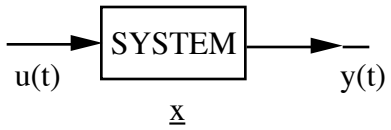
CHAPTER 2
INTRODUCTION TO CIRCUIT SIMULATION BY STATE EQUATION APPROACH

2.1 'State equation' form [1, 2]

$$\left.\begin{aligned} \dot{\underline{x}} &= f(\underline{x}, \underline{u}) \\ \underline{y} &= g(\underline{x}, \underline{u}) \end{aligned}\right\} \quad x, u, y \text{ are vectors} \tag{1}$$



$\underline{x}$ is a state variable

Fig. 2.1 System with state $\underline{x}$, input $\underline{u}$ and output $\underline{y}$

i) State Variables (SV) $\Rightarrow$ minimal set of variables giving the condition or 'state' of the system. The 'state' of the system uniquely defines how it will respond to a given input (i.e., defines the history). Also, all other quantities associated with the system can be expressed as linear combinations of the SVs and the inputs.

eg:

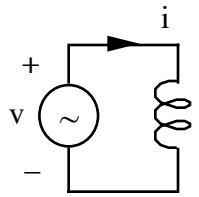


Fig. 2.2 Simple Inductive Circuit

$$\frac{di}{dt} = \frac{v}{L}, \qquad \begin{aligned} v &\rightarrow \text{ input} \\ i &\rightarrow \text{ state variable} \end{aligned}$$

ii) Note, the solution depends on state i and input v.

iii) State variables have to be assigned initial values. Note that for any given initial condition i (o), the future history is uniquely determined.

iv) Linear Form:

$$\left.\begin{aligned} \dot{\underline{x}} &= [A]\underline{x} + [B]\,\underline{u} \\ \underline{y} &= [C]\underline{x} + [D]\,\underline{u} \end{aligned}\right\} \tag{2}$$

S.V. Equation may be discretized, i.e. if we know x at time t, we may estimate x at time $t+\Delta t$, for a given u(t). There are many 'integration methods' to do this, for example, with rectangular integration, $\dot{x} = f(\underline{x}, \underline{u})$

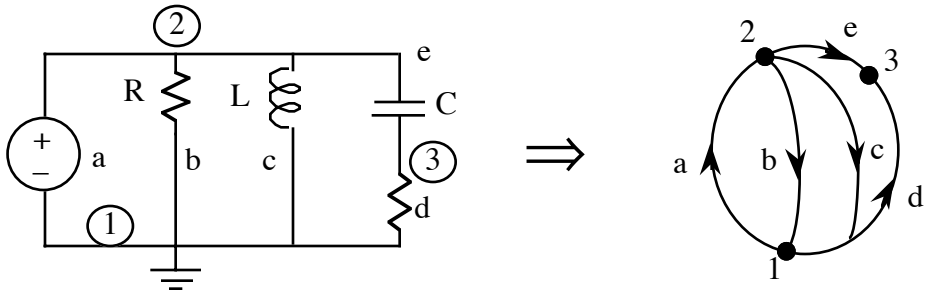
$$\left.\begin{aligned} &\frac{d\underline{x}}{dt} = f(\underline{x}, \underline{u}) \\ \text{or}\quad &\Delta x \approx f(\underline{x}(t), \underline{u}(t))\,\Delta t \\ \text{and}\quad &\underline{x}\,(t+\Delta t) \approx \underline{x}(t) + \Delta x \end{aligned}\right\} \qquad (3)$$

Other more complicated methods such as the Runge-Kutte Methods, or the predictor-corrector methods may also be used for SE (State Equation) integration. Methods like these are available in computer programs such as CSMP (Continuous System Modelling Program) by IBM. Programs such as EMTP or EMTDC also utilize such methods.

2.2 Obtaining SE's of the form (1) for circuits with RLC — elements [1]

We first need to define the following terms:

1) Graph of a circuit – is a directed interconnection diagram as shown in Fig. 2.3b.



(a) Network b) Graph

Fig. 2.3 Example of network and graph

The graph does not show the individual elements as RLC elements. The directions on the graph may be chosen arbitrarily.

2) A graph has nodes (i.e. 1,2,3 in Fig. 2.3b) and branches (a,b,c,d & e)

3) A tree is a minimal set of branches connecting all nodes.

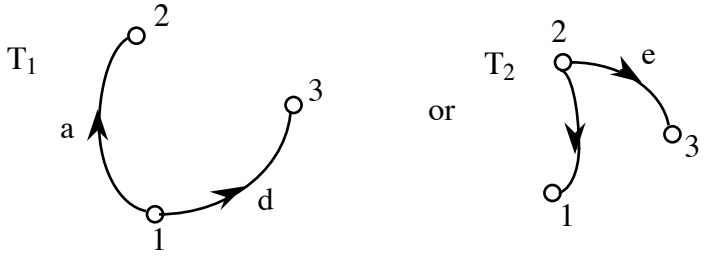


Fig. 2.4 Two possible trees for the above graph

are examples of trees. We may express this as $T_1 = \{a,d\}$ or $T_2 = \{b,e\}$ are 'trees'. Branches a and d are tree branches or twigs. A tree has no loops.

4) The 'Cotree' of a tree is the set of edges remaining when the tree is removed from the graph, e.g.

$$C_1 = G - T_1$$

or $$C_1 \cup T_1 = G$$

$C_1 = \{b,c,e\}$ is the cotree corresponding to tree T_1.

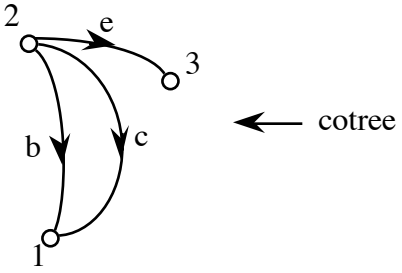


Fig. 2.5 Cotree of tree T_1

Cotree elements are 'links'!

5) With every cotree and only one co-tree branch, there are associated tree branches such that together they form a 'fundamental loop' shown below. The bold line is the cotree branch.

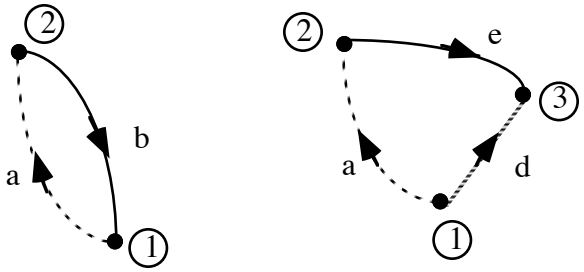


Fig. 2.6 Examples of Fundamental Loops

KVL equations may be written for loops, such as

$$v_a + v_b = 0$$

or $$v_a + v_e - v_d = 0$$

6) With every tree branch are associated cotree links which together with the tree branch form a fundamental cutset. e.g. (with tree branches shown in bold lines). Removing a cutset from a graph divides the graph into two disconnected sets of node: Thus cutsets satisfy KCL equations.

e.g.

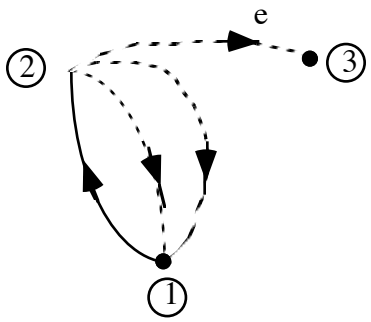


Fig. 2.7 Fundamental Cutsets

so that $i_a - i_b - i_c - i_e = 0$.

So far we have not considered the actual elements in a graph. We now do so.

7) A normal tree is formed by placing as many voltage sources and capacitors into tree edges as possible. (Unless an all capacitor or all voltage source loop exists, it is usually possible to put all voltages sources and capacitors into the normal tree). A normal tree should contain as few inductors and current sources as possible. The priority for membership in a normal tree is in the sequence E, C, R, L, J (E for voltage source J for current source).

8) A 'normal cotree' has priority of membership J, L, R, C, E.

9) Steps in writing SE's for a network
   i) Select a normal tree. Write fundamental loop equations. Introduce device characteristic to get differential and linear equations. Inductor currents are the 'state variables'.
   ii) Write cutset current equations, and again assign capacitor voltages as SV's.
   iii) There should be enough linear equations to express all other voltages and currents in terms of state variables. KVL for loops with capacitors and KCL for cutsets with inductors will yield the differential equations.

Examples:

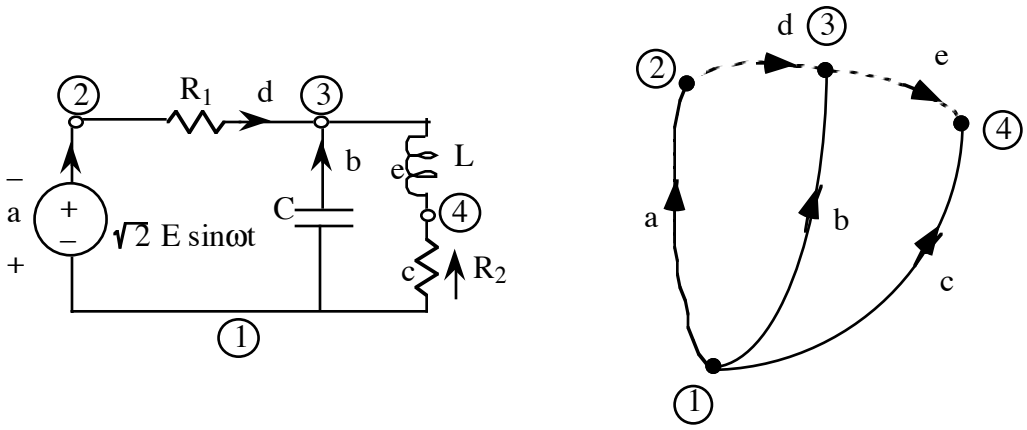


Fig. 2.8 A Network and its Graph

The normal tree has been shown in the graph in bold lines.

KVL equations (Fundamental loops)

$$v_a + v_d - v_b = 0 \tag{1}$$

$$v_e - v_c + v_b = 0 \tag{2}$$

KCL equations (Fundamental cutsets)

$$i_a - i_d = 0 \tag{3}$$

$$i_b + i_d - i_e = 0 \tag{4}$$

$$i_e + i_c = 0 \tag{5}$$

Those loop equations with an inductor in the loop yield the first set of SES.

Thus,

$$L\frac{di_e}{dt} - v_c + v_b = 0 \tag{2'}$$

Likewise cutset equations with capacitors yield another set of state equations.

$$C\frac{dv_b}{dt} + i_d - i_e = 0 \tag{4'}$$

We still have v_c and i_d in equations 2' and 4' which are not state variables.

But $v_c = -i_e R_2$

and $i_d = +\left(\frac{v_b - v_a}{R_1}\right) = \frac{v_b + \sqrt{2}\,E\sin\omega t}{R_1}$ (6)

Substituting 6 into 2' and 4', we get our state equations.

$$c\frac{dv_b}{dt} + \frac{v_b + \sqrt{2}E\sin\omega t}{R_1} - i_e = 0 \tag{2"}$$

$$L\frac{di_e}{dt} + i_e R_2 + v_b = 0 \tag{4"}$$

which can be rewritten as

$$\frac{dv_b}{dt} = -\frac{1}{R_{1C}}v_b + \frac{i_e}{c} - \frac{\sqrt{2}\,E\sin\omega t}{R_1 C}$$

and $$\frac{di_e}{dt} = -\frac{R_2}{L} i_e - \frac{v_b}{L}$$

or

$$\frac{d}{dt}\begin{bmatrix} v_b \\ i_e \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1 C} & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R_2}{L} \end{bmatrix} \begin{bmatrix} v_b \\ i_e \end{bmatrix} + \frac{1}{R_1 C} \begin{bmatrix} \sqrt{2}\ E \sin\omega t \\ 0 \end{bmatrix}$$

$$\dot{X} = A + B\ u \qquad (7)$$

which is in the classical form, and can be solved by proper numerical integration. All other voltages and currents may be expressed in terms of the sources and these state variables.

2.3 Presence of all-inductor cutset

Ex. 2:

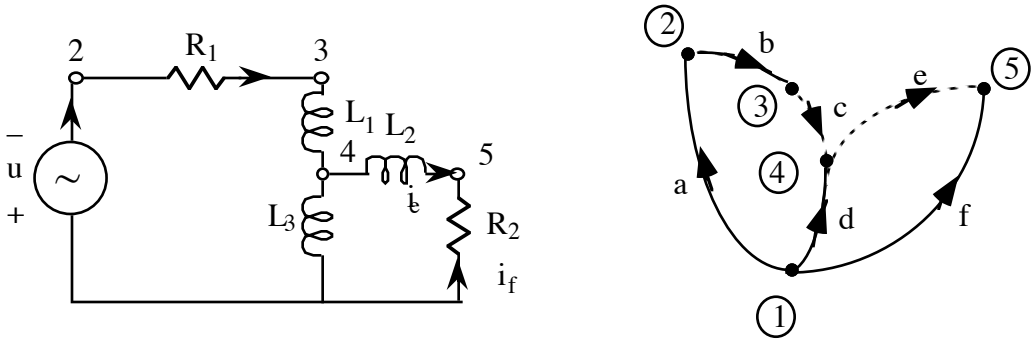


Fig. 2.9 Network with all inductiro cutset

Note, as there is an all inductor cutset, we must include one inductor in the normal tree

Fundamental loop equations (for cotree links)

$$u - i_b R_1 - L_1 \frac{d\, i_c}{dt} + L_3 \frac{d\, i_d}{dt} = 0$$

and (1)

$$-L_2 \frac{d\, i_e}{dt} + i_f R_2 - L_3 \frac{d\, i_d}{dt} = 0$$

We now choose state variables as

i_d and i_c then $i_e = i_c + i_d$

also fundamental cutset equations yield

$i_b = i_c$, $i_f = -i_e$ $(= -(i_c + i_d))$ (2)

Substituting equation (2) into (1)

$$u - i_c R_1 - L_1 \frac{d\, i_c}{dt} + L_3 \frac{d\, i_d}{dt} = 0$$

and (3)

$$-L_2 \frac{d}{dt}\{i_c+i_d\} - (i_c+i_d)R_2 - L_3 \frac{d\, i_d}{dt} = 0$$

Giving:

$$-L_1 \frac{d\, i_c}{dt} + L_3 \frac{d\, i_d}{dt} = i_c R_1 - u$$

$$-L_2 \frac{d\, i_c}{dt} - (L_2 + L_3) \frac{d\, i_d}{dt} = i_c R_2 + i_d R_2 \qquad (4)$$

or

$$\begin{bmatrix} -L_1 & L_3 \\ -L_2 & -(L_2+L_3) \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_c \\ i_d \end{bmatrix} = \begin{bmatrix} R_1 & 0 \\ R_2 & R_2 \end{bmatrix} \begin{bmatrix} i_c \\ i_d \end{bmatrix} - \begin{pmatrix} u \\ 0 \end{pmatrix} \qquad (5)$$

which is not in the classical form. This always happens when all-inductor cutsets or all-capacitor loops exist.

Eqn. 5 has the form

$$L\ \dot{x} = A'\ x + B'\ u$$

where L is the matrix shown. It can readily be converted to the classical form.

$$\dot{x} = L^{-1} A'\ x + L^{-1} B'\ u$$
$$= A\ \underline{x} + B\ \underline{u}$$

where $L^{-1}A' = A$, etc.

Again, such cases will always arise with all-inductor cutsets or all capacitor loops. This is so, because, say for an all capacitor loop with n capacitors,

$v_{C1} + v_{C2} + ... + v_{Cn} = 0$, and

thus all the capacitor voltages are not linearly independent.

2.5 Modelling thyristors and diodes:

There are several methods of modelling such devices. One way is to treat resulting circuits with the diode (or Thyristor) on and off as two different circuits, and solve them independently with matched boundary conditions, eg:

A

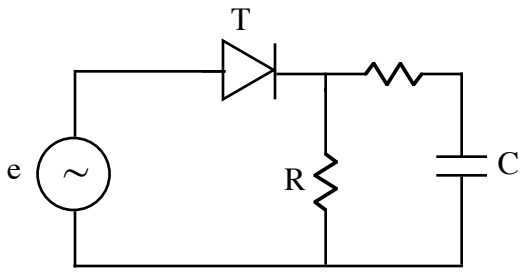


Fig. 2.10 Network with Switching Element

Thyristor on:

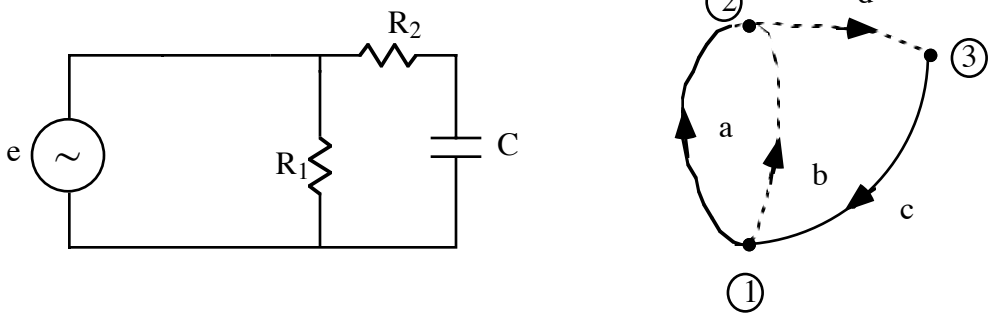


Fig. 2.11 Resultant Network with switch in on position (and accompanying graph)

From the fundamental cutset eqns.

$$c\frac{dV_c}{dt} = -i_d \left(= -\frac{v_d}{R_2}\right)$$

with

$v_d = e - v_c$ from fundamental loop eqns.

Giving the SE as

$$\frac{dV_c}{dt} = \frac{-V_c}{R_2C} + \frac{e}{R_2C} \qquad (1)$$

Thyristor off:

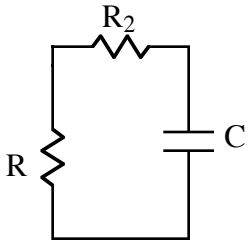


Fig. 2.12 Circuit with switch off

Likewise:

$$c\frac{dV_c}{dt} = \frac{-v_c}{(R_1 + R_2)C} \qquad (2)$$

Boundary Condition: v_c continuous

We may thus draw a state transition diagram [4]

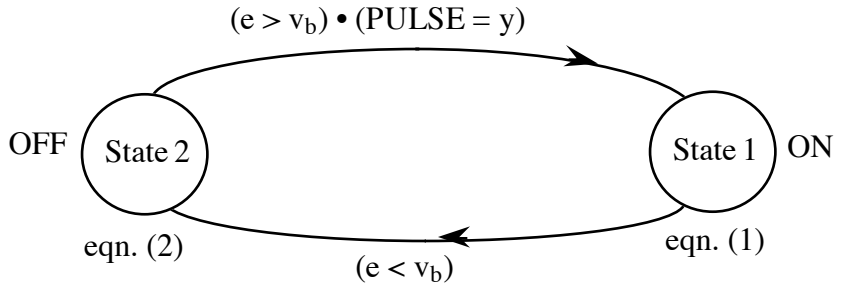


Fig. 2.13 State diagram for circuit in Fig. 2.10

The problem with this approach is that it yields many different states, and it is important to write SEs for each state, and also carefully implement the state transition logic. The advantage is a cleaner solution, that is, if the switches (diodes, thyristors, etc.), are off. They are treated as exact open circuits and if on, exact short circuits. The resulting systems of equations are usually smaller, and easier to solve.

B. Treating the diode resistance as a variable resistance

The circuit just discussed may be modelled as follows:

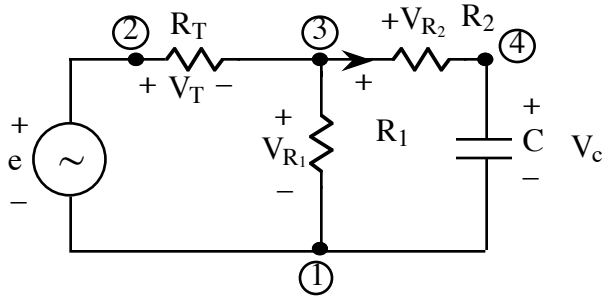


Fig. 2.12 Representation of the circuit in Fig. 2.10 with non-varying topology

which, after doing the usual manipulation yields the equation

$$\frac{d\,v_c}{dt} = \frac{-(R_1+R_T)}{R_1R_T + R_1R_2 + R_TR_2}\,v_c + \frac{eR_1}{R_1R_T + R_1R_2 + R_TR_2}$$

where $R_T = 1\Omega$ (say), if T is on

$= 1000\Omega$ (say), if T is off

The off $\leftrightarrow$ on transition is determined by the usual switching conditions.

Computer Tools:

1) Direct programming in Fortran for eqn. (3) of page 3.
2) Simulation programs like CSMP. Many standard block diagrams are already available

in the IBM CSMP program.

3) Simulation programs such as EMTDC of SPICE.

2.6 Integration of State Equations and Stability Implications [2]

The equation

$$\dot{X} = A\,X + B\,u \tag{1}$$

can be numerically integrated using various methods:

1) Rectangular integration (not usually recommended)

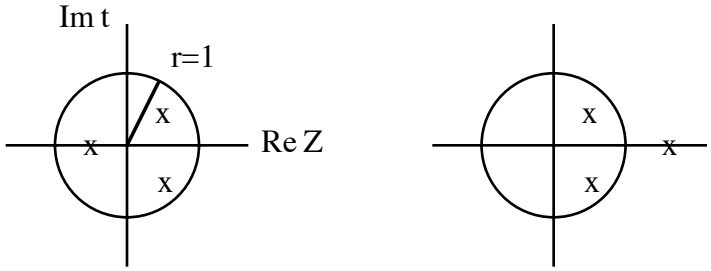
$$\underline{X}(t) = \int_{z=t-\Delta t}^{t} [A\,\underline{X}(z) + B\,\underline{u}(z)]\,dz + \underline{X}(t-\Delta t) \tag{2}$$

$$(I + A\Delta t)\,\underline{X}\,(t-\Delta t) + (B\Delta t)\,\underline{u}$$

Thus a difference equation of the form:

$$\underline{X}(t) = G\,\underline{X}(t-\Delta t) + H\,\underline{u} \tag{3}$$

results. For stability of numerical solution for bounded inputs [4] (BIBO Stability) a necessary and sufficient condition is that the eigenvalues of G lie within the unit circle as in Fig. 2.15.



a. Stable System b. Unstable System

Fig. 2.15 Pole positions for typical systems

2) Trapezoidal Integration

$$\dot{X} = A\,X + B\,u$$

$$\underline{X}(t) \approx \underline{X}(t-\Delta t) + \frac{A\underline{X}(t) + A\underline{X}(t-\Delta t)}{2} \bullet \Delta t$$

$$+ B\,\frac{u(t) + u(t-\Delta t)}{2} \bullet \Delta t$$

or

$$\left(I - \frac{A\Delta t}{2}\right)\underline{X}(t) = \left(I + \frac{A\Delta t}{2}\right)\underline{X}(t) + (B\Delta t)\,u^{*} \tag{4}$$

where $u^* = \frac{u(t) + u(t-\Delta t)}{2}$ and is known

or $$\underline{X}(t) = \left[I - \frac{A\Delta t}{2}\right]^{-1} \left[I + \frac{A\Delta t}{2}\right] \underline{X}(t) + \left[I - \frac{A\Delta t}{2}\right]^{-1} (B\Delta t)\, u^*(t) \qquad (5)$$

which is agin in the classical form

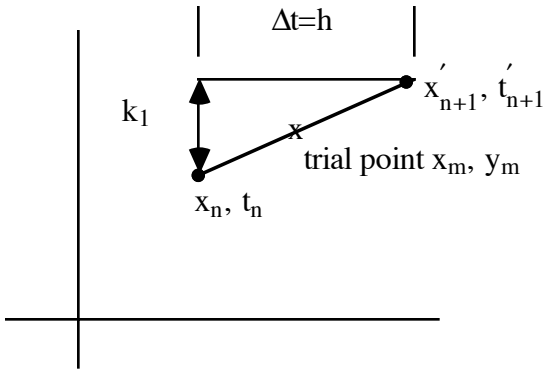
$$\underline{X}(t) = G\, \underline{X}(t-\Delta t) + H\, u(t)$$

Trapezoidal integration is stability preserving, i.e. if eigenvalues of A are in the left hand plane then eigenvalues of G are in the unit circle and conversely. The trapezoidal rule cannot readily be applied to non-linear systems, unless the system functions are invertible (Note, we were required to take an inverse of a matrix multiplication function).

In such cases we have to resort to implicit methods such as the Adam's second order formula or the Gear method [5].

Runge-Kutte Methods [6]

Consider the integration of an arbitrary function $\dot{\underline{X}} = \underline{f}(\underline{X}, t)$ (Note, since u is a function of t, we have eliminated it from our equation)



Let $x_n = x(t-\Delta t)$

$x_{n+1} = x(t)$

$\Delta t = h$

Fig. 2.16 Runge-Kutte of Order 2

estimate $\underline{k}_1 = \underline{f}(\underline{X}_n, t_n) \bullet \Delta t = hf(\underline{X}_n, t_n)$

estimate position of trial midpoint

$$\underline{X}_m = \underline{X}_n + \frac{\underline{k}_1}{2}, \quad t_m = t_n + \frac{h}{2}.$$

then $\underline{k}_2 = f(\underline{X}_m, t_m)$ is a value for the increment of $\underline{X}_n$, and is perhaps? a better value as it is evaluated at the midpoint. Then $\underline{X}_{n+1} \approx \underline{X}_n + \underline{k}_2$, $t_{n+1} = t_n + h$ is the next point on our solution. This is an example of a second order method.

The fourth order Runge-Kutte is often a popular method, where we make two trial estimates at the midpoint and one at the expected endpoint, i.e.,

$$\left.\begin{aligned} \underline{k}_1 &= h\,\underline{f}(\underline{X}_n, t_n) \\ \underline{k}_2 &= h\,\underline{f}\left(\underline{X}_n + \frac{\underline{k}_1}{2}, t_n + \frac{h}{2}\right) \\ \underline{k}_3 &= h\,\underline{f}\left(\underline{X}_n + \frac{\underline{k}_2}{2}, t_n + \frac{h}{2}\right) \\ \underline{k}_4 &= h\,\underline{f}\left(\underline{X}_n + \underline{k}_3, t_n + h\right) \end{aligned}\right\} \qquad (6)$$

and $\underline{X}_{n+1} = \underline{X}_n + \frac{\underline{k}_1}{6} + \frac{\underline{k}_2}{3} + \frac{\underline{k}_3}{3} + \frac{\underline{k}_4}{6} + O\left(h^5\right)$

is the esimate of the endpoint. Note that in these methods, it is a trivial matter to vary the timestep.

There are several other methods with various advantages and disadvantages listed in literature [5,6].

Ex.

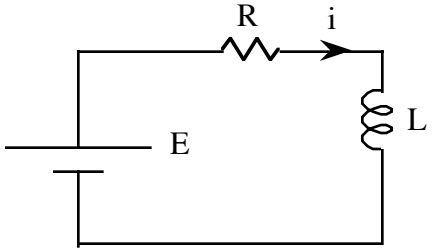


Fig. 2.16 An R-L circuit

$$\frac{di}{dt} = \frac{E}{L} - \frac{R}{L}\,i$$

by rectangular integration

$$i(t) = i(t-\Delta t) + \frac{E\Delta t}{L} - \frac{R}{L}\,i(t-\Delta t)\,\Delta t$$

$$= \left(1 - \frac{R}{L}\Delta t\right) i(t-\Delta t) + \frac{E\Delta t}{L}$$

For stable solution, we require

ev. of $\left(1 - \frac{R}{L}\Delta t\right)$ to be in the unit circle

i.e. $\left|1 - \frac{R}{L}\Delta t\right| < 1$

or $\left|1 - \frac{R}{L}\Delta t\right| = 1$ will give the boundary value on Δt. Thus $\Delta t = \frac{2L}{R}$ is the maximum timestep allowable. (Try this.)

With trapezoidal rule.

$$i(t) = \left[\frac{1 - \frac{R}{L}\Delta t}{1 + \frac{R}{L}\Delta t} \right] i(t-\Delta t) + \ldots$$

This time, as long as $\Delta t > 0$, the solution is always stable as is to be expected because we are applying a trapezoidal algorithm to a stable system, and we know that it preserves stability.

ASSIGNMENT #2:

1) Using the canonical approach to developing state equations, repeat the solution for the problem in assignment 1. Use the a) rectangular and b) trapezoidal altorithms for integration.

For a) and b) above calculate the values of the theoretically largest possible timestep for preserving a stable solution. Confirm by simulation.

References

[1] Balbanian, Bickert and Seshu, Network Analysis.

[2] Ogata, K., "Modern Control Engineering", Prentice Hall, 1970.

[3] R.J. Brown & W.F. Tinney, “Digital Solution for Large Power Networks”, Trans. AIEE, Vol. 76, Part 3, 1957.

[4] Katz, R.M., “Contemporary Logic Design”, Benjamin/Cumms Publishing co., 1994, ISBN 0-8053-2703-7.

[5] Robert W. Hornbeck, "Numerical Methods", Quantum Publishing, Prentice Hall, 1975.

[6] William H. Preis, et al., "Numerical Recipies, The Art of Scientific Computing", Cambridge University Press, ©1986.

Accuracy versus Stability

As seen in chapter 2, the stability of the algorithm can be determined from the eigenvalues of the discretized state updating matrix G in the equation:

$$x(t) = G(\Delta t)x(t-\Delta t) + H(\Delta t)u \,. \quad (1)$$

Here x is the simulated state variable sampled at intervals Δt, and u the input. As discussed earlier, G and H are obtained from the matrices A and B of the continuous state variable equation by considering the integration method being used. For example, if the rectangular rule is applied to the initial state equation: $\dot{\underline{x}} = Ax + Bu$, then (with I as the appropriate n x n identity matrix):

$$\Delta \underline{x} = A\underline{x}\Delta t + B\underline{u}\Delta t \quad (2)$$

$$\underline{x}(t) = \underline{x}(t) + \Delta t = (I + A\Delta t)\underline{x}(t) + (B\Delta t)\underline{u}$$

$$or \qquad G = I + A\Delta t \text{ and } H = B\Delta t$$

Similarly, if the trapezoidal rule is used:

$$\underline{x}(t) = \left(I - \frac{A\Delta t}{2}\right)^{-1}\left(I + \frac{A\Delta t}{2}\right)\underline{x}(t-\Delta t) + \left(I - \frac{A\Delta t}{2}\right)^{-1}\frac{B\Delta t}{2}(u(t) + u(t-\Delta t)) \quad (3)$$

$$\underline{x}(t) = G\underline{x}(t-\Delta t) + H\underline{u}^*$$

However, the fact that an algorithm is stable for a stable system doesn't tell us anything about the accuracy of the solution. A frequency response plot is a good way to look at the accuracy of a solution. Expressing Eqn. 1 in the form of a z transform, one obtains:

$$\underline{X}(z) = (I - G(\Delta t)z^{-1})^{-1}H(z)\underline{U}^*(z) \quad (4)$$

$$\underline{X}(z) = \Psi(z)\underline{U}^*(z)$$

For plotting the frequency response, we substitute $z = e^{j\omega\Delta t}$ from the well known relationship between the z domain and the s domain ($z = e^{s\tau}$), where τ is the sampling period. We can then plot the frequency response of each of the components $\Psi_{m,n}(j\omega)$ and observe how it compares with the required frequency response of the continuous system T(s), s=jω obtained from:

$$\underline{X}(s) = [(sI - A)^{-1}B]\underline{U}(s) \quad (5)$$

$$\underline{X}(j\omega) = T(j\omega)\underline{U}(j\omega)$$

Let us see how this works with the first order R-L circuit discussed earlier and shown in Fig. 1

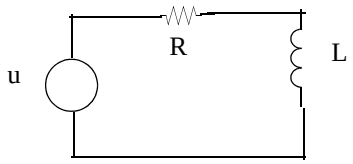


Fig 1: R-L Circuit

The appropriate state equation is:

$$\dot{i} = \left(\frac{-R}{L}\right) i + \frac{1}{L} u \qquad (6)$$

The appropriate transfer functions for the rectangular and trapezoidal rules respectively are then given by:

(7)

$$Gr(z, \Delta t) := \frac{\Delta t}{L} \cdot \frac{z^{-1}}{1 - \left(1 - \frac{R \cdot \Delta t}{L}\right) \cdot z^{-1}}$$

$$Gt(z, \Delta t) := \frac{\frac{1}{L} \cdot \frac{\Delta t}{2} \cdot \left(1 + z^{-1}\right) \cdot \left(1 + \frac{R \cdot \Delta t}{2 \cdot L}\right)^{-1}}{1 - \left(\frac{1 - \frac{R \cdot \Delta t}{2 \cdot L}}{1 + \frac{R \cdot \Delta t}{2 \cdot L}}\right) \cdot z^{-1}}$$

The frequency response can then be plotted by substituting $z = e^{j\omega\Delta t}$. For R = 1Ω, L = 100 mH and Δt = 0.01 s, the magnitude plots for the rectangular and trapezoidal rules are superposed on the actual frequency response. From the graphs it appears that both the rectangular and trapezoidal rules have similar accuracy and are good to about 25 Hz.

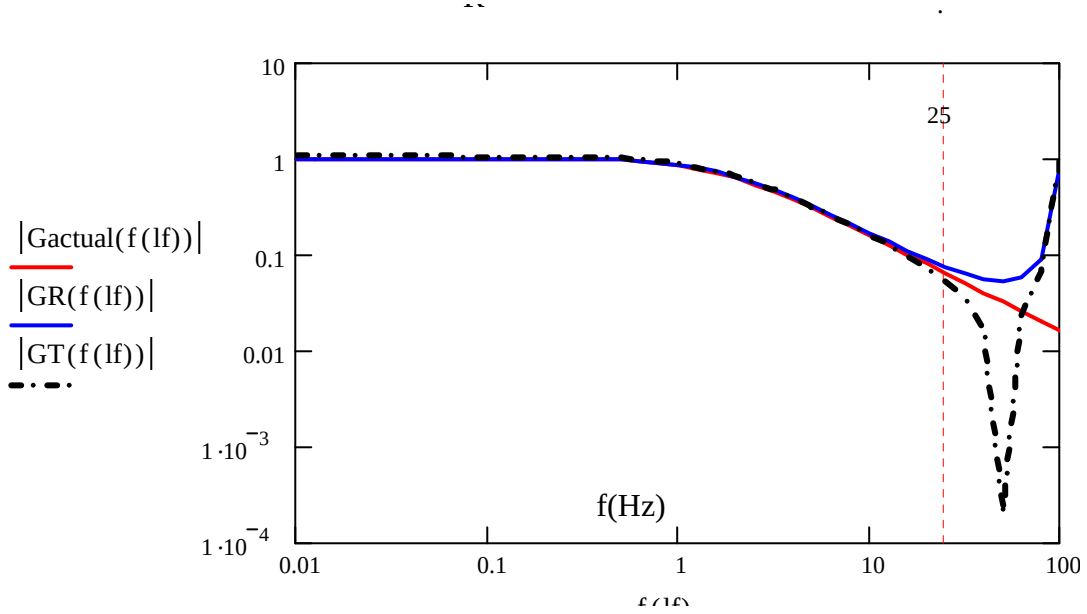


Fig 2: Magnitude response, actual, simulated (trapezoidal rule) and simulated (rect. rule)

However, as one can see, the trapezoidal rule has somewhat better *phase* response as compared to the rectangular rule.

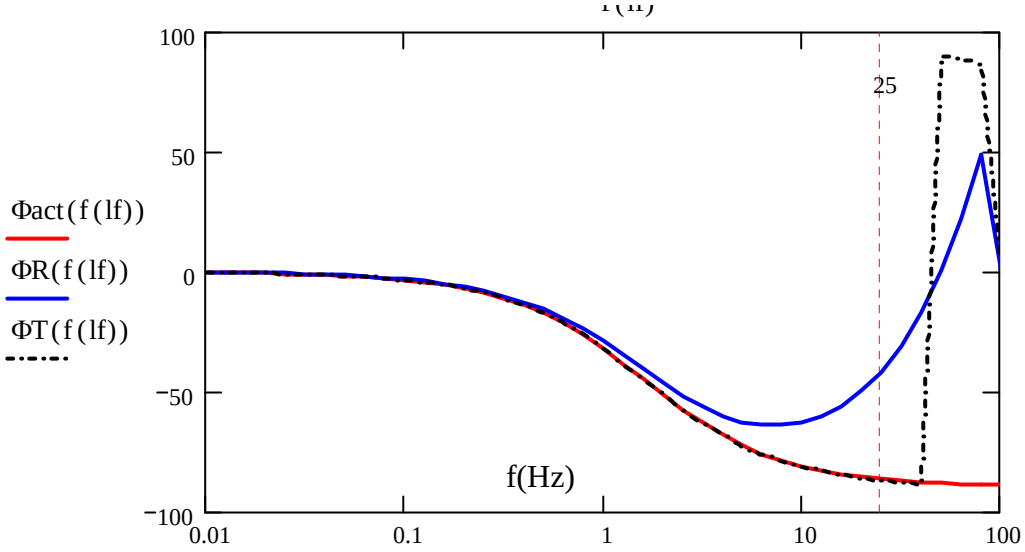


Fig 2: Phase response, actual, simulated (trapezoidal rule) and simulated (rect. rule)

However if the eigenvalues of G are plotted as a function of time-step, one sees that the rectangular rule is unstable for Δt > 0.2 *(L/R) and that the trapezoidal rule is always stable.

So note, that though the trapezoidal rule is always stable, its accuracy decreases as the timestep is increased. In the above example, it appears that the accuracy is good to about 25 Hz or so.

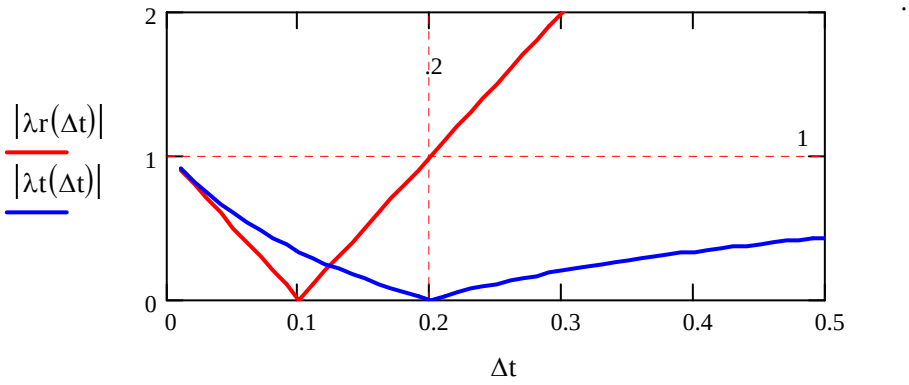


Fig 3: eigenvalues v/s Δt: Rectangular and Trap rules