Chapter 4
Formal description of feedback

The design of feedback amplifiers involves the analysis and synthesis of the feedback loop to prevent oscillatory behavior. Various manual feedback analysis methods are developed since the very beginning [401,403], using loopgain and other well-known concepts. In addition, various automated circuit analysis methods are available since the development of circuit simulators.

Feedback analysis differs from circuit analysis. It is aimed at simplifying the transfer description of the signal flow in the feedback loop. This simplification is crucial for handling the complexity of feedback synthesis, as is discussed in chapter 5.

Restriction of conventional methods

Manual feedback analysis methods are adequate for relatively simple feedback networks at relatively low frequencies, e.g. below 1% of the transition frequency f_T of the individual transistors. The more sophisticated the feedback loop of a wideband amplifier is, the more its feedback analysis relies on automated circuit analysis.

Automated circuit analysis, using transistor models with estimated parameter values, may be adequate over a wider frequency interval. As a rule of the thumb, this applies when designing second order[1] feedback loops, up to 10% of f_T. The higher the feedback order and the wider the frequency band of interest, the more crucial a reliable circuit analysis will be.

The previous chapters 2 and 3 discussed in detail how to characterize the transfer of semiconductor devices, such as transistors. Furthermore, it was discussed how to extract adequate models from two-port measurements that take full account for relevant parasitic effects. These basic provisions facilitate an adequate circuit analysis, preferably with a simulator.

Wideband feedback design is limited by parasitic effects that are dominant. It requires a feedback analysis that takes full account for these effects. This holds especially when dealing with the stability of feedback loops, which is one of the main problems in wideband feedback design.

So far, we found no more than approximating methods in the literature to analyze feedback loops. Furthermore, there is in general no answer to the question of dealing with parasitic effects that modifies the dominant behavior of the feedback loop.

A novel approach

This chapter provides mathematical tools, to analyze feedback loops and their stability. It starts from a generalized feedback analysis that fully accounts for parasitic effects. Similarly to conventional synthesis methods, the proposed method is focussed on loop gain, however the transfer function is evaluated using multi-port methods.

[1] The term feedback *order* is adopted from filter theory, and refers to the dominant order of the loop-gain transfer-function. *Higher order* feedback is reserved for second order loops, third order loops or higher.

Furthermore, the proposed method extracts pole-zero representations for this loop gain, and converts these intermediate results into the simplest pole-zero representation that fulfills the accuracy requirements.

The highlights of this chapter, which are developed in this study, are:

- Development of a generalized feedback theory, that takes full account for parasitic effects and that is suitable for implementation in circuit simulators.
- Development of a new simple and robust algorithm that extracts the poles and zeros of a transfer function, for reliable pole-zero analysis. This was essential to the applicability of the feedback theory proposed here.
- Development of automated deflation algorithms that reduce the transfer order of the loop gain into its dominant form.
- Exact loop gain calculation, without ignoring any (linear) parasitic effects, by two specific cuts in the loop. The loop gain is derived from the two-port parameters of both the isolated parts of the loop.
- Accurate approximation of loop gain, by one cut in the loop at an arbitrary location. The loop gain is derived from the two-port parameters of the opened loop.
- Development of manual analysis methods for the contribution of zeros in the right half plane (RHP) to the overall loop transfer function.

4.1. Superposition analysis of feedback amplifiers

A linear circuit description using N-port matrix parameter is an exact representation. A tabular format, with numerical values for each frequency and node of interest is quite suitable for measurement and (automated) analysis. A basic drawback of this method is its large number of numerical data, which makes N-port matrix parameters inconvenient for manual transfer analysis and for manual compensation synthesis to realize stable feedback amplifiers.

The simplification of the problem is a successful approach in the design of feedback amplifiers. The elementary feedback model of Black's feedback patent [401], shown in figure 4.1, is based on this approach. Unfortunately, this model is too simple for wideband feedback design, because it requires transfer functions that are unilateral and not affecting each other,

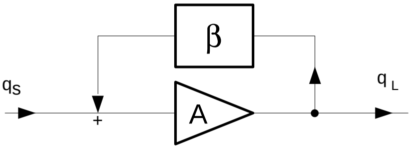


Fig 4.20 *Flow diagram of Black's feedback model, which is too simple for wideband feedback design.*

Nordholt [406] has described some extended models that are more suitable for this purpose. He formalized the superposition feedback analysis of Cherry and Hooper [405] with the description of the *superposition model*. This superposition model facilitates an exact representation of a feedback amplifier.

Further, the referred publication has reduced this superposition model to the *asymptotic gain model* which is simpler and therefore more convenient for manual analysis and synthesis. As a result, the synthesis is focussed on this asymptotic gain model and does not discribe in detail how to extract all superposition parameters of an arbitrary circuit.

This section generalizes the theory of the superposition model.

4.1.1. Superposition model parameters

In a well-designed feedback amplifier, the source signal is amplified by one or more cascaded amplifier stages and then fed to a load impedance. This enables a designer to designate a forward signal flow, and a feedback flow.

Figure 4.2 symbolizes this signal flow for an arbitrary feedback circuit. Source and load impedance are included in figure 4.2, because they affect the transfer and the stability of a feedback amplifier.

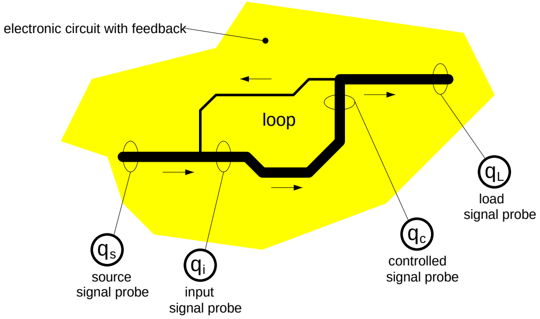


Fig 4.20 *Signal flow in an arbitrary feedback amplifier, including the source and load. The four probes sense the signal flow through a reference plane, including voltage, current, wave, charge, or any other related quantity.*

A superposition feedback analysis of the circuit in figure 4.2 requires that the signal flow is sensed by probes at four locations. The different probes may sense different quantities, including voltage, current, wave, charge or any other related quantity. It is the designer's choice what kind of signal is sensed, and as long as the signal dimension is undefined, they are referred as q_s, q_i, q_c and q_L.
The probes do not sense the signal at a single circuit node but through a reference plane. This means that a voltage-probe senses between a node and ground, a current-probe senses the current flow from one node to another, a wave-probe senses both voltage and current to reconstruct the wave, etcetera.

A superposition feedback analysis simplifies the circuit analysis as follows:

- It cuts the circuit in figure 4.2 at the four reference planes, associated with q_s, q_i, q_c and q_L, to split the circuit in (1) a one-port *source*, (2) a one-port *load*, (3) a two-port *forward amplifier* and (4) a four-port *feedback network* in which all remaining components resides.
- Then it reconstructs the original signal flow in the feedback network by external current injection in port q_s and q_c and proper termination of all the four ports with specific impedances. According to the superposition principle, the relation between the signal quantities has the following general matrix form:

$$\begin{bmatrix} q_L \\ q_i \end{bmatrix} = \begin{bmatrix} \rho & \nu \\ \xi & \beta \end{bmatrix} \cdot \begin{bmatrix} q_s \\ q_c \end{bmatrix}$$

- Next it reconstructs the original signal flow in the forward amplifier by external current injection in port q_i and proper termination of all the two ports with specific impedances. The relation between the resulting signal quantities has then the following general (matrix) form:

$$[q_c] = [A] \cdot [q_i]$$

Since the signal flow in the individual pieces is made identical to that of the overall circuit, the analysis of the individual pieces facilitates a faithful reproduction.

The superposition feedback analysis includes the presence of four signal quantities that are interrelated by five superposition parameters. We will refer to them by the following names[2] :

$q_c = A \cdot q_i$	q_s = *source quantity*	A = *forward gain*
$q_L = \rho \cdot q_s + \nu \cdot q_c$	q_L = *load quantity*	β = *feedback factor*
$q_i = \xi \cdot q_s + \beta \cdot q_c$	q_i = *input quantity*	ξ = *input conversion*
	q_c = *controlled quantity*	ν = *output conversion*
		ρ = *forward leakage*

Fig 4.3 *Superposition (model) parameters.*

One extraction method of the five superposition parameters ρ, ν, ξ, β and A starts from a full matrix description of the individual N-ports. The required terminations are then extracted from all these matrix parameters and the five superposition parameters are evaluated from these intermediate results. Since a full matrix description of an N-port requires a set with N^2 matrix parameters, this might be an inconvenient approach.

Another method starts from the observation that all source, load and termination impedances have specific values during the superposition analysis. This means that there is no need to evaluate all N-port matrix parameters. As a result, it will not isolate parts from the circuit in terms of individual N-ports but in terms of *ideal* sources, loads and amplifiers.

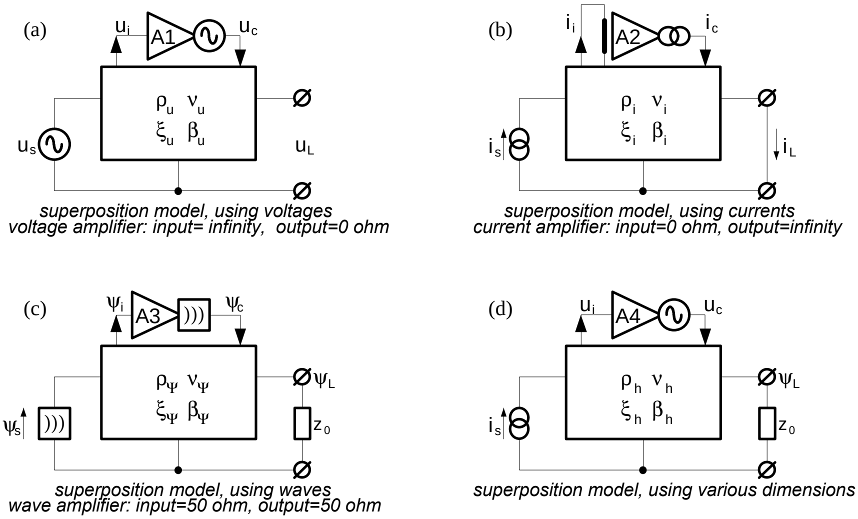
This approach yields the superposition *model*, and the block diagrams in figure 4.4 show different examples of this approach. The concept 'ideal' is closely related with the chosen dimension of the various signals. This means that output impedance of ideal sources is *zero* for a voltage source, *infinity* for a current source and *matched* for a wave source. Similar, the input impedance of voltage, current or wave probes are respectively infinity, zero or matched.

Figure 4.4a shows an example of the superposition model, in the special case that all signals of interest are voltages. As a result, it includes: an ideal voltage source with *zero output impedance* to generate q_s, an ideal voltage probe with *infinity input impedance* to sense q_L, and an ideal voltage to voltage amplifier with *infinity input impedance* to sense q_i and *zero output impedance* to generate q_c. All impedances that makes the actual source, load and amplifier non-ideal are embedded in a common feedback blackbox.

Figure 4.4b shows an example of the same feedback amplifier, in the case that the current is analyzed instead of the voltage. Using proper Thevenin to Norton transformations the physical source, load and amplifier are modified into ideal current sources and probes. As a result, it includes an ideal current source with *infinity output impedance* to generate q_s, an ideal current probe with *zero input impedance* to sense q_L, and an ideal current to current amplifier with *zero input impedance* to sense q_i and

[2] The names that are chosen in this thesis are reconciled with their physical interpretation. Alternative names that were found in the literature [406] for the transfer functions A, ε and ν are *reference variable, input signal loss* and *output signal loss*.

infinity output impedance to generate q_c. All remaining impedances reside in a common feedback blackbox, which differs from that in figure 4.4a.



superposition model, using voltages
voltage amplifier: input= infinity, output=0 ohm

superposition model, using currents
current amplifier: input=0 ohm, output=infinity

superposition model, using waves
wave amplifier: input=50 ohm, output=50 ohm

superposition model, using various dimensions

Fig 4.20 *Examples of different implementations of the superposition model of feedback amplifiers, using signals with different dimensions. In these block diagrams all sources, probes and amplifiers are ideal by definition. This means, for instance, that A2 is an ideal current amplifier with zero input impedance and infinity output impedance. The four-port represents the feedback blackbox, in which all remaining components of the circuit reside.*

Figure 4.4c shows an example of the same feedback amplifier, in the case that all signals of interest are waves. Similar to the example in figure 4.4b, the source load and amplifier are transformed to ideal components that are perfectly matched to a chosen Z_0 (e.g. 50Ω).
In the example of figure 4.4d, all signals of interest have different dimensions.

It is irrelevant whether the forward amplifier is a single amplifier stage or a cascade of stages. The superposition feedback analysis models this cascade by an ideal forward amplifier surrounded by input, output and shunted impedances. It then extracts these impedances from the forward amplifier and embeds them in the feedback network. All remaining circuit components reside in a common multi-terminal network, the *feedback network*.

4.1.2. Superposition flow parameters

The superposition model originates from well-considered mathematical manipulations with the feedback circuit, however that approach estranges the model of a physical interpretation. The superposition flow diagrams in figure 4.5 represent the equations of the superposition model in a graphic way to relax the interpretation of the superposition parameters.

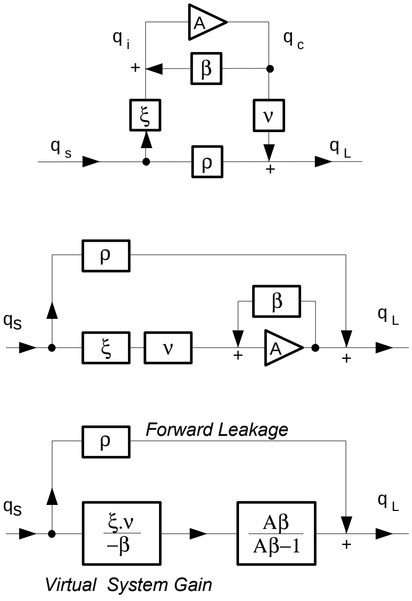


Fig 4.20 *Three identical flow diagrams of the superposition model. Some transfer functions are grouped to emphasize their influence on the overall result.*

We define some additional transfer functions, and their associated names, because their physical interpretation is meaningful. The last flow diagram in figure 4.5 illustrates the meaning of these superposition *flow* parameters.

- The *loop gain* $(A\beta)$ is a measure for the transfer of all amplifier stages and feedback attenuation in the feedback loop.
- The *virtual system gain* $(-\xi\cdot\nu/\beta)$ is a measure for the system gain in the case of zero leakage $(\rho\rightarrow 0)$ and infinity loop gain $(A\beta\rightarrow\infty)$. When the loop gain reduces below the unity gain value, then the virtual system gain differs significantly from the realized system gain.
- The *asymptotic system gain* $(\rho-\xi\cdot\nu/\beta)$ is a measure comparative to the virtual system gain, without neglecting the forward leakage.
- The *realized system gain* is the transfer that is actually achieved.

The *forward leakage* (ρ) is the superposition *model* parameter that was defined in section 4.1.1. It is a measure for the isolation between input and output. When the forward gain reduces to zero $(A\rightarrow 0)$, then the gain of the overall system becomes equal to this leakage value.

loop gain:	$\mathring{H}$	$= A\cdot\beta$
forward leakage	A_{t0}	$= \rho$
virtual system gain	$A_{v\infty}$	$= \; -(\xi\cdot\nu/\beta)$
asymptotic system gain	$A_{t\infty}$	$= \rho - (\xi\cdot\nu/\beta)$

realized system gain:

$$A_t = (A_{t\infty} - A_{t0}) \cdot \left(\frac{A\beta}{A\beta - 1}\right) + A_{t0} \approx A_{t\infty} \cdot \left(\frac{A\beta}{A\beta - 1}\right)$$

Fig 4.6 *Extraction of superposition flow-parameters from y-parameters.*

The superposition model is an exact representation of the feedback amplifier, and does not neglect parasitic effects. It is not required to embed *all* active devices in the forward gain block. Further, the choice of the dimension of the input quantity and the choice of the controlled quantity is arbitrary.

In the special case of a multi loop feedback configuration, where the controlled amplifier has more than two output nodes, it is sufficient to embed a *few* active devices in the forward gain block and to embed all remaining active and passive devices in the multi terminal feedback blackbox. As a result, the choice of the superposition parameters is ambiguous.

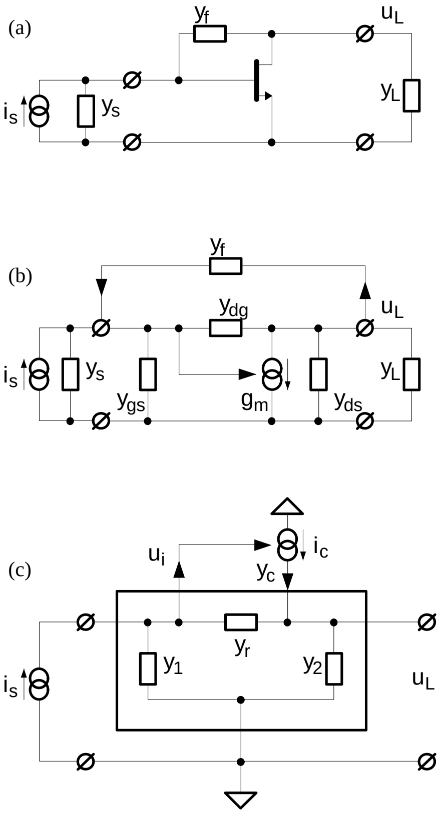
This will be illustrated with an example. Consider a situation in which the input and output nodes of the forward gain block are chosen. Then a variation in the dimension of the chosen input and controlled quantity will not affect the forward leakage, the loop gain and the virtual system gain. To prove this, change the input quantity q_i to $q'_i = \alpha_i\cdot q_i$ and change the controlled quantity q_c to $q'_c = \alpha_c\cdot q_c$. Then the modified superposition parameters are related as follow:

$q_c = q'_c/\alpha_c$	$A = A'\cdot\alpha_c/\alpha_i$	$(A\cdot\beta)$	= *invariant*
$q_i = q'_i/\alpha_i$	$\beta = \beta'\cdot\alpha_i/\alpha_c$		
	$\xi = \xi'\cdot\alpha_i$	$(\xi\cdot\nu/\beta)$	= *invariant*
	$\nu = \nu'/\alpha_c$		
	$\rho = \rho'$	ρ	= *invariant*

Although the choice of the nodes and the dimension of the internal signal is arbitrary from a mathematical point of view, the choice is most handy when all active devices are embedded in the forward amplifier block. This choice minimizes the gap between conventional feedback analysis and superposition feedback analysis.

4.1.3. Example of the extraction of superposition parameters

A simple example how to evaluate the superposition model is draw in figure 4.7. An electronic circuit with transimpedance feedback and a field effect transistor amplifies a current and generates a voltage for a load impedance. The associated superposition model is hybrid because source and load dimension are different.



$$y_c = -g_m$$
$$y_r = y_f + y_{dg}$$
$$y_1 = y_s + y_{gs}$$
$$y_2 = y_l + y_{ds}$$

$$i_s = y_1 \cdot u_i + y_r \cdot (u_L - u_i)$$
$$i_c = y_2 \cdot u_L + y_r \cdot (u_i - u_L)$$
$$i_c = y_c \cdot u_i$$

$$\begin{bmatrix} u_L \\ u_i \end{bmatrix} = \begin{bmatrix} \rho & \nu \\ \xi & \beta \end{bmatrix} \cdot \begin{bmatrix} i_s \\ i_c \end{bmatrix}$$
$$[i_c] = [A] \cdot [u_i]$$

$$\begin{bmatrix} \rho & \nu \\ \xi & \beta \end{bmatrix} = \frac{1}{\Delta} \cdot \begin{bmatrix} 1 & 1+y_1/y_r \\ 1+y_2/y_r & 1 \end{bmatrix}$$
$$A = y_c$$
$$\Delta = y_1 + y_2 + y_1 \cdot y_2/y$$

Fig 4.20 *Example of the evaluation of the superposition model for transimpedance feedback. The electronic circuit (A) is modeled by a linear* P*-network (B). Then, all shunt impedances are combined (C) and some ideal (controlled) sources are isolated from the remaining elements.*

The first step in the evaluation of the superposition model is the replacement of the active device by an appropriate circuit model. The example in figure 4.7b uses a Π-FET model with three impedances and a controlled source to represent the circuit of figure 4.7a.

The next step is to isolate the (controlled) sources from the rest of the circuit and to combine all remaining components in a common blackbox. The example in figure 4.7c combines all shunted impedances to reduce the complexity of the blackbox. The result is a distinct separation between feedback network, forward amplifier and source and load

quantities. Further, the figure summarizes the relation between circuit parameters and superposition parameters.
With some mathematical effort figure 4.7 results in:

$$\boxed{\begin{array}{llll}
A = y_c & & & \\[4pt]
\beta = \dfrac{1}{y_1 + y_2 + y_1\cdot y_2/y_r} & & \dfrac{A\beta}{A\beta-1} = \dfrac{y_c}{y_c-y_1-y_2-y_1\cdot y_2/y_r} & \\[10pt]
A\beta = \dfrac{y_c}{y_1 + y_2 + y_1\cdot y_2/y_r} & & A_{t\infty} = \dfrac{-1}{y_r} & \\[10pt]
\xi\cdot\nu/\beta = \dfrac{(y_r+y_1)\cdot(y_r+y_2)}{y_1 + y_2 + y_1\cdot y_2/y_r} & & A_{t0} = \dfrac{1}{y_1 + y_2 + y_1\cdot y_2/y_r} & \\[10pt]
\rho = \dfrac{1}{y_1 + y_2 + y_1\cdot y_2/y_r} & & A_t = \dfrac{1}{(y_1-y_c)\cdot(y_2+y_r)/(y_c+y_r) + y_2} &
\end{array}}$$

Fig 4.8 *Extraction of superposition flow parameters from y-parameters.*

Note that $A_{t\infty}$ is in this example not equivalent to the desired system gain ($-1/y_f$). To simplify this example, the parasitic feedback impedance y_{dg} (capacitance) is incorporated in the external feedback impedance (y_f). Section 4.2 describes a general applicable method.

4.1.4. Conclusions

In conclusion, the superposition model for an arbitrary feedback amplifier is described in general terms. The dimension of the signal flow of interest makes the superposition model more specific. This has been illustrated for signals such as voltages, currents and waves.

The superposition model originates from a pure mathematical definition, with five *superposition model parameters*. The interpretation of this model is relaxed using flow diagrams, and some transfer functions are grouped to reduce the number of parameters.

The superposition model is an exact representation of the feedback amplifier, however, the superposition parameters are not unique. The choice of the internal signal quantities and their dimensions is decisive. It has been demonstrated that a transformation in internal signal dimension will transform most superposition parameters. Some combined quantities remain invariant under this transformation. This holds for the loop gain ($A\beta$), the virtual system gain ($-\varepsilon\cdot\nu/\beta$) and the forward leakage (ρ).

The evaluation of all superposition parameters of a simple transimpedance amplifier has been demonstrated as example.

4.2. Superposition parameter calculation

The previous section described the superposition model in general terms. Feedback analysis, using the superposition model, requires the evaluation of all relevant superposition parameters for arbitrary feedback amplifier configurations.

Nordholt [406] has solved this by simplifying the analysis to the asymptotic gain model. Since this asymptotic gain model ignores the forward leakage (ρ), it is inadequate for wideband feedback design. The previous section described a restricted solution for a specific configuration.

This study has resulted in a generalized method to extract all superposition parameters, using N-port techniques. The feedback circuit is split in two sub circuits. The associated N-port parameters are evaluated with a circuit simulator and represented in a y-parameter format. The various superposition parameters are calculated using these y-parameters. This section describes the proposed generalized method.

4.2.1. Calculation of loop gain, forward gain and feedback factor

An arbitrary feedback circuit may contain one or more active devices, and thus may contain several non-ideal controlled sources with (parasitic) local feedback. From a mathematical point of view it is correct to choose one of these controlled sources as forward amplifier. The other controlled sources can be placed among the components in the feedback network. This emphasizes that the feedback network is not confined to passive components only; active components are also allowed.

From a physical point of view, an arbitrary choice is a very inconvenient approach. A clumsy choice for A and β will be compensated by inconvenient conversion values for ε and ν. However, this approach estranges the model from its traditional physical meaning. The best choice, from a physical point of view, causes unity conversion at input and output (ε=1, ν=1, both with the associated dimension) and an equal dimension for forward gain (A) and system gain (A_t).

An example of an inconvenient choice for forward gain is the use of the controlled source in a *single* amplifier stage of a multi stage loop.

One of the highlights of this study is the idea that an arbitrary two-port is fully characterized by a virtual circuit with four elements. This was discussed in detail in section 2.3.2 when introducing virtual circuit parameters. Figure 4.9 shows an elaboration of this approach, that is more suitable for physical interpretation than methods that isolate the controlled source in a *single* amplifier stage.

The cascade of amplifier stages is represented by a two-port network, and the associated two-port y-parameters facilitate the extraction of the four elements of an equivalent circuit. As a result, the extracted source represents a cascade of various active devices with a *single* controlled source. This approach facilitates the evaluation of the superposition parameters in a way, similar to the example of subsection 4.1.3.

The method proposed here can be implemented simply in a circuit simulator that is conversant with N-ports as a compound circuit element.

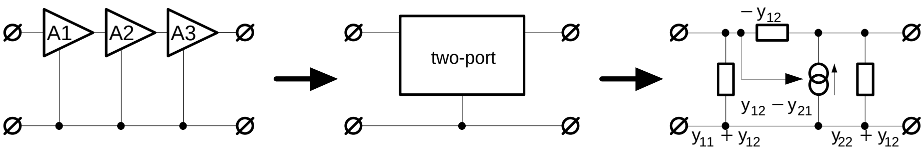
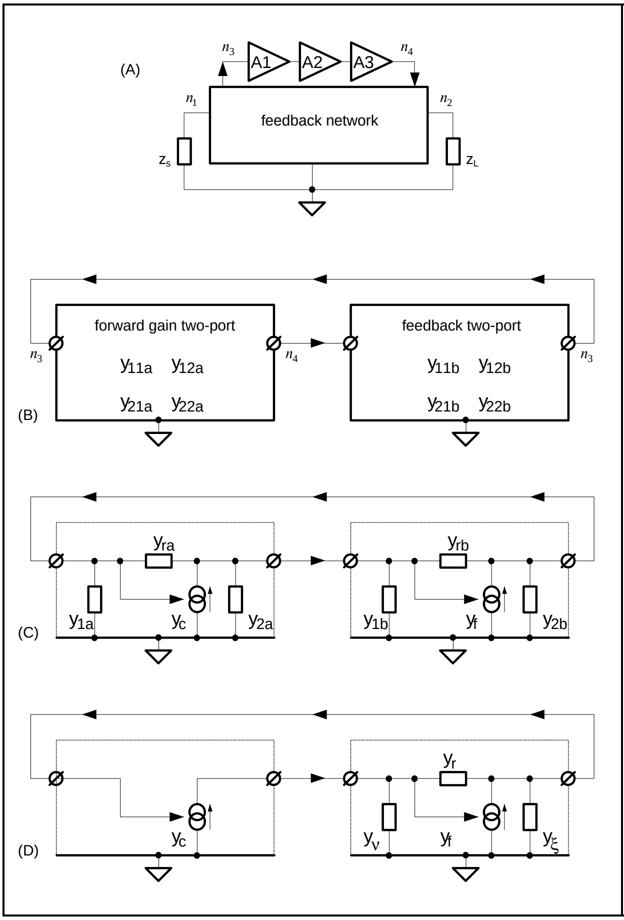


***Fig 4.20** The ideal controlled source, embedded in a cascade of amplifier stages, is de-embedded by a transformation of two-port Y-parameters into* P*-model-parameters. Note that the circuit representation of the* P*-model-parameters holds for any arbitrary two-port and is not restricted to FET's.*

Figure 4.10 shows a more general approach for the loop gain calculation with a circuit simulator. Since loop gain Aβ is dimensionless and independent on the chosen dimension of A or β, the dimension of transconductance is used as a temporary choice.

- The simulation of an arbitrary circuit starts with defining a path of forward signal flow and isolates four sub-circuits, as shown in figure 4.10a. These sub-circuits are (1) source impedance, (2) load impedance, (3) a four-port feedback network and (4) a two-port forward amplifier. The multi-port parameters of each sub-circuit are subsequently evaluated with a circuit simulator, for instance as described in section 2.2.4.
- Then, the multi-port parameters of sub-circuit 1, 2 and 3 are combined and reduced for extracting additional two-port parameters with node n_4 as input port node and node n_3 as output port node. As is shown in figure 4.10b, the circuit has been split into two networks: one forward gain two-port and a reduced feedback two-port. The required algorithms are discussed in section 2.2.3 and 2.2.4.
- Next, these two-port parameters are transformed in (Π model) virtual circuit parameters for generating the virtual circuits as shown in figure 4.10c. The required algorithms are discussed in section 2.3.2.
- Subsequently, the simulation 'moves' the equivalent passive circuit elements from the forward gain two-port to the feedback two-port for de-embedding the controlled source parameter y_c. Figure 4.10d shows how the equivalent loop model is simplified using this approach for representing the feedback loop in a standard format.
- The standard format of the feedback network facilitates the calculation of the superposition parameters and the loop gain in a simple way, similar to the example in section 4.1.3. The loop gain is calculated similar to the methods discussed for figure 4.7.



$$y_{1a} = y_{11a} + y_{12a}$$
$$y_{2a} = y_{22a} + y_{12a}$$
$$y_c = y_{12a} - y_{21a}$$
$$y_{ra} = -y_{12a}$$

$$y_{1b} = y_{11b} + y_{12b}$$
$$y_{2b} = y_{22b} + y_{12b}$$
$$y_f = y_{12b} - y_{21b}$$
$$y_{rb} = -y_{12b}$$

$$y_\xi = y_{1a} + y_{2b}$$
$$y_v = y_{2a} + y_{1b}$$
$$y_r = y_{ra} + y_{rb}$$

Fig 4.20 *Intermediate calculation steps for the evaluation of the loop gain A*b. *The source and load impedances are enclosed in the feedback blackbox (A), and both the amplifier blackbox and the feedback blackbox are represented by two-port networks (B). The y-parameters of these two-ports are transformed into virtual circuit-parameters (C), (see section 2.3.2). The equivalent input, output and shunt impedances are stripped from the amplifier and combined with the feedback network (D). The loop gain is finally evaluated from this simplified model.*

- Next, the dimension of the forward gain is chosen, which is most meaningful when its dimension equals that of the realized system gain. The default choice in figure 4.10 is [1/Ω] because of the voltage controlled current source y_c.
  The transformation of the forward amplifier into another controlled source requires additional impedance at input and/or output. The input admittance y_{1a} and the output admittance y_{2a} are appropriate default choices. The forward gain A is one of the following list:

voltage to current: $A_y = y_c$
current to current: $A_i = y_c/y_{1a}$
voltage to voltage: $A_u = y_c/y_{2a}$
current to voltage: $A_z = y_c/(y_{1a}\cdot y_{2a})$

- Finally, the feedback factor β is determined by the quotient of loop gain and forward gain.

The above procedure is suitable for manual calculation however it is in its current form, with equivalent circuit elements, too cumbrous to implement in a circuit simulator. More appropriate is a direct evaluation of the required quantities from the two-port parameters, for instance from the y-parameters. With some mathematical effort the following expressions can be obtained for loop gain and some basic forms of the forward gain:

forward gain (I/U)	$A_y = y_c$	$y_\xi = y_{11a}+y_{22b}+y_{12a}+y_{12b}$
forward gain (I/I)	$A_i = y_c/y_{1a}$	$y_\nu = y_{11b}+y_{22a}+y_{12a}+y_{12b}$
forward gain (U/U)	$A_u = y_c/y_{2a}$	$y_r = -(y_{12a} + y_{12b})$
forward gain (U/I)	$A_z = y_c/(y_{1a}\cdot y_{2a})$	$y_c = y_{12a} - y_{21a}$
		$y_f = y_{12b} - y_{21b}$
loop gain	$A\beta = \dfrac{y_c\cdot(y_f+y_r)}{y_\xi\cdot y_\nu + y_r\cdot(y_\xi+y_\nu-y_f)}$	$y_{1a} = y_{11a} + y_{12a}$
		$y_{2a} = y_{22a} + y_{12a}$

Fig 4.11 *Extraction of loop gain and forward gain from y-parameters.*

The previous approach is valid for grounded amplifier stages only. In the case that the input or output amplifier stage has floating external nodes, then the N-port description of the forward amplifier requires at least three ports. These configurations are not considered here, however the loop gain has to be calculated in a similar way.

4.2.2. Calculation of asymptotic and virtual gain and forward leakage

The asymptotic system gain is the system gain under the hypothetical situation that the loop gain goes to infinity ($\mathring{H}=A\beta\rightarrow\infty$). The same applies for the forward leakage, under the situation that the loop gain goes to zero ($\mathring{H}=A\beta\rightarrow 0$). The value of realized system gain is somewhere between these extreme values. The relation between loop gain variation γ and associated variable system gain $A_t(\gamma)$ is as follows:

variable system gain: $$A_t(\gamma) = (A_{t\infty}-A_{t0})\cdot\left(\frac{\delta\cdot\mathring{H}}{\delta\cdot\mathring{H}-1}\right) + A_{t0}$$

Since the loop gain $\mathring{H}$ is known from previous calculations, the evaluation of these transfer functions is feasible by system gain calculations under variation of loop gain.

forward leakage: $A_{t0} = A_t(0)$

asymptotic system gain: $A_{t\infty} = (A_t(1) - A_t(0)) \cdot \left(\frac{\mathring{H}-1}{\mathring{H}}\right) + A_t(0)$

virtual system gain: $A_{v\infty} = A_t(1) - A_t(0)$

***Fig 4.12** Extraction of forward leakage, asymptotic system gain and virtual system gain from y-parameters.*

Arithmetic variation of loop gain requires the isolation of the forward amplifier from the circuit as a two-port. Variation of the transconductance y_c of the equivalent circuit representation of this two-port is similar to the variation of loop gain.

A successful mathematical approach for circuit simulators is the evaluation of the y-parameters of the two-port, the modification of the y_{21} parameter, and then the evaluation of the overall system transfer with this modified two-port. Linear variation of loop gain requires a variation of $y'_{21} = \gamma \cdot y_{21} - (1-\gamma) \cdot y_{12}$.

Figure 4.13 shows the circuit equivalent of this procedure. Forward leakage calculation requires a modification of $y'_{21} = y_{12}$ to perform zero loop gain.

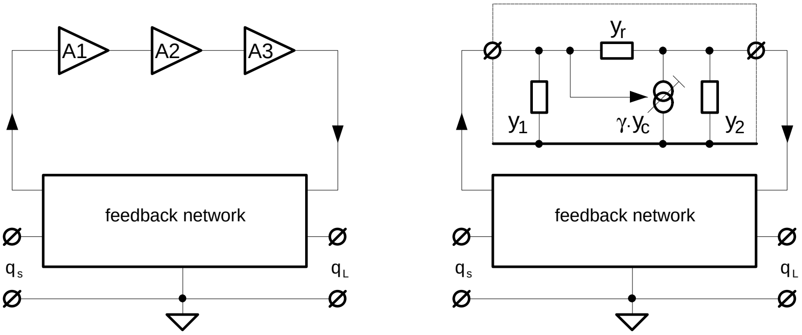


***Fig 4.20** Arithmetic variation of loop gain by modification of the* y_{21} *parameter of the cascade of amplifier stages. When the controlled source transfer* y_c *has to change to* $y_c' = \gamma \cdot y_c$*, then the* y_{21} *parameter has to change to* $y_{21}' = \gamma \cdot y_{21} - (1-\gamma) \cdot y_{12}$

4.2.3. Conclusions

In conclusion, a generalized method is demonstrated for extracting all superposition parameters from arbitrary feedback circuits. The application of the proposed method is primarily intended for usage with circuit simulators, equipped with N-port facilities.

The designer splits the circuit in two sub-circuits and defines the dimension of the internal and external signals. The simulator provides the five superposition model parameters.

4.3. Loop gain estimation using single-cut methods

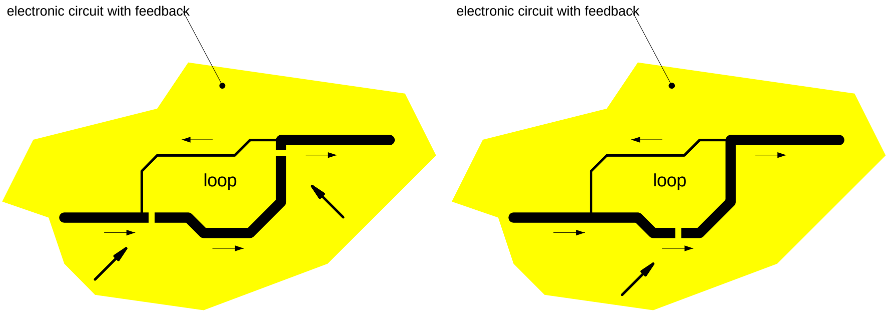
An *exact* superposition analysis requires an exact calculation of the superposition parameters. The wider the passband of the amplifier, the more the analysis must rely on exact values. When future circuit simulators give full support to superposition analysis, then exact analysis is always preferred.

Sometimes, an *estimated* superposition analysis is useful, for instance when the relevant parameters are directly obtained from two-port measurements. It requires that internal nodes of the circuit under test are made available for external measurement, such as a modified circuit with opened feedback loop. Further, estimated analysis might be useful to obtain global insight on stability performance with minimal calculation effort. This holds especially when appropriated (software) tools are not available.

Estimated feedback analysis is primarily focused on loop gain calculation since most stability problems are mainly related on loop gain. It may serve as an intermediate step between conventional analysis and exact superposition analysis.

This section describes mathematical methods to estimate the loop gain. This estimated result will be referred in this text by the name: *open* loop gain.

Further, this section links the exact definition of loop gain with the estimated methods that are proposed here and with conventional loop gain estimations. This facilitates the assessment of errors originating from open loop gain and conventional methods.


electronic circuit with feedback
loop
electronic circuit with feedback
loop

two-cut method, for loop gain calculation at begin and end of forward amplifier

one-cut method, for loop gain estimation at an arbitrary location in the loop

Fig 4.20 *Signal flow in a feedback amplifier. Two cuts in the loop will isolate the forward amplifier from feedback network and facilitate an exact calculation of the loop gain. With a single cut in the loop, the determination of the loop gain is restricted to an approximation.*

Figure 4.14 illustrates the basic idea, on the basis of signal flow in an arbitrary feedback amplifier.

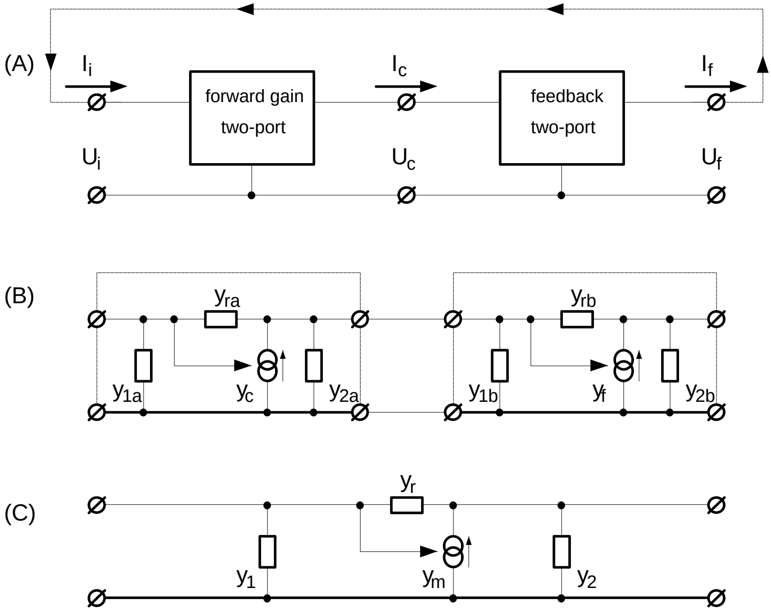
With two cuts at the begin and end of the forward amplifier, this amplifier is isolated from the feedback network, which facilitates an exact loop gain calculation. Two cuts in the loop yields two individual two-ports and eight two-port parameters. An exact loop gain calculation, make use of all these parameters.

A single cut yields an single (open) loop two-port and four two-port parameters. This restricts the calculated result to a fair estimation of loop gain. It will be demonstrated that this estimation is independent of the position of the cut.
Although two-cut methods are preferred, due to their accuracy, one-cut methods are sometimes attractive because of their simplicity.

4.3.1. One-cut loop gain estimation using two-port parameters

Single cut methods start from the assumption that the transfer of the forward amplifier is close to unilateral at all frequencies of interest. This means that the forward transfer (gain) of that two-port is significantly higher than the reverse transfer (isolation). Note that this condition is not required for the *individual* amplifier stages; the more stages are used in the overall cascade, the better the overall transfer is unilateral.
The concept of *sufficient* isolation of the forward amplifier is related to the transfer of the feedback network. The feedback must dominate the isolation. In general, the more stages are used in the overall cascade, the lower the feedback is and the higher the isolation demands will be.
The higher the frequency of interest, the lower the isolation will be. As a result, the application of one-cut loop gain methods are mainly restricted to frequencies that are significantly lower than transition frequencies f_T of the transistors.
Figure 4.15 shows various representations of the opened loop.

Figure 4.15a shows the result from the two-cut method. Forward amplifier two-port and the feedback network are isolated from the circuit. Because the source and load impedance are embedded in the feedback network, it is represented here as two-port. Both two-ports represent the loop.
Figure 4.15b shows the two-port parameters using a virtual circuit representation. The element values are extracted from the Π-model virtual circuit-parameters, as was previously described in section 2.3.2. When all eight numbers from both two-ports are available, then an exact loop gain calculation is feasible.
Figure 4.15c shows the cascade of both two-ports using a virtual circuit representation. This two-port is the open loop two-port and the element values are derived from the two-port parameters from figure 4.15b.



$y_1 = y_{1a} + y_{ra}\cdot(y_v - y_c)/(y_v + y_r)$ $\qquad$ $y_\xi = y_{1a} + y_{2b}$

$y_2 = y_{2b} + y_{rb}\cdot(y_v - y_c)/(y_v + y_r)$ $\qquad$ $y_v = y_{2a} + y_{1b}$

$y_{rc} = (y_{ra}\cdot y_{rb})/(y_v + y_r)$ $\qquad$ $y_r = y_{ra} + y_{rb}$

$y_m = (y_c\cdot(y_f + y_r) - y_{ra}(y_c - y_f)) / (y_v + y_r)$

Fig 4.20 *Approximation of the loop gain $\mathring{H}=A$*b *by the open loop gain. If the isolation of the forward amplifier is high (y_{ra}*fi *0) then the open loop gain is close to the loop gain.*

definition of open loop gain

The evaluation of the loop gain $\mathring{H}$, from the individual two-port parameters of forward amplifier and feedback network, was derived section 4.2.1. We have found a fair estimation of $\mathring{H}$, which is defined as $\overset{\ominus}{H} \overset{\text{def}}{=} (y_m)/(y_1+y_2)$, that is evaluated from the open loop two-port parameters. It can be derived from the equations in figure 4.15 that both quantities are related as follows:

$$\textit{loop gain:} \qquad \mathring{H} = \frac{y_c\cdot(y_f+y_r)}{y_\xi\cdot y_v + y_r\cdot(y_\xi+y_v-y_f)}$$

$$\textit{open loop gain:} \qquad \overset{\ominus}{H} \overset{\text{def}}{=} \frac{y_c\cdot(y_f+y_r) - \delta}{y_\xi\cdot y_v + y_r\cdot(y_\xi+y_v-y_f) - \delta} = \frac{y_m}{y_1+y_2}$$

$$\textit{with } \delta = y_{ra}\cdot(y_c - y_f)$$

When y_{ra} is small enough to ignore δ then the *open loop gain* $\overset{\ominus}{H}$ is close to the *loop gain* $\mathring{H}$.

relation with other two-port parameters

The definition of the open loop gain is based on the virtual circuit parameters (y_m, y_1, y_2). However these two-port parameters are not always direct available. In practice, it is often handy to use s-parameters for measurement and y-parameters for circuit simulation, however any two-port parameter set applies. Therefore it is convenient to evaluate the open loop gain directly from arbitrary two-port parameters.

Using the equations of figure 4.15, and the two-port parameter definitions of section 2.3.1, it can be demonstrated that the following relations hold for $\overset{\ominus}{H}$:

open loop gain for various two-port parameters:

$\overset{\ominus}{H} = \frac{y_m}{y_1+y_2}$	$\overset{\ominus}{H} = \frac{h_{21}+h_{12}}{2\cdot h_{12}-1-\Delta_h}$	$\overset{\ominus}{H} = \frac{s_{12}-s_{21}}{2\cdot s_{12}+\Delta_s-1}$
$\overset{\ominus}{H} = \frac{y_{12}-y_{21}}{y_{11}+y_{22}+2\cdot y_{12}}$	$\overset{\ominus}{H} = \frac{1-\Delta_a}{a_{11}+a_{22}-2\cdot\Delta_a}$	$\overset{\ominus}{H} = \frac{1-\Delta_t}{t_{11}+t_{22}-2\cdot\Delta_t}$
$\overset{\ominus}{H} = \frac{z_{21}-z_{12}}{z_{11}+z_{22}-2\cdot z_{12}}$	$\overset{\ominus}{H} = \frac{1-\Delta_{abcd}}{A+D-2\cdot\Delta_{abcd}}$	

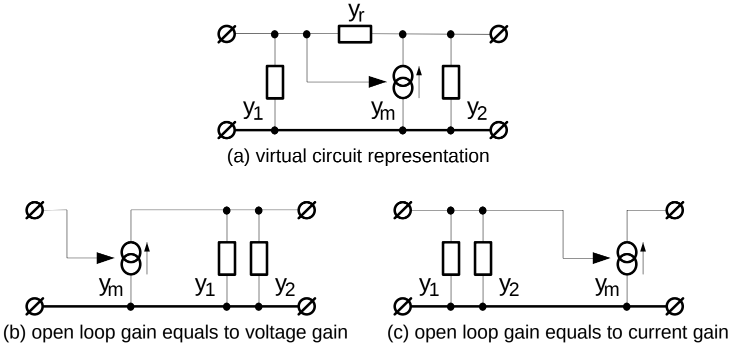
$\Delta_a = a_{11}\cdot a_{22}-a_{12}\cdot a_{21}$ $\Delta_s = s_{11}\cdot s_{22}-s_{12}\cdot s_{21}$

$\Delta_h = h_{11}\cdot h_{22}-h_{12}\cdot h_{21}$ $\Delta_t = t_{11}\cdot t_{22} - t_{12}\cdot t_{21}$

$\Delta_{abcd} = A\cdot D-B\cdot C$

relation with current gain and voltage gain

The open loop gain $\overset{\ominus}{H}$ is a transfer function somewhere in-between current gain H_i and voltage gain H_u. Figure 4.16 demonstrates this by means of three circuits with identical open loop gain. Figure 4.16a is the virtual circuit representation of an open loop two-port. Figure 4.16b and 4.16c are different two-ports, but have in common that the open loop gain is equal for all three two-ports. The major difference is that the input impedance is moved to the output (or reverse), and that the shunt impedance is removed. The current gain for the two-port in figure 4.16b equals the open loop gain, while the same applies for the voltage gain in figure 4.16c. As a result, when $y_1 \approx 0$ then $\overset{\ominus}{H}$ approaches the voltage gain and when $y_2 \approx 0$ then $\overset{\ominus}{H}$ approaches the current gain.



(a) virtual circuit representation

(b) open loop gain equals to voltage gain

(c) open loop gain equals to current gain

Fig 4.16 *Three circuits with identical open loop gain.*

position invariance

The location of the loop cut is irrelevant, as long as the cyclic order remains unchanged. This means that the open loop gain of a two-port cascade with order (1,2,3,4) equals that of the cascade with order (2,3,4,1) or (3,4,1,2) or (4,1,2,3). As a result, the value of the open loop gain is unique for a cascaded loop of arbitrary two-ports, and invariant of the cut location.

The property that open loop gain is *position invariant* is simply verified with a matrix representation of the individual two-ports. The simplest verification uses t-parameters, because the T-matrix of a cascade of two-port is equal to matrix product of all individual T-matrices. When the matrices T_a, T_b and T_c are representations of arbitrary two-ports, then the open loop gain of $T_a \cdot T_b \cdot T_c$ equals both the open loop gains of $T_b \cdot T_c \cdot T_a$ and $T_c \cdot T_a \cdot T_b$. This property holds for arbitrary t-parameter sets and does not require a unilateral behavior of the individual two-ports.

Figure 4.17 illustrates the benefits of this position invariance. The loop is cut at an arbitrary node in the signal path through the amplifier stages. The impedance value of input and output impedance of the separated amplifier stages is irrelevant.

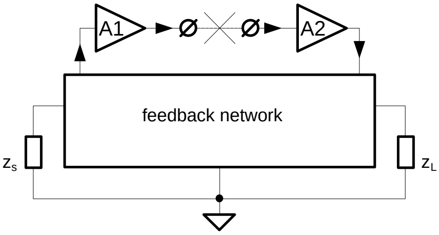


Fig 4.20 *Approximation of the loop gain $\overset{\circ}{H}=A$b by the measurement of the open loop gain $\overset{\ominus}{H}$. The loop is opened at an arbitrary location, and $\overset{\ominus}{H}$ is reconstructed from the measured two-port parameters. Note that $\overset{\ominus}{H}$ is a unique parameter of a loop of cascaded two-ports, as is described in the text.*

4.3.2. One-cut loop gain estimation using open voltage and current gain

One-cut loop gain estimations, using two-port parameters, are can be implemented simply in circuit simulators with two-port parameter facilities. This result in the open loop gain, that is an approximation for the exact loop gain. Manual loop gain calculation methods prefer the use of currents and voltages. Commonly used manual methods evaluate the open loop *voltage* gain or the open loop *current* gain, which are approximations of the open loop gain.

This subsection relates the open loop gain with open loop *voltage* gain and the open loop *current* gain, to assess whether these approximations are applicable or not.

Tuinenga [407: chapter 6.9] described an elegant way to define the open loop gain for voltages and for currents. The basic principle is outlined in figure 4.18. An external source injects a signal in the loop and a voltage or current ratio meter senses the injected signals. The current ratio H_i=i_y/i_x is referred in this text as *open loop current-gain* and the voltage ratio H_u.=u_y/u_x as *open loop voltage-gain*.

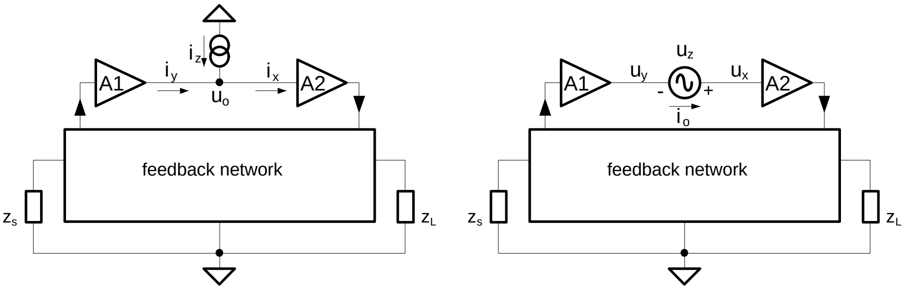


Fig 4.20 *Approximation of the open loop gain $\overset{\ominus}{H}$ by measurement of the open loop current-gain H_i=i_y/i_x or the open loop voltage-gain H_{u}.=u_y/u_x*

The calculation of the injected signal ratio of the open loop two-port is facilitated by a y-parameter representation. Since these parameters (y_{11}, y_{21}, y_{12}, y_{22}) are related with the virtual circuit parameters (y_m, y_1, y_2, y_r) of figure 4.15t can be demonstrated that the following conditions hold:

$$\textit{open loop current gain}\qquad \begin{bmatrix} i_x \\ -i_y \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \cdot \begin{bmatrix} u_0 \\ u_0 \end{bmatrix} \Rightarrow H_i \overset{\text{def}}{=} \frac{i_y}{i_x} = -\frac{y_{22}+y_{21}}{y_{11}+y_{12}} = \frac{y_m-y_2}{y_1}$$

$$\textit{open loop voltage gain}\qquad \begin{bmatrix} i_0 \\ -i_0 \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \cdot \begin{bmatrix} u_x \\ u_y \end{bmatrix} \Rightarrow H_u \overset{\text{def}}{=} \frac{u_y}{u_x} = -\frac{y_{11}+y_{21}}{y_{22}+y_{12}} = \frac{y_m-y_1}{y_2}$$

The value of the reverse isolation y_r= $-y_{12}$ is irrelevant in this measurement setup, because the associated feedback impedance is virtually shorted during the measurements. The current or voltage ratio approaches the open loop gain $\overset{\ominus}{H}$ when the magnitudes of the ratio (y_1/y_2) differ significantly from one. It can be demonstrated that H_i and H_u are as follows related to $\overset{\ominus}{H}$:

$$H_i = \overset{\ominus}{H} + (y_2/y_1)\cdot(\overset{\ominus}{H}-1) \qquad \textit{if} \quad |y_2/y_1| \ll 1 \quad \textit{then} \quad H_i \approx \overset{\ominus}{H}$$

$$H_u = \overset{\ominus}{H} + (y_1/y_2)\cdot(\overset{\ominus}{H}-1) \qquad \textit{if} \quad |y_2/y_1| \gg 1 \quad \textit{then} \quad H_u \approx \overset{\ominus}{H}$$

Both ratio measurements are poor approximations of the open loop gain, however combining H_i and H_u facilitates the evaluation of the correct value. It can be demonstrated that the following relation holds:

$$\boxed{\overset{\ominus}{H} = \frac{H_i \cdot H_u - 1}{H_i + H_u - 2}}$$

The open loop current gain or open loop voltage gain dominates in the above expression when the magnitudes of input and output impedances are significantly unbalanced. The *lowest* of these two approximates the open loop gain $\overset{\ominus}{\mathrm{H}}$.
Current gain approximations yield fair results, for instance, for cascaded bipolar transistors in common emitter configuration. The output impedance of each amplifier stage is relatively high compared to the input impedance of the succeeding stage, so that most of the available output current flows into the succeeding stage.
Voltage gain approximations yield fair results, for instance, to simple opamp configurations, in which the input impedance of the opamp is relatively high and the output impedance relatively low.

Note that the current gain of a two-port is quite different from the *open loop* current gain of that two-port. The current gain of a two-port is the ratio between output current and input current, under the condition that the input current is generated using an ideal current source and that the output current is sensed when it is shorted. The same applies for voltage gain and *open loop* voltage gain.
To evaluate the open loop gain from voltage gain and current gain, an additional 'measurement' is required to complete the missing information.

4.3.3. Conclusions

In conclusion, it has been demonstrated that two-cut methods are required to evaluate an exact solution for the loop gain, however one-cut methods provides fair estimations for this exact value. This estimated result is referred as *open loop gain,* and is a characteristic value of two-ports. The open loop gain function is evaluated for various types of two-port parameters.
Further, it has been demonstrated that the open loop gain is independent of the position of the cut. This is a remarkable property because the two-port parameters of the open loop gain two-port change with the position of the cut. It holds for any cascade of two-ports, as long as their cyclic order remains unchanged.
The accuracy of the open loop gain, as approximation for the (exact) loop gain, is higher that that of the open loop *current* gain or the open loop *voltage* gain. These two quantities are representative for the results of commonly used manual loop gain calculations. The relation between all these quantities facilitates a validation assessment of conventional analysis methods.
Two-cut methods are generally preferred, since they provide exact solutions. However, the simplicity of one-cut methods may be decisive in situations with simple bandwidth demands.

4.4. Loop gain representation by poles and zeros

The loop gain transfer function is an important figure of merit in the assessment of the pass band transfer function. The previous sections demonstrated how to determine loop gain, with sufficient accuracy for reliable wideband feedback analysis.

Stability is an important aspect of the pass band transfer function. The determination of loop gain, in terms of poles and zeros, provides valuable information whether a circuit is oscillating or not.

One of the basic topics of this study is the development of a simple but robust pole-zero extraction algorithm. This iterative curve fitting technique fits rational transfer functions to tabular data. The development of this algorithm was essential to the synthesis of stable wideband feedback loops using pole-zero patterns and root-locus techniques.

This section describes how to extract the poles and zeros of a loop with this algorithm, and demonstrates that higher order pole-zero models are reliable description methods for loop gain transfer functions.

4.4.1. The purpose of pole-zero patterns

Loop gain and system gain are transfer functions that can be determined by measurement or simulation. Network analyzers and tabular[3] circuit simulators generate frequency tables with magnitude and phase information at distinct frequencies of interest for specifying transfer functions.

The graphical presentation of these tables using Bode plots is complete, however stability performance is difficult to interpret. This is because the difference between stable and unstable performance is implied in a subtle difference in the phase plot of a transfer function. Transfer *measurements* on actual systems provide a direct indication of oscillation since the measurement fails in these situations. Transfer *simulation* may result in innocent-looking Bode plots, lacking any warning of oscillation.

Resolving this uncertainty requires the use of additional presentation methods for tabular transfer functions. A loop gain representation, using pole-zero patterns, is more suitable for stability analysis. This facilitates a reliable assessment of stability because the imaginary axis is the arbitration line between stable poles in the LHP (left half plane) and unstable poles in the RHP (right half plane). The drawback of pole-zero plots is that they provide less information about transfer. Therefore, both Bode plots and pole-zero patterns are required.

4.4.2. Strategy of pole-zero extraction methods

The extraction of poles and zeros from a feedback loop is a delicate and a complex process. Manual as well as automated methods may be used to carry out the extraction *simultaneously* or *sequentially*[4] with the evaluation of the loop gain. Several methods are listed below:

[3] The addition 'tabular' is used in this text to distinct tabular simulators from analytical simulator. A tabular simulator provides a transfer function as sampled data and an analytical simulator as coefficients of functions such as polynomials or rational functions.

[4] The subdivision in 'simultaneous' and 'sequential' methods is devised in this text.

- *Manual simultaneous* methods use a cascade of first order estimations; each amplifier stage being modeled by one or at most two poles. This approach simplifies the complexity of the overall problem.
- *Manual sequential* methods calculate Bode plots of the loop gain and then estimate its magnitude asymptotes. From these, they estimate the poles and zeros from the location of the corner frequencies. This approach assumes a minimum-phase transfer function that is consistent with all poles and zeros in the LHP.
- *Automated simultaneous* methods translate the complete circuit netlist to a frequency independent matrix format to describe, for instance, the frequency dependent admittance matrix $\mathbf{Y}=\mathbf{Y}_{\text{re}}+j\omega\cdot\mathbf{Y}_{\text{im}}$. The poles and zeros of the transfer function of interest are directly extracted from $\mathbf{Y}_{\text{re}}$ and $\mathbf{Y}_{\text{im}}$ using eigenvalue calculations. This approach circumvents the evaluation of the complex matrix **Y**, and is implemented in analytical circuit simulators, such as the programs ANP3 [408,409] and LOCUS [113-118].
- *Automated sequential* methods translate the complete circuit netlist to matrix format, however they repeat the translation for any discrete frequency of interest. The transfer function value, associated with that frequency, is evaluated using matrix inversion calculations. Next, it extracts the poles and zeros using a curve-fit to this transfer table. This approach appears to be cumbrous, however it is wider applicable (measured data, elements with delay) and enables a piece wise evaluation of a complex circuit (significant reduction of round-off errors). It is implemented in tabular circuit simulators, such as the commercial available programs Touchstone© [124], MDS© [126] and many variants of Spice [127,128].

It is out of question that the automated methods are more accurate than the manual methods, however the difference in performance between simultaneous and sequential methods is not obvious.

Analytical circuit simulators suffer more from round-off errors then tabular circuit simulators. This is because simple circuits with element models based on delay may generate a huge number of poles and zeros, while most of them are located far outside the frequency band of interest. A tabular circuit simulator models the same element with a single complex number per frequency and remains applicable when analytical circuit simulators fail. Further, the referred tabular circuit simulators facilitate a piece wise evaluation of a complex circuit, using sub circuits.

As a result, it is plausible that automated sequential methods yield better results than automated simultaneous methods when applied to complex circuits.

One of the results of this study was the development of a simple but robust curve fitting algorithm for the extraction of poles and zeros from tabular transfer functions (see section 3.1.1). Appendix *D* describes in detail the mathematics of this method. The linear approach, used in this algorithm, combines fast and robust convergence properties and no need to estimate starting values. The proposed algorithm can be implemented simpler than commonly used iteration methods [412,413] such as Levenberg-Marquardt [410] or Gauss-Newton methods [413].

4.4.3. Practical demonstration of the pole-zero extractor

The robustness of the proposed extraction method extends the application of pole-zero analysis to wideband feedback design, since the extracted singularities are representative of actual circuits. This is demonstrated in figure 4.19, in which three realistic examples of lightwave receivers with third order transimpedance feedback are shown.

The basic form of the amplifier stages in figure 4.19a is a common-emitter (CE) stage, to maximize the loop gain. The active decoupling of the stages in figure 4.19b with common-base (CB) stages improves the reverse isolation of the CE-stages because the parasitic local feedback in the preceding transistor is reduced by the low input impedance of the CB-stage. The combined amplifier stages are called *cascode* stages, and cascoding increases the loop gain and improves the smoothness of its magnitude.

The dynamic decoupling of the stages in figure 4.19c facilitates the use of common-collector (CC) stages. These stages are called *long tailed pairs,* and combine the advantages of cascoding with non-inverting transfer, at the cost of a minor reduction of loop gain.

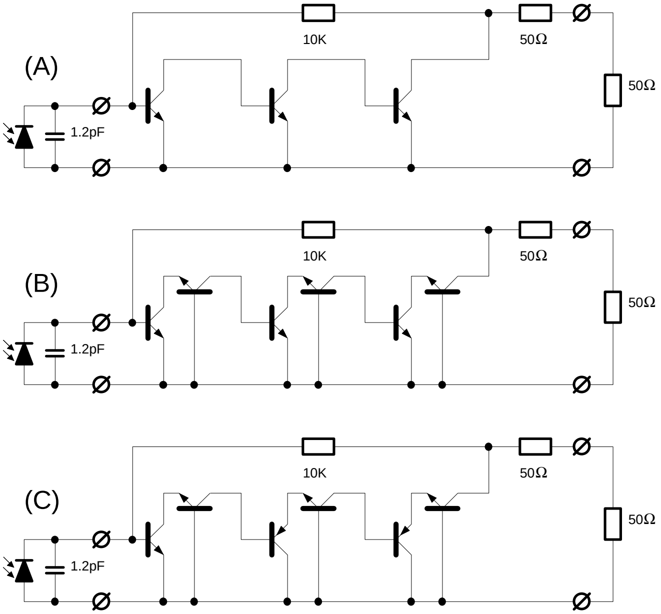


Fig 4.20 *Three examples of a transimpedance feedback amplifier with three cascaded amplifier stages: (a) all CE-stages, (b) all cascoded stages and (c) a cascode stage cascaded with two long-tailed pairs.*

Figure 4.20 shows the calculated loop gain for these examples. Measured two-port s-parameters of the transistors were used in the simulation to avoid the intermediate step of the extraction of a transistor model. This demonstrated the reliability of the pole-zero analysis. The other circuit elements were modeled.

The same s-parameter data set used here was that used for the BJT transistor model extraction in section 3.2. All transistors are assumed to be equal.

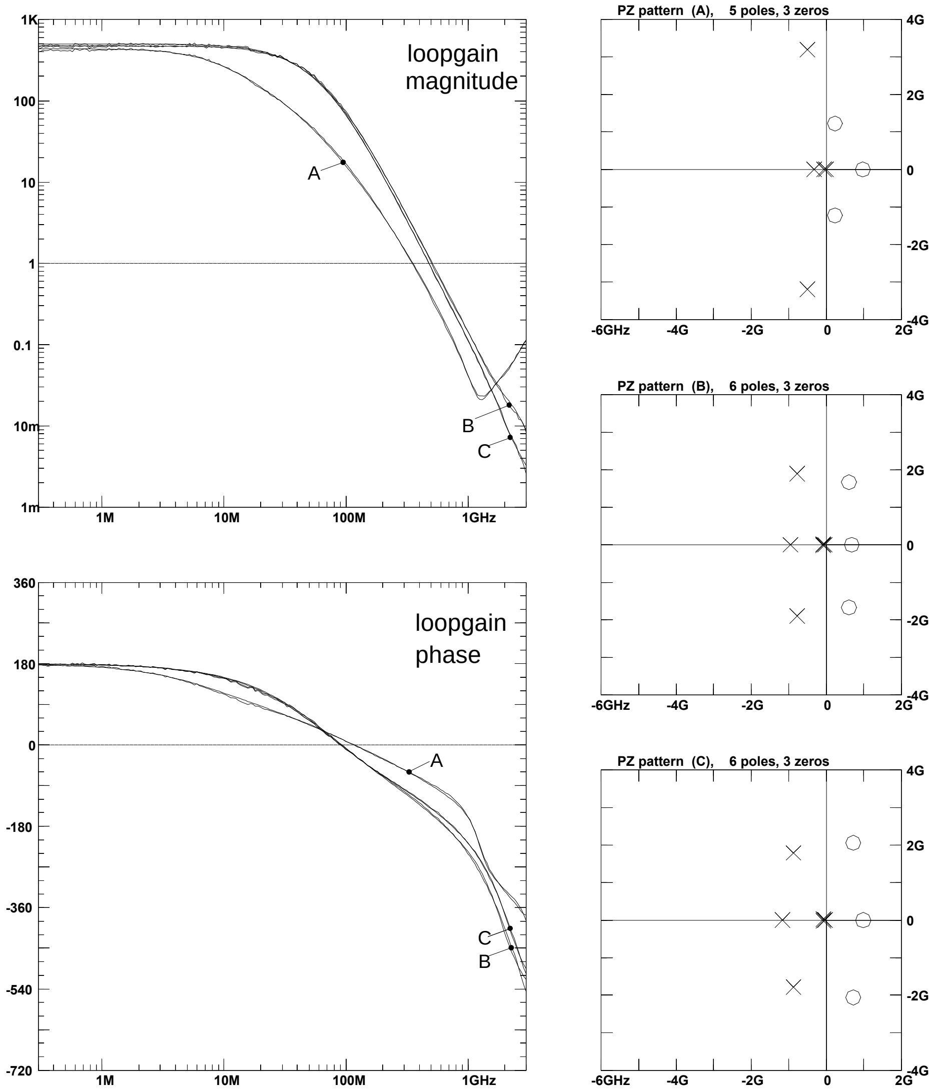


Fig 4.20 Bode plots and pole-zero patterns of the loop gain of the three configurations in figure 4.19. The three data curves in the Bode plots are overlaid with three fitted curves. Note the exceptional fit between model and measured data, while 201 frequency points were used

After the (two-cut) loop gain calculation the simulation fitted three curves of rational functions to the loop gain tables with 201 frequency points each. The choice of the *number* of poles and zeros is a simple process of trial and error, and the algorithm extracts their location. An optimal choice is usually accompanied by a minimum of iteration steps.

The three data curves in the Bode plots of figure 4.20 are overlaid with these fitted curves, and the close match between model and data demonstrates the robustness and the applicability of the method. The associated pole-zero plots in figure 4.20 are therefore accurate representations of the loop gain up to 3 GHz.

One of the most startling observations in figure 4.20 is the presence of RHP-zeros *below* the transition frequency f_T=3.3 GHz of the used transistors. These zeros are important obstacles for the synthesis of stable feedback loops, and are discussed in section 5.3.
These zeros originate from parasitic effects, and are hardly noticed in Bode plots. They illustrate the importance of reliable pole-zero analysis of loop gain, which will be emphasized in the succeeding sections.

4.4.4. Interpretation of pole-zero patterns

This section is completed with an overview of some interpretation rules for pole-zero patterns.

- Poles and zeros determine the corner frequencies of transfer functions. Above this value, the transfer of poles decreases with the frequency while the transfer of zeros increases. Their magnitude defines the value of the corner frequency and their phase distinguishes between stable, resonant or unstable.
- All LHP-poles indicate a stable transfer function. Poles, closer to the imaginary axis than to the real axis, cause a transfer function with resonation peaks.
- One or more poles located in the right complex half plane of the pole-zero pattern indicate that the circuit is oscillating.
- Zeros are always stable. LHP-zeros are the counterpart of LHP-poles. Pairs of poles and zeros at the same location are a neutral pair.
- RHP-zeros are *not* the counterpart of RHP-poles, because they do not affect the stability of the transfer function. Therefore, these zeros cannot be canceled out by poles.

  An alternative is a restricted cancellation by LHP-poles at a mirror position. Mirroring with respect to the imaginary axis causes a complete compensation of the magnitude response, however doubles the phase response.

Figure 4.21 shows some examples of standard pole-zero patterns, with their associated Bode plots.

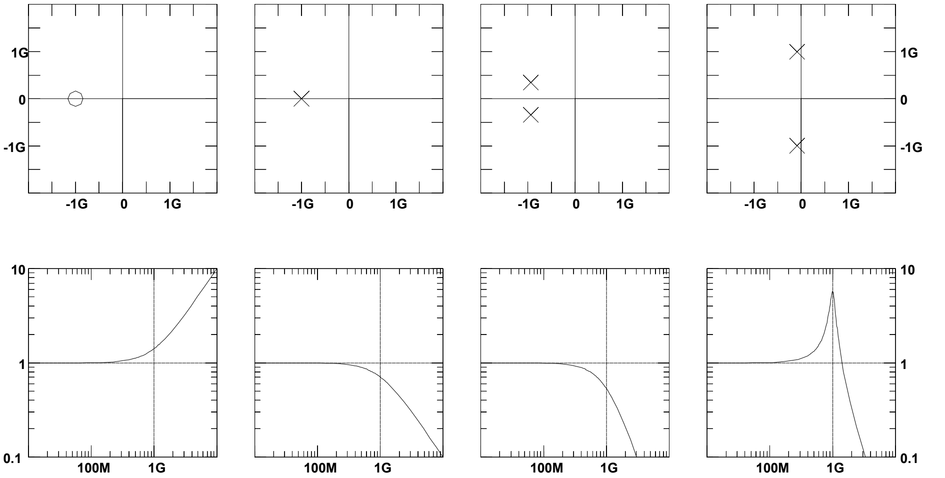


__Fig 4.21__ Examples of some typical pole or zero patterns, with their associated Bode plot. The axis of the pole-zero patterns are the real and imaginaire values of the singularities. The axis of the Bode plots are magnitude versus frequency.

4.4.5. Conclusions

In conclusion, the risks are discussed of feedback analysis that fully relies on Bode plots alone. We emphasized the importance of additional analysis, using pole-zero patterns.

Various strategies are implemented to find the poles and zeros of the loop. It has been observed that automated *simultaneous* methods are not very successful in studying complex and wideband circuits. It is plausible that automated *sequential* methods (extraction of poles and zeros, using curve-fitting techniques) yield better results.

A simple but robust algorithm for pole-zero extraction is developed, and its robustness is demonstrated with realistic examples. Exceptional good fits, over a very wide frequency band, illustrates the reliability of the proposed method. RHP-zeros were correctly extracted.

The proposed pole-zero analysis does not require any model of the individual transistors. We used *measured* two-port parameters of the individual transistors, combined them in a circuit simulator, and evaluated the (exact) loop gain in tabular form. Pole-zero extraction was the last step of the analysis.

Since an increasing number of manufacturers specify their devices in terms of two-port parameters (a table with measured s-parameters), a powerful tool is developed to perform *reliable* pole-zero analysis, using *realistic* data.

4.5. Loop gain deflation algorithms

Synthesis of stable feedback loops is a delicate process. A *fully* automated procedure for syntheses is currently not available and therefore the assistance of *manual* synthesis remains a substantial part of the overall procedure. The complexity of synthesis requires a simplification of the feedback problem, particularly in the case of manual synthesis.

The major reduction of the complexity of a transfer function is the extraction of poles and zeros from a tabular transfer function. For instance, the example of figure 4.20 has reduced a table with 300 complex numbers to a table with 6 poles and 3 zeros. Yet, any further deflation of the loop gain is desirable to relax the synthesis problem.

Since parasitic effects are essential for wideband feedback design, the commonly used low frequency approach of ignoring them is inadequate.

This study has resulted in powerful reduction algorithms for manual and automated deflation[5] of loop gain, including:

- A manual deflation method that models all parasitic singularities by *pseudo delay*.
- An automated deflation method that cancels pole-zero pairs that does not coincide *exactly*.
- An automated deflation method that approximates a transfer function with its dominant poles and zeros, within a *predefined* frequency interval.

These reduction techniques are important for the purposes of synthesis, as is described in section 5.3. This section describes the new deflation methods that expand low-frequency synthesis techniques for wideband applications.

4.5.1. Manual deflation using pseudo delay

For the purpose of practical synthesis, it is convenient to distinguish the poles and zeros in two groups: the *primary*[6] singularities and the *parasitic* singularities. This facilitates a reduction of the system order, the *dominant* system order. The combined influence of all parasitic singularities is then represented by a single real scalar: *pseudo delay*. The here proposed deflation method is mainly intended for manual interpretation of higher-order effects in pole-zero patterns.

The choice whether a singularity is primary or parasitic is arbitrary, and is mainly based on the chosen frequency band of interest. By approximation, poles and zeros have unity magnitude responses for frequencies below half the corner frequency $\omega < ½\,|p|$, thus all singularities with corner frequencies above twice the highest frequency of interest are good candidates to be classified as parasitic.

Ignorance of all parasitic singularities, which is commonly used in low-frequency design, is not suitable. It was the inspiration of this study to recognize that group delay or phase delay facilitates a fair estimation of the combined parasitic phase effects. As a result, the system order of the loop gain is reduced by the following approximation:

$$\mathring{H}(s) \approx \mathring{h}(s) \cdot \exp(-s \cdot \tau_p) \qquad \textit{within the basic frequency band}$$

[5] The word 'deflation' has been adopted from root finding algorithms for polynomials.

[6] The subdivision in 'primary' and 'parasitic' singularities is devised in this text.

The phase delay of a real zero (Z) or a conjugate pair of complex zeros (Z, Z') at zero frequency (ω→0) is derived as follows:

$$H_1(j\omega) = (1 - j\omega/Z) \quad \Rightarrow \phi \approx \tan(\phi) = -\omega/Z \quad \Rightarrow -\left(\frac{\phi}{\omega}\right) \approx \frac{1}{Z}$$

$$H_2(j\omega) = (1 - j\omega/Z)(1 - j\omega/Z') \quad \Rightarrow \phi \approx \tan(\phi) = -\omega(1/Z + 1/Z') \quad \Rightarrow -\left(\frac{\phi}{\omega}\right) \approx \frac{1}{Z} + \frac{1}{Z'}$$

The same applies for poles, however with opposite sign, and therefore we define the overall pseudo delay as follows:

$$\boxed{\tau_p = -\left(\frac{\phi}{\omega}\right) = \left[\sum 1/Z_k - \sum 1/P_k\right]_{\text{parasitic}}}$$

The magnitude response of a transfer function with pure delay is frequency independent, however this does not apply for poles and zeros. This effect limits the bandwidth of the pseudo delay approximation.

Since poles and zeros have a complementary magnitude response, any mirrored pole-zero pair with respect to the imaginary axis forms an all-pass transfer function. These combinations expand the usable bandwidth of the pseudo delay approximation. When the pairs of poles and zeros lie in Bessel position, then the combined phase delay has a maximally flat frequency response over a wide frequency band.

As is demonstrated in figure 4.22 the same applies for the group delay, however with smaller bandwidth.

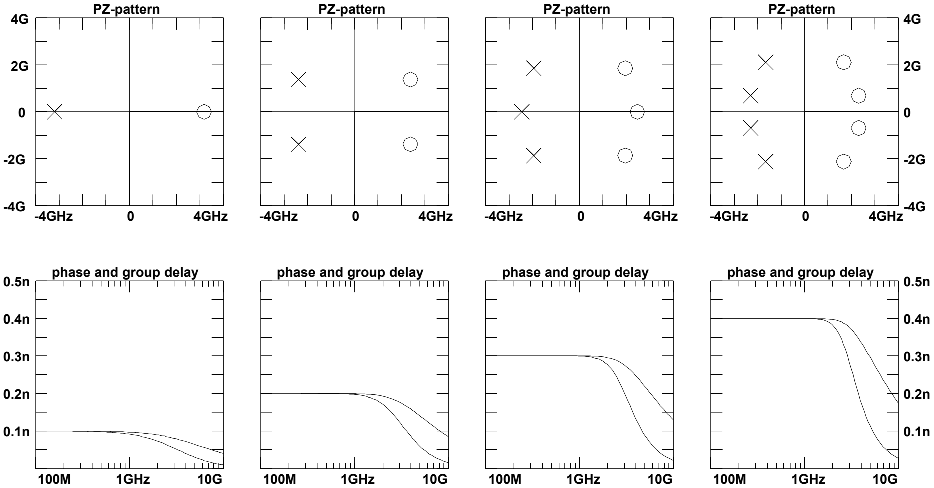


Fig 4.22 *Pole-zero patterns and group delay plots of various all-pass filters in which the poles are located in Bessel position. The transfer of these filters is characterized by maximally flat group delay.*

The most notable singularities of the specific examples in figure 4.20 are the RHP-zeros. In the examples 4.20b and 4.20c these zeros are associated with poles closely located to the mirror position. As a result, the replacement of these singularities by pseudo delay is a good approximation over a wide frequency band. When all singularities above 500

MHz are classified as parasitic singularities, then the associated parasitic pseudo delay has the following value:

A:	$t_p =$	0.23 [ns]	*3 transistors*
B:	$t_p =$	0.52 [ns]	*6 transistors, all cascoded*
C:	$t_p =$	0.41 [ns]	*6 transistors, cascode and two long-tailed pairs*

The deflation method with pseudo delay is mainly intended for manual synthesis. It is therefore handy to estimate this delay directly from the transistor parameters (see section 3.2.2), without the effort of evaluating loop gain.

As a rule of the thumb, the overall pseudo delay of a cascade of transistors is close to the sum of the delay in all the individual transistors. Cascoding is associated with the highest pseudo delay.

To indicate the importance of these delay figures, a delay of 0.3 ns is equivalent to 9 cm transition time of EM-waves in free space and 4.5 cm in standard epoxy print.

4.5.2. Automated deflation by pole-zero cancellation

Poles are canceled out by zeros that are located at the same position. In that special case, the deflation is achieved by simply removing these pairs. When their positions do not coincide but match within a predefined accuracy, e.g. 10%, then simple removal does not conform to the best feasible approximation. An improved deflation technique must therefore combine the removal of matching pairs with the adjustment of the remaining poles and zeros.

This study has resulted in a powerful deflation algorithm that uses a generalized definition for polynomial division. This subsection describes the application of weighted[7] polynomial division for reduction of the transfer order.

Pole-zero cancellation is nothing more than division on the numerator T(s) as well as the denominator N(s) of a rational transfer function H(s)=T(s)/N(s) using a common divisor polynomial X(s). An additional complication arises when the quotient is not divisible, which is similar to the round-off error problem in root finding algorithms. As a result, the deflation resolves itself to the following approximation:

$$H(s) \;=\; \frac{T(s)}{N(s)} \;\approx\; \frac{T_0(s)\cdot X(s)}{N_0(s)\cdot X(s)} \;=\; \frac{T_0(s)}{N_0(s)}$$

Various root finding algorithms, as described in section 9.5 of Numerical Recipes [413], divide the calculated roots into the original polynomial. They reduce round-off errors by both forward and backward deflation, followed by polynomial polishing[8] [413]. A deflation step uses a polynomial division with remainder according to one of these definitions:

$T(s) = q_{1r}(s)\cdot X(s) + r_{1r}(s)$	*normal (backward) division with remainder*
$T(s) = q_{2f}(s)\cdot X(s) + r_{2f}(s)\cdot s^{m-k}$	*reverse (forward) division with remainder*
$T(s) = q_{0g}(s)\cdot X(s) + r_{0g}(s)\cdot W(s)$	*weighted division with remainder*

Appendix *G* describes an algorithm for our weighted division. It can also be used for a conventional polynomial division, although classic methods are simpler.

[7] The words 'weighed division' are devised in this text.

[8] Root polishing is an algorithm that reduces round-off errors in polynomials when its roots are divided out.

A good fit for high frequencies arises when normal division is applied and the remainder polynomial (backward deflation) is ignored. Reverse division results in a good fit for low frequencies (forward deflation). To minimize the impact of increasing errors of successively deflated polynomials, some root finding algorithms polish the deflated result. One solution is the combination of the first coefficients in a forward deflation with the last coefficients in a backward deflation. This results in a fair fit for both low and high frequencies, but concentrates the error in the midband region. Study of various deflation results has shown that this concentrated error may cause unreliable results.
A change of the number of coefficients that are taken from the forward quotient polynomial may spread these errors out over a wider frequency band. This type of manual correction improves the result but prevents a full automated solution.

The algorithm proposed here performs deflation and polishing simultaneously by a weighted division and a proper choice for the weighing polynomial W(s). This approach smoothes the error and spreads it out over a wider frequency band.
The relative error δ(s) of the deflated result is as follows:

$$\delta(s) = \frac{r(s)}{X(s)} \cdot \frac{W(s)}{q(s)}$$

$$\frac{T(s)}{X(s)} = q(s) \cdot (1+\delta(s))$$

An algorithm for the optimal weighing polynomial W(s) was not found, however a significant improvement was observed when W(s)/q(s) approaches the value one for all midband and highband frequencies and goes to zero for low frequencies. Since the order of r(s) is lower than that of X(s), the proposed choice causes δ(s) to be zero for zero and infinite frequencies.
The algorithm proposed here starts with an estimation of the quotient q(s) by a conventional mix of forward and backward deflation. It then improves this quotient with an iterative loop of weighted polynomial divisions, using W(s)=q(s)-q(0) as weighing polynomial. Each iteration step performs a successive improvement of the approximation for q(s), and the iteration continues until the variation in q(s) has lost its significance.

4.5.3. Automated deflation using dominant singularities

The concept of dominant singularities is widely used, however its definition is merely intuitive. In low frequency design the concept of *dominant* singularities is equivalent to the concept of *primary* singularities, as defined in section 4.5.1. This approach is not suitable for automated synthesis because a computer aided synthesis procedure requires an unambiguous definition.
This study has resulted in a more suitable definition for dominant singularities that facilitates the reliable prediction of loop gain, using dominant approximations. The definition is just the result of the algorithm that is described in appendix *H*.
Dominant poles and zeros are adjusted primary singularities, modified so that the approximation error of the deflated transfer goes to zero at *both* ends of a predefined frequency interval. The algorithm performs this in two steps:

1. Transform the poles and zeros into polynomial form, in increasing order. Remove the higher order power terms contributing less than a predefined amount, e.g. 30%.

The roots of this reduced polynomial differ from the primary poles and zeros, and facilitate a deflation with better accuracy.

2. Polish this intermediate result by adjustment of the two highest polynomial coefficients. This forces the error to zero at the highest frequency of interest.

This polishing step is crucial for the automated compensation synthesis. It facilitates an accuracy improvement at a frequency of choice. The elaboration of the algorithm is described in appendix *H*.

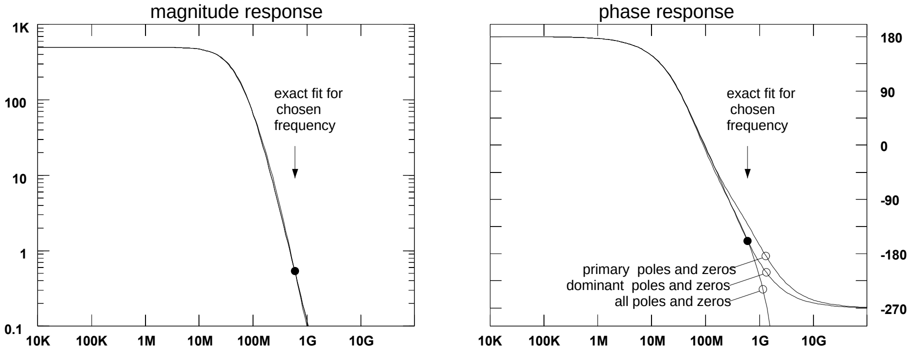


Fig 4.23 *Deflation example of a transfer function by primary poles and zeros and by dominant poles and zeros. The polishing step of the dominant deflation caused an exact fit at the chosen frequency (marked). In this example, the magnitude accuracy of both methods is comparative, however the phase accuracy of a dominant approximation is significantly better.*

Figure 4.23 demonstrates what can be achieved with the improved method. The transfer function of the example in figure 4.20c was deflated from 6 poles and 3 zeros into 4 poles and 1 zero with different methods. The primary singularities were obtained by discarding the poles and zeros with the highest magnitude. The dominant singularities were obtained by this method.

In this example, the improvement of the phase plot demonstrates the true dominant deflation method is applicable over a wider frequency interval than quasi dominant methods based on primary singularities.

4.5.4. Automated overall deflation

A full automated deflation mixes methods of pole-zero cancellation and dominant approximation. This comprises a cancellation step, followed by a dominant approximation, and completed by a second cancellation step to eliminate pole-zero pairs of intermediate results.

4.5.5. Conclusions

In conclusion, various deflation algorithms are developed for reducing the number of poles and zeros representing the loop gain transfer function. These reduction techniques are important for the purpose of synthesis. The more reduction of complexity the simpler the synthesis will be, assuming adequate accuracy in a chosen frequency interval.

- A manual deflation method is described to represent all parasitic singularities by a single number: pseudo delay. This provides an attractive figure of merit in estimating an upper limit for the available amplifier bandwidth. Compensation of pure delay is impossible because it would require non-causal compensation networks.
- An automated deflation method is described to cancel pairs of poles and zeros. When they coincide, then a simple removal of these pairs is adequate. Difficulties arise when they match within a predefined accuracy. The proposed algorithm spreads the deflation error out over a wide frequency interval, to find the most plausible solution.
- An automated deflation method is described to replace sets of poles and zeros by sets of dominant poles and zeros. The algorithm provides an exact description of the original transfer function at both ends of a chosen frequency interval. The deflation error is spread the out over the inner frequency interval, while the outer frequency interval is not considered. The performance of this algorithm is superior to simply removing all singularities that exceed a chosen frequency limit.

The proposed algorithms are robust and capable of an adequate reduction of the transfer functions in figure 4.20.