Synthesis of passband filters with asymmetric transmission zeros

Giuseppe Macchiarella

Polytechnic of Milan, Italy

Electronic and Information Department

MILANO

Passband filters with asymmetric zeros

- An asymmetric response allows a more flexible assignment of selectivity requirements, allowing at the same time to reduce the overall filter order
- Placing asymmetric zeros respect the center of the passband (f_0) produce a response which is no more geometrically symmetric around f_0
- The synthesis techniques based on the lowpass – bandpass classical transformation cannot be directly employed (they implies a geometric symmetry in the bandpass domain)

Extension of the circuit components class

- In addition to capacitors, inductors, resistors and inverters, a new component is now introduced:

The frequency-invariant reactance (FIR) / susceptance (FIB):
$Z=jX$, $Y=jB$

- Circuits including this new component present network functions with more general properties. In particular, the response around zero frequency can be asymmetric
- The synthesis of a lowpass prototype with FIR (FIB) components allows to obtain an asymmetric response (around f_0) in the passband domain (after application of the classical lowpass – bandpass frequency transformation)

Are FIR significant from a physical point of view?

- Strictly speaking FIR are not physically realizable, so synthesized networks containing FIR are not meaningful
- This is especially true around the zero frequency, where a FIR can not be even approximated with real component (concentrated or distributed)
- In the bandpass domain however, it is possible to obtain a reactance (susceptance) which does not present a relevant variation in a small range of frequencies.
- So, a synthesized bandpass network containing FIR is significant from a practical point of view because it can be approximated with real components

Positive and Positive-real functions

- Impedances (admittances) of networks with FIR (FIB) components are <u>positive function</u> in the complex frequency variable *s* (not *positive-real* as in case of R,L,C networks).
- A rational function *f(s)* in *s* is a positive function if Re{f(s)}≥0 for Re{s}≥0. (It is a positive-real function if f(s) is real for s real)
- Most of the properties of positive and positive-real functions are similar. The main differences are:
  - The coefficient of polynomials at numerator and denominator of a positive-real function are real (complex for positive functions)
  - The roots of the polynomials occur in complex conjugate pairs for positive-real function (no such restriction for positive function)

Characteristic polynomials for positive networks

Assuming to be in the normalized domain s, the characteristic polynomials define the scattering parameters of a lossless 2-port network:

$$S_{11}(s)=\frac{F(s)/\varepsilon_R}{E(s)}, \qquad S_{21}=\frac{P(s)/\varepsilon}{E(s)}, \qquad S_{22}(s)=\frac{F_2(s)/\varepsilon_R}{E(s)}$$

All polynomials are assumed monic. The coefficients ε and ε_R are real number which are related each other.

<u>Conditions to be verified (positive requirement)</u>:

-Coefficients of *E* and *F* are complex (with same degree *n*)

-Roots of *E* have negative real part (*Hurwitz* polynomial)

-Degree of *P(s)* ≤ *n*

Unitary of S matrix (Lossless condition)

$$\mathbf{S}\cdot\tilde{\mathbf{S}}^* = \mathbf{U} \qquad \Rightarrow \qquad \begin{aligned} S_{11}(s)S_{11}(s)^* + S_{21}(s)S_{21}(s)^* &= 1 \\ S_{22}(s)S_{22}(s)^* + S_{12}(s)S_{12}(s)^* &= 1 \\ S_{11}(s)S_{12}(s)^* + S_{21}(s)S_{22}(s)^* &= 0 \end{aligned}$$

<u>Paraconjugation (s=jω)</u>: Q(s)*=Q*(s*)=Q*(-s)

$$Q(s) = q_0 + q_1 s + q_2 s^2 + ... + q_n s^n$$

$$\Rightarrow Q(s)^* = Q^*(-s) = q_0^* - q_1^* s + q_2^* s^2 - .... + q_n^* s^n \ \ (n \text{ even})$$

$$- q_n^* s^n \ \ (n \text{ odd})$$

The roots *zQ* of *Q(s)** are those of *Q(s)* with the real part of opposite sign: *zQ*=-*zQ**.

$$Q(s)^* = (-1)^n \cdot \prod_{k=1}^{n} \left(s + zQ_k^*\right)$$

Characteristic polynomials of lossless networks

Roots of *P*:

-Imaginary

-Complex pairs with opposite real part

Polynomial *P* (degree n_z) must be multiplied by *j* when (n-n_z) is even (consequence of matrix **S** unitary)

Roots of F_2:

- Equal to the negative conjugate of the roots of F:

$$zF_2 = -zF^* \quad \Rightarrow \quad F_2(s) = (-1)^n F(s)^*$$

Feldtkeller equation:

$$\frac{P(s)P(s)^*}{\varepsilon^2} + \frac{F(s)F(s)^*}{\varepsilon_r^2} = E(s)E(s)^*$$

E(s) is defined once *P(s)* and *F(s)* are known

Relationship between ε and ε_r

- When $n_z<n$:

$$S_{21}(j\infty)=\frac{P(j\infty)}{\varepsilon E(j\infty)}=0, \quad \left|S_{11}(j\infty)\right|^2=1-\left|S_{21}(j\infty)\right|^2=1 \;\Rightarrow\; \frac{\left|F(j\infty)\right|}{\varepsilon_R\left|E(j\infty)\right|}=1 \;\Rightarrow\; \varepsilon_R=1$$

- When $n_z=n$ (fully canonical condition)

$$\frac{P(j\infty)P(j\infty)^*}{\varepsilon^2 E(j\infty)E(j\infty)^*}+\frac{F(j\infty)F(j\infty)^*}{\varepsilon_r^2 E(j\infty)E(j\infty)^*}=1 \;\Rightarrow\; \frac{1}{\varepsilon^2}+\frac{1}{\varepsilon_r^2}=1$$

$$\varepsilon_r=\frac{\varepsilon}{\sqrt{\varepsilon^2-1}}$$

ε is determined once RL is imposed:

$$RL=10\log\left(\left|\frac{E(j)}{F(j)}\right|^2\right)=10\log\left(1+\left|\frac{P(j)/\varepsilon}{F(j)}\right|^2\right) \;\Rightarrow\; \varepsilon^2=\left|\frac{P(j)}{F(j)}\right|^2\frac{1}{10^{RL/10}-1}$$

The approximation problem: the characteristic function Cn and polynomials P, F

$$A(\Omega)=1+\varepsilon'^2 C_n^2(\Omega)=1+\varepsilon^2\frac{|F(j\Omega)|^2}{|P(j\Omega)|^2}=1+\varepsilon^2\frac{F(j\Omega)F(-j\Omega)}{P(j\Omega)P(-j\Omega)}\quad\Rightarrow$$

$$C_n(j\Omega)C_n(-j\Omega)=\frac{\varepsilon^2}{\varepsilon'^2}\frac{F(j\Omega)F(-j\Omega)}{P(j\Omega)P(-j\Omega)}$$

Applying the analytic continuation ($j\Omega \rightarrow s$):

$$C_n(s)=\frac{\varepsilon}{\varepsilon'}\frac{F(s)}{P(s)}$$

Given $C_n(\Omega)$ (order *n* and imposed transmission zeros), it is possible to compute the characteristic polynomials

The generalized Chebycheff characteristic function

$$
C_n(\Omega) = \begin{cases} \cos\left[(n-n_z)\cos^{-1}(\Omega) + \sum_{k}^{1,n_z} \left| \mathrm{Re}\left\{ \cos^{-1}\left(\dfrac{1-\Omega\cdot\Omega_{z,k}}{\Omega-\Omega_{z,k}} \right) \right\} \right| \right] & |\Omega| \le 1 \\ \cosh\left[(n-n_z)\cosh^{-1}(\Omega) + \sum_{k}^{1,n_z} \left| \mathrm{Re}\left\{ \cosh^{-1}\left(\dfrac{1-\Omega\cdot\Omega_{z,k}}{\Omega-\Omega_{z,k}} \right) \right\} \right| \right] & |\Omega| > 1 \end{cases}
$$

$\Omega_{z,k}$ are the assigned transmission zeros: $zP_k = j\Omega_{z,k}$

zP_k are the roots of *P(s)*, which must be imaginary or complex pairs with opposite real part. $C_n(\Omega)$ can be expressed in terms of the roots of *P(s)* and *F(s)*

$$
C_n(\Omega) = \frac{\varepsilon}{\varepsilon'} \frac{\prod_{k=1}^{n} \left(\Omega - zF_k / j \right)}{\prod_{k=1}^{n_z} \left(\Omega - zP_k / j \right)}
$$

Evaluation of polynomials P(s), F(s) given $C_n(\Omega)$

- Assign the order *n* and the transmission zeros zP_k
- Evaluate $C_n(\Omega)$ (with $\Omega_{z,k}=zP_k/j$) for $\Omega_i = \Omega_1,\ldots\Omega_N$, in the interval $-1< \Omega_i <1$ ($N>2n$)
- Generate the vector $F'(\Omega_i) = C_n(\Omega_i)\cdot\prod_{k=1}^{n_z}(\Omega_i - zP_k/j)$
- Find the coefficient of polynomial $F'(\Omega)$ by fitting $F'(\Omega_i)$ with a polynomial of order *n*
- Find the roots $\Omega_{F,k}$ of $F'(\Omega)$: $\Omega_{F,k}=zF_k/j \rightarrow zF_k=j\Omega_{F,k}$
- Generate the polynomials *F(s)* and *P(s)* from their roots (zP_k, zF_k). Multiply *P(s)* by *j* if ($n-n_z$) is even

Evaluation of E(s) using lossless condition

The imposed RL determines ε:

$$RL = 10\log\left(\left|\frac{E(j)}{F(j)}\right|^2\right) = 10\log\left(1+\left|\frac{P(j)/\varepsilon}{F(j)}\right|^2\right) \quad \Rightarrow \quad \varepsilon^2 = \left|\frac{P(j)}{F(j)}\right|^2 \frac{1}{10^{RL/10}-1}$$

If n_z=n ε_r is evaluated with the expression previously shown. Otherwise, ε_r=1.
The polynomial $E_2(s)$ is then evaluated as:

$$E^2(s) = E(s)E(s)^* = \frac{P(s)P(s)^*}{\varepsilon^2} + \frac{F(s)F(s)^*}{\varepsilon_r^2}$$

The roots of E^2 are computed and those with negative real part define the roots of E(s).

Finally, E(s) is obtained from its roots (it is monic)

More efficient method (for imaginary zF_k)

Let consider the following factorization of E^2:

$$E^2(s) = E(s)E(s)^* = \left(\frac{P(s)}{\varepsilon} + \frac{F(s)}{\varepsilon_r}\right) \cdot \left(\frac{P(s)^*}{\varepsilon} + \frac{F(s)^*}{\varepsilon_r}\right) = E_a \cdot E_b$$

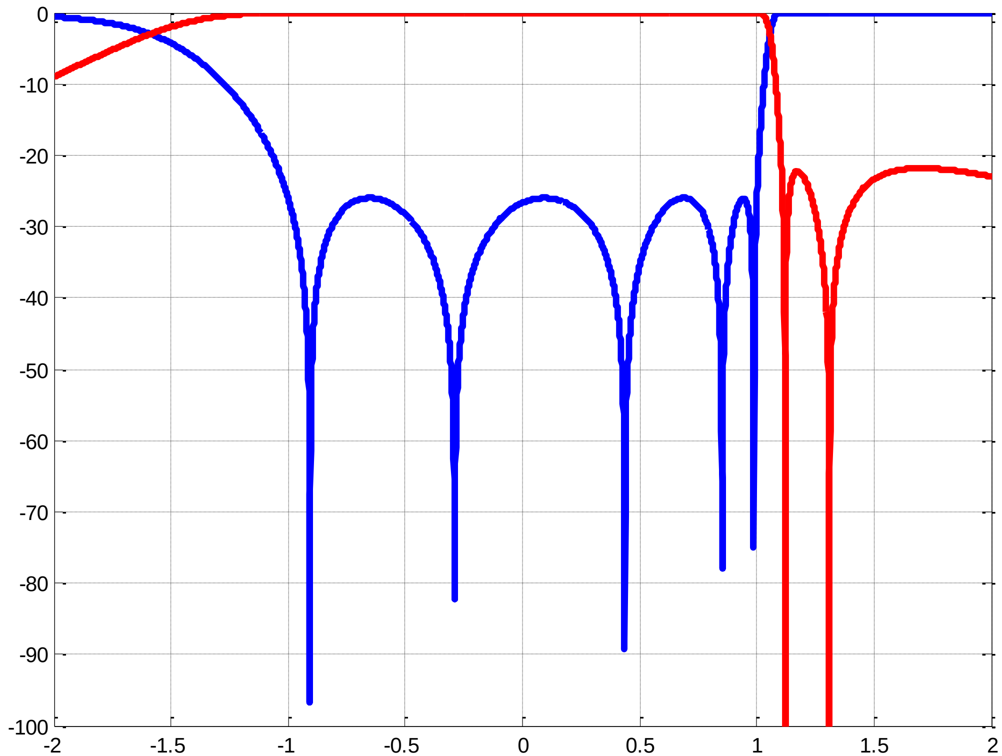
The equality holds if:

$$\left(\frac{P(s)^* F(s) + P(s)F(s)^*}{\varepsilon \varepsilon_r}\right) = 0 \quad \Rightarrow \quad P(s)^* F(s) = -P(s)F(s)^*$$

The last equation is verified if the roots of F are imaginary or symmetric with respect the imaginary axis.

The roots of *E(s)* are obtained from the roots of E_a (or E_b), by assigning all the real part negative (i.e. by changing the sign of those which are positive)

Example: n=5, RL=26dB, zP={1.12i, 1.31i}



P/ε ={1.4224, -3.4564i, -2.0869} $\varepsilon' = 0.050182$

F={1, -1.0794i, 0.81597, -1.0023i, -0.0079246, -0.096402i}

E={1, 2.64-1.079i, 4.3-3.16i, 3.624-5.58i, 0.5864-5.316i, -1.1655-1.7338i}

Group Delay Evaluation

- Group Delay in Ω domain:

$$\tau = -\frac{\partial}{\partial\Omega}\left[\angle S_{21}(\Omega)\right] = -\frac{\partial}{\partial\Omega}\left[\angle P(\Omega) - \angle E(\Omega)\right]$$

$$\angle P(\Omega) = \angle\prod_{k=1}^{n_z}\left(j\Omega - zP_k\right), \qquad \angle E(\Omega) = \angle\prod_{k=1}^{n}\left(j\Omega - zE_k\right)$$

The phase of *P(Ω)* is independent on *Ω*: in fact, the roots zP_k are on imaginary axis (0 contribute) or in pair with opposite real part (contributes of opposite sign). Then:

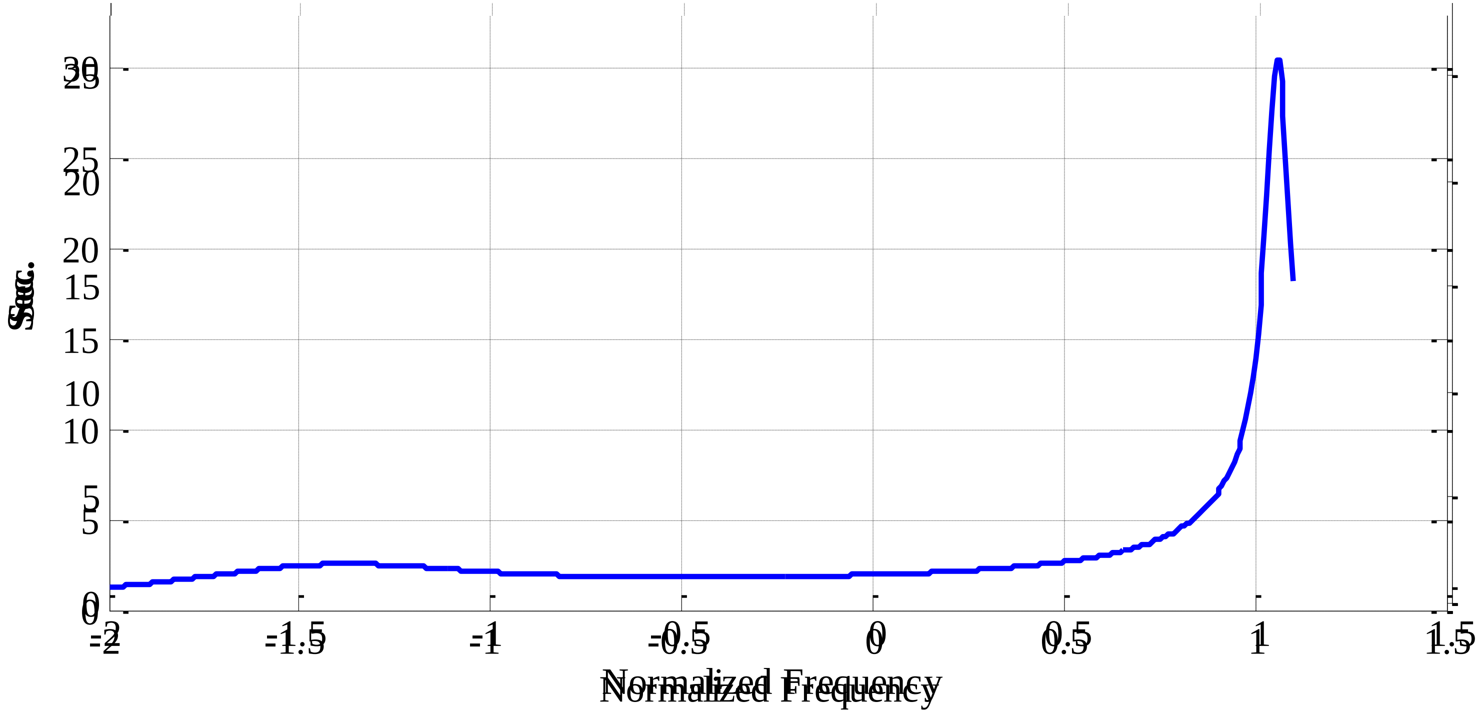
$$\angle S_{21}(\Omega) = -\angle E(\Omega) = -\sum_{k=1}^{n}\tan^{-1}\left(\frac{\Omega - \mathrm{Im}(zE_k)}{\mathrm{Re}(zE_k)}\right)$$

$$\tau = -\frac{\partial}{\partial\Omega}\left[-\angle E(\Omega)\right] = \sum_{k=1}^{n}\frac{\mathrm{Re}(zE_k)}{\left[\mathrm{Re}(zE_k)\right]^2 + \left[\Omega - \mathrm{Im}(zE_k)\right]^2}$$

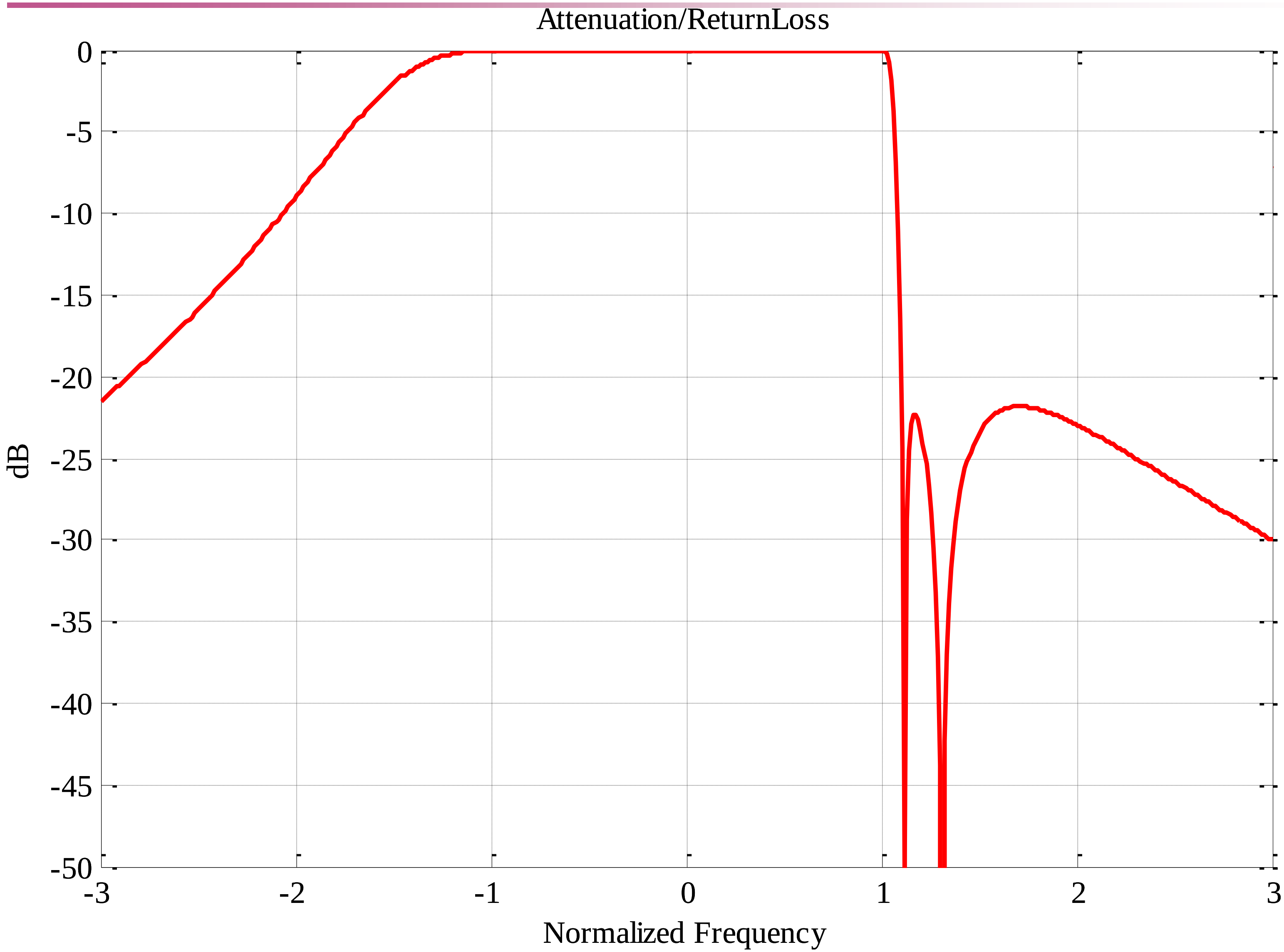
Complex zero for phase equalizing

Group Delay of the previous example:
Max value in passband: 14 sec

Introducing two complex zeros ($\pm0.25-1.05i$):
Max value in passband: 9 sec



Attenuation worsens!

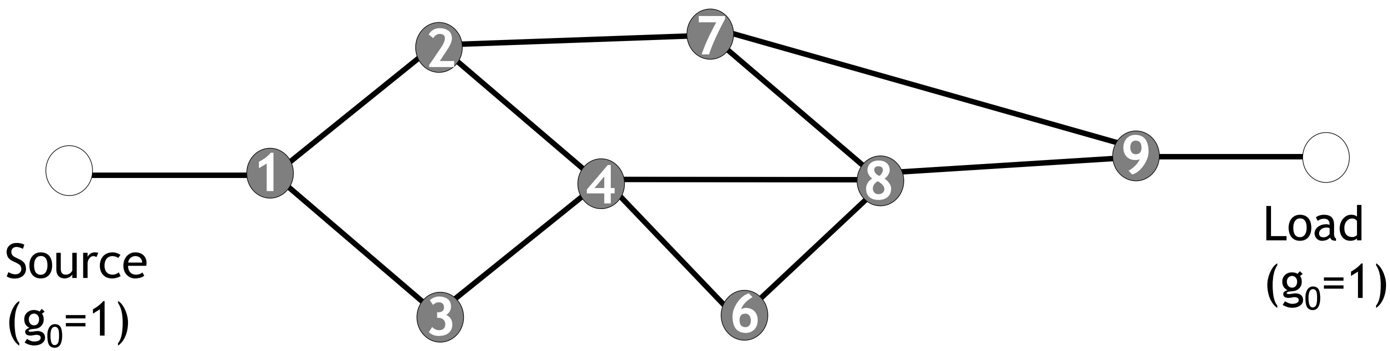


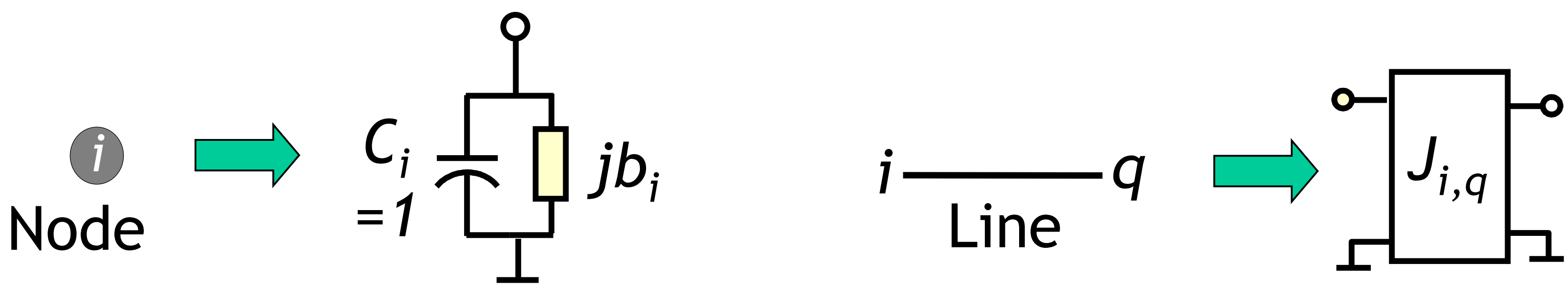
Synthesis of the filter in the normalized domain (Ω)

- Filtering networks presenting transmission zeros can be classified in two very general categories:
  - <u>Crossed-coupled networks</u>: the transmission zeros are generated by means of multiple paths which allow the output signal to vanish at some frequencies
  - <u>Extracted pole networks</u>: each transmission zero (pure imaginary) is realized by means of a suitable impedance (admittance) which blocks the transmission between input at output at a specific frequency
- The networks synthesized in Ω are called *prototypes*. To obtain the network in the final bandpass domain ω it is necessary to perform a *de-normalization* process.
- Prototypes with the number of transmission zeros equal to the number of poles are called *fully canonical*

Cross-coupled prototype networks

- General topology: conventional representation





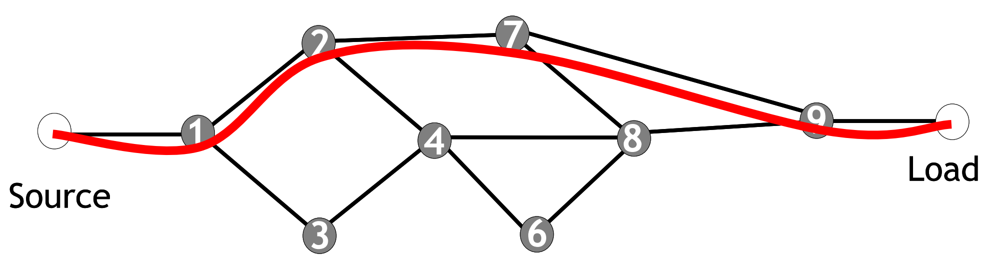
Given a set of polynomials F,P,E defining the generalized Chebyceff characteristic, it is always possible to find one (or more) topology for the cross-coupled network implementing the response determined by the assigned polynomials

Minimum path rule

- Given a cross-coupled topology, the maximum number of transmission zeros which can be accommodated is determined by the "*minimum path rule*":

"The maximum number of transmission zeros is equal to the prototype order (*n*) minus the number of nodes touched for going from the source to the load (*np*)"

$$n_z = n - np$$



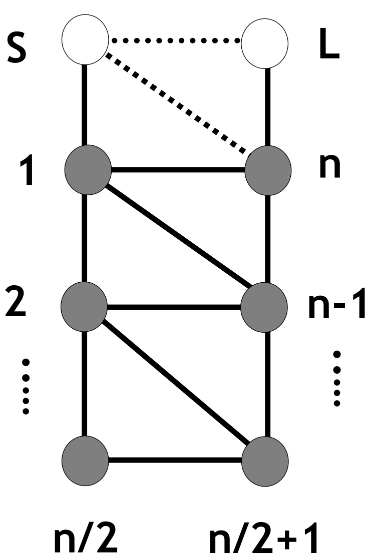
n_z=9-4=5

n=9, *np*=4

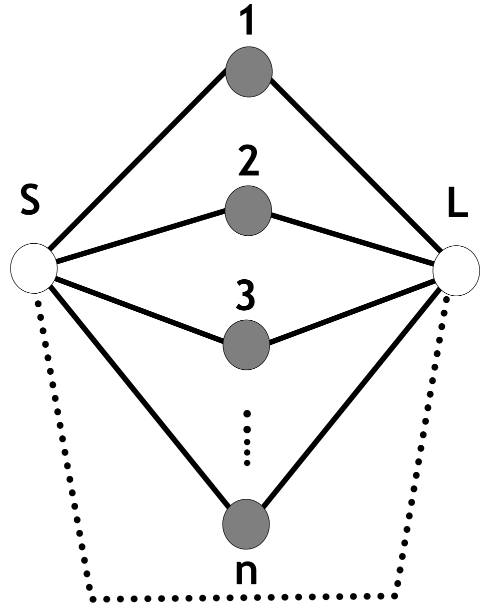
Canonical Prototypes

- There is a particular prototypes category whose topology can always be synthesized once the characteristic polynomials are assigned. The prototypes obtained are called *canonical*

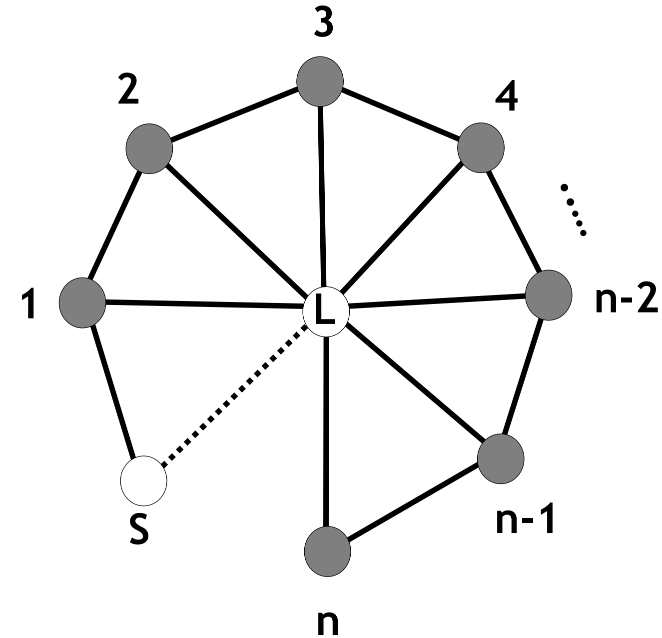
<u>Most important canonical prototypes</u>:



Folded



Transversal

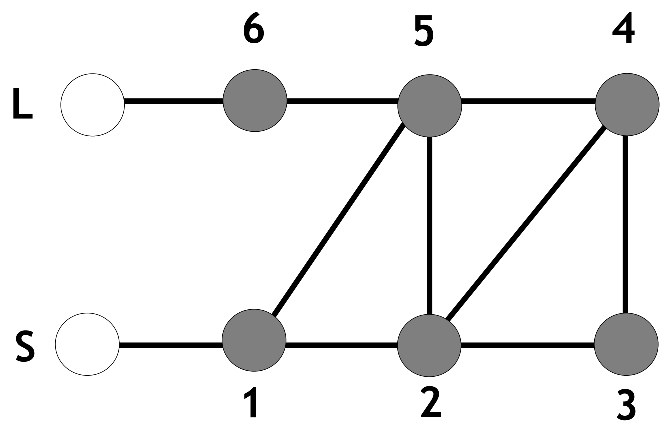


Wheel

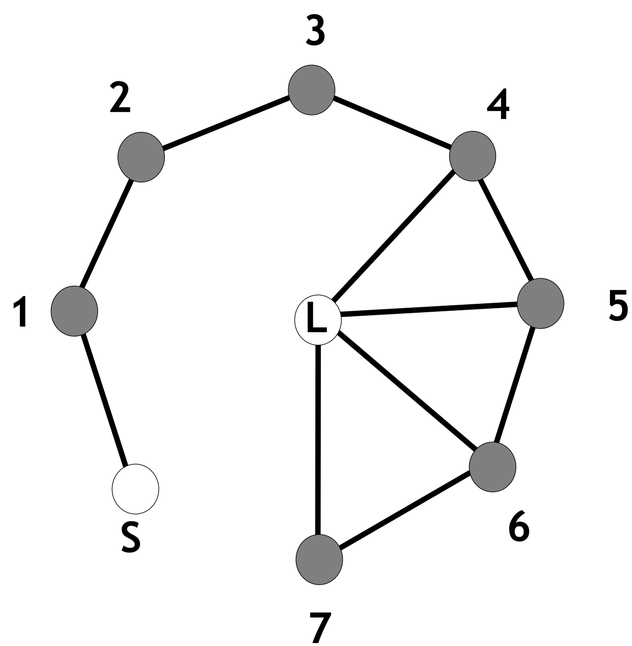
Notes on Canonical Prototypes

- Not all the couplings are different from zero (this is true only for transversal prototype)
- In folded and wheel prototypes there are two kinds of couplings: the *direct* (those connecting sequential nodes) and the *cross* (those between not-consecutive nodes).
- The number of not-zero cross couplings depends on the number of transmission zeros (according to the minimum path rule)
- In the folded prototype cross couplings involving source and load are necessary when $n_z>n-2$
- For fully canonical prototypes ($n_z=n$) a coupling between load and source is requested

Examples



n=6, np=3 → nz=3



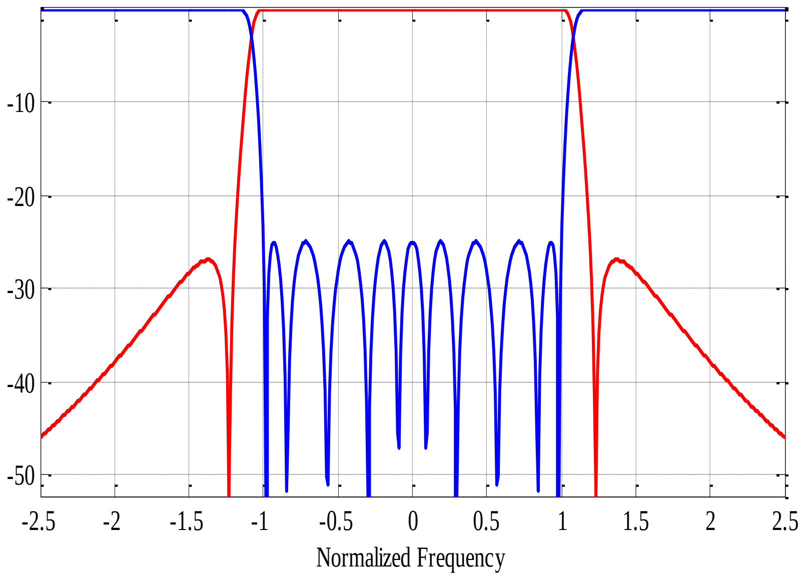
n=7, np=4 → nz=3

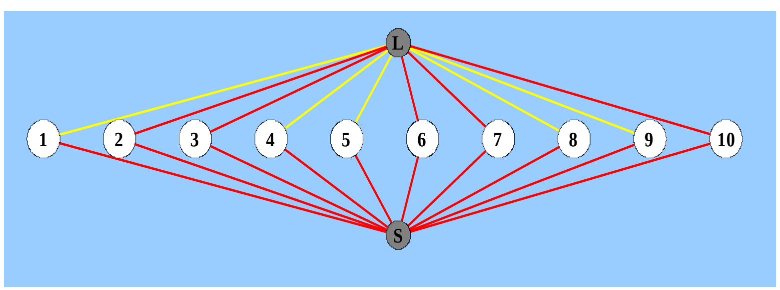
Canonical prototypes with symmetric response

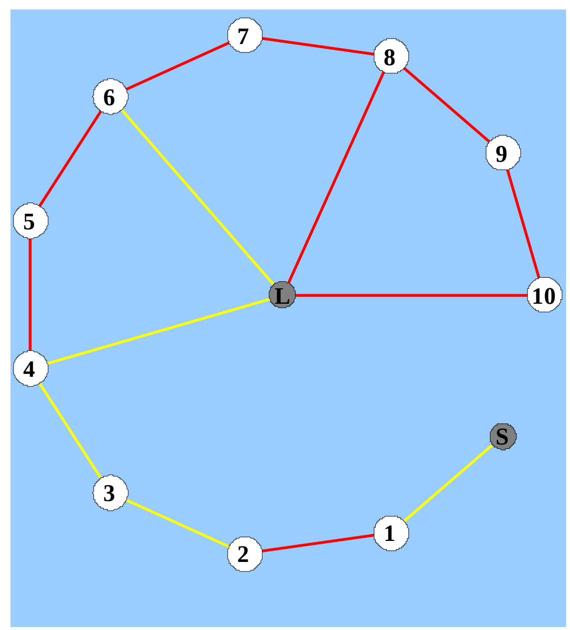
- Symmetric response in the normalized domain Ω is obtained with transmission zeros symmetrically placed around real axis
- The diagonal elements of M are null
- The corresponding prototype does not include FIR (FIS) elements (positive-defined network)
- The canonical prototypes networks have specific properties:
  - **Folded**: oblique cross couplings are zero
  - **Wheel**: cross couplings terminating on the load vanish alternately
  - **Transversal**: couplings $M_{1,k}$ and $M_{k,n+2}$ have the same magnitude

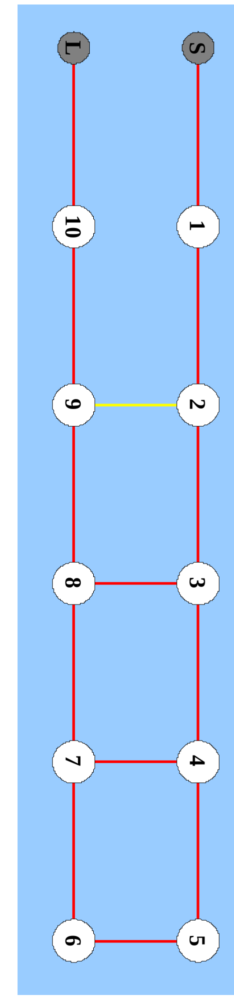
Example

N=10, transmission zeros: [±1.23i, ±0.3±0.1i], RL=25

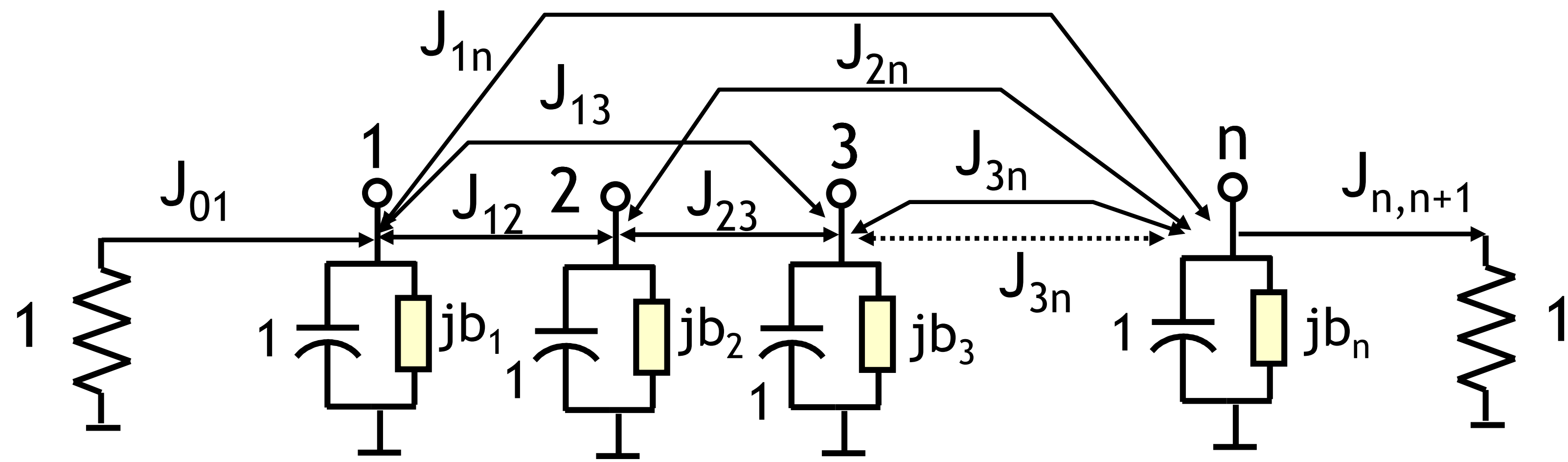


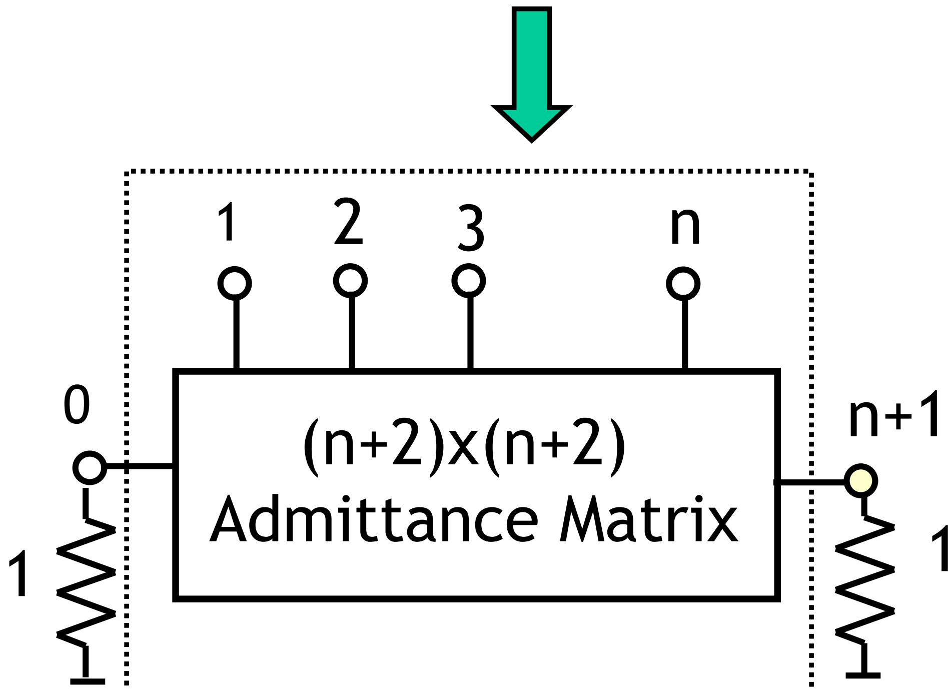






Circuit analysis of cross-coupled prototypes: formation of the (n+2) x (n+2) admittance matrix Y





$$Y_{i,j} = jJ_{i,j}$$

$$Y_{1,1} = Y_{n+2,n+2} = 0$$

$$Y_{i,i}\Big|_{i\neq 0} = s + jb_i$$

Normalized Coupling Matrix M

$$\mathbf{Y} = \begin{bmatrix} 0 & jJ_{01} & 0 & \ldots & 0 \\ jJ_{01} & s + jb_1 & jJ_{12} & \ldots & jJ_{1n} \\ jJ_{02} & jJ_{12} & s + jb_2 & \ldots & jJ_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & jJ_{n,n+1} & 0 \end{bmatrix} = s\mathbf{U_n} + j\mathbf{M}$$

$$\mathbf{M} = \begin{bmatrix} 0 & J_{01} & 0 & \ldots & 0 \\ J_{01} & b_1 & J_{12} & \ldots & J_{1n} \\ J_{02} & J_{12} & b_2 & \ldots & J_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & J_{n,n+1} & 0 \end{bmatrix}$$

Normalized Coupling Matrix

Evaluation of the scattering parameters from M

- The Y matrix is computed at frequencies $s=j\Omega_k$:

$$\mathbf{Y}_k = j\Omega_k \mathbf{U_n} + j\mathbf{M}$$

The Z matrix is obtained by inverting Y:

$$\mathbf{Z}_k = \mathbf{Y}_k^{-1}$$

- The matrix Z' (2x2) is extracted from Z_k by cancelling all rows and columns except the first and last:

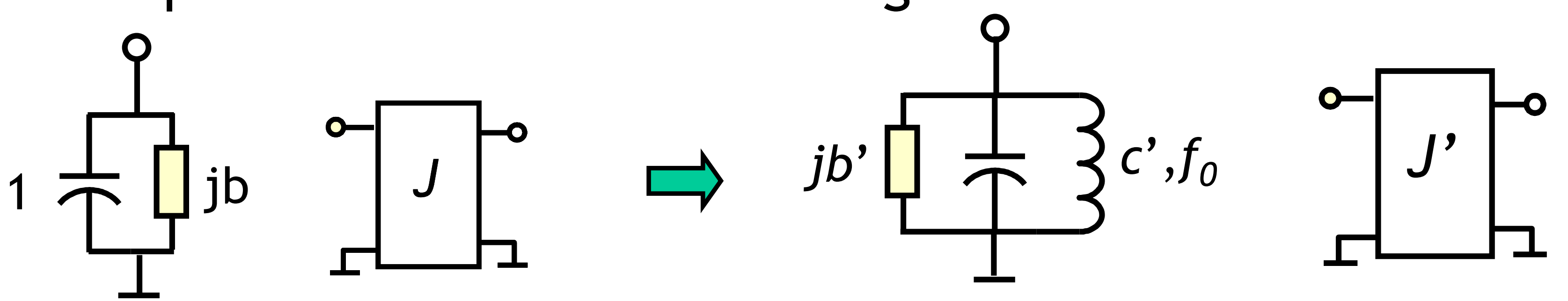
$$\mathbf{Z}'_k = \begin{bmatrix} Z_{0,0} & Z_{0,n+1} \\ Z_{0,n+1} & Z_{n+1,n+1} \end{bmatrix}$$

- The scattering matrix of the prototype is computed from Z':

$$\mathbf{S}_k = \left(\mathbf{Z}'_k - \mathbf{U}\right)\cdot\left(\mathbf{Z}'_k + \mathbf{U}\right)^{-1}$$

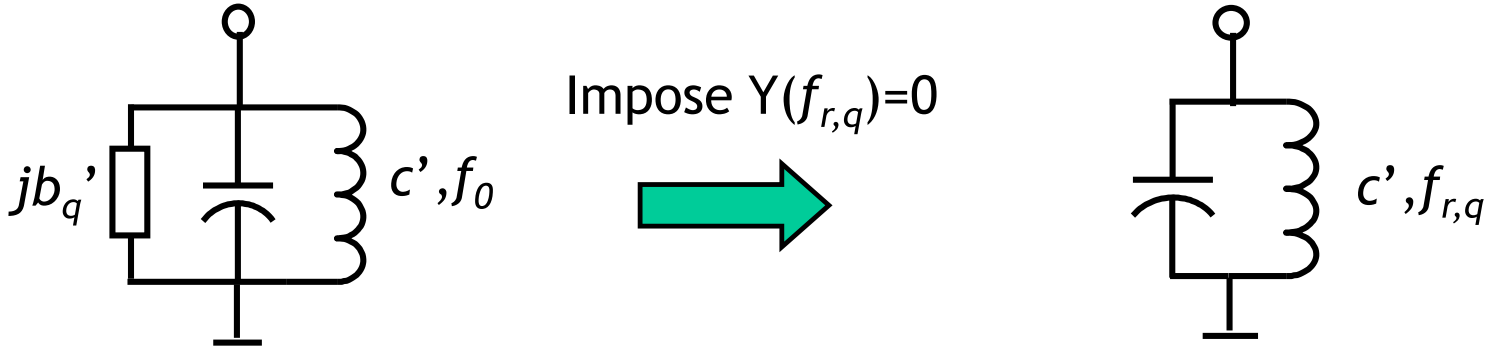
De-normalization of prototype networks

- De-normalization consists in the network transformation from the normalized domain Ω to the bandpass domain f, using the classical frequency transformation: $\Omega=(f_0/B)(f/f_0-f_0/f)$
- At circuit level, this transformation is obtained by replacing the unit capacitance with a shunt resonator. If also the external loads are scaled from 1 to G_0 the correspondence between normalized and de-normalized components are is the following:



$$c' = \frac{G_0}{2\pi B}, \quad b' = b \cdot G_0, \quad J' = J \cdot G_0$$

Approximated de-normalized resonators



$$\omega_0 c'_q \left(\frac{f_{r,q}}{f_0} - \frac{f_0}{f_{r,q}} \right) + b'_q = \frac{f_0}{B} \left(\frac{f_{r,q}}{f_0} - \frac{f_0}{f_{r,q}} \right) + b_q = 0 \quad \Rightarrow$$

$$\frac{f_{r,q}}{f_0} = -\frac{B}{f_0}\frac{b_q}{2} + \sqrt{\left(\frac{B}{f_0}\frac{b_q}{2} \right)^2 + 1}$$

The resonators in a cross-coupled bandpass filter are in general <u>no synchronous</u>. The approximation is typically acceptable for (B/f_0)<<1.

Note that the b'_q do not influence the coupling coefficients, which must be evaluated at f_0

De-normalized bandpass network

- The de-normalized network is constituted by coupled resonators with the following coupling coefficients:

$$k_{i,j} = \frac{J'_{i,j}}{\omega_0 c'} = \frac{B}{f_0} J_{i,j} = \frac{B}{f_0} M_{i,j}$$

- The resonant frequency of *i-th* shunt admittance results:

$$\frac{f_{ris,i}}{f_0} = -\frac{B}{f_0}\frac{b_i}{2} + \sqrt{\left(\frac{B}{f_0}\frac{b_i}{2}\right)^2 + 1} = -\frac{B}{f_0}\frac{M_{i,i}}{2} + \sqrt{\left(\frac{B}{f_0}\frac{M_{i,i}}{2}\right)^2 + 1}$$

- External Q produced by the *q-th* resonator coupled to source (load):

$$Q_{E,q} = \frac{2\pi f_0 \cdot c'}{J'^2_{0,q}/G_0} = \frac{1}{(B/f_0)\, J^2_{0,q}} = \frac{1}{(B/f_0)\, M^2_{0,q}}$$

Dependence of filter response on the coupling parameters

- Once the parameters $k_{i,j}$, $f_{ris,i}$ and $Q_{E,q}$ are defined, also the filter response is uniquely determined.
- This means that there are infinite networks, differing for the circuit component values but with same coupling parameters, which present the same response (identical scattering parameters)
- The circuit component values have however an influence on the voltages and currents along the filter; moreover, there could be some combinations of components values which result in an easier implementation while other values may even not allow the physical realization of the filter

Synthesis of canonical prototypes: the circuit approach

- Starting point: evaluation of Chain Matrix from characteristic polynomials:

$$[ABCD] = \frac{1}{jP(s)}\begin{bmatrix} A(s) & B(s) \\ C(s) & D(s) \end{bmatrix}$$

$$2E(s) = A(s)+B(s)+C(s)+D(s), \qquad 2F = A(s)+B(s)-C(s)-D(s)$$

$$A(s) = E_e(s)+F_e(s), \qquad B(s) = E_o(s)-F_o(s)$$

$$C(s) = E_o(s)+F_o(s), \qquad D(s) = E_e(s)-F_e(s)$$

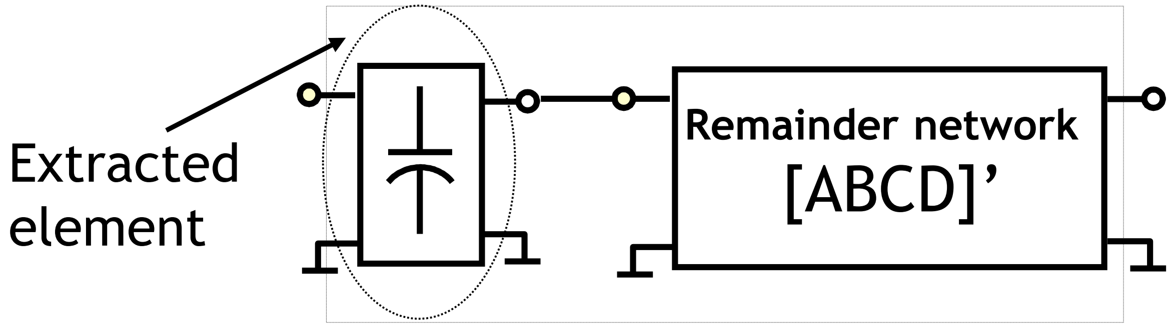
Where the subscript *e*,*o* define the even and odd part of a polynomial:

$$Q(s) = Q_e(s)+Q_o(s)$$

$$Q_e(s) = \frac{Q(s)+Q(s)^*}{2}, \qquad Q_o(s) = \frac{Q(s)-Q(s)^*}{2}$$

Synthesis by extraction

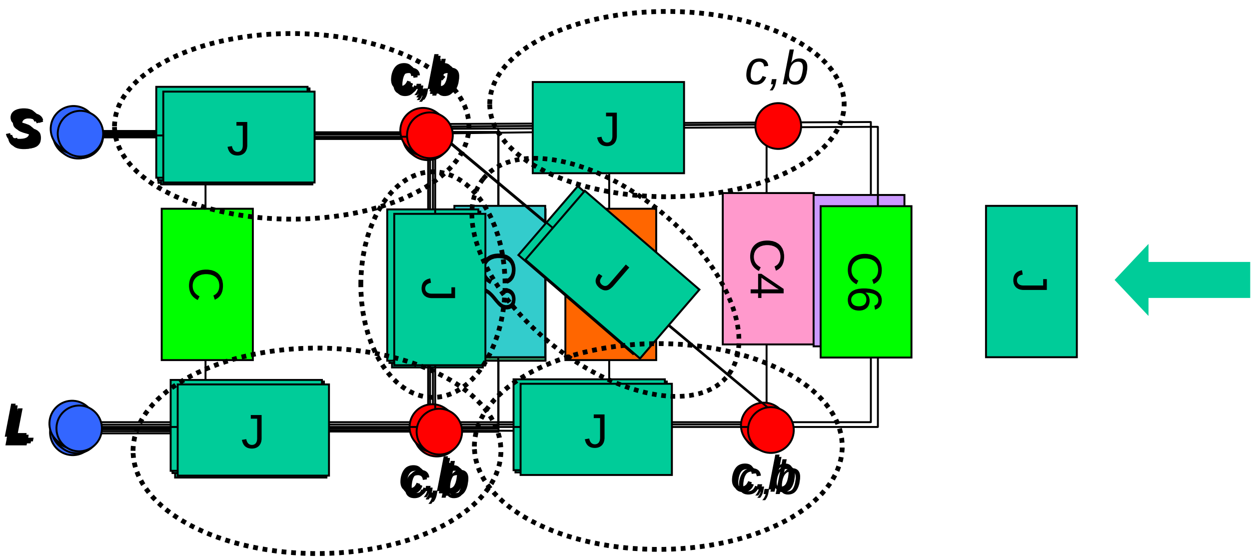
- The synthesis is performed by subsequent extractions from the [ABCD] matrix of the prototype:



[ABCD] matrix

- Suitable rules are available for the synthesis of specific canonical prototypes (see the works of Cameron on the *folded* prototype)

Example of synthesis (folded prototype)



Scaling of resonator nodes

- The circuit synthesis does not produce in general a normalized prototype (i.e. the capacitances are not all equal to 1)
- Using the conservation of the coupling coefficient $k_{i,j}$, it is easy to evaluate the elements of the coupling matrix $M_{i,j}$ resulting from synthesized components $J_{i,j}$, c_i, c_j:

$$M_{i,j} = \frac{J_{i,j}}{\sqrt{c_i \cdot c_j}}$$

- The elements $M_{i,i}$ resulting from the synthesized frequency-invariant b_i are given by:

$$M_{i,i} = \frac{b_i}{c_i}$$

Direct Synthesis of the Coupling Matrix

- This technique consists in the direct evaluation of the coupling matrix M without resort to an explicit circuit synthesis
- The method has been proposed by Cameron and allows the evaluation of the transversal canonical prototype
- Details of the method, which rely on the relationship between the Coupling matrix and the short-circuited Admittance Matrix of the transversal prototype, can be found in the literature
- The computation procedure, can be easily automated in a computer program (input data: the characteristic polynomials)

Coupling Matrix reconfiguration

- Once a canonical prototype is available, it is possible to derive other topologies by performing suitable *transformations* of the synthesized coupling matrix
- The transformation must conserve the response of the network
- A class of topological transformations with such a property is represented by the *Similarity Transform* (Given’s Rotation)
- Starting from the Transversal Prototype, specific transformations are available for obtaining the other canonical prototypes (folded, wheel)

SynFil: A software for filters synthesis

- SynFil is a software for the synthesis of microwave filters with cross-coupled and extracted-pole topologies. It can be used freely during this Course (until the end of 2021).
- It can be downloaded from this link:

  http://macchiarella.faculty.polimi.it/Dottorato2015/SynFil.zip

- A password is required to unpack the downloaded file. Matlab 2020 should be installed in the PC. Otherwise, version 9.8 (R2020a) of Matlab Runtime is required.
- This Runtime can be downloaded from:

https://www.mathworks.com/products/compiler/mcr/index.html

De-normalized coupling matrix M’

- The element of matrix **M’** are defined as:

$$M'_{i,j} = \left(B/f_0\right) \cdot M_{i,j} \qquad \left(i > 0, \quad j < N+1, \;\; i \neq j\right)$$

$$M'_{0,j} = \left(B/f_0\right) \cdot M_{0,j}^2 \;\; \left(j < N+1\right), \quad M'_{i,N+1} = \left(B/f_0\right) \cdot M_{i,N+1}^2 \;\; \left(i > 0\right)$$

- The off main diagonal elements $M'_{i,j}$ represent the coupling coefficient $k_{i,j}$ *(i>0, j<N+1)*
- The diagonal elements $M'_{q,q}$ determine the resonance frequencies of *q-th* node:

$$\frac{f_{ris,q}}{f_0} = -\left(M'_{q,q}/2\right) + \sqrt{\left(M'_{q,q}/2\right)^2 + 1}$$

- The elements $M'_{0,j}$ ($M'_{i,N+1}$) are the inverse of the external Q of resonators *j* (*i*):

$$Q_{E,j} = 1/M'_{0,j} \qquad Q_{E,i} = 1/M'_{i,N+1}$$

Evaluation of the de-normalized response from M' including losses

- The **Y** matrix vs. frequency (for G_0=1) can be written as:

$$\mathbf{Y}(f)=\left(\frac{f_0}{B}\right)\left[\frac{1}{Q_0}\mathbf{U_n}+j\left\{\left(\frac{f}{f_0}-\frac{f_0}{f}\right)\mathbf{U_n}+\mathbf{M}'\right\}\right]\simeq$$

$$=\left(\frac{f_0}{B}\right)\left[\frac{1}{Q_0}\mathbf{U_n}+j\left\{diag\left(\frac{f}{f_{ris,i}}-\frac{f_{ris,i}}{f}\right)+\mathbf{M}''\right\}\right]$$

where Q_0 is the unloaded Q of the resonators.
The approximated expression assumes that the frequency invariant elements are represented by de-tuned resonators (from f_0 to $f_{ris,i}$, see next slide); M'' is then obtained from M' by putting the element of the main diagonal to zero.