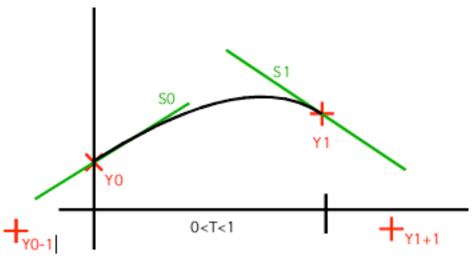
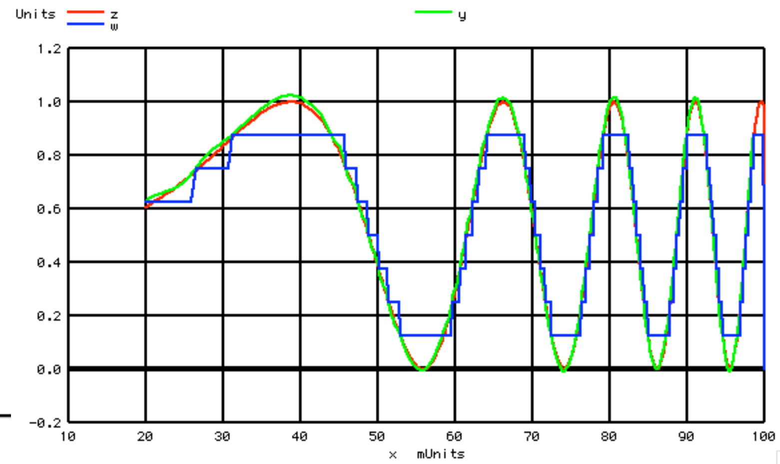
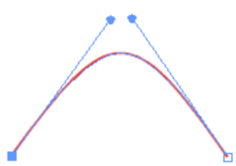
========Extrapolation_Actually_Works=================

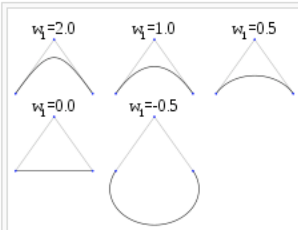




The extrapolation requires four points. Beginning and end slopes need to be provided.
The curve on the right shows the extrapolated curve in green on top of the input curve in red.
The data points for the green curve are where the blue curve touches the green curve.
There are only about 60 points.

This is using to simplest version of curve fitting as possible.
Even with that, the two curves come close.
Remember, the data points consist of a precise voltage values at a precise time points.
And there are more advanced interpolation methods which can include a weight value.
So given the sampling only takes place when something happens,
how well can the original signal get reconstructed?





Sections of conic sections represented exactly by rational Bézier curves

Rational Bézier curves

The rational Bézier curve adds adjustable weights to provide closer approximations to arbitrary shapes.
The **numerator is a weighted Bernstein-form Bézier curve** and the denominator is a weighted sum of Bernstein polynomials.
Rational Bézier curves can, among other uses, be used to represent segments of conic sections exactly.[4]
Given n + 1 control points Pi, the rational Bézier curve can be described by:

$$\mathbf{B}(t) = \frac{\sum_{i=0}^{n} b_{i,n}(t)\mathbf{P}_i w_i}{\sum_{i=0}^{n} b_{i,n}(t) w_i}$$

or simply

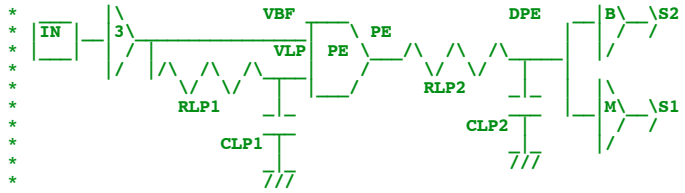
$$\mathbf{B}(t) = \frac{\sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} \mathbf{P}_i w_i}{\sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} w_i}.$$

========Spice_Code=================================

```
Level_Crossing_Freq_Sweep_Extrapolated
**
*       VIN
*
*      __|_________________|      __________|   |
*     /  |  \    |  TH0   _|_     |  TH1   _|_  |  TH0
*    / /-\ \     |  |\   / _ \    |  |\   / _ \ |  |\         _______________
*    \     \/ /  |  | \ -\     /  |  | \ -\     /  | \       |        |      \
*     \_______/  |__|  >- \___/ __|__|  >- \___/ __|  >-     |        |  DAC  \__|VOUTA|
*         |         | /  |  OFF0     | /  |  OFF1   | /  |   |   _____|       /  |_____|
*        ///        |/   |           |/   |         |/   |   |  |     |      /
*                        |D0 |            |D1 |          |D2 |__|  |  __|_____/
*                        |___|            |___|          |___|     |  |
*                          |                |__________________________|  |
*                          |__________________________________________|
```

```
*

*=========Create_Signal==================
Vtime     Vtime  0      DC     0      PWL( 0 0 1 1)
BVIN      VIN    0      V =          .5*sin(27000*(V(Vtime))^3) +.5

BTH0      D0     0      V =    u( V(VIN)  -1/2)
BOFF0     VIN    OFF0   V =    V(D0)/2
BTH1      D1     0      V =    u( V(OFF0) -1/4)
BOFF1     OFF0   OFF1   V =    V(D1)/4
BTH2      D2     0      V =    u( V(OFF1) -1/8)
BOFF2     OFF1   OFF2   V =    V(D2)/8
BTH3      D3     0      V =    u( V(OFF2) -1/16)
BOFF3     OFF2   OFF3   V =    V(D3)/16
BTH4      D4     0      V =    u( V(OFF3) -1/32)
BDAC      VOUTA  0      V =    (V(D0)/2+V(D1)/4+V(D2)/8 )

Btrig     trig   0      V =    1*u( V(VOUTA)- V(VIN) +5m )
CHP       VOUTA  HP     .1n
RHP       HP     0      100K
BTRIG2    trig2  0      V = u( 100*abs(V(HP))  -.1   )

XTOGLE    trig2   D0B    TOGLE
XTOGLE0   D0B     D1B    TOGLE
XTOGLE1   D1B     D2B    TOGLE
XTOGLE2   D2B     D3B    TOGLE
XTOGLE3   D3B     D4B    TOGLE
XTOGLE4   D4B     D5B    TOGLE
XTOGLE5   D5B     D6B    TOGLE
BDAC2     VOUTB   0      V =    128-(V(D6B)*64+V(D5B)*32+V(D4B)*16+V(D3B)*8+V(D2B)*4 +V(D1B)*2 +V(D0B))

XPOS_E2   trig2   PEOUT  POS_E2
XS_H1     Vtime   trig   VtimeS      SH

.control
*TRAN     TSTEP  TSTOP  TSTART TMAX   ?UIC?
tran      2u    100m      0      2u
set       pensize = 2

*=========Plot_reference_AADC=============================
plot     vin  vouta  voutb/50  xlimit  1m  100m
plot     vin  vouta            xlimit  1m  100m

*=========Create_Arrays============================
let  numb = length(vtimes)-101
unlet index
unlet val
unlet index
let index = vector(62)
let val   = vector(62)*0 +.5
let tref  = vector(62)*0 +.1
let dt    = vector(62)*0


*=========Use_Trigger_To_Store_Value_and_Time============================
let i = 100
repeat  $&numb
let tt = peout[i]

if (tt > .1)
let k = voutb[i]
let index[k] = k
let value = vin[i]
let val[k] = value
let tref[k] = vtimes[i]
end

let i = i+1
end

*=========Find_delta_Time============================
let tref[0] = 0
let dt    = vector(62)*0

let m = 1
repeat 60
let dt[m] = tref[m] -tref[m-1]
let m = m+1
end
plot val dt*50  vs index

*=========Curvefit_Arrays============================
unlet  x
unlet  indext
unlet  indexn
unlet  dx
unlet  dy
let x      = vector(630)*0 +.1
let indext = vector(630)
let indexn = vector(630)*0
let y      = vector(630)*0 +.5
let z      = vector(630)*0 +.5
let dx     = vector(630)*0 +1m
let w      = vector(630)*0
*=========Curvefit_Points============================
let i = 0
let j = 1
let n = 0
*=========There_are_about_60_Points============================
repeat  60
let S0scale = ( tref[j+1] -tref[j]  )/( tref[j] -tref[j-1]  )
let S2scale = ( tref[j+1] -tref[j]  )/( tref[j+2] -tref[j+1])
```

```
let S0  = ( val[j]-val[j-1]    )*S0scale
let S1  = ( val[j+1]-val[j   ] )
let S2  = ( val[j+2]-val[j+1] )*S2scale
let S01 = (S1+S0)
let S12 = (S2+S1)
let HH  = val[j+1]-val[j]
let A   = val[j]
let B   = S01
let C   =  3*HH -2*S01 -S12
let D   = -2*HH   +S01 +S12
*=========Will be adding 10 extrapolation_Points==============================
repeat  10
let indext[i] = i
let x[i] = tref[j] +  n*(tref[j+1]-tref[j])/10
let tt = n/10
let indexn[i] = n/10
let y[i] =  A  +B*tt  +C*tt^2 +D*tt^3
let z[i] =  .5*sin( 360*4297.1*x[i]^3 ) +.5
let w[i] = val[j]
let n = n + 1
let i = i + 1
end
let n = 0
let j = j + 1
end

plot z y w   vs x

.endc

.MODEL    SW      SW(    VT=.5 VH=.1  RON=1 ROFF=100MEG)
.MODEL    DD2      D( IS=3.15e-18 RS=1  CJ0=2f)

*=========POS_Edge2============================================
*
*   ___   |\             VBF ____               ___
* |IN |__|3\_______________|    \   ___    ___  |   \   ___
* |___|   | /  |          VLP|  PE \_|OUT|  |IN |_|PE  \_|OUT|
*         |/  |/\  /\  /\____|     / |___|  |___| |    / |___|
*                \/  \/   |  |___/                |__/
*               RLP      _|_
*                        ___
*                   CLP   |
*                        _|_
*                        ///
*
.SUBCKT   POS_E2  IN    OUT
 BBUF     VBF    0      V =   2*u( v(IN )-.5 )
 RLP      VBF    VLP    10k
 CLP      VLP    0      .1n   IC=0
 BAND     OUT    0      V =   u( u(v(VBF )-.5)*u(.5 -v(VLP ) ) -.1)
.ENDS     POS_E2

*=========Sample_Hold============================================
*
*            IN2
*     ___  |\    /         IN3 |\      ___
*   |IN |_|1\__/   ___________|2\____|OUT|
*   |___|  |  / S1   _|_       |  /   |___|
*          |/        ___       |/
*                 C1  |
*                     |
*                    _|_
*                    ///
.SUBCKT   SH     IN     CNTL      OUT
 B1       IN2    0      V =       v(IN )
 S1       IN2    IN3    CNTL     0    SW
 C1       IN3    0      .1u

 R1       IN3    0      100Meg
 B2       OUT    0      V =       v(IN3 )
.ENDS     SH

*=========POS_Edge============================================
*
*   ___   |\             VBF ____               ___
* |IN |__|3\_______________|    \   ___    ___  |   \   ___
* |___|   | /  |          VLP|  PE \_|OUT|  |IN |_|PE  \_|OUT|
*         |/  |/\  /\  /\____|     / |___|  |___| |    / |___|
*                \/  \/   |  |___/                |__/
*               RLP      _|_
*                        ___
*                   CLP   |
*                        _|_
*                        ///
*
.SUBCKT   POS_E  IN     OUT
 BBUF     VBF    0      V =   u( v(IN )-.5 )
 RLP      VBF    VLP    10k
 CLP      VLP    0      .1n   IC=0
 BAND     OUT    0      V =    u( u(v(VBF )-.5)*u(.5 -v(VLP ) ) -.1)
.ENDS     POS_E

*=========TOGGLE============================================
*        __________________________________________
*       |                                          |
*       |   /    VC1 |\  _NVC1 /    VC2   |\       |  ___
*       |__/  _______| \/ \___/  _________|  \___|_|OUT|
*          S1   _|_  | /\_/  S2   _|_    |  /      |___|
*               ___  |/               ___  |/
*            C1  |                  C2  |
*                |                      |
*               _|_                    _|_
*               ///                    ///                 |\
```




```
.SUBCKT   TOGLE   IN     OUT2
 BBUF     VBF     0      V =    u( v(IN )-.5 )
 RLP      VBF     VLP    1k
 CLP      VLP     0      5n
 BAND     VPE     0      V =    u( u(v(VBF )-.5)*u(.5 -v(VLP ) ) -.5)
 RLP2     VPE     VLP2   .3k
 CLP2     VLP2    0      2n
 BBRK     BRK     0      V =     1-u(v(VLP2 )-.2)
 BMAK     MAK     0      V =     u(v(VLP2 )-.9)
 S1       OUT     VC1    MAK     0        SW
 S2       NVC1    VC2    BRK     0        SW
 C1       VC1     0      10n
 C2       VC2     0      10n
 R1       VC1     0      10Meg
 R2       VC2     0      10Meg
 BINV     NVC1    0      V =     1-u(v(VC1)-.2)
 BOUT     OUT     0      V =     u(v(VC2)-.2)
 BOUT2    OUT2    0      V =     1- u(v(VC2)-.2)
.ENDS     TOGLE

.end
```

8.18.11_3.31PM
dsauersanjose@aol.com
Don Sauer