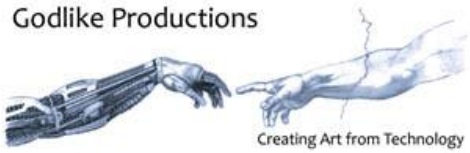
Author: Jason Fewings BSc (Hons), PgDip

Date: 30 April 2023

https://godlike.com.au



WHITE PAPER

Capacitor Modelling on Eventide H9000

MODELLING WITH PICODELAY

The Capacitor Modelled DADSR Envelope is a sample accurate model of a capacitor charging and discharging. The output is a unipolar envelope that ranges from 0 to 1, and which should have timings accurate regardless of sampling rate.

The VSIG file's are available free of charge at https://godlike.com.au/index.php?id=420 (current as of 2023). The specific address may change in the future if we update the website backend, but it will remain available under H9000 resources on the website.

A capacitor can be modelled with the following differential equation:

$$\frac{dV}{dt} = kV \quad Equation\ 1$$

V = Voltage
t = Time

where

$$k = \frac{-1}{RC} \quad Equation\ 2$$

R = Resistance
C = Capacitance

let

$$\tau = RC \quad Equation\ 3$$

τ is known as the Time Constant. At 3 x τ, the capacitor is assumed to be 95% charged or discharged (Tutorials, 2022), and we have used this as the envelope time in the model. The voltage, charge and current of the capacitor approach the applied voltage, charge or current, asymptotically.

We can rearrange equation 1 and substitute 2 and 3 in to derive

$$V = -\tau\frac{dV}{dt} \qquad \textit{Equation 4}$$

We will model the voltage changes to the capacitor as disturbances with respect to time, d(t), which gives us

$$V = -\tau\frac{dV}{dt} + d(t) \quad \textit{Equation 5}$$

Defining s as the time between samples, we can convert this to a difference equation as such

$$V_n - V_{n-1} = \frac{s(V_{n-1} - d(t))}{-\tau} \qquad \textit{Equation 6}$$

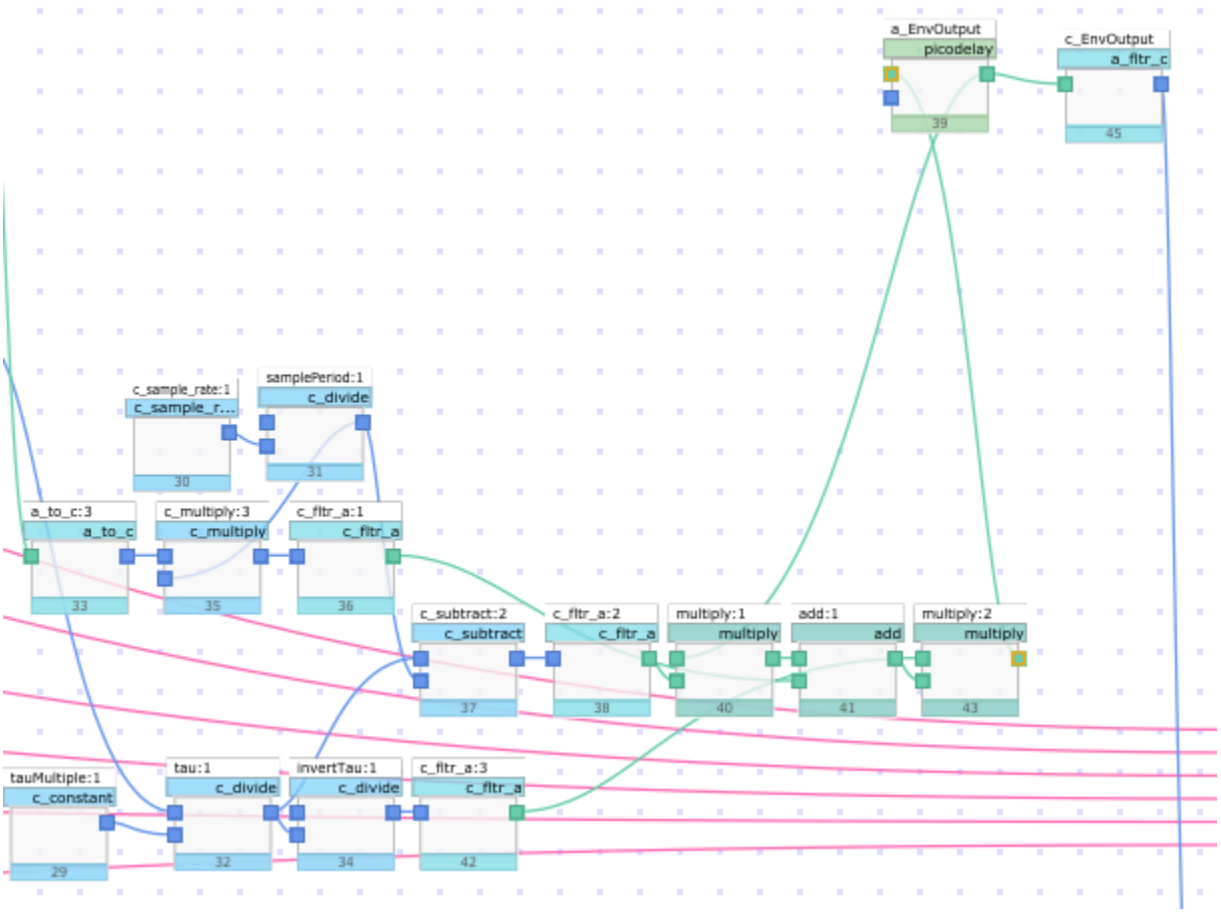
This simplifies to

$$V_n = \frac{sd(t) + V_{n-1}(\tau - s)}{\tau} \qquad \textit{Equation 7}$$

We can implement this directly in VSIG in discrete time. (V_{n-1}) is a B (Backstep) operator, or the output value at the last sampling interval.

This forms the capacitor modelling aspect.

This is represented in the VSIG model by these blocks.



The Backstep operator or shift register is implemented by the 1 sample picodelay. The outputs 39 and 43 both give the envelope shape, 1 sample apart. Pick whichever one you want to feed the parameter you want mapped to the envelope.

The 2 variables become d(t), which is the forcing function with respect to time, and τ, which is the time constant of the capacitor.

We model d(t) directly in VSIG as a series of steps triggered when we press down on the trigger button. This fires a 1 into a delay. After the delay period the output rises to 1, and we start the attack timer. At the conclusion of this, we set the step level to the sustain level. This is maintained until we release the trigger at which time d(t) falls to 0.

I have implemented some additional logic so that the release stage starts from whenever the button is released, and if the button is released prior to the attack stage starting, the envelope remains untriggered.

OK, this is the target levels taken care of. This emerges at block 28 of the VSIG model.

At each of these breakpoints we need to update τ in line with the rate of the relevant stage. The calculations that end at block 25 determine the times. We do this by running c_hold blocks to hold the level at 1 for each envelope stage and then multiplying that by the relevant segment time, and then summing them. The result of this is that the time of the current segment is the only segment not multiplied by 0, and this then feeds the time constant calculation of the capacitor model.

We only require the time constant of the model, not individual resistor or capacitor values. These are always linked. A large capacitor will require a smaller resistor to charge with constant current in a certain time. R x C will always be constant for given charge or discharge time, so we simplify things by directly setting τ, a luxury that is not afforded in electrical design.

There is another level of optimization that can be achieved on this design by removing the audio rate delay blocks 9 and 14 and instead basing them off blocks 19 and 23 instead, however blocks 26, 27 and 28 will require different logic to the current logic. I will address this in the next revision of this algorithm.

To use this algorithm in your own design, please ensure that the code comments block (44) remains. We would also appreciate a link to our website (https://godike.com.au) or an acknowledgement in any credits or readme files. The menu page (block 7) has all of your envelope controls (though you can disconnect the hmonitor), and audio and control rate envelopes are available at blocks 39 and 45 respectively.

CAPACITOR MODELLING USING THE INTEGRATOR BLOCK

An alternative and theoretically a more efficient way to model a capacitor is to use the integrator block.

From first principles if we have a supply voltage (which we can set as a forcing function) with a capacitor in series of capacitance C driving a resistive load R we can use the following equations.

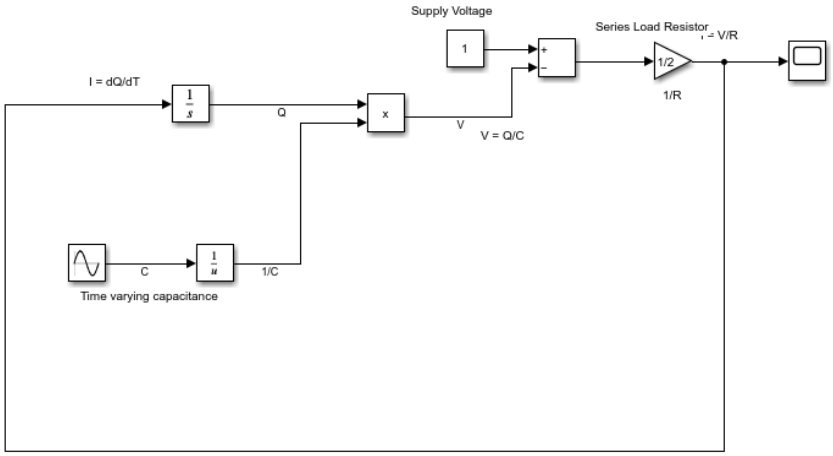
$$I = \frac{V}{R} \quad \textit{Equation 8}$$

$$I = \frac{dQ}{dt} \quad \textit{Equation 9}$$

$$Q = \int I\,dt + n \quad \textit{Equation 10}$$

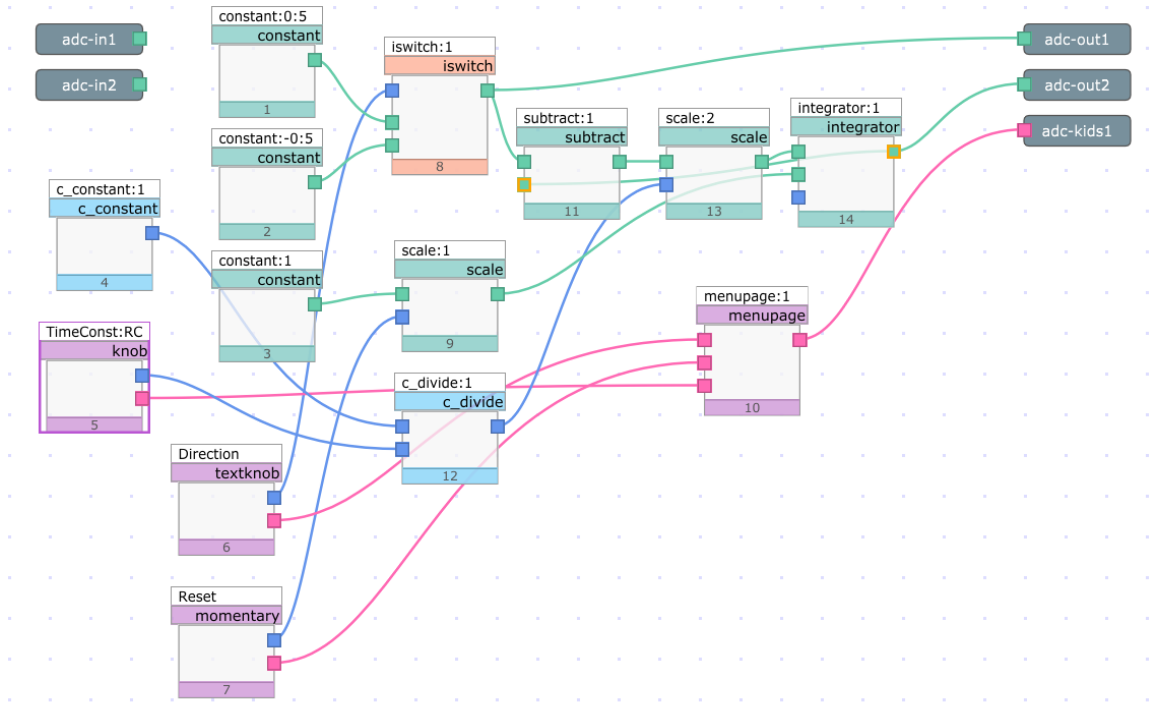
$$V = \frac{Q}{C} \quad \textit{Equation 11}$$

The capacitor can be modelled in Simulink, in the following manner (Jon, 2022)



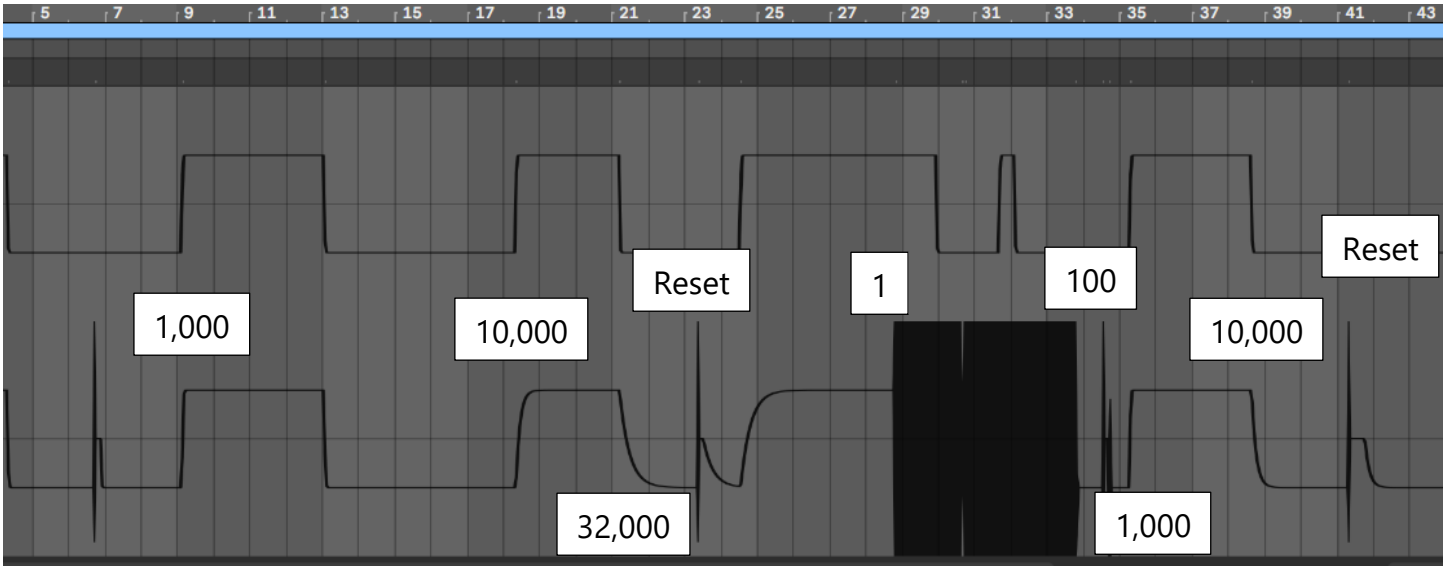
In our case, we will set the capacitance and load resistance as a variable, collect them by simplifying the block diagram to create a time constant (RC) and vary the voltage as the signal with a switch (0.5 and -0.5).

Implementing this in VSIG we can model it as per the diagram below. It is worthy to note that this is the same as a PI controller if you are familiar with industrial automation or process control. The switch can be though of adjusting the setpoint and the time constant is the integrator time constant, combined with the gain.



A signal (forcing function) can be substituted in place of blocks 8 and its predecessors (for implementation in an application). In the example above provides a step function that varies between -0.5 and +0.5. The top line below is the input, and the bottom line below is the output. The RC at the start is 1000, rising to 10,000, 32,000 and then 1, 100, 1000 and 10,000. At bar 23 and 41 the reset is pressed resetting the integrator to 0.

At an RC of 1, the output is unstable, essentially the gain is too high for stability (gain is tied up with RC)



If you find any errors or have suggestions for improvements, we are keen to hear from you, and we are also keen so see what you build with this. Feel free to hit us up at https://godlike.com.au/index.php?id=contact

REFERENCES

Jon. (2022, July 29). *Modelling a variable (i.e. nonlinear) capacitor in Simulink: How to make it stable?* Retrieved from Mathworks: https://au.mathworks.com/matlabcentral/answers/1763665-modelling-a-variable-i-e-nonlinear-capacitor-in-simulink-how-to-make-it-stable

Tutorials, E. (2022). *RC Discharging Circuit*. Retrieved from https://www.electronics-tutorials.ws/rc/rc_2.html