Maskinteknikk 2 - D&V delen - RUL

Jørn Vatn
eMail: jorn.vatn@ntnu.no

Updated 2024-01-09

RUL og betydning for vedlikehold

Vedlikehold er definert som “En kombinasjon av alle tekniske og administrative aktiviteter, inkludert ledelsesaktiviteter, som har til hensikt å opprettholde eller gjenvinne en tilstand som gjør en enhet i stand til å utføre en krevd funksjon”. Videre defineres *forebyggende* vedlikehold som “Vedlikehold som utføres etter forutbestemte intervaller eller ifølge forutbestemte kriterier, og som har til hensikt å forlenge levetider, redusere sannsynligheten for svikt eller funksjonsnedsetting (degradering)”. Vi skal heller ikke glemme *korrigerende* vedlikehold som er definert ved “Vedlikehold som utføres etter at en feil (tilstand) er oppdaget, og som har til hensikt å bringe en enhet tilbake i en tilstand som gjør det mulig å utføre en krevd funksjon”.

I grove trekk ønsker vi å minimere korrigerende vedlikehold. For å redusere mengden korrigerende vedlikehold utføres ulike typer forebyggende vedlikehold. Vedlikeholdsoptimalisering dreier seg om å balansere fordeler og ulemper ved å utføre vedlikehold. For å gjøre dette trenger vi informasjon som kan si oss “hvor kritisk” det er å utføre forebyggende vedlikehold “nå”. Dersom du velger retningen “Drift og vedlikehold” vil du lære mer om dette. I denne presentasjonen vil vi kun gi en liten smakebit.

Forkortelsen RUL står for Remaining Usefull Lifetime og kan på norsk oversettes til gjenværende nyttig levetid. Her er det tre viktige forhold:

1. Remaining/Gjenværende: Dette henspeiler på at vi på nåtidspunktet, t_0, ønsker å benytte kunnskap og informasjon vi har til å se “framover”. Informasjon kan være alder til en enhet og/eller tilstand til enheten.

2. Usefull/nyttig: Dette henspiller på at enheten vi betrakter degraderer over tid, og vil etter en stund ikke være nyttig lenger i den tilstand den vil befinne seg i. Hva som er nyttig må defineres. Noen ganger er dette enkelt, den er ikke nyttig dersom den har sviktet. I andre situasjoner kan den fortsatt fungere, men kostnaden ved å “kjøre videre” er så stor at vi ikke betrakter enheten som nyttig lenger.

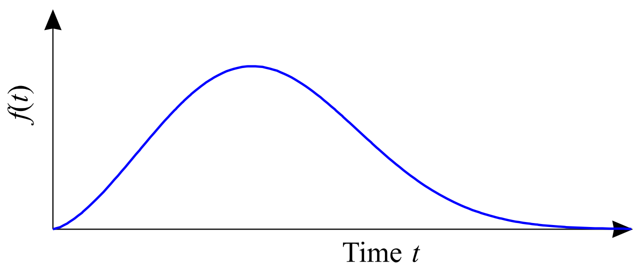
3. Lifetime/levetid: Dette henspiller på tidspunktet fram i tid, T, når enheten ikke lenger er nyttig.

Kunnskapen vi har om RUL vil være viktig for å sette inn pro-aktive tiltak. Dette kan være å bytte enheten, oppgradere enheten, bytte slitedeler osv. Selv om vi kan identifisere fornuftige tiltak, er det ikke rett frem å bestemme når vi skal utføre tiltak. En viktig årsak til dette er at RUL ikke er kjent, vi har kun en formening om RUL. Formelt benyttes stor bokstav T i det følgende om RUL, og T er en stokastisk variabel. Ett annet begrep som ofte benyttes er å si at T er en tilfeldig størrelse. Med tilfeldig menes at vi ikke vet verdien T vil anta, dvs når enheten ikke lenger er nyttig. Dersom vi ikke får gjort noe før enheten ikke er nyttig, må vi typisk utføre en korrigerende vedlikeholdsaksjon, som ofte er mye dyrere enn å gjøre et forebyggende tiltak.

For å finne tidspunktet for når vi skal "sette inn søtet", trenger vi å uttrykke usikkerheten i T. Budskapet er derfor at RUL ikke er et tall, men det kan være en prediksjon i form av et tall og med tilhørende *usikkerhet*.

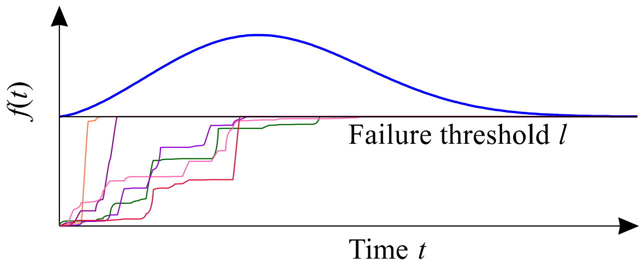
Sannsynlighetsfordeling og tilfeldige størrelser

For å beskrive tilfeldige størrelser benyttes ofte en sannsynlighetsfordeling. Den sier noe om hvor sannsynlig hver enkelt verdi av den tilfeldige størrelsen T er. Figur 1 viser en slik sannsynlighetsfordeling.



Figur 1: Sannsynlighetsfordeling, $f(t)$

Figur 1 viser ingenting som indikerer at enheten degraderer over tid, den viser kun sannsynligheter for ulike verdier. Ofte tenker vi at det er en underliggende degraderingsprosess som leder til at enheten ikke lenger er nyttig. Slike degraderingsprosesser er også "tilfeldige", og denne tilfeldigheten bunner ofte i at belastninger, bruk osv varierer over tid. Figur 2 viser ulike løp som leder til at enheten ikke lenger er nyttig. Ofte tenker vi at enheten ikke er nyttig lenger dersom degraderingen overskrider en grense, "failure threshold".



Figur 2: Ulike løp for degradering som leder til en tilfeldig tid for "ubrukbar enhet"

1 Helseindikator og RUL

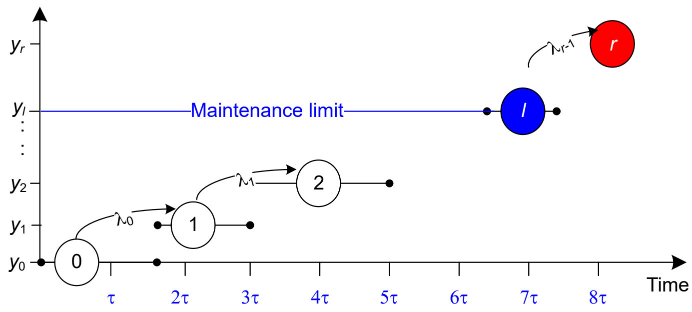
Prediktivt vedlikehold er en vedlikeholdsstrategi som baserer seg på at vi benytter kunnskap om tilstand til en enhet, samt kunnskap om tilstandsutvikling for å foreta prediksjoner mht utvikling over tid for enheten som vurderes.

Den såkalte helseindikatoren er da informasjonsbæreren om tilstanden. Mange teknikker finnes for å finne en slik helseindikator, det betyr både teknikker for å "måle", f eks ultralyd eller vibrasjonsmålinger, men også teknikker for å "analysere" måleverdiene, for eksempel "Fast Fourier Transform, FFT". I denne presentasjonen vil vi ikke utdype slike teknikker, men anta at vi er i stand til å fremskaffe en nyttig helseindikator som kan benyttes for å lage prediksjoner.

Figur 2 viser ulike løp for utviklingen av en kontinuerlig helseindikator. Ofte er det naturlig å tenke på helseindikatoren som en kontinuerlig størrelse. I andre situasjoner kan det være mer hensiktsmessig å diskretisere helseindikatoren. Mange selskaper benytter kun et fåtall tilstander for å beskrive sin helseindikator. For eksempel Equinor benytter verdiene "Frisk", "Uvell", "Syk" og "Død" som tilstander i sitt vedlikeholdsstyringssystem.

2 Markovmodell

I dette notatet presenterer vi en modell hvor helseindikatoren er diskret. Vi lar $Y(t)$ være verdien til helseindikatoren ved tid t. Figur 3 viser utvikling av $Y(t)$ som funksjon av tiden. Tilstandene gis nummer $0,1,2,\ldots,r$ hvor en perfekt enhet har tilstand 0 svarende til en "fysisk tilstand", dvs y_0, osv opp til tilstand r som er tilstanden hvor enheten ikke lenger er nyttig, f eks at den har sviktet. Vi har også indikert en såkalt "vedlikeholdsgrense" (Maintenance limit). Dersom enheten når tilstand ℓ, antar vi at når dette oppdages utføres et tiltak som setter tilstanden tilbake til så god som ny, dvs til til-



Figur 3: Markov modell

stand 0. Enheten inspiseres ved jevne mellomrom, dvs med intervaller av lengde τ. Tilstanden til enheten er kun kjent like etter en inspeksjon. I og med at tilstanden er diskretisert, så vil enheten oppholde seg i de ulike tilstandene en viss tid. Denne tiden er tilfeldig. Det er ikke illustrert i Figur 3, men vi har indikert at det er en viss "overgangssannsynlighet" mellom tilstandene. Konkret defineres overgangsrater λ_i hvor $\lambda_i \Delta t$ er tilnærmet lik sannsynligheten for at en enhet som er i tilstand i ved tidspunkt t vil være i tilstand $i+1$ ved tidspunkt $t+\Delta t$.

For å forstå overgangene her, og "tilfeldighetenes spill", kan vi tenke en veldig enkel bruddmekanisk modell. $Y(t)$ er lengden av en sprekk ved tidspunkt t. Vi starter med $Y(0)=0$, og så tenker vi oss at for hvert lite tidsintervall av lengde Δt er det en viss sannsynlighet, $\lambda_i \Delta t$ for at vi får en "stor belastning" som driver sprekken til neste nivå, hvor vi da har diskretisert sprekklengden i r nivåer før enheten går til brudd.

Matematisk er det en del antagelser vi må gjøre knyttet til en slik tilfeldig prosess. Og i noen tilfeller kan vi finne at antagelsene leder til en såkalt Markovprosess. Vi vil derfor forfølge situasjonen hvor Markovantagelsene er oppfylt uten at vi dykker ned i argumenter for dette.

Markovmodellen er en av de enkleste modellene vi kan benytte for å finne RUL-fordelingen. Vi trenger ingen avansert sannsynlighetsregning, men for å kunne gjøre noe fornuftig trenger vi å kunne programmere. Til slutt vil vi vise Python-kode som kan benyttes til å finne RUL-fordelingen.

Dersom vi nå lar $P_i(t)$ betegne sannsynligheten for at enheten er i tilstand i ved tidspunkt t, kan vi enkelt argumentere for følgende tilnærming:

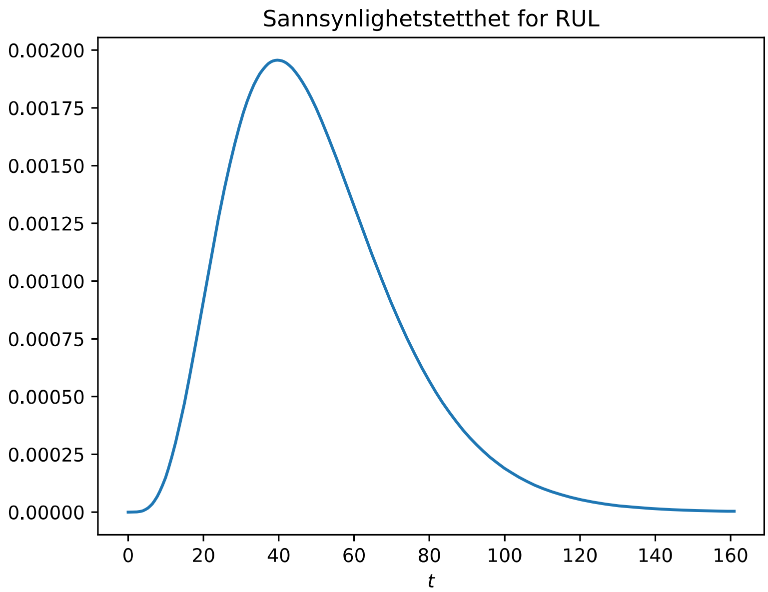
$$P_i(t+\Delta t) \approx P_i(t)(1-\lambda_i \Delta t) + P_{i-1}(t)\lambda_{i-1}\Delta t \qquad (1)$$

Her er argumentet at for å være i tilstand i ved tidspunkt $t+\Delta t$ så må vi enten ha vært i tilstand i ved tidspunkt t, og ikke "hoppet" til tilstand $i+1$ i løpet av tiden Δt, eller så var vi i tilstand $i-1$ ved tidspunkt t og "hoppet" til tilstand i i løpet av tiden Δt. Sannsynligheten for å "stå i ro" er $1-\lambda_i \Delta t$, mens sannsynligheten for å hoppe er $\lambda_{i-1}\Delta t$.

For en ny enhet kan vi nå benytte ligning (1) til å beregne $P_i(t)$ for ulike t verdier. Initielle betingelser er:

$$\begin{aligned} P_0(0) &= 1 \\ P_i(0) &= 0 \text{ for } i > 0 \end{aligned} \tag{2}$$

hvor vi spesielt er interessert i å finne $P_r(t)$ som er sannsynlighetsfordelingen til T, dvs til RUL. Figur 4 viser PFD for situasjonen med $r = 5$, og $[\lambda_0, \lambda_1, \ldots \lambda_{r-1}] = [1/12, 1/11, 1/10, 1/9, 1/8]$. Merk at vi her har latt λ-verdiene bli større og større. Dette er ofte situasjonen fordi degraderingen øker ofte som funksjon av degraderingsnivået. Dvs en degradert enhet slites raskere enn en ny enhet.



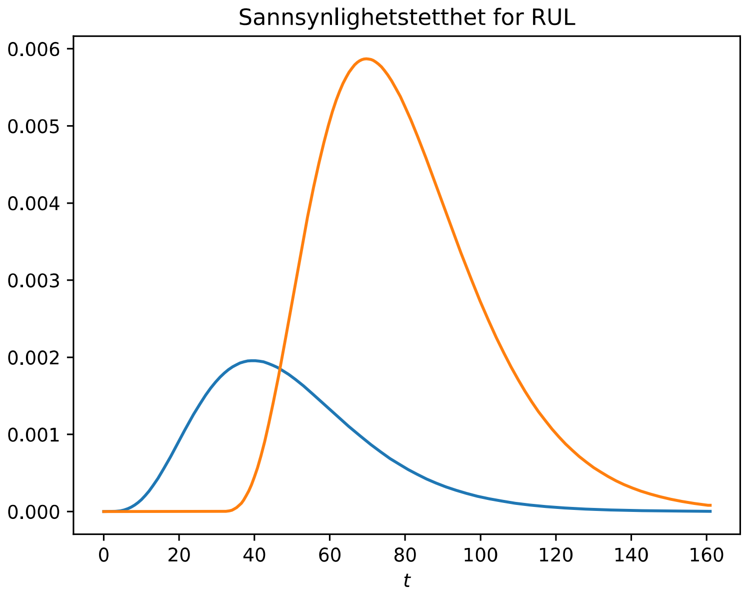
Figur 4: Sannsynlighetsfordeling, $f(t)$, gitt at vi starter i tilstand 0 ved tid $t = 0$

Dersom vi har observert at tilstanden er f eks l ved tidspunkt t_0, dvs ved en potensiell vedlikeholdsgrense, kan vi også finne RUL-fordelingen, men da med initiale betingelser:

$$\begin{aligned} P_\ell(t_0) &= 1 \\ P_i(t_0) &= 0 \text{ for } i \neq \ell \end{aligned} \tag{3}$$

Vi kan nå gjenta beregningene, og vi har antatt at $t_0 = 30$, og $\ell = 3$, og Figur 5 viser nå den betingede RUL-fordelingen gitt denne informasjonen i den orange kurven, mens den blå kurven er den samme som i Figur 4.

Dersom du vil se Python-koden for å lage plottene, så finner du den her: `https://folk.ntnu.no/jvatn/eLearning/MAST1102/markovsimple.zip`.



Figur 5: Sannsynlighetsfordeling, $f(t)$, gitt at vi starter i tilstand 3ved tid $t_0 = 30$

Midlere tid til svikt

Et sentralt begrep innenfor fagområdet sikkerhet, pålitelighet og vedlikehold er MTTF. MTTF er forkortelsen for Mean Time To Failure, og på norsk blir dette midlere tid til svikt. MTTF er da tiden det i gjennomsnitt tar fra en ny enhet settes i drift til den svikter. Det kan vises at MTTF kan beregnes ved:

$$\text{MTTF} = \int_0^\infty t f(t) dt = \int_0^\infty R(t) dt \tag{4}$$

hvor $R(t)$ er overlevelsessannsynligheten og er gitt ved $R(t) = \int_t^\infty f(u) du$. I Python koden er det vist hvordan vi også kan beregne MTTF'en numerisk i Markovmodellen. Dette gjøres ved å benytte at $R(t) = 1 - P_r(t)$, slik at vi har:

$$\text{MTTF} = \int_0^\infty [1 - P_r(t)] dt \tag{5}$$

I Python-koden er det vist hvordan vi kan beregne MTTF i denne situasjonen. Merk at for oss så er da “useful lifetime” det samme som tid til svikt fra nå av.

Vedlikeholdsmodeller

I forelesningen vil vi ikke ha tid til å lage noen modeller for optimalt vedlikehold. I eksemplet som er illustrert i Figur 3 er det to størrelser vi ønsker å finne optimale verdier for, den ene størrelsen er τ som angir hvor ofte enheten inspiseres, mens den andre størrelsen er ℓ, dvs vedlikeholdsgrensen. Siden optimalisering er å finne balansen mellom fordeler og ulemper ønsker vi å uttrykke dette i form av en matematisk funksjon som skal minimeres, dvs vi ønsker å minimere de forventede kostnadene. Funksjonen vi her benytter er:

$$C(\tau, \ell) = c_{\mathrm{I}}/\tau + (c_{\mathrm{F}} + c_{\mathrm{CM}})\lambda_{\mathrm{E}}(\tau, \ell) + c_{\mathrm{RC}}\rho_{\mathrm{E}}(\tau, \ell) \tag{6}$$

hvor

- ℓ = vedlikeholdsgrensen
- c_{I} = kostnad for å utføre en inspeksjon
- c_{F} = totale kostnader ved en svikt, dvs nedetidskostnader, sikkerhetskostnader osv
- c_{CM} = kostnader for å reparere enehten (CM = Corrective Maintenance = korrigerende vedlikehold)
- c_{RC} = kostnadene for fornyelse av en degradert enhet, dvs den har ikke sviktet, men tilstanden har nådd vedlikeholdsgrensen
- $\lambda_{\mathrm{E}}(\tau, \ell)$ = den effektive feilraten, dvs forventede antall svikt per tidsenhet for en enhet som inspiseres ved faste intervaller av lengde τ og fornyes når vedlikeholdsgrensen er nådd, og vi har oppdaget dette ved en inspeksjon
- $\rho_{\mathrm{E}}(\tau, \ell)$ = forventet antall fornyelse per tidsenhet for en enhet som inspiseres ved faste intervaller av lengde τ og fornyes når vedlikeholdsgrensen er nådd, og vi har oppdaget dette ved en inspeksjon

Det er forholdsvis enkelt å modifisere Python-koden for å beregne både effektiv feilrate og fornyelsesraten. Når vi integrerer Markov-ligningene må vi "flytte" sannsynlighetene over vedlikeholdsgrensen tilbake til 0 hver gang tiden passerer $\tau, 2\tau, \ldots$ svarende til at vi fornyer. Videre må vi telle opp hvor ofte dette skjer for å finne fornyelsesraten. Vi må også telle opp hvor ofte vi går til tilstand r. Dersom du velger retningen "Drift og vedlikehold" og går videre på et 2-årig masterprogram i sikkerhet, pålitelighet og vedlikehold, vil du lære hvordan du kan integrere disse modellene, og mye annet som både er spennende og nyttig.

Kontinuerlig helseindikator

Selv om Markovmodellen kan virke litt komplisert å arbeide med, så er det generelt mer utfordrende å jobbe med modeller hvor helseindikatoren er kontinuerlig. I noen situasjoner kan vi finne analytiske modeller for RUL-fordelingen basert på de underliggende degraderingsmekanismene. Men generelt må vi utføre "integrasjon" på samme måte som for Markov-modellen, men nå er y-aksen kontinuerlig, som gjør integrasjonen betydelig mer utfordrende. Spesielt vanskelig blir dette dersom degraderingen øker med degraderingsnivå så vel som alder. Dette er ofte tilfellet fordi kreftene som driver degraderingen ofte øker både med alder og degradert tilstand. Fra brudmekanikken vet vi for eksempel at en sprekk vokser hurtigere jo lenger sprekken er.