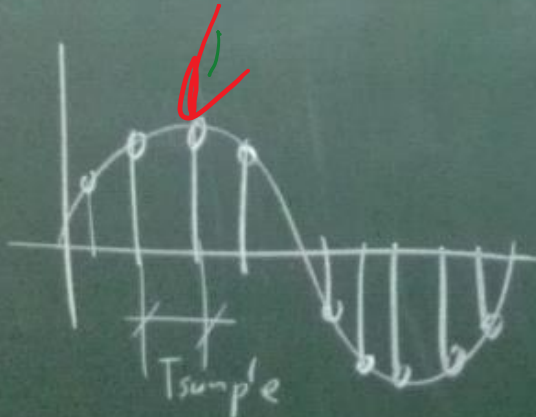


1 Digitaler Funktionsgenerator


Digitale
Funktionsgenerator



$$F_{\text{sample}} = \frac{1}{T_{\text{sample}}}$$

Festliche Formeln für dig. Functen siehe ProcVis-bl-VS.doc

Parameter (einstellbar)

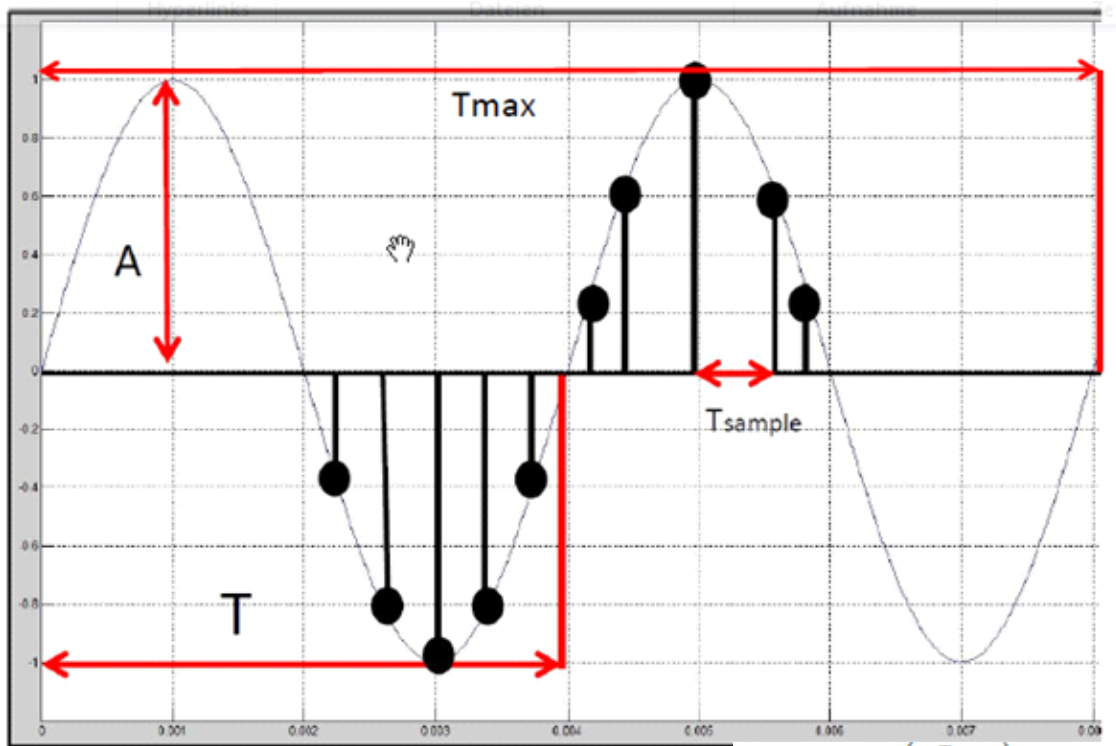
- Frequenz (Periodendauer)
- Amplitude fix auf 1.0
Einstellbare Amplitude durch
nachgestelltes Koeffizientenglied
- Signalform 
(Tastverhältnisse)

Gute Faustformel: $F_{\text{sig max}} = \frac{F_{\text{sample}}}{10}$

Abtastfrequenz ist bei dig. Funktionsgeneratoren
so wichtig, daß die F_{sig} immer bezogen auf
 F_{sample} angegeben wird

1b Abgetastete periodische Signale

Abgetastete periodische Signale



$$F_{sig} = \frac{1}{T_{sig}}$$

$$F_{sump} = \frac{1}{T_{sump}}$$

F_{sig} ...Signalfrequenz

F_{sump} ...Abtastfrequenz

A ...Signal-Amplitude

T_{mux} ...Länge des Signals in sec

mel

T_{samp}

$\left(\frac{F_{sig}}{F_{sample}}\right) \dots$ Signalfrequenz bezogen auf die Abtastfrequenz ist immer zwischen 0 und 0,5

$$\frac{\text{Points}}{\text{Periode}} = \frac{1}{\frac{F_{sig}}{F_{sample}}} \dots \text{Abtastpunkte pro Signalperiode}$$

Mit F_{sump} und T_{sump} ist F_{samp} und T_{samp} gemeint

Bsp.: ein 10Hz Sinus Signal mit 100Hz abgetastet.

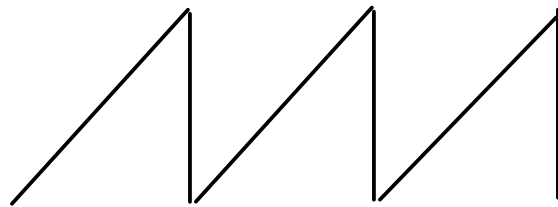
$F_{sig}/F_{sample} = 0,1$; $\text{Points}/\text{Period} = 10$;

Erkenntnisse aus BspSignale\Ue1.m

- Wir sollten verstanden haben was aliasing ist und wie sich die Quantisierung auswirkt
- Das Ohr ist ein Fourier-Analysator

2 Sägezahn, Dreieck, Sinus, Rechteckgenerator mit objektorientierter Programmierung

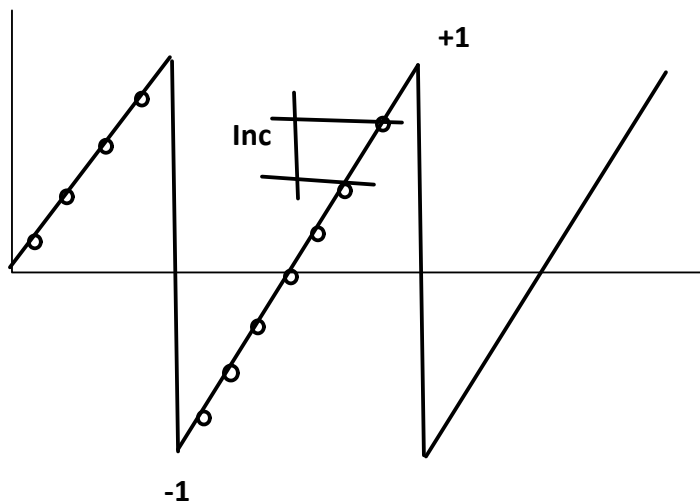
Baisklasse: IFuncGen



← Rampen oder Sägezahngenerator

Alle Funktionsgeneratoren liefern die Amplitude 1
Lautstärke durch nachgeschaltete Multiplikationen (Abschwächer)

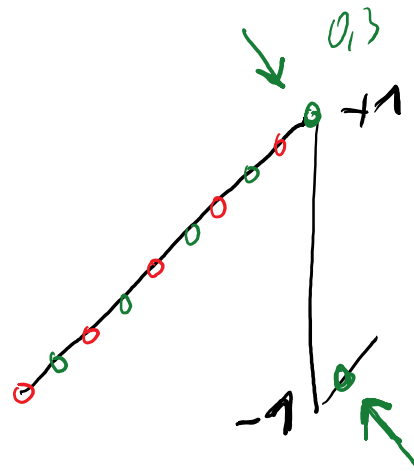
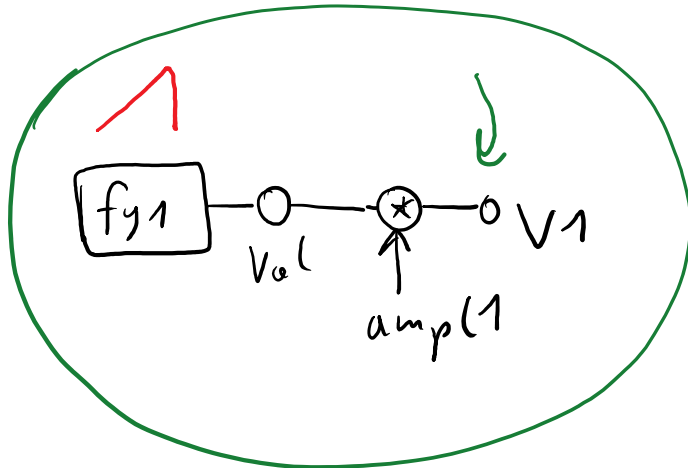
FloatRampGenSwd



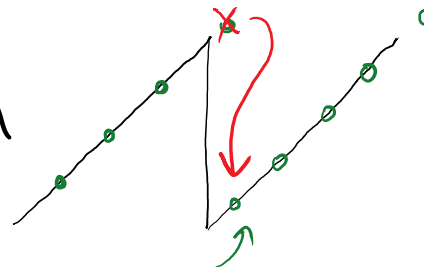
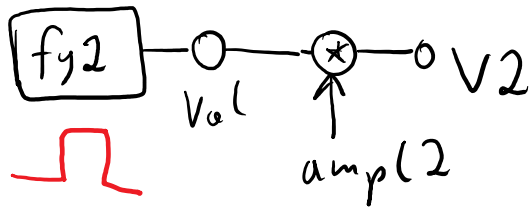
```
void CalcOneStep()
{
    val = val + Inc;
}
```

```
Inc = 2.0/PointsPerPeriod
Inc = 2.0*(Fsig/Fsample)
```

2b Weitere Details zum Sägezahn Beispiel



```
void SignedRampGen::CalcOneStep()
{
    val = val + _inc;
    if( val > 1.0 )
        val = -1 + (val - 1.0);
}
```



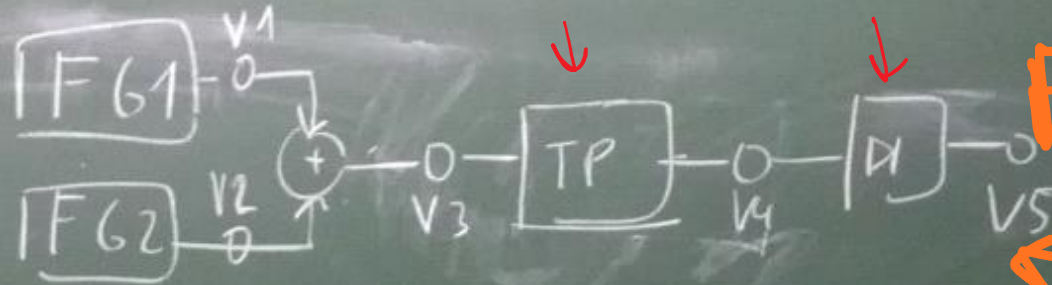
Um den Betrag den val über **+1** geht
wird bei **-1** korrigiert

2 0,1 0,01 ← Frequ
 fy1 fy2

3 0,5 0,7 ← Ampl

3 Signalverarbeitungskette Echtzeit vs. Simulation

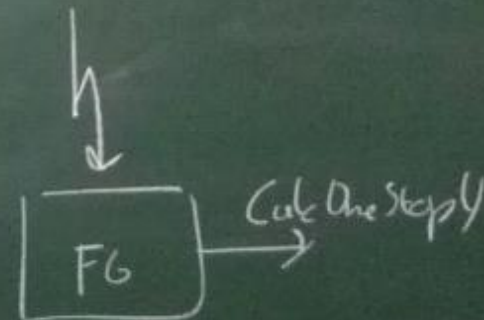
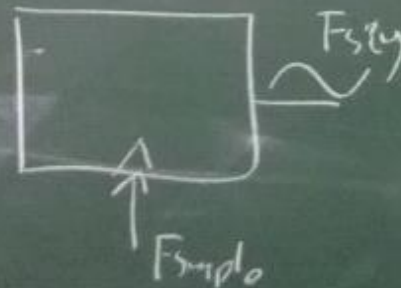
Signalverarbeitungskette



Calc One Step

Exec Signal Chain

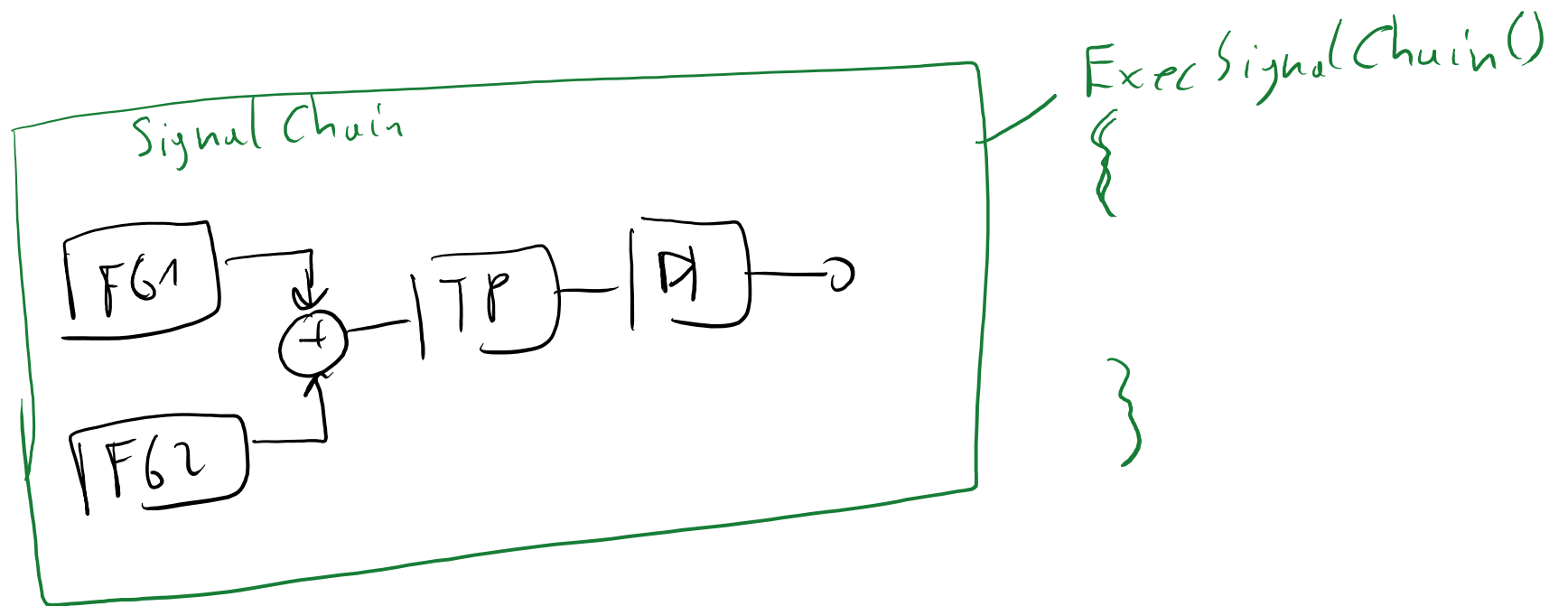
$F_{sig} = 171 \text{ kHz}$
 $F_{smp} = 44 \text{ kHz}$



Calc One Step() wird mit F_{smp} aufgerufen

- Echtzeit: Signalverarbeitungskette wird mit 40 kHz aufgerufen
- Simulation/Analyse: —//— wird mit 100 Hz aufgerufen

3.1 Signalverarbeitungskette Echtzeit vs. Simulation



ExecSignalChain:

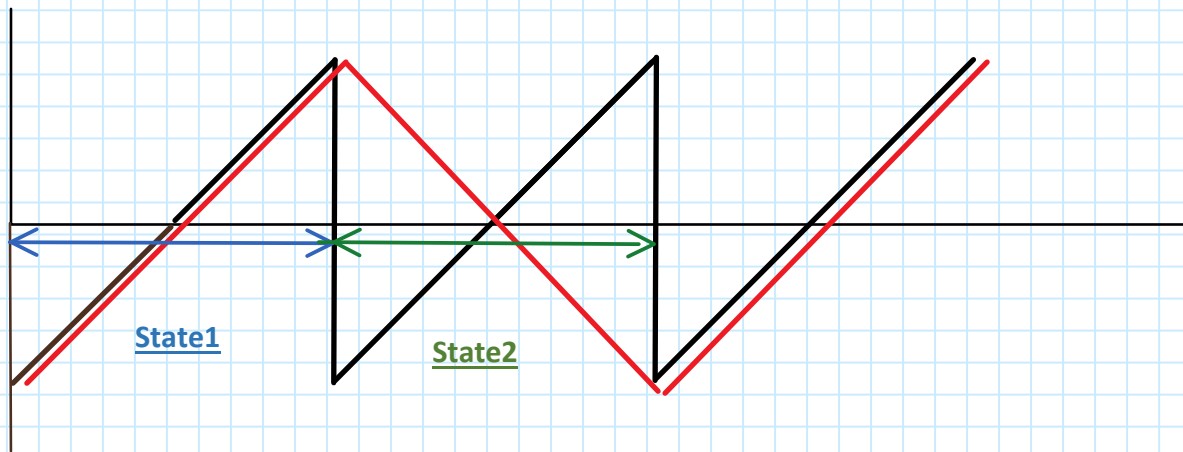
Kann mit 100Hz Timer

oder 44Khz Timerinterrupt aufgerufen werden



4 Triangle Generator

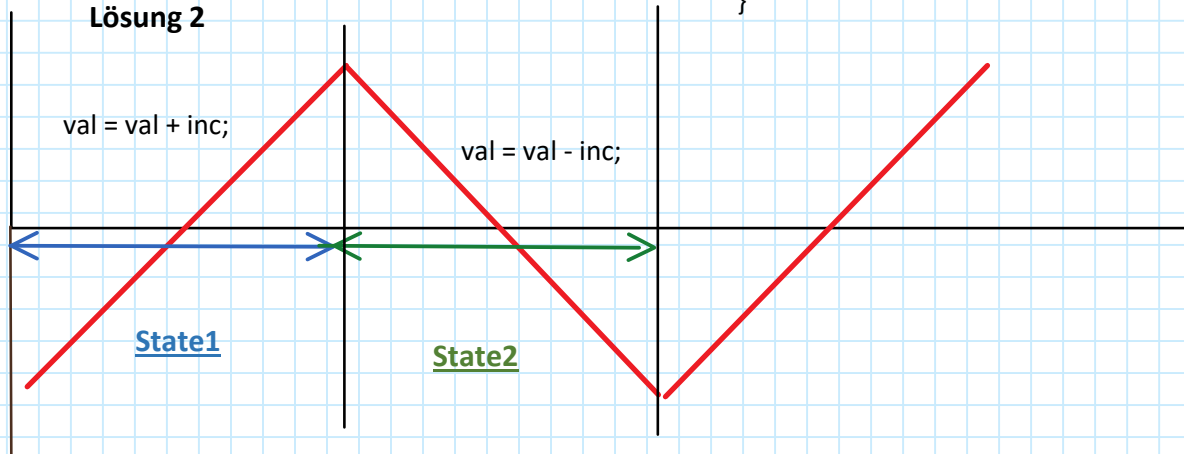
Lösung 1



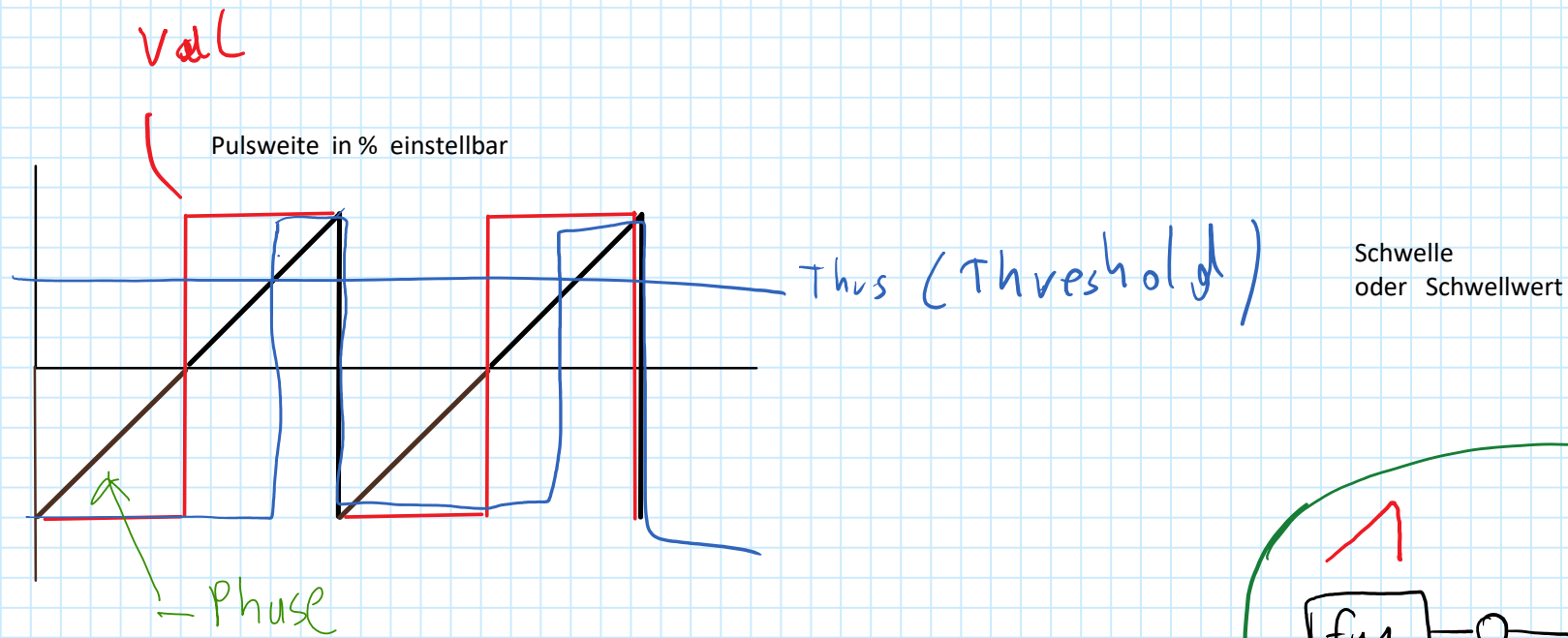
- **phase** = Sägezahn
- **val** = Ausgangswert des Dreieckgenerators
- Im State1 gilt: **val** = **phase**
- Im State2 gilt: **val** = **--phase**

```
void TriangleGen::SetPointsPerPeriod(int aPoints)
{
    _inc = 4.0/aPoints;
}
```

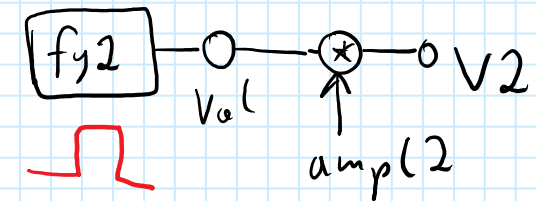
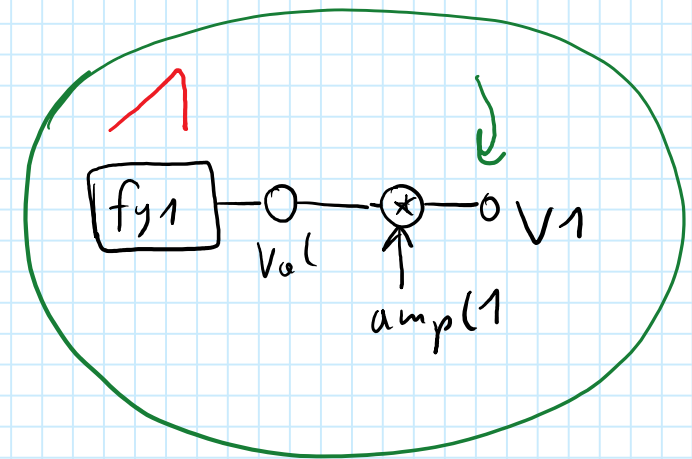
Lösung 2



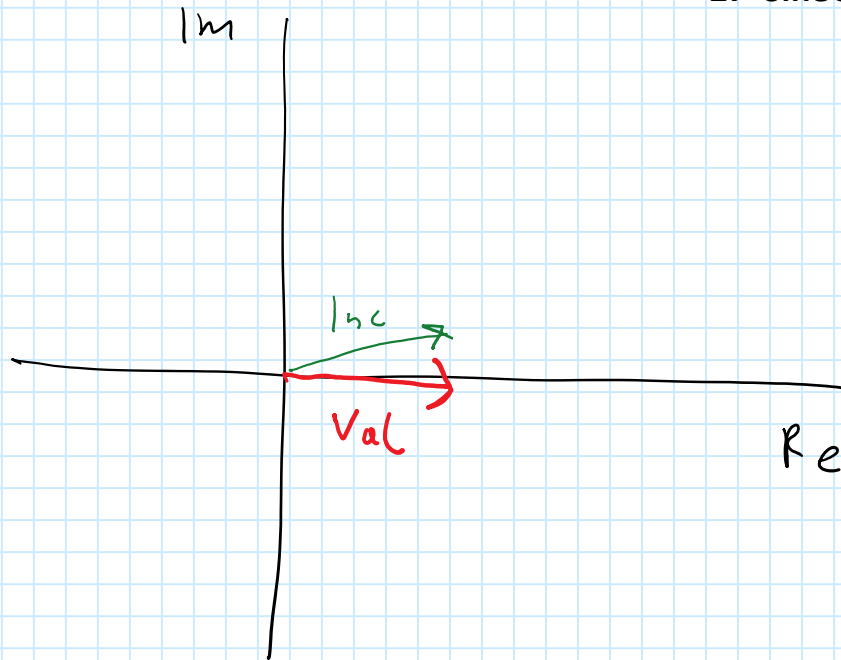
5 Rechteck Funktionsgenerator



```
If _phase < 0  
    val = -1;  
else  
    val = +1;
```



27 SineGen1

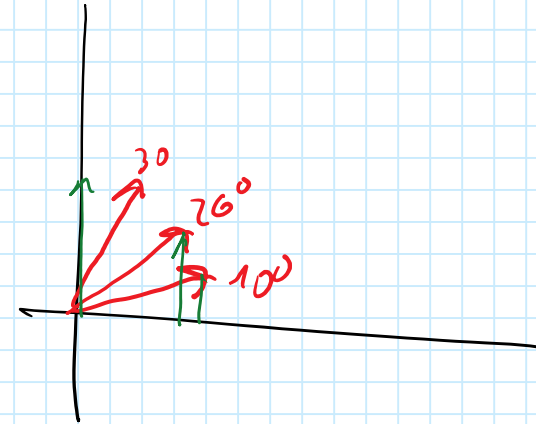


$$Val = 1 \angle 0^\circ$$

$$Inc = 1 \angle 10^\circ$$

$$Val = Val * Inc;$$

$$Val = Val \times Inc;$$



$$Doc = Val \cdot \sin // \sin$$

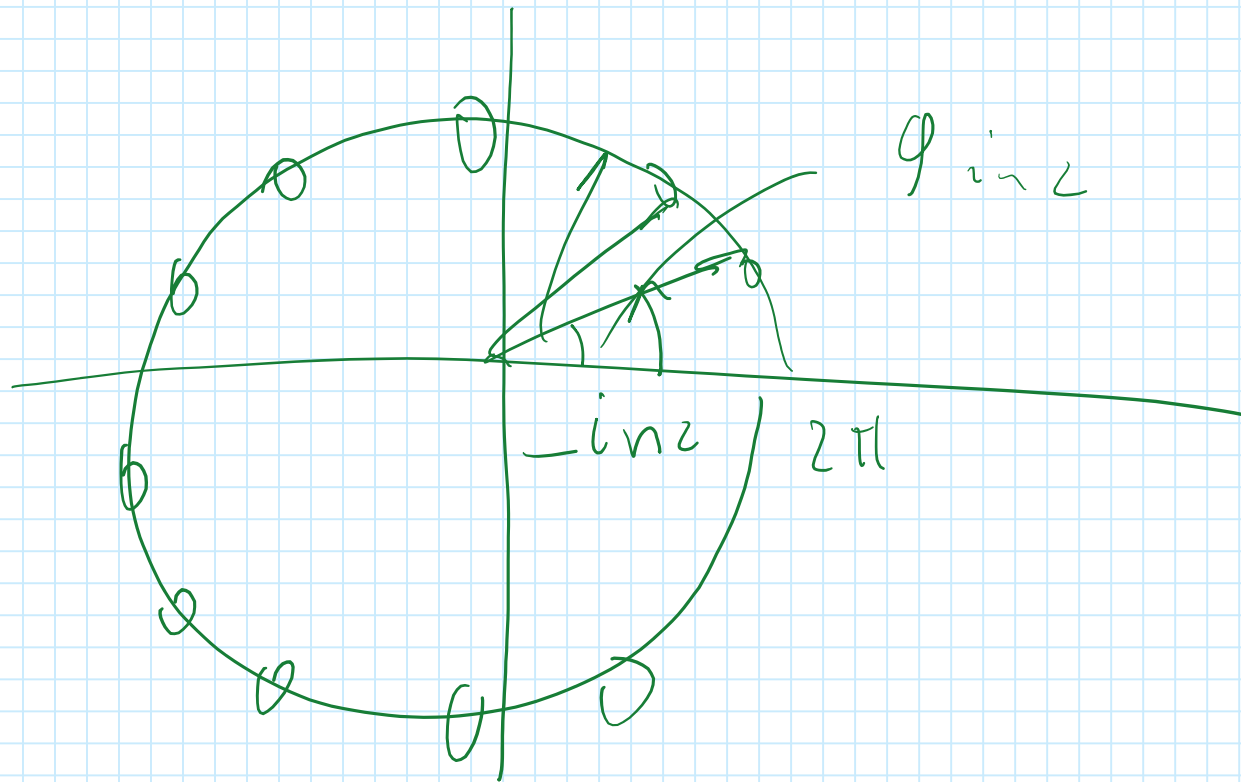
$$Doc = Val \cdot \cos // \cos$$

$$10 \text{ Plote} / \text{Pa'ole} = inc = 36 = \frac{2\pi}{10}$$

$$y = y + inc;$$

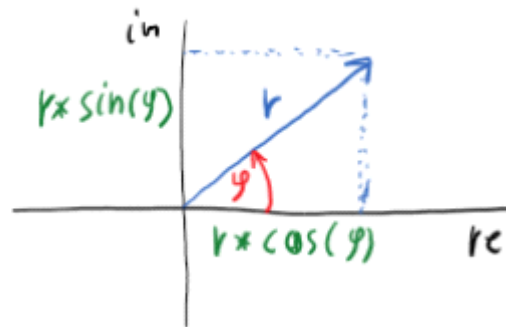
$$Val = \sin(y);$$

Complex Multiplikation in Rechteck-Koordinaten



29 SineGen3

$$(Ar + j \cdot Ai) \cdot (Br + j \cdot Bi) = (Ar \cdot Br - Ai \cdot Bi) + j \cdot (Ai \cdot Br + Ar \cdot Bi)$$



Reset:

`val = val*_inc;`

`val = val*_inc;`



28 SineGen4

```
class Complex {
public:
    float re, im;
public:
    Complex()
    { re=0; im=0; }

    void SetPolar(float aR, float aPhi)
    { re = aR*cos(aPhi); im = aR*sin(aPhi); }

    Complex Mult(Complex aB)
    {
        Complex prod;
        prod.re = re*aB.re - im*aB.im;
        prod.im = im*aB.re + re*aB.im;
        return prod;
    }
};
```

```
class SineGen {
public:
    Complex _inc;
public:
    Complex val;
public:
    SineGen()
    { Init(); }

    void Init()
    { Reset(); SetPointsPerPeriod(50.0); }

    void Reset()
    { val.SetPolar(1.0,0.0); }

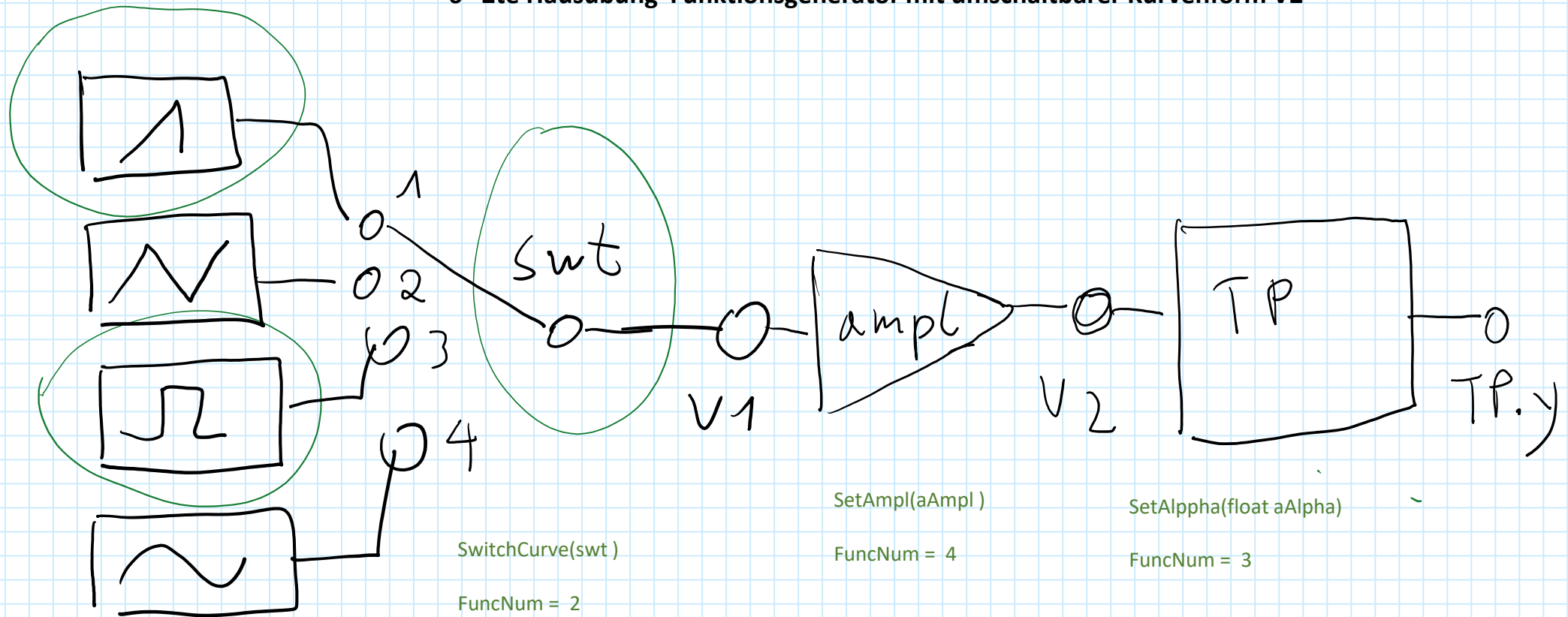
    void SetPointsPerPeriod(float aPoints)
    { _inc.SetPolar(1.0, (2*M_PI)/aPoints); }

    void SetFrequ(float aFrequ)
    { SetPointsPerPeriod(1.0f/aFrequ); }

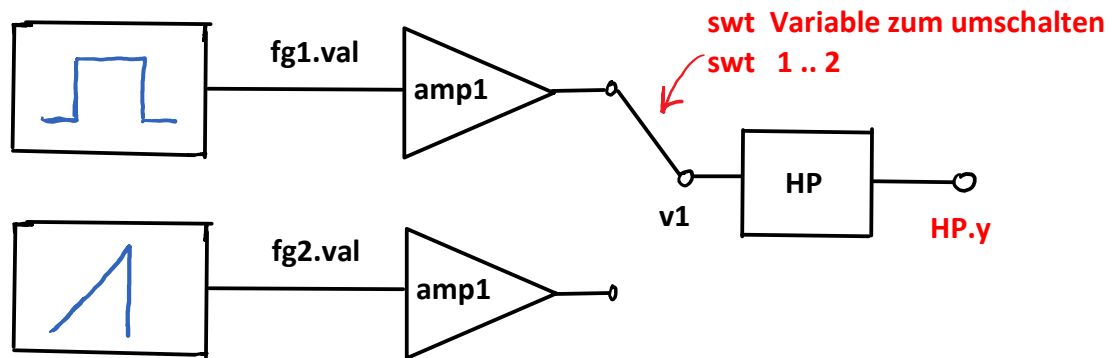
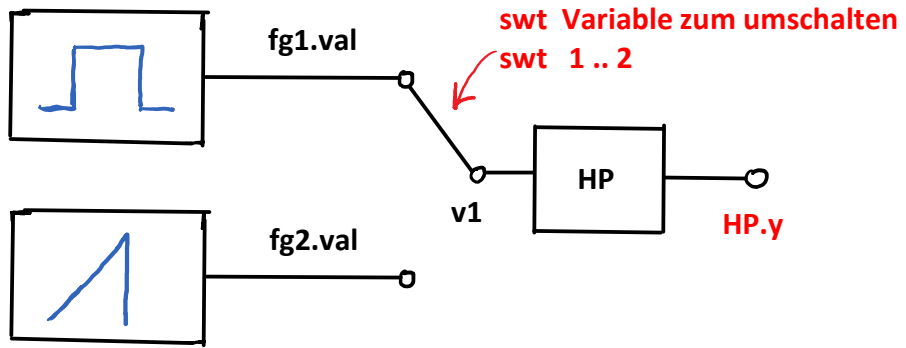
    void CalcOneStep()
    { val = val.Mult(_inc); }
};
```

29 SineGen5

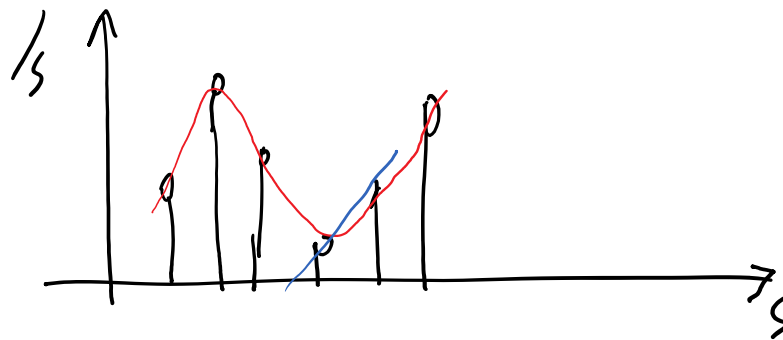
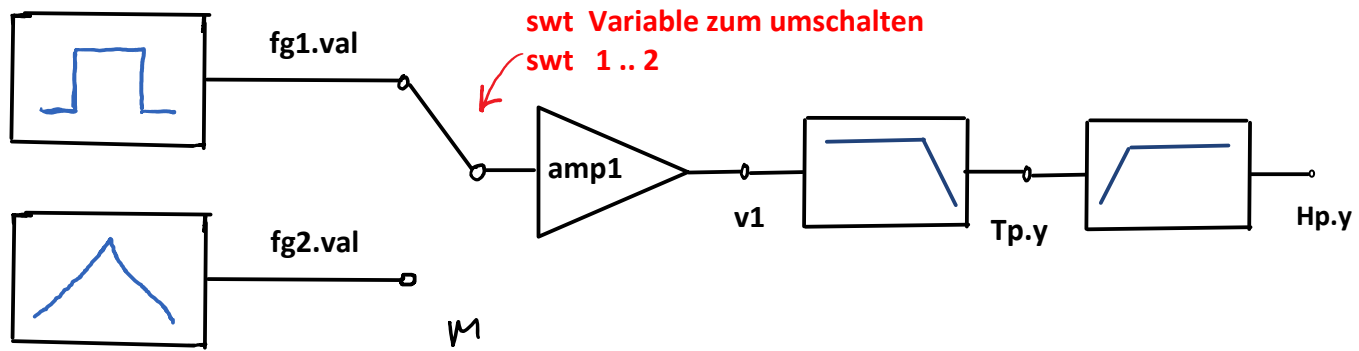
6 2te Hausübung Funktionsgenerator mit umschaltbarer Kurvenform V2



7 Schaltung2



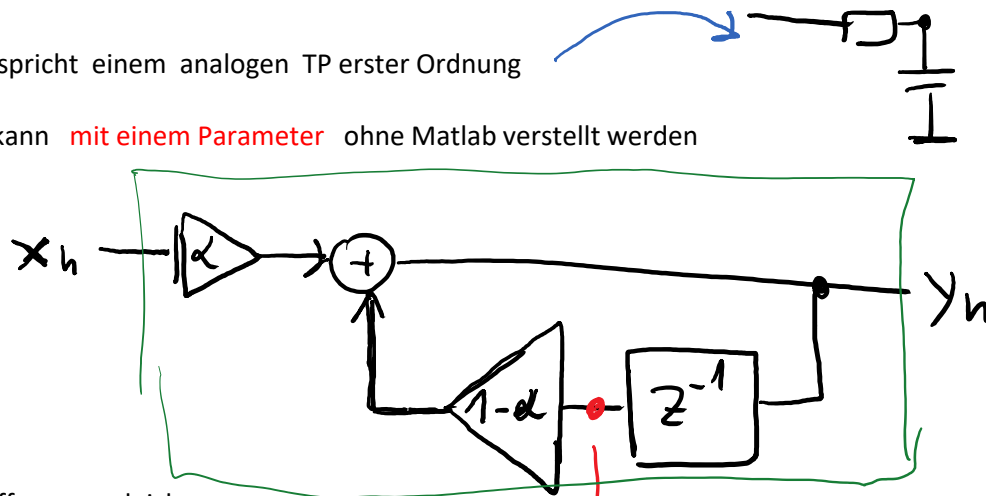
8 Schaltung 3



9 Supereinfacher TP 1er Ordnung

Entspricht einem analogen TP erster Ordnung

Fg kann mit einem Parameter ohne Matlab verstellt werden



Differenzengleichung:

$$y_n = \alpha \cdot x_n + \underbrace{(1-\alpha)}_{\beta} \cdot y_{n-1}$$

Übertragungsfunktion:

$$Y(z) = \alpha \cdot X(z) + (1-\alpha) \cdot z^{-1} \cdot Y(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\alpha}{1 - (1-\alpha) \cdot z^{-1}}$$

y_{n-1}

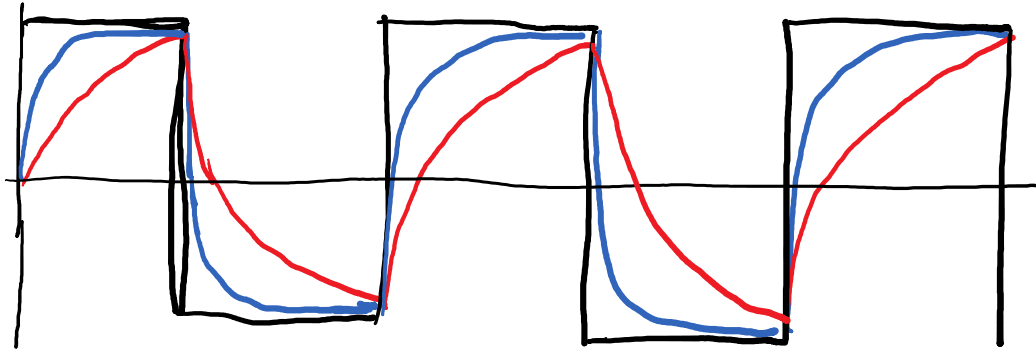
Frequenzgang: für $z^{-1} \rightarrow e^{-j\omega \cdot T_s}$ einsetzen

\triangleright mit $K_{\text{Konst.}}$

\oplus Addieren

$\boxed{z^{-1}}$ verz. um eine Abtastzeit

10 Antwort des Filters auf ein Rechtecksignal

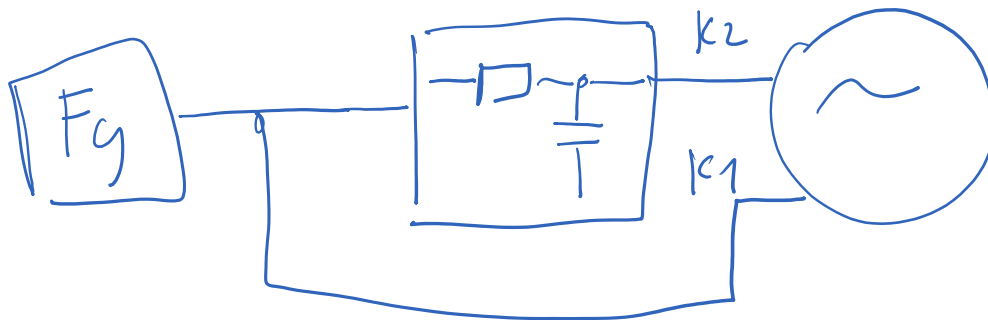


— τ klein f_y groß
— τ groß f_y klein

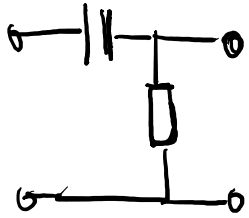
$$f_y = \frac{1}{\tau}$$

$$\tau \approx \frac{1}{\omega_y}$$
$$\tau \approx \frac{1}{\alpha}$$

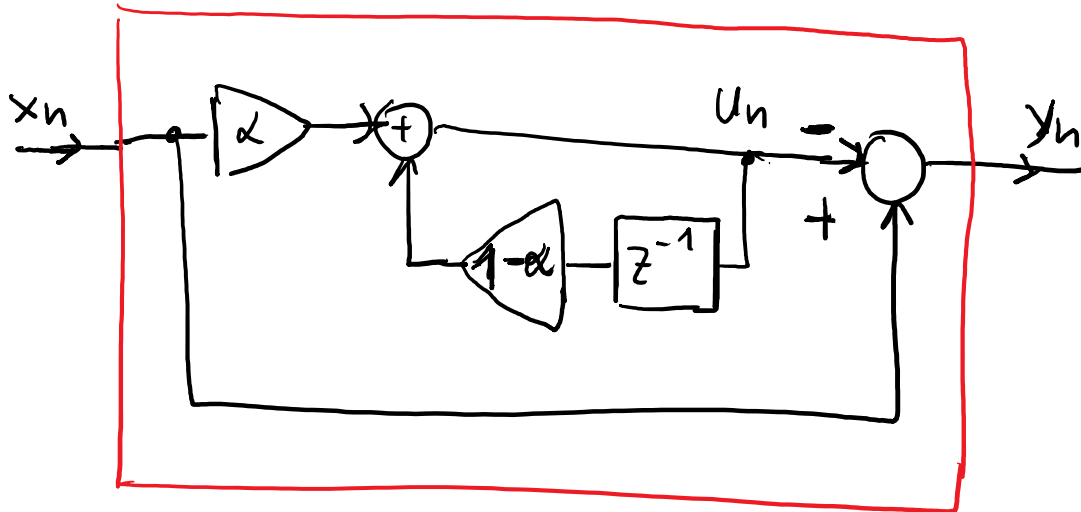
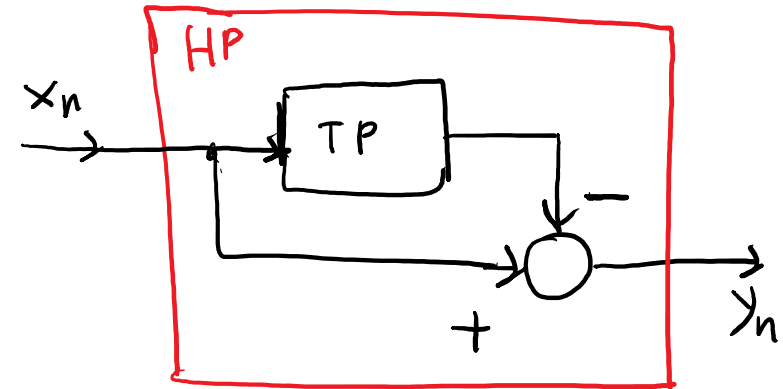
großes $\alpha \rightarrow$ kleines τ
kleines $\alpha \rightarrow$ großes τ



11 Digitale Version eines HP 1er Ordnung

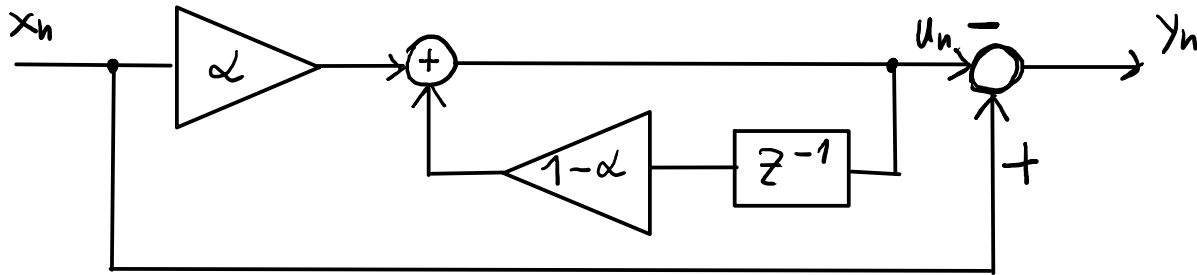


HP erhält man indem man das Ausgangssignal eines TP vom Gesamtsignal subtrahiert



$$\left. \begin{aligned} u_n &= x_n \cdot \alpha + u_{n-1} \cdot (1-\alpha) \\ y_n &= x_n - u_n \end{aligned} \right\} \text{Differenzengl. für Progr. am MC}$$

12 HP 1er Ordnung



$$u_n = x_n \cdot \alpha + u_{n-1} \cdot (1 - \alpha)$$

$$y_n = x_n - u_n$$

13 HP 1er Ordnung

$$\left. \begin{aligned} u_n &= x_n \cdot \alpha + u_{n-1} \cdot (1-\alpha) \\ y_n &= x_n - u_n \end{aligned} \right\} \text{Differenzengl. für Progr. am } nC$$

~~$$u = x - y$$~~

← neu berechnen

~~$$x - y = x \cdot \alpha + y \cdot z^{-1} \cdot (1-\alpha)$$~~

$$b_0 = (1-\alpha) \quad b_1 = -(1-\alpha)$$

$$a_0 = 1 \quad a_1 = -(1-\alpha)$$

↳ noch Umformung $H(z) = \frac{y}{x} = \frac{(1-\alpha) - (1-\alpha) \cdot z^{-1}}{1 - (1-\alpha) \cdot z^{-1}} = \frac{b_0 + b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}}$

Antwort des HP Filters auf ein Rechtecksignal

