موضوع ارایه:......................

1.معرفی:

ما در این ارایه سعی داریم ابتدا با چند مدل مختلف سری های زمانی برای پیش بینی میانگین قیمت معاملاتی در بازه ی یک دقیقه ای آینده مدلی ارایه کنیم.همواره در تخمین های اولیه ای که ما به وسیله ی مدل خود بدست می آوریم یک عدم قطعیت ذاتی وجود دارد.به همین خاطر قصد داریم با مشاهده ی قیمت open و مقایسه آن با تخمین های اولیه یک تخمین بهبود یافته تر بدست دهیم.برای این کار از فیلتر های کالمن استفاده میکنیم.برای اینکه مدلسازی ما با فیلتر کالمن منطبق شود فرض میکنیم مشاهده ی ما از قیمت open در واقع یک مشاهده ی نویزی با توزیع نرمال از همان میانگین قیمت معاملاتی در یک دقیقه آینده است که قصد مطالعه آنرا داریم.

مطالعات ما در این ارایه مربوط به نحوه ی تغییرات میانگین قیمت معاملاتی Bitcoin در طول بازه های زمان یک دقیقه ای است.بدیهی است که نتایج نهایی کمابیش قابل پیاده سازی در سایر بازارها و در مورد سایر پارامتر های بازار و سایر بازه های زمانی نیز میباشد.

2.مقدمات و پیشنیاز ها:

2.1:فیلتر کالمن:

فیلتر کالمن الگوریتمی است که با با داشتن تقریب های خام اولیه و مشاهده و اندازه گیری هایی ازسیستم (که فرض میشود هم مشاهده ما از سیستم و هم تقریب های خام هر دو دارای خطا هستند) این دو رو با یکدیگر ترکیب کرده و با استفاده از هر دو در خروجی یک تقریب بهبود یافته تر بدست میدهد.فرض کنید xt  متغیری باشد که نحوه ی تغییرات آن در طول زمان به طور "حدودی" به وسیله ی سیستم دینامیکی خطی :

xt=Fxt-1+But-1+wt-1 E1

قابل توصیف باشد.که در آن wt-1 یک متغیر تصادفی است که نویز یا خطای مدل نام دارد. wt-1~N(0,Qt-1)که در آن Qt-1ماتریس کوواریانس متغیر wt-1 است.

از طرفی فرض کنید برای هر t به جای دسترسی مستقیم به xtبه یک ورژن نویزی از آن یعنی zt دسترسی داریم که :

zt=xt+vt E2

و در اینجا vt یک متغیر تصادفی با توزیع vt~N(0,Rt)است.

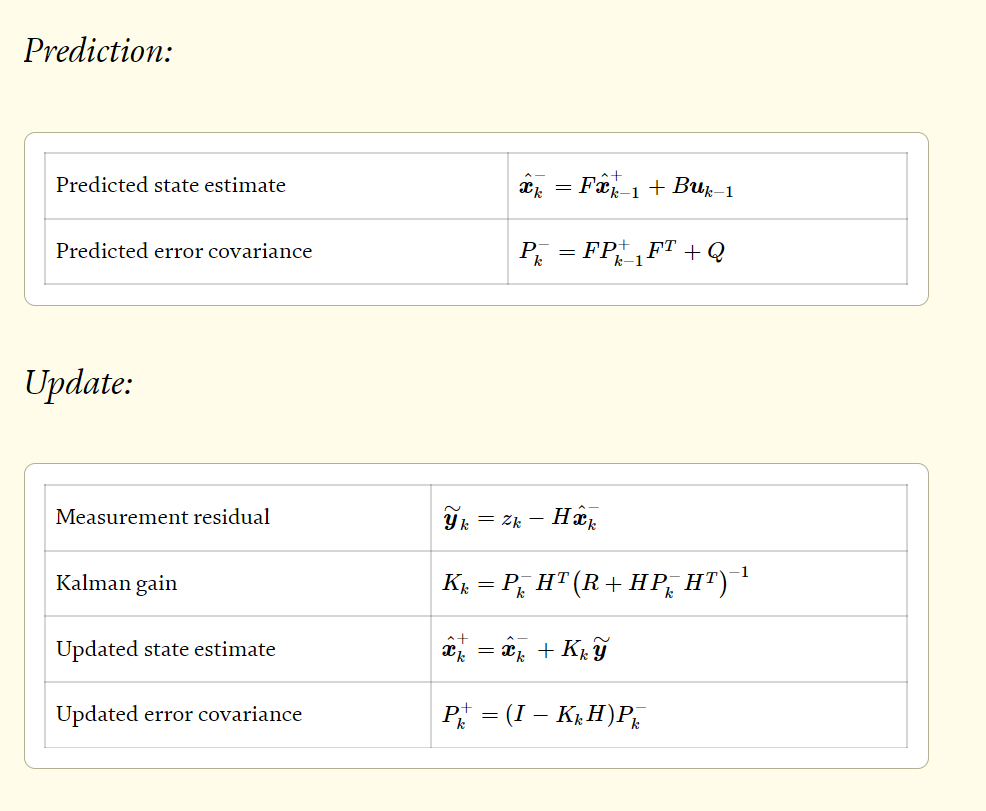
تا به این جا ما دو تقریب از مقدار xtبدست آورده ایم که عبارتند از :

تقریب با استفاده از دانستن دینامیک سیستم :Fxt-1+But-1

تقریب با استفاده از مشاهده و اندازه گیری:zt

الگوریتم کالمن به طریق زیر یک تقریب بهبود یافته تر نسبت به هردوی تقریب های قبلی بدست میدهد:

الگوریتم یک:

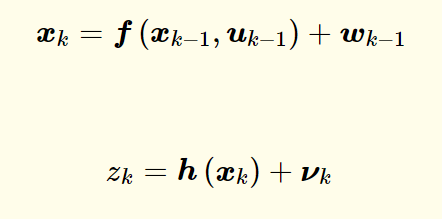


در این تصویر علامت منفی روی تقریب ها به معنی اینست که این تقریب خام اولیه است وعلامت مثبت یعنی تقریب بعد از انجام الگوریتم کالمن بهبود یافته است.

آنچه که در تصویر بالا در قسمت محاسبات مربوط به prediction مطرح شده است در واقع تخمین اولیه و میزان عدم قطعیتی است که ما از تخمین خود داریم.این تقریب از فرض ما بدست می آید که نحوه ی تغییرات xt به وسیله سیستم دینامیک خطی E1 توضیح داده میشود.

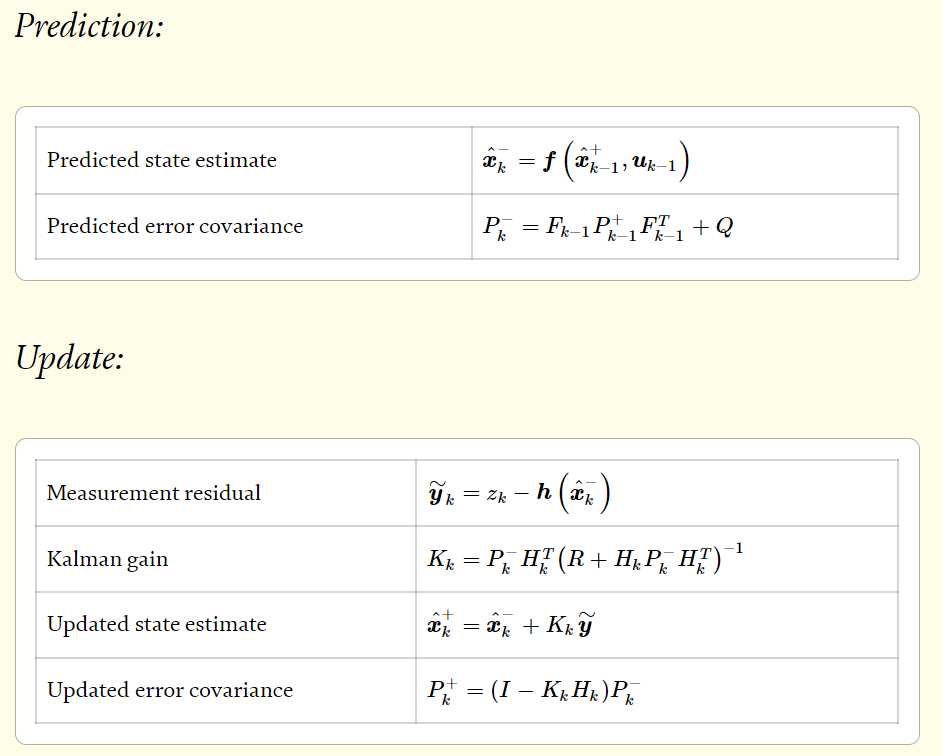
2.2:توسعه الگوریتم کالمن به سیستم های غیرخطی:

اگر xtدر طول زمان به جای دینامیک خطی بالا از تابع غیر خطی f پیروی کند به طوری که:

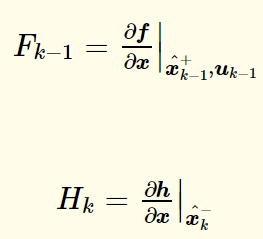


الگوریتم کالمن مشابه قبل خواهد بود.

الگوریتم دو:



با این تفاوت که H ,Fمقدارشان از ژاکوبین h,fدر آخرین مرحله بدست می آید.یعنی :



میتوان به طرق مختلف الگوریتم کالمن را تعمیم داد به عنوان مثال :

نرمال بودن نویزهای سیستم:هر تخمین به همراه نویز آن متناظر با توزیع احتمال است.با فرض نرمال بودن نویزها داریم:

رابطه یک :

|  |  |  |  |
| --- | --- | --- | --- |
|  | مقدار تخمین | Covariance نویز | توزیع متناظر |
| مدل | x̂k | Pk | x~N(x̂k,Pk) |
| اندازه گیری | yk | Rk | y~N(yk,Rk) |

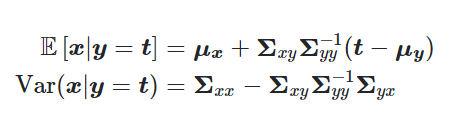
که در واقع رابطه ی مقدار و عدم قطعیت تخمین با توزیع احتمال متناظر با آن به شکل زیر است.اگر توزیع احتمال مربوط به یک متغیر را داشته باشیم آنگاه :

رابطه دو :

مقدار تخمین = (متغیر) E

عدم قطعیت = (متغیر) Var

در مورد فیلتر کالمن متغیر تصادفی متناظر با predicted state estimation را با x داده ایم و متغیر تصادفی مربوط به اندازه گیری را با y .توزیع این دو متغیر در رابطه ی یک آورده شده است.داریم:

رابطه 3. 

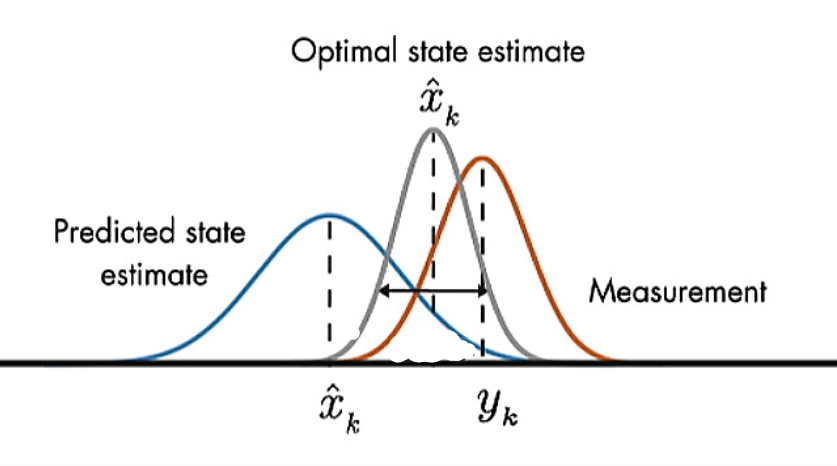
و اگر این روابط را به شکل رابطه های خود بازنویسی کنیم فیلتر کالمن به شکل زیر در می آید:

رابطه 4:

تخمین بدست آمده از فیلتر کالمن = (y=yk+1|x) E

عدم قطعیت موجود هنگام استفاده از فیلتر کالمن = (y=yk+1|x) Var

برای دیدن جزییات اثبات به مرجع [1] مراجعه کنید.پس در نتیجه اگر ما بتوانیم مقادیر سمت چپ معادلات رابطه 4 را محاسبه کنیم لزومی به فرض نرمال بودن نویز نیست.



میتوان سایر فرض ها را نیز به حالت کلی تعمیم دهیم.مثلا میتوان فرض کرد دینامیک تغییرات xt از رابطه ی E1 پیروی نمیکند.در آنصورت تخمین خام اولیه و عدم قطعیت متناظر با آن (xt-,Pt-) را باید به طریق دیگری محاسبه کرد و باز هم تخمین موجود در رابطه 4 را میتوان به عنوان تخمین بهبود یافته از رابطه کالمن در نظر گرفت.

برای مشاهده ی حالات تعمیم یافته به منبع [4] مراجعه کنید.

2.3:سری های زمانی

فرض کنید بردار یا اسکالر xt حاوی اطلاعات بازار در بازه ی t \_ام باشد. برای اینکه از مشاهدات M روز گذشته بتوانیم تقریبی خطی برای امروز بدست آوریم از روش (Auto Regressive Model with Memory Length M) به طریق زیراستفاده میکنیم :

رابطه 5: x̂t=a1xt-1+…+aMxt-M

اگر xt بردار باشد از رابطه ی :

رابطه 6: x̂t=A1xt-1+…+AMxt-M

استفاده میکنیم.که ورژن برداری رابطه ی قبل است.البته ما دو رابطه ی قبل را به شکل زیر بازنویسی میکنیم:

رابطه 7 :

Ai ها میتوانند ماتریس 1 در 1 (اسکالر) باشند.

زیرا این نحوه ی نوشتار به ساختار رابطه E1نزدیک تر است.

ztاطلاعات ثبت شده برای قیمت open بازه ی t\_ام باشد و مقدار xtمقدار متوسط آن در همان بازه باشد داریم :

zt=xt+wt

که wt نویز در لحظه ی t\_ام است.

جمع آوری داده ها :

داده های در فایل data.csv ذخیره و ارسال شده اند.برای دسترسی به کدی که داده ها را جمع آوری کرده به مرجع [2]مراجعه کنید.در این فایل اطلاعات مربوط به Bitcoin در بازه های یک دقیقه ای به صورت سطری ثبت و گزارش شده است.در این فایل علاوه بر ویژگی های معمول که در بازار از آنها برای توصیف وضعیت استفاده میشود(Open,High,Low,Close,Volume,Value,Taker Buy Volume,Taker Buy Value …) از 4ویژگی دیگر (با Feature 1 …Feature 4 نمایش داده شده اند.) استفاده کرده ایم که در واقع تابع سایر ویژگی های در همین سطر هستند و ما به جای پارامتر های مرسوم قصد داریم این 4 ویژگی را با سری زمانی مورد مطالعه خود قرار دهیم.

مدل های مورد مطالعه ما:

مدلی که ما برای پیش بینی xt استفاده میکنیم تعمیمی از رابطه 7 است.

رابطه 9:

که در آن F همان ماتریس موجود در رابطه 7 است.u(n) یک بردار از ویژگی های دیگری است که برای تخمین از آنها کمک میگیریم.

w(n) نویز سیستم است.و x(n)به شکل زیر است.

رابطه 10:

که در آن xn میانگین قیمت معاملاتی سهام در دقیقه ی n \_ ام است و u(n) مجموعه ی ویژگی های کمکی ایست که در فایل data.csvذخیره شده است در بازه ی زمانی متناظر با دقیقه ی n\_ام.

فرض میکنیم F و G دارای sparsity pattern زیر باشند.

رابطه11:

و ماتریس F مشابه رابطه 7

برای پیدا کردن F و G مناسب باید مسیله بهینه سازی زیر را حل کنیم.

رابطه 12:

Minimize F,G

که معادلست با :

Minimize F,G

اگر F و G ماتریس های تنک(sparse) نبودند برای حل این مسیله آخر توجه داشته باشید که مسیله 13.15 مرجع [6] (VMLS) مشابه این مسیله است و میتوانستیم از راه حل آن برای این حالت استفاده کنیم.(مدل مورد بحث در این جا از ماتریس های تنک استفاده میشود اما در آینده در مدل 3 ماتریس های چگال را نیز بررسی میکنیم.)

اما برای ماتریس های تنک از راه زیر میتوان مسیله را حل کرد.

که برای حل آن مسیله ی کمترین مربعات زیر را داریم:

رابطه 13 :

Minimizef\_i , g\_i

این رابطه مناسب ترین F و G برای یک بازه ی زمانی را به ما میدهد و احتمال overfitting وجود دارد.ما به جای رابطه 12 از رابطه ی زیر استفاده میکنیم.

رابطه 14 :

Minimize F,G

که برای حل آن به رابطه ی زیر میرسیم:

رابطه 15:

A=

Minimizef\_i , g\_i

مدل ارایه شده در رابطه 7 نهایتا در این جا تکمیل شد و با تست کردن آن در عمل بر روی تعدادی داده ی تصادفی به این نتیجه رسیدیم که این مدل کمی بهتر از قیمت openقیمت را پیسش بینی میکند.و در واقع واریانس نتایج این مدل از واریانس نتایج ناشی از اندازه گیری صرف بدون استفاده از رابطه 7 واریانس کمی داشت.به همین دلیل ما مدل های متفاوت تر و در صورت امکان غیر خطی را نیز در ادامه مورد بررسی قرار میدهیم.

در داده های تست داریم:

رابطه:16 [≈](https://fa.wikipedia.org/wiki/%D8%AA%D9%82%D8%B1%DB%8C%D8%A8) Var(z-x)< Var(x̂kalman-x)

بهتر است به جای مطالعه مستقیم روند تغییرات قیمت یا سایر شاخص های مرسوم از 4 ویژگی موجود در data.csv استفاده کنیم.به دلیل این که بتجربه ما این ویژگی ها قابل مطالعه تر هستند.در واقع روند تغییرات قیمت تابع بسیاری از اطلاعاتی است که ما به آنها دسترسی نداریم و رفتار آن با چند ویژگی کلی مثل قیمت لحظه ی قبل و ...قابل محاسبه نیست.

از طرفی استفاده از Auto Regressive Models که تخمینی خطی از یک پارامتر بر حسب مقادیر قبلی آن بدست میدهد در مورد پارامتر های بازار که رفتاری بسیار دور از رفتار خطی از خود بروز میدهند دلیل دیگری است که نتیجه نهایی دقت پایینی داشت.

ما در ادامه قصد داریم چندین برآوردگر غیرخطی مختلف تولید کنیم که هر کدام تقریبی از xtبدست میدهند.

مدلهای مختلف پیشنهاد شده.

در قسمت قبل نحوه ی کار الگوریتم کالمن و مدل یک توضیح داده شد.در مورد سایر مدل ها روند مشابهی وجود دارد به همین خاطر در این قسمت به نوشتن خلاصه روند در مورد مدل های دیگر بسنده میکنیم.

اجرای الگوریتم کالمن در مدل های خطی (3 مدل اول):

مدل 1 :

دینامیک مدل :

که در آن :

wn~N(0,s2)

و در آن F,G ماتریس های تنک با ویژگی های زیر هستند:

برای پیدا کردن f , gمناسب باید مسیله بهینه سازی موجود در رابطه 15 را حل کنیم داریم :

مدل 2:دینامیکی مشابه مدل پیش با این تفاوت که به جای مسیله بهینه سازی آن مسیله ی زیر را جایگزین میکنیم :

Minimizef\_i , g\_i

که در آن W یک ماتریس قطری به شکل زیر است :

که در آن .1>a>0

در واقع تفاوت مدل 2 با مدل 1 در اینست که ضرایب f , g در مدل 2 حساسیت بیش تری نسبت به داده های پایانی ای دارند که به عنوان ورودی دریافت میکنند.

برای پیدا کردن f , gمناسب باید مسیله بهینه سازی موجود در رابطه 15 را حل کنیم داریم :

مدل 3: برای این مدل نیز دینامیکی مشابه با دو مدل قبل در نظر بگیرید با این تفاوت که قید های مربوط به sparsity pattern دو مدل قبل را برمیداریم و حال F , Gدو ماتریس کاملا چگال میتوانند باشند.

برای پیدا کردن F , G مناسب در این مدل باید مسیله بهینه سازی زیر را حل کنیم.

که همانطور که در تمرین 13.15 کتاب VMLS به آن اشاره شد داریم :

چون دینامیک سه مدل اول کاملا منطبق بر دینامیک استاندارد الگوریتم کالمن است همان الگوریتم را عینا باید اجرا کرد.تفاوت نماد گذاری به این صورتست که کافیست به جای H ماتریس همانی I را قرار دهید و به جای ماتریس B ماتریس G را قرار دهید.

مدل های غیر خطی:

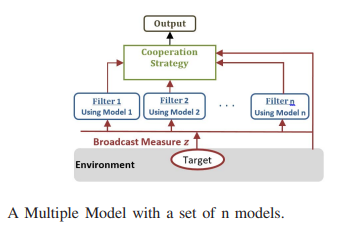
مدل 4:

مدل 5:

تلفیق چندین الگوریتم برآوردگر مختلف:

یک چالش پیش رو این هست که احتمالا هیچ یک از برآورد گرهای قبلی یک غلبه ی کامل در همه ی حالات نسبت به سایرین ندارد و هر کدام تحت شرایط خاصی بهتر از بقیه عمل میکند.در شرایطی که در الگوریتم کالمن فقط یک برآورد گر داریم .

حال ما میخواهیم الگوریتم کالمن را به نحوی توسعه دهیم که در حالت وجود چندین برآوردگر اولیه نیز کارآمد باشد.اساس این روش به این صورت خواهد بود که میانگین وزن داری از برآوردگر ها و دستگاه های اندازه گیری را(برآوردگر یا ابزار اندازه گیری i\_ام را با x̂i نمایش میدهیم) را به عنوان برآورد نهایی خود قرار دهیم.و نحوه ی وزن دهی به برآوردگر ها نسبت معکوس با عدم قطعیت موجود(واریانس) در آن برآوردگر داشته باشد.



به طرق مختلف میتوان این کار را انجام داد.به عنوان مثال ابتدا هر مدل را با مشاهدات خود از قیمت open فیلتر کنیم و بعد میانگین وزن داری از آنها را بگیریم.یا ابتدا میانگین وزن داری از برآوردگر ها را محاسبه کنیم و مقدار آنرا به عنوان تقریب خام اولیه به الگوریتم کالمن بدهیم.یک راه دیگر اینست که برآوردگر i\_ام را با برآوردگر i+1\_ام به طور سری تلفیق کنیم(در این جا برآوردگر i+1\_ام نقش دستگاه اندازه گیری را بازی میکند) و نتیجه را با برآوردگر i+2\_ام تلفیق کنیم ...

روش های مختلف وزن دهی برای میانگین های وزن دار نیز میتواند الگوریتم های تلفیقی متفاوتی را تولید کند.

کدام یک از روش های پیشنهاد شده در بالا مطلوب تر است؟

مشابه با اثبات دومی که در مرجع [1]ذکر شده است میتوان محاسبات مشابهی انجام داد و رابطه ای مشابه رابطه 4 تولید کرد و در واقع با این کار رابطه صریحی برای مرحله update الگوریتم بدست می آید که بر حسب تمام x̂i ها و عدم قطعیت متناظر با آنها خواهد بود.  
........................

نمودار ها......................

روش های برای اندازه گیری نویز موجود در سیستم و ماتریس های کوواریانس.......................

ادامه:

به طریق مختلفی میتوان این مسیله را تعمیم داد در زیر به راهکارهایی برای تعمیم به حالت کلی تر سوال میپردازیم.

* چهار ویژگی ذکر شده بنا به تجربه قابل مطالعه تر از شاخص های مرسوم بازار بودند.میتوان تعداد زیادی شاخص تصادفی تولید کرد و با استخراج ویژگی های جدید مناسب .از آنها برای استفاده در سری زمانی استفاده کرد.
* مقدار حافظه سری زمانی یا مقدار داده ی ای که برای محاسبه پارامتر های سری های زمانی استفاده کرده ایم به طور دستی در این مقادیر تنظیم شده بود.میتوان روشهایی برای تنظیم خودکار آنها بوجود آورد
* الگوریتم های بکار رفته در کدها دارای انواع پتانسیل ها برای بهبود بخشیدن چه از نظر سرعت اجرا چه از نظر دقت و ... میباشد.

با تشکر از حسن توجه شما

محسن قدرتی \_محمد صالح بهرامی

مرداد 1401

بهینه سازی در علوم داده

مراجع:

1. <https://math.stackexchange.com/questions/840662/an-explanation-of-the-kalman-filter>

2. <https://github.com/MohsenGhodrati/cvx_project_Kalman_Filter>

3. <https://www.intechopen.com/chapters/63164>

4. [https://www.kalmanfilter.net/default.aspx](https://www.kalmanfilter.net/default.aspx%20%0d5)

5. <https://www.youtube.com/watch?v=mwn8xhgNpFY&list=PLn8PRpmsu08pzi6EMiYnR-076Mh-q3tWr>

# 6. Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares [Stephen Boyd](https://www.stanford.edu/~boyd/) & [Lieven Vandenberghe](https://www.ee.ucla.edu/~vandenbe/)

# 7. <https://web.stanford.edu/~boyd/papers/low_dim_pred_time_series.html>

# 8. <https://web.stanford.edu/~boyd/papers/auto_ks.html>

# 9. <https://web.stanford.edu/~boyd/papers/lfd_lqr.html>

# 11. <https://profdoc.um.ac.ir/articles/a/1073637.pdf>