*=============**Infinite_Response**=========================

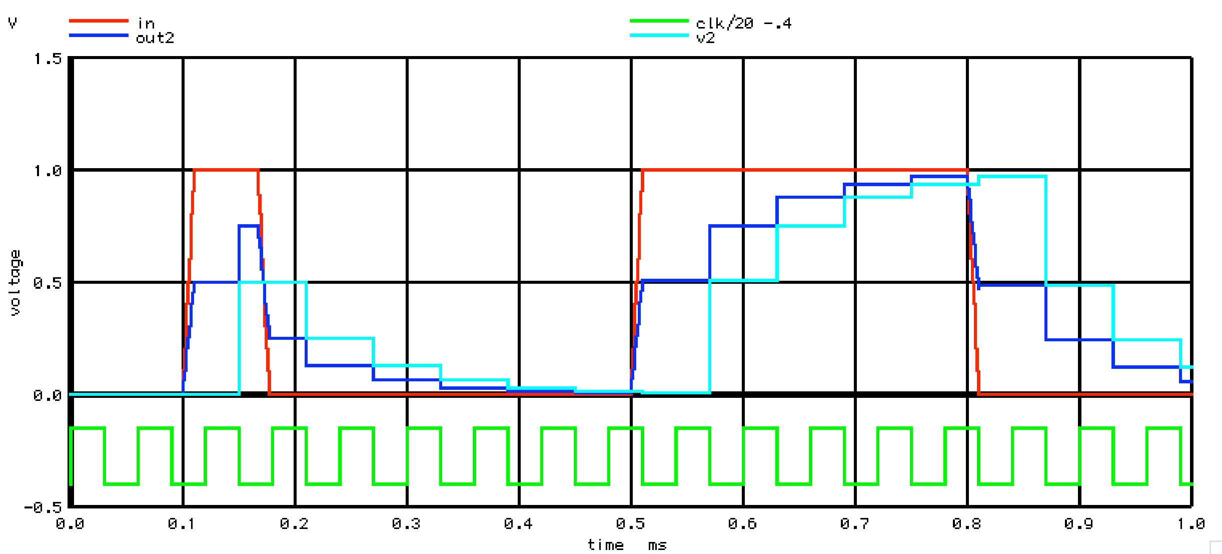
A Infinite Impulse Response (IIR) Digital filter involves adding feedback to delay elements. Doing a trial and error calculation to converge on an answer is the same thing.

===

```
Infinite_Response
*
*
*            |CLK|
*            |___|                    /|  BINV
*              |                 /  |
*    |V2 |__|ADELAY|_|V1 |______/  |________
*    |___|  |______| |___|      \  |       |
*      |   XDELAY1               \ |       |
*      |                          \|       |
*      |     R1  1K     VINN               |
*      |____/\  /\  /\__     R4   500       |
*             \/  \/   |                   |
*          R2          |__/\  /\  /\_      |
*    IN_____/\  /\  /\__|    \/  \/  |   |
*      |      \/  \/   |              |   |
*     _|_   R3   1K    |   |\         |   |
*    /   \             |___|-\        |  _|_
*   //  \  \               |  \_______|_|OUT|
*   \    \//               |  /         |___|
*    \___/              ___|+/
*     _|_              _|_ |/ BOPA
*     ///              ///
*
```

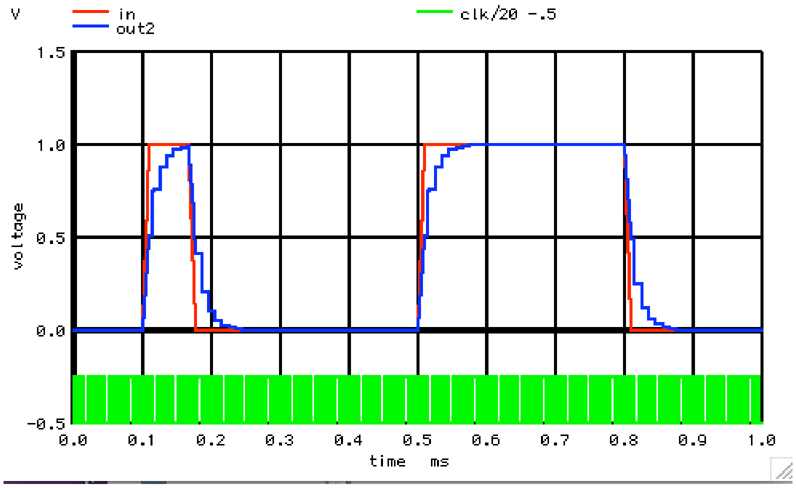
===

In this example, half of a delayed output signal is being negatively fed back across an Op Amp. Ultimately that negative feedback cancels out the input signal at the Op Amp's input. But is does so by reducing the error by one half for each clock cycle. In theory, this one half error reduction process goes on forever, kind of like the half life of uranium.



===

So the output square wave response resembles a simple RC lowpass filter. Notice how the difference between input and output starts off at one half. The first clock reduces it to one quarter, then on to one eight, etc.. The error is seeing a half life with each clock.



===

When one speeds up the clock, the same half life reduction of error is going on, only faster.

In Analog filters, this delay element is often a Op Amp and capacitor hooked up as an integrator.

======================FILTERS_STATE_VARIABLE===============================

```
                                            ___||_          ____||_|Y(S)|
                                            |      |        |      ||____|
                                            | |\  C|        | |\ C |
                         Rk       R         | -\   |   R    | -\   |
                      |U(S)|_/\  /\__/\  /\_|_|  \_|_/\  /\_|_|  \_|
                      |____|    \/  |    \/   |  /|     \/    |  /|
                                    |     _____|+/|      ____|+/ |
                                    |   _|_   | / |    _|_   | /  |
                                    |   ///       |    ///        |
                              Rsum  |             |               |
                                    |         Ra2 |               |
                            _/\  /\_|_______/\  /\|               |
                          _|_  \/    |        \/         Ra1      |
                          ///        |____________________/\  /\__|
                                     K  =Rsum/(Rsum+Rk)     \/
                                     a2 =Rsum/(Rsum+Ra2)
                                     a1 =Rsum/(Rsum+Ra1)
                                     s=1 when Xc = R
                                      ___    ..         .
                      U(S)    ___    / _ \  y(t) _____   y(t) _____   y(t)
                          _\|  k  |_\/ \   \_____| 1/s |_____| 1/s |_______\ Y(S)
                           /|_____|_/\ /_  /     |_____|    ||_____|  |    /
                                     \___/                  |         |
                                      ^ ^                   |         |
                                      |  |___|-a2|__________|         |
                                      |      |_____|   ____           |
                                      |______________| -a1 |__________|
                                                     |_____|

                      Y(S)/U(S)  =K/( s^2 +a2*s +a1)
```

==

The negative feedback defines the poles of the filter.
The feedforward defines the zeros of the filter.

======================FILTERS_STATE_VARIABLE_ANALOG==========================

```
Filters               high_pass =  (s^2                )/(s^2 +(wo/Q)*s +wo^2)
                      band_pass =  (    +(wo/Q)*s      )/(s^2 +(wo/Q)*s +wo^2)
                      low_pass  =  (               wo^2)/(s^2 +(wo/Q)*s +wo^2)
                      all_pass  =  (s^2 -(wo/Q)*s +wo^2)/(s^2 +(wo/Q)*s +wo^2)
                      band_rej  =  (s^2            +wo^2)/(s^2 +(wo/Q)*s +wo^2)
```

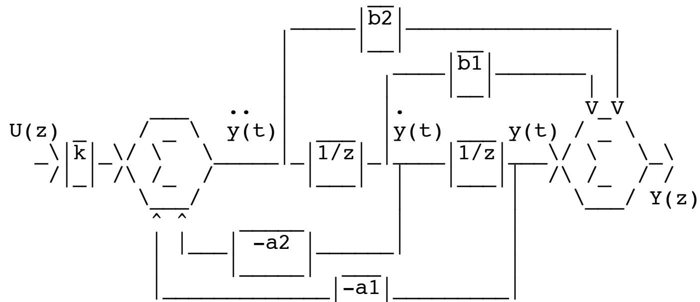
```
                                                  __
                                            _____|b2|________________
                                           |     |__|                |
                                           |       |____|b1|_______  |
                                           |       |    |__|       | |
                                        .. |        .             V_V
                      U(S) _    ___   y(t) |        y(t) ___ y(t) / _ \
                        _\|k|_\/ \   \______|_|1/s|_|____|1/s|___\/ \   _\
                         /|_|_/\ /_  /       |___|  |   |___||   /\ /_  / /
                                  \___/              |        |    \___/ Y(S)
                                  ^ ^                |        |
                                  |  |___|-a2|_______|        |
                                  |      |_____|              |
                                  |_____________|-a1|__________|
                                                |___|
                      Y(S)/U(S)  =K (1+ s*b1 + s*b2) /( s^2 +a2*s +a1)
```

==

Using the State variable format above makes it easy to make any type of filter.

Digital filters are just the same filter where the integrators all get replaced with analog delay elements.

=====================FILTERS_STATE_VARIABLE_ANALOG==========================




```
Y(z)/U(z)  =K (1+ z*b1 + z*b2) /( z^2 +a2*z +a1)
```

==

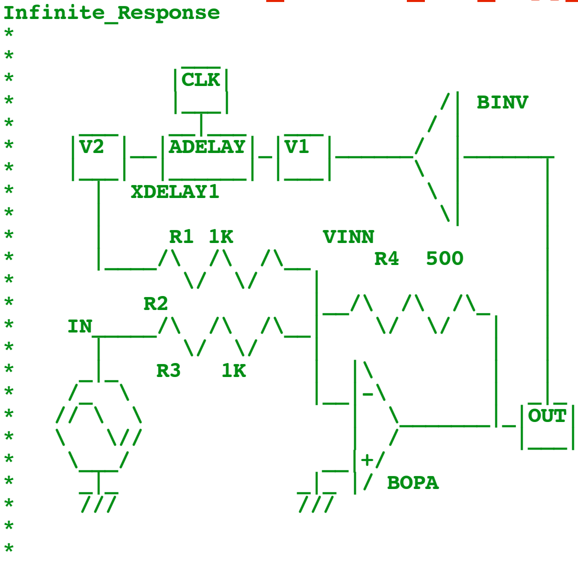
The digital filter is effectively just the same filter which uses delay signal (z terms) to replace integrators (s term). The discreet world is not exactly the same as the analog world. To convert transfer functions from H(s) to H(z), use the following.

$$s = \frac{2}{T}\frac{z-1}{z+1}$$ from Laplace to z (Tustin transformation);

$$z = \frac{2+sT}{2-sT}$$ from z to Laplace.

===========**Full_Netlist_For_Copy_Paste**=======================

```
Infinite_Response
```




```
*V_PULSE# NODE_P NODE_N DC     VALUE  PULSE( VINIT  VPULSE TDELAY TRISE  TFALL  PWIDTH PERIOD )
VCLK      CLK    0      DC     0      PULSE( 0      5      1n     1n     1n     30u    60u    )
*V_PWL#   NODE_P NODE_N DC     VALUE  PWL(   T1     V1     T2     V2     T3     V3     ...>)
VIN       IN     0      DC     0      PWL( 0 0 .1m 0 .11m 1 .167m 1 .177m 0 .5m 0 .51m 1 .8m 1 .81m 0 )
XDELAY1   V1     V2     CLK    ADelay
BINV      V1     0      V=     -V(out)
BOPA      OUT    0      V=     5*tanh( V(VINN)*1000)
R1        V2     VINN   1K
R2        IN     VINN   1K
R4        VINN   OUT    500

.MODEL    SWP    SW(    VT=2.6 VH=.2  RON=1  ROFF=10MEG)
.control
*TRAN     TSTEP  TSTOP  TSTART TMAX
 tran     1u     1m     0      1u
 set      pensize = 2
 let      out2   = -out
 plot     in     clk/20 -.5    out2   ylimit -.5 1.5
.endc
```

```
*=========Analog_Delay=========================
.SUBCKT   ADelay IN     OUT    CLK
 R0       CLK    VTD    100k
 C0       VTD    0      1p
 BRE      VRE    0      V = 5*u(V(CLK) -V(VTD)-.1)
 BFE      VFE    0      V = 5*u(V(VTD) -V(CLK)-.1)
*SXXXXXX  N+     N-     NC+    NC-    MODEL
 S1       V1     V2     VRE    0      SWP
 S2       V3     V4     VFE    0      SWP
 C1       V2     0      30n
 C2       V4     0      30n
 BIN      V1     0      V =    V(IN)
 BMID     V3     0      V =    V(V2)
 BOUT     OUT    0      V =    V(V4)
.ENDS     ADelay
*  ___
* |CLK|_/\  /\  /\_ VTD   VRE   VFE
* |___|   \/  \/  _|_    _|_   _|_
*              C0 ___   /BRE\  /BFE\
*       R0         |    \___/  \___/
*                 _|_    _|_    _|_
*                 ///    ///    ///
*
*  ___   |\V1 S1 V2 |\V3 S2 V4 |\   ___
* |IN |_| \_/ _____| \_/ _____| \_|OUT|          ___
* |___| | /  : _|_  | /  : _|_  | / |___|         |CLK|
*       |/  VRE___  |/  VFE___  |/                |___|
*           C1 _|_         _|_            ___    __|___   ___
*      BIN     /// BMID C2/// BOUT       |IN |__|ADELAY|_|OUT|
*                                        |___|  |______| |___|
*
*===============================================

.end
```

4.11.10_4.54PM
dsauersanjose@aol.com
Don Sauer
http://www.idea2ic.com/