

03.3 ss.graph2nt Normal trees

1 Before we introduce the algorithm for constructing the state-space model in [Lecture 03.4 ss.nt2ss](#), we introduce the first step from the system graph to the state-space model: the **normal tree**. It is a **subgraph** of the system's linear graph.

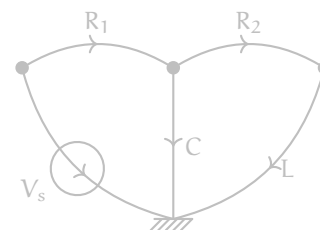
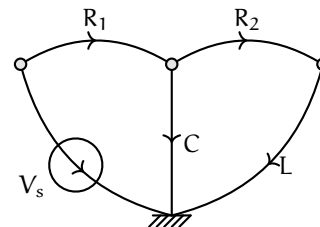
2 In the following, we will consider a connected graph with E edges, of which S are sources. There are $2E - S$ unknown across- and through-variables, so that's how many equations we need. We have $E - S$ elemental equations and for the rest we will write continuity and compatibility equations. N is the number of nodes.

3 The following rules must be respected.

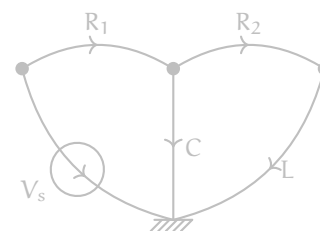
R1. There can be no loops.

R2. Every node must be connected.

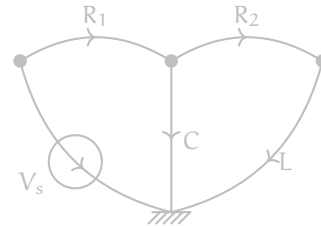
Form a normal tree with the following steps. For an inline example, we will construct a normal tree from the linear graph for an electronic system, shown at right.



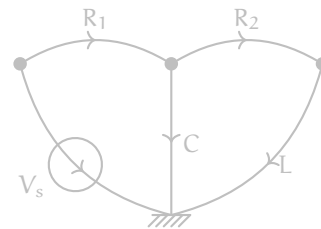
2. Include all across-variable sources.



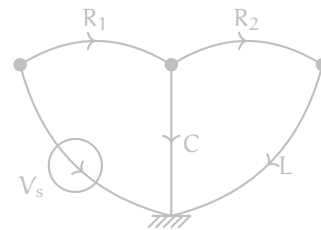
3. Select as many as possible A-type elements.



4. Select as many as possible D-type elements.



5. Select as many as possible T-type elements.



4 We call those edges in the normal tree its **branches** and those not, the **links**.

5 *A-type elements not in* and *T-type elements in* the normal tree are called **dependent energy storage elements**. All other A- and T-types are **independent energy storage elements**. The energy in these can be independently controlled.

6 In order to avoid an artificial excess in state variables and construct what is called a **controllable** model, whenever **A-types in series** (sharing one node) or **T-types in parallel** (sharing two nodes) appear, we should combine them to form equivalent elements in accordance with the formulas

$$C_e = \frac{1}{\sum_i 1/C_i} \quad \text{or} \quad (1a)$$

$$L_e = \frac{1}{\sum_i 1/L_i}. \quad (1b)$$

7 There are special names for power-flow variables associated with an element, depending on whether the element is a branch or link. **Primary**

variables are: *across*-variables on branches and *through*-variables on links.

Secondary variables are: *through*-variables on branches and *across*-variables on links.