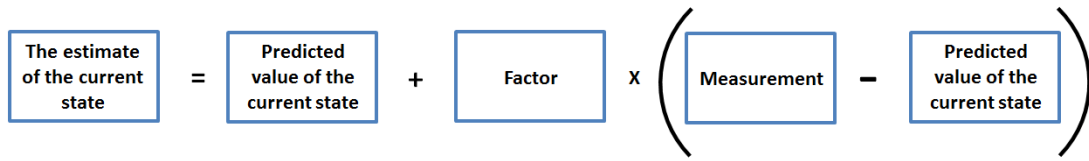


Figure 3.4: *State Update Equation.*

The factor $1/n$ is specific to our example. We will discuss the vital role of this factor later, but right now, I would like to note that in “Kalman Filter language,” this factor is called the **Kalman Gain**. It is denoted by K_n . The subscript n indicates that the Kalman Gain can change with every iteration.

The discovery of K_n was one of Rudolf Kalman’s significant contributions.

Before we get into the guts of the Kalman Filter, we use the Greek letter α_n instead of K_n .

So, the State Update Equation looks as follows:

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \alpha_n (z_n - \hat{x}_{n,n-1}) \quad (3.3)$$

The term $(z_n - \hat{x}_{n,n-1})$ is the “measurement residual,” also called **innovation**. The innovation contains new information.

In this example, $1/n$ decreases as n increases. In the beginning, we don’t have enough information about the current state; thus, the first estimation is based on the first measurement $\frac{1}{n}|_{n=1} = 1$. As we continue, each successive measurement has less weight in the estimation process, since $1/n$ decreases. At some point, the contribution of the new measurements will become negligible.

Let’s continue with the example. Before we make the first measurement, we can guess (or rough estimate) the gold bar weight simply by reading the stamp on the gold bar. It is called the **Initial Guess**, and it is our first estimate.

The Kalman Filter requires the initial guess as a preset, which can be very rough.

3.1.1 Estimation algorithm

The following chart depicts the estimation algorithm that is used in this example.

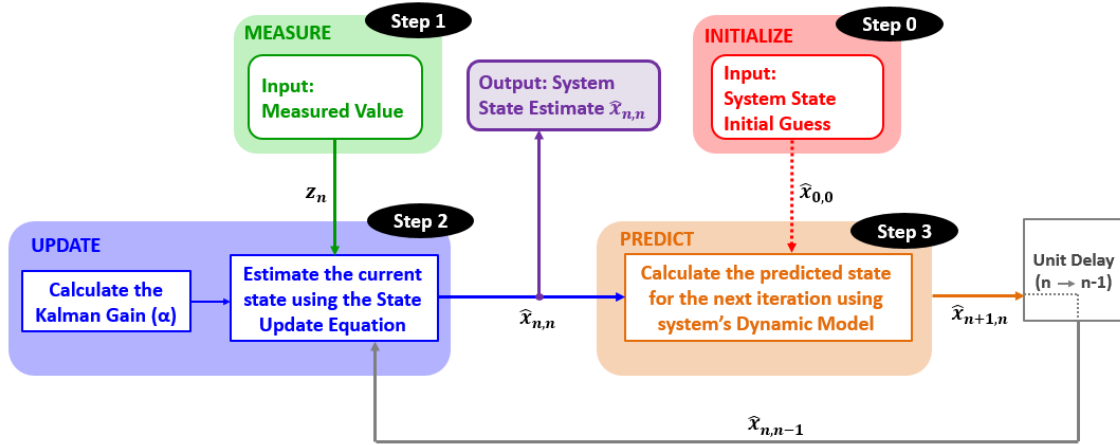


Figure 3.5: State Update Equation.

Now we are ready to start the measurement and estimation process.

3.1.2 The numerical example

3.1.2.1 Iteration Zero

Initialization

Our initial guess of the gold bar weight is 1000 grams. The initial guess is used only once for the filter initiation. Thus, it won't be required for successive iterations.

$$\hat{x}_{0,0} = 1000g$$

Prediction

The weight of the gold bar is not supposed to change. Therefore, the dynamic model of the system is static. Our next state estimate (prediction) equals the initialization:

$$\hat{x}_{1,0} = \hat{x}_{0,0} = 1000g$$

3.1.2.2 First Iteration

Step 1

Making the weight measurement with the scales:

$$z_1 = 996g$$

Step 2

Calculating the gain. In our example $\alpha_n = 1/n$, thus:

$$\alpha_1 = \frac{1}{1} = 1$$

Calculating the current estimate using the State Update Equation:

$$\hat{x}_{1,1} = \hat{x}_{1,0} + \alpha_1 (z_1 - \hat{x}_{1,0}) = 1000 + 1 (996 - 1000) = 996g$$



The initial guess could be any number in this specific example. Since $\alpha_1 = 1$, the initial guess is eliminated in the first iteration.

Step 3

The dynamic model of the system is static; thus, the weight of the gold bar is not supposed to change. Our next state estimate (prediction) equals to current state estimate:

$$\hat{x}_{2,1} = \hat{x}_{1,1} = 996g$$

3.1.2.3 Second Iteration

After a unit time delay, the **predicted estimate** from the previous iteration becomes the **prior estimate** in the current iteration:

$$\hat{x}_{2,1} = 996g$$

Step 1

Making the second measurement of the weight:

$$z_2 = 994g$$

Step 2

Calculating the gain:

$$\alpha_2 = \frac{1}{2}$$

Calculating the current estimate:

$$\hat{x}_{2,2} = \hat{x}_{2,1} + \alpha_2 (z_2 - \hat{x}_{2,1}) = 996 + \frac{1}{2}(994 - 996) = 995g$$

Step 3

$$\hat{x}_{3,2} = \hat{x}_{2,2} = 995g$$

3.1.2.4 Third Iteration

$$z_3 = 1021g$$

$$\alpha_3 = \frac{1}{3}$$

$$\hat{x}_{3,3} = 995 + \frac{1}{3}(1021 - 995) = 1003.67g$$

$$\hat{x}_{4,3} = 1003.67g$$

3.1.2.5 Fourth Iteration

$$z_4 = 1000g$$

$$\alpha_4 = \frac{1}{4}$$

$$\hat{x}_{4,4} = 1003.67 + \frac{1}{4}(1000 - 1003.67) = 1002.75g$$

$$\hat{x}_{5,4} = 1002.75g$$

3.1.2.6 Fifth Iteration

$$z_5 = 1002g$$

$$\alpha_5 = \frac{1}{5}$$

$$\hat{x}_{5,5} = 1002.75 + \frac{1}{5}(1002 - 1002.75) = 1002.6g$$

$$\hat{x}_{6,5} = 1002.6g$$

3.1.2.7 Sixth Iteration

$$z_6 = 1010g$$

$$\alpha_6 = \frac{1}{6}$$

$$\hat{x}_{6,6} = 1002.6 + \frac{1}{6}(1010 - 1002.6) = 1003.83$$

$$\hat{x}_{7,6} = 1003.83g$$

3.1.2.8 Seventh Iteration

$$z_7 = 983g$$

$$\alpha_7 = \frac{1}{7}$$

$$\hat{x}_{7,7} = 1003.83 + \frac{1}{7}(983 - 1003.83) = 1000.86g$$

$$\hat{x}_{8,7} = 1000.86g$$

3.1.2.9 Eighth Iteration

$$z_8 = 971g$$

$$\alpha_8 = \frac{1}{8}$$

$$\hat{x}_{8,8} = 1000.86 + \frac{1}{8}(971 - 1000.86) = 997.125g$$

$$\hat{x}_{9,8} = 997.125g$$

3.1.2.10 Ninth Iteration

$$z_9 = 993g$$

$$\alpha_9 = \frac{1}{9}$$

$$\hat{x}_{9,9} = 997.125 + \frac{1}{9}(993 - 997.125) = 996.67g$$

$$\hat{x}_{10,9} = 996.67g$$

3.1.2.11 Tenth Iteration

$$z_{10} = 1023g$$

$$\alpha_{10} = \frac{1}{10}$$

$$\hat{x}_{10,10} = 996.67 + \frac{1}{10}(1023 - 996.67) = 999.3g$$

$$\hat{x}_{11,10} = 999.3g$$

We can stop here. The gain decreases with each measurement. Therefore, the contribution of each successive measurement is lower than the contribution of the previous measurement. We get pretty close to the true weight, which is 1000g. If we were making more measurements, we would get closer to the true value.

The following table summarizes our measurements and estimates, and the chart compares the measured values, the estimates, and the true value.

n	1	2	3	4	5	6	7	8	9	10
α_n	1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}
z_n	996	994	1021	1000	1002	1010	983	971	993	1023
$\hat{x}_{n,n}$	996	995	1003.67	1002.75	1002.6	1003.83	1000.86	997.125	996.67	999.3
$\hat{x}_{n+1,n}$	996	995	1003.67	1002.75	1002.6	1003.83	1000.86	997.125	996.67	999.3

Table 3.2: *Example 1 summary.*

3.1.3 Results analysis

The following chart compares the true, measured, and estimated values.

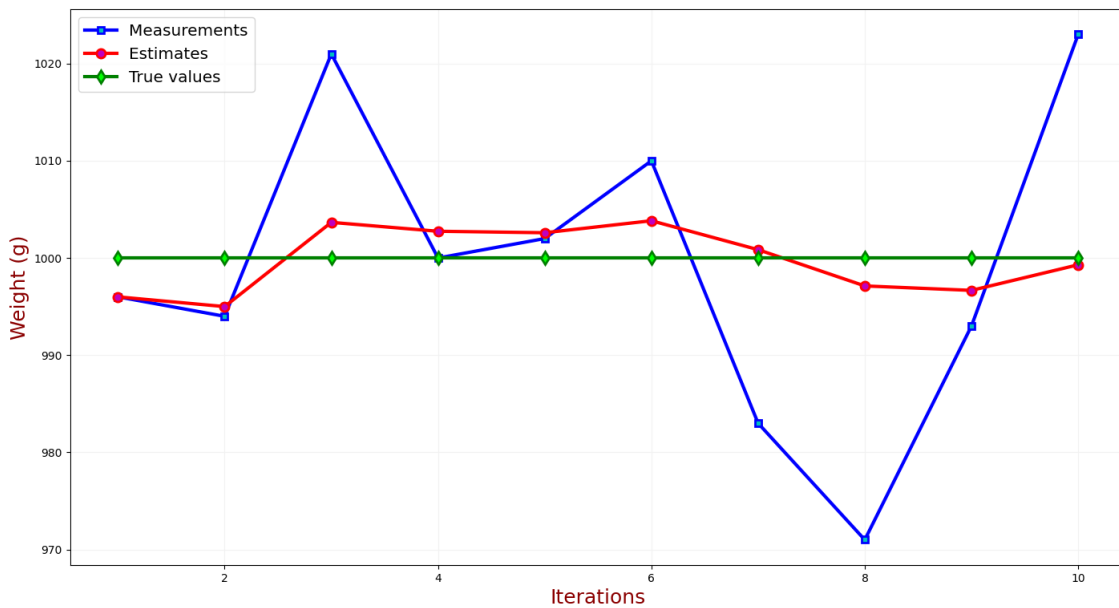


Figure 3.6: *Example 1: Measurements vs. True value vs. Estimates.*

The estimation algorithm has a smoothing effect on the measurements and converges toward the true value.

3.1.4 Example summary

In this example, we've developed a simple estimation algorithm for a static system. We have also derived the state update equation, one of the five Kalman Filter equations. We will revise the state update equation in subsection 4.1.4.

Bibliography

Articles

- [1] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Trans. of the ASME - Journal of Basic Engineering* (1960), pages 35–45 (cited on page 5).
- [2] Tarunraj Singh Dirk Tenne. “Optimal design of $\alpha - \beta - (\gamma)$ filters”. In: *Proceedings of the American Control Conference* 6 6 (Feb. 2000), pages 4348–4352. DOI: [10.1109/ACC.2000.877043](https://doi.org/10.1109/ACC.2000.877043) (cited on page 73).
- [3] P. R. Kalata. “A generalized parameter for $\alpha - \beta$ and $\alpha - \beta - \gamma$ target trackers”. In: *The 22nd IEEE Conference on Decision and Control* (Dec. 1983). DOI: [10.1109/CDC.1983.269580](https://doi.org/10.1109/CDC.1983.269580) (cited on page 73).
- [4] Wayne E. Hoover. “Algorithms For Confidence Circles and Ellipses”. In: *NOAA Technical Report NOS 107 CGS 3* (Sept. 1984) (cited on page 148).
- [5] L. Campo Y. Bar-Shalom. “The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance”. In: *Advances in Control Systems* 3 (1966), pages 293–340. DOI: <https://doi.org/10.1016/B978-1-4831-6716-9.50011-4> (cited on page 237).
- [6] S. F. Schmidt L. A. Mcgee. “Discovery of the Kalman filter as a practical tool for aerospace and industry”. In: *ech. Rep., NASA-TM-86847* (1985). DOI: [10.1109/TAES.1986.310815](https://doi.org/10.1109/TAES.1986.310815) (cited on page 237).
- [7] Kai O. Arras. “An Introduction To Error Propagation: Derivation, Meaning and Examples of Equation $C_y = F_x C_x F_x^T$ ”. In: *EPFL-ASL-TR-98-01 R3* (1998). DOI: <https://doi.org/10.3929/ethz-a-010113668> (cited on page 246).
- [8] Jeffrey K. Uhlmann Simon J. Julier. “New extension of the Kalman filter to nonlinear systems”. In: *Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI* (July 1997). DOI: <https://doi.org/10.1117/12.280797> (cited on pages 283, 284, 337).
- [9] Eric A. Wan Rudolph Van Der Merwe. “Sigma-point kalman filters for probabilistic inference in dynamic state-space models”. In: *Oregon Health and Science University* (2004). DOI: <https://doi.org/10.6083/M4Z60KZ5> (cited on pages 285, 337).
- [10] H.F. Durrant-Whyte S. Julier J. Uhlmann. “A new method for the nonlinear transformation of means and covariances in filters and estimators”. In: *IEEE*

- Transactions on Automatic Control* 45.3 (Mar. 2000), pages 477–482. DOI: [10.1109/9.847726](https://doi.org/10.1109/9.847726) (cited on page 298).
- [11] Herman Bruyninckx Tine Lefebvre and Joris De Schutter. “Comment on A new method for the nonlinear transformation of means and covariances in filters and estimators”. In: *IEEE Transactions on Automatic Control* 47.8 (Aug. 2002), pages 1406–1408. DOI: [10.1109/TAC.2002.800742](https://doi.org/10.1109/TAC.2002.800742) (cited on page 298).
- [12] Rudolph van der Merwe Eric A. Wan. “The Unscented Kalman Filter for Nonlinear Estimation”. In: *IEEE Proceedings of IEEE 2000 adaptive systems for signal processing, communication and control symposium* (Oct. 2000). DOI: <https://doi.org/10.3390/s21020438> (cited on page 320).
- [13] René van de Molengraft Jos Elfring Elena Torta. “Particle Filters: A Hands-On Tutorial”. In: *Sensors* 2021, 21(2) (2021), page 438 (cited on page 337).
- [14] Y. Bar-Shalom K. C. Chang R. K. Saha. “On optimal track-to-track fusion”. In: *IEEE Transactions on Aerospace and Electronic Systems* 33.8 (Oct. 1997), pages 1271–1276. DOI: [doi:10.1109/7.625124](https://doi.org/10.1109/7.625124) (cited on page 348).
- [15] L. Campo Y. Bar-Shalom. “The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance”. In: *IEEE Transactions on Aerospace and Electronic Systems* AES-22.6 (Nov. 1986), pages 803–805. DOI: [10.1109/TAES.1986.310815](https://doi.org/10.1109/TAES.1986.310815) (cited on page 348).
- [16] P.A. Bromiley. “Products and Convolutions of Gaussian Probability Density Functions”. In: *Internal Report* (Aug. 2014) (cited on pages 421, 427).

Books

- [17] Joseph P.D. Bucy R.S. *Filtering for Stochastic Processes with Applications to Guidance, Chapter 16*. Interscience, New York, 1968 (cited on page 182).
- [18] Uhlmann Jeffrey. *Dynamic map building and localization: new theoretical foundations, Thesis (Ph.D.)*. University of Oxford, 1995 (cited on pages 283, 320).
- [19] Tom M. Apostol. *Calculus, theorem 8.3*. John Wiley and Sons, Inc, 1967 (cited on page 404).

Index

Symbols

$\alpha - \beta$ filter 49
 $\alpha - \beta$ track filtering equations 52
 $\alpha - \beta$ track update equations 52
 $\alpha - \beta - \gamma$ filter 66

A

Accuracy 36
analytic linearization 221
approximated models 221

B

Bias error 65
biased 37

C

Cholesky decomposition 223
Cholesky factorization 223
control input 205
control matrix 151
control variable 151
Covariance Extrapolation Equation 99,
160
covariance matrix 136
Covariance Update Equation 83
cumulative probability 383

D

Dynamic error 65
Dynamic Model 28

E

elliptical scale factor 148
Estimate 36
estimate error 77
Expected Value 29

G

g-h filter 52
g-h-k filter 66
Gaussian 33
governing equation 394
ground truth 374

H

Hidden State 30

I

Initial Guess 43
innovation 43, 175
input transition matrix 151
input variable 151

K

Kalman Filter 28
Kalman Gain 43, 82
Kalman Gain Equation 82

L

lag error 65, 114
Linear Approximation 221
Linear systems 158

Linear Time-Invariant 158
 linearization error 280

M

Mahalanobis distance 356
 Mean 29
 measurement error 77
 Measurement Noise 28
 measurement uncertainty 38, 77
 moments 36
 Monte-Carlo method 337
 multivariate normal distribution... 136
 multivariate random variable 136

N

Normal Distribution 33

O

observation matrix... 168, 172, 193, 209
 optimal filter 80, 180, 221

P

Particle Filter 337
 PDF (Probability Density Function) 33
 Precision 36
 predicted estimate 45
 Prediction Equation 50
 Predictor Covariance Equation 160
 prior estimate 45
 Process Noise 28, 99
 Process Noise Variance 99

R

random measurement error 38
 random variable 35

S

Sigma point transform 285

Sigma Points 284
 Standard Deviation 30
 standard normal distribution 384
 standardized score 384
 State Extrapolation Equation 50
 state extrapolation equation 151
 State Space Model 28
 state space representation 391
 state transition matrix 151, 401
 State Update Equation 42
 state vector 125
 statistical distance 284
 statistical linear regression 298
 statistical linearization 221
 sub-optimal 221
 system dynamics matrix 401
 system function 159
 System State 28
 Systematic error 65

T

time-invariant system 159
 trace 180
 track-to-track 347
 Transition Equation 50
 Truncation error 65

U

unbiased 37
 uncertainty 77
 Unscented Transform 284

V

Variance 30

Z

z-score 384