A Phase–Frequency Transfer Description of Analog and Mixed–Signal Front–End Architectures for System–Level Design

Ewout Martens* and Georges Gielen
Katholieke Universiteit Leuven, Department of Electrical Engineering, ESAT-MICAS
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
{emartens|gielen}@esat.kuleuven.ac.be

Abstract

A novel approach for the modeling of front-end architectures is presented. Architectures are described as a system transforming polyphase harmonic signals through building blocks modeled by polyphase harmonic transfer matrices and distortion tensors. The major goal of the method is to provide a model that is suited for systematic architectural exploration during front-end system design. An example of a downconversion architecture describes the system non-idealities as the result of parasitic transfers between phases and frequencies.

1. Introduction

In the last years there has been a strong growth of applications of various wireless and wireline transmission standards (e.g. GSM, EDGE, Bluetooth, xDSL). This trend has resulted in the development of several architectures for the analog and mixed-signal front-ends. Moreover, these architectures tend to become continuously more complex. On the one hand, they must be able to handle different standards, and on the other hand several techniques for improving the performance of existing architectures are applied.

Models for front-end architectures can be roughly divided into two groups. As a first approach, a high-level model for a particular architecture with almost ideal building blocks is created and simulated in tools such as Matlab/Simulink or ADS. More accurate models are used for the building blocks separately. Another approach tries to find an efficient simulation model for the architecture (e.g. [4]).

The first approach is more an *ad hoc* method that is not applicable in a systematic architectural exploration. In addition, it has a limited accuracy and simulation speed making incremental modeling difficult. The second method is efficient for verification purposes, but it does not clearly identify the information flow throughout the architecture: signals forming a logical set are split up into different signals. Moreover, parasitic signal flows will not be separated from the information flow.

This paper presents a method for describing front-end architectures that is more suited for systematic architectural exploration than the other two approaches. The goals of the approach are threefold:

- There should be a clear link between the model for the architecture and the underlying signal processing algorithm that it implements. Changes in the algorithm can easily be reflected in the architecture. This is achieved by modeling the information flows throughout the architecture instead of the real signals.
- The model makes it possible to distinct wanted signals from unwanted signals. In this way, problems can be identified. Moreover, during the evaluation of the performance of the architecture, the unwanted signals do not have to be calculated with full accuracy: mostly, the power of these signals is sufficient.
- Incremental modeling of the architecture should be possible. On the one hand, the study of the effects of a particular non-ideality must be possible. On the other hand, the model should be able to include almost real building blocks, leading to a model that is able to represent blocks at a wide range of accuracy levels.

The proposed modeling method represents all signals in the system as polyphase harmonic signals, which contain a set of polyphase bandpass signals at different carrier frequencies. The link between the information signals and the real signals in the system is established via base transformations of the polyphase signals. A more accurate representation is achieved by adding more frequencies and by increasing the model order of the signal components. The signal representation is described in section 2.

*Research Assistant of The Fund for Scientific Research – Flanders (Belgium)

1530-1591/04 $20.00 (c) 2004 IEEE

The signal transformations of the different blocks in the architecture are modeled as polyphase harmonic transfer matrices. Each element of such a matrix denotes the conversion of a signal at a certain frequency and in a certain phase to another frequency and/or phase. In this way, all linear systems — time-invariant or periodic time-variant — can be represented in a formal way. Section 3 elaborates the linear representation of building blocks. This is typically the desired behavior of the blocks.

Nonlinear behavior of the blocks is represented by distortion tensors. The input–output relation of building blocks is represented by inner products that usually can be simplified to matrix multiplication for analysis purposes. In section 4 the model for the nonlinear behavior is explained.

Section 5 illustrates the use of the proposed modeling method with an example of a downconversion architecture. The non-ideal signal flows are modeled as transfers between phases and frequencies, suggesting ways to filter them out.

Finally, conclusions are presented in section 6.

2. Signal representation

The proposed modeling method uses a signal representation that can deal with a wide variety of signals available in front-end architectures. In this section, polyphase harmonic signals are presented as an harmonic signal built up out of polyphase signals. First a brief overview of polyphase signals is given, and next the extension to polyphase harmonic signals is made.

2.1. Polyphase signals

In general, analog and mixed-signal front-ends contain building blocks performing operations on sets of signals. These sets of signals are lumped together in polyphase signals. As defined in [2], a polyphase signal is a set of N signals with the same frequency, but with different phases and/or amplitudes. Furthermore, the signals are usually bandpass signals which are efficiently represented by using the equivalent low-pass transformation. The resulting N-phase signal is a vector of complex signals:

$$\mathcal{S}(t) = \begin{bmatrix} s_1(t) & \ldots & s_N(t) \end{bmatrix}^T \tag{1}$$

Representation (1) can also be regarded as the time-varying coordinates of the signal $\mathcal{S}(t)$ with respect to the N single-phase base vectors $\mathcal{e}_k(t)$, with a signal in one phase and no signals in the other phases:

$$\mathcal{e}_k(t) = [\underset{1}{0} \quad \ldots \quad 0 \quad \underset{k}{1} \quad 0 \quad \ldots \quad \underset{N}{0}]^T \tag{2}$$

However, other bases can be chosen. A base transformation can be defined by a base transformation matrix:

$$\begin{bmatrix} \mathcal{e}'_1(t) & \ldots & \mathcal{e}'_N(t) \end{bmatrix} = \begin{bmatrix} \mathcal{e}_1(t) & \ldots & \mathcal{e}_N(t) \end{bmatrix} \cdot \mathbf{B} \tag{3}$$

The signal is converted to $\mathcal{S}'(t) = \mathbf{B}^{-1} \cdot \mathcal{S}(t)$ as can be found from general linear theory.

Apart from the single-phases base, two bases are of special interest: a base consisting of common-/differential-mode vectors and a base with symmetrical components. The first one reflects the way circuits are usually constructed whereas the second one emphasizes the path of the information signal. The latter is especially useful in the exploration and synthesis of front-end architectures. A symmetrical polyphase signal is a polyphase signal with the same amplitude for all components and with a constant phase difference between two successive components in the set [2]. For an N-phase signal, the base conversion matrix from single-phases to symmetrical-phases is

$$\mathbf{B}_N^{sp\to sc} = \begin{bmatrix} \alpha^0 & \alpha^0 & \ldots & \alpha^0 \\ \alpha^0 & \alpha^{-1} & \ldots & \alpha^{-(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^0 & \alpha^{-(N-1)} & \ldots & \alpha^{-(N-1)(N-1)} \end{bmatrix} \tag{4}$$

where $\alpha = \mathrm{e}^{\mathrm{j}\frac{2\pi}{N}}$. The symmetrical components with the phase difference $2\pi/N$ and $(N-1)2\pi/N$ are the positive and negative sequences respectively. These base transformations provide the link between the structure of the architecture (single-phase or common-/differential-mode) and the information flow (symmetrical components).

The polyphase signal can also be represented in the frequency domain:

$$\mathcal{S}(\nu) = \begin{bmatrix} \mathcal{F}\{s_1(t)\} & \ldots & \mathcal{F}\{s_N(t)\} \end{bmatrix}^T \tag{5}$$

where ν is used to denote the equivalent baseband frequency.

2.2. Polyphase harmonic signals

Signals in front-end architectures usually contain components at different frequencies. To include all these components, the signal representation defined by the proposed model is a polyphase harmonic signal. It is a vector with the equivalent baseband polyphase signals at the frequencies $\{f_1, f_2, f_3, \ldots\}$ ($f_i \geq 0$ Hz) as elements:

$$\tilde{\mathcal{S}}(t) = \begin{bmatrix} \mathcal{S}_1(t)^T & \mathcal{S}_2(t)^T & \ldots \end{bmatrix}^T \tag{6}$$

To simplify the signal representation, it is assumed that all polyphase signals $\mathcal{S}_i(t)$ have the same polyphase base.

Note that the set of frequencies $\{f_1, f_2, f_3, \ldots\}$ contains only positive frequencies. Since each component of the polyphase harmonic signal represents a physical signal, the negative frequencies contain no extra information. If the wanted information signal is twice present in a signal, the information flow throughout the architecture becomes less clear. Moreover, if there would be different information on the negative frequencies, the signals would be complex, making it impossible to use the equivalent baseband transformation.

However, some building blocks perform operations on the components on the negative frequencies. To describe these operations, the operator C_i is defined:

$$\mathrm{C}_0 f(\nu) = f(\nu) \tag{7a}$$
$$\mathrm{C}_1 f(\nu) = [f(-\nu)]^* \tag{7b}$$

It is straightforward that an implementation of the model for performance evaluation will reduce the number of components in the polyphase harmonic signals if, for example, one is not interested in all the common-mode signals. The proposed methodology provides the flexibility to include more signals to enhance the complexity. Also, on the signal components the principle of incremental modeling can be applied: a first analysis may contain signal components independent of ν whereas subsequent analyses may use refined models. This give a system designer the possibility to gradually refine the detail of the front-end architecture.

3. Linear representation of building blocks

Analog front-end architectures contain filters, phase-converters, mixers, A-to-D converters, etc. This section defines the formal reference frame of the polyphase harmonic transfer matrix for the modeling of a linear approximation of these building blocks. The next section will extend the model to also include weakly nonlinear behavior.

3.1. Polyphase filters

A polyphase filter can be represented by a polyphase transfer matrix $\mathcal{H}(\nu)$ built out of phase transfer functions. This matrix can be used as an ordinary transfer function:

$$\mathcal{Y}(\nu) = \mathcal{H}(\nu) \cdot \mathcal{X}(\nu) \tag{8}$$

where $\mathcal{X}(\nu)$ and $\mathcal{Y}(\nu)$ are the frequency-domain representation of the input and output signal respectively.

If a base transformation with characteristic matrix $\mathbf{B}$ is performed on the signals, then the polyphase transfer matrix is converted according to equation (9):

$$\mathcal{H}'(\nu) = \mathbf{B}^{-1} \cdot \mathcal{H}(\nu) \cdot \mathbf{B} \tag{9}$$

Note that equation (9) implies that if the single-phase representation of a polyphase filter is a circulant matrix, then a symmetrical component with phase difference $\Delta\phi$ is mapped on a symmetrical component with the same characteristic $\Delta\phi$. This kind of filter is a well-known symmetrical polyphase filter. Furthermore, if the symmetrical component of order k has a phase difference $k\frac{2\pi}{N}$, then the phase transfer function of the k-th and the $(N-k+2)$-th symmetrical component are linked to each other as follows:

$$H_{N-k+2}(\mathrm{j}2\pi\nu) = [H_k(-\mathrm{j}2\pi\nu)]^* \tag{10}$$

A polyphase filter can be represented as a mapping operator for polyphase harmonic signals defined on the frequencies $\{f_1, f_2, \ldots\}$ using the block diagonal matrix $\tilde{\mathcal{H}}(\nu)$:

$$\tilde{\mathcal{H}}(\nu) = \mathrm{diag}\left(\mathcal{H}_1(\nu), \ldots, \mathcal{H}_N(\nu)\right) \tag{11}$$

with $\mathcal{H}_i(\nu) = \mathcal{H}(\nu + f_i)$. $\tilde{\mathcal{H}}(\nu)$ is an example of a polyphase harmonic transfer matrix (PHTM). The polyphase harmonic input signal $\tilde{\mathcal{X}}(\nu)$ is mapped onto $\tilde{\mathcal{Y}}(\nu)$ by a matrix multiplication:

$$\tilde{\mathcal{Y}}(\nu) = \tilde{\mathcal{H}}(\nu) \cdot \tilde{\mathcal{X}}(\nu) \tag{12}$$

A PHTM can also be regarded as an extension to multiple phases of a truncated harmonic transfer matrix as defined in [3]. However, only the positive frequency components are included.

3.2. Phase-converters

A phase-converter converts a polyphase signal to a different number of phases. It also performs a filtering operation resulting in a similar equation as (9). This polyphase matrix can also be converted to a PHTM using (11).

3.3. Polyphase mixing

A linear mixer used for up- or downconversion can be regarded as a linear periodic time-varying system. Such a system can be modeled using harmonic transfer matrices (HTM) [3]. A polyphase mixing operation consists of a set of mixers whose operation can be described by a PHTM $\tilde{\mathcal{M}}(\nu)$ written as a function of the HTMs of the individual mixers. The input–output relation of the polyphase mixer with input $\tilde{\mathcal{X}}$ and output $\tilde{\mathcal{Y}}$ can then be written as follows:

$$\tilde{\mathcal{Y}}(\nu) = \tilde{\mathcal{M}}(\nu) \cdot \tilde{\mathcal{X}}(\nu) \tag{13}$$

A polyphase mixer is modeled as a two-stage operation as illustrated in figure 1 for a polyphase mixer with $N = 4$ input phases and $M = 4$ output phases. The first stage contains a set of mixers of which each multiplies a part of the input signal with a part of the oscillator signal. The result of the first stage is a polyphase signal with, in general, more phases than the input or output signal. In the second stage this polyphase signal is converted to the desired number of phases using a phase-converter. For each stage a PHTM can be derived. The PHTM of the entire polyphase mixer is then found as the product of the two PHTMs of the stages.

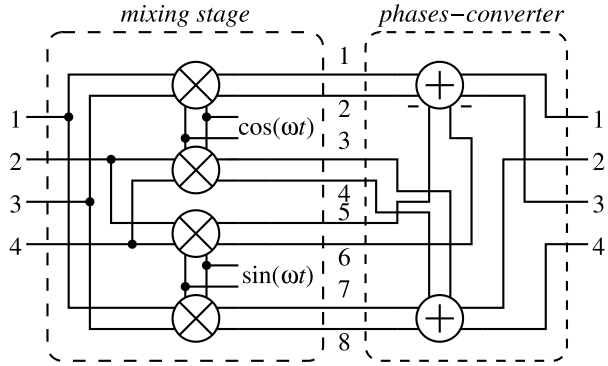

Figure 1. Example of a polyphase mixer modeled as a two-stage operation.

Mixing stage. The N-phase input signal $\tilde{\mathcal{X}}$ is split up into $\hat{N}$ parts $\tilde{\mathcal{X}}^q$ ($\hat{N}=2$ in figure 1). It is assumed that the polyphase bases of the input signal and of the parts are chosen to satisfy the following equation at each carrier frequency f_i:

$$\mathcal{X}_i(\nu) = \left[\mathcal{X}_i^1(\nu)^T \quad \ldots \quad \mathcal{X}_i^{\hat{N}}(\nu)^T\right]^T \tag{14}$$

Each mixer in the polyphase mixer is assigned a unique pair: mixer (k,l) is the mixer with the k-th part of the N-phase input signal and the l-th part of the oscillator signal as inputs. The number of mixers for each part of the input signal is denoted by $\hat{P}$ ($\hat{P}=2$ in figure 1).

The output signal of mixer (k,l) is the polyphase harmonic signal $\tilde{\mathcal{Z}}^{(k,l)}$. The phase transfer from phase p to phase q can be represented by an harmonic transfer matrix:

$$\begin{bmatrix} C_1\tilde{z}_q^{(k,l)}(\nu) \\ C_0\tilde{z}_q^{(k,l)}(\nu)\end{bmatrix} = \sum_q \tilde{H}_{q,p}^{(k,l)}(\nu)\cdot\begin{bmatrix} C_1\tilde{x}_p^k(\nu) \\ C_0\tilde{x}_p^k(\nu)\end{bmatrix} \tag{15}$$

where the operators C_0 and C_1 are defined by (7). Since our goal is to model the information flow throughout the architecture, it is preferable to model the information signal only by one component (at positive frequencies) instead of by two components (at positive and negative frequencies). Therefore, polyphase harmonic signals contain only the positive frequency components and the operators C_0 and C_1 are put into the PHTMs. This also simplifies the analysis of the information power flow throughout the architecture.

By proper arrangement of the elements of the HTMs of (15), the output of mixer (k,l) can be calculated with a PHTM:

$$\tilde{\mathcal{Z}}^{(k,l)}(\nu) = \tilde{\mathcal{H}}^{(k,l)}(\nu)\cdot\tilde{\mathcal{X}}^k(\nu) \tag{16}$$

where $\tilde{\mathcal{H}}^{(k,l)}(\nu)$ is built up out of polyphase transfer matrices:

$$\tilde{\mathcal{H}}^{(k,l)}(\nu) = \begin{bmatrix} \mathcal{H}_{1,1}^{(k,l)}(\nu) & \ldots & \mathcal{H}_{1,\tilde{N}}^{(k,l)}(\nu) \\ \vdots & \ddots & \vdots \\ \mathcal{H}_{\tilde{P},1}^{(k,l)}(\nu) & \ldots & \mathcal{H}_{\tilde{P},\tilde{N}}^{(k,l)}(\nu)\end{bmatrix} \tag{17}$$

where $\tilde{N} = N/\hat{N}$ and $\tilde{P} = P/\hat{P}$. Each polyphase transfer matrix $\mathcal{H}_{j,i}^{(k,l)}$ represents the polyphase transfer from frequency f_i to frequency f_j in mixer (k,l).

The signals $\tilde{\mathcal{Z}}^{(k,l)}$ are lumped together into the polyphase harmonic signal $\tilde{\mathcal{Z}}$:

$$\tilde{\mathcal{Z}}(\nu) = \left[\tilde{\mathcal{Z}}^{(1,1)}(\nu)^T \quad \ldots \quad \tilde{\mathcal{Z}}^{(\hat{N},\hat{P})}(\nu)^T\right]^T \tag{18}$$

It can be shown that $\tilde{\mathcal{Z}}(\nu)$ can be calculated out of the input signal $\tilde{\mathcal{X}}(\nu)$ using the PHTM $\tilde{\mathcal{T}}(\nu)$ in which the transfer from frequency f_i to frequency f_j is given by $\tilde{\mathcal{T}}_{j,i}(\nu)$:

$$\tilde{\mathcal{T}}_{j,i}(\nu) = \mathrm{diag}\left(\begin{bmatrix}\mathcal{H}_{j,i}^{(1,1)} \\ \vdots \\ \mathcal{H}_{j,i}^{(1,\hat{P})}\end{bmatrix}, \ldots, \begin{bmatrix}\mathcal{H}_{j,i}^{(\hat{N},1)} \\ \vdots \\ \mathcal{H}_{j,i}^{(\hat{N},\hat{P})}\end{bmatrix}\right) \tag{19}$$

Phase-converter. The second stage consists of the conversion of the $(N\cdot\hat{P})$-phase output signal of the mixing stage to the M-phase output signal. This stage can be characterized by a PHTM $\tilde{\mathcal{A}}(\nu)$. The PHTM of the characteristic equation (13) for the polyphase mixer is then found as the cascade connection of the two PHTMs of the two stages:

$$\tilde{\mathcal{M}}(\nu) = \tilde{\mathcal{A}}(\nu)\cdot\tilde{\mathcal{T}}(\nu) \tag{20}$$

4. Weakly nonlinear representation of architecture building blocks

4.1. Single-phase weakly nonlinear behavior.

In order not to complicate the description of the modeling approach, it is first assumed that the nonlinearity can be described in the time domain by a polynomial equation:

$$y(t) = \sum_{m=1}^{\infty} K_m\,[x(t)]^m \tag{21}$$

This corresponds to the case of a memoryless nonlinearity.

Assume that the input signal $x(t)$ contains frequency components at frequencies $\{f_1,\ldots,f_a\}$ and that the output signal $y_m(t) = K_m\,[x(t)]^m$ is described by the components at frequencies $\{f_1',\ldots,f_b'\}$. They are represented by the harmonic signals $\tilde{x}(\nu)$ and $\tilde{y}_m(\nu)$. One can proof that $\tilde{y}_m(\nu)$ can be calculated by repeated calculation of inner products between a distortion tensor $\boldsymbol{D}_m(\nu)$ of dimension $(m+1)$ and the single-phase harmonic signal $\tilde{x}(\nu)$:

$$\tilde{y}_m(\nu) = \left\langle\cdots\left\langle\left\langle\boldsymbol{D}_m(\nu),\tilde{x}(\nu)\right\rangle_2,\tilde{x}(\nu)\right\rangle_2'\cdots\right\rangle_2' \tag{22}$$

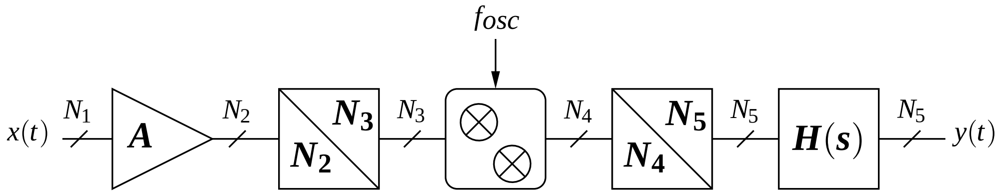

Figure 2. A generic architecture for a downconversion. The symbols N_i denote the number of phases.

The inner product between the $(m+1)$-dimensional tensor $\boldsymbol{T}(\nu)$ and $\tilde{x}(\nu)$ is an m-dimensional tensor where each element can be calculated as follows [1]:

$$\left(\langle \boldsymbol{T}(\nu), \tilde{x}(\nu)\rangle_2\right)_{j,i_2,\ldots,i_m} = \sum_{q=1}^{a} t_{j,q,i_2,\ldots,i_m}(\nu)\cdot x_k(\nu) \qquad (23)$$

where $t_{j,q,i_2,\ldots,i_m}(\nu)$ are the elements of the tensor $\boldsymbol{T}(\nu)$ (contravariant in the first variable and covariant in the other variables). For $\langle\cdot,\cdot\rangle'$ the convolution is used instead of the normal product. Subscript '$_2$' indicates that the inner product is taken over the second index of the distortion tensor.

Like the elements of the PHTM $\tilde{\mathcal{M}}(\nu)$ of (13), the elements of the distortion tensors contain operators to take into account the components at negative frequencies. These operators are defined recursively:

$$\begin{aligned} &\mathrm{C}_{n_1,\ldots,n_q,n_{q+1},\ldots,n_p}\left\{f_1(\nu)\otimes\cdots\otimes f_q(\nu)\right\} = \\ &\qquad \left[\mathrm{C}_{n_1}\left\{f_1(\nu)\right\}\otimes\cdots\otimes \mathrm{C}_{n_q}\left\{f_q(\nu)\right\}\right]\otimes \mathrm{C}_{n_{q+1},\ldots,n_p} \end{aligned} \qquad (24)$$

where n_i is 0 or 1, and C_0 and C_1 are defined by (7). $\otimes$ denotes the convolution operation.

For example, for a third-order nonlinearity with factor K_3 in (21), $\boldsymbol{D}_3$ can be calculated as follows:

$$\begin{aligned} &(d_3)_{j,p_1,p_2,p_3} = \\ &\begin{cases} \frac{3/2K_3}{f_{occ}(p_1,p_2,p_3)}\mathrm{C}_{0,\substack{0\\1},\substack{0\\1}} & f_n = f_{p_1}\pm f_{p_2}\pm f_{p_3} \\ \frac{3/2K_3}{f_{occ}(p_1,p_2,p_3)}\mathrm{C}_{1,\substack{0\\1},\substack{0\\1}} & f_n = -f_{p_1}\pm f_{p_2}\pm f_{p_3}\neq 0\text{ Hz} \\ 0 & \textit{otherwise} \end{cases} \end{aligned} \qquad (25)$$

for $f_{p_3}\geq f_{p_2}\geq f_{p_1}$ (otherwise it is zero). The occurrence function f_{occ} calculates the product of the faculties of the occurrences of its arguments.

Note that the elements of the distortion tensor are independent of ν. This results from the choice for a memoryless nonlinearity. A distortion tensor with non-constant elements models a nonlinearity with memory.

During analysis of an architecture, the last inner product in (22) can be approximated by a matrix multiplication. For a third-order nonlinearity this means the approximation:

$$\begin{aligned} &\langle\langle \boldsymbol{D}_3(\nu),\tilde{x}(\nu)\rangle_2,\tilde{x}(\nu)\rangle_2' \approx \\ &\qquad \left[\langle\langle \boldsymbol{D}_3(\nu),\tilde{x}(\nu)\rangle_2,\tilde{x}(\nu)\rangle_2'\right]_{\nu=0} \end{aligned} \qquad (26)$$

In this way, frequency transfers are directly described by the matrix multiplication. Each element m_{ji} of (26) represent the transfer from frequency f_i to f_j by the multiplication of the component at f_i by the components at f_k and f_l with $f_k, f_l \leq f_i$. If one wants to know the total contribution of the component at f_i to the resulting signal component at f_j, one has to take into account also the transfers via frequencies f_k and f_l which may be greater than f_i. These can be determined by calculating the inner products:

$$\langle\langle \boldsymbol{D}_3(\nu),\tilde{x}(\nu)\rangle_2,\tilde{x}(\nu)\rangle_2' \qquad f_k\leq f_l\leq f_i \qquad (27a)$$
$$\langle\langle \boldsymbol{D}_3(\nu),\tilde{x}(\nu)\rangle_2,\tilde{x}(\nu)\rangle_3' \qquad f_k\leq f_i\leq f_l \qquad (27b)$$
$$\langle\langle \boldsymbol{D}_3(\nu),\tilde{x}(\nu)\rangle_3,\tilde{x}(\nu)\rangle_3' \qquad f_i\leq f_k\leq f_i \qquad (27c)$$

4.2. Polyphase weakly nonlinear behavior.

The distortion tensors described above can easily be extended to the case of signals represented by polyphase harmonic vectors instead of harmonic vectors. The inner product of (23) does not change, but the sum must be taken over all elements of the harmonic vectors.

5. Example

At a high abstraction level, a downconversion operation can be characterized by the following equation ($\mathcal{H}\{\cdot\}$ denotes the Hilbert-transform):

$$y(t) = \Re\left\{\left[x(t)+\mathrm{j}\mathcal{H}\{x(t)\}\right]\cdot \mathrm{e}^{-\mathrm{j}2\pi f_{osc}t}\right\} \qquad (28)$$

This equation can easily be translated into the generic downconversion model of figure 2. As described in section 3.3 the polyphase mixing consists of a mixing stage followed by a phases-converter. Furthermore, a gain factor (A) and a filtering operation ($H(s)$) to obtain more degrees of freedom in the architecture. The translation from the equation defining the operations and the operators of a generic architecture is easily generalized for the major part of operations encountered in front-end architectures.

As an example, the number of phases are chosen as follows: $N_1 = 1$, $N_2 = 2$, $N_3 = 4$, $N_4 = 8$ and $N_5 = 4$. This choice results in the same polyphase mixer as the one depicted in figure 1. The architecture is analyzed using a prototype implementation of the presented model in Matlab [5].

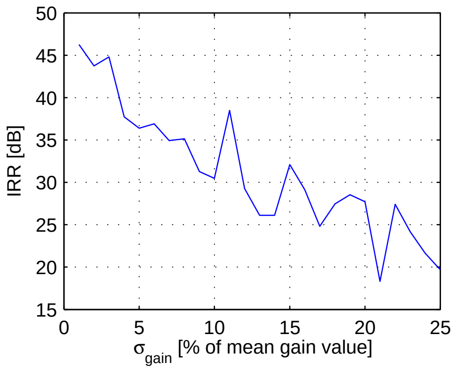

Figure 3. *IRR* degradation due to mismatch between the gain of the mixers.

Mismatch. The gain mismatch between the mixers of the polyphase mixer results in a decrease of the image rejection ratio. Figure 3 can be used for determining the maximum allowable gain mismatch. It can be shown that an analysis according to the first modeling approach (see section 1) leads to a similar result.

In contrast with the other two approaches, the phase-transfer model also shows the origin of the *IRR* degradation as a result of the transfers between phases and frequencies. An example of these transfers is shown in figure 4 for the transfer from the image frequency band to the IF frequency band. The transfers drawn with thick lines correspond to the transfers from the signal frequency band to the IF frequency band. This means that the signal frequency is transfered from phase 4 to phase 4. The parasitic transfers that degrade the *IRR* are indicated by the colored paths. Without mismatch, the two paths compensate each other.

Note that to increase the *IRR*, a solution will be to filter out phase 2 before the mixing stage (i.e. by an appropriate N_2-to-N_3 phase-converter). However, another solution is to use another N_4-to-N_5 converter since the main parasitic signal flow is via phase 4 whereas the main information signal flow is via phase 8.

Weak nonlinearities. To extend the model to weak nonlinearities, distortion tensors should be added parallel to the PHTMs. They add extra phase-frequency transfers.

As an example, a third-order distortion tensor has been added in parallel to the mixing stage modeling mixers. This tensor models the third-order nonlinearity of mixers with an IP_3 of 3 dBm (and a conversion gain of 5 dB). Assume the input contains signal components at 900 Mhz, 900.8 Mhz and 901.6 Mhz of -94 dBm, -43 dBm and -43 dBm. This results in a main distortion component in phase 4 of -109 dBm which is 7.5 dB higher than the component in phase 8.

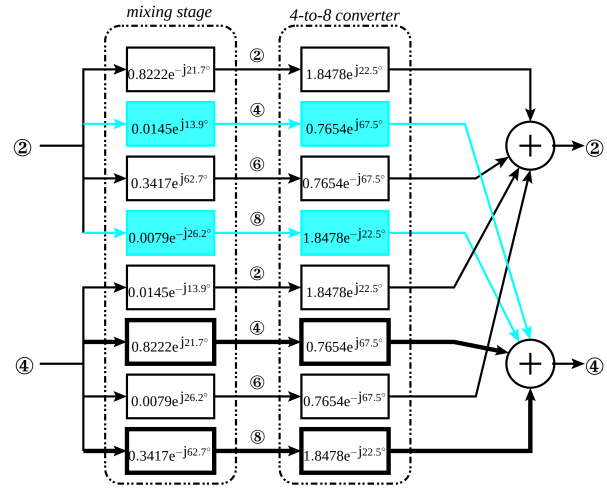

Figure 4. Transfers between symmetrical components for the transfer from f_{im} to f_{IF}.

6. Conclusions

A systematic way for the modeling of signal flows in front-end architectures has been proposed. The modeling method using polyphase harmonic signals, polyphase harmonic transfer matrices and distortion tensors leads to an insight in the phase and frequency transfers within the system. In this way, the non-ideal signal flows can be identified and solutions can be provided in a structured manner.

In addition, the method is also able to evaluate the performance of the architecture at different levels of accuracy. These properties make the method suited for systematic front-end architectural exploration.

References

[1] George Arfken. *Mathematical Methods for Physicists*. Academic Press, Inc., San Diego, 1985.

[2] M.J. Gingell. “Single sideband modulation using sequence asymmetric polyphase networks”. *Electrical Communication*, 48(1–2):21–25, 1973.

[3] P. Vanassche, G. Gielen and W. Sansen. “Symbolic modeling of perdiodically time-varying systems using harmonic transfer matrices”. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 21(9):1011–1024, Sep. 2002.

[4] P. Wambacq *et. al.* “Dataflow Simulation of Mixed-Signal Communication Circuits Using a Local Multirate, Multicarrier Signal Representation”. *IEEE Tran. Circuits and Systems–I: Fundamental Theory and Applications*, 49(11):1554–1562, Nov. 2002.

[5] The MathWorks. *MATLAB: The language of technical computing*. `http://www.mathworks.com/`.