

Kalman Filter book using Jupyter Notebook. Focuses on building intuition and experience, not formal proofs. Includes Kalman filters,extended Kalman filters, unscented Kalman filters, particle filters, and more. All exercises include solutions.

 1,217 commits

 4 branches

 0 releases

 60 contributors

Branch: master ▾





































New pull request

Create new file

Upload files

Find file

Clone or download ▾

 rlabbe	Fixed link to table of contents. #238	...	Latest commit 47cbc4b 3 days ago
 Supporting_Notebooks	Text used the wrong parameter name for hist() call, and I misattributed		5 months ago
 animations	Weights should be reset to 1/N after resampling.		a year ago
 experiments	moved test functions to correct file.		26 days ago
 figs	made gh_estimate3 consistent with text		2 years ago
 kf_book	moved test functions to correct file.		26 days ago
 pdf	Fixed chapter numbering		2 months ago
 .gitignore	Disable interactive plots for PDF.		2 years ago
 00-Preface.ipynb	Test fix for link to nbviewer. #238		3 days ago
 01-g-h-filter.ipynb	Fixed link to table of contents. #238		3 days ago
 02-Discrete-Bayes.ipynb	Fixed link to table of contents. #238		3 days ago
 03-Gaussians.ipynb	Fixed link to table of contents. #238		3 days ago
 04-One-Dimensional-Kalman-Filt...	Fixed link to table of contents. #238		3 days ago
 05-Multivariate-Gaussians.ipynb	Fixed link to table of contents. #238		3 days ago
 06-Multivariate-Kalman-Filters.ipy...	Fixed link to table of contents. #238		3 days ago
 07-Kalman-Filter-Math.ipynb	Fixed link to table of contents. #238		3 days ago
 08-Designing-Kalman-Filters.ipynb	Fixed link to table of contents. #238		3 days ago
 09-Nonlinear-Filtering.ipynb	Fixed link to table of contents. #238		3 days ago
 10-Unscented-Kalman-Filter.ipynb	Fixed link to table of contents. #238		3 days ago
 11-Extended-Kalman-Filters.ipynb	Fixed link to table of contents. #238		3 days ago
 12-Particle-Filters.ipynb	Fixed link to table of contents. #238		3 days ago
 13-Smoothing.ipynb	Fixed link to table of contents. #238		3 days ago
 14-Adaptive-Filtering.ipynb	Fixed link to table of contents. #238		3 days ago
 Appendix-A-Installation.ipynb	Fixed link to table of contents. #238		3 days ago
 Appendix-B-Symbols-and-Notati...	Fixed link to table of contents. #238		3 days ago
 Appendix-D-HInfinity-Filters.ipynb	Fixed link to table of contents. #238		3 days ago
 Appendix-E-Ensemble-Kalman-Fil...	Fixed link to table of contents. #238		3 days ago
 Appendix-G-Designing-Nonlinear...	Fixed link to table of contents. #238		3 days ago
 Appendix-H-Least-Squares-Filter...	Fixed link to table of contents. #238		3 days ago
 Appendix-I-Analytic-Evaluation-of...	Fixed link to table of contents. #238		3 days ago
 README.md	Added instructions for cloning the book		2 months ago
 __init__.py	Started altering to use filterpy project.		4 years ago
 book_format.py	Updated for FilterPy 1.4.0		2 months ago
 environment.yml	Setup conda environment and update readme		6 months ago
 license.html	Fixed CC License		3 years ago
 requirements.txt	Interactive plots with 5matplotlib inlien		2 years ago

 README.md

Kalman and Bayesian Filters in Python

Introductory text for Kalman and Bayesian filters. All code is written in Python, and the book itself is written using Jupyter Notebook so that you can run and modify the code in your browser. What better way to learn?

"Kalman and Bayesian Filters in Python" looks amazing! ... your book is just what I needed - Allen Downey, Professor and O'Reilly author.

Thanks for all your work on publishing your introductory text on Kalman Filtering, as well as the Python Kalman Filtering libraries. We've been using it internally to teach some key state estimation concepts to folks and it's been a huge help. - Sam Rodkey, SpaceX

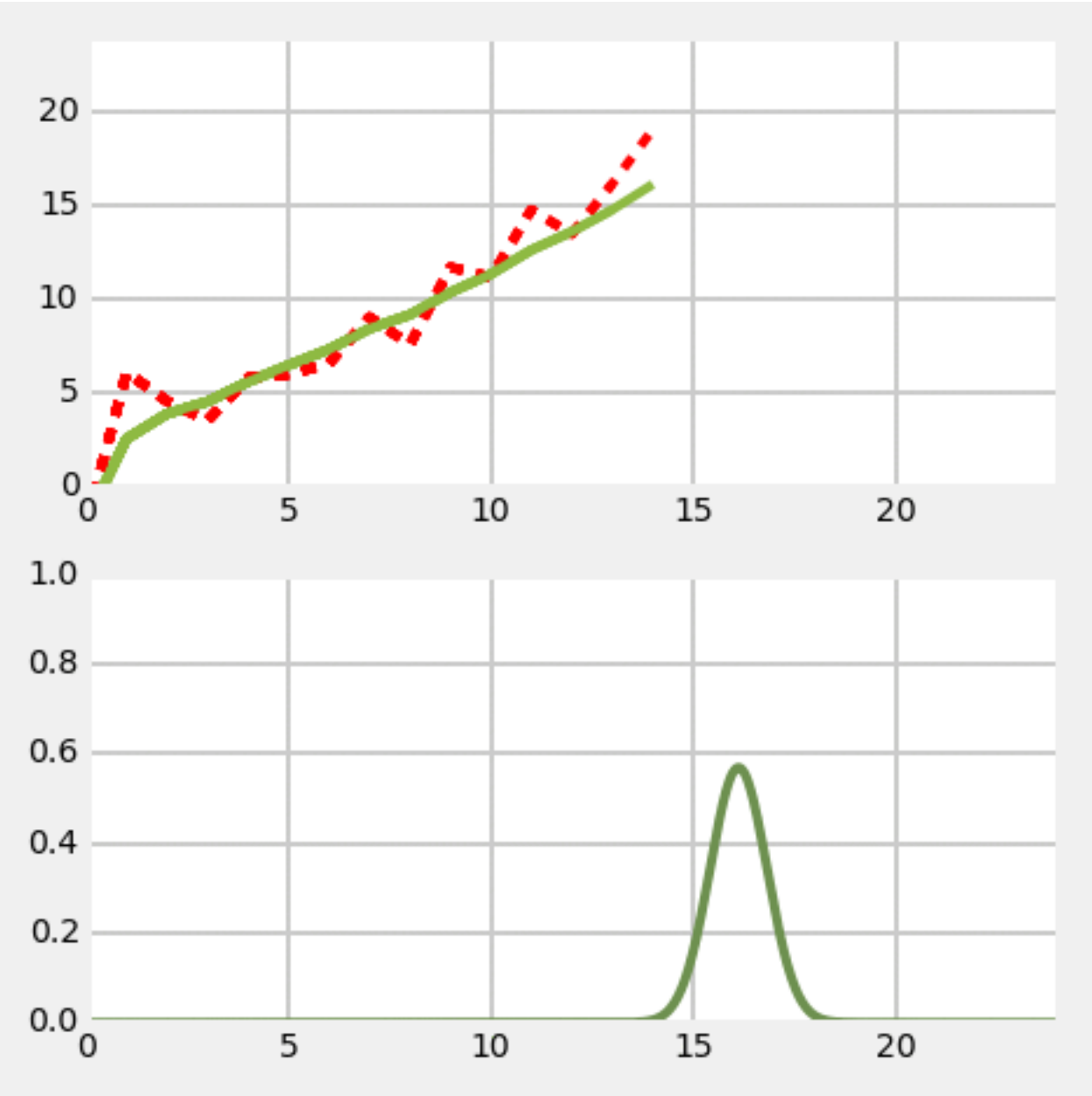
Start reading online now by clicking the binder or Azure badge below:

launch

binder

launch

azurenb



What are Kalman and Bayesian Filters?

Sensors are noisy. The world is full of data and events that we want to measure and track, but we cannot rely on sensors to give us perfect information. The GPS in my car reports altitude. Each time I pass the same point in the road it reports a slightly different altitude. My kitchen scale gives me different readings if I weigh the same object twice.

In simple cases the solution is obvious. If my scale gives slightly different readings I can just take a few readings and average them. Or I can replace it with a more accurate scale. But what do we do when the sensor is very noisy, or the environment makes data collection difficult? We may be trying to track the movement of a low flying aircraft. We may want to create an autopilot for a drone, or ensure that our farm tractor seeded the entire field. I work on computer vision, and I need to track moving objects in images, and the computer vision algorithms create very noisy and unreliable results.

This book teaches you how to solve these sorts of filtering problems. I use many different algorithms, but they are all based on Bayesian probability. In simple terms Bayesian probability determines what is likely to be true based on past information.

If I asked you the heading of my car at this moment you would have no idea. You'd proffer a number between 1° and 360° degrees, and have a 1 in 360 chance of being right. Now suppose I told you that 2 seconds ago its heading was 243° . In 2 seconds my car could not turn very far so you could make a far more accurate prediction. You are using past information to more accurately infer information about the present or future.

The world is also noisy. That prediction helps you make a better estimate, but it also subject to noise. I may have just braked for a dog or swerved around a pothole. Strong winds and ice on the road are external influences on the path of my car. In control literature we call this noise though you may not think of it that way.

There is more to Bayesian probability, but you have the main idea. Knowledge is uncertain, and we alter our beliefs based on the strength of the evidence. Kalman and Bayesian filters blend our noisy and limited knowledge of how a system behaves with the noisy and limited sensor readings to produce the best possible estimate of the state of the system. Our principle is to never discard information.

Say we are tracking an object and a sensor reports that it suddenly changed direction. Did it really turn, or is the data noisy? It depends. If this is a jet fighter we'd be very inclined to believe the report of a sudden maneuver. If it is a freight train on a straight track we would discount it. We'd further modify our belief depending on how accurate the sensor is. Our beliefs depend on the past and on our knowledge of the system we are tracking and on the characteristics of the sensors.

The Kalman filter was invented by Rudolf Emil Kálmán to solve this sort of problem in a mathematically optimal way. Its first use was on the Apollo missions to the moon, and since then it has been used in an enormous variety of domains. There are Kalman filters in aircraft, on submarines, and on cruise missiles. Wall street uses them to track the market. They are used in robots, in IoT (Internet of Things) sensors, and in laboratory instruments. Chemical plants use them to control and monitor reactions. They are used to perform medical imaging and to remove noise from cardiac signals. If it involves a sensor and/or time-series data, a Kalman filter or a close relative to the Kalman filter is usually involved.

Motivation

The motivation for this book came out of my desire for a gentle introduction to Kalman filtering. I'm a software engineer that spent almost two decades in the avionics field, and so I have always been 'bumping elbows' with the Kalman filter, but never implemented one myself. As I moved into solving tracking problems with computer vision the need became urgent. There are classic textbooks in the field, such as Grewal and Andrew's excellent *Kalman Filtering*. But sitting down and trying to read many of these books is a dismal experience if you do not have the required background. Typically the first few chapters fly through several years of undergraduate math, blithely referring you to textbooks on topics such as Itô calculus, and present an entire semester's worth of statistics in a few brief paragraphs. They are good texts for an upper undergraduate course, and an invaluable reference to researchers and professionals, but the going is truly difficult for the more casual reader. Symbology is introduced without explanation, different texts use different terms and variables for the same concept, and the books are almost devoid of examples or worked problems. I often found myself able to parse the words and comprehend the mathematics of a definition, but had no idea as to what real world phenomena they describe. "But what does that *mean*?" was my repeated thought.

However, as I began to finally understand the Kalman filter I realized the underlying concepts are quite straightforward. A few simple probability rules, some intuition about how we integrate disparate knowledge to explain events in our everyday life and the core concepts of the Kalman filter are accessible. Kalman filters have a reputation for difficulty, but shorn of much of the formal terminology the beauty of the subject and of their math became clear to me, and I fell in love with the topic.

As I began to understand the math and theory more difficulties present themselves. A book or paper's author makes some statement of fact and presents a graph as proof. Unfortunately, why the statement is true is not clear to me, nor is the method for making that plot obvious. Or maybe I wonder "is this true if $R=0$?" Or the author provides pseudocode at such a high level that the implementation is not obvious. Some books offer Matlab code, but I do not have a license to that expensive package. Finally, many books end each chapter with many useful exercises. Exercises which you need to understand if you want to implement Kalman filters for yourself, but exercises with no answers. If you are using the book in a classroom, perhaps this is okay, but it is terrible for the independent reader. I loathe that an author withholds information from me, presumably to avoid 'cheating' by the student in the classroom.

From my point of view none of this necessary. Certainly if you are designing a Kalman filter for a aircraft or missile you must thoroughly master of all of the mathematics and topics in a typical Kalman filter textbook. I just want to track an image on a screen, or write some code for an Arduino project. I want to know how the plots in the book are made, and chose different parameters than the author chose. I want to run simulations. I want to inject more noise in the signal and see how a filter performs. There are thousands of opportunities for using Kalman filters in everyday code, and yet this fairly straightforward topic is the provenance of rocket scientists and academics.

I wrote this book to address all of those needs. This is not the book for you if you program navigation computers for Boeing or design radars for Raytheon. Go get an advanced degree at Georgia Tech, UW, or the like, because you'll need it. This book is for the hobbiest, the curious, and the working engineer that needs to filter or smooth data.

This book is interactive. While you can read it online as static content, I urge you to use it as intended. It is written using Jupyter Notebook, which allows me to combine text, math, Python, and Python output in one place. Every plot, every piece of data in this book is generated from Python that is available to you right inside the notebook. Want to double the value of a parameter? Click on the Python cell, change the parameter's value, and click 'Run'. A new plot or printed output will appear in the book.

This book has exercises, but it also has the answers. I trust you. If you just need an answer, go ahead and read the answer. If you want to internalize this knowledge, try to implement the exercise before you read the answer.

This book has supporting libraries for computing statistics, plotting various things related to filters, and for the various filters that we cover. This does require a strong caveat; most of the code is written for didactic purposes. It is rare that I chose the most efficient solution (which often obscures the intent of the code), and in the first parts of the book I did not concern myself with numerical stability. This is important to understand - Kalman filters in aircraft are carefully designed and implemented to be numerically stable; the naive implementation is not stable in many cases. If you are serious about Kalman filters this book will not be the last book you need. My intention is to introduce you to the concepts and mathematics, and to get you to the point where the textbooks are approachable.

Finally, this book is free. The cost for the books required to learn Kalman filtering is somewhat prohibitive even for a Silicon Valley engineer like myself; I cannot believe they are within the reach of someone in a depressed economy, or a financially struggling student. I have gained so much from free software like Python, and free books like those from Allen B. Downey [here](#). It's time to repay that. So, the book is free, it is hosted on free servers, and it uses only free and open software such as IPython and mathjax to create the book.

Reading Online

The book is written as a collection of Jupyter Notebooks, an interactive, browser based system that allows you to combine text, Python, and math into your browser. There are multiple ways to read these online, listed below.

binder

binder serves interactive notebooks online, so you can run the code and change the code within your browser without downloading the book or installing Jupyter.

 launch binder

nbviewer

The website <http://nbviewer.org> provides an Jupyter Notebook server that renders notebooks stored at github (or elsewhere). The rendering is done in real time when you load the book. You may use [this nbviewer link](#) to access my book via nbviewer. If you read my book today, and then I make a change tomorrow, when you go back tomorrow you will see that change. Notebooks are rendered statically - you can read them, but not modify or run the code.

nbviewer seems to lag the checked in version by a few days, so you might not be reading the most recent content.

GitHub

GitHub is able to render the notebooks directly. The quickest way to view a notebook is to just click on them above. However, it renders the math incorrectly, and I cannot recommend using it if you are doing more than just dipping into the book.

PDF Version

A PDF version of the book is available [here](#)

The PDF will usually lag behind what is in github as I don't update it for every minor check in.

Downloading and Running the Book

However, this book is intended to be interactive and I recommend using it in that form. It's a little more effort to set up, but worth it. If you install IPython and some supporting libraries on your computer and then clone this book you will be able to run all of the code in the book yourself. You can perform experiments, see how filters react to different data, see how different filters react to the same data, and so on. I find this sort of immediate feedback both vital and invigorating. You do not have to wonder "what happens if". Try it and see!

The book and supporting software can be downloaded from GitHub by running this command on the command line:

```
git clone --depth=1 https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python.git
pip install filterpy
```

Instructions for installation of the IPython ecosystem can be found in the Installation appendix, found [here](#).

Once the software is installed you can navigate to the installation directory and run Jupyter notebook with the command line instruction

```
jupyter notebook
```

This will open a browser window showing the contents of the base directory. The book is organized into chapters. To read Chapter 2, click on the link for chapter 2. This will cause the browser to open that subdirectory. In each subdirectory there will be one or more IPython Notebooks (all notebooks have a .ipynb file extension). The chapter contents are in the notebook with the same name as the chapter name. There are sometimes supporting notebooks for doing things like generating animations that are displayed in the chapter. These are not intended to be read by the end user, but of course if you are curious as to how an animation is made go ahead and take a look.

This is admittedly a somewhat cumbersome interface to a book; I am following in the footsteps of several other projects that are somewhat repurposing Jupyter Notebook to generate entire books. I feel the slight annoyances have a huge payoff - instead of having to download a separate code base and run it in an IDE while you try to read a book, all of the code and text is in one place. If you want to alter the code, you may do so and immediately see the effects of your change. If you find a bug, you can make a fix, and push it back to my repository so that everyone in the world benefits. And, of course, you will never encounter a problem I face all the time with traditional books - the book and the code are out of sync with each other, and you are left scratching your head as to which source to trust.

Companion Software

 pypi v1.4.1

I wrote an open source Bayesian filtering Python library called **FilterPy**. I have made the project available on PyPi, the Python Package Index. To install from PyPi, at the command line issue the command

```
pip install filterpy
```

If you do not have pip, you may follow the instructions here: <https://pip.pypa.io/en/latest/installing.html>.

All of the filters used in this book as well as others not in this book are implemented in my Python library FilterPy, available [here](#). You do not need to download or install this to read the book, but you will likely want to use this library to write your own filters. It includes Kalman filters, Fading Memory filters, H infinity filters, Extended and Unscented filters, least square filters, and many more. It also includes helper routines that simplify the designing the matrices used by some of the filters, and other code such as Kalman based smoothers.

FilterPy is hosted github at (<https://github.com/rllabbe/filterpy>). If you want the bleeding edge release you will want to grab a copy from github, and follow your Python installation's instructions for adding it to the Python search path. This might expose you to some instability since you might not get a tested release, but as a benefit you will also get all of the test scripts used to test the library. You can examine these scripts to see many examples of writing and running filters while not in the Jupyter Notebook environment.

Alternative Way of Running the Book in Conda environment

If you have conda or miniconda installed, you can create environment by

```
conda env update -f environment.yml
```

and use

```
source activate kf_bf
```

and

```
source deactivate kf_bf
```

to activate and deactivate the environment.

Issues or Questions

If you have comments, you can write an issue at GitHub so that everyone can read it along with my response. Please don't view it as a way to report bugs only. Alternatively I've created a gitter room for more informal discussion. [gitter](#) [join chat](#)

License



Kalman and Bayesian Filters in Python by [Roger R. Labbe](#) is licensed under a [Creative Commons Attribution 4.0 International License](#).

All software in this book, software that supports this book (such as in the the code directory) or used in the generation of the book (in the pdf directory) that is contained in this repository is licensed under the following MIT license:

The MIT License (MIT)

Copyright (c) 2015 Roger R. Labbe Jr

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contact

