

Python Drift Scan script.

Indledning.

Formål med scriptet er at logge en måling af Solstøj i en datafile mens Solen passerer gennem fokus på en parabol. Forår og efterår vil der være en dato, hvor Solen vil være i centrum på en fast monteret QO100 parabol, og når man logger støjen fra Solen kan man efterfølgende få et indtryk af parabolens effektivitet.

Formålet med målingen er, at få et antal grader for parabolens 3 dB åbningsvinkel (HPBW), som kan være forstyret af faktorer som dårlig indstilling, LNB's belysning af parabolen, TX-helix-spiral påvirkning af LNB osv.

Datoer for Solen's passage af fokuspunktet kan findes med værktøjer som <https://www.suncalc.org/> . Desuden findes der nogle Java-script på Internettet af Joachim Koppen, DF3GJ, som bruges til at lave driftscan kurver oprindelig af Måne-støj på meget høje frekvenser fra 10 GHz og opad (<https://portia.astrophysik.uni-kiel.de/~koeppen/JS/LunarDriftScans.html>). Scriptet kan også bruges til at finde parabolens åbningsvinkel HPBW med Solstøj på 10 GHz, hvor QO100 satelliten har sit download. Desuden har Joachim, DF3GJ skrevet en artikel i Dubus 3/2023, "How to Measure the Width of a Narrow Antenna Beam" *1

Metoden med optagelse af en driftscan kurve kan komme i betragtning for en parabol med en åbningsvinkel, som er mindst 3 gangen Solens vinkel på himlen på ca. 0.5 grad. For disse paraboler vil Solen kunne betragtes som en punktformet støjkilde. I praksis vil dette nok gælde for QO100-parabol mindre end ca. 1 m.

Forberedelse.

Hvis man skal være klar til fredag den 28. februar 2025, skal Python-scriptet installeres og afprøves på en PC med en RTL-dongle tilsluttet en USB-port. Man skal nok være klar mindst dagen før og lave en afprøvning, hvor man sikrer logfilen. På dagen har man kun en chance og skal vente ½ år igen.

Min opsætning er på en Ubuntu-clon PC med Python version 3. Hvordan scriptet kan afvikles på en Windows PC kan jeg desværre ikke hjælpe med, men det burde kunne gøres. Det første man kan sikre sig er, at man med RTL-dongle og et SDR-program som SDR# eller SDR++, kan høre QO100's narrowband beacon på 10.489,500 MHz.

QO100 installation.

Min QO100 opsætning består i korthed af:

- En ombygget MAXIMUM ST-11 LNB (Harald Nyborg LNB efter OZ2OE's konstruktion) med en Helix TX-spiral.
- En 90 cm Nokia parabol
- En RTL-dongle ver. 3 (den i metalhus)

Gennemgang af Python-script.

Meget af koden i Python-scriptet er hentet fra <https://pysdr.org/>, som har en sektion med RTL-donglen. Koden bærer også flittig brug af ChatGPT. Jeg ved ikke om koden kræver

version 3 af Python, fordi det ikke er afprøvet på tidligere versioner. Men man skal installere pyrtlsdr-pakken. På Linux:

```
sudo pip install pyrtlsdr
```

I scriptet køres målingen med RTL-donglen i en selvstændig tråd, og donglen startes med parametre, som skal sættes direkte i koden. Som udgangspunkt er der testet med:

600 MHz, 1 msp/s og gain=20

Og scriptet måler signalstyrke hvert 2 sek. ved 2000 samples som smides væk og efterfølgende 2000 samples. Ud fra I- og Q-samples findes signalstyrken af støjen med dette:

```
signal_strength = np.sqrt(np.mean(np.abs(samples)**2))
```

Signalstyrken gemmes i en Python-liste sammen med antal sekunder siden start af målingen.

Den typiske måleværdi af signalstyrken er et decimaltal mellem 0.013 og 0.006 for henholdsvis den kraftigste og svageste støj. Første linie i logfilen kunne se sådan ud:

Time, 0.00, Strength, 0.012387

Styrken af signalet skulle gerne repræsentere power af sol- og egenstøj af et 1 MHz bredt frekvensbånd ved 600 MHz, som er mellemfrekvensen fra LNB'ens LO= 9.750 MHz. Gain er forsigtigt sat til 20 dB, og det formodes ikke at AGC-funktion er slået. Der findes en parameter til RTL-donglen, som hedder AGC=AUTO, som ikke skulle være aktiv. Man kunne gå højere op i frekvens til 700 MHz for at opnå bedre forstærkning i LNB'en. Men QO100 transponderen er en del af EsHail2 satellitten, som indeholder mange videokanaler. Omkring 700 MHz ses ikke noget af støjgulvet fra narrowband transponter, som slutter på mellemfrekvensen på 739,5 MHz.

Python script startes fra kommandolinien med: python3 monitor_rtl.py. Logningen af en halv driftscan kurve tager på en 125 cm parabol højst 10 min. Tid og signalstyrken udskrives undervejs på skærmen, og efter nogle af parabolens sideløops kan man se, at signalstyrken ikke falder længere. Man skal herefter taste EX og Return på keyboardet, hvorved tråden til RTL-donglen stopper. Logfilen hedder sun_noise_log.txt, og det er MEGET vigtigt for at sikre logfilen ved at tage en kopi, som gives et andet navn og måske sikres i en anden mappe. Scriptet, som er meget primitivt lavet, SLETTER den gamle logfile, når det startes forfra igen. Men prøv derfor at starte / stoppe script'et nogle gange inden den endelige test, så man kender det. Man har kun ET FORSØG på dagen!

Python-script'et afslutter med en rapport, hvor det på baggrund af et relativt simpel gennemløb af logdata, udskriver parabolens 85%-15% og 50% kurvepunkter:

Tråden lukker ned

Der er 303 målinger i listen. Finder min/max værdi af Strength

Min Strength: 0.006932 på index 277 i data

Max Strength: 0.013013 på index 8 i data

Value85: 0.012101, Value15: 0.007845
Stop_15_tid: 358.81 på Index_15%: 179 i data
Stop_85_tid: 88.20 på Index_85%: 44 i data
Start_max_tid: 0.00 på Index_100%: 0 i data
Stop_50_tid: 218.50 på Index_50%: 109 i data
Rim width for 15%/85%: 1.11 Find HPBW i Tabel 1
HPBW 3dB width for 50%: 1.80

Behandling af logfilen.

Data i logfilen udskrives, som beskrevet overfor, i en tekstfile af linier adskilt af Newline (\n). Hvis logfilen ikke kan læses på en Windows PC, kan det traditionelt være p.g.a. forskellen mellem Windows og Linux. Windows bruger vistnok CarriageReturn (CR eller CR/LF).

Den bedste behandling af logfilen fåes hvis den indlæses i et regneark, hvor felter i logfilen er adskilt af komma (","). Når det er lykkedes at indlæse logfilen i et regneark, vil sekunder stå i 2. kolonne og signalstyrke i 4. kolonne.

Find tiden for max. og min.

Hvis man i regnearket kan lave en kurve med tiden i sekunder på x-aksen og signalstyrken på y-aksen, fås et billede af forløbet af Solens støj i parabolen. Det er nu vigtigt at finde tiden for maksimum og minimum signal.

Når man skal omregne tiden for Solen til grader, gælder det for ved Ækvator:

Grader pr. sekund = 365 dage / (24timer * 60 min. * 60 sek.), giver ca. 0.00417 deg/s

Ifølge Jørgen, OZ7TA vil omregningen i DK incl. declinationen for fredag den 28. februar være 0,247 grader/minut, altså ca. 0.00412 grader/sekund.

HPBW ved 50%, 3dB.

Man skal nu finde tidspunktet, hvor signalstyrken er faldet med 50%. Dette fås ved i regnearket/kurven at finde den bedste signalværdi, som har en værdi af:

50% signal = $\frac{1}{2} * (\text{max. signal} - \text{min. signal}) + \text{min. signal}$;

Nu gåes fra 50% signalet på y-aksen ned til antal sekunder på x-aksen. Tiden for halveringen af signalstyrken udregnes med:

50% HPBW = Tid i sekunder ved 50% minus Tid for max. signal.

Den fulde HPBW i grader udregnes nu som 2 gange 50% tid omregnet til grader:

HPBW = 2 * 50% sekunder * 0.000412

Andre værdier.

Man kan også finde andre værdier, f.eks. 85% og 15% signalværdier i kurven, og så gå ind på Joachim, DF3GJ JavaScript eller en udvidet del af Tabel 1 i *1. Tabellen i artiklen er skrevet for større parabolen med HPBW mindre end 1 grad.

Her er en udvidelse af Tabel 1 for HPBW større en 1 grad:

HPBW gr.	0.15/0.85
1.1	0.66
1.2	0.72
1.3	0.778
1.4	0.835
1.5	0.892
1.6	0.95
1.7	1.01
1.8	1.07
1.9	1.12
2.0	1.18
2.1	1.24
2.2	1.3
2.4	1.41
2.5	1.47

Afslutning.

Det er vigtigt, at man afprøver scriptet nogle gange og lader det prøvekøre i lang tid inden man stoppe ved at taste EX og Return. Husk at starte i god tid inden Solen passerer, således man får et tydeligt maksimum med. Prøv også at køre med scriptet i dagene før og måske efter fredag den 28. februar. Solen stiger i de dage med omkring 0.3 grader pr. dag, så på gode paraboler vil det måske have indflydelse på resultatet hvor godt den rammer fokus den 28. februar.

Prøv også at overføre logfilen til et regneark. Hvis den ikke kan læses logfilen på en Windows PC, kan man måske rette slutningen af linie 89 til "\c" eller "\c\n":

```
out_text = "Time, " + "{:.2f}".format(tid_brugt) + ", Strength, " + "{:.6f}".format(strength) + "\n"
```

Noter:

*1 Joachim Koppen, DF3GJ, Dubus 3/2023, "How to Measure the Width of a Narrow Antenna Beam"