

# Discretization of an RC Lowpass Filter

Ross Bencina  
<http://www.rossbencina.com>  
[rossb@audiomulch.com](mailto:rossb@audiomulch.com)

January 9, 2015\*

## 1 Introduction

This note derives a digital version of an analog RC lowpass filter<sup>1</sup> using nodal analysis and trapezoidal integration. It follows the method used by Andrew Simper in his recent technical notes.<sup>2</sup> The aim is to lay out the mathematical steps that underpin the method, more or less from first principles. It's a first step towards modelling analog circuits using numerical methods.

Some knowledge of calculus and a little familiarity with linear circuits is assumed (e.g. application of Ohm's Law and Kirchoff's Laws to form nodal equations). Familiarity with digital IIR filters is also assumed (i.e. the idea of a digital filter with state variables and a state update procedure that computes an output sample from an input sample at each time step).

## 2 The circuit

The analog RC lowpass circuit is shown in Figure 1.  $R$  is the resistance of the resistor.  $C$  is the capacitance of the capacitor. Our goal is to create a discrete time digital implementation of this circuit. We want a program that takes voltage  $V_{in}$  at each time step and gives us  $V_C$ .

## 3 Voltages across the components

The process of deriving the digital implementation involves solving equations for the circuit using Ohm's Law and Kirchoff's laws. This process is called nodal

---

\*An earlier version appeared as a Google document in July 2014. The current version is archived at <https://github.com/RossBencina/dsp-notes> and may be revised from time to time. Corrections and bug reports are welcome.

<sup>1</sup>[https://en.wikipedia.org/wiki/RC\\_circuit#Series\\_circuit](https://en.wikipedia.org/wiki/RC_circuit#Series_circuit)

<sup>2</sup>See Simper's "Direct Numerical Integration of a One Pole Linear Low Pass Filter" and other technical papers located at <http://www.cytomic.com/technical-papers>.

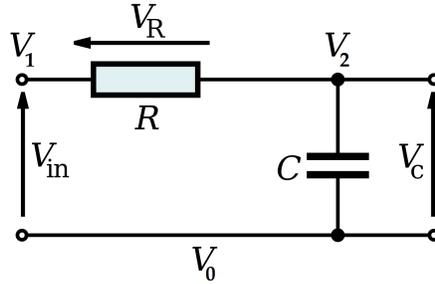


Figure 1: Analog RC lowpass circuit. Image source: Wikipedia

analysis.<sup>3</sup>

We apply the node voltage method<sup>4</sup> to produce an equation for the voltage across the resistor ( $V_R$ ). The voltage across the capacitor  $V_C$  is an unknown that we will solve for later.

Assume that electron current flows in the direction of the arrows.<sup>5</sup> Applying the formula ( $V_{highest} - V_{lowest}$ ) for the voltage across a component yields  $V_R$  as follows:

$$V_1 = V_0 + V_{in} \quad (1)$$

$$V_2 = V_0 + V_C \quad (2)$$

$$V_R = V_1 - V_2 \quad (3)$$

$$\implies V_R = (V_0 + V_{in}) - (V_0 + V_C) \quad (4)$$

$$\implies V_R = V_{in} - V_C \quad (5)$$

## 4 Current flow for each component

We need to know the (effective<sup>6</sup>) current flow through each component.

Recall Ohm's Law for a general resistor:

$$I = \frac{V}{R} \quad (6)$$

Using the equation for  $V_R$  from above (5) gives us the current through the resistor:

<sup>3</sup>[http://en.wikipedia.org/wiki/Nodal\\_analysis](http://en.wikipedia.org/wiki/Nodal_analysis)

<sup>4</sup>[http://en.wikipedia.org/wiki/Nodal\\_analysis](http://en.wikipedia.org/wiki/Nodal_analysis)

<sup>5</sup>For our method to succeed, the direction of the arrows doesn't matter so long as they are kept consistent for the whole calculation. See for example this discussion at physicsforums.com.

<sup>6</sup>Depending on your point of view, current may not flow "through" capacitors. In our model we're going to assume current flows through capacitors and call it  $I_C$ .

$$I_R = \frac{1}{R}V_R \quad (7)$$

$$\implies I_R = \frac{1}{R}(V_{in} - V_C) \quad (8)$$

The equivalent law for capacitors is:<sup>7</sup>

$$\boxed{I = C \frac{dV}{dt}} \quad (9)$$

So the current in and out of the capacitor is:

$$I_C = C \frac{d}{dt}V_C \quad (10)$$

## 5 A differential equation for the circuit

Kirchoff's current law (KCL)<sup>8</sup> tells us that the sum of currents into a node equals the sum of currents out of the node. For node  $V_2$  this yields

$$I_R = I_C \quad (11)$$

Substituting the expressions above for  $I_R$  (8) and  $I_C$  (10):

$$\frac{1}{R}(V_{in} - V_C) = C \frac{d}{dt}V_C \quad (12)$$

Our goal is to produce a numerical solution to equation 12 for  $V_C$  (the circuit output) given  $V_{in}$  (the circuit input) at each time step. To do this we discretize (approximate) the capacitor using a linear approximation and then expand the KCL equations using the discretized expression for  $I_C$ .

## 6 Aside: current-voltage relation for a capacitor

This section is a bit of an aside. We're going to compute a result that's needed in the next section. It relates capacitor voltage and current. If you're happy to take the relation as given you can skip this section. In summary, we will show that equation 10:

$$I_C = C \frac{d}{dt}V_C$$

implies:

$$V_C(t) = \frac{1}{C} \int_{t_0}^t I_C(\tau) d\tau + V_C(t_0)$$

<sup>7</sup>[https://en.wikipedia.org/wiki/Capacitor#Current.E2.80.93voltage\\_relation](https://en.wikipedia.org/wiki/Capacitor#Current.E2.80.93voltage_relation)

<sup>8</sup>[https://en.wikipedia.org/wiki/Kirchhoff's\\_circuit\\_laws](https://en.wikipedia.org/wiki/Kirchhoff's_circuit_laws)

This is a well known relation that expresses voltage across a capacitor at time  $t$  as the sum of the voltage at an earlier time  $t_0$ , plus an integration of  $I_C$  (current) over the time range  $t_0$  to  $t$ . In the next section we'll use the relation to construct a discrete time time-step update equation for the voltage across a capacitor.

To derive the current voltage relation, recall equation 10 for current flow in and out of a capacitor:

$$I_C = C \frac{d}{dt} V_C, \quad (\text{equation 10})$$

rearranging and making time an explicit parameter  $\tau$  :

$$\frac{d}{dt} V_C(\tau) = \frac{1}{C} I_C(\tau)$$

Integrating both sides from  $t_0$  to  $t$ :<sup>9</sup>

$$\begin{aligned} \int_{t_0}^t \frac{d}{dt} V_C(\tau) d\tau &= \frac{1}{C} \int_{t_0}^t I_C(\tau) d\tau \\ [V_C(\tau)]_{t_0}^t &= \frac{1}{C} \int_{t_0}^t I_C(\tau) d\tau \\ V_C(t) - V_C(t_0) &= \frac{1}{C} \int_{t_0}^t I_C(\tau) d\tau \\ V_C(t) &= V_C(t_0) + \frac{1}{C} \int_{t_0}^t I_C(\tau) d\tau \end{aligned} \quad (13)$$

Notice that given two times  $t_0$  and  $t$ , equation 13 gives us a way to compute  $V_C(t)$  as a function of  $V_C(t_0)$  and a definite integral over  $I_C$  from time  $t_0$  to time  $t$ .

## 7 Discretizing a capacitor using the trapezoidal rule

In the previous section we derived a general time-domain relation (equation 13) for the capacitor currents and voltages at times  $t_0$  and  $t$ . We'll use the general relation to derive a time-step update rule (difference equations) for our uniformly sampled (digital) system. The update rule takes us from the previous time step to the current time step.

Let  $sr$  be the sampling rate in samples per second.

Let  $T$  be the sampling period:  $T = 1/sr$

Let  $t$  be the time of the current time step

---

<sup>9</sup> See: [http://www.site.uottawa.ca/mathasatool/01unit/12topic/focus/voltage\\_current/p09.htm](http://www.site.uottawa.ca/mathasatool/01unit/12topic/focus/voltage_current/p09.htm)

Take  $t_0$  to be the time of the previous time step, that is  $t_0 = t - T$

Discretizing the current/voltage relation for time  $t$ , first substitute  $t$  and  $T$  into equation 13 from the previous section:

$$V_C(t) = V_C(t_0) + \frac{1}{C} \int_{t_0}^t I_C(\tau) d\tau \quad (\text{from equation 13}) \quad (14)$$

$$= V_C(t - T) + \frac{1}{C} \int_{t_0}^t I_C(\tau) d\tau \quad (t_0 = t - T) \quad (15)$$

Now we apply the trapezoidal rule to discretize the integral. The trapezoidal rule states:<sup>10</sup>

$$\int_a^b f(x) dx \approx (b - a) \frac{(f(a) + f(b))}{2} \quad (16)$$

We will use the trapezoidal rule to approximate the following integral:

$$\frac{1}{C} \int_{t_0}^t I_C(\tau) d\tau \approx ? \quad (17)$$

Making the following substitutions to the trapezoidal rule (16):

$$\begin{aligned} f(x) &: I_C(x) \\ a &: (t - T), \\ b &: (t), \\ (b - a) &= t - (t - T) = t - t + T \\ \implies (b - a) &: T \end{aligned}$$

yeilds:

$$\frac{1}{C} \int_{t_0}^t I_C(\tau) d\tau \approx \frac{T}{2C} (I_C(t - T) + I_C(t)) \quad (18)$$

Substituting this approximation into equation 15 gives an expression for  $V_C(t)$  in terms of voltages and currents at the current and previous time steps:

$$V_C(t) \approx V_C(t - T) + \frac{T}{2C} (I_C(t - T) + I_C(t)) \quad (19)$$

The trapezoidal rule is an approximation, but we will treat equation 19 as an equivalent expression for  $V_C$  from here on.

<sup>10</sup> Trapezoidal rule: [https://en.wikipedia.org/wiki/Trapezoidal\\_rule](https://en.wikipedia.org/wiki/Trapezoidal_rule) [http://www.mathwords.com/t/trapezoid\\_rule.htm](http://www.mathwords.com/t/trapezoid_rule.htm)

Let's switch our notation to the discrete time domain and to something more closely resembling computer code. We'll notate multiplication with ".". Take  $n \in \mathbb{Z}$  as the current time step and  $n - 1$  as the previous time step. Switch notation as follows:

$$\begin{aligned} V_C(t) &\rightarrow vc[n] \\ V_C(t - T) &\rightarrow vc[n - 1] \\ I_C(t) &\rightarrow ic[n] \\ I_C(t - T) &\rightarrow ic[n - 1] \end{aligned}$$

Rewriting the trapezoidally integrated equation for  $V_C$  (19):

$$vc[n] = vc[n - 1] + T/(2C).(ic[n - 1] + ic[n]) \quad (20)$$

Later, when applying KCL we will want an expression for  $ic[n]$ . So, solve equation 20 for  $ic[n]$ :

$$\begin{aligned} \implies vc[n] - vc[n - 1] &= T/(2C).ic[n - 1] + T/(2C).ic[n] \\ \implies vc[n] - vc[n - 1] - T/(2C).ic[n - 1] &= T/(2C).ic[n] \\ \implies ic[n] &= (2C)/T.(vc[n] - vc[n - 1]) - ic[n - 1] \end{aligned} \quad (21)$$

Letting  $gc = (2C)/T$ :

$$ic[n] = gc.vc[n] + (-gc.vc[n - 1] - ic[n - 1]) \quad (22)$$

Following a convention used by QUCS using "iceq":<sup>11</sup>

Let

$$\begin{aligned} iceq[n - 1] &= -gc.vc[n - 1] - ic[n - 1] \\ ic[n] &= gc.vc[n] + iceq[n - 1]. \end{aligned} \quad (23)$$

Or equivalently:

$$\begin{aligned} ic[n] &= gc.vc[n] + iceq[n - 1] \\ iceq[n] &= -gc.vc[n] - ic[n]. \end{aligned} \quad (24)$$

We can further simplify  $iceq$  by substituting for  $ic[n]$

$$\begin{aligned} iceq[n] &= -gc.vc[n] - ic[n] \\ \implies iceq[n] &= -gc.vc[n] - (gc.vc[n] + iceq[n - 1]) \\ \implies iceq[n] &= -2.gc.vc[n] - iceq[n - 1]. \end{aligned} \quad (25)$$

<sup>11</sup> See e.g. equations 6.60 and 6.63 in the QUCS documentation

## 8 Solving the KCL equation(s) using the linear equivalent capacitor current

Remember the differential equation from section 5? We're ready to solve the equivalent linear equation. Instead of  $I_C = C \frac{d}{dt} V_C$  we'll be using the linear approximation  $ic[n]$  derived in the previous section (24).

Recall,

$$\begin{aligned} I_R &= (1/R)(V_{in} - V_C) && \text{(current through the resistor)} \\ \rightarrow ir[n] &= (1/R).(vin[n] - vc[n]) \end{aligned}$$

$$\begin{aligned} gc &= (2C)/T && (26) \\ iceq[n] &= -2.gc.vc[n] - iceq[n - 1] \\ ic[n] &= gc.vc[n] + iceq[n - 1] && \text{(current "through" the capacitor)} \end{aligned}$$

Equating current in and current out at our node of interest:

$$\begin{aligned} I_R &= I_C && \text{(KCL: currents in and out of a node are equal)} \\ \rightarrow ir[n] &= ic[n] \\ \Rightarrow (1/R).(vin[n] - vc[n]) &= gc.vc[n] + iceq[n - 1] && (27) \end{aligned}$$

*(In general we would have a set of simultaneous equations to solve, one equation for each node. However, in this case we have only one node.)*

Now the final step. Given our input  $vin[n]$  we want to compute  $vc[n]$ .

Solving the previous equation for  $vc[n]$ :

$$\begin{aligned} (1/R).(vin[n] - vc[n]) &= gc.vc[n] + iceq[n - 1] \\ \Rightarrow (1/R).vin[n] - (1/R).vc[n] - gc.vc[n] - iceq[n - 1] &= 0 \\ \Rightarrow (1/R).vin[n] - iceq[n - 1] &= (1/R).vc[n] + gc.vc[n] \\ \Rightarrow vin[n] - R.iceq[n - 1] &= vc[n] + R.gc.vc[n] && \text{(times R)} \\ &= vc[n].(1 + R.gc) \\ \Rightarrow vc[n] &= (vin[n] - R.iceq[n - 1]) / (1 + R.gc) && (28) \end{aligned}$$

So our final equations are:

$$gc = (2C)/T \quad (29)$$

$$vc[n] = (vin[n] - R.iceq[n - 1]) / (1 + R.gc) \quad (30)$$

$$iceq[n] = (-2.gc.vc[n] - iceq[n - 1]) \quad (31)$$

or, letting  $gr = 1/R$

$$vc[n] = (gr.vin[n] - iceq[n - 1]) / (gr + gc) \quad (32)$$

$$iceq[n] = (-2.gc.vc[n] - iceq[n - 1]) \quad (33)$$

Or equivalently:

$$vc[n] = (gr.vin[n] + iceq[n - 1]) / (gr + gc) \quad (34)$$

$$iceq[n] = 2.gc * vc[n] - iceq[n - 1] \quad (35)$$

Andrew Simper suggests setting  $gc = 1$ ,  $gr = g$

$$vc[n] = (g.vin[n] + iceq[n - 1]) / (1 + g) \quad (36)$$

$$iceq[n] = 2.vc[n] - iceq[n - 1] \quad (37)$$

In pseudo-code:

```
init :
    iceq = 0

step (input: vin, output: vc):
    vc = (g * vin + iceq) / (1 + g)
    iceq = 2 * vc - iceq
```

## 9 Acknowledgements

The method described here is an interpretation of Andrew Simper's worked example,<sup>12</sup> and his related papers.<sup>13</sup> Thanks to Andrew and other members of the #musicdsp IRC channel for their very kind assistance in guiding me through this.

---

<sup>12</sup><http://www.cytomic.com/files/dsp/OnePoleLinearLowPass.pdf>

<sup>13</sup><http://www.cytomic.com/technical-papers>