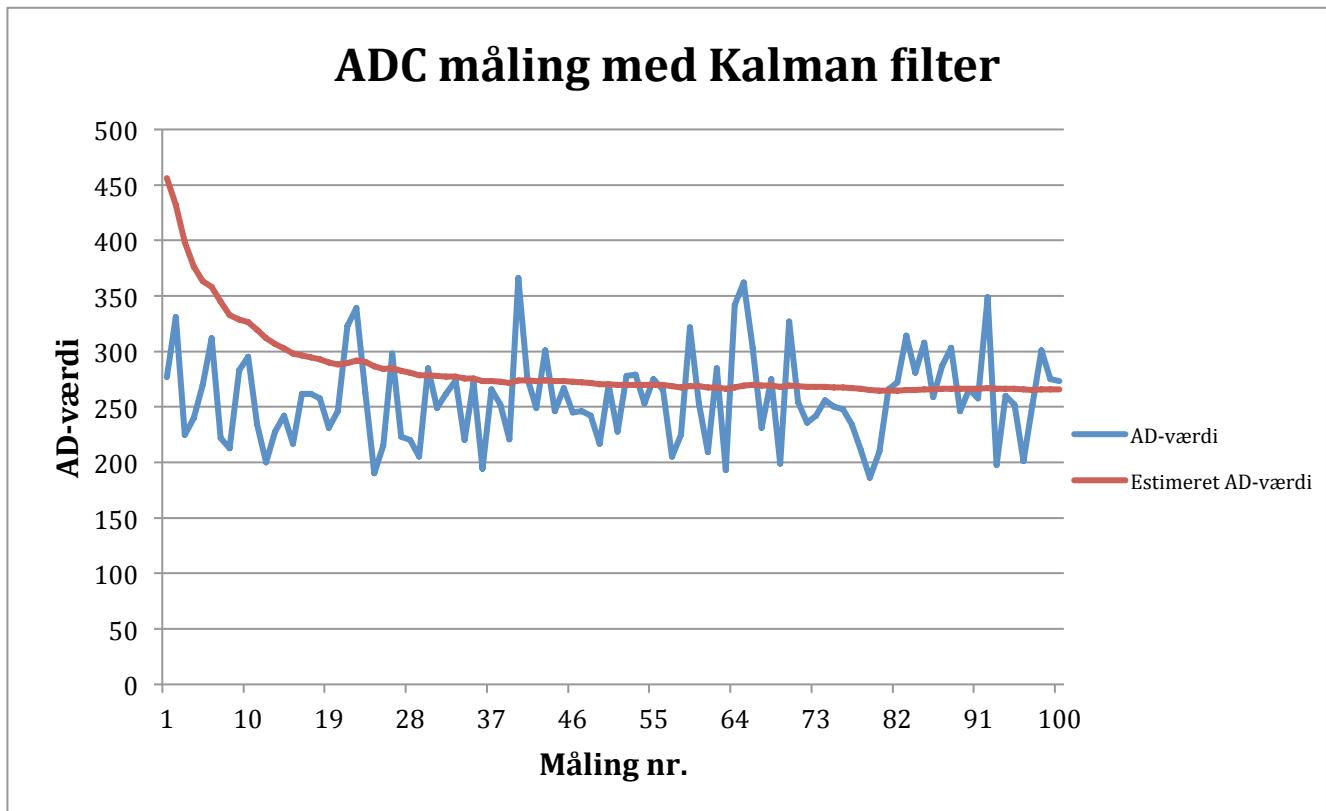


**Holstebro Tekniske Gymnasium**

SRP 2011 – Mat/El-teknik

*Kalman filteret*

Udarbejdet af:

Kristian Sloth Lauszus, 3b

**Udleveret:** Tirsdag d. 13. december 2011**Afleveret:** Tirsdag d. 20. december 20111. Vejleder:  
Jens Holmark Petersen2. Vejleder:  
Bent Arnoldsen

## Abstract

This paper examines the basic concepts of the Kalman filter and how to implemented it in a basic Analog to Digital converter using a microcontroller. It is not a theory analysis of the Kalman filter itself, as it is very complex. The purpose is instead to give a practical approach on how to implement a Kalman filter in a one-dimensional system.

At first it introduces the elementary of statistics and probability theory including frequency, mean, variance, and standard deviation. Finally it sums up by explaining the relationship between covariance and variance. It then looks at normal distribution and how they are formed. Furthermore it explains how to estimate if a set of data is normally distributed using a normal probability plot.

With the knowledge obtained from these chapters, an introduction to the Kalman filter and the theory behind it. Also the Kalman filter equations, which is needed to implement the filter, is explained.

It then moves on to a quick introduction to Analog to Digital converters and explains how an AD-converter introduces noise to the signal.

Finally this theory is used in a real life example using an Arduino Mega. The AD-converter inside the microcontroller measures which is then filtered by the Arduino Mega using a Kalman filter.

Through the experiments it is observed that the Kalman filter can very accurate cancel out the noise, if the noise variances is known.

**Indholdsfortegnelse**

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| <b>Abstract .....</b>                                               | <b>1</b>  |
| <b>Opgavebesvarelse .....</b>                                       | <b>3</b>  |
| <b>Indledning .....</b>                                             | <b>3</b>  |
| <b>Stokastisk variabler .....</b>                                   | <b>4</b>  |
| <b>Frekvens, middelværdi, varians og spredning .....</b>            | <b>5</b>  |
| <b>Kovarians .....</b>                                              | <b>6</b>  |
| <b>Normalfordeling .....</b>                                        | <b>7</b>  |
| <b>Fraktildiagram .....</b>                                         | <b>8</b>  |
| <b>Redegørelse for Kalman filteret .....</b>                        | <b>9</b>  |
| <b>Udtrykket for Kalman filteret .....</b>                          | <b>12</b> |
| <b>Analog til Digital konvertering .....</b>                        | <b>14</b> |
| <b>Normalfordelt hvid støj .....</b>                                | <b>16</b> |
| <b>Analyse af målestøjen .....</b>                                  | <b>17</b> |
| <b>Implementering af Kalman filter .....</b>                        | <b>19</b> |
| <b>Konklusion .....</b>                                             | <b>22</b> |
| <b>Perspektivering .....</b>                                        | <b>23</b> |
| <b>Litteraturliste .....</b>                                        | <b>24</b> |
| <b>Bøger .....</b>                                                  | <b>24</b> |
| <b>Artikler .....</b>                                               | <b>24</b> |
| <b>Internetsider .....</b>                                          | <b>24</b> |
| <b>Programmer .....</b>                                             | <b>25</b> |
| <b>Figurer .....</b>                                                | <b>25</b> |
| <b>Bilag 1 – Skematiske tegning og billede af opstilling .....</b>  | <b>26</b> |
| <b>Bilag 2 – Arduino kode: Analyse af målestøj .....</b>            | <b>27</b> |
| <b>Bilag 3 – Normalfordelingskurve for målestøjen .....</b>         | <b>28</b> |
| <b>Bilag 4 – Arduino kode: AD-konverter med Kalman filter .....</b> | <b>29</b> |
| <b>Bilag 5 – Arduino kode: AD med Kalman filter top 8msb .....</b>  | <b>31</b> |
| <b>Bilag 6 – Graf for Kalman filter top 8msb .....</b>              | <b>33</b> |
| <b>Bilag 7 – CD-ROM .....</b>                                       | <b>34</b> |
| <b>Opgaveformulering .....</b>                                      | <b>35</b> |

## Opgavebesvarelse

### Indledning

I 1960 offentliggjorde Rudolf Kalman sin berømte artikel omhandlende Kalman filteret. Et Kalman filter kan bestemme et systems tilstand over tid ud fra en række indirekte målinger eller ud fra unøjagtige målinger af systemet selv.<sup>1</sup> Disse målinger kan være fx være forstyrret af tilfældig målestøj eller støj i selve systemet.

Selve Kalman filteret består af en række ligninger, der kan beregne et systems tilstand det ikke er mulig at observere. Selvom det er mere end 50 år siden, at Kalman filteret blev udviklet, forskes der stadig i dets muligheder og anvendelse indenfor digital elektronik.

Oprindelig blev Kalman filteret udviklet til rumfartsindustrien, men i dag bruges den i alt lige fra radioer, navigation, selvkørende biler, helikoptere, robotter og til at beregne kraftig nedbør<sup>2,3</sup>.

I denne rapport vil jeg fokusere på Kalman filteret og dets praktiske anvendelser indenfor elektronik, nærmere bestemt AD-konvertering. Jeg vil således se, om jeg ved hjælp af et Kalman filter kan bestemme den faktiske spænding mere præcist end det output AD-konverteren tilkendegiver. Dette kunne fx være relevant i en robot, hvor AD-konverteren er placeret tæt på en såkaldt motorcontroller, der udsender elektrisk støj, som derved vil påvirke AD-målingen.

I stedet for at bruge det omtalte eksempel, vil jeg i stedet simulere støjen ved at påvirke signalet med normalfordelt hvid støj.

For at kunne gøre dette er det imidlertid nødvendigt at kende teorien bag et Kalman filter og AD-konvertering, henholdsvis indenfor fagområderne matematik og el-teknik.

Indenfor matematikken vil jeg begrænse mig til én dimension, så jeg undgår matriceregning, da omfanget af teorien bag dette er meget omfattende. I stedet vil jeg fokusere på mere generelle begreber indenfor sandsynlighedsregning og statistik såsom varians, spredning, normalfordeling, kovarians, diskrete- og kontinuerlige stokastiske variabler mm der alle er nødvendige for at forstå basiteorien bag Kalman filteret, samt implementationen af denne.

---

<sup>1</sup> Simon, Dan: *Kalman Filtering*. I: <http://academic.csuohio.edu/>, 1.6.2011, s. 72-79

<sup>2</sup> Steffensen, Michael: *Logistisk Kalman filter for kraftig nedbør*. I: [www.dmi.dk](http://www.dmi.dk), 1.12.12, s. 2-2

<sup>3</sup> Welch, Greg og Gary Bishop: *An Introduction to the Kalman Filter*. I: <http://www.cs.unc.edu/>, 24.07.2006, s. 1-15

Ud fra denne teori vil jeg derefter forklare baggrunden og ligningen for Kalman filteret. Jeg vil dog ikke bevise selve ligningerne, da udledningen og forklaringen af disse ligger langt over det teoretiske niveau af, hvad det kunne forventes af en universitetsstuderende med hovedfag i matematik.

I el-teknik vil jeg beskrive teorien bag AD-konvertering og komme ind på to konkrete AD-konvertere af typen successiv approksimations AD-konverter og Flash AD-konverter.

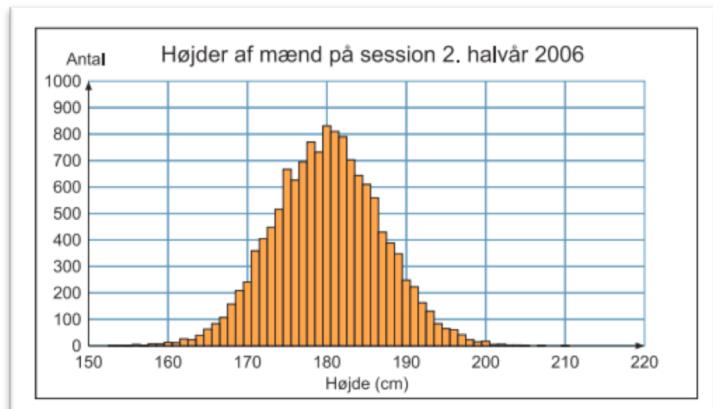
Derefter vil jeg benytte Kalman filteret og den beskrevne teori til at implementere Kalman filteret i en AD-konvertering foretaget af en mikroprocessor. Denne AD-konvertering stammer fra et kendt signal tillagt en normalfordelt målestøj, så jeg derved kan se hvor tæt og hvor hurtigt Kalman filteret angiver på den faktiske spænding.

### Stokastisk variabeler

Deskriptiv statistik, også kaldet beskrivende statistik, er forskellige måder at repræsentere data på. Man kunne fx bruge deskriptiv statistik til analyse af en stokastisk variabel.

En stokastisk variabel er en tilfældig variabel hvor man på forhånd ikke kan forudsige udfaldet og alle udfald er uafhængige. Hvis udfaldsrummet  $U$  er kendt, kaldes det en diskret stokastisk variabel. Dette kunne fx være en kast med en terning, hvor udfaldsrummet, som bekendt er:  $U = \{1,2,3,4,5,6\}$ . Hvis den stokastiske variabel kan antage enhver værdi på tallinen eller i et interval kaldes den en kontinuerlig stokastisk variabel<sup>4</sup>. Et eksempel på dette kunne være en populations højde. En stokastisk variabel betegnes ofte  $X, Y$  eller  $Z$ , hvor  $X, Y$  eller  $Z$  er en funktion, som til hvert udfald  $u \in U$  knytter et reelt tal  $X(u)$ . Tilsammen udgør disse udfald udfaldsrummet  $U$ . Indenfor statistik er man ofte interesseret i at beregne forskellige sandsynligheder og sammenhænge for et givet udfald. Til dette bruges sandsynlighedsregning, der giver metoder til at foretage analyser og beregninger af fx sandsynligheden for et givet udfald, spredningen mm.

Mange fænomener fordeler sig efter en såkaldt normalfordeling. Dette kunne fx være et kast med terning eller højden på en



Figur 1 – Grafen ovenfor viser tydeligt, at en højde på gruppe værnehæftige tilnærmedesvis følger en normalfordeling.  
Kilde: [www.matematiksider.dk](http://www.matematiksider.dk)

<sup>4</sup> Halse, Søren m.fl.: Matematik, Gyldendals Minilex. 1. udg. Gyldendal, 2005

given gruppe. Figur 1 illustrerer 13427 værnepligtiges højde i et histogram. Hver værnepliktig er afrundet til cm, derved opstår der en histogram der tilnærmelsesvis en normalfordeling. Grafen viser tydeligt at der er flest mænd med en højde på ca. 180 cm. Derudover viser den at jo længere væk fra denne middelværdi man går jo færre observationer er der med den angivne værdi. Dette siger dog ikke noget konkret om, hvad sandsynligheden er for, at den næste person, der går ind på sessionslægen kontor, er fx 170 cm. Endnu sværere er det umiddelbart at bedømme hvad sandsynligheden er for, at personen er under 170 cm – dette skrives også  $P(X < 170)$ .

### Frekvens, middelværdi, varians og spredning

Før jeg vil forklare, hvordan man kan beregne sådanne sandsynligheder for en normalfordeling, er det imidlertid nødvendigt at kende en række begreber indenfor statistik og sandsynlighedsregning.

Hypighed  $h(x_i)$  angiver antallet af observationer, der enten er det samme eller ligger inde for samme interval i observationssættes. Ud fra denne kan man frekvensen  $f(x)$ , der er givet ved:

$$f(x_i) = \frac{h(x_i)}{n} \quad (1.1)$$

Hvor  $i$  angiver hvilken observation, der er tale om og  $n$  angiver antallet af observationer. Det næste begreb er middelværdi. Middelværdi angives som  $\mu$  og beregnes således:

$$\mu = \sum_{i=1}^n x_i * f(x_i) \quad (1.2)$$

Der er dog det problem, at hvis observationen stammer fra et diskret stokastisk eksperiment fx ved at slå med to terninger, er det ikke sikkert, at det opfylder denne middelværdi, da det netop er tilfældigt. Ved simple beregninger kan det vises, at middelværdien  $\mu$  er 7, men dette behøver ikke være tilfældet, hvis man fx slår tre gange. Derfor omtaler man nogle gange en stokastisk variabels middelværdi som forventningsværdien  $E(X)$ . Denne udregnes tilsvarende som middelværdien. Den benytter sig blot af den forventede frekvens  $P(X = x_i)$  i stedet for:

$$E(X) = \sum_{i=1}^n x_i * P(X = x_i) \quad (1.3)$$

Udfaldsrummene  $U_1 = \{1,2,3,4,5,6,7,8,9\}$  og  $U_2 = \{5,5,5,5,5,5,5,5,5\}$  har begge en middelværdien  $\mu$  på 5. Middelværdien siger imidlertid ikke særlig meget om, hvordan observationerne spreder sig i forhold til middelværdien. Det er derfor nødvendigt at indføre begrebet varians,  $Var$ . Varians for en

diskret stokastisk variabel og er defineret ved middelværdien for den kvadratiske afvigelse af observationerne fra middelværdien. Udtrykt som formel er det givet:

$$Var(X) = \sum_{i=1}^n (x_i - E(X))^2 * P(X = x_i) \quad (1.4)$$

Dette kan også omskrives til:

$$Var(X) = E[(X - E(X))^2] \quad (1.5)$$

Spredningen eller standardafvigelsen  $\sigma$  er lig kvadratrod af variansen:

$$\sigma = \sqrt{Var(X)} \quad (1.6)$$

Derfor opskrives variansen også nogle gange som standardafvigelsen i anden  $\sigma^2$ .

Bemærk at disse udtryk kun gælder for en diskret stokastisk variabel. Hvis den stokastiske variabel i stedet var kontinuerlig så var  $n$  ikke kendt og så benyttes i stedet lidt andre formler. Det vil jeg komme ind på i afsnittet ”Implementering af Kalman filter”.<sup>5,6,7</sup>

## Kovarians

Kovariansen er et mål for i hvor høj grad to stokastiske variabler varierer sammen eller modsatrettet hinanden. Kovariansen defineres ved middelværdien af produktet af de to variablers afvigelse fra middelværdien:

$$\text{cov}(X, Y) = E[(X - E(X)) * (Y - E(Y))] \quad (2.1)$$

Hvis man ønsker at finde kovariansen af den samme stokastiske variabel, er udtrykket deraf givet ved:

$$\text{cov}(X, X) = E[(X - E(X)) * (X - E(X))] = E[(X - E(X))^2] \quad (2.2)$$

Ifølge (1.5) er kovariansen mellem den samme stokastiske variabel derved lig variansen:<sup>4</sup>

$$\text{cov}(X, X) = Var(X) \quad (2.3)$$

<sup>5</sup> www.matematiksider.dk: Normalfordeling. Udgivet af Erik Vestergaard

<sup>6</sup> Bohnstedt, Allan m.fl.: MAT A, htx. Side 236-284. 1. udg. Systime, 2008

<sup>7</sup> Carstensen, Jens og Jesper Frandsen: MAT 3A. Side 194-214 og 230-258. 1. udg. Systime, 1999

## Normalfordeling

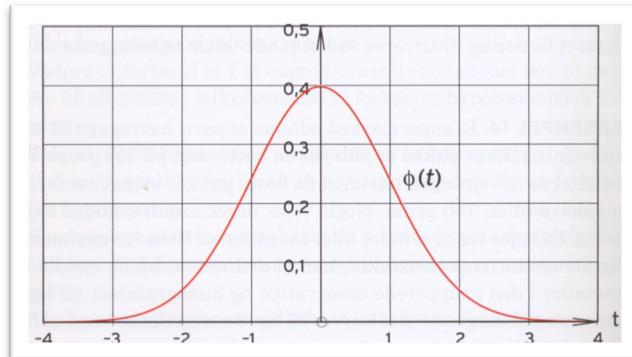
Som tidligere omtalt fordeler mange data sig i en såkaldt normalfordeling. Hvis man foretog en række observationerne, der fordelte sig som en normalfordeling – det kunne være højden på et uendeligt stort antal personer – vil man se, at disse data fordelte sig som en klokkeformet kurve, også kaldet en Gauss-kurve efter matematikeren Carl Friedrich Gauss<sup>7</sup>. Han viste, at en normalfordeling med observationerne hen ad x-aksen og sandsynligheden (frekvensen) for denne observation op ad y-aksen, har følgende tæthedsfunktion:

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi} * \sigma} * e^{-0,5 * \left(\frac{x-\mu}{\sigma}\right)^2} \quad (3.1)$$

Ved hjælp af tæthedsfunktionen kan man altså beregne frekvensen for den givne observation med middelværdien  $\mu$  og standardafvigelsen  $\sigma$ .

En normalfordeling med middelværdien  $\mu$  0 og standardafvigelsen  $\sigma$  1, kaldes en standard normalfordeling. Hvis  $X$  er normalfordelt med middelværdien  $\mu$  og variansen  $\sigma^2$  opskrives dette  $X \sim N(\mu, \sigma^2)$ .

Standard normalfordelingen skrives herfor  $X \sim N(0,1)$ . Gauss-kurven for standard normalfordelingen kan ses i Figur 2.



**Figur 2 – Gauss-kurven for standard normalfordelingen.**  
Kilde: Carstensen, Jens og Jesper Frandsen: MAT 3A, side 278

Bemærk at x-aksen er en asymptote. Derved ses det, at arealet under grafen er lig 1:

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi} * \sigma} * e^{-0,5 * \left(\frac{x-\mu}{\sigma}\right)^2} dx = 1 \quad (3.2)$$

Man kan altså finde mængden af observationer, der er indenfor et givet interval, ved at tage integralet fra minus uendelig til et reelt tal  $x \in R$  af frekvensfunktionen for normalfordelingen. Dette kaldes også fordelingsfunktionen og er udtrykt således:

$$F(x, \mu, \sigma) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi} * \sigma} * e^{-0,5 * \left(\frac{x-\mu}{\sigma}\right)^2} dx \quad (3.3)$$

Denne funktion beregner således den kumulerede frekvens fra minus uendelig til  $x$ . Fordelingsfunktionen er således stamfunktion til tæthedsfunktionen. På Figur 3 ses grafen for den kumulerede frekvens for standard normalfordelingen. Ved at bruge fordelingsfunktionen kan man altså beregne sandsynligheden for at  $X$  ligger under eller indenfor et givet interval. Værdien hvor 25%, 50% og 75% af alle observationer ligger kaldes henholdsvis 1. kvartil, 2. kvartil og 3. kvartil.

Det ses, at sandsynligheden for at en observation i en normalfordeling ligger indenfor intervallerne  $\pm 1$ ,  $\pm 2$ ,  $\pm 3$  og  $\pm 4$  standardafvigelser  $\sigma$  fra middelværdien  $\mu$  er:

$$F(1,0,1) - F(-1,0,1) = 68,27\%$$

$$F(2,0,1) - F(-2,0,1) = 95,45\%$$

$$F(3,0,1) - F(-3,0,1) = 99,73\%$$

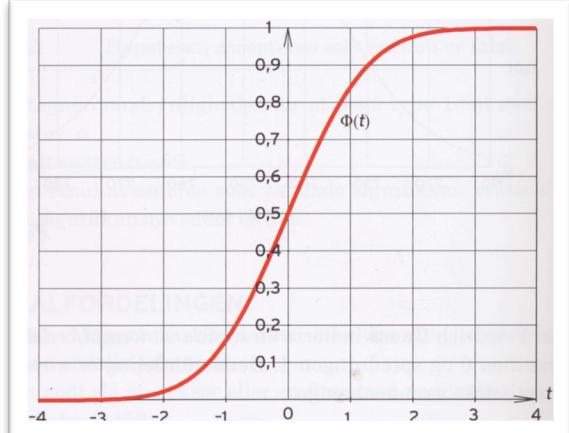
$$F(4,0,1) - F(-4,0,1) = 99,99\% \approx 100\%$$

Det ses altså at stort set alle observationer vil være at finde indenfor  $\pm 4$  af standardafvigelsen  $\sigma$  fra middelværdien  $\mu$ . Denne fordeling af observationer gælder således for alle normalfordelinger. Figur 4 illustrerer sammenhængen mellem standardafvigelsen og observationernes fordeling.

Tæthedsfunktionen og fordelingsfunktionen ved standard normal fordelingen angives også henholdsvis  $\varphi(x)$  og  $\Phi(x)$ .<sup>7</sup>

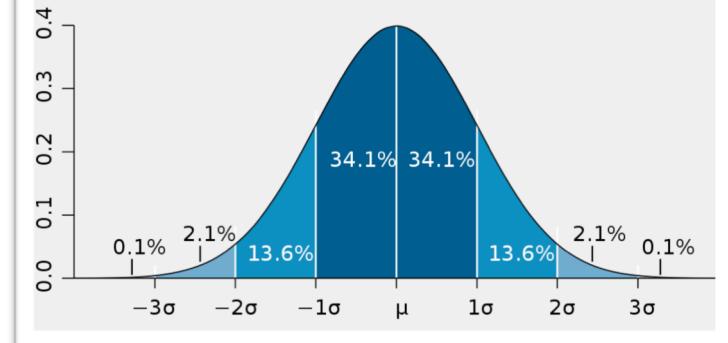
### Fraktdiagram

For at undersøge om et datasæt er normalfordelt bruges et såkaldt fraktdiagram. Dette udnytter det faktum, at hvis en stokastisk variabel  $Y$  er normalfordelt  $Y \sim N(\mu, \sigma^2)$ , så må den transformerede standard



**Figur 3 – Kurve for den kumulerede frekvens af standard normalfordelingen.**

Kilde: Carstensen, Jens og Jesper Frandsen: MAT 3A, side 278



**Figur 4 - Sammenhæng mellem standardafvigelse og observationers fordeling.**

Kilde: <http://en.wikipedia.org>

normalfordelte stokastiske variabel  $X \sim N(0,1)$  være givet ved:

$$X = \frac{Y - \mu}{\sigma} \quad (4.1)$$

Dette kan omskrives til følgende:

$$Y = \sigma * X + \mu \quad (4.2)$$

Ved at indtægne dette i et (X,Y)-koordinatsystem vil punkterne således ligge på en lige linje med standardafvigelsen  $\sigma$  som hældningskoefficienten og krydse y-aksen i punktet  $(0,\mu)$ .<sup>8</sup>

I et fraktildiagram er Y således vores observationer. X beregnes ud fra den kumulerede frekvens af observationerne vha. den inverse fordelingsfunktion til standard normalfordelingen også kaldet fraktilfunktionen til standard normalfordelingen:

$$X = \phi^{-1}(F(x)) \quad (4.3)$$

Fraktilfunktionen er imidlertid meget kompliceret og angives typisk blot  $\phi^{-1}(p)$ <sup>9</sup>. Man kan altså vha. fraktilfunktionen finde den observation (fraktil), som den angivne kumulerede frekvens ville svare til, hvis den var perfekt standard normalfordelt.

Hvis den undersøgte stokastiske variabel er normalfordelt, vil den således forme en ret linje med fraktilerne fra standard normalfordelingen. Det er derfor hensigtsmæssigt at bruge lineær regression på grafen for at undersøge, hvor tæt disse punkter ligger på en ret linje og ud fra det vurdere, om det er en normalfordeling eller ej.<sup>5</sup> Et konkret eksempel kan ses i afsnittet ”Analyse af målestøjen”.

### Redegørelse for Kalman filteret

Kalman filterets formål er at lave et estimat af en ukendt variabel  $x$  til tiden  $k$ . Udtrykket for  $x$  til tiden  $k$ , kan skrives således:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (5.1)$$

Hvor  $x_k$  og  $x_{k-1}$  er et udtryk for tilstanden til henholdsvis tiden  $k$  og  $k-1$ ,  $u_{k-1}$  er et valgfrit kontrol input til tiden  $k-1$  og  $w_{k-1}$  er et udtryk for processtøjen til tiden  $k-1$ . Både  $x$ ,  $u$  og  $w$  er alle vektorer

<sup>8</sup> Laursen, Jan og John Olesen: Grundlæggende statistik. Side 261-263. 4. udg. Forlaget JaVol, 2006

<sup>9</sup> Følgende kilde angiver et udtryk for fraktilfunktionen:

[http://en.wikipedia.org/wiki/Normal\\_distribution#Quantile\\_function](http://en.wikipedia.org/wiki/Normal_distribution#Quantile_function)

og  $A$  og  $B$  er kendte matricer<sup>10</sup>. Det er imidlertid ikke muligt at observere eller måle  $x_k$ . Denne er i stedet observeret i form af  $z_k$ , der er givet ved:

$$z_k = Hx_k + v_k \quad (5.2)$$

$z_k$  er målingen,  $x_k$  er tilstanden og  $v_k$  er målestøjen. Derudover er alle vektorer til tiden  $k$ .  $H$  er et kendt matrice.

Kort sagt er man altså interesseret i at bestemme  $x_k$ , men da denne er påvirket af støj, er det kun muligt at observere målingen  $z_k$ .

Det er værd at bemærke, at  $x_k$  er en såkaldt Markovproces, da den således kun afhængig af den forrige tilstand  $x_{k-1}$  og ikke de foreliggende tilstænde.  $z_k$  er samtidig også en Markovproces, hvor den kun afhænger af  $x_k$ . Dette kan udtrykkes således:

$$x_k | x_{k-1}, x_{k-2}, \dots \quad (5.3)$$

$$z_k | x_k, x_{k-1}, \dots \quad (5.4)$$

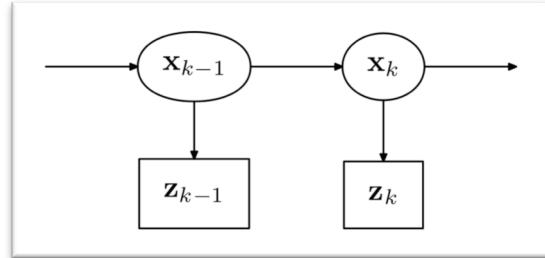
Dette læses  $x_k$  er givet  $x_{k-1}$ , som er givet  $x_{k-2}$  osv. (5.3) og (5.4) er et udtryk for en såkaldt skjult Markovkæde, da  $x_k$  ikke kan observeres. Figur 5 illustrerer principippet bag den skjulte Markovkæde for Kalman filteret. For mere information henvises til følgende fodnoter<sup>11,14</sup>.

Processtøjen  $w$  og målestøjen  $v$  er begge uafhængige stokastiske normalfordelte vektorer med en middelværdi  $\mu$  på 0:

$$p(w) \sim N(0, Q) \quad (5.5)$$

$$p(v) \sim N(0, R) \quad (5.6)$$

Det ses heraf, at  $x_k$  givet  $x_{k-1}$  må være normalfordelt med følgende middelværdi og varians:



Figur 5 – Illustration af den skjult Markovkæde for Kalman filteret.

Kilde: <http://en.wikipedia.org>

<sup>10</sup> Da det kommende eksempel i afsnittet ”Implementering af Kalman filter” er én-dimensional, er alle matricer blot 1x1 matricer og har derfor ingen betydning har for implementeringen af Kalman filteret i dette tilfælde. Herfor vil jeg således ikke komme ind på matriceregning og dets indhold.

<sup>11</sup> Markov-proces. Udgivet af Gyldendal. Internetadresse:

[http://www.denstoredanske.dk/It\\_teknik\\_og\\_naturvidenskab/Matematik\\_og\\_statistik/Sandsynlighedsregning\\_og\\_stokastisk\\_proces/Markov-proces](http://www.denstoredanske.dk/It_teknik_og_naturvidenskab/Matematik_og_statistik/Sandsynlighedsregning_og_stokastisk_proces/Markov-proces)

$$p(x_k | x_{k-1}) \sim N(Ax_{k-1} + Bu_{k-1}, Q) \quad (5.7)$$

Ligeledes gælder det for  $z_k$  givet  $x_k$  at:

$$p(z_k | x_k) \sim N(Hx_k, R) \quad (5.8)$$

Hvor  $p$  angiver tæthedsfunktionen for henholdsvis  $x_k$  og  $z_k$ .

$Q$  og  $R$  er henholdsvis proces- og målestøj kovarians matricer. Hvor kovarians matricen er givet ved henholdsvis:

$$Q = E[(w - E(w))(w - E(w))^T] \quad (5.9)$$

$$R = E[(v - E(v))(v - E(v))^T] \quad (5.10)$$

Da middelværdien  $\mu$  for begge udtryk er lig 0, kan disse omskrives til:

$$Q = E[w w^T] \quad (5.11)$$

$$R = E[v v^T] \quad (5.12)$$

Hvor  $T$  angiver den transponerede matrice. Hvis der arbejdes i én dimension er den transponerede vektor  $w^T$  og  $v^T$  den samme som henholdsvis vektor  $w$  og  $v$ . Derved kan udtrykket omskrives til følgende, hvis man arbejder med et én-dimensionalt system:

$$Q = E[w^2] \quad (5.13)$$

$$R = E[v^2] \quad (5.14)$$

Ifølge (2.3) er dette derved lig variansen. Hermed er proces- og målestøjen for et én-dimensional system givet ved variansen af henholdsvis proces- og målestøjen:

$$Q = Var(w) \quad (5.15)$$

$$R = Var(v) \quad (5.16)$$

En anden vigtig egenskab ved en Markovkæde er, at man, ved at kende de tidligere tilstande, kan beregne sandsynligheden for en given observation i fremtiden. Man kan altså forudsige, hvad systemets tilstand er i fremtiden ud fra de tidligere tilstande<sup>4</sup>. Derfor vil man altså i teorien ikke blot kunne beregne, hvad systemets tilstand er, men også forudsige hvad den sandsynligvis vil være i fremtiden.

## Udtrykket for Kalman filteret

Vha. af meget komplicerede udledning kan (5.1) og (5.2) omskrives til<sup>12</sup>:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (6.1)$$

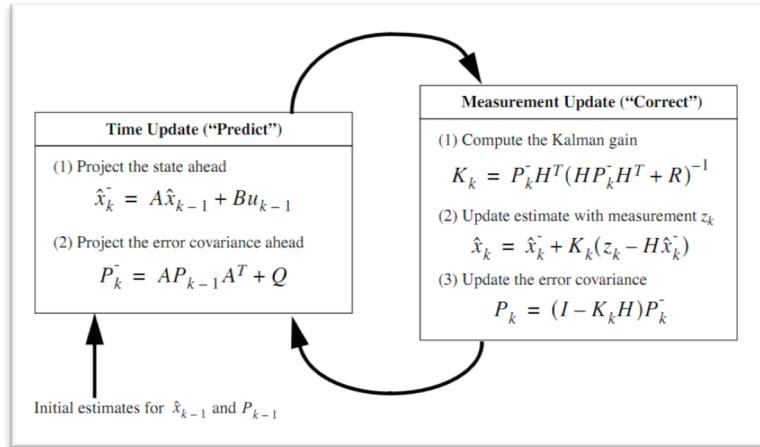
$$P_k^- = AP_{k-1}A^T + Q \quad (6.2)$$

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (6.3)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (6.4)$$

$$P_k = (I - K_k H)P_k^- \quad (6.5)$$

Disse udtryk kan således bruges i alle situationer, der kan opstilles som et lineært system. Dette kunne fx være, at man ønskede en meget nøjagtig estimering af en rakets position ud fra rakettens hastighed og acceleration. I dette tilfælde har man således brug for matriceregning for at kunne omregne fra henholdsvis  $\frac{m}{s}$  og  $\frac{m}{s^2}$  til m – man regner altså i flere dimensioner. Figur 6 illustrerer Kalman filterets virkemåde.



**Figur 6 – Illustration af de forskellige trin i Kalman filteret.**  
Kilde: Welch, Greg og Gary Bishop: *An Introduction to the Kalman Filter*, side 6

Disse udtryk kan dog simplificeres en del, hvis det ønskede estimat er i samme dimension, som den anvendte input. Herved kan matricerne A,B og H udelades og Q og R erstattes af henholdsvis variansen af proces- og målestøjen, som vist i (5.15) og (5.16). Derved kan udtrykkende omskrives til følgende hvis man regner i én dimension:

$$\hat{x}_k^- = \hat{x}_{k-1} + u_{k-1} \quad (6.6)$$

$$P_k^- = P_{k-1} + Var(w) \quad (6.7)$$

$$K_k = P_k^- (P_k^- + Var(v))^{-1} \quad (6.8)$$

<sup>12</sup> For mere information se fodnote 14 og 15

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-) \quad (6.9)$$

$$P_k = (1 - K_k)P_k^- \quad (6.10)$$

Hvor  $\hat{x}_k^-$  angiver estimatet af  $x$  á priori, dvs. inden målingerne, til tiden  $k$ , mens  $\hat{x}_{k-1}$  angiver estimat for  $x$  fra sidste observation.  $u$  er et ekstra kontrol input og derfor kan (6.6) udelades hvis dette ikke benyttes. Bemærk at hvis (6.6) ikke medtages skal  $\hat{x}_k^-$  erstattes af  $\hat{x}_{k-1}$  i ligningen (6.9).

$P_k^-$  er et udtryk for den estimerede fejl-varians før målingen. Denne er således givet ved summen af den tidligere fejl-varians  $P_{k-1}$  og variansen af processtøjen  $Var(w)$ .

$K_k$  er populært sagt et udtryk for, hvor meget man kan stole på den kommende måling til tiden  $k$ . Denne kaldes også for "Kalman Gain". Det ses, at hvis variansen af målestøjen  $Var(v)$  er stor, vægtes inputtet således ikke særlig meget, hvorimod hvis den er meget lille, vægtes den højere, da  $K_k$  dermed er større.

$\hat{x}_k$  er den estimerede værdi af  $x_k$ . Denne er således lig summen af den á priori estimerede værdi  $\hat{x}_k^-$  og produktet af  $K_k$  og forskellen mellem inputtet  $z_k$  og  $\hat{x}_k^-$ .

$P_k$  er et udtryk for den estimerede fejl-varians á posteriori, dvs. efter målinger, til tiden  $k$ .

Hatten  $\wedge$  over  $x$ -erne angiver, at det er et estimat af tilstanden  $x_k$ , da denne jo ikke kan observeres.  $\hat{x}_k$  er således et udtryk for middelværdien af  $x_k$  given inputtet  $z_k$ , som er en Markovkæde. Derved er  $\hat{x}_k$  givet ved:

$$\hat{x}_k = E[x_k | z_k, z_{k-1}, z_{k-2}, \dots] \quad (6.11)$$

Estimeringen er således et udtryk for den betingede middelværdi af  $x_k$ . Hvor den betingede middelværdi er defineret ud fra det bedste skøn af en ubekendt stokastisk variabel ud fra data<sup>13</sup>. Kort sagt udregner Kalman filteret en betinget middelværdi ud som et estimat for systemet uobserverede tilstand  $x_k$  ud fra inputtet  $z_k$ .

For mere information omkring teorien bag Kalman filteret henvises til følgende fodnoter.<sup>13,14,15,16</sup>

<sup>13</sup>Betinget Middelværdi. Udgivet af Gyldendal. Internetadresse:

[http://www.denstoreddanske.dk/It\\_teknik\\_og\\_naturvidenskab/Matematik\\_og\\_statistik/Sandsynlighedsregning\\_og\\_stokastisk\\_proces/betinget\\_middelv%C3%A6rdi](http://www.denstoreddanske.dk/It_teknik_og_naturvidenskab/Matematik_og_statistik/Sandsynlighedsregning_og_stokastisk_proces/betinget_middelv%C3%A6rdi)

<sup>14</sup>Vaseghi, Saeed V.: Advanced Digital Signal Processing and Noise Reduction. Side 147-172 og 193-206. 4. udg. Wiley, 2008

<sup>15</sup>Haykin, Simon: Modern Filters. Side 349-375. 1. udg. Macmillan Publishing Company, 1989

## Analog til Digital konvertering

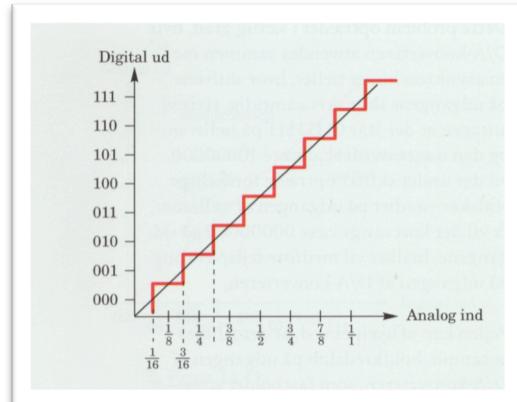
Formålet med en AD-konverter er at konvertere et analog signal til et digitalt signal, som fx kan aflæses af en mikroprocessor i form af en binær værdi. Dette kunne fx være en 10-bit konverter med en referencespænding på 5V. Denne vil således fordele 0V-5V på et interval bestående af 1024 trin ( $2^{10}$ ). Svarende til 0-1023 på udgangen. Det ses, at længden af hvert trin derfor er givet ved:

$$\frac{5V}{1024} = 4,883mV \quad (7.1)$$

Det ses altså, at den binære værdi kun er en tilnærmelse af den analog spænding. Derudover ses det at den analoge spænding er kontinuerlig, da den kan optræde som enhver værdi i intervallet. Derimod er den binære værdi en diskret variabel, da udfaldsrummet U er kendt, som beskrevet i afsnittet "Stokastisk variabler". En AD-konverter vil således indføre støj på det virkelige signal. Figur 7 illustrerer princippet for en 3-bits AD-konverter, hvor den lige linje angiver den faktiske spænding, mens den røde angiver værdien, som den angivne spænding ville svare til for en 3-bit AD-konverter.

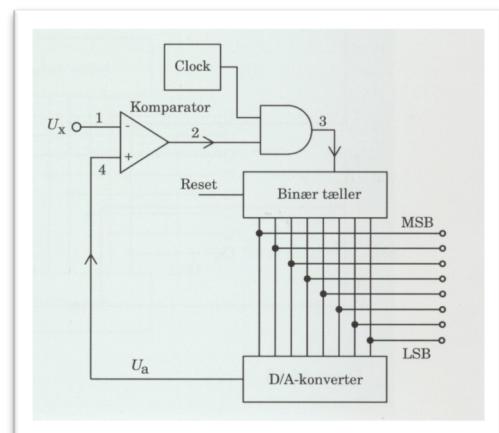
For at konvertere den analoge spænding til en binær værdi sammenlignes denne med en kendt spænding. Ved at bruge en komperator hvor den kendte spænding indføres på minus og den ukendte spænding på plus, vil udgangen henholdsvis være 1, hvis den ukendte spænding er større end den kendte spænding, mens den vil være 0, hvis den er mindre end den kendte spænding.

Derved vil der, ved at indføre en stigende kendt spænding, ske et skift fra 1 til 0. Et kredsløb, der netop gør dette, kan ses i Figur 8. Det fungerer ved at for hver clockimpuls, der sendes ind, tæller den binære tæller én op. Denne værdi omsættes af DA-konverteren<sup>17</sup> til en analog spænding som



Figur 7 - Princippet for en 3-bit AD-konverter.

Kilde: Andersen, Leif Møller og Kim Rundblad: Digital og mikroprocessor teknik, side 194



Figur 8 – AD-konverter med rampetæller.

Kilde: Andersen, Leif Møller og Kim Rundblad: Digital og mikroprocessor teknik, side 194

<sup>16</sup> Betinget fordelinger. Udgivet af Københavns Universitet. Internetadresse:

<http://www.math.ku.dk/~erhansen/stat2b/notater/notat4-3.pdf>

<sup>17</sup> DA står for "Digital to Analog", en DA-konverter er således et digitalt signal til en analog spænding

sendes ind på komparatorens input. Hvis den kendte spænding således overskridt den ukendte spænding  $U_x$ , er udgangssignalet fra AND-gaten 0 og tælleren vil derfor ikke tællere videre. Bits'ne vil således være en udtryk for den analoge spænding  $U_x$ .

MSB og LSB står for henholdsvis "Most Significant Bit" og "Least Significant Bit", hvor MSB således angiver den højeste bit og LSB angiver den laveste bit. Hvis man kender referencespænding kan en given værdi let omregnes til en spænding. Hvis man fx har et output fra en 8-bit AD-konverter på 10110010 svarende til 178 og referencespændingen er 3,3V, vil spændingen kunne beregnes således:

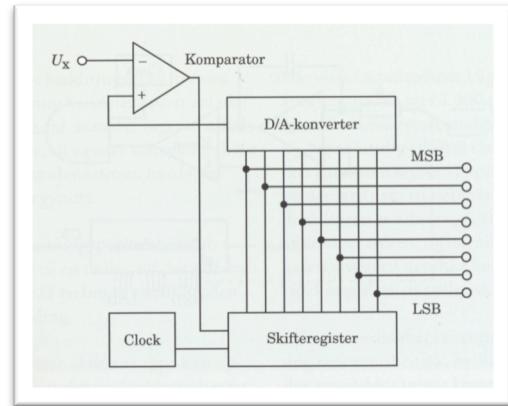
$$\frac{178}{255} * 3,3V = 2,3V \quad (7.2)$$

Bemærk at dette er ikke den sande værdi, da det jo er begrænset af opløsningen. Derfor vil aflæsningen således have en unøjagtighed på  $\pm 0,5\text{LSB}$ , svarende til:

$$\frac{0,5}{255} * 3,3V = 6,5mV \quad (7.3)$$

Spændingen vil altså være  $2,3V \pm 6,5\text{mV}$ . Dette kaldes kvantiseringsstøjen og skyldes afrundingen af signalværdien – i dette tilfælde spændingen – til et digital signal<sup>18</sup>. For mere information se afsnittet "Implementering af Kalman filter".

Problemet ved denne type AD-konverter er, at konverteringstiden afhænger af den ukendte spændings størrelse, da en spænding på 0V således kun vil tage 1 clockimpuls, mens en spænding på 5V vil tage hele 1024 clockimpulser. Derfor bruges denne type AD-konverter ikke i praksis. En anden type, der altid tager den samme tidsmængde og samtidig foregår væsentlig hurtigere, er successiv approksimations AD-konverteren. Figur 9 beskriver principippet bag denne type AD-konverter. I stedet for at tælle nede fra og op er den udstyret med en skifteredister, der skifter én bit af gangen. Det fungerer ved, at alle bits først sættes til 1 og derefter sammenlignes spændingen



**Figur 9 – Princip bag en successiv approksimation AD-konverter.**

Kilde: Andersen, Leif Møller og Kim Rundblad: Digital og mikroprocessor teknik, side 197

<sup>18</sup> Støj. Udgivet af Gyldendal. Internetadresse:

[http://www.denstoredanske.dk/I%2c\\_teknik\\_og\\_naturvidenskab/Elektronik%2c\\_teletrafik\\_og\\_kommunikation/Elektronik%2c\\_radio\\_og\\_tv/st%C3%B8j](http://www.denstoredanske.dk/I%2c_teknik_og_naturvidenskab/Elektronik%2c_teletrafik_og_kommunikation/Elektronik%2c_radio_og_tv/st%C3%B8j)

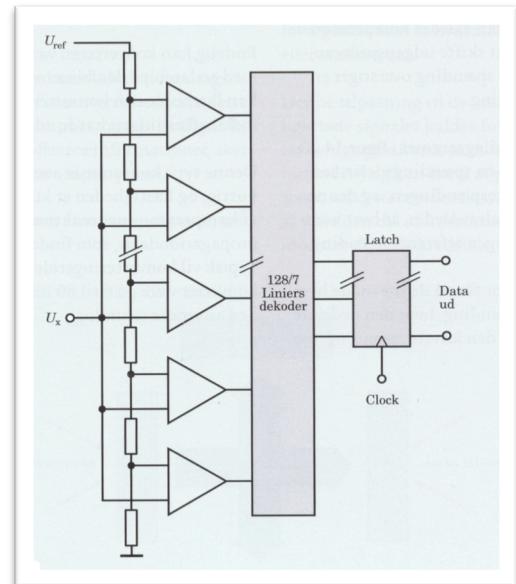
med den ukendte spænding  $U_x$ . Hvis komperatorens output derefter giver 1, er den ukendte spænding  $U_x$  altså større end referencespændingen. Hvis dette ikke er tilfældet vil outputtet i stedet være 0. I dette tilfælde skifter skifteredisteret MSB og igen sammenlignes spændingerne. Sådan fortsætter systemet fra MSB til LSB indtil den har været alle bits igennem. Efter dette kan den ukendte spændings  $U_x$  binære værdi aflæses på udgangen. Som det fremgår tager denne altid  $n$  clockimpulser, hvor  $n$  er AD-konverterens opløsning, således vil en 10-bit AD-konverter altid tage 10 clockimpulser.

En anden type er Flash AD-konverteren. Som navnet antyder er denne en utrolig hurtig AD-konverter. Figur 10 illustrerer principippet bag en Flash AD-konverter. Denne type er opbygget af en række spændingsdelere, der deler referencespændingen  $U_{ref}$  op i lige store intervaller. Disse spændinger er derefter koblet til hver deres komparator. På den anden indgang er tilført den ukendte spænding  $U_x$ . Udgangene på komperatorerne vil derved være 0, hvis referencespænding er større eller lig den ukendte spænding  $U_x$ . Omvendt vil udgangen være 1, hvis referencespændingen er mindre end den ukendte spænding  $U_x$ . Ved at føre udgange af komperatorerne ind i en 128/7 dekoder, der omsætter værdierne til en binær værdi, kan den ukendte spænding  $U_x$  bestemmes. Derudover kan der tilføres en latch på udgangen, så den binære værdi fastholdes.

Det ses heraf, at opløsningen er givet spændingsfaldet mellem hvert interval i spændingsdelen. For at hæve opløsningen er der derfor nødvendigt at gøre spændings-intervallerne mindre imellem hvert led. Dette medfører dog, at man skal tilføre en komparator for hver gang, man ønsker opløsningen hævet med blot én værdi, men til gengæld har man en ekstrem hurtig AD-konverter, der kun afhænger af gateforsinkelsen i komperatorerne, dekoderen og eventuelt latchen.<sup>19</sup>

### Normalfordelt hvid støj

Som omtalt i afsnittet ”Redegørelse for Kalman filteret” kræver Kalman filteret, at målestøjen er en normalfordelt stokastisk variabel med middelværdien  $\mu = 0$ , og hvor variansen er kendt. Det viser sig



Figur 10 – Principippet bag en Flash AD-konverter.

Kilde: Andersen, Leif Møller og Kim Rundblad: Digital og mikroprocessor teknik, side 200

<sup>19</sup> Andersen, Leif Møller og Kim Rundblad: Digital og mikroprocessor teknik. Side 194-200

imidlertid, at naturligt støj typisk er normalfordelt med en middelværdi omkring den målte signal – middelværdien  $\mu$  er altså 0. Dette kaldes også normalfordelt hvid støj og har lige stor intensitet i hele frekvensområdet<sup>20</sup>. Dette kunne fx stamme fra en motor, hvor man samtidig ønsker at kende temperaturen indeni vha. af en analog temperatursensor. Denne AD-konverteringen ville dog ikke give noget præcist bud af temperaturen, da målestøjen vil påvirke konverteringen.

Det er samtidig relativt let, at simulere normalfordelt hvid støj. I det kommende afsnit vil jeg således benytte mig af 60 sekunders normalfordelt hvid støj generet af en gratis tone generator fundet på internettet<sup>21</sup>. For at høre lydfilen henvises til ”60s - Gaussian White Noise.wav” i ”Bilag 7 – CD-ROM”.

### Analyse af målestøjen

Før målestøjen kan filtreres fra via Kalman filteret, skal den være normalfordelt med middelværdien  $\mu = 0$  og variansen for målestøjen skal være kendt. Det er derfor relevant, at undersøge om den generede lydfil opfylder disse krav. Dette kan gøres med et fraktildiagram, som er beskrevet i afsnittet ”Fraktildiagram”. Hvis dette er tilfældet, kan variansen for normalfordelingen derefter bestemmes ud fra målinger, som beskrevet i afsnittet ”Frekvens, middelværdi, varians og spredning”.

Det første step er at implementere støjen ovenpå signalet, således at middelværdien  $\mu$  er lig 0. Dette gøres med et højpasfilter, der ”lægger” AC signalet, som stammer fra lydfilen, oveni DC signalet, som stammer fra en strømforsyning. Højpasfilteret vil samtidig filtrere de lavere frekvenser fra, bestemt ud fra overgangsfrekvens  $f_0$  for højpasfilteret. Denne er givet:

$$f_0 = \frac{1}{2\pi * R * C} \quad (8.1)$$

Ifølge den skematiske tegning, se ”Bilag 1 – Skematiske tegning og billede af opstilling”, kan man således beregne overgangsfrekvensen  $f_0$  for kredsløbet til følgende:

$$f_0 = \frac{1}{2\pi * 8700\Omega * 1\mu F} = 18,29\text{Hz} \quad (8.2)$$

---

<sup>20</sup> Hvid Støj. Udgivet af Gyldendal. Internetadresse:[http://www.denstoredanske.dk/I%e5\\_teknik\\_og\\_naturvidenskab/Elektronik,\\_teletrafik\\_og\\_kommunikation/Kommunikationssteori\\_og\\_teletrafikteori/hvid\\_st%C3%B8j](http://www.denstoredanske.dk/I%e5_teknik_og_naturvidenskab/Elektronik,_teletrafik_og_kommunikation/Kommunikationssteori_og_teletrafikteori/hvid_st%C3%B8j)

<sup>21</sup> George W. Taylor: Tone generator. <http://www.tropicalcoder.com/AudioTestFileGen.htm>

Højpasfilteret fungerer således, at det reducerer amplituden af frekvenser under overgangsfrekvensen  $f_0$ . Højpasfilteret reducerer altså støjen for signaler med frekvenser under 18,29Hz, til sammenligning har lydsignalet en frekvens på 44100Hz.<sup>22</sup>

Målinger foretages med en Arduino Mega<sup>23</sup>, som er en Open Source mikrocontroller-platform. Koden brugt til at foretage målingerne kan ses i ”Bilag 2 – Arduino kode: Analyse af målestøj”.

Fraktdiagrammet for målingen kan ses i Figur 11, mens normalfordelingskurven kan ses i ”Bilag 3 – Normalfordelingskurve for målestøjen”. Da forklaringsgraden  $R^2$  er meget tæt på 1 må målestøjen altså siges at være normalfordelt – det ses også at regressionens lignings hældningskoefficient  $a$  og  $b$  er meget tæt på henholdsvis den beregnede standardafvigelse og middelværdi.

Middelværdien og variansen af målestøjen er beregnet til følgende:

$$\mu = 566,433 \quad (8.3)$$

$$Var(v) = 1636,931 \quad (8.4)$$

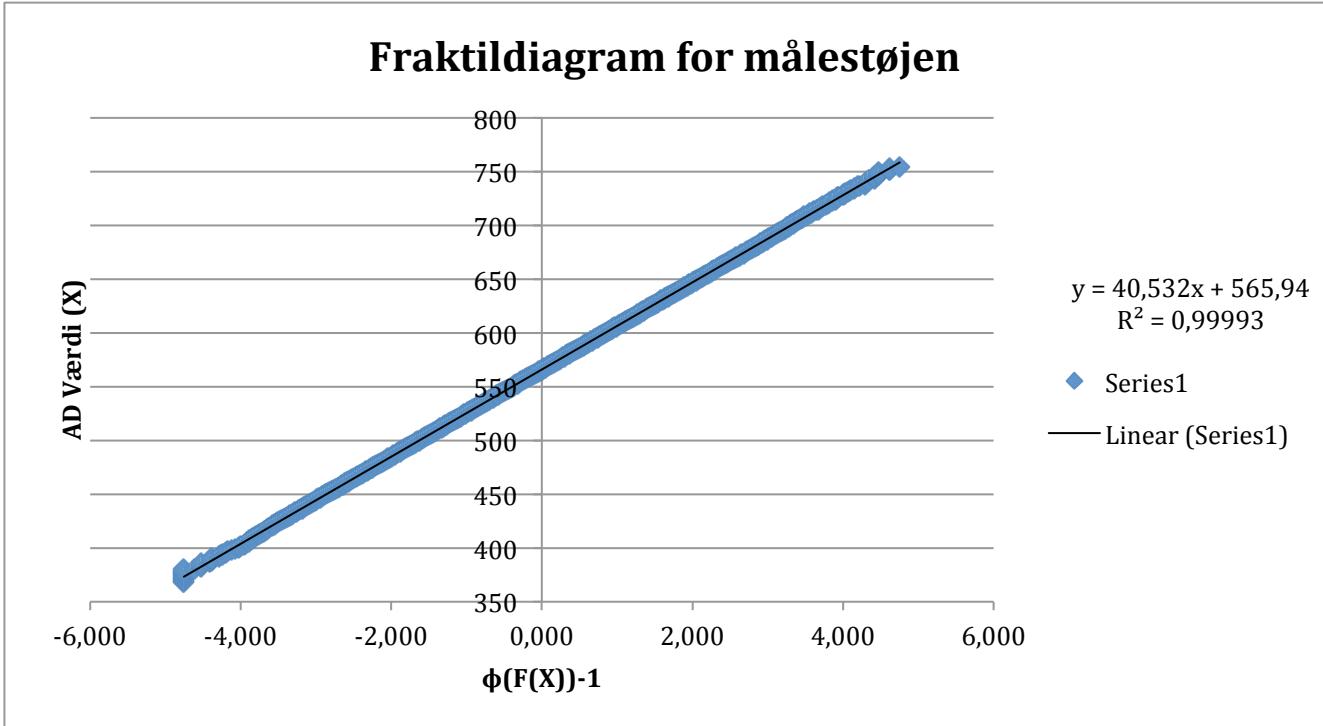
Hvor middelværdien svarer til  $2,768V\left(\frac{566,433}{1023} * 5V\right)$ . Til sammenligning målte mit multimeter strømforsyningen til 2,765. Vi kan altså konkludere at målestøjen er normalfordelt med følgende middelværdi  $\mu$  og varians:

$$v \sim N(0; 1636,931) \quad (8.5)$$

Målingerne og alle beregninger kan ses i ”Bilag 7 – CD-ROM” i filen ”Analyse af Målestøj.xlsx”.

<sup>22</sup> Rasmussen, Egon: Analog Teknik. Side 247-248. 2. udg. Industriens forlag, 2001

<sup>23</sup> Arduino Mega. Udgivet af Arduino. Internetadresse: <http://arduino.cc/en/Main/ArduinoBoardMega>



**Figur 11 – Fraktildiagram for målestøjen**

Kilde: Se ”Analyse af Målestøj.xlsx” i ”Bilag 7 – CD-ROM”

### Implementering af Kalman filter

I det kommende afsnit vil jeg implementere Kalman filteret til AD-måling, så støjen filtreres fra.

Støjen består imidlertid ikke kun af målestøj, men også af processtøj. I dette tilfælde er dette primært i form kvantiseringsstøjen forårsaget af AD-konverteren, som omtalt i afsnittet ”Analog til Digital konvertering”. Derudover kan processtøjen også stamme fra støj internt i mikroprocessoren, denne er dog utrolig lille og derfor medregnes den ikke. Variansen af kvantiseringsstøjen beregnes dog en lille smule anderledes, da det er en ligefordelt kontinuerlig stokastisk variabel, da spændingen kan have uendelig mange udfald indenfor intervallet. Dette interval er således længden af hvert trin – se Figur 7. Ifølge (1.5) har vi, at variansen er givet ved følgende:

$$Var(X) = E[(X - E(X))^2] \quad (9.1)$$

Dette kan også omskrives til:<sup>24</sup>

$$Var(X) = E[(X - E(X))^2] = \int_a^b (x - \mu)^2 * f(x) dx \quad (9.2)$$

<sup>24</sup> Calculus Applied to Probability and Statistics. Udgivet af Zweig Media. Internetadresse: <http://www.zweigmedia.com/RealWorld/cprob/cprob3.html>

Hvor  $f(x)$  angiver tæthedsfunktionen, a og b indikerer intervallet. For en ligefordelt kontinuerlig variabel er tæthedsfunktionen givet ved.<sup>25</sup>

$$f(x) = \frac{1}{b-a} \quad a \leq x \leq b \quad (9.3)$$

Da længden af hvert trin er 1, må 0,5 inde derfor være centrum. Dette vil således også være middelværdien. Hvis man ser middelværdien som centrum, er intervallet derfor  $[-0,5;0,5]$  med en middelværdi  $\mu$  på 0. Variansen af kvantiseringsstøjen er derved:

$$Var(X) = \int_a^b (x - \mu)^2 * f(x) dx = \int_{-0,5}^{0,5} (x - 0)^2 * \frac{1}{0,5 - (-0,5)} dx = 0,083 \quad (9.4)$$

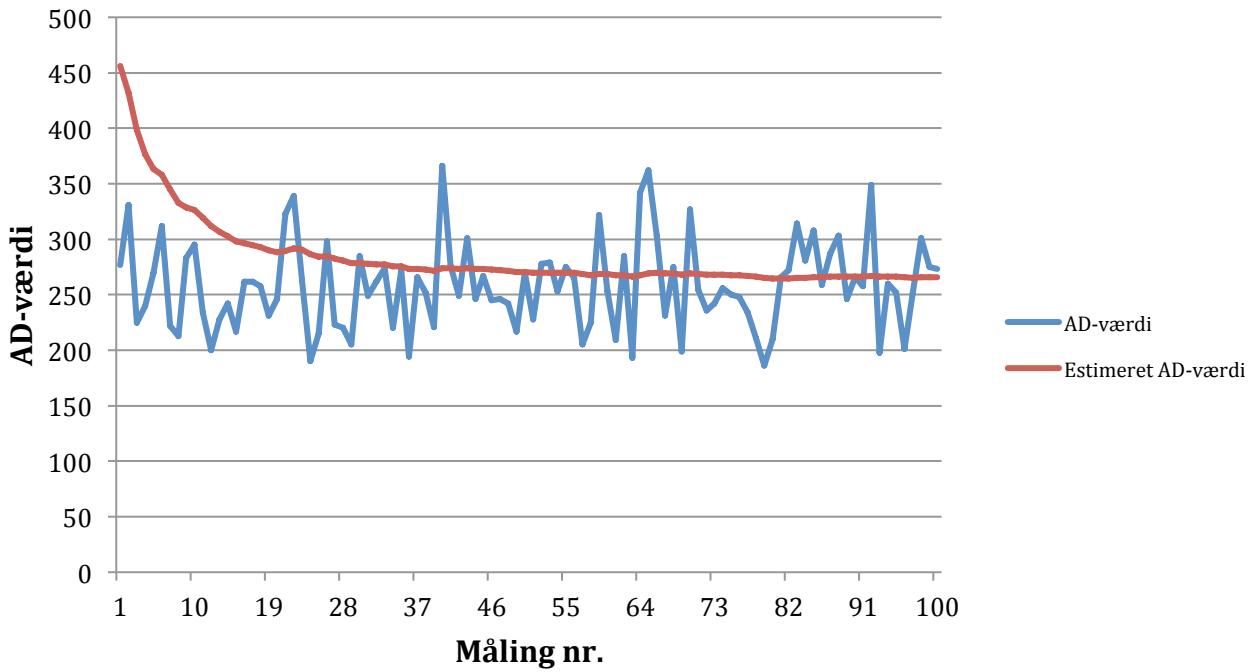
Denne kan dog ikke umiddelbart bruges i Kalman filteret, da det ikke er en normalfordeling. Derfor sættes variansen af processtøjen blot til 0. Koden for AD-konverteringen med implementeret Kalman filter kan ses i ”Bilag 4 – Arduino kode: AD-konverter med Kalman filter”.

Figur 12 viser tydeligt hvor effektivt Kalman filteret er til at fjerne støjen. Det ses, at Kalman filterets estimerede værdi, til sidst har stabiliseret sig omkring 265 og 259 efter henholdsvis 100 og 1000 målinger. Det estimerede output er altså ca.  $1,295V \left( \frac{265}{1023} * 5V \right)$  og  $1,266V \left( \frac{259}{1023} * 5V \right)$ . Til sammenligning målte mit multimeter strømforsyningens spændingen til 1,272V. Estimeringen er således utrolig tæt på det rigtige signal. Dette er selvfølgelig et optimalt eksempel, da variansen af støjen er kendt og processtøjen er utrolig lille. I praksis er støjen dog ofte ukendt og man bliver derfor nødt til at enten prøve eller beregne sig frem til denne. Derudover viser Figur 12 også, at Kalman filteret først skal indstille sig. Det tager således ca. 20 målinger før den har stabiliseret sig. Den er imidlertid først rigtig tæt på den faktiske værdi efter ca. 60 målinger. Da port-registrene for AD-konverteren i Arduino'en er sat til 10kHz vil det derfor tage ca. 6ms  $\left( \frac{60}{10kHz} \right)$  før man kan regne med estimatet. Da Kalman filteret i koden er indstillet til at starte på 2,5V, vil opstartstiden derfor maksimalt være ca. 12ms.

---

<sup>25</sup> Park, Sung Y. og Anil K. Bera: Maximum entropy autoregressive conditional heteroskedasticity mode. I: Journal of Econometric, 8.1.2009, s. 221

## ADC måling med Kalman filter



**Figur 12 – Grafen ovenfor viser ADC målingen. Det blå signal er værdien målt af AD-konverteren, mens det røde er den estimerede AD-værdi beregnet af Kalman filteret.**

Kilde: Se "AD med Kalman filter.xlsx" i "Bilag 7 – CD-ROM".

For at se alle 1000 ADC-målingerne henvises til filen "AD med Kalman filter.xlsx" i "Bilag 7 – CD-ROM".

Udover overstående forsøg prøvede jeg også at hæve processtøjen ved, at fjerne de fire nederste bits for AD-konverteren. Alle værdier ville således blive rundet ned til de øverste 6 bits. Derved er variansen givet ved:

$$Var(X) = \int_a^b (x - \mu)^2 * f(x)dx = \int_{-8}^8 (x - 0)^2 * \frac{1}{8 - (-8)} dx = 21,333 \quad (9.5)$$

Som man kan se i "Bilag 6 – Graf for Kalman filter top 8msb" er Kalman filteret ikke i stand til, at fjerne støjen når processtøjen er sat til 21,333. Hvis processtøjen derimod er sat til 0, filtreres støjen helt væk. Den estimerede værdi er dog på ca. 252 hvilket svarer til ca. otte under den faktiske værdi på ca. 260. Dette svarer helt til forventningen, da AD-værdierne i gennemsnit vil blive nedrundet med otte ved at sætte de fire bits til 0, da AD-konverteren derved minimum kan variere med 16.

Hvordan variansen for den ligefordelte kvantiseringsstøj konverteres til en normalfordelt processtøj er på nuværende tidspunkt uklart. Dette kunne således være interessant, at undersøge forholdet mellem processtøjen og kvantiseringsstøjen, for en AD-konverter, i fremtidige undersøgelser.

Det er værd at bemærke at et Kalman filter kun kan fjerne systematisk støj. Hvis støjen variere over tid, dvs. dens varians ændrer sig vil den således ikke være i stand til at estimere tilstanden præcist.

Derudover vil den ikke kunne fjerne støjen hvis det ikke har en middelværdi  $\mu$  på 0, som beskrevet i afsnittet ”Redegørelse for Kalman filteret”. Kalman filteret vil således fx ikke være i stand til at filtrere støjen væk fra et DC-signal, hvis støjens middelværdi ligger forskudt fra signalet. Der findes imidlertid andre type filtre, evt. kombineret med et Kalman filter, som kan løse nogle af disse problemer. Dette vil jeg dog ikke komme ind på i denne rapport, da det ikke har relevans for opgaven.

## Konklusion

I denne rapport er der gennemgået nogle grundbegreber indenfor statistik og sandsynlighedsregning, såsom varians, middelværdi, spredning mm og deres relation til en normalfordeling. Middelværdierne er et udtryk for det forventede gennemsnit. Varians og spredning/standardafvigelse er derimod et udtryk for hvor bred denne kurve er. Bredden af kurven siger således noget om hvordan fordelingen er, der er et udtryk for hvor meget værdierne i gennemsnit afviger fra middelværdien.

Den grundlæggende teori bag Kalman filteret er herefter gennemgået med henblik på, at udlede en fem ligninger der kunne bruges i til at filtrere støjen fra et DC-signal. Denne støj optræder ofte normalfordelt og blev genereret af en lydfil. Dette signal var herefter ”lagt” oven på et DC-signal vha. et højpasfilter. Denne støj blev derefter analyseret og vurderet om det var normalfordelt vha. et fraktildiagram. Målestøjen viste sig at være næsten fuldstændig normalfordelt.

Signalet indeholdt imidlertid også processtøj, i form af kvantiseringsstøjen, forsaget af AD-konverteren ved afrunding fra et analogt- til digitalt signal. Denne var dog ikke normalfordelt og kunne dermed ikke umiddelbart sættes ind i Kalman filteret, da Kalman filteret kræver, at støjen skal være normalfordelt med en middelværdi på 0. Det viste sig at kvantiseringsstøjen var så lille at den stort set ingen betydning havde. Vha. Kalman filteret lykkedes det at filtrere denne støj fra, så den estimerede værdi kom utroligt tæt på det virkelige signal. Det viste sig dog, at Kalman filteret ved opstart først skulle indstille sig, før den estimerede værdi lå meget tæt på den virkelige. Den maksimale opstartstid var således ca. 12ms, hvilket må siges at være mere end acceptabelt i langt de fleste situationer. Den estimerede værdi er således et udtryk for den betingede middelværdi af den uobserverede tilstand.

Jeg prøvede dog at hæve processtøjen ved at sætte de fire nederste bits til 0. Ved at skrue op for variansen for processtøjen stabiliserede Kalman filteret sig dog ikke og er derfor ikke anvendeligt. Det bedste resultat fremkom, overraskende nok, ved at sætte variansen af processtøjen til 0. Dette resul-

terede dog blot i, at den estimerede værdi var otte fra den virkelige. Dette hang dog fint sammen med, at AD-værdien i gennemsnit bliver forkortet med otte.

### Perspektivering

Kalman filteret har som beskrevet utroligt mange anvendelsesmuligheder indenfor moderne elektronik, hvor man ønsker at kende et systems tilstand uden at være i stand til at måle eller observere det på nogen anden måde. Det ville derfor være spændende at arbejde videre på Kalman filteret og forsøge at implementere det i andre digitale kredsløb. Det helt naturlig forløb ville være at implementere det i et flerdimensionalt system, såsom en robot udstyret med en GPS og et Wifi-modul. Ved at kombinere positionsdatene fra de to moduler ville man derved kunne opnå en meget nøjagtigt position for systemet. Derudover ville det også være interessant, at undersøge hvordan man kunne implementere kvantiseringsstøjen i Kalman filteret til AD-konvertering. Min vurdering er at det vil kræve et Kalman filter i flere dimensioner, for at konvertere kvantiseringsstøjen til en normalfordelt støj.

## Litteraturliste

### Bøger

4. Halse, Søren m.fl.: Matematik, Gyldendals Minilex. 1. udg. Gyldental, 2005
6. Bohnstedt, Allan m.fl.: MAT A, htx. Side 236-284. 1. udg. Systime, 2008
7. Carstensen, Jens og Jesper Frandsen: MAT 3A. Side 194-214 og 230-258. 1. udg. Systime, 1999
8. Laursen, Jan og John Olesen: Grundlæggende statistik. Side 261-263. 4. udg. Forlaget JaVol, 2006
14. Vaseghi, Saeed V.: Advanced Digital Signal Processing and Noise Reduction. Side 147-172 og 193-206. 4. udg. Wiley, 2008
15. Haykin, Simon: Modern Filters. Side 349-375. 1. udg. Macmillan Publishing Company, 1989
19. Andersen, Leif Møller og Kim Rundblad: Digital og mikroprocessor teknik. Side 194-200. 4. udg. Erhvervsskolernes Forlag, 2009
22. Rasmussen, Egon: Analog Teknik. Side 247-248. 2. udg. Industriens forlag, 2001

### Artikler

1. Simon, Dan: *Kalman Filtering*. I: <http://academic.csuohio.edu/>, 1.6.2011, s. 72-79. Internetadresse: <http://academic.csuohio.edu/simond/courses/eec644/kalman.pdf> Besøgt d. 13.12.2011
2. Steffensen, Michael: *Logistisk Kalman filter for kraftig nedbør*. I: [www.dmi.dk](http://www.dmi.dk/), 1.12.12, s. 2-2. Internetadresse: <http://www.dmi.dk/dmi/tr02-28.pdf> Besøgt d. 13.12.2011
3. Welch, Greg og Gary Bishop: *An Introduction to the Kalman Filter*. I: <http://www.cs.unc.edu/>, 24.07.2006, s. 1-15. Internetadresse: [http://www.cs.unc.edu/~welch/media/pdf/kalman\\_intro.pdf](http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf) Besøgt d. 13.12.2011
5. [www.matematiksider.dk](http://www.matematiksider.dk): Normalfordeling. Udgivet af Erik Vestergaard. Internetadresse: [http://www.matematikfysik.dk/mat/noter\\_tillaeg/normalfordeling.pdf](http://www.matematikfysik.dk/mat/noter_tillaeg/normalfordeling.pdf) - Besøgt d. 14.12.2011
25. Park, Sung Y. og Anil K. Bera: Maximum entropy autoregressive conditional heteroskedasticity mode. I: Journal of Econometric, 8.1.2009, s. 221. Internetadresse: <http://www.wise.xmu.edu.cn/Master/Download/..%5C..%5CUploadFiles%5Cpaper-masterdownload%5C2009519932327055475115776.pdf> - Besøgt d. 18.12.2011

### Internetsider

9. Wikipedia: Normal distribution. Internetadresse: [http://en.wikipedia.org/wiki/Normal\\_distribution#Quantile\\_function](http://en.wikipedia.org/wiki/Normal_distribution#Quantile_function) - Besøgt d. 16.12.2011
11. Markovproces. Udgivet af Gyldental. Internetadresse: [http://www.denstoredanske.dk/It\\_teknik\\_og\\_naturvidenskab/Matematik\\_og\\_statistik/Sandsynlighedsregning\\_og\\_stokastisk\\_proces/Markov-proces](http://www.denstoredanske.dk/It_teknik_og_naturvidenskab/Matematik_og_statistik/Sandsynlighedsregning_og_stokastisk_proces/Markov-proces) - Besøgt d. 17.12.2011
13. Betinget Middelværdi. Udgivet af Gyldental. Internetadresse: [http://www.denstoredanske.dk/It\\_teknik\\_og\\_naturvidenskab/Matematik\\_og\\_statistik/Sandsynlighedsregning\\_og\\_stokastisk\\_proces/betinget\\_middelv%C3%A6rdi](http://www.denstoredanske.dk/It_teknik_og_naturvidenskab/Matematik_og_statistik/Sandsynlighedsregning_og_stokastisk_proces/betinget_middelv%C3%A6rdi) - Besøgt d. 17.12.2011
16. Betinget fordelinger. Udgivet af Københavns Universitet. Internetadresse: <http://www.math.ku.dk/~erhansen/stat2b/notater/notat4-3.pdf> - Besøgt d. 17.12.2011
18. Støj. Udgivet af Gyldental. Internetadresse: [http://www.denstoredanske.dk/It%2c\\_teknik\\_og\\_naturvidenskab/Elektronik%2c\\_teletrafik\\_og\\_kommunikation/Elektronik%2c\\_radio\\_og\\_tv/st%C3%B8j](http://www.denstoredanske.dk/It%2c_teknik_og_naturvidenskab/Elektronik%2c_teletrafik_og_kommunikation/Elektronik%2c_radio_og_tv/st%C3%B8j) - Besøgt d. 17.12.2011

20. Hvid Støj. Udgivet af Gyldendal. Internetadresse:  
[http://www.denstoredanske.dk/I%e5\\_teknik\\_og\\_naturvidenskab/Elektronik,\\_teletrafik\\_og\\_kommunikation/Kommunikationsteori\\_og\\_teletrafikteori/hvid\\_st%C3%B8j](http://www.denstoredanske.dk/I%e5_teknik_og_naturvidenskab/Elektronik,_teletrafik_og_kommunikation/Kommunikationsteori_og_teletrafikteori/hvid_st%C3%B8j) - Besøgt d. 17.12.2011

23. Arduino Mega. Udgivet af Arduino. Internetadresse: <http://arduino.cc/en/Main/ArduinoBoardMega> - Besøgt d. 17.12.2011

24. Calculus Applied to Probability and Statistics. Udgivet af Zweig Media. Internetadresse:  
<http://www.zweigmedia.com/RealWorld/cprob/cprob3.html> - Besøgt d. 18.12.2011

## Programmer

21. George W. Taylor: Tone generator. <http://www.tropicalcoder.com/AudioTestFileGen.htm>, 16.12.2011

## Figurer

Figur 1: Se kilde nr. 5, Side 23

Figur 2: Se kilde nr. 6, Side 278

Figur 3: Se kilde nr. 6, Side 278

Figur 4: Standard Deviation Diagram. Udgivet af Wikipedia. Internetadresse:  
[http://en.wikipedia.org/wiki/File:Standard\\_deviation\\_diagram.svg](http://en.wikipedia.org/wiki/File:Standard_deviation_diagram.svg) - Besøgt d. 15.12.2011

Figur 5: Kalman filter. Udgivet af Wikipedia. Internetadresse:  
[http://en.wikipedia.org/wiki/File:HMM\\_Kalman\\_Filter\\_Derivation.svg](http://en.wikipedia.org/wiki/File:HMM_Kalman_Filter_Derivation.svg) - Besøgt d. 17.12.2011

Figur 6: Se kilde nr. 3, Side 6

Figur 7: Se kilde nr. 19, Side 194

Figur 8: Se kilde nr. 19, Side 195

Figur 9: Se kilde nr. 19, Side 197

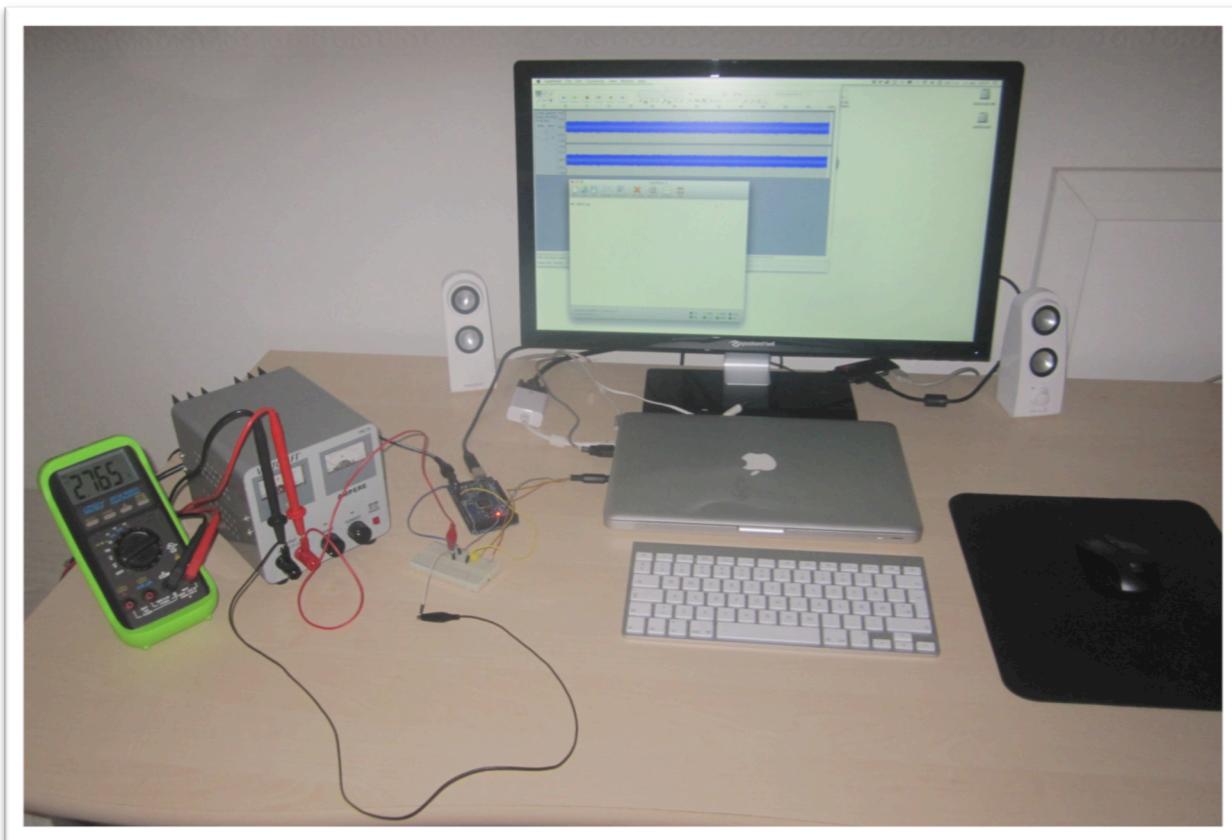
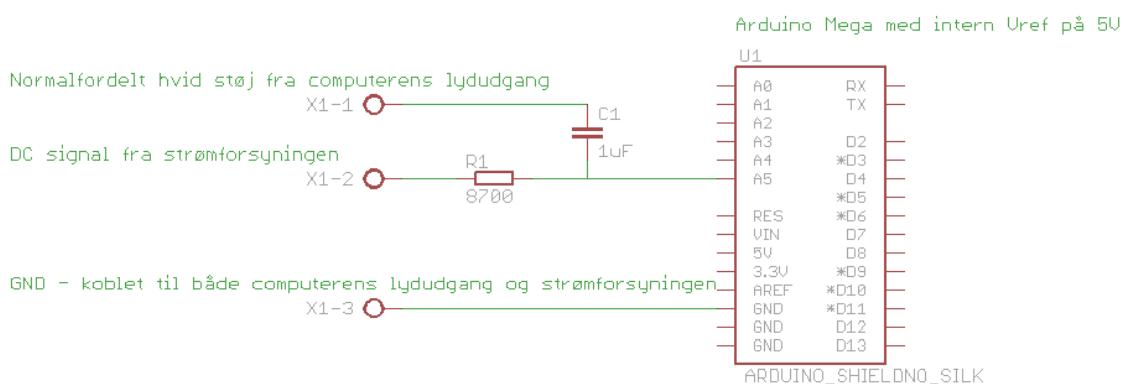
Figur 10: Se kilde nr. 19, Side 200

Figur 11: Se filen "Analyse af Målestøj.xlsx" i "Bilag 7 – CD-ROM"

Figur 12: Se filen "AD med Kalman filter.xlsx" i "Bilag 7 – CD-ROM"

## Bilag 1 – Skematiske tegning og billede af opstilling

Nedenfor ses et billede af henholdsvis den skematiske tegning og et billede af opstillingen. Til venstre i opstillingen ses et multimeter koblet direkte til en strømforsyning. Derefter ses et fumlebræt med højpasfilteret og efter dette Arduino Mega'en. Til højre ses computeren der afspiller en lydfil, som er forbundet til højpasfilteret via computerens lydudgang. Arduino Mega'en kommunikerer med computeren vha. seriel kommunikation, så målinger efterfølgende kan gemmes og behandles i Excel.

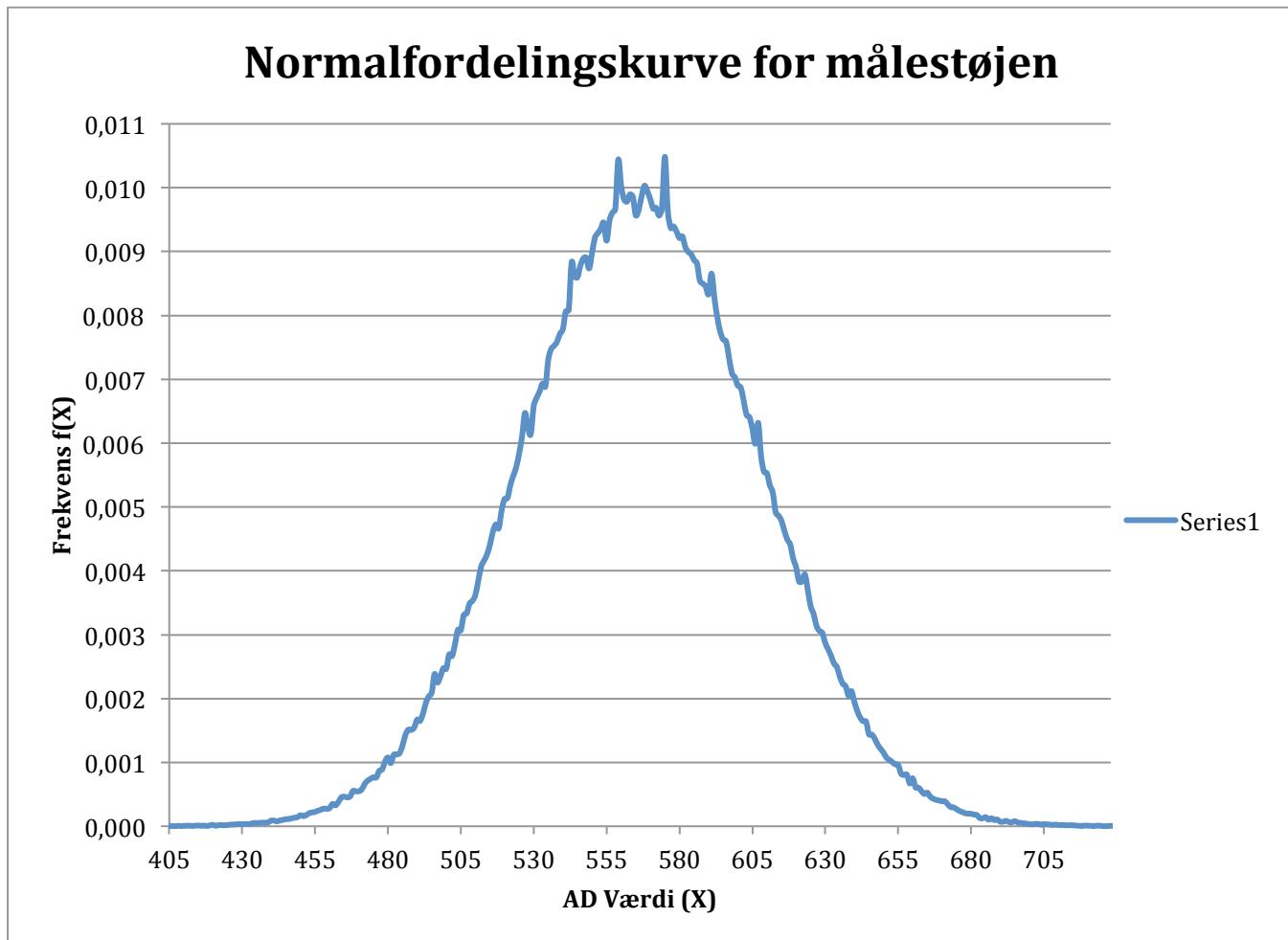


## Bilag 2 – Arduino kode: Analyse af målestøj

```
1. long timer = 0;//Timer til at se hvor lang tid målingen tog
2. const unsigned long READINGS = 1000000;//Ca. 1 minut og 40 sekunder, da den mäter med en frekvens
   på ca. 10000Hz
3. const unsigned int LENGTH = 1024;//Længde på data array'et - den er 1024, da AD-
   værdierne kan være 0-1023
4. unsigned int input[LENGTH];
5.
6. void setup()
7. {
8.   Serial.begin(115200);//Setup seriell kommunikation inklusiv set seriell baud rate
9.   Serial.println("\nADC Maaling");//Indiker at den er startet
10. }
11.
12. void loop()
13. {
14.   timer = millis();//Nulstil timeren
15.
16.   for(long i=0;i<READINGS;i++)//Læs analog indgang 0, 100000 gange
17.     input[analogRead(0)]++;
18.
19.   timer = millis() - timer;//Indstil timeren
20.
21.   for(int i = 0; i<LENGTH;i++)//Skriv alle dataene til computeren vha. serielparten
22.     Serial.println(input[i]);
23.
24.   //Skriv hvor lang tid målingen tog - ',' går at Excel automatisk kan opdele værdierne
25.   Serial.print("\ndtimer:,");
26.   Serial.println(timer);
27.   Serial.print("I sekunder:,");
28.   Serial.println(timer/1000);
29.
30.   Serial.print("Antal maalinger (n):");//Skriv antallet af målinger
31.   Serial.println(READINGS);
32.
33.   while(true);//Sæt den i et uendeligt while loop, så det kun foretages en gang
34. }
```

### Bilag 3 – Normalfordelingskurve for målestøjen

Normalfordelingskurven eller Gauss-kurven for målestøjen kan ses nedenfor.



## Bilag 4 – Arduino kode: AD-konverterer med Kalman filter

```
1. long timer = 0;//Timer til at se hvor lang tid målingen tog
2. const unsigned int LENGTH = 1000;//Angiver antal målinger og dermed også længden af alle arrays
3.
4. //Input variabler
5. unsigned int input;//AD Input
6. unsigned int rawInput[LENGTH];//Array til at gemme AD input
7. float kalmanInput[LENGTH];//Array til at gemme den estimerede AD værdi
8.
9. //Kalman filter konstanten
10. const float Var_v = 1636.931;//Varians af målestøj
11. const float Var_w = 0;//Varians af processtøj
12.
13. //Kalman filter variabler
14. float K = 0;
15. float X = 512;//Første bud er at spændingen ligger på 2,5V -
   dette er smart da det er midt i intervallet for AD-konverteren
16. float P = 512;//Samtidig skal den ikke vægte den indkomne data særlig højt
17.
18. void setup()
19. {
20.   Serial.begin(115200);//Setup seriel kommunikation inklusiv set seriel baud rate
21.   Serial.println("\nADC Maaling med Kalman filter");
22.   Serial.print("AD-vaerdi");
23.   Serial.print(",");
24.   Serial.println("Estimeret AD-vaerdi");
25. }
26.
27. void loop()
28. {
29.   timer = millis();//Nulstil timeren
30.
31.   for(int i=0;i<LENGTH;i++)//Kør for loopen det det antal gange der står i LENGTH
32.   {
33.     input = analogRead(0);//Læs analog port 0
34.     rawInput[i] = input;//Gem i arrayet
35.     kalmanInput[i] = kalmanCalculate((float)input);//Gem den estimerede AD-værdi i arrayet
36.   }
37.
38.   timer = millis() - timer;//Indstil timeren
39.
40.   for(int i = 0; i<LENGTH;i++)//Skriv alle dataene til computeren vha. serielparten
41.   {
42.     Serial.print(rawInput[i]);
43.     Serial.print(",");
44.     Serial.println(kalmanInput[i], DEC);
45.   }
46.
47. //Skriv hvor lang tid målingen tog - ',' gør at Excel automatisk kan opdele værdierne
48. Serial.print("\ndtime:");
49. Serial.println(timer);
50. Serial.print("In seconds:");
51. Serial.println(timer/1000);
52.
53. Serial.print("Antal maalinger (n):");//Skriv antallet af målinger
54. Serial.println(LENGTH);
55.
56. while(true);//Sæt den i et uendeligt while loop, så det kun foretages én gang
57. }
58.
59. float kalmanCalculate(float ADValue)
60. {
61.   //Xk += ADValue2;//Ekstra kontrol input - denne bruges ikke
```

```
62. P += Var_w;//Estimerede fejl-varians á priori, dvs. før målingen
63. K = P/(P+Var_v);//Kalman gain
64. X += K*(ADValue - X);//Estimerede AD-værdi
65. P = (1-K)*P;//Estimerede fejl-varians á posteriori, dvs. efter målingen
66. return X;
67. }
```

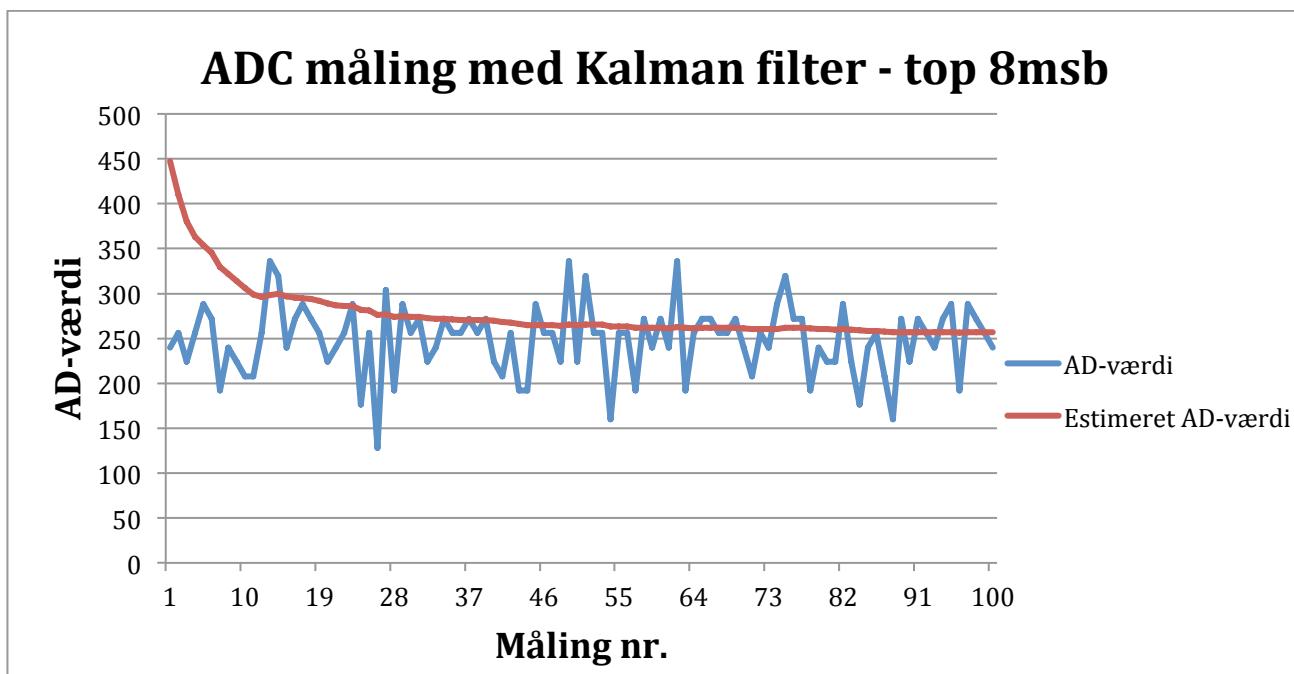
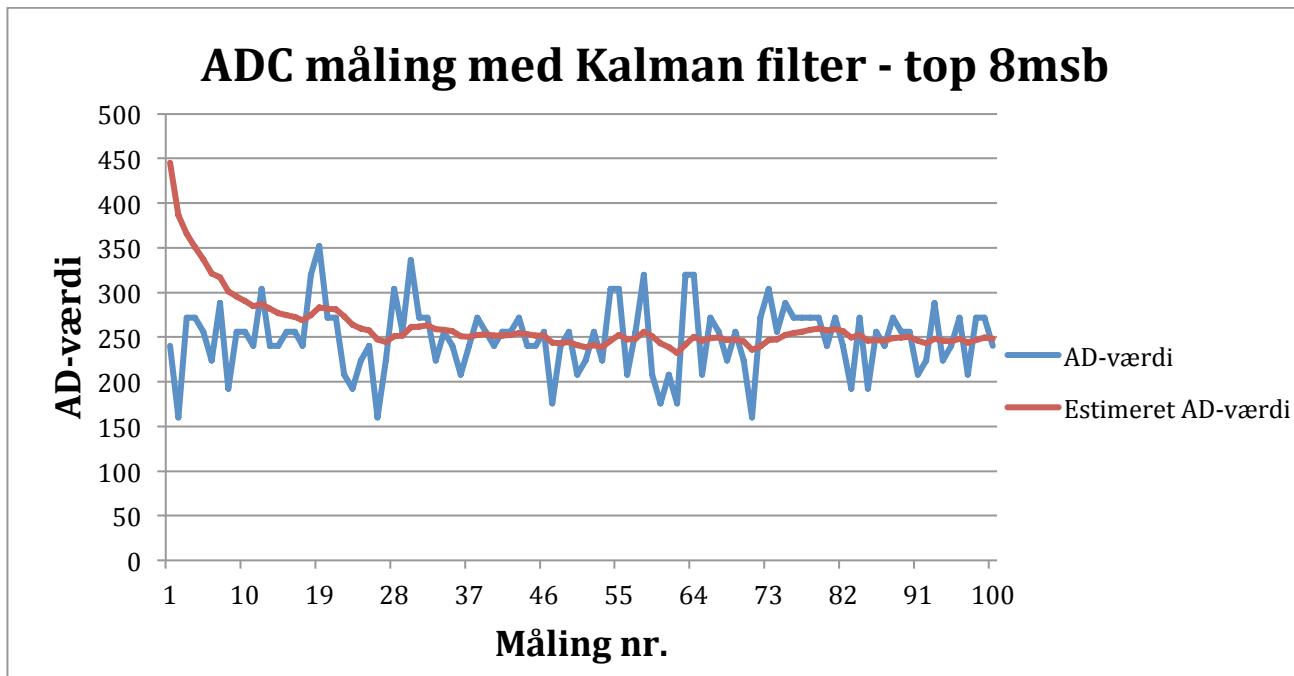
## Bilag 5 – Arduino kode: AD med Kalman filter top 8msb

```
1. long timer = 0;//Timer til at se hvor lang tid målingen tog
2. const unsigned int LENGTH = 1000;//Angiver antal målinger og dermed også længden af alle arrays
3.
4. //Input variabler
5. unsigned int input;//AD Input
6. unsigned int rawInput[LENGTH];//Array til at gemme AD input
7. float kalmanInput[LENGTH];//Array til at gemme den estimerede AD værdi
8.
9. //Kalman filter konstanten
10. const float Var_v = 1636.931;//Varians af målestøj
11. const float Var_w = 21.333;//Varians af processtøj - i andet forsøg var denne sat til 0
12.
13. //Kalman filter variabler
14. float K = 0;
15. float X = 512;//Første bud er at spændingen ligger på 2,5V - dette er smart da det er midt i intervallet for AD-konverteren
16. float P = 512;
17.
18. void setup()
19. {
20.     Serial.begin(115200);//Setup seriel kommunikation inklusiv set seriel baud rate
21.     Serial.println("\nADC Maaling med Kalman filter");
22.     Serial.print("AD-værde");
23.     Serial.print(",");
24.     Serial.println("Estimeret AD-værde (Kalman filter)");
25. }
26.
27. void loop()
28. {
29.     timer = millis();//Nulstil timeren
30.
31.     for(int i=0;i<LENGTH;i++)//Kør for loopen det det antal gange der står i LENGTH
32.     {
33.         input = analogRead(0) & 0x3F0;//Læs analog port 0
34.         rawInput[i] = input;//Gem i arrayet
35.         kalmanInput[i] = kalmanCalculate((float)input);//Gem den estimerede AD-værde i arrayet
36.     }
37.
38.     timer = millis() - timer;//Indstil timeren
39.
40.     for(int i = 0; i<LENGTH;i++)//Skriv alle dataene til computeren vha. serielporten
41.     {
42.         Serial.print(rawInput[i]);
43.         Serial.print(",");
44.         Serial.println(kalmanInput[i], DEC);
45.     }
46.
47. //Skriv hvor lang tid målingen tog - ',' gør at Excel automatisk kan opdele værdierne
48. Serial.print("\ndtime:,");
49. Serial.println(timer);
50. Serial.print("In seconds:,");
51. Serial.println(timer/1000);
52.
53. Serial.print("Antal maalinger (n):,");
54. Serial.println(LENGTH);
55.
56. while(true);
57. }
58.
59. float kalmanCalculate(float ADValue)
```

```
60. {  
61. //Xk += ADValue2;//Ekstra kontrol input - denne bruges ikke  
62. P += Var_w;//Estimerede fejl-varians a priori, dvs. før målingen  
63. K = P/(P+Var_v);//Kalman gain  
64. X += K*(ADValue - X);//Estimerede AD-værdi  
65. P = (1-K)*P;//Estimerede fejl-varians a posteriori, dvs. efter målingen  
66. return X;  
67. }
```

## Bilag 6 – Graf for Kalman filter top 8msb

Nedenfor ses de to grafer for målingerne, hvor de fire nederste bit er sat til 0. Den øverste er en måling, hvor processtøjen er sat til 21,333, mens den nedenunder er sat til 0. Som man kan se, er den øverste således ikke i stand til at filtrere støjen ordentlig fra. Det er den nederste derimod. Den estimerede AD-værdi er imidlertid otte mindre end den faktiske AD-værdi på ca. 260. For at se alle målingerne henvises til henholdsvis ”AD med Kalman filter top 8msb.xlsx” og ”AD med Kalman filter top 8msb\_Var0.xlsx” i ”Bilag 7 – CD-ROM”.



## Bilag 7 – CD-ROM

Denne CD-ROM indeholder følgende filer:

- ”60s - Gaussian White Noise.wav” – dette er lydfilen bestående af 60 sekunders normalfordelt hvid støj.
- ”Analyse af Målestøj.xlsx” – ADC målingerne brugt til analyse af målestøjen
- ”AD med Kalman filter.xlsx” – ADC målingerne med implementeret Kalman filter
- ”AD med Kalman filter top 8msb.xlsx” – ADC målingerne med implementeret Kalman filter og hvor sidste fire bits er sat til 0 – processtøjen var sat til 21,333.
- ”AD med Kalman filter top 8msb\_Var0.xlsx” – Samme som overstående – processtøjen var blot sat til 0.

## Opgaveformulering

Vedlæg opgave formuleringen