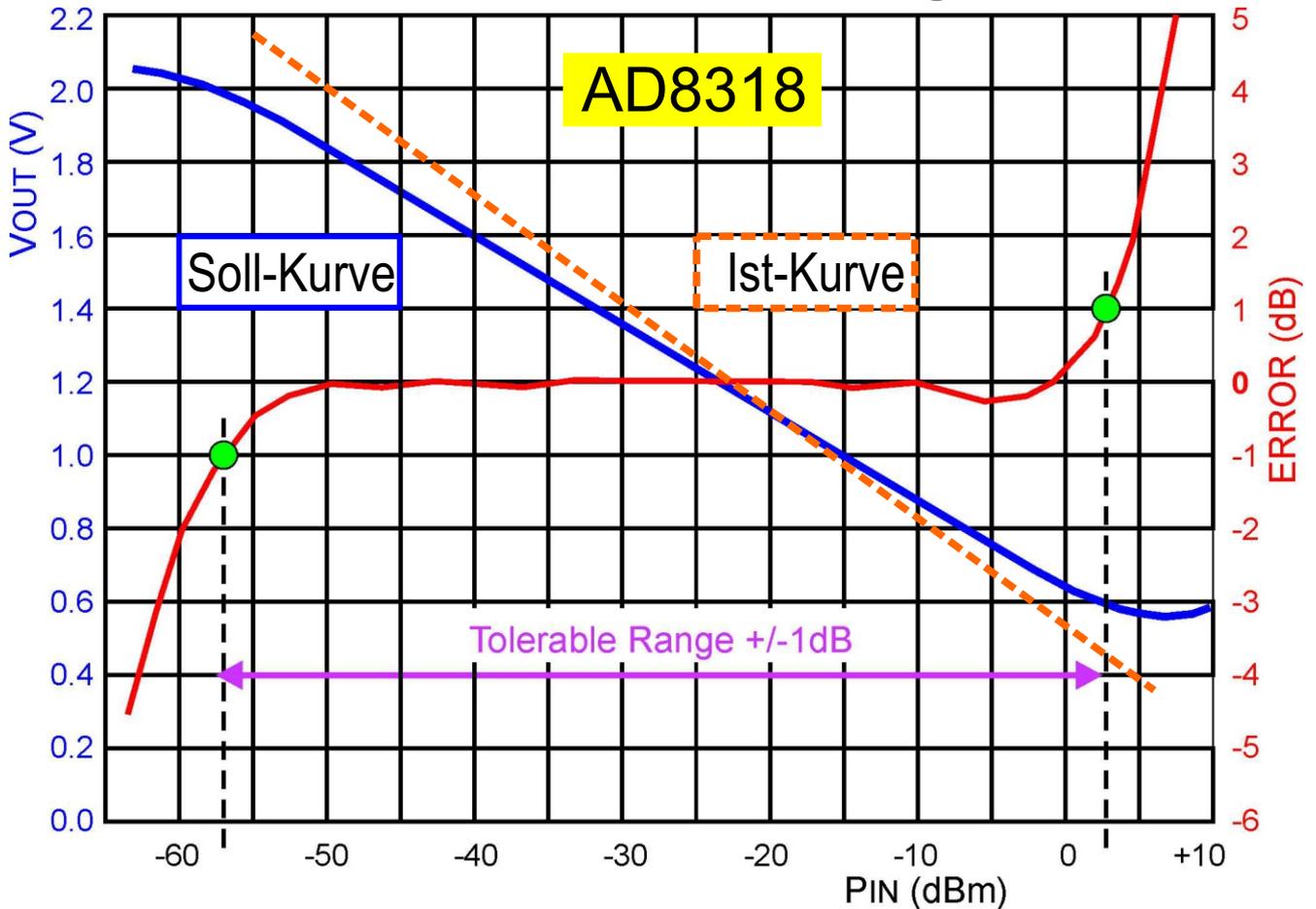


AD8318 - Kalibrierung

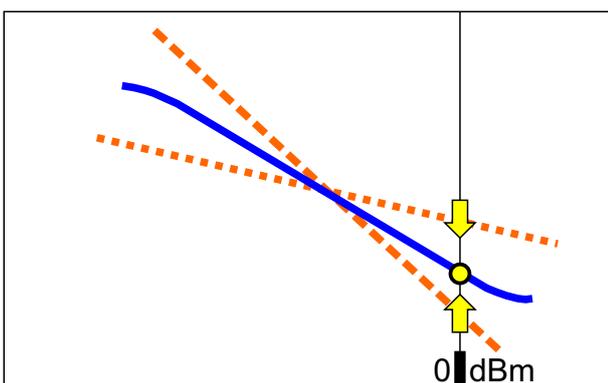


Im Datenblatt des AD8318 sind die 6 Eichkurven nur graphisch abgebildet. Für die Verwendung der Kurven in Software müssen diese als mathematischer Ausdruck vorliegen. Da die Sollkurve im tolerablen Bereich nahezu linear ist kann hierzu die Geradengleichung $y = m \cdot x + c$ benutzt werden. Der AD8318 gibt die Spannung V_{OUT} (y) aus. Gesucht ist jedoch PIN (x).

Deshalb muss die Gleichung umgestellt werden:

(Syntax im Arduino Sketch siehe S.2)

$$x = (y - c) / m$$

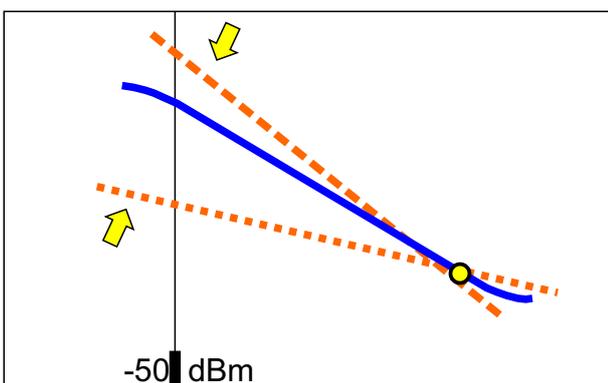


Für die Eichung ist ein präziser Generator erforderlich.

Schritt 1 :

Generator auf 0 dBm einstellen.

Die Konstante c in der Gleichung verschiebt die Ist-Kurve parallel nach oben oder unten. Wert von c erhöhen bzw. verringern bis das Display 0 dBm anzeigt.



Schritt 2 :

Generatorpegel auf den tiefsten Wert im linearen Bereich der Eichkurve einstellen, z.B. auf -50 dBm.

Die Konstante m in der Gleichung dreht die Ist-Kurve rechts bzw links um den 0 dBm-Punkt.

Wert von m erhöhen bzw. verringern bis das Display -50 dBm anzeigt.

$$x = (y - c) / m$$

```
void calculate_power_CH_1()    // used straight line equation:
                              // y = mmm*x + ccc -> x = (y - ccc)/mmm
{
    level_CH_1 = - (voltage_CH_1 - ccc) / mmm + att_CH1;
    level_CH_1 = floor(level_CH_1 + 0.5);           // round and convert to integer
    power_W_1 = pow(10,level_CH_1/10);             // convert dBm to mW
}
```

```
void AD8318_use_curve_2ghz2()
{
    mmm = 0.025;
    ccc = 0.45;
    error_limit_LOW = -60;
    error_limit_HIGH = -4;
}
```

PS: Das Minus-Zeichen vor - (voltage_CH_1) habe ich verwendet da beim AD8313 die Steigungen der Kurven negativ sind. Dadurch konnten die 6 mmm-Werte positiv eingetragen werden.