

Tiefpassfilter erster Ordnung ("Exponentialfilter")

$T_{\max} := 5 \cdot s$ Eingestellte Simulationszeit (in Sekunden) $dt := 0.05 \cdot s$ Zeitschritt der Simulation

$n_{\max} := \text{rund}\left(\frac{T_{\max}}{dt}\right)$ Berechnung der Anzahl der Simulationsschritte $n_{\max} = 100$

$n := 0..(n_{\max} - 1)$ Erzeugung eines Laufzeit-Indizes "n" und des Zeitvektors "t" $t_n := n \cdot dt$

$In_n := 1$ Erzeugung und Initialisierung eines Vektors "In" mit n_{\max} Elementen als Einheitssprung

$\text{Tau} := 1 \cdot s$ Festlegung der Zeitkonstante für die Filterfunktion

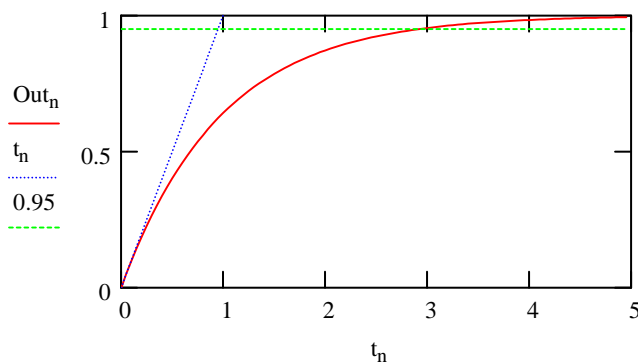
$dt_div_Tau := \frac{dt}{\text{Tau}}$ Berechnung einer Hilfsgröße für den Algorithmus $dt_div_Tau = 0.05$

Beschreibung des Algorithmus

```
ExpFi(In) :=
  Erg_0 ← 0
  z ← 1
  while z < n_max
    Erg_z ← Erg_{z-1} + dt_div_Tau · (In_z - Erg_{z-1})
    z ← z + 1
  Erg
```

Berechnung des Ergebnisvektors: $\text{Out} := \text{ExpFi}(\text{In})$

Darstellung des Ergebnisses:



Bemerkungen:

Es ist gut erkennbar, wie die Nullpunkt-Tangente an den Anstieg den Endwert beim Zeitwert Tau (1s) schneidet. Ebenso gut ist erkennbar, dass bei 3 Tau 95% vom Endwert erreicht sind. Nach allem, was wir in der Schule gelernt haben: die Reaktion auf den Einheitssprung ist klar die Exponentialfunktion $1 - \exp(-t/\text{Tau})$.

Nun kann dieser Algorithmus zur Verarbeitung verrauschter Eingangssignalverläufe verwendet werden.

$\text{In_cos}_n := \text{rnd}(\sin(n \cdot 0.025))$ $\text{Out_cos} := \text{ExpFi}(\text{In_cos})$

