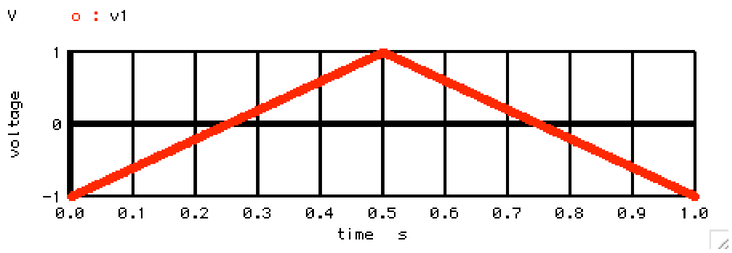
*=============**Transient_Timing_PWL_1msec**===========

It is not so easy to see the inconsistent timing.

```
=====================================================================
*V_PWL#  NODE_P NODE_N DC     VALUE  PWL(   T1   V1   T2     V2   T3     V3     ...>)
V_PUL    V1     0      DC     0      PWL(   0    -1   .25    0    .5     1      .75  0 1 -1 )
*TRAN    TSTEP  TSTOP  TSTART TMAX   ?UIC?
.tran    100m   1      0      1m
plot     v1
plot     v1      pointplot
```



===

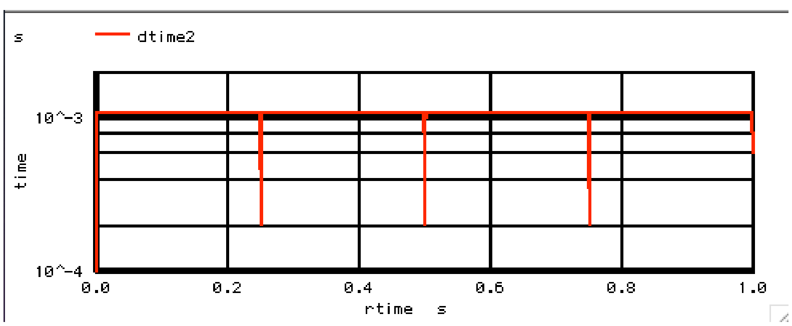
This is where constructing a timing plot comes in handy.

```
=====================================================================
let      num = length(time)-2
compose dtime start = 0 stop = $&num step =1
compose rtime start = 0 stop = $&num step =1
let      i = 0
repeat   $&num
let      i = i +1
let      dtime[i] = time[i +1] -time[i]
let      rtime[i] = time[i]
end
let      dtime2 = abs(dtime)+100u
plot     dtime2 vs rtime ylog
```

===

The timing is only changing an order of magnitude.
But it is enough to effect the RMS value. There must be enough small timing points at each time point of the PWL wave to be able to do this.

```
=====================================================================
let      vrms1_cdhw = sqrt(mean(v1*v1))
echo    "INPUT RMS PWL prelinear   =   $&vrms1_cdhw"
```



INPUT RMS Square prelinear   =   0.583932

===

The **Linearize** statement appears to limit the small changes to just the beginning of the waveform. So the RMS comes much closer to perfection.

```
=====================================================================
linearize
plot     v1     pointplot

let      vrms1_cdhw = sqrt(mean(v1*v1))
echo    "INPUT RMS PWL postlinear   =   $&vrms1_cdhw"
```

==
INPUT RMS Square postlinear = 0.577929

sqrt(0.3333333333333333) = 0.5773502691896257

===========Full_Netlist_For_Copy_Paste======================

```
RMS_PWL_!ms
.Option srcsteps = 1 set Gmin = 1.0000E-02
*=======Circuit_Netlist========================
V_PUL   V1    0     DC     0    PWL( 0  -1 .25 0 .5 1 .75 0 1 -1 )

*TRAN   TSTEP TSTOP TSTART TMAX ?UIC?
.tran   1m    1     0      1m
.control
run
set     pensize = 2
plot    v1    pointplot
plot    v1

let     vrms1_cdhw = sqrt(mean(v1*v1))
echo   "INPUT RMS PWL prelinear  =  $&vrms1_cdhw"

let     num = length(time)-2
compose dtime start = 0 stop = $&num step =1
compose rtime start = 0 stop = $&num step =1
let     i = 0
repeat  $&num
let     i = i +1
let     dtime[i] = time[i +1] -time[i]
let     rtime[i] = time[i]
end
let     dtime2 = abs(dtime)+100u
plot    dtime2 vs rtime ylog

linearize
plot    v1    pointplot
let     vrms1_cdhw = sqrt(mean(v1*v1))
echo   "INPUT RMS PWL postlinear  =  $&vrms1_cdhw"

let     num = length(time)-2
compose dtime start = 0 stop = $&num step =1
compose rtime start = 0 stop = $&num step =1
let     i = 0
repeat  $&num
let     i = i +1
let     dtime[i] = time[i +1] -time[i]
let     rtime[i] = time[i]
end

plot    dtime vs rtime


.endc
.end
```

7.12.10_10.31AM

dsauersanjose@aol.com
Don Sauer
http://www.idea2ic.com/