Herleitung von ɤ und Wellenwiderstand einer elektrischen Leitung mit Verlusten

$$\gamma=\sqrt{(R'+j\omega L')\cdot(G'+j\omega C')}$$

mit L'   Induktivitätsbelag in H/m
mit C'   Kapazitätsbelag in F/m
mit R'   Widerstandsbelag in Ohm/m
mit G'   dieelektrische Verluste in Siemens/m = 1/(Ohm *m)

aus z0² = L'/C' und ceff² = 1/(L' * C')

ergibt sich mit $ceff=\frac{c0}{\sqrt{\varepsilon r \mu r}}=ceff*vf$ und c0=2.997925e8 [m/s]

L' = z0/ceff und C' = 1/(ceff * z0)

eingesetzt:

$$\gamma=\sqrt{(R'+\frac{j\omega z0}{ceff})\cdot(G'+\frac{j\omega}{ceff\ z0})}$$

allgemein gilt:

ɤ * l = ɤ * ceff * delay = ɤd * delay,

daraus ergibt sich durch Multiplikation mit ceff:

$$\gamma d=\sqrt{(R'ceff+j\omega z0)\cdot(G'ceff+\frac{j\omega}{z0})}$$

wir substituieren nun r‘ = R‘ * ceff [Ω/s] und g‘ = G‘ * ceff [1/(Ω*s)]

$$\gamma d=\sqrt{(r'+j\omega z0)\cdot(g'+\frac{j\omega}{z0})}$$

Diese Substitution entspricht auch der allgemeinen Definition der ohmschen Verluste in der HP Literatur:

Offset loss is in GΩ/sec. It is the propagation loss per unit length of the transmission line at a normalization frequency, such as 1 GHz, multiplied by the speed of light in the transmission medium. For coaxial devices, it can be calculated from the loss magnitudedata at 1 GHz.

nährungsweise ergibt sich r‘ zu

$$r'=\sqrt{rg'^2+rs'^2}$$

mit rg‘ = Gleichstromwiderstand [Ω/s] und der Beitrag des Skineffekts rs‘.

rs‘ wird angenähert durch:

$$rs'=s'\cdot\sqrt{f}=s'\cdot\sqrt{f\cdot\frac{f0}{f0}}=s'\cdot\sqrt{f0}\cdot\sqrt{\frac{f}{f0}}=hps'\cdot\sqrt{\frac{f}{f0}}$$ mit **f0 = 1Hz**

dadurch, dass man die Einheit aus dem Wurzelausdruck herauszieht ergibt sich die Einheit von hps‘ zu Ω/s, während s‘ die Einheit $\Omega/\sqrt{s}$ hat. Mit der Wahl der

Konstanten f0 = 1 Hz sind s‘ und hps‘ allerdings zahlenmäßig gleich. Es stimmt übrigens nicht, dass die HP network analyzer den offset loss (= hps‘) in Gigaohm/s erwarten. Man kann zwar bei der manuellen Eingabe die Einheit wählen, intern wird jedoch der Wert Ω/s abgespeichert (HP 8753XX). Die Konstante hps‘ wird nach HP aus der Messung des S21 und des delay bei 1 GHz bestimmt.

g‘ ergibt sich mit gd‘ analog zu rs‘ näherungsweise, nur dass hier der Zusammenhang linear ist, zu:

$g'=gd'\cdot\frac{f}{f0}$ mit f0 = 1 Hz und der Einheit von gd‘ $\frac{1}{\Omega\cdot s}$

somit ergibt sich insgesamt

$$\gamma d=\sqrt{\left(\sqrt{rg'^2+hps'^2\cdot\frac{f}{f0}}+j2\pi f\,z0\right)\cdot\left(gd'\cdot\frac{f}{f0}+\frac{j2\pi f}{z0}\right)}$$

für die Kalibrierstandards, bei denen rg‘ und gd‘ keine Berücksichtigung finden, verkürzt sich der Ausdruck auf:

$$\gamma d=\sqrt{\left(hps'\cdot\sqrt{\frac{f}{f0}}+j2\pi f\,z0\right)\cdot\frac{j2\pi f}{z0}}$$

Ab hier kann man von einer weiteren Umformung absehen, weil die Berechnung numerisch erfolgt und die weitere Umgestaltung der Formel dazu nichts beiträgt.

Damit lässt sich nun der S11 am Ende der Leitung (s11e) auf den S11 am Anfang der Leitung(s11a) zurück transformieren:

$$S11a=S11e\cdot e^{j\gamma d\cdot(-2\cdot delay)}$$

Die Gleichung für den Wellenwiderstand ergibt sich über ein analoges Vorgehen. Aus:

$$Zw=\sqrt{\frac{R'+j\omega L'}{G'+j\omega C'}}$$

durch Substitution wie oben ergibt sich:

$$Zw=\sqrt{\frac{R'+\frac{j\omega z0}{ceff}}{G'+\frac{j\omega}{ceff\,z0}}}$$

Durch Multiplikation des Ausdrucks unter der Wurzel mit ceff/ceff ergibt sich:

$$Zw=\sqrt{\frac{R'ceff+j\omega z0}{G'ceff+\frac{j\omega}{z0}}}$$

Entsprechend der Berücksichtigung der Näherungen ergibt sich analog zu oben:

$$Zw=\sqrt{\frac{\sqrt{rg'^2+hps'^2\cdot\frac{f}{f0}}+j2\pi f z0}{gd'\cdot\frac{f}{f0}+\frac{j2\pi f}{z0}}}$$

verkürzt für die Kalibrierstandards wiederum:

$$Zw=\sqrt{\frac{hps'\cdot\sqrt{\frac{f}{f0}}+j2\pi f z0}{\frac{j2\pi f}{z0}}}$$

Das Ganze nun in Python umgesetzt:

```
def Zw(f, z0, hps, rg=0, gd=0):
    return (complex(math.sqrt(rg**2.0 + hps**2.0 * f), 2*math.pi*f*z0) /
            complex(gd*f, 2*math.pi*f/z0))**0.5
#end zwave



# der komplexe Ausbreitungskoeffizient
def gammadelay(f, z0, hps, rg=0, gd=0):
    return (complex(math.sqrt(rg**2.0 + hps**2.0 * f), 2*math.pi*f*z0)*
            complex(gd*f, 2*math.pi*f/z0))**0.5
#end gammadelay



# Berechnung des transformierten s11 am TX-Messport (reflektierte Welle) des VNA
# za ist die Abschlussimpedanz der Leitung
# ZR ist die Systemimpedanz und in der Regel = z0
def s11(f, za, delay, z0, hps, rg=0, gd=0):
    """
    zline = Zw(f, z0, hps, rg, gd)
    s11e = (za-zline)/(za+zline)
    s11a = s11e * math.e**(-2*gammadelay(f, z0, hps, rg, gd)*delay)
    zin = zline *(1+s11a)/(1-s11a)
    return (zin-ZR)/(zin+ZR)
    Die vorstehende Berechnung liefert den selben Wert wie die nachfolgende,
    die in der HP-Literatur verwendet wird
    """
    gt = (za-ZR)/(za+ZR)
    zc = Zw(f, z0, hps, rg, gd)
    g1 = (zc-ZR)/(zc+ZR)
    eg = math.e**(-2*gammadelay(f, z0, hps, rg, gd)*delay)
    eg1 = 1.0-eg
    return (g1*(eg1-g1*gt)+eg*gt)/(1.0-g1*(eg*g1+gt*eg1))
#end s11

def s11open(f, delay, z0, hps, C0, C1, C2, C3):
    c = C0*1e-15+C1*1e-27*f+C2*1e-36*f**2+C3*1e-45*f**3
    if c == 0.0:
        za = 1.0
    else:
        za = complex(0, -1/(2*math.pi*f*c))
    #end if
    return s11(f, za, delay, z0, hps)
#end s11open
```

```
def s11short(f, delay, z0, hps):
    return s11(f, 0.0, delay, z0, hps,)
#end s11short
```

```
# Parallelschaltung von R und parasitärer Kapazität
def s11load(f, delay, z0, hps, r, c):
    za = 1.0/complex(1.0/r, 2*math.pi*f *c)
    return s11(f, za, delay, z0, hps)
#end s11load
```