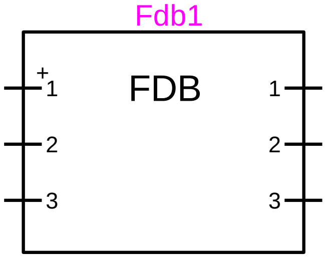
Frequency dependent branch device



Jean Mahseredjian, 12/29/2013 1:27 AM

1 Available versions

The FDB (Frequency Dependent Branch) device is a multiphase device. When only 3 phases are used, it is feasible to force 3-phase pin usage on both sides of the FDB device through its drawing tab options.

2 Applications

The FDB device model consists of a number of parallel RLC branches of the general form shown in Figure 1. In a given FDB branch there are n component branches. This model can be used to represent the behavior of a any frequency dependent device which can be synthesized with the type of branches show in Figure 1.

The model data file required by this device can be automatically produced using FDBFIT (Transformer data calculation function, “Data from FDBFIT”) or a by a dedicated support program. The equivalents represent the network in modal domain. The FDB can be used for transfer functions that can be realized with a rational function that has a finite number of poles and zeros and no time delay.

See also the documentation for the FDBFIT (“Data from FDBFIT”) device.

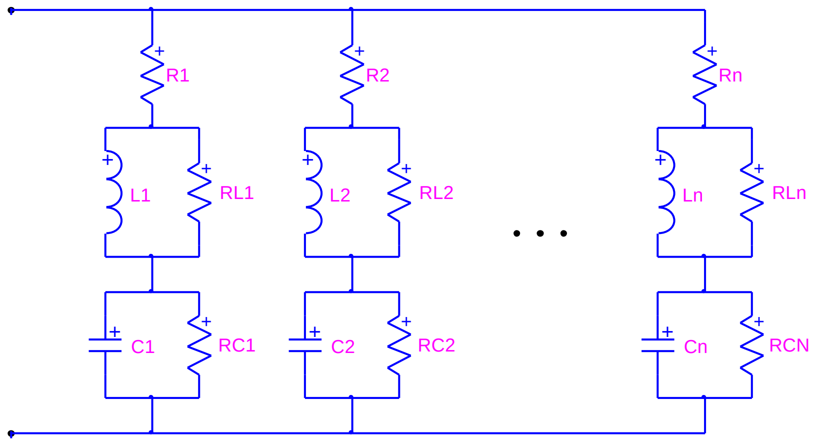


Figure 1 Generic structure of an FDB branch

3 Data tab

It is only required to enter the model data file name of this device on its first data tab. The following generic formats illustrate the expected data fields in the model data file. The first two data lines are given:

```
n_phase=number_of_phases
balanced=ibal
```

The *number_of_phases* variable following the equality sign provides the number of phases for the selected FDB. The *ibal* code is set to 1 for indicating a balanced network and 0 for indicating unbalanced case data.

When a balanced case is selected, the following data lines are required.

```
N0                     ← number of zero-sequence RLC branches
R(i) L(i) RL(i)        ← stacked branch data for i=1,2,…N0
C(i) RC(i)
N1                     ← number of positive-sequence RLC branches
R(i) L(i) RL(i)        ← stacked branch data for i=1,2,…N1
C(i) RC(i)
```

When an unbalanced case is selected, the following data lines become required:

```
N1                     ← number of RLC branches for the first mode
R(i) L(i) RL(i)        ← stacked branch data for i=1,2,…N1
C(i) RC(i)
N2                     ← number of RLC branches for the second mode
R(i) L(i) RL(i)        ← stacked branch data for i=1,2,…N2
C(i) RC(i)
...
Q                      ← modal transformation matrix
```

The last mode last branch data is immediately followed by the modal transformation matrix Q (used in EMTP for calculating the phase domain matrix from the modes). It is noticed that the matrix Q must be entered in its transposed version Q^t. Matrix data is entered row-by-row. Each row can appear on one or more lines.

Specific rules:

- All numbers are entered in free-format and separated by one or more blanks.
- The ith branch data of a given mode is entered using two lines (as shown above).
- R is expressed in Ohms, L is expressed in Henrys and C is in Farads.
- R can be set to 0.

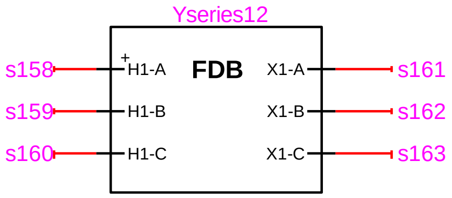
- If RL is 0 then it is replaced by an open circuit.
- If RC is 0 then it is replaced by an open circuit.
- If L is 0 then it is replaced by an open circuit.
- If C is 0 then it is replaced by an open circuit.
- If both RL and L are zero then this branch is replaced by a short-circuit.
- If both RC and C are zero then this branch is replaced by a short-circuit.

An optional data line (appearing first in the model data file) can be used to automatically identify device pins on its symbol. This line is a comment line and must start with a capital “C”. The code BUS= is followed by comma separated bus (node) names. Example:

```
C BUS= H1-A  ,X1-A  ,H1-B  ,X1-B  ,H1-C  ,X1-C  ,
```

4 Netlist

The following Netlist section example is taken from the example provided in the FDBFIT (“Data from FDBFIT”) device documentation. This device has 3 phases, for a total of 6 pins.

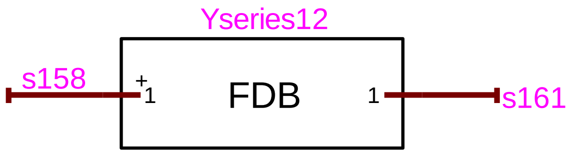


```
_FDB;Yseries12;6;6;s158,s159,s160,s161,s162,s163,
3,1,
test_fdb_demo_serie12.pun,
```

Field	Description
FDB	Part name
Fdb1	Instance name, any name.
6	Total number of pins
6	Number of pins given in this data section
s158	Signal name connected to k1-pin, top phase
s159	Signal name connected to k2-pin
s160	Signal name connected to k3-pin
s161	Signal name connected to m1-pin
s162	Signal name connected to m2-pin
S163	Signal name connected to m3-pin
3	Number of phases, 3 in this case
1	Relative path usage in the following data file name
file name	The model data file name

Relative path usage is the recommended option and it indicates that the model data file is found in a directory below the current design directory. All data is stored into the ParamsA attribute of the device.

When the device is 3-phase it can be also optionally set to use 3-phase pins on both sides.



The Netlist is now changed to a 3-phase device:

```
_FDB;Yseries12a;6;2;s158a,s161a,
3,1,
test_fdb_demo_serie12.pun,
_FDB;Yseries12b;6;2;s158b,s161b,
_FDB;Yseries12c;6;2;s158c,s161c,
```

The first phase carries the model data and the following phases provide the signals connected to phases b and c. The phase identification character (a, b or c) is automatically appended to the device instance name and signals. There is now a total of 6 pins with 2 pins listed per instance (phase) line.

5 Steady-state model

The model data file provided on the data tab is used in the steady-state solution. The admittance matrix of the model is calculated at each steady-state solution frequency.

6 Initial conditions

Automatic initial conditions are found from the steady-state solution. Manual initial conditions are not available.

7 Frequency Scan model

Similar to the steady-state. The branch impedance is found at each frequency.

8 Time-domain model

The device equations are discretized according to the integration time-step and solved at each simulation time-point.