Small-signal modeling

In the previous chapter, we strived to describe and simulate our basic structures (buck, boost and buck-boost) in the time domain using switching elements. This is the *switched* model approach where we implement switching elements like MOSFETs and diodes: we are in the time domain where the variable *t* rules the circuit operation. Simulation conditions are thus fixed via a *.TRAN* statement, testifying for a transient analysis. In the time-domain, SPICE works like a sample and hold system featuring a variable step: it first fixes a step (or a time interval between two simulated points, the so-called timestep) and tries to converge toward the right solution, reached when the result (i.e. the final error) fits the simulation tolerance options. This internal variable is labeled *Tstep*. If it fails, the engine reduces *Tstep* and tries to converge again. If it does, the point is accepted and stored, otherwise the point is rejected. As a result, on quiet signals (e.g. *di/dt* or *dv/dt* close to zero), SPICE choses a large time step because almost no changes occur. To the opposite, as soon as a transition appears, the engine reduces the time interval to capture enough data-points. Figure 2-1 clearly shows this typical behavior:

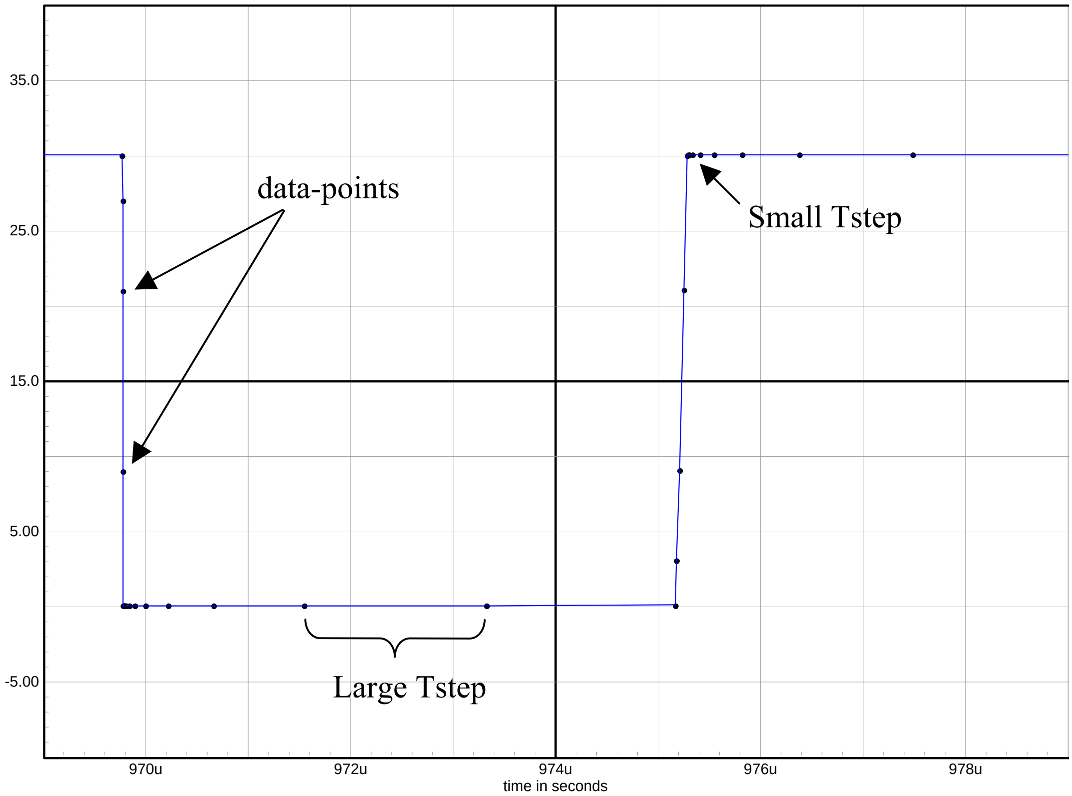

Figure 2-1: on slowly changing signals, SPICE takes large time intervals. As changes occur, SPICE reduces its time step…

When simulating small portions of working time, e.g. a few milliseconds, this technique delivers quick analysis results. Unfortunately, when you need to assess the transient response of a low-bandwidth system switching at 100 kHz, you might need to simulate tens – if not hundered – of millisecond before you obtain a usable result. With a variable timestep system like SPICE, it can easily take tens of minutes (sometimes hours) before you get a single waveform. Once the simulation has ended, it might even take a few minutes more before you can extract and display data buried into the 100 Mo file built on the disk by the simulator! A fixed timestep simulator like PSIM elegantly solves this problem and represents an excellent solution when this phenomenon hampers your simulation time. A typical example is the Power Factor Correction (PFC) circuit which switches at a high frequency but implements a 10 Hz bandwidth… We will come back on this later on.

Another solution, consists in deriving an *averaged* model. An averaged model implies the disappearance of any switching event (hence discontinuities) to the benefit of a smoothly variying, continuous, signal. To clarify the concept, let us explain what the term, averaging, means: averaging usually means that a period of a periodic time function *f(t)* is integrated during a cycle and divided by the duration of that cycle. This is the well known definition:

$$\left\langle f(t)\right\rangle_T = \frac{1}{T}\int_0^T f(t)dt \qquad \text{eq. 2-1}$$

where $\left\langle f(t)\right\rangle_T$ represents the average value of *f(t)* over a cycle of *T* duration.

Applying equation 2-1 to a time evolving signal would lead to a succession of separate discrete values, all representing the average value of *f(t)* calculated at consecutive periods: *0-T*, T-T_1, T_1-T_2... If you link all these points with a continuous function that has the same values as the averaged function at the end of each period and is essentially smooth, then you get an "averaged and continuous" representation of the original function. This is what your eye naturally does when you observe a program on TV: the moving image you can see is actually made of a succession of fixed frames but they are displayed at a much quicker pace your eye can sustain. You therefore observe a continuous flow of images…

Also, thanks to the averaging process, the ripple has disappeared. This is because we assume that the modulating period is much larger than the switching period: your eye response time is larger than the rate at which images are displayed (20 ms in Europe). This concept is described by figure 2-2 where you see how a sinusoidal waveform modulates the duty-cycle and imposes a current variation in the inductor. The duty-cycle is modulated via the following law:

$$d(t) = D_{dc} + D_{mod}\sin\omega_{mod}t \qquad \text{eq. 2-2}$$

D_{dc} represents the steady-state duty-cycle corresponding to a given operating point, whereas D_m describes the modulating peak value. Both D_{dc} and D_{mod} are constant with the condition $|D_{mod}| << D_{dc}$ implying that the system under study stays linear in the modulated region. The modulation period f_{mod} is also much larger than the converter switching frequency f_{sw}. The averaged and continuous function is similar to the filtered waveform, but is not exactly the same, since it is a mathematical abstraction rather than a real time-dependent physical variable: as figure 2-2 shows, the ripple has been neglected.

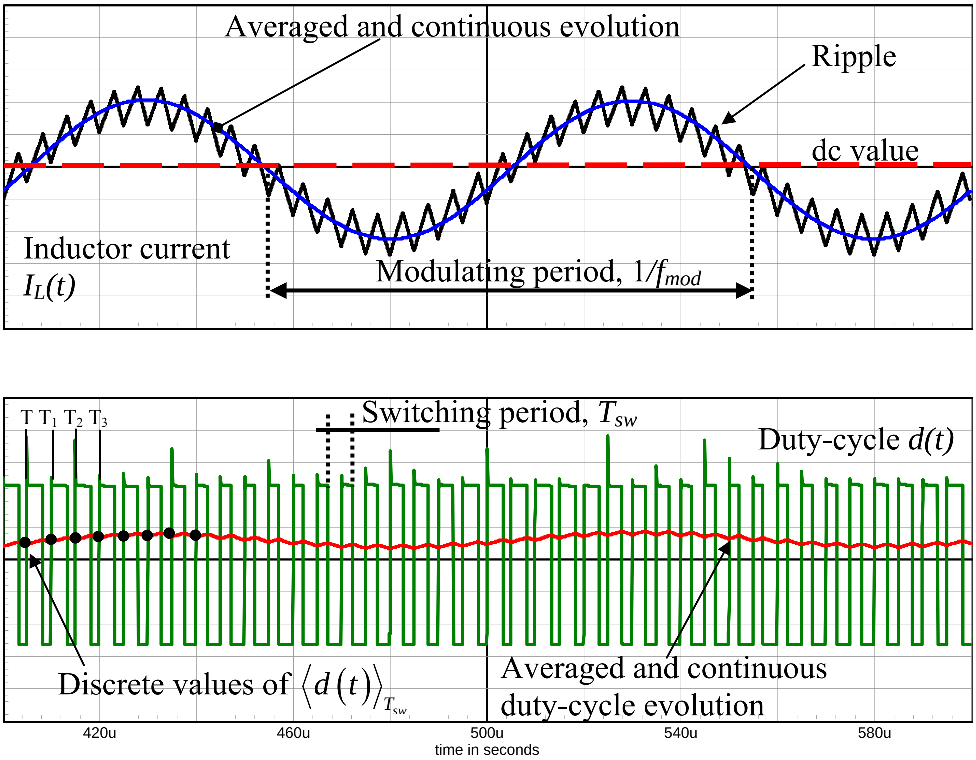

Figure 2-2: an ac modulated inductive current with its slowly varying low-frequency component. Please note the difference between the modulating frequency and the switching frequency: $f_{mod} << F_{sw}$.

Thanks to the above technique, we will have a way to describe the behavior of a particular structure (a buck for instance) via averaged equations. Once linearized across an operating point, these equations will lead to a so-called averaged small-signal model, usefull to unveil the ac response of the structure under study or check the stability once the loop is closed and compensated.

As you can imagine, both models, switched or averaged, have their PROs and CONs. Here they are listed for you:

Averaged models

- ♦ *Small-signal response*: draw Bode or Nyquist plots in a snap-shot and assess the stability of your converter, get the input impedance variations and check if the filter is compatible without damping.
- ♦ *No switching component*: simulation results are immediate! You can visualize long transient effects of several tens milliseconds (e.g. in a low bandwidth system like a Power Factor Corrector) and see if the output overshoot is within the specifications or not or if the distorsion is close to the initial specification.

- ❖ It is difficult to see the effects of parasitic elements, like inductor ohmic losses or diodes forward drops, although some good models now include them.
- ❖ You cannot evaluate the switching losses of semiconductors.

Transient models

- ♦ *They can include parasitic elements*: see the effect of the leakage inductance and quantify the main switch voltage stress or the poor resulting cross-regulation in multiple output converters.
- ♦ *Reflects reality*: if the right component models are used, you can build a virtual breadboard where ripple levels, conduction and switching losses and so on, can easily be assessed.

- ❖ Because of the numerous switching events, the simulation time can be very long.
- ❖ Difficult if not impossible, to obtain a Bode plot from a transient simulation.
- ❖ Transient response is difficult and long to assess in low bandwidth applications.

As you will discover in this chapter, several options exist to model or derive the small-signal equivalent circuit of a given topology: a) you either write the converter state equations corresponding to the two switch(es) positions and average them over a switching cycle b) you average the switches waveforms only and create an equivalent model out of it. The common denominator of those methods remains the final small-signal linearization needed to extract the equivalent model. The first method a) is described through the State-Space Averaging (SSA) technique while the other option uses the averaged switch modeling technique. Let's start to study the first one:

State-Space Averaging

The SSA technique applied to power converters was first described by Dr ĆUK in the mid-70's [1] then documented in another founding paper together with Dr Middlebrook in reference [2]. A lot of books discuss in details this modeling technique and we will not re-introduce it again in this chapter [3, 4]. However, to give you a small taste of the SSA technique, we have represented a simple buck converter and highlighted its *state* variables (figure 2-3):

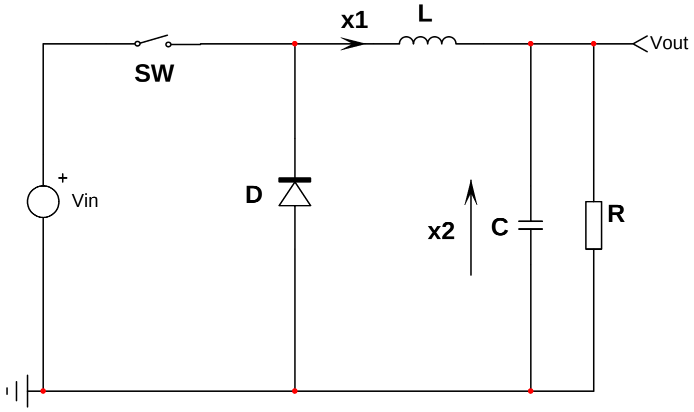

Figure 2-3: highlighted state variables for a buck converter

State variables are usually associated with storage elements like capacitors and inductors. If we know the state of the circuit at a given time, e.g. at $t = 0$, the so-called intitial conditions, then we should be able to solve the system equations for other $t > t_0$. For a circuit including a capacitor and an inductor, the state variables will respectively be the capacitor voltage, noted x_2 and the inductor current, noted x_1.

The state equations of a system S can be put under the compact form:

$$\dot{x} = \mathbf{A}x(t) + \mathbf{B}u(t)$$ general form of the state equation eq. 2-3

$$y(t) = \mathbf{M}x(t) + \mathbf{N}u(t)$$ general form of the output equation eq. 2-4

where:

$x(t)$ represents the state vector and contains all states variables of the circuit under study. In a LC circuit, it would contain our variables x_1 and x_2 discussed above. The input vector $u(t)$, contains the system independent inputs, for instance the input voltage V_{in}.

For a second order system as our buck circuit, we have:

$$\dot{x}_1 = a_{11}x_1 + a_{12}x_2 + b_{11}u_1 + b_{12}u_2$$ eq. 2-3a

$$\dot{x}_2 = a_{21}x_1 + a_{22}x_2 + b_{21}u_1 + b_{22}u_2$$ eq. 2-3b

Using the general matrix notation according to equation 2-3a and 2-3b, we obtain:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$ eq. 2-3c

$y(t)$ is a called the output vector and offers a way to place external waveforms that can be expressed as linear combinations of the state and input vectors. For instance, a current generator placed on the output of a network to derive its output impedance could be part of the output vector. An other output vector could be the input current of the same network if the designer needs to derive the input impedance…

$\dot{x} = \dfrac{dx}{dt}$, the time derivative of the state variable x_1.

A = state equation state coefficient matrix
u = source vector coefficient matrix
B = state equation source coefficient matrix
M = output equation state coefficient matrix
N = output equation source coefficient matrix

As equations 2-3 and 2-4 describe our system S in the time domain t and provided that they are linear, we can derive the Laplace transfer function of our system. Thanks to the Laplace property of derivatives, we can write:

$$sX(s) = \mathbf{A}X(s) + \mathbf{B}U(s)$$ eq. 2-5

$$Y(s) = \mathbf{M}X(s) + \mathbf{N}U(s)$$ eq. 2-6

Re-arranging equation 2-5 gives:

$$sX(s) - \mathbf{A}X(s) = \mathbf{B}U(s)$$ eq. 2-7

$$X(s)\left(sI - \mathbf{A}\right) = \mathbf{B}U(s)$$ eq. 2-8

and finally:

$$X(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}U(s) \qquad \text{eq. 2-9}$$

where $(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ represents $T(s)$, the general transfer equation linking state and source variables. **I** is the unity matrix.

Now plugging equation 2.9 into equation 2.6, we obtain:

$$Y(s) = \mathbf{M}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}U(s) + \mathbf{N}U(s) \qquad \text{eq. 2-10}$$

Factoring *U(s)* gives:

$$Y(s) = \left[\mathbf{M}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{N}\right]U(s) = T(s)U(s) \qquad \text{eq. 2-11}$$

Hence the generalized transfer function of a N_{th} order linear system:

$$T(s) = \left[\mathbf{M}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{N}\right] \qquad \text{eq. 2-12}$$

When further manipulating these equations, one can show that:

- det $(s\mathbf{I} - \mathbf{A})^{-1}$ represents the characteristic polynomial
- det $(s\mathbf{I} - \mathbf{A})^{-1} = 0$ is the characteristic equation whose roots are the transfer function poles.
- The Routh – Hurwitz criterion implies that stability is met if all coefficients of the characteristic polynomial are positive. This technique shows to be extremely useful, for instance, when you try to know if a linear system is stable without deriving its complete transfer function: write the state equation matrix A, evaluate det $(s\mathbf{I} - \mathbf{A})^{-1}$ and look at the coefficients…

The first sequence of the SSA thus consists in identifying matrixes coefficient corresponding to all the converter states. In CCM, we can identifiy two states (power switches on or off), during which all state variables exist within state durations. In DCM, however, a third state exists as both switches are now open. In this third state, the inductor state variable x_1 is zero since it has disappeared at the end of the diode conduction time. Each state will lead to a set of linear equations, formated via our matrix approach: $\mathbf{A_1}$ and $\mathbf{B_1}$ for the on state, $\mathbf{A_2}$ and $\mathbf{B_2}$ for the off state. We will later on see how to link them…

SSA at work for the buck converter – first step

To begin with, we will write all node and mesh equations pertinent to figure 2-3 in the two distinct switch positions, on and off. The converter operates in CCM and V_{in} is replaced by u_1 to better stick to the initial notation:

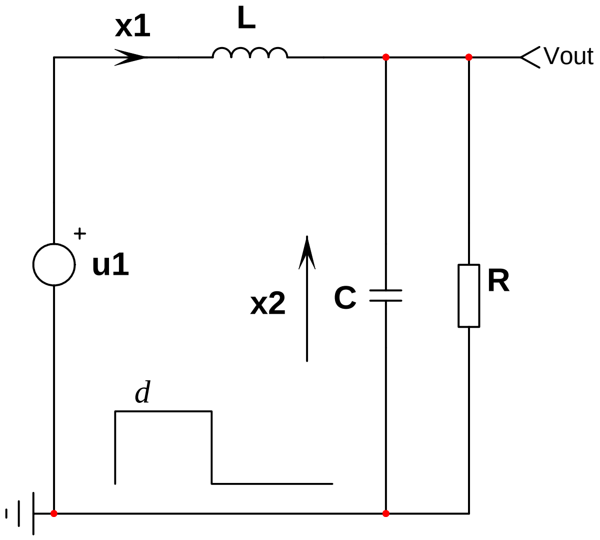

Figure 2-4a: the SW switch in the closed-state

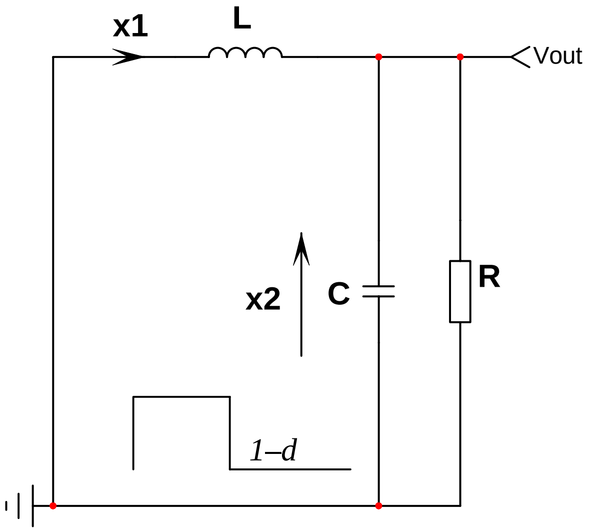

Figure 2-4b: the diode D is now conducting

State 1: SW is closed, the diode is open (figure 2-4a). Solve the equation with the state variables x_1 and x_2 in order to find their respective derivatives:

$$u_1 = L\dot{x}_1 + x_2 \qquad \text{eq. 2-13}$$

Once re-aranged:

$$\dot{x}_1 = -\frac{1}{L}x_2 + \frac{1}{L}u_1 \qquad \text{eq. 2-14}$$

$$C\dot{x}_2 = x_1 - \frac{1}{R}x_2 \qquad \text{eq. 2-15}$$

After manipulation:

$$\dot{x}_2 = \frac{1}{C}x_1 - \frac{1}{RC}x_2 \qquad \text{eq. 2-16}$$

Then, feed the $\mathbf{A_1}$ and $\mathbf{B_1}$ matrixes according to equation 2-3. Here, we do not use equation 2-4 as the needed information, V_{out}, is obtained via x_2.

$$\mathbf{A_1} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \qquad \mathbf{B_1} = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}$$

$\mathbf{B_1}$, as it is, cannot be multiplied by $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ to stick to equation 2-3c format. However, as equations 2-14 and 2-16 do not include u_2, we can add null b_{12} b_{22} coefficients in $\mathbf{B_1}$ to create an another column:

$$\mathbf{B_1} = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & 0 \end{bmatrix}$$

We can now combine equations 2-14 and 2-16 together, using $\mathbf{A_1}$ and the newly defined $\mathbf{B_1}$:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad \text{eq. 2-17}$$

State 2: SW is now open and the diode conducts (figure 2-4b). Let's solve the equation with the state variables x_1 and x_2 in order to find their respective derivatives :

$$0 = L\dot{x}_1 + x_2 \qquad \text{eq. 2-18}$$

Once arranged:

$$\dot{x}_1 = -\frac{1}{L}x_2 \qquad \text{eq. 2-19}$$

$$x_2 = R\left(x_1 - \dot{x}_2 C\right) \qquad \text{eq. 2-20}$$

After a different arrangement:

$$\dot{x}_2 = \frac{1}{C}x_1 - \frac{1}{RC}x_2 \qquad \text{eq. 2-21}$$

Then, feed the $\mathbf{A_2}$ and $\mathbf{B_2}$ matrixes according to equation 2-3 as done above:

$$\mathbf{A_2} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \qquad \mathbf{B_2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Gathering all terms, and creating a second column in $\mathbf{B_2}$ knowing that all u_1 and u_2 coefficient are zero:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad \text{eq. 2-22}$$

At this point, we need to link both states: $\mathbf{A_1}$ and $\mathbf{A_2}$, $\mathbf{B_1}$ and $\mathbf{B_2}$. If you look at the buck schematic and the equations we wrote, you certainly remark that A_1 and B_1 apply for the first (on) interval, or during d^{th} of the switching time, while $\mathbf{A_2}$ and $\mathbf{B_2}$ exist during the $(1-d)^{th}$ (off) switching time interval. Using this remark, we can combine both matrixes by the following equations:

$$\mathbf{A} = \mathbf{A_1}d + \mathbf{A_2}\left(1-d\right) \qquad \text{eq. 2-23}$$

$$\mathbf{B} = \mathbf{B_1}d + \mathbf{B_2}\left(1-d\right) \qquad \text{eq. 2-24}$$

Thanks to the multiplication by *d* and *(1-d)*, equations 2-23 and 2-24 smooth the discontinuity associated to the switching event and become continue. Updating equation 2-3 leads to:

$$\dot{x} = \left[\mathbf{A}_1 d + \mathbf{A}_2(1-d)\right]x(t) + \left[\mathbf{B}_1 d + \mathbf{B}_2(1-d)\right]u(t) \qquad \text{eq. 2-25}$$

Applying our previous equations to fit 2-22 and 2-23 expressions, leads to:

$$\mathbf{A} = \mathbf{A_1}d + \mathbf{A_2}\left(1-d\right) = \begin{bmatrix} 0 & -\frac{d}{L} \\ \frac{d}{C} & -\frac{d}{RC} \end{bmatrix} + \begin{bmatrix} 0 & -\frac{1-d}{L} \\ \frac{1-d}{C} & -\frac{1-d}{RC} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \qquad \text{eq. 2-26}$$

$$\mathbf{B} = \mathbf{B_1}d + \mathbf{B_2}\left(1-d\right) = \begin{bmatrix} \frac{d}{L} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{d}{L} & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{eq. 2-27}$$

We now have a set of *continuous* state equations which look like:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{d}{L} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad \text{eq. 2-28}$$

If we expand the equation set 2-28

$$\dot{x}_1 = -\frac{1}{L}x_2 + \frac{d}{L}u_1 \qquad \text{eq. 2-29}$$

$$\dot{x}_2 = \frac{1}{C}x_1 - \frac{1}{RC}x_2 \qquad \text{eq. 2-30}$$

and compare them to equations 2-14 and 2-16, we can see that the only difference lies in the presence of the d term affecting the input voltage, u_1. Because d will vary via the feedback loop to keep a selected parameter constant, for instance the output voltage, equation 2-29 and 2-30 are now non-linear. However, we can use them to build a non-linear, also called large-signal, model as depicted by figure 2-4c, knowing that u_1 is actually V_{in}:

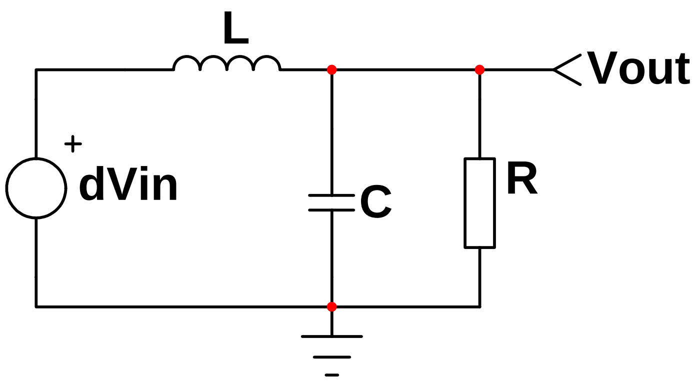

Figure 2-4c: a continuous non-linear model of the buck converter

The dc transformer

Nevertheless, from figure 2-4c, dV_{in} in reality is not the input source. V_{in} actually is. To improve figure 2-4c visibility, we will use a so-called dc transformer. This transformer does not have a physical meaning, as everyone knows that transformers only work in ac. But in our case, it will multiply V_{in} by the ratio d and will