ETS-RT - 2016-002

IMPLEMENTATION DETAILS OF A HARDWARE ABSTRACTION-BASED DESIGN METHOLODY FOR RADIOFREQUENCY CIRCUITS

Examples of Linear Devices

SABEUR LAFI
AMMAR KOUKI
JEAN BELZILE

ETS-RT - 2016-002

IMPLEMENTATION DETAILS OF A HARDWARE ABSTRACTION-BASED DESIGN METHOLODY FOR RADIOFREQUENCY CIRCUITS
Examples of Linear Devices

DÉTAILS D'IMPLÉMENTATION D'UNE MÉTHODOLOGIE DE CONCEPTION DE CIRCUITS RADIOFRÉQUENCES BASÉE SUR L'ABSTRACTION MATÉRIELLE
Exemples de Composants Linéaires

ETS TECHNICAL REPORT

SABEUR LAFI
AMMAR KOUKI
JEAN BELZILE

ELECTRICAL ENGINEERING DEPARTMENT

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MONTRÉAL, MAY 18, 2016

ETS-RT - 2016-002

IMPLEMENTATION DETAILS OF A HARDWARE ABSTRACTION-BASED DESIGN METHOLODY FOR RADIOFREQUENCY CIRCUITS

Examples of Linear Devices

SABEUR LAFI
AMMAR KOUKI
JEAN BELZILE
ELECTRICAL ENGINEERING DEPARTMENT
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

Electronic version of this technical report is available on the École de technologie supérieure web site (http://www.etsmtl.ca).

In order to request a paper copy, please contact :

Service de la bibliothèque
École de technologie supérieure
1100, rue Notre-Dame Ouest
Montréal (Québec)
H3C 1K3

Phone : (514) 396-8946
Fax : (514) 396-8633
Email : biblio@etsmtl.ca

ISBN 978-2-921145-86-2

Legal Deposit : Bibliothèque et Archives nationales du Québec, 2016
Legal Deposit : Library and Archives Canada, 2016

DÉTAILS D'IMPLÉMENTATION D'UNE MÉTHODOLOGIE DE CONCEPTION DE CIRCUITS RADIOFRÉQUENCES BASÉE SUR L'ABSTRACTION MATÉRIELLE
Exemples de Composants Linéaires

SABEUR LAFI
AMMAR KOUKI
JEAN BELZILE

SOMMAIRE

L'émergence de nouvelles applications et services mobiles contribuent à la hausse des attentes des utilisateurs en ce qui a trait aux performances des systèmes de radiocommunications. Toutefois, la conception de tels systèmes devient de plus en plus difficile à cause de la multitude de normes en usage, l'encombrement du spectre électromagnétique et l'hostilité accrue des environnements d'exploitation. Afin de faire face à ces exigences, des améliorations remarquables des approches de conception des systèmes radiofréquences doivent avoir le jour. À cet effet, plusieurs défis de conception tels que les problèmes de productivité, collaboration entre concepteurs, automatisation, conception et réutilisation des solutions ainsi que l'insertion de nouvelles technologies doivent avoir une attention particulière.

Dans l'optique de contribuer à la résolution de ces problèmes de conception, nous avons proposé dans des publications scientifiques récentes une nouvelle méthodologie de conception des circuits radiofréquences basée sur l'abstraction matérielle. Nous avons particulièrement détaillé les piliers de cette approche se basant sur un cycle de conception de cinq étapes qui est conçue autour d'une structure de données multidimensionnelle, la matrice Q, et à laquelle est projetée une stratégie d'abstraction. Nous avons choisi de valider la méthodologie proposée par le biais de cas d'études correspondant à des applications réelles. Dans ce rapport, nous détaillons le processus de conception d'un composant de sélection de fréquences. Deux scénarios différents et les étapes de conception sont particulièrement soulignés.

Mots clefs: Conception radiofréquence, abstraction matérielle, matrice Q, validation.

IMPLEMENTATION DETAILS OF A HARDWARE ABSTRACTION-BASED DESIGN METHOLODY FOR RADIOFREQUENCY CIRCUITS
Examples of Linear Devices

SABEUR LAFI
AMMAR KOUKI
JEAN BELZILE

ABSTRACT

The emergence of new applications and services raises consumer expectations regarding future radio systems' performance. However, radio design is becoming more challenging due to multi-standard functionality, spectrum crowdedness and harsh operating environments. In order to keep pace with the emerging requirements, notable enhancements in radiofrequency design approaches are required. At this regard, several design challenges should be addressed especially in terms of productivity, design collaboration, automation and reuse as well as ensuring better technology insertion.

To tackle all these challenges, we have proposed in recent research publications a proposal of a new radiofrequency design framework based on hardware abstraction. We have particularly detailed its key foundations: a five-step design scheme that is streamlined to a novel abstraction strategy and built around a multidimensional data structure, namely the Q-matrix. We have undertaken the validation process of the proposed framework through selected design case studies corresponding to real-world applications. In this technical report, we present in some detail the framework validation results for the design of a radiofrequency selection device. Two scenarios were investigated and the overall design process was detailed.

Keywords: radiofrequency design, hardware abstraction, Q-matrix, framework validation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 OVERVIEW OF THE HARDWARE ABSTRACTION-BASED METHODOLOGY FOR RADIOFREQUENCY DESIGN

1.1 Introduction

In previous publications (e.g., (Lafi, 2015)), we presented the foundations of a radiofrequency (RF) design methodology based on hardware-abstraction. This design methodology streamlines a five-step design scheme with an abstraction strategy and a central data structure (i.e., Q matrix). In this chapter, we review the design stages of the proposed framework from a practical point of view.

1.2 Design Framework

As shown in Figure 1.1, the proposed framework consists of five-stage design scheme that is built around the Q-matrix. This design scheme is streamlined with a hardware abstraction strategy that defines three modeling views: Requirements and Platform-Independent Models, Platform-Specific Models and Platform Models. Each view expresses a given design viewpoint that can be associated to a given abstraction level. These theoretical concepts need to be implemented from a practical standpoint in a set of comprehensive steps where inputs and outputs of each step are clear. Figure 1.2 depicts a flat representation of the design framework from the designer's perspective. The design process goes step-by-step through the five-stage design cycle as illustrated in the flowchart of Figure 1.3:

1.2.1 UML/SysML Models

This step starts the design process. It requires a mandatory input, namely specifications. These specifications may be in text format, spreadsheets, etc. The designer uses a model creation tool to capture these specifications in SysML models (other standard modeling languages can also be used). The models can be created from scratch or using existing modeling templates (e.g., modeling profiles). If the latter solution is adopted, the designer's derives the required functionality from the existing constructs (e.g., describing predefined functionalities). Depending on that functionality, default

parameters' values are used to define the device's models. It is worth noting that both requirements and platform-independent models are produced in this step. As SysML/UML standard languages are concerned, there is a number of commercial software packages that can be used to capture specifications and produce visual RM/PIM models. Most of these software tools are standalone packages that were mainly optimized to be used in software engineering. Thus, they require some modifications to be used in RF design. Ideally, any integrated design environment that is intended to make this framework effective requires a dedicated software tool or plugin that is fully optimized for RF modeling using standard modeling languages. At the end of UML/SysML modeling, the output of this step is a set of files which holds the visual UML/SysML models representing the device's RM/PIM models.

1.2.2 XML Description

The RM/PIM models produced in the previous step are used at this level to generate corresponding XML description. For this purpose, the designer uses a XML generation tool to convert visual models into XML markup language files that can be automatically processed, updated and exchanged between different tools, designers and design environments. Similarly to modeling packages, there are currently existing tools mostly originating from software engineering domain that fulfill this task. They can be adapted to RF design as well. The generation of XML files is accompanied with the creation of the Q-matrix. At this level, it is populated only with a subset of the data captured in RM/PIM models. Since these models are not yet verified, they may contain errors and incoherencies and subsequently contaminate the Q-matrix.

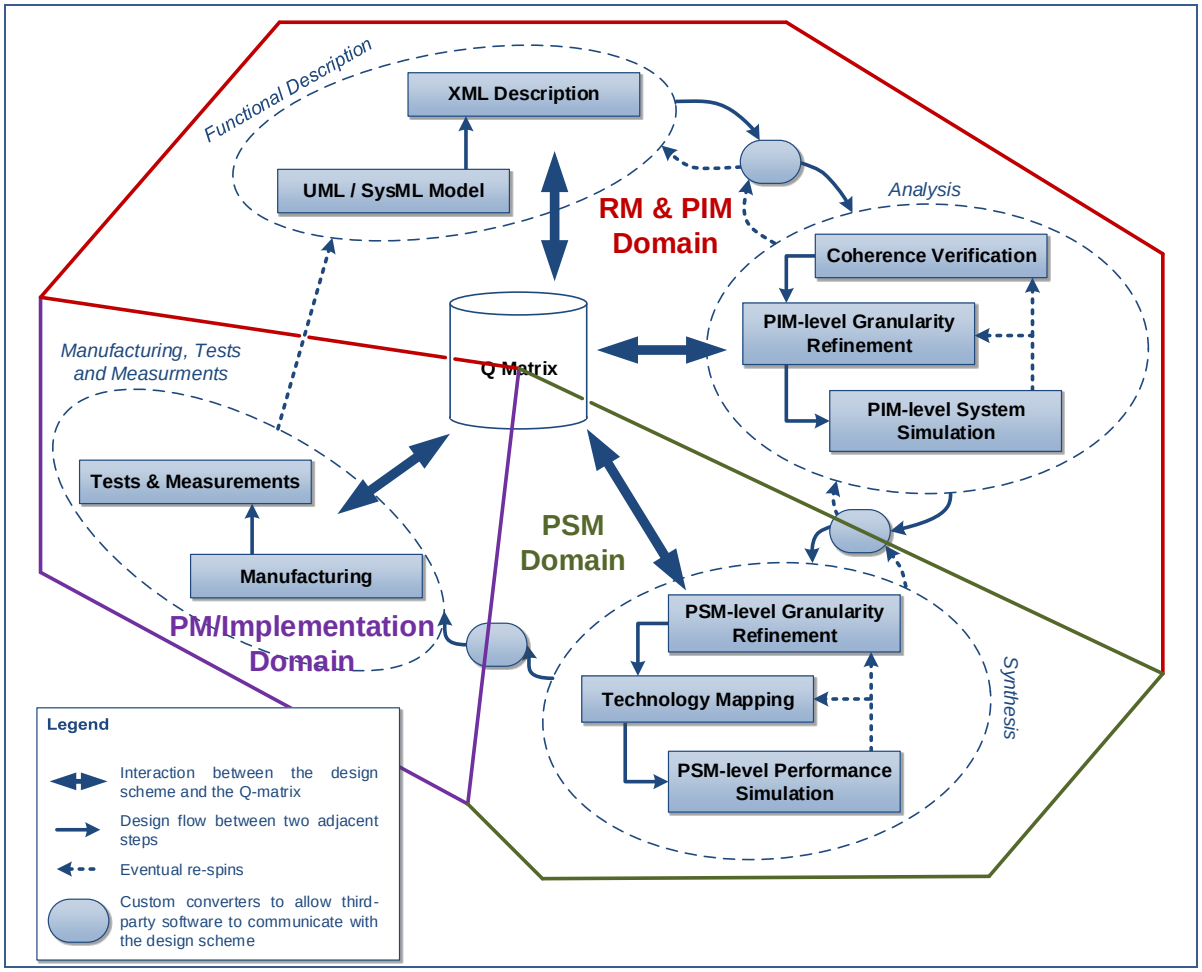


Figure 1.1 Streamlining the proposed design framework with the RF/microwave abstraction strategy

1.2.3 Coherence Verification

In this step, the RM/PIM models and XML files along with the associated Q-matrix are validated to ensure the coherence of the models' parameters and data. To this end, the designer uses a model validation tool that checks the RM/PIM models based on a set of coherence verification rules which are provided also as input. The tool produces a validation report where any potential inconsistencies and validation errors are outlined. If errors reported, the designer proceeds to the revision and correction of the RM/PIM models before it attempts to generate again the corresponding XML description and Q-

matrix and carry out another coherence verification test. If all models and the related Q-matrix are proven to be coherent, the designer can proceed to the next step where a design solution can be searched and optimized. Unfortunately, there are no existing tools that can be used with no or at least minor modifications to carry out the task of coherence verification. To overcome this weakness, we used scripting languages to accomplish this process. Some specialized software libraries can also be used for the same purpose (e.g., analog filter libraries).

1.2.4 PIM-level Granularity Refinement

Based on the RM/PIM models, the designer undertakes a space exploration process to look for a solution that satisfies the initial requirements. If a promising design solution is found, this step may be skipped. If no satisfactory solution is found, the designer proceeds to the refinement of the PIM models in order to relax the design constraints. For example, if there are any parameters that are overestimated, they might be given more reasonable values. If any key design parameters were assigned to default values (particularly when design templates are used), these defaults may be subject to review in order to make design solution search more efficient. In all these cases, another coherence verification test is mandatory to keep the RM/PIM models consistent and be sure that the new modifications did not compromise their coherence. Having done this and no satisfactory design solution is found, the next action is PIM-level granularity refinement that consists of changing the design view without changing neither the abstraction level nor the design viewpoint. This is practically demonstrated by the derivation of additional information within the models in the purpose of reducing the design space search area. For every granularity refinement iteration, a new QBlock is added to the Q-matrix to hold the data related to the RM/PIM models corresponding to that iteration. To succeed the granularity refinement process, the designer uses jointly two main tools: a model refiner and a model optimizer. Among the existing design tools, we noticed that scripting languages may be used as refinement (and optimization) tools at a reasonable time and effort cost. Some specialized commercial packages might also be used for the

optimization of certain models (often with significant overhead). In fact, an effort of manual models conversion should be carried out in order to transform them into an acceptable input to these tools.

1.2.5 PIM-level Simulation

In this step, the designer carries out different simulations in order to ensure that the design solution meets the requirements expressed by the RM/PIM models. Generally, it consists of iterative performance evaluations (based on metrics mostly depending on the selected functionality) and optimizations. Designers can use scripting and programming languages as well as available APIs to carry out the required simulations for performance assessment. Most system-level tools can be used but require an overhead due to manual adaptation of RM/PIM models to the target design environment. During the PIM-level simulations, the Q-matrix is regularly updated. At every iteration, one or many new QBlocks can also be added. Feeding continuously the Q-matrix with data is useful for various purposes (e.g., conducting a performance comparison between multiple candidate design solutions). Using a relevant model-to-model transformation, one or many PSMs are synthesized from the best PIM-level candidate design solution. Therefore, the output of this step is one or many platform-specific design solutions (i.e., PSM). In addition, an up-to-date Q-matrix which holds the history of the design process so far is also got. The original models (especially RMs often used for validation and traceability) are frequently attached to the resulting PSMs for the next design steps.

1.2.6 PSM-level Granularity Refinement

It is the first step in the synthesis design stage. The platform-specific design solution (i.e., PSM) already created using a PIM-to-PSM transformation is augmented with platform-specific information only (e.g., target implementation technology such as distributed lines). Nevertheless, it does not necessary encompass detailed platform information (e.g., substrate data, technology constraints, etc.). That is why technology input is required to complete the model information. Before technology mapping, the

PSM is assessed against the requirements. If the PSM does not satisfy the requirements, granularity refinement attempts to enhance the quality of that design solution. Idem, a new QBlock is added to the Q-matrix in order to capture the corresponding design data.

1.2.7 Technology Mapping

At this step, the resulting PSM is augmented with detailed technology information. In practice, this means that each PSM element is enriched with corresponding technology features and items (e.g., substrate, physical characteristics and constraints, physical shapes and dimensions, etc.). Technology information may be provided using component libraries, electrical models or some existing tools (such as ADS LineCalc) that can be used to synthesize the physical properties of each PSM element. Ideally, this process is fully automated which implies the use of specialized tools that can automatically map each PSM component to the corresponding technology details. For the following case studies, we used scripting and programming languages for this purpose.

1.2.8 PSM-level Performance Simulation

Given a PSM with detailed technology information, the designer conducts at this level several performance simulations in order to evaluate the PSM's response against the requirements. At this step, the performance assessment of the PSM is more accurate than its predecessor (i.e., PIM) due to the availability of technology platform information. If the PSM's performance is judged not satisfactory, several optimization iterations might be conducted in order to enhance the PSM's quality. Once the requirements are met, the PSM can be converted using a PSM-to-PM transformation into a platform model. In practice, this consists to accurately replace PSM elements by their detailed physical artwork artifacts (e.g., layers, materials, physical connections, etc.). The resulting layout is submitted after verification to the manufacturing step. Tools such as layout editors, design rules checkers and layout versus schematic tools can be used to ensure the absence of defects and design constraints violations in the final PM model (i.e., layout).

1.2.9 Manufacturing

Given the final PM, the RF circuit is manufactured, packaged and integrated using the suitable machinery.

1.2.10 Tests and Measurements

The fabricated circuit is then submitted to several tests and measurements in order to validate its actual performance. This takes place using commercial software and hardware tools (such as VNAs). All the measurements are reported and stored in the Q-matrix.

As shown in Figure 1.1, 1.2 and 1.3, the Q-matrix lifecycle starts at the end of the functional description. Following its initial validation, the Q-matrix can be queried and modified at any design step by several parties (including involved designers and tools). This makes the Q-matrix a central piece of the framework that centralizes design data and allows an effective use of it (not only during the design process but also even after manufacturing).

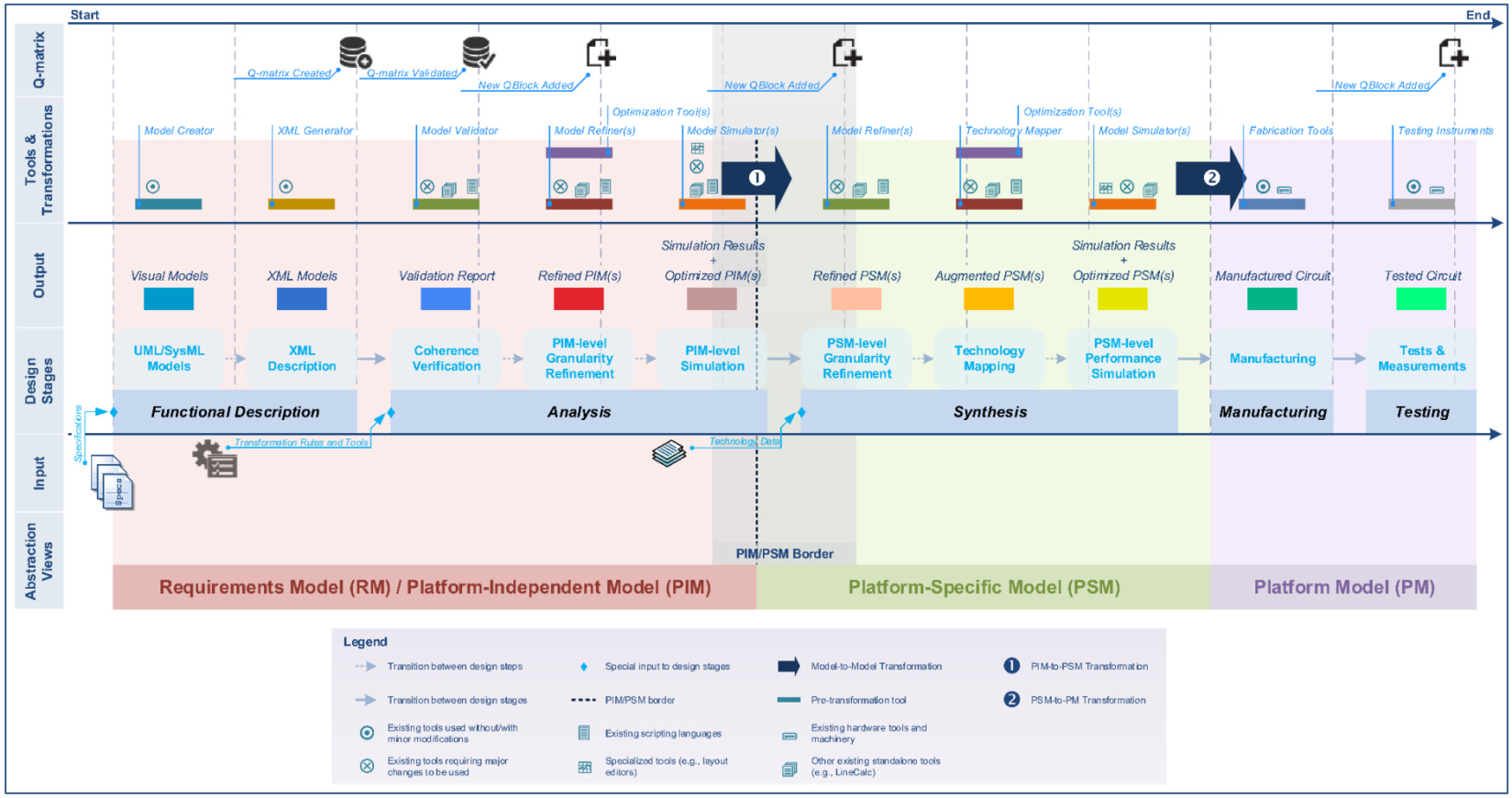


Figure 1.2 Detailed flowchart of the proposed design framework

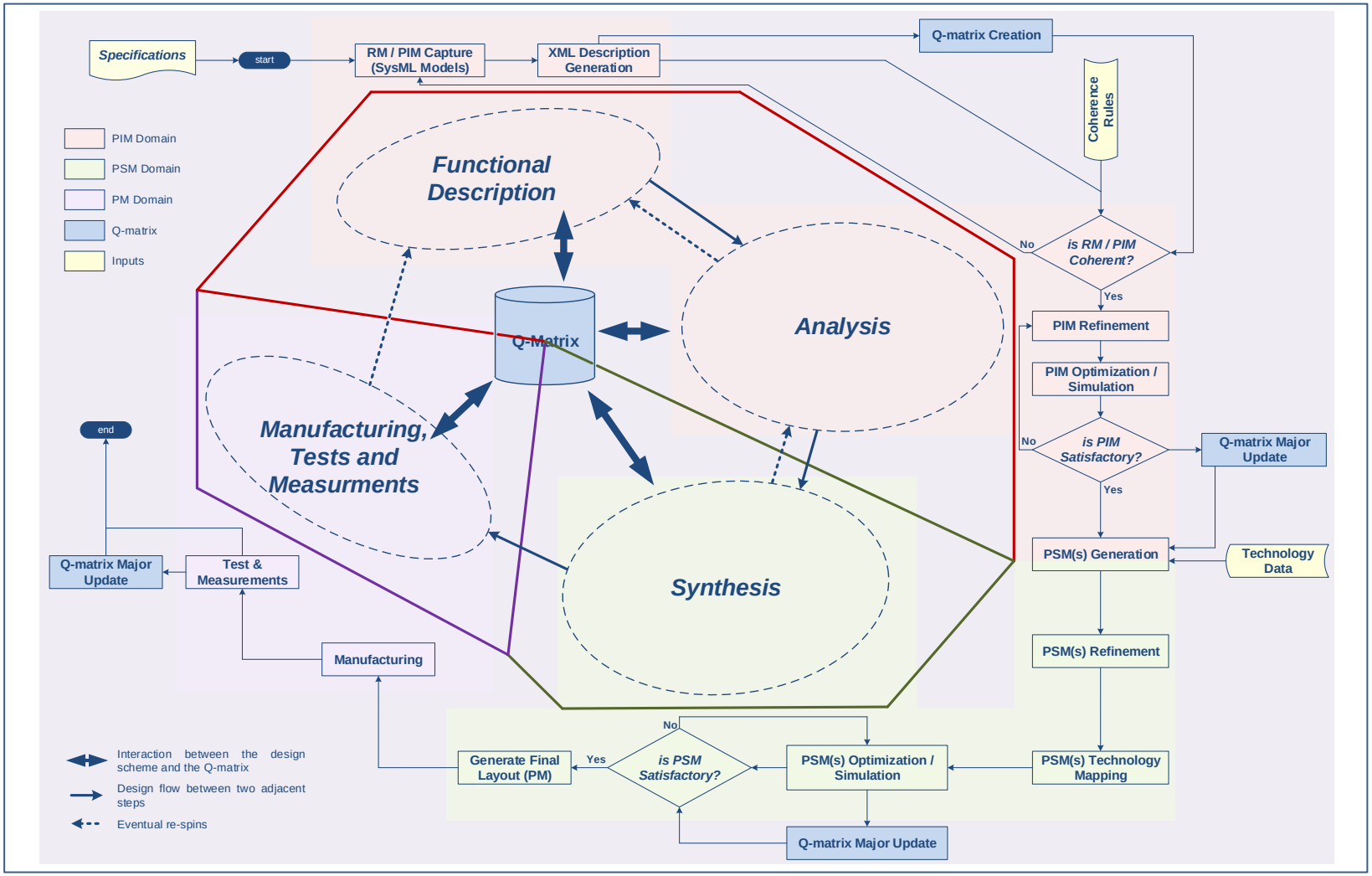


Figure 1.3 A detailed filter design flowchart extracted from the design framework

1.3 Methodology validation through case studies

To validate the proposed design framework, we proceed in this technical report to the implementation of real-world RF functionality (i.e., radiofrequency selection device) through selected case studies. In each case study, we present in some details the design steps, artefacts and simulation results related to the design of such functionality using the proposed methodology.

2 CASE STUDY I: FREQUENCY SELECTION DEVICE

2.1 Introduction

A frequency selection functionality (such as a RF filter) is a RF component that is used within a RF front-end to operate frequency selection. It suppresses signals at undesired frequencies and attempts to accept the wanted signals with the least possible attenuation. Various types of filters exist and can be built using several technologies. Generally, two prevalent approaches are used to design RF filters: Image parameter and insertion loss methods. In this chapter, we consider the traditional filter synthesis process that consists of three main steps:

1. Lowpass prototype design according to the insertion loss method,
2. Filter network transformation (including frequency and impedance scaling), and
3. Filter network realization using lumped elements and/or distributed lines.

For the elaboration of the filter's functional description, we use the SysML profile for RF devices (see details in (Lafi, 2016)) to implement the filter RM/PIM models (the package diagram of Figure 2.1 gives an overview of these models structure).

An acceptable initial design solution satisfying the filter's PIM is then derived using a traditional filter approximation (e.g., Chebyshev, inverse Chebyshev, maximally flat, elliptic, Bessel, etc.). Based on this approximation, an ideal network of resonators (see Figure 2.2) which meets the initial specifications is then easily derived using a relevant PIM-to-PSM transformation. This network represents the filter PSM which is subsequently transformed into an adequate implementation (i.e., PM) given the appropriate technology data. At this regard, a filter can be realized using different technologies (i.e., waveguides, lumped components, LTCC, distributed lines, etc.). Accordingly, technology input can be of various formats. For illustration, see Figure 2.3 shows a distributed-line implementation of a RF filter.

As shown in Figure 1.2, each transformation may require external tools to conduct in part or in whole, PIM-to-PSM and PSM-to-PM transformations (e.g., a tool to generate a distributed-line layout from the corresponding PSM). Ideal resonators' networks are derived from SysML RM/PIM models and filter layout artworks are derived for each PSM. The electrical data are stored in the Q-matrix throughout the design process. Both models and transformations are depicted in Figure 2.4.

Based on the detailed design flowchart of Figure 1.3, the inputs to the functional description and synthesis stages are specifications and technology data respectively. Furthermore, the coherence verification rules are implicitly considered as an input to the coherence verification condition (i.e., entitled "is RM/PIM coherent?"). In addition to the RM/PIM models and XML description, the Q-matrix coherence is checked at the same flowchart condition. The task "Q-matrix Major Update" indicates that a new QBlock might be created to hold data from a new source (e.g., design stage, other designers or tools, etc.).

To put this design process into application, we implement in this chapter, two frequency selection devices. The first is 450-MHz lumped-component bandpass filter for satellite communications. The second is a 1-GHz microstrip lowpass filter for intermediate frequency applications.

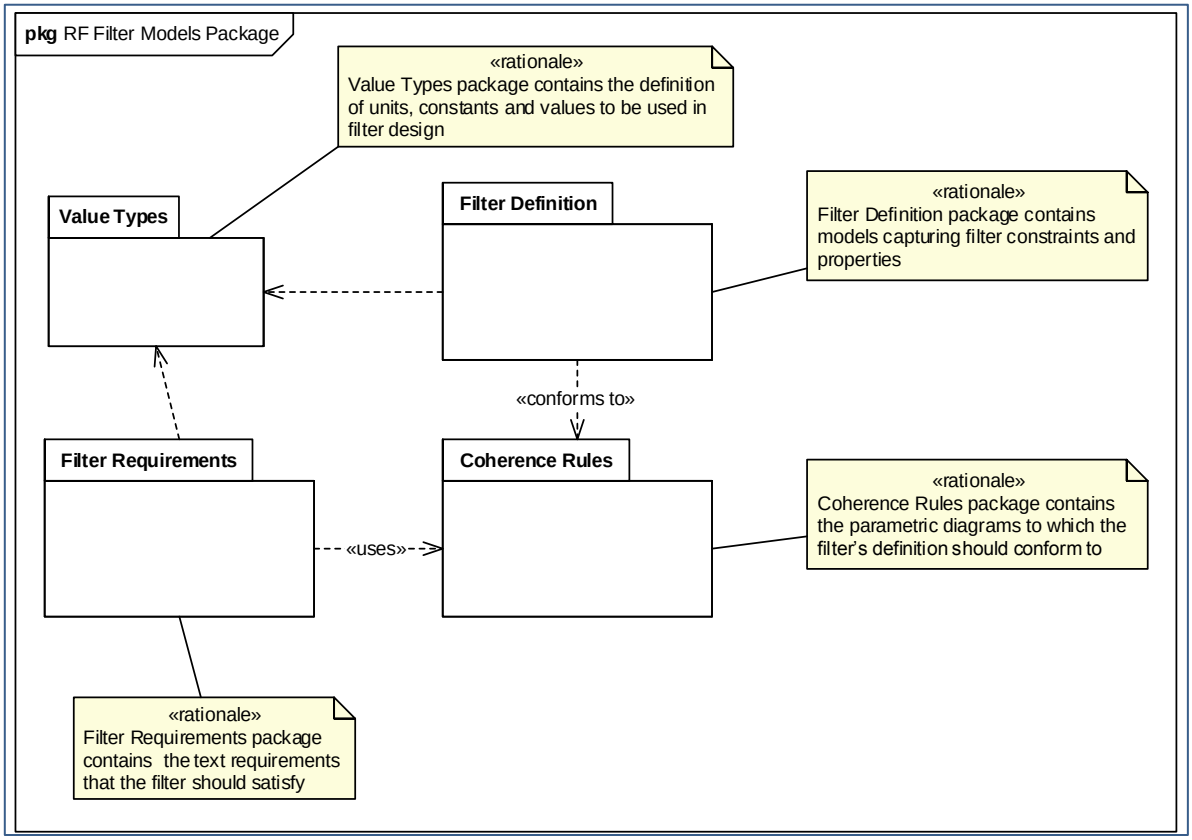


Figure 2.1 This package diagram gives an overview of the filter's SysML RM/PIM models

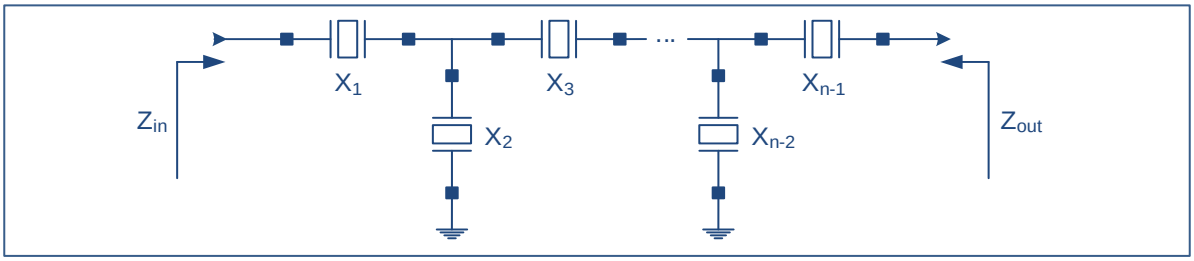


Figure 2.2 The filter's PSM consists of an ideal resonators' network

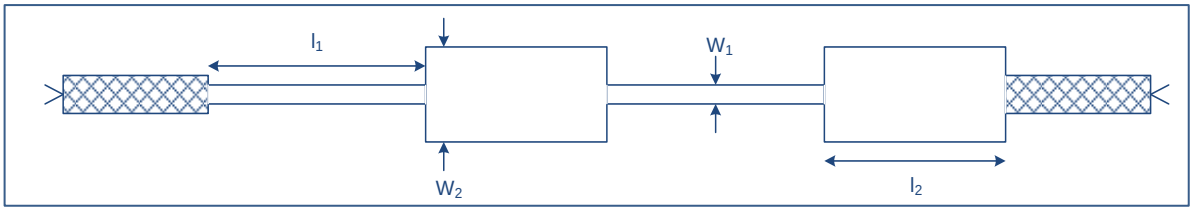


Figure 2.3 A distributed-line layout artwork is a potential filter PM model

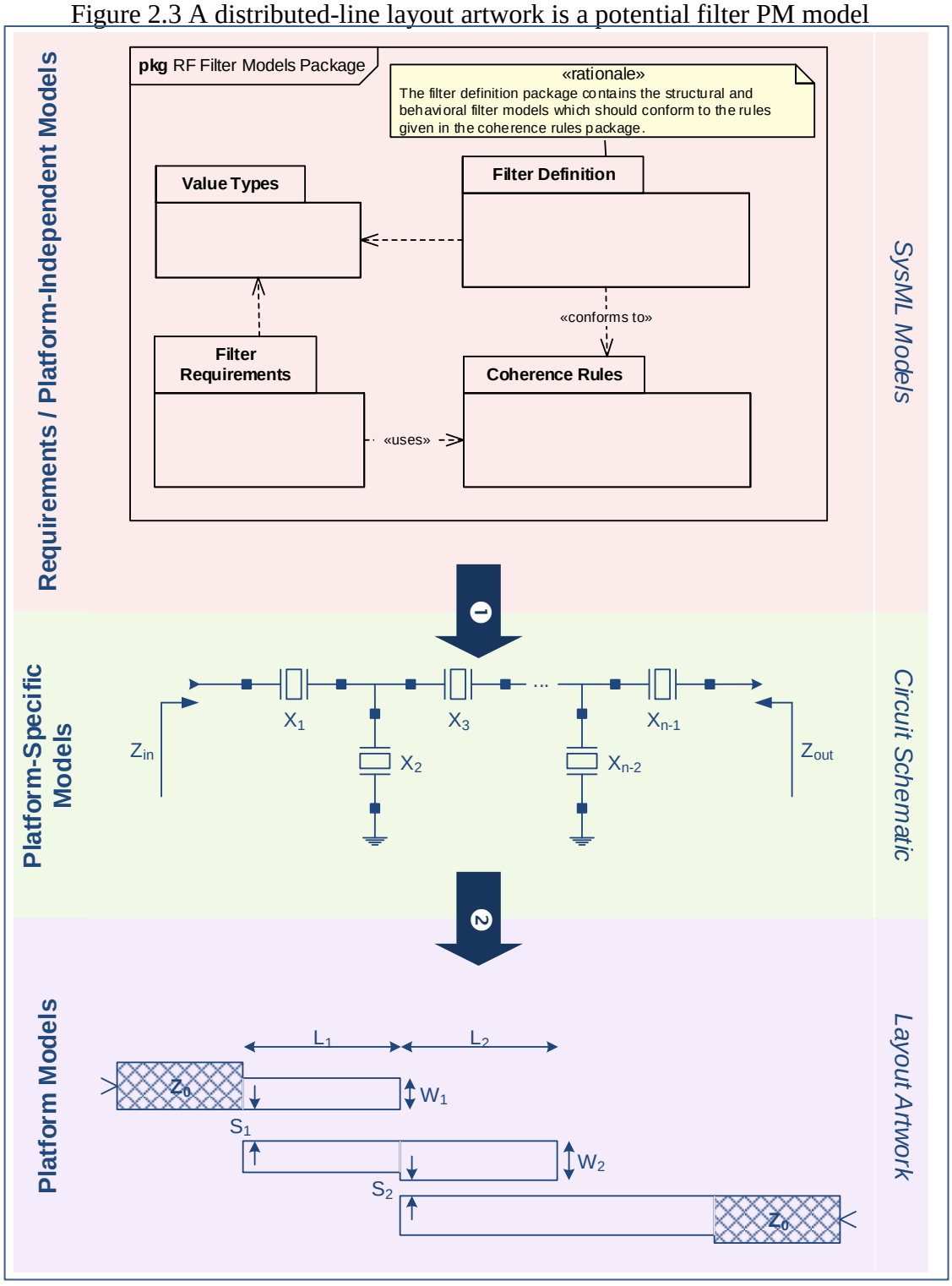


Figure 2.4 Model-to-model transformations convert RM/PIM to PSMs and PSMs into PMs

2.2 A 450-MHz lumped-component bandpass filter

In this first example, we aim to design a bandpass filter for satellite communications. We attempt to detail as much as possible the framework's design steps.

2.2.1 Specifications

A bandpass filter enables the selection of a given frequency band while it suppresses all the remaining lower and higher frequencies (see Figure 2.5). Traditionally, the specifications consist to assign a value to the filter parameters. In this case study, Table 2.1 shows the specifications of the desired bandpass filter.

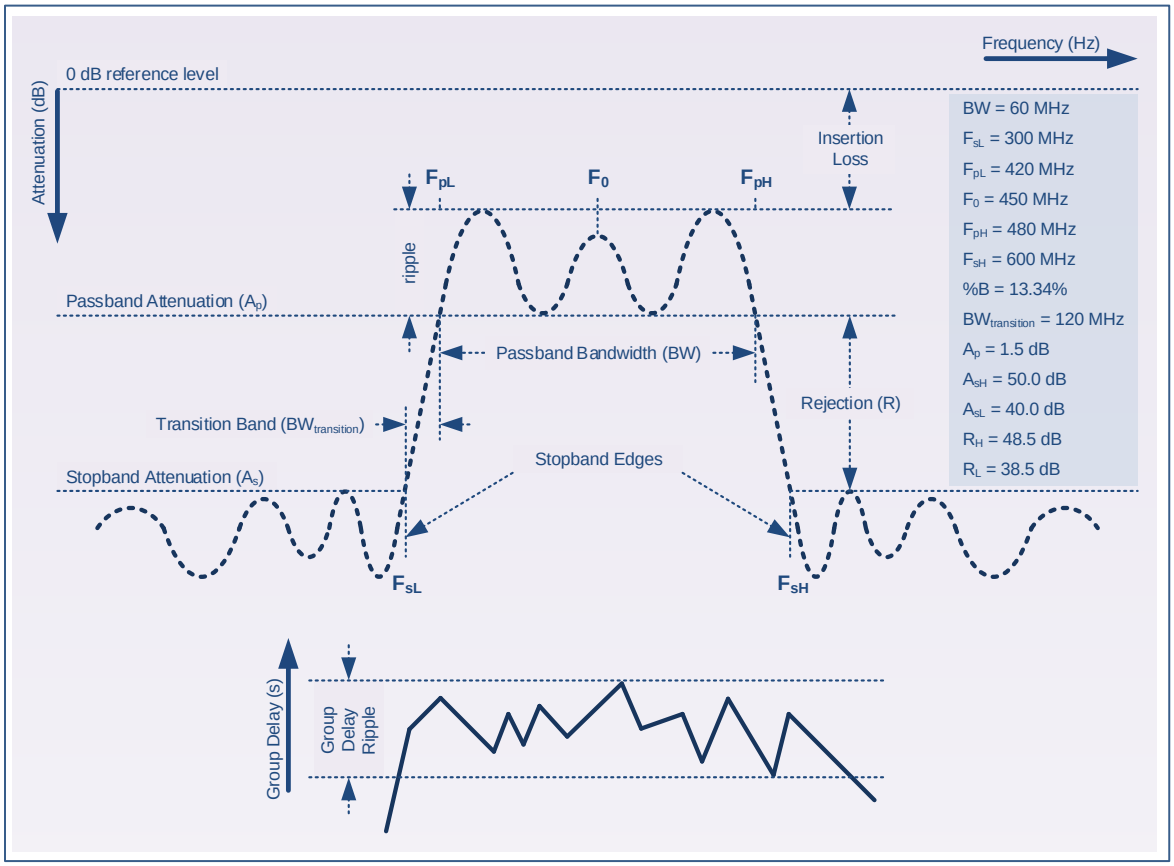


Figure 2.5 The main parameters of a RF bandpass filter

2.2.2 Functional description: UML/SysML Models

Given the specifications of Table 2.1, the first design stage is the functional description of the desired bandpass filter. It consists of elaborating the RM/PIM models and generating the corresponding XML description.

Table 2.1 Typical bandpass filter requirements for a 450-MHz satellite radio

Filter Type	Bandpass
Passband Attenuation (dB)	< 1.5
Stopband Attenuation (dB)	> 40.0 @ 300 MHz
	> 50.0 @ 600 MHz
Upper Passband Edge Frequency (MHz)	480
Lower Passband Edge Frequency (MHz)	420
Upper Stopband Edge Frequency (MHz)	600
Lower Stopband Edge Frequency (MHz)	300
Termination Impedance (ohm)	50
Other Requirements	– Small form factor – Lightening protection – Small radiation fingerprint (MIL-STD-461) – Operating non-condensing humidity up to 80%

- **SysML Models (RM/PIM)**

Using SysML and based on the "SysML profile for RF devices" detailed in (Lafi, 2016), we capture the specifications of Table 2.1 in RM/PIM models. These models do not only capture the filter parameters, structure and behavior but also other requirements. To capture all these design aspects, we define four main SysML packages:

- Filter Definition: contains the models which describe the filter properties and hierarchy,
- Filter Requirements: captures other text requirements related to the filter,
- Coherence verification rules: assembles the design constraints and the coherence rules that should be satisfied by the other packages (particularly the filter definition), and
- Value Types: defines the units, constants, values and other domain-specific artifacts related to filter design.

The package diagram of Figure 2.6 presents an overview of the relationships between the four packages. Both filter "Definition" and "Requirements" packages are associated to the "Value Types" package with a dependency relationship denoting that the models of the former packages use the definitions introduced within the latter. Furthermore, the models contained in the "Filter Definition" package should satisfy the rules and constraints captured in the "Coherence Rules" package. This is expressed by the association "conforms to" linking both packages. Similarly, the association "uses" indicates that the "Filter Requirements" models use the models within the "Coherence Rules" package.

- **Filter properties and constraints**

From a functional viewpoint, a filter is a frequency selection device. Its frequency response is traditionally of four types: lowpass, highpass, bandpass or bandstop. A custom filter can also be defined by combining two or more among these traditional frequency responses.

From an architectural viewpoint, a filter is a two-port linear network. It is composed of two ports through which RF signals come in and go out of the filtering circuit. The SysML block definition diagram (bdd) of Figure 2.7 presents a hierarchy of blocks that derives the bandpass filter functionality. Starting at the "Generic Filter" block, this

diagram indicates that inherits the properties of a “Two-Port Network” (including its two ports) and “Linear Device” blocks. In addition, the “Generic Filter” block functionally and semantically representing a frequency selection device can be specialized in three other devices. The first is “Mid-Band Filter” representing bandpass and bandstop filters. The second is “Single-Side Filter” representing highpass and lowpass filters. The third is “Custom” block that captures complex-response filters (e.g., a dual-band filter).

The bdd of Figure 2.7 is detailed in the bdds of Figure 2.8 and Figure 2.9. The blocks of each of these diagrams are expanded. Except the leaf blocks (i.e., lowpass, highpass, bandpass and bandstop filter) having no specific value properties, the other blocks have four compartments:

- Values: It contains the block’s value properties. Each among these parameters has a given type that is defined in the “Value Types” package (illustrated in Figure 2.10);
- Default values: Some outstanding value properties are assigned to default values. These defaults are overridden by the values given in the specifications. If no values are specified, the defaults are considered in the design only if they do not cause any inconsistencies in coherence verification;
- Weights: The value properties to be considered in the selection process of a candidate design solution are assigned constant weights. Each weight indicates the importance of the corresponding value property;
- Constraints: It is a set of rules that defines how the variable weights are calculated for each value property considered in the previous compartment.

It is worth noting that the generic bdd of Figure 2.7 is expanded in two detailed bdds (see Figure 2.8 and Figure 2.9) only for clarity.

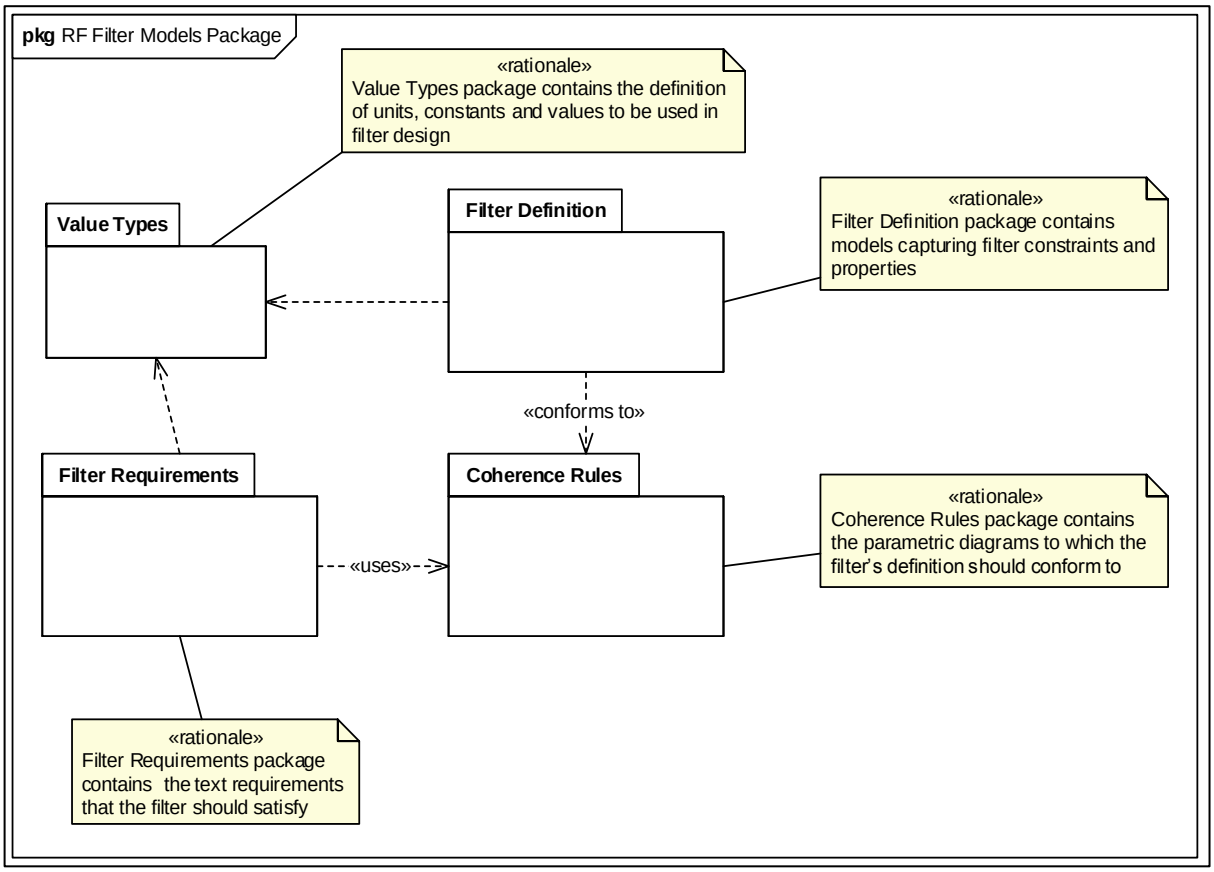


Figure 2.6 An overview of filter RM/PIM models developed in SysML

As previously mentioned, any block which represents a specialization of another one inherits its properties (i.e., values, default values, weights and constraints) including those constructed by a specific relationship with other blocks (e.g., composition relationship relating the "Two-Port Network" block to its two ports). The complete list of a bandpass filter value properties are enumerated in Table 2.2 and Table 2.3.

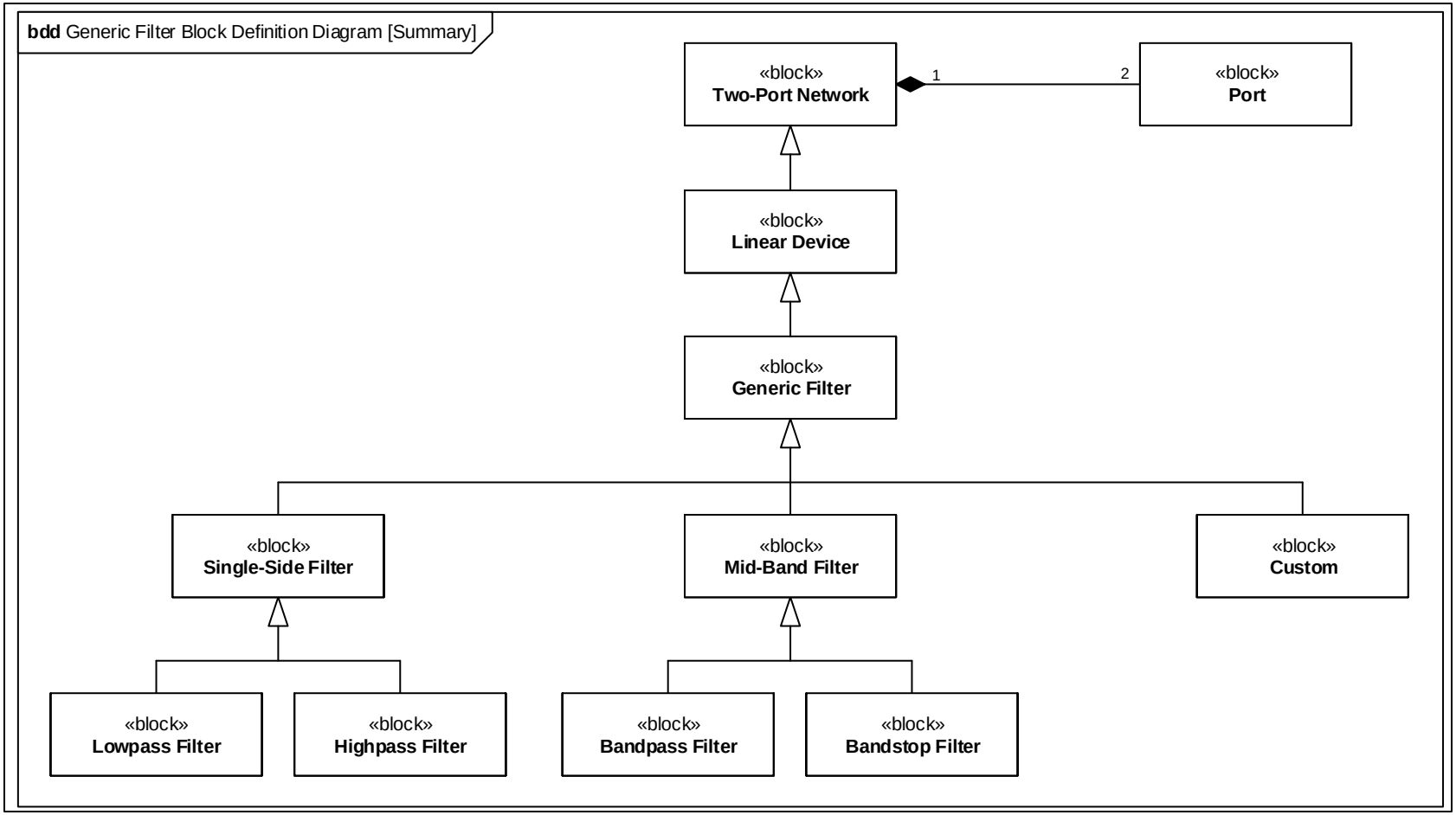


Figure 2.7 A filter is a two-port linear device

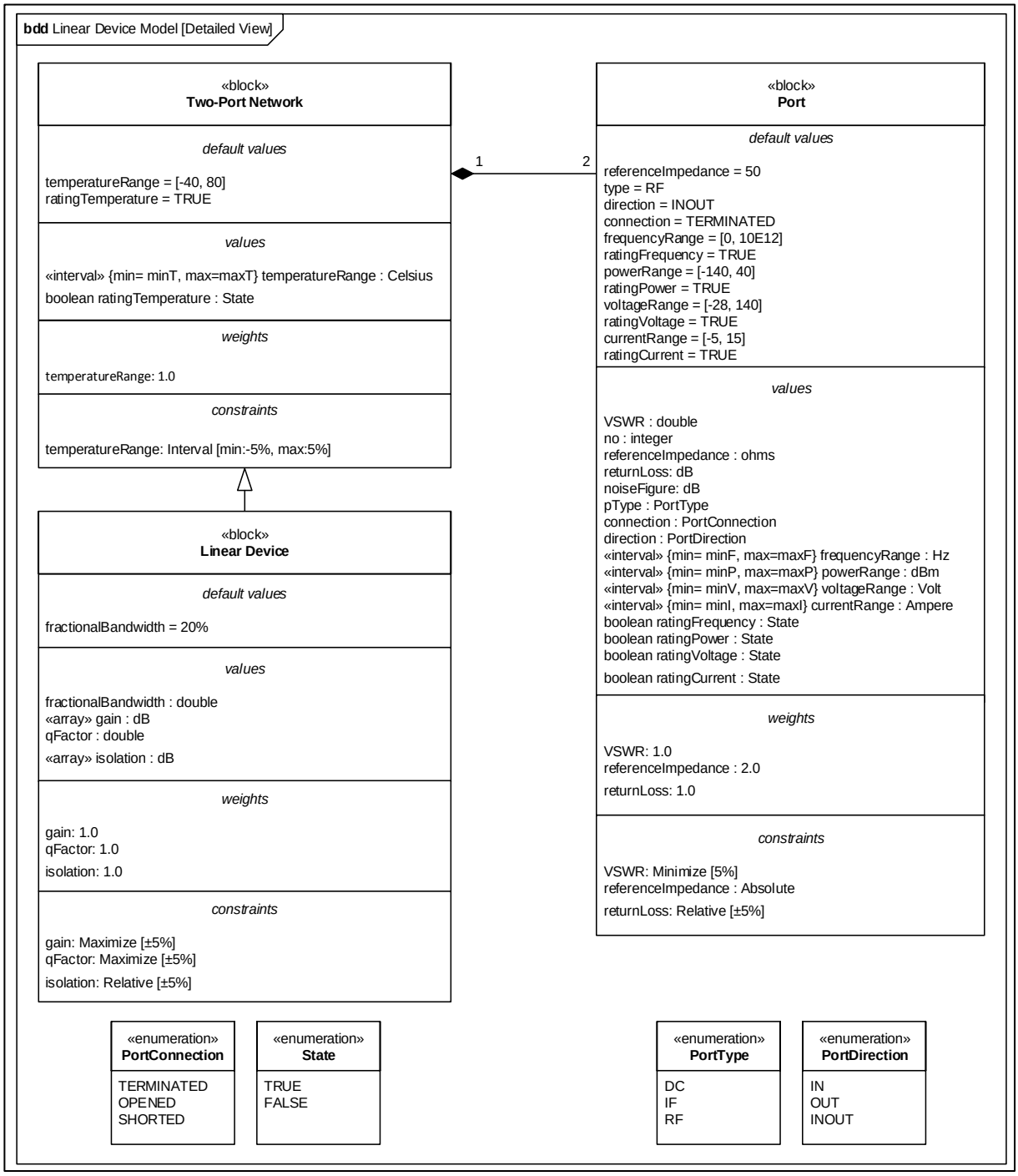


Figure 2.8 Linear device detailed block definition diagram

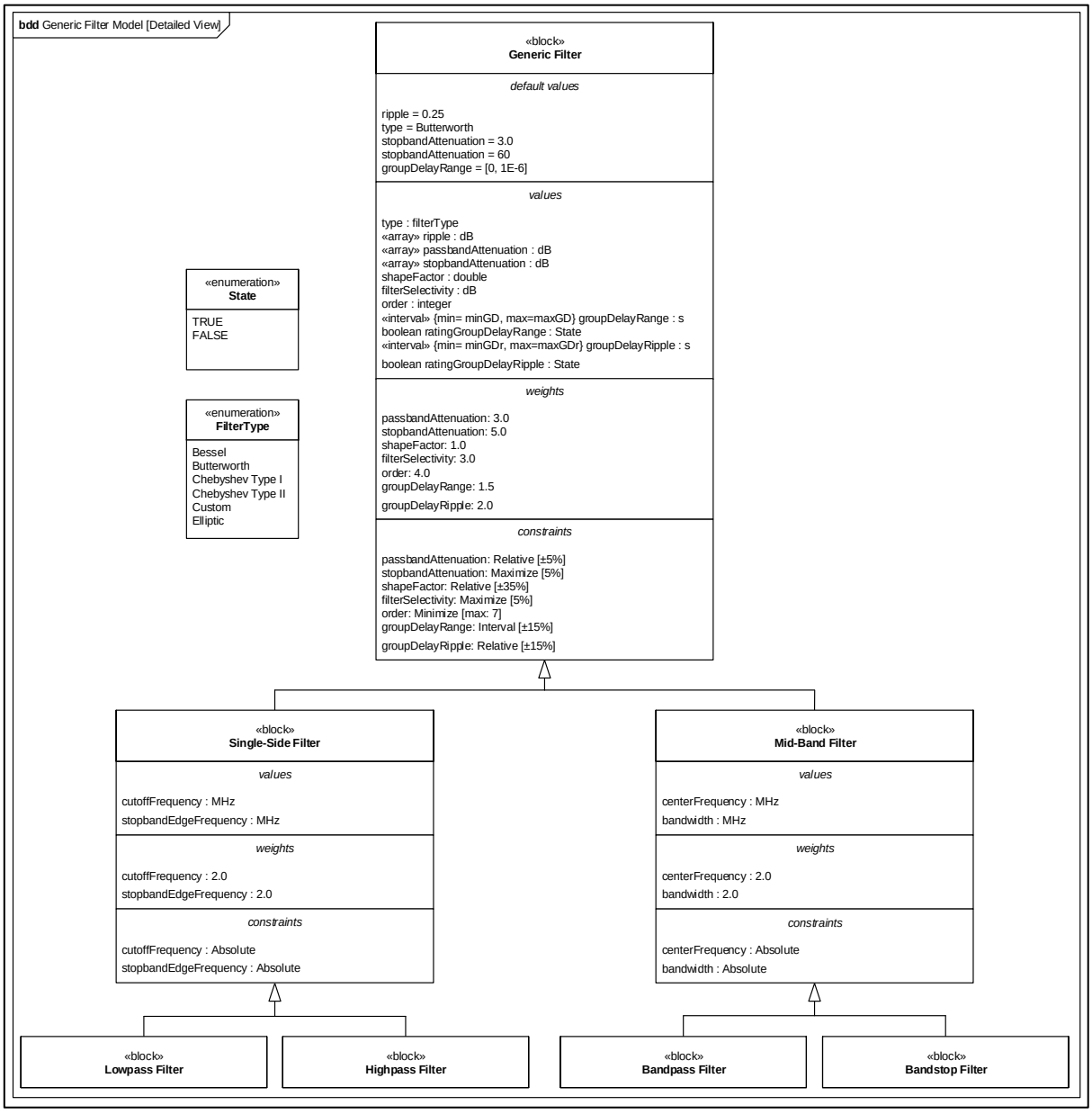


Figure 2.9 Generic filter detailed block definition diagram

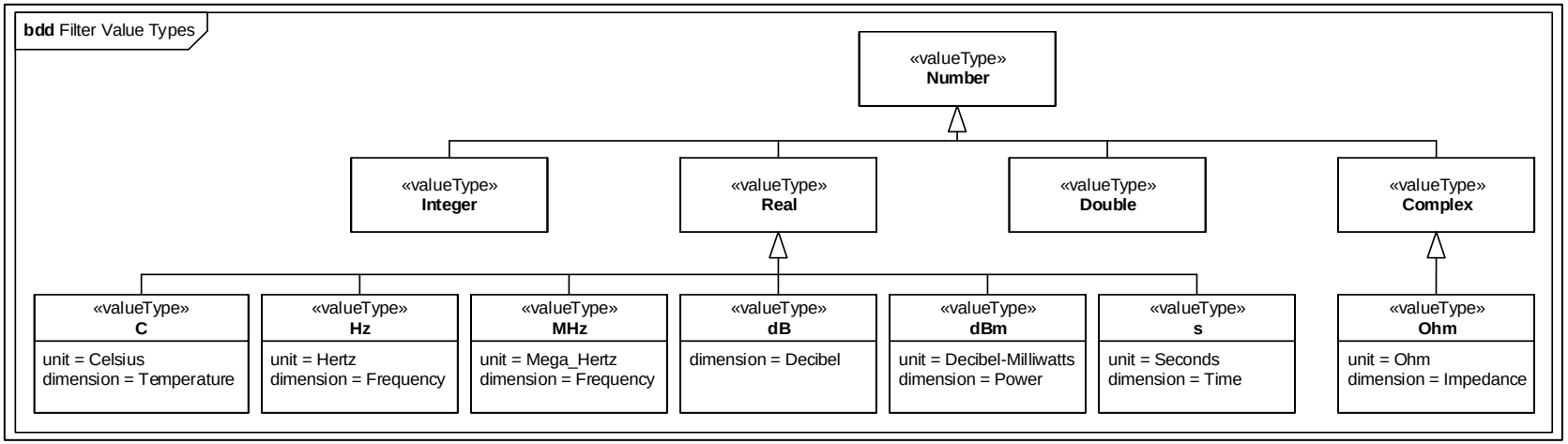


Figure 2.10 Value types captured in a bdd for the filter's RM/PIM models

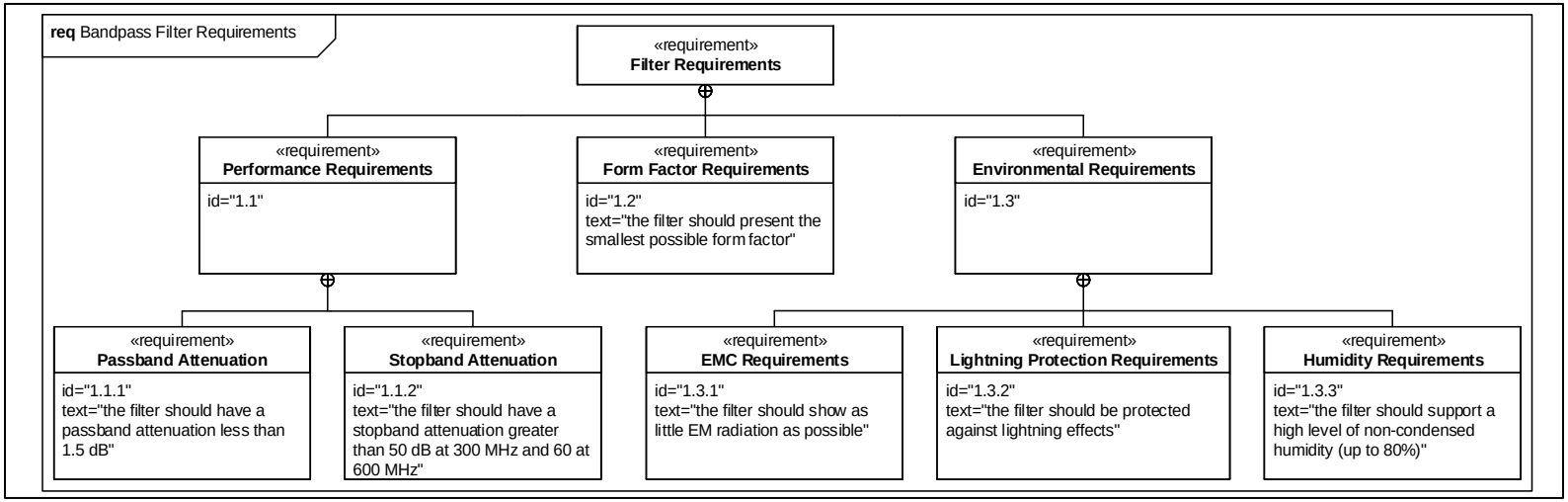


Figure 2.11 Detailed filter requirements diagram

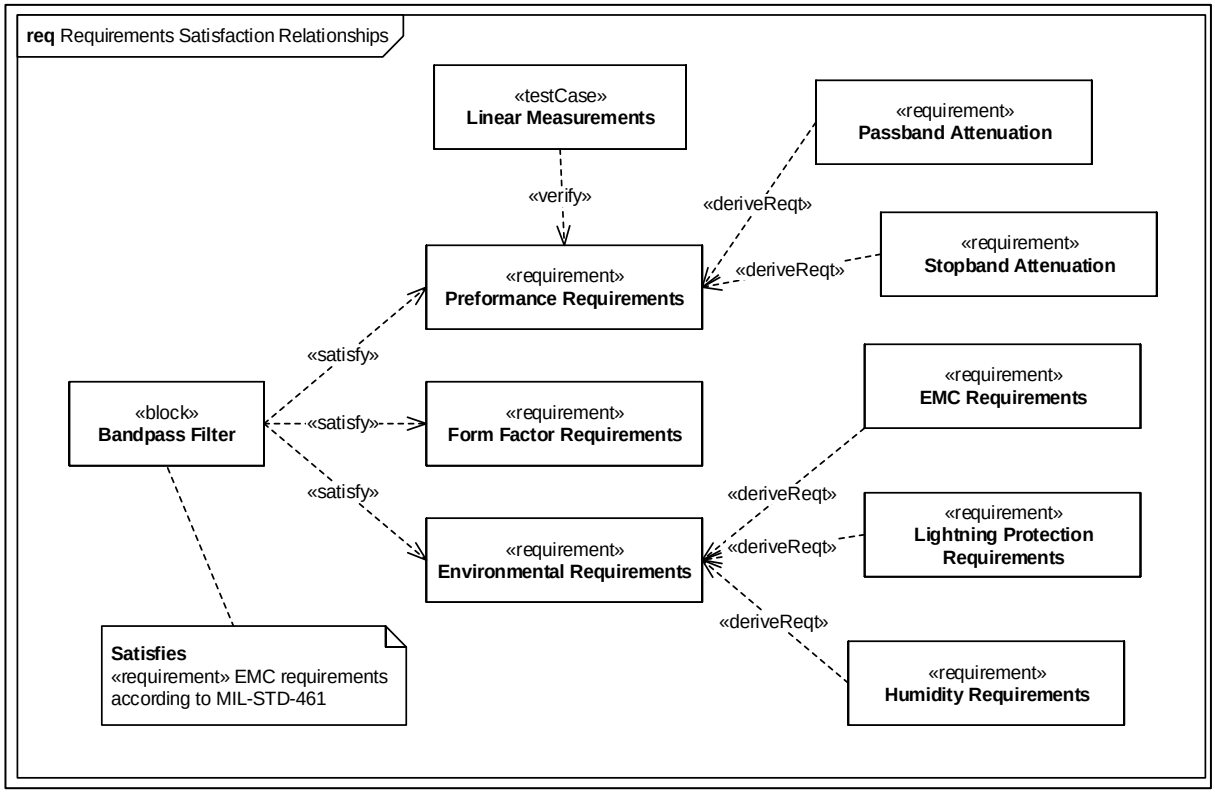


Figure 2.12 The bandpass filter might be explicitly associated to its requirements and testcases

Table 2.2 List of bandpass filter value properties as presented in the bdds of Figure 2.8 and 2.9

Value Property	Type	Default Value	Remarks	Specifications Value
temperatureRange	interval (Celsius)	[-40, 80]		
ratingTemperature	Boolean	TRUE	TRUE FALSE	
portList	Object (Port)			
portsNo	integer	2		
fractionalBandwidth	double	20%		13.34%
gain	array (dB)			Overridden by *passbandAttenuation*

Value Property	Type	Default Value	Remarks	Specifications Value
				and *stopbandAttenuation*
qFactor	double			
isolation	dB			Overridden by *filterSelectivity*
type	filterType	Butterworth	Bessel Butterworth Chebyshev Type I Chebyshev Type II Custom Elliptic	
ripple	array (dB)	0.25		
passbandAttenuation	array (dB)	3.0		
stopbandAttenuation	array (dB)	60.0		40.0 @300 MHz 50.0 @600 MHz
groupDelayRange	interval (seconds)	[0, 5E-6]		
ratingGroupDelay	Boolean		TRUE FALSE	
groupDelayRipple	interval (seconds)			
ratingGroupDelayRipple	Boolean			
order	integer			
shapeFactor	double			
filterSelectivity	dB			
centerFrequency	MHz			450.0
bandwidth	MHz			60.0

Table 2.3 List of bandpass filter port value properties as presented in the bdd of Figure 2.8

Value Property	Type	Default Value	Remarks	Specifications Value
VSWR	double			
no	integer			1 (2)
referenceImpedance	double (ohms)	50.0		50.0 (50.0)
returnLoss	dB			
noiseFigure	dB			
pType	PortType	RF	DC IF RF	
direction	PortDirection	INOUT	IN OUT INOUT	
connection	PortConnection	TERMINATED	TERMINATED OPENED SHORTED	
frequencyRange	interval (Hz)	[0, 10E12]		
ratingFrequency	Boolean	TRUE	TRUE FALSE	
powerRange	interval (dBm)	[-140, 40]		
ratingPower	Boolean	TRUE	TRUE FALSE	
voltageRange	interval (Volts)	[-28, 140]		
ratingVoltage	Boolean	TRUE	TRUE FALSE	
currentRange	interval (Amperes)	[-5, 15]		
ratingCurrent	Boolean	TRUE	TRUE FALSE	

- **Filter requirements**

The specifications of the filter are not exclusively composed of properties that can be captured in the PIM structural models. It might also have additional requirements that cover other aspects of the filter operation (e.g., operation environment). These requirements (mostly text-based) can be captured using the SysML requirements diagram. For instance, the last row of Table 2.1 enumerates requirements related to the form factor of the bandpass filter and its operation environment conditions. The SysML requirements diagram allows visualizing these requirements in a comprehensive graphical hierarchy (see Figure 2.11). It enables also to associate the requirements as well as the corresponding testcases to each block using the relevant relationships (see Figure 2.12). Such diagrams contribute significantly to better communication between designers and requirements traceability.

The requirements diagram of Figure 2.11 subdivides the filter requirements into three categories:

– Performance requirements: including passband and stopband attenuation specifications,
– Form Factor requirements: representing form factor constraints, and
– Environmental requirements: covering radiation and humidity operating conditions as well as lightning protection.

As shown in Figure 2.12, the requirements can be associated to the filter blocks with specific relationships for validation and verification purposes. For example, the “satisfy” relationship means that the filter block should satisfy the indicated requirements. However, the “verify” relationship associates a testcase to a requirement.

- **PIM coherence rules**

The coherence rules are used to verify the consistency of the functional description. This input can either be provided within the RM/PIM or separately. In the case study, we

include the coherence rules in SysML models as illustrated in Figure 2.13. To do so, we use the parametric diagram to capture the various equations, algorithms and constraints.

The coherence rules captured in the parametric diagrams of Figure 2.14, Figure 2.15 and Figure 2.15 will be detailed in the next design stage.

2.2.3 Functional Description: XML description

SysML models developed in the previous design step are ideal for visualization but require to be saved in a comprehensive file format for exchange and automated processing. To this end, we use the standard language XML to format both models and design data.

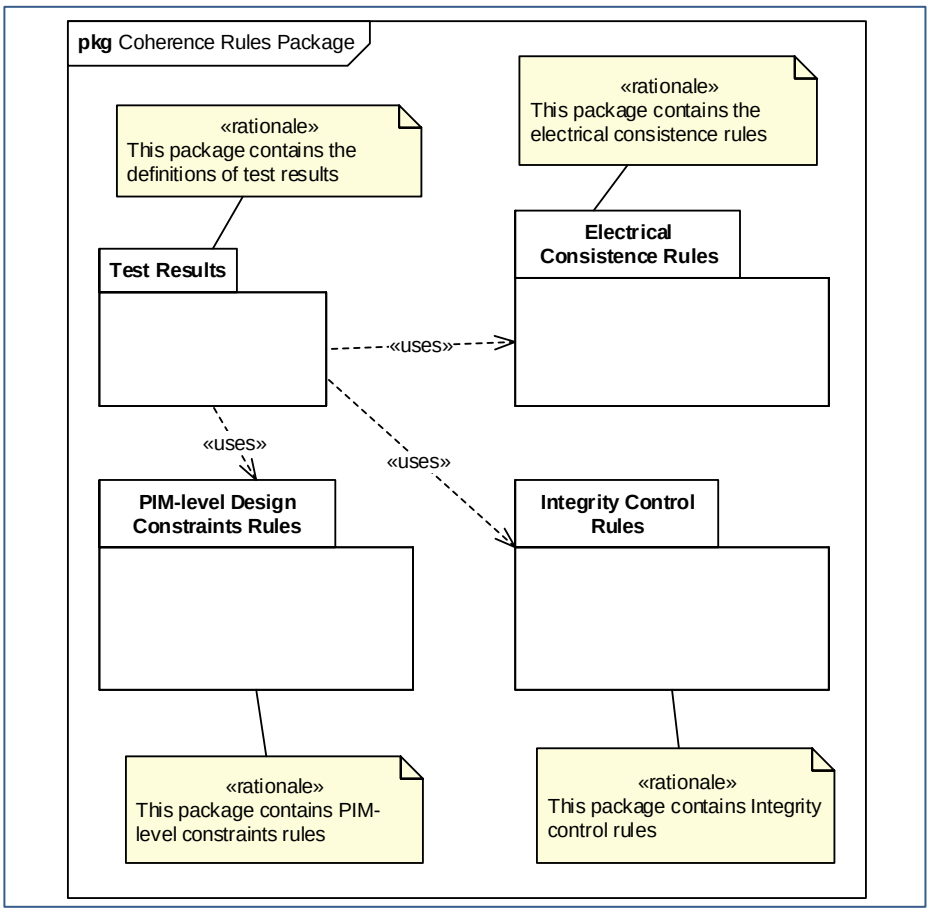


Figure 2.13 The bandpass filter coherence rules package consists of four interrelated sub-packages

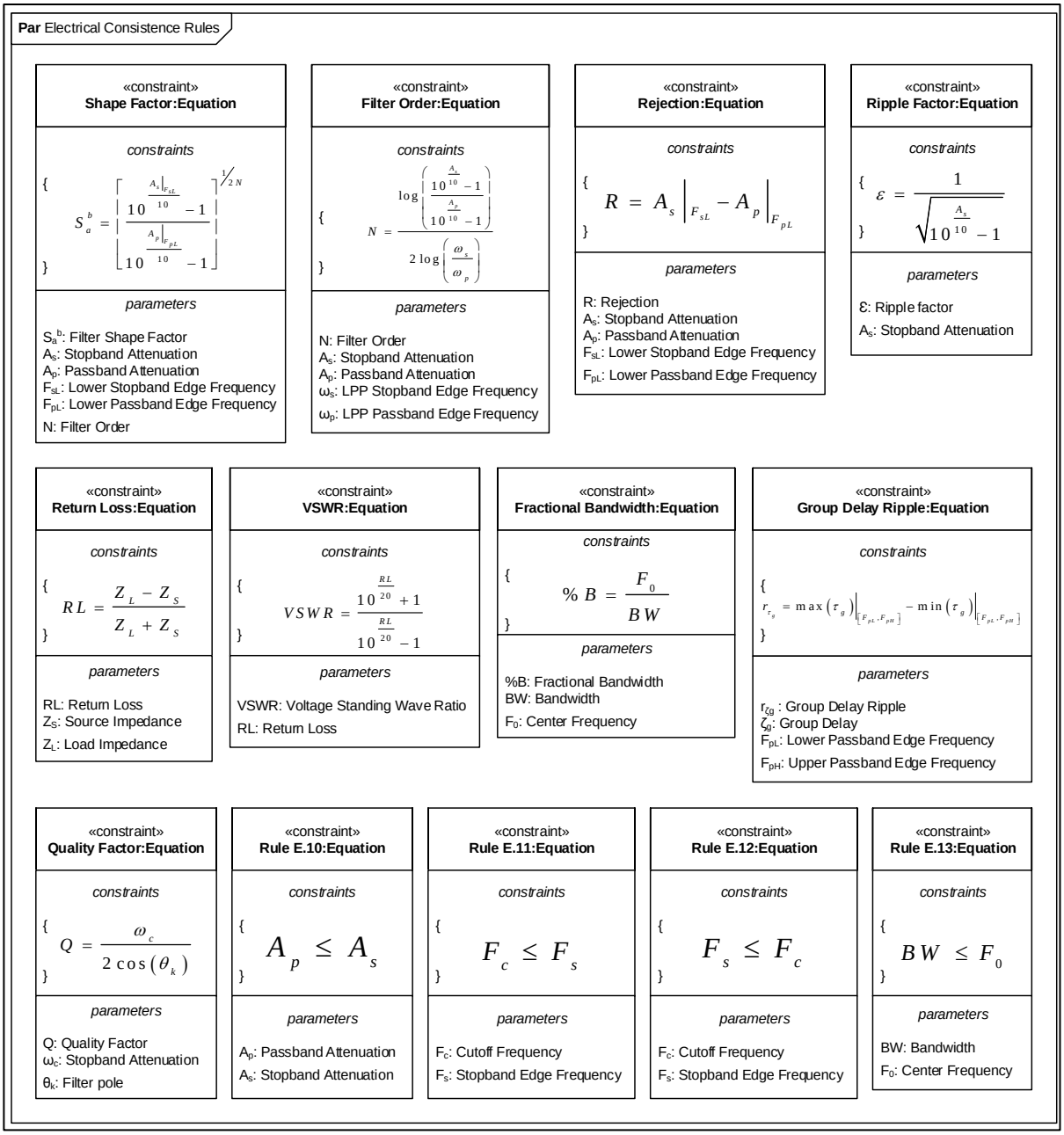


Figure 2.14 Electrical consistence rules illustrated in a parametric diagram

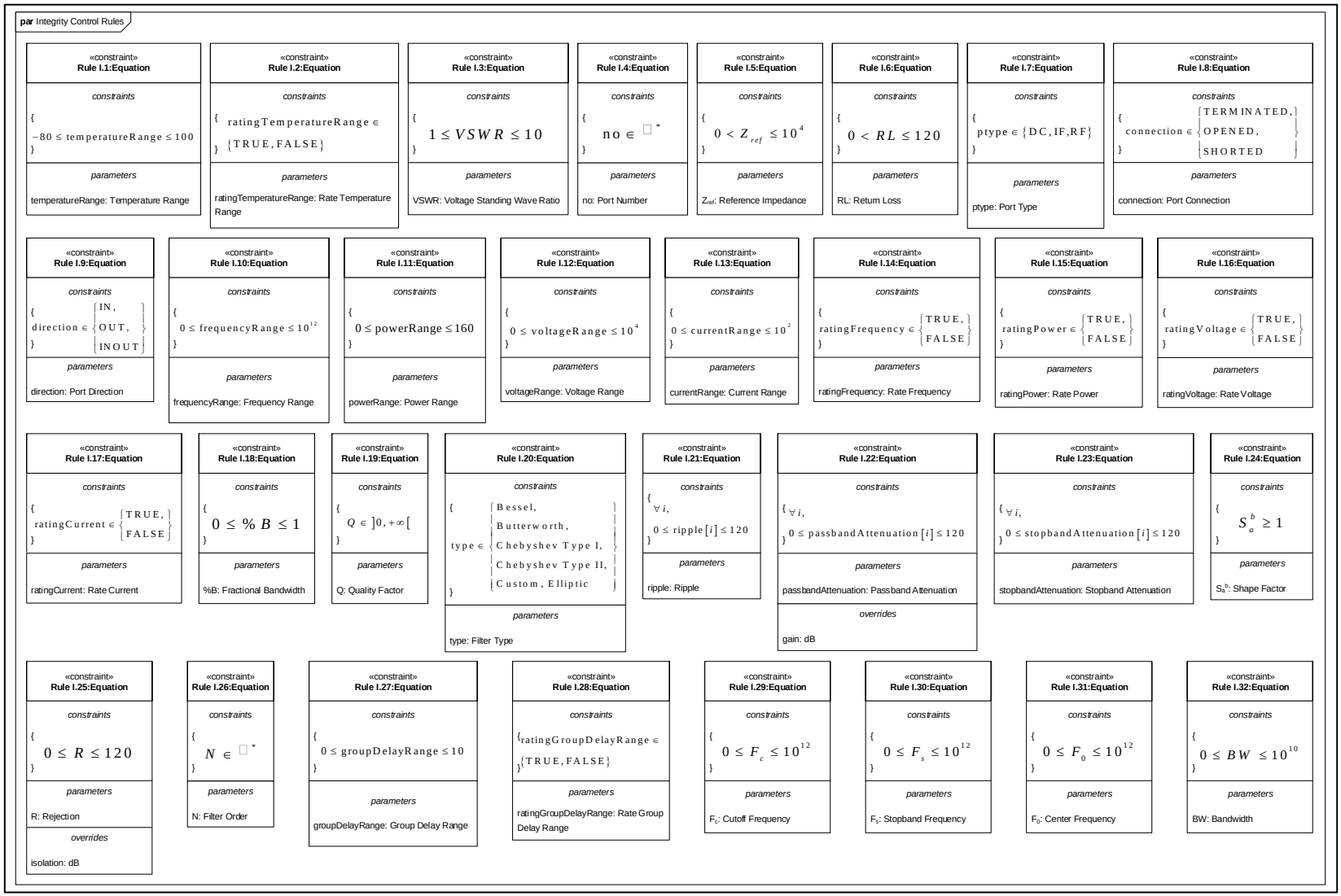


Figure 2.15 Integrity control rules captured using in a parametric diagram

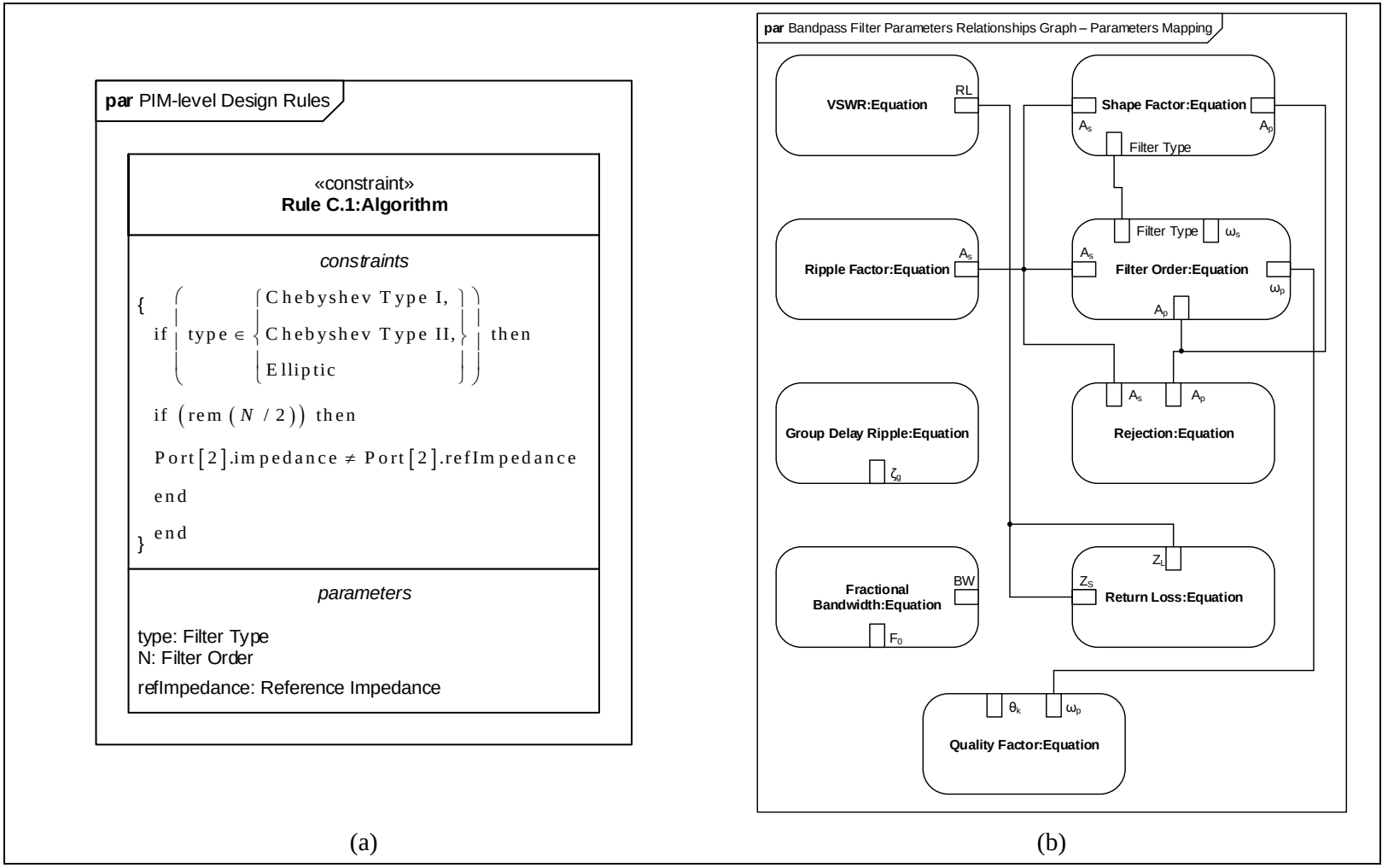


Figure 2.16 Parametric diagram has multiple uses (a) PIM-level design constraints, (b) Mapping of equations' parameters

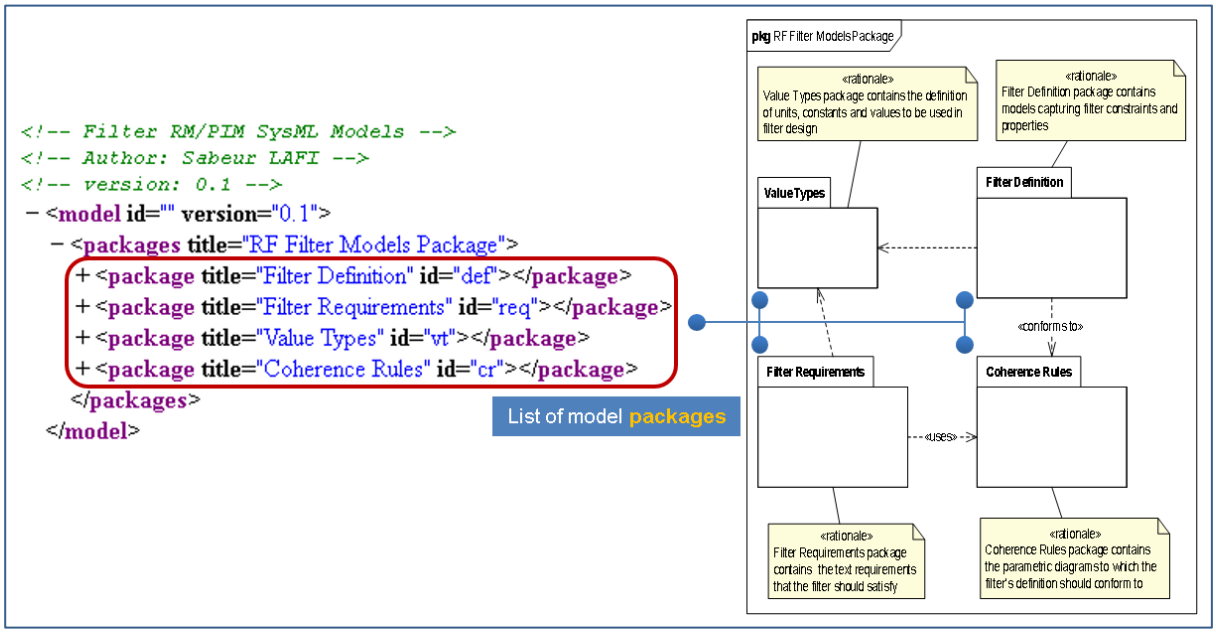


Figure 2.17 Overview of the RM/PIM's XML description file

- **RM/PIM description**

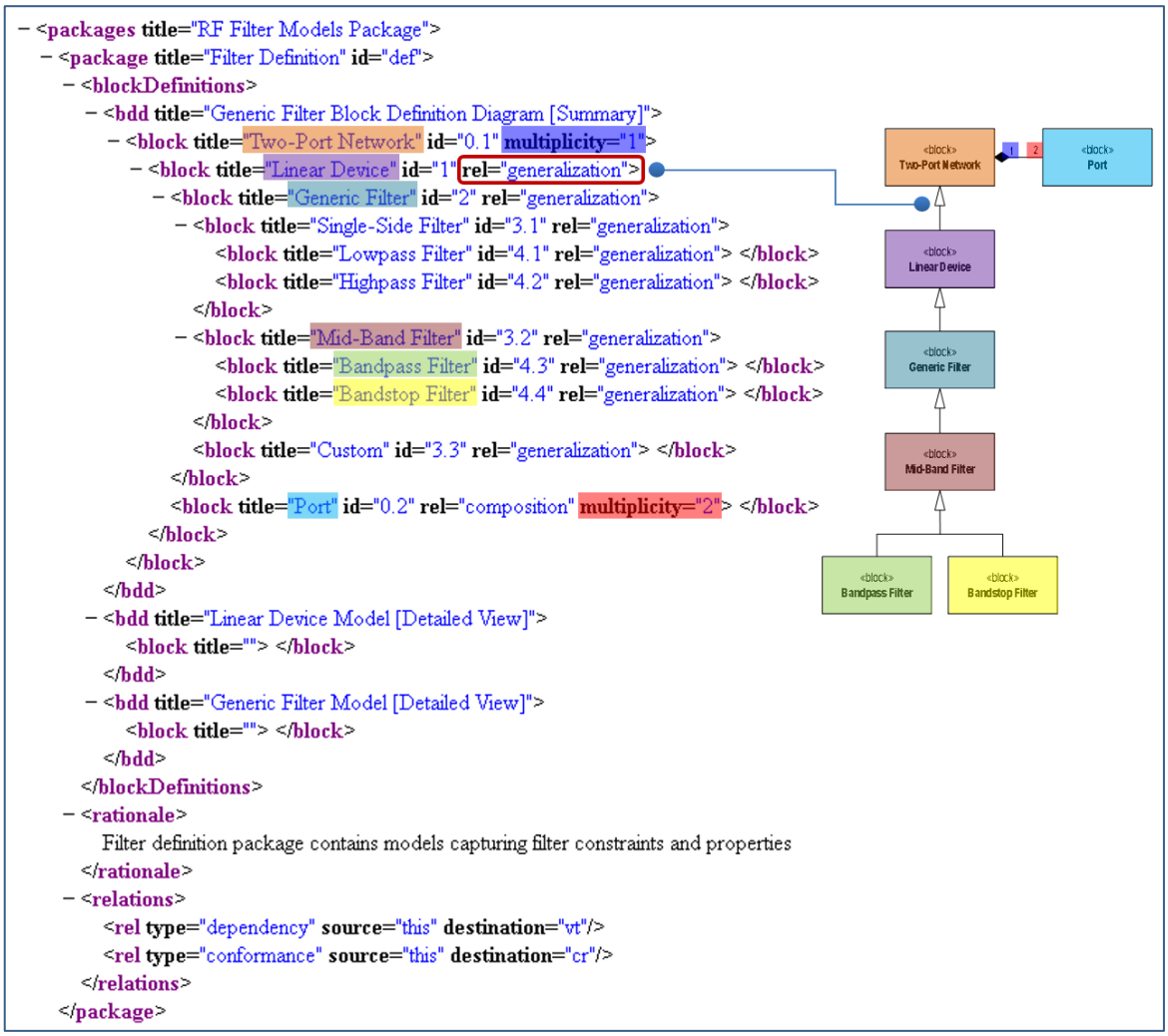
The models developed for the functional description of the bandpass filter are formatted in XML. The Figure 2.17 shows the resulting XML file. A branch of the XML tree is dedicated to each modeling package (e.g., filter definition, filter requirements, coherence rules and value types).

As illustrated in Figure 2.18, the "Filter Definition" package is captured in the XML tag "package" whose title is "Filter Definition". The hierarchy of blocks, their relationships and their multiplicity are transcribed using XML.

Figure 2.19 shows the XML structure capturing the "Filter Requirements" including the entire hierarchy of the requirements and their relationships. The color-shaded sections are mapped to their corresponding blocks in the SysML model. Similarly, value types and coherence rules are faithfully transliterated using XML (see Figure 2.20 and Figure 2.21).

- **Q-matrix creation**

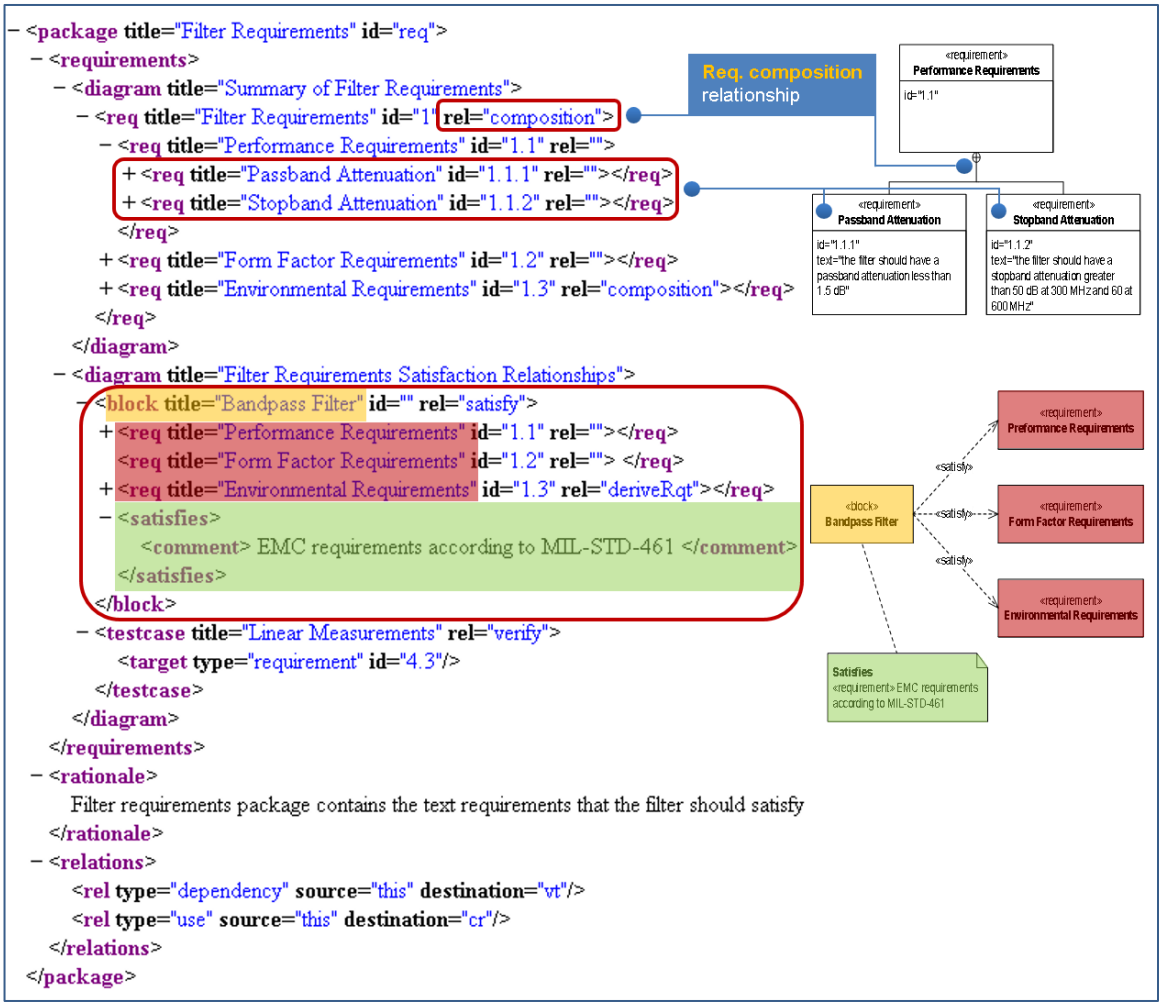
Once the XML description is generated, the Q-matrix is created. Its initial version contains a QBlock that is populated with the electrical parameters captured in the functional description. At this level, there is no guarantee that the Q-matrix is coherent because the RM/PIMs are not yet submitted to coherence verification. That is why the Q-matrix should be also verified and updated if any errors are detected later in the stage of coherence verification.




```
– <packages title="RF Filter Models Package">
  – <package title="Filter Definition" id="def">
    – <blockDefinitions>
      – <bdd title="Generic Filter Block Definition Diagram [Summary]">
        – <block title="Two-Port Network" id="0.1" multiplicity="1">
          – <block title="Linear Device" id="1" rel="generalization">
            – <block title="Generic Filter" id="2" rel="generalization">
              – <block title="Single-Side Filter" id="3.1" rel="generalization">
                  <block title="Lowpass Filter" id="4.1" rel="generalization"> </block>
                  <block title="Highpass Filter" id="4.2" rel="generalization"> </block>
                </block>
              – <block title="Mid-Band Filter" id="3.2" rel="generalization">
                  <block title="Bandpass Filter" id="4.3" rel="generalization"> </block>
                  <block title="Bandstop Filter" id="4.4" rel="generalization"> </block>
                </block>
                <block title="Custom" id="3.3" rel="generalization"> </block>
              </block>
              <block title="Port" id="0.2" rel="composition" multiplicity="2"> </block>
            </block>
          </block>
        </bdd>
      – <bdd title="Linear Device Model [Detailed View]">
          <block title=""> </block>
        </bdd>
      – <bdd title="Generic Filter Model [Detailed View]">
          <block title=""> </block>
        </bdd>
    </blockDefinitions>
  – <rationale>
      Filter definition package contains models capturing filter constraints and properties
    </rationale>
  – <relations>
      <rel type="dependency" source="this" destination="vt"/>
      <rel type="conformance" source="this" destination="cr"/>
    </relations>
  </package>
```

Figure 2.18 The XML description corresponding to the “Filter Definition” package

As previously mentioned, we use XML for the storage and exchange of the Q-matrix data. In general, the file resulting from Q matrix manipulation is voluminous because it is augmented at each design step with many data points. In this case, we show only some outstanding snippets of that file in order to illustrate how the Q-matrix evolves throughout the design cycle.




```
- <package title="Filter Requirements" id="req">
  - <requirements>
    - <diagram title="Summary of Filter Requirements">
      - <req title="Filter Requirements" id="1" rel="composition">
        - <req title="Performance Requirements" id="1.1" rel="">
          + <req title="Passband Attenuation" id="1.1.1" rel=""></req>
          + <req title="Stopband Attenuation" id="1.1.2" rel=""></req>
          </req>
        + <req title="Form Factor Requirements" id="1.2" rel=""></req>
        + <req title="Environmental Requirements" id="1.3" rel="composition"></req>
        </req>
      </diagram>
    - <diagram title="Filter Requirements Satisfaction Relationships">
      - <block title="Bandpass Filter" id="" rel="satisfy">
        + <req title="Performance Requirements" id="1.1" rel=""></req>
          <req title="Form Factor Requirements" id="1.2" rel=""> </req>
        + <req title="Environmental Requirements" id="1.3" rel="deriveRqt"></req>
        - <satisfies>
            <comment> EMC requirements according to MIL-STD-461 </comment>
          </satisfies>
        </block>
      - <testcase title="Linear Measurements" rel="verify">
          <target type="requirement" id="4.3"/>
        </testcase>
      </diagram>
    </requirements>
  - <rationale>
      Filter requirements package contains the text requirements that the filter should satisfy
    </rationale>
  - <relations>
      <rel type="dependency" source="this" destination="vt"/>
      <rel type="use" source="this" destination="cr"/>
    </relations>
</package>
```

Figure 2.19 The XML description corresponding to the “Filter Requirements” package

As illustrated in the equations (3.2)-(3.3) in (Lafi, 2015) introducing the Q-matrix mathematical formalism, the data captured by the functional description should be

streamlined in order to fit within that mathematical construction before it is inserted in the right QBlock. This is relatively easy to do for linear passive circuits (such as filters) because these devices are always characterized using the scattering parameters which fit directly within the Q-matrix definition.

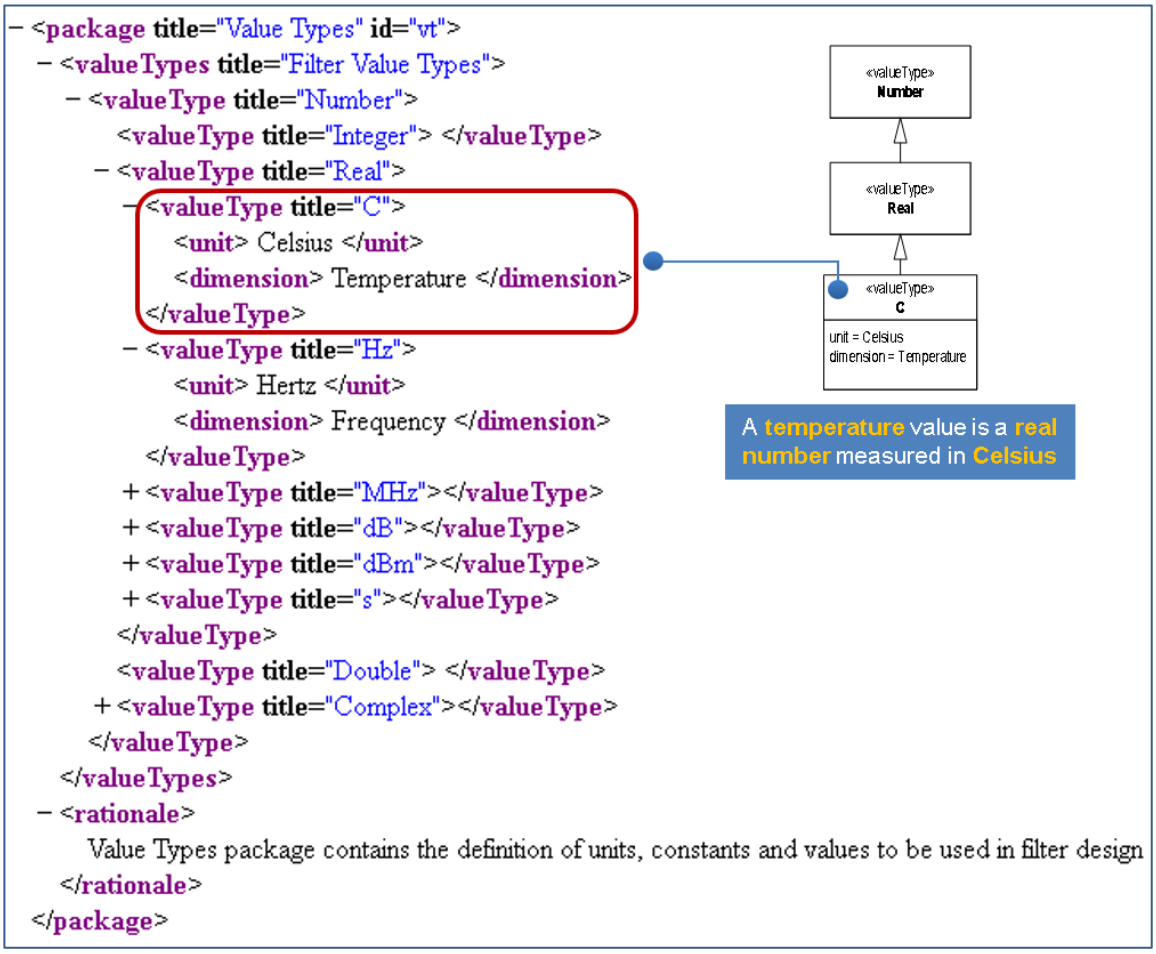
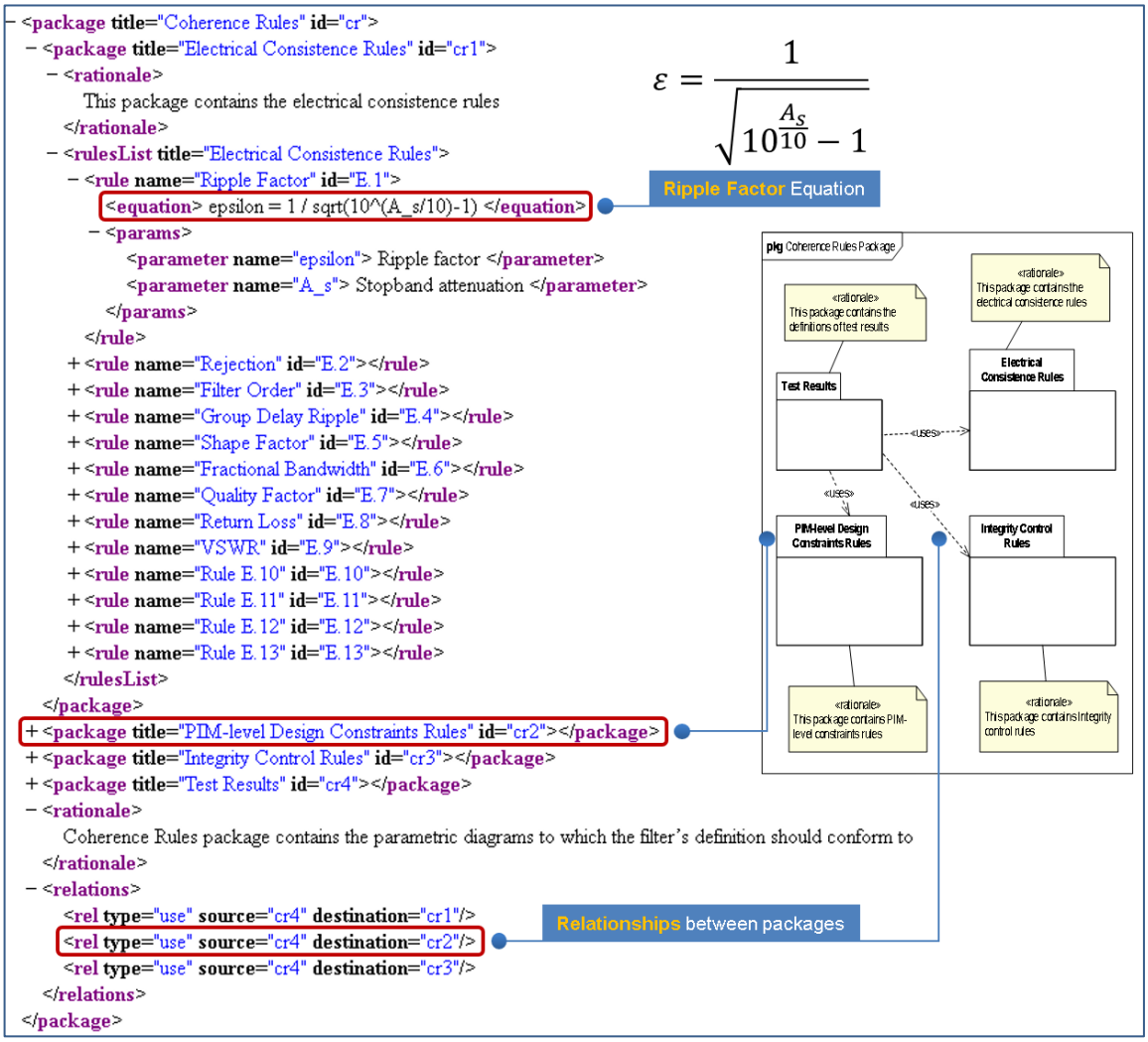


Figure 2.20 The XML description corresponding to the “Value Types” package




```
- <package title="Coherence Rules" id="cr">
  - <package title="Electrical Consistence Rules" id="cr1">
    - <rationale>
        This package contains the electrical consistence rules
      </rationale>
    - <rulesList title="Electrical Consistence Rules">
      - <rule name="Ripple Factor" id="E.1">
          <equation> epsilon = 1 / sqrt(10^(A_s/10)-1) </equation>
        - <params>
            <parameter name="epsilon"> Ripple factor </parameter>
            <parameter name="A_s"> Stopband attenuation </parameter>
          </params>
        </rule>
      + <rule name="Rejection" id="E.2"></rule>
      + <rule name="Filter Order" id="E.3"></rule>
      + <rule name="Group Delay Ripple" id="E.4"></rule>
      + <rule name="Shape Factor" id="E.5"></rule>
      + <rule name="Fractional Bandwidth" id="E.6"></rule>
      + <rule name="Quality Factor" id="E.7"></rule>
      + <rule name="Return Loss" id="E.8"></rule>
      + <rule name="VSWR" id="E.9"></rule>
      + <rule name="Rule E.10" id="E.10"></rule>
      + <rule name="Rule E.11" id="E.11"></rule>
      + <rule name="Rule E.12" id="E.12"></rule>
      + <rule name="Rule E.13" id="E.13"></rule>
      </rulesList>
    </package>
  + <package title="PIM-level Design Constraints Rules" id="cr2"></package>
  + <package title="Integrity Control Rules" id="cr3"></package>
  + <package title="Test Results" id="cr4"></package>
  - <rationale>
      Coherence Rules package contains the parametric diagrams to which the filter's definition should conform to
    </rationale>
  - <relations>
      <rel type="use" source="cr4" destination="cr1"/>
      <rel type="use" source="cr4" destination="cr2"/>
      <rel type="use" source="cr4" destination="cr3"/>
    </relations>
</package>
```

$$\varepsilon = \frac{1}{\sqrt{10^{\frac{A_s}{10}} - 1}}$$

Figure 2.21 The XML description corresponding to the "Coherence Rules" package

In the XML snippet shown in Figure 2.22, a first QBlock named "bpf_specs" is created. It holds the design data already provided in the specifications. In the configuration sub-block (i.e., "config"), the data type is assigned to "S" (i.e., scattering parameters) and format to "MA" (which stands for Magnitude/Angle). Then, the configuration related to each port (i.e., two ports for filters) is inserted as defined in the RM/PIM. In this case, some configuration parameters (e.g., port type and direction) are assigned to the default values given in the RM/PIM because the specifications do not provide explicit values for

them. The next sub-block to be created is the data block. The time and temperature dimensions are set to "U" (i.e., unknown / unspecified) because both of them are still indefinite.

The sub-block "attached ports" captures the configuration of every port for each data item (i.e., q_{ij}). The XML snippet of Figure 2.22 shows only one data item. It specifies the filter's frequency response at the lower passband edge (i.e., 420 MHz). For this reason, the frequency parameter of input and output ports is set to 0.42 GHz. No power level was specified in the RM/PIM at this frequency for both ports, which explains why this parameter is void.

The attenuation level required at 420 MHz corresponds to the passband attenuation (i.e., equal to 1.5 dB1). That is why the transmission coefficients q_{12} and q_{21} (named "S12" and "S21" respectively) have a value equal to (0.841395, 0). The first refers to the magnitude while the second is the angle. Since the value of the required attenuation is real, the imaginary part of both coefficients is set zero. However, the reflection coefficients q_{11} and q_{22} (i.e., "S11" and "S22") are void because there is no data specified for them. In case of the return loss value property in the port block is specified or assigned to a default value, it becomes mandatory to calculate q_{11} and q_{22} at this step.

This initial Q-matrix reflects only the electrical data already captured by the functional description (i.e., RIM/PIM). At this step, the Q-matrix is not complete and may be not coherent. It is mandatory to review the Q-matrix at the coherence verification step in order to avoid any errors. In the following design steps and stages, the Q-matrix continues to be populated with additional data from various sources.

[1] Mathematically speaking, an attenuation of 1.5 dB corresponds to a gain equal to -1.5 dB.

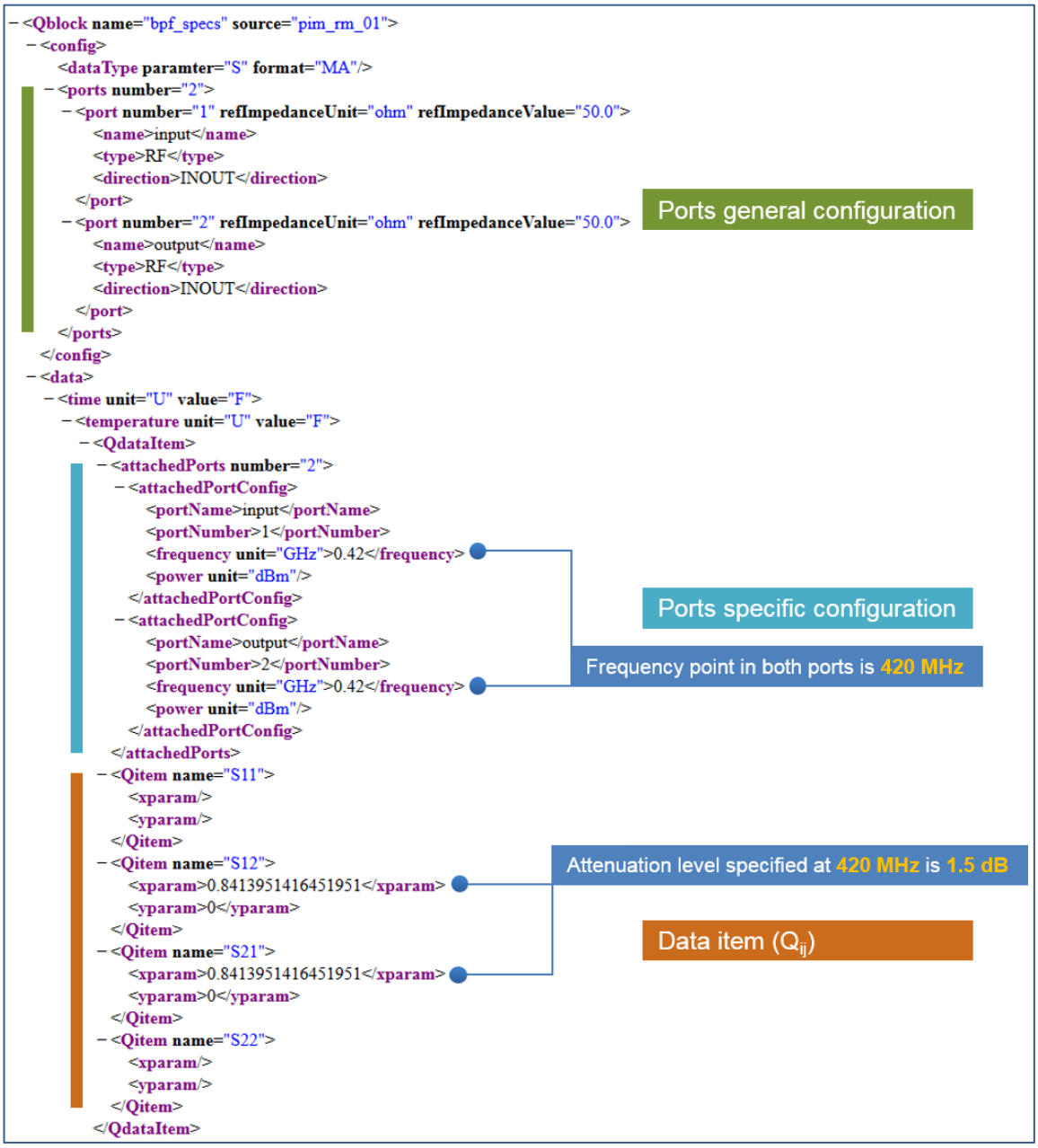


Figure 2.22 A snippet from the created Q-matrix showing initial electrical data item at 420 MHz

2.2.4 Analysis: Coherence verification

The functional description and the Q-matrix have not been submitted so far to any kind of verification. Before starting the analysis step in order to find out an initial design

solution meeting the requirements, a coherence verification test should take place. This process uses a set of coherence rules that are used to validate the consistency of the functional description. In this case study, these rules are part of the RM/PIM and not provided as a separate input. We used SysML parametric diagram to capture three types of coherence rules.

Let us first consider the following notations:

N	Filter order
BW	Bandwidth
$\%B$	Fractional Bandwidth
F_0	Central Frequency
F_c	Cutoff Frequency
F_s	Stopband Edge Frequency
S_a^b	Shape Factor
r	ripple
τ_g	Group Delay
A_p	Passband Attenuation
A_s	Stopband Attenuation
R	Filter Selectivity or Rejection
Z_{ref}	Reference Impedance
Q	Quality Factor
RL	Return Loss
NF	Noise Figure

- **Electrical consistence rules**

The electrical consistence rules check out the validity of the specified values regarding the relationships linking them. For the bandpass filter, we developed the parameters relationships graph (PRG) of Figure 2.23 in order to determine the links between its value properties. The equations of Table 2.4 illustrate these links. The parametric

diagram of Figure 2.16.b illustrates how to map the same parameters in different equations for each rule. Such mapping avoids ambiguity when complex mathematical formalisms are involved. It is worth noting that the relationships given in Table 2.4 are not exhaustive. We enumerate the remaining ones in Table 2.5. The SysML parametric diagram of Figure 2.14 captures all these rules.

Table 2.4 Relationships between bandpass filter parameters given in Figure 2.23

Arrow Color	Mathematical Relationship	Parameter
●	$R = A_s\|_{F_{sL}} - A_p\|_{F_{pL}}$	Rejection (filter selectivity)
●	$N = \dfrac{log\left(\dfrac{10^{\frac{A_s}{10}} - 1}{10^{\frac{A_p}{10}} - 1}\right)}{2log\left(\dfrac{\omega_s}{\omega_p}\right)}$	Filter Order (Butterworth)
●	$\varepsilon = \dfrac{1}{\sqrt{10^{\frac{A_s}{10}} - 1}}$	Ripple Factor (Chebyshev)
●	$r_{\tau g} = \max(\tau_g)\|_{[F_{pL},F_{pH}]} - \min(\tau_g)\|_{[F_{pL},F_{pH}]}$	Group Delay Ripple
●	$S_a^b = \left[\dfrac{10^{A_s\|_{F_s}/10} - 1}{10^{A_p\|_{F_p}/10} - 1}\right]^{1/2N}$	Shape factor (Butterworth)
●	$Q = \dfrac{\omega_c}{2\cos\theta_k}$	Quality Factor (Butterworth)
●	$\%B = \dfrac{F_0}{BW}$	Fractional Bandwidth
●	$VSWR = \dfrac{10^{\frac{RL}{20}} + 1}{10^{\frac{RL}{20}} - 1}$	Voltage Standing Wave Ratio
●	$RL = \dfrac{Z_L - Z_S}{Z_L + Z_S}$	Return Loss

Table 2.5 Main filter electrical consistence rules

No	Rule	if test fails
E.1	$A_p \leq A_s$	Error
E.2	For lowpass filter: $F_c \leq F_s$	Error
E.3	For highpass filter: $F_s \leq F_c$	Error
E.4	For bandpass and bandstop filters: $BW \leq F_0$	Warning

- **PIM-level design constraints**

These rules verify if the functional description leads to a non-feasible design solution. For this bandpass filter, a single PIM-level design constraint is considered. This rule is expressed by the algorithm of Table 2.6 that is captured in the parametric diagram Figure 2.16.a.

- **Integrity control rules**

This type of rules ensures that the RM/PIM value properties are within a predefined range in order to prevent errors. A set of 35 rules are considered to control the bandpass filter RM/PIM. These rules are enumerated in Table 2.7 and captured in the parametric diagram of Figure 2.15. The rules that are effectively used for the coherence verification of the bandpass filter are I.1 through I.31, I.34 and I.35. The rules I.32 and I.33 are used only for single-side filters.

- **Coherence verification report**

After developing the functional description and the coherence rules of the bandpass filter, we made the test. As shown in Figure 2.24, the RM/PIM passes the coherence verification test. Consequently, there is no need to alter the RM/PIM and recreate the

corresponding Q-matrix. The next step is the PIM analysis where one or many candidate design solutions should be figured out.

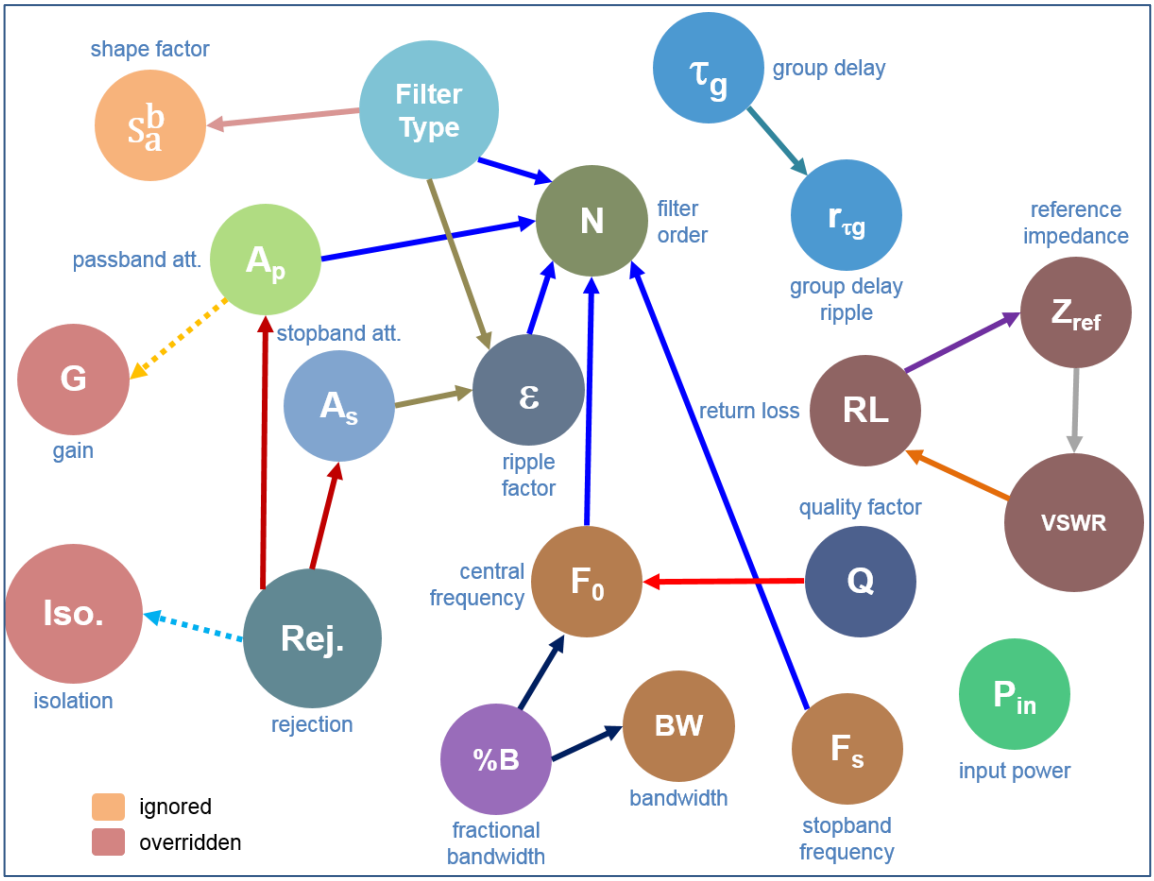


Figure 2.23 Bandpass filter parameters relationships graph

Table 2.6 Bandpass filter PIM-level design constraints rules

No	Rule	if test fails

C.1	if type ∈ {Chebyshev Type I, Chebyshev Type II, Elliptic} then
	if rem(order/2) then
	$Port[2].impedance \neq Port[2].refImpedance$
	end
	end
		Error or Warning

Table 2.7 Bandpass filter integrity control rules

No.	Rule	if test fails
I.1	$-80 \leq$ temperatureRange ≤ 100	Error or/and Warning
I.2	ratingTemperatureRange ∈ {TRUE,FALSE}	Error
I.3	$1 \leq$ VSWR ≤ 10	Error or/and Warning
I.4	no $\in \mathbb{N}^*$	Error
I.5	$0 < Z_{ref} \leq 10^4$	Error or/and Warning
I.6	$0 < RL \leq 120$	Error or/and Warning
I.7	$0 < NF \leq 100$	Error or/and Warning
I.8	ptype ∈ {DC,IF,RF}	Error
I.9	connection ∈ {TERMINATED,OPENED,SHORTED}	Error
I.10	direction ∈ {IN,OUT,INOUT}	Error
I.11	$0 \leq$ frequencyRange $\leq 10^{12}$	Error or/and Warning
I.12	$0 \leq$ powerRange ≤ 160	Error or/and Warning
I.13	$0 \leq$ voltageRange $\leq 10^4$	Error or/and Warning
I.14	$0 \leq$ currentRange $\leq 10^2$	Error or/and Warning
I.15	ratingFrequency ∈ {TRUE,FALSE}	Error
I.16	ratingPower ∈ {TRUE,FALSE}	Error
I.17	ratingVoltage ∈ {TRUE,FALSE}	Error
I.18	ratingCurrent ∈ {TRUE,FALSE}	Error
I.19	$0 \leq$ %B ≤ 1	Error
I.20	Q ∈]0,∞[	Error or/and Warning
I.21	type ∈ {Bessel, Butterworth, Chebyshev Type I, Chebyshev Type II, Custom, Elliptic}	Error
I.22	$\forall i, 0 \leq$ ripple[i] ≤ 120	Error or/and Warning
I.23	$\forall i, 0 \leq$ passbandAttenuation[i] ≤ 120	Error or/and Warning
I.24	$\forall i, 0 \leq$ stopbandAttenuation[i] ≤ 120	Error or/and Warning
I.25	$S_a^b \geq 1$	Error
I.26	$0 \leq$ R ≤ 120	Error or/and Warning
I.27	N $\in \mathbb{N}_+^*$	Error
I.28	$0 \leq$ groupDelayRange ≤ 10	Error or/and Warning

No.	Rule	if test fails
I.29	ratingGroupDelayRange ∈ {TRUE,FALSE}	Error
I.30	$0 \leq$ groupDelayRipple ≤ 10	Error or/and Warning
I.31	ratingGroupDelayRipple ∈ {TRUE,FALSE}	Error
I.32	$0 \leq F_c \leq 10^{12}$	Error or/and Warning
I.33	$0 \leq F_s \leq 10^{12}$	Error or/and Warning
I.34	$0 \leq F_0 \leq 10^{12}$	Error or/and Warning
I.35	$0 \leq BW \leq 10^{10}$	Error or/and Warning

```
******** Coherence Verification Test ********
TARGET PIM: 450-MHz BANDPASS FILTER (C:\Users\slafi\Documents\Design\bpf450_pim.xml)

-> Electrical Consistence Rules (11 found)
    ..................................... PASS
-> PIM-level Design Constraints Rules (1 found)
    ..................................... PASS
-> Integrity Control Rules (33 found)
    ..................................... PASS
    Summary: (0) Warnings (0) Errors
```

Figure 2.24 Bandpass filter coherence verification report

2.2.5 Analysis: PIM-level granularity refinement and simulation

At this step, the functional description passed the coherence verification test. We need to find out design solutions that meet the requirements of that functional description.

- **Design space exploration**

To figure out a candidate design solution that fits with the filter's functional description, we associate a black-box model to it (see details in section 1.3.2 of (Lafi, 2015)). This black-box model is mathematically defined by the equation (A-1, p. 114) in APPENDIX B. Hence, the functional description provides an abstract representation of the filter functionality while the black-box model provides a theoretical realization of that functionality. From this viewpoint, the relationship between the functional description and the associated black-box model is similar (but not identical) to the relationship

between a class and an object (i.e., an instantiation of that class) in object-oriented programming.

We consider four traditional approximations for the filter transfer function on which the black-box model is based. These approximations are:

- Butterworth,
- Chebyshev Type I,
- Elliptic (aka Cauer), and
- Bessel.

The equations (A-5, p. 115) and (A-8, p. 115) in APPENDIX A give the transfer functions related to Butterworth and Chebyshev Type I approximations. The reader can refer to (Wanhammar, 2009) for a thorough dissertation about the mathematical background of all these approximations.

Based on the PIM, the analysis of the black-model (i.e., four filter prototypes) gives the frequency and group delay responses shown in Figure 2.25-2.28. The Table 2.8 shows a detailed comparison between the performances of these prototypes. This comparison is based on the requirements already captured in the filter's RM (see Figure 2.11) as well as the detailed block definition diagram capturing the filter's value properties, their weights and constraints (see Figure 2.8 and Figure 2.9). The metrics we consider in the following are:

- Form factor: expressed by the filter order,
- Performance: expressed by passband and stopband attenuation, rejection, group delay, frequency characteristics, output impedance and shape factor.

Based on the results reported in Figures 2.25, 2.26, 2.27, 2.28 and Table 2.8, the performance of some prototypes considered in the black-box model is relatively satisfactory. Then, we proceed directly to the selection of the best candidate design solution without performing granularity refinement.

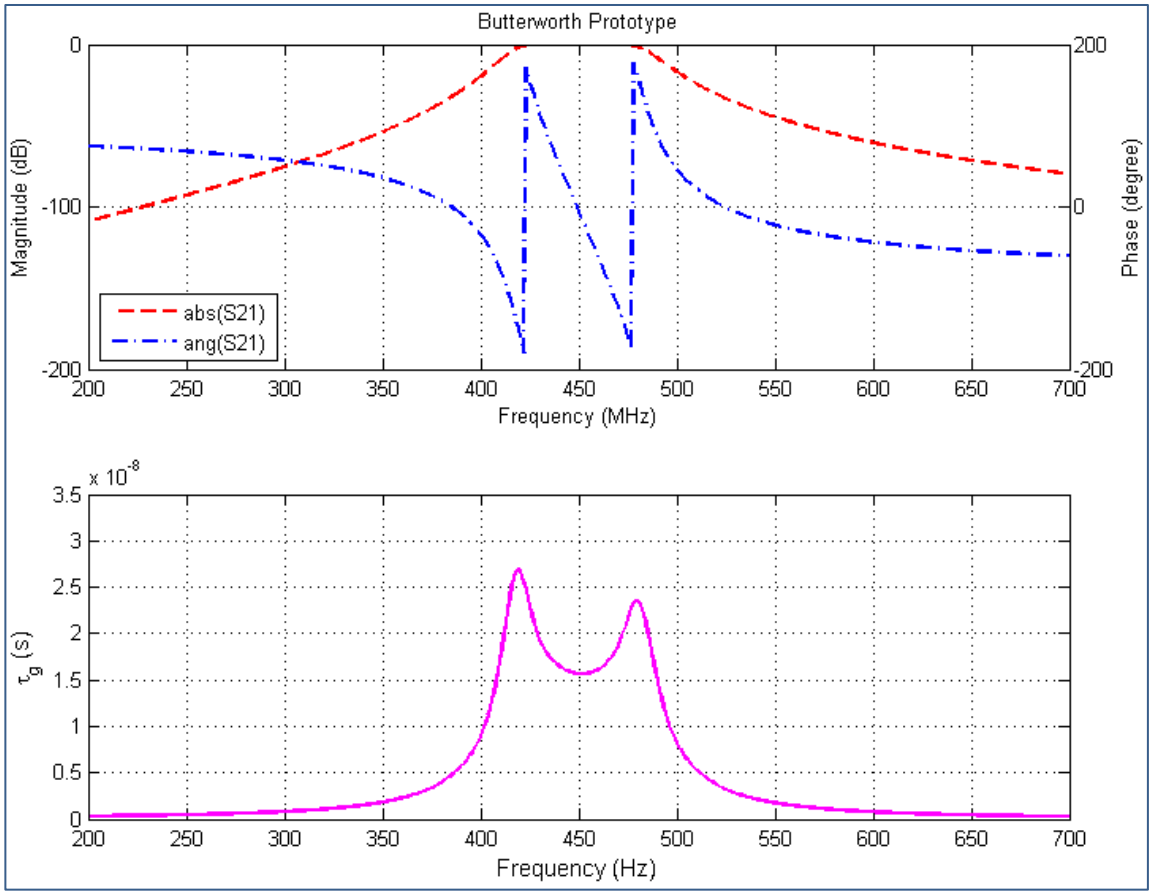


Figure 2.25 PIM analysis: Attenuation and group delay for Butterworth prototype

- **Decision making: design solution selection**

Given the four candidate design solutions (i.e., filter prototypes), we proceed to the selection of the best one that fits with the filter's requirements. To do so, we use an objective function. The constant weight of each value property is already given in the "Generic Filter" bdd of Figure 2.9. The variable weights are calculated based on the property constraints given in the same bdd. The Table 2.9 summarizes the constant weights and property constraints of each decision variable while the Table 2.10 details the calculations of the objective function of each filter prototype.

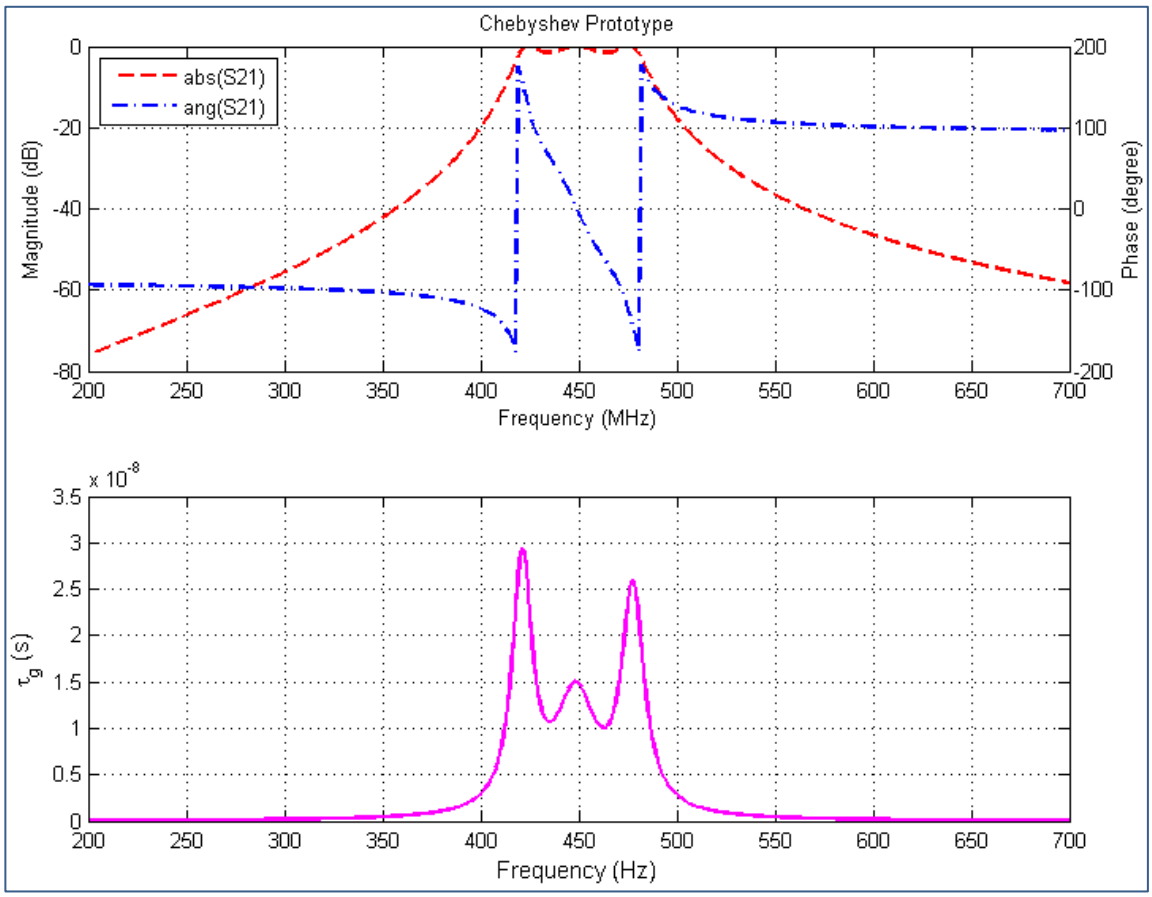


Figure 2.26 PIM analysis: Attenuation and group delay for Chebyshev type I prototype

As shown in Table 2.10, each filter prototype has its own objective function. Since we look for the maximum objective function, the best candidate solution found is the Butterworth approximation. Then, the Chebyshev Type I and Elliptic are respectively ranked second and third with a small difference. This is because both approximations have similar response characteristics in the passband for the same filter order. The worst design solution is the Bessel prototype. It is the only prototype that has negative variable weights for stopband attenuation and rejection decision variables (i.e., X_4, X_5, X_{12} and X_{13}). This is mainly due to the significant gap between the prototype performance and the filter requirements.

This example shows that careful modeling of the functional description (such as providing weights and constraints) enables the automation of the decision making process and contributes to the enhancement of the automation level in the overall design cycle.

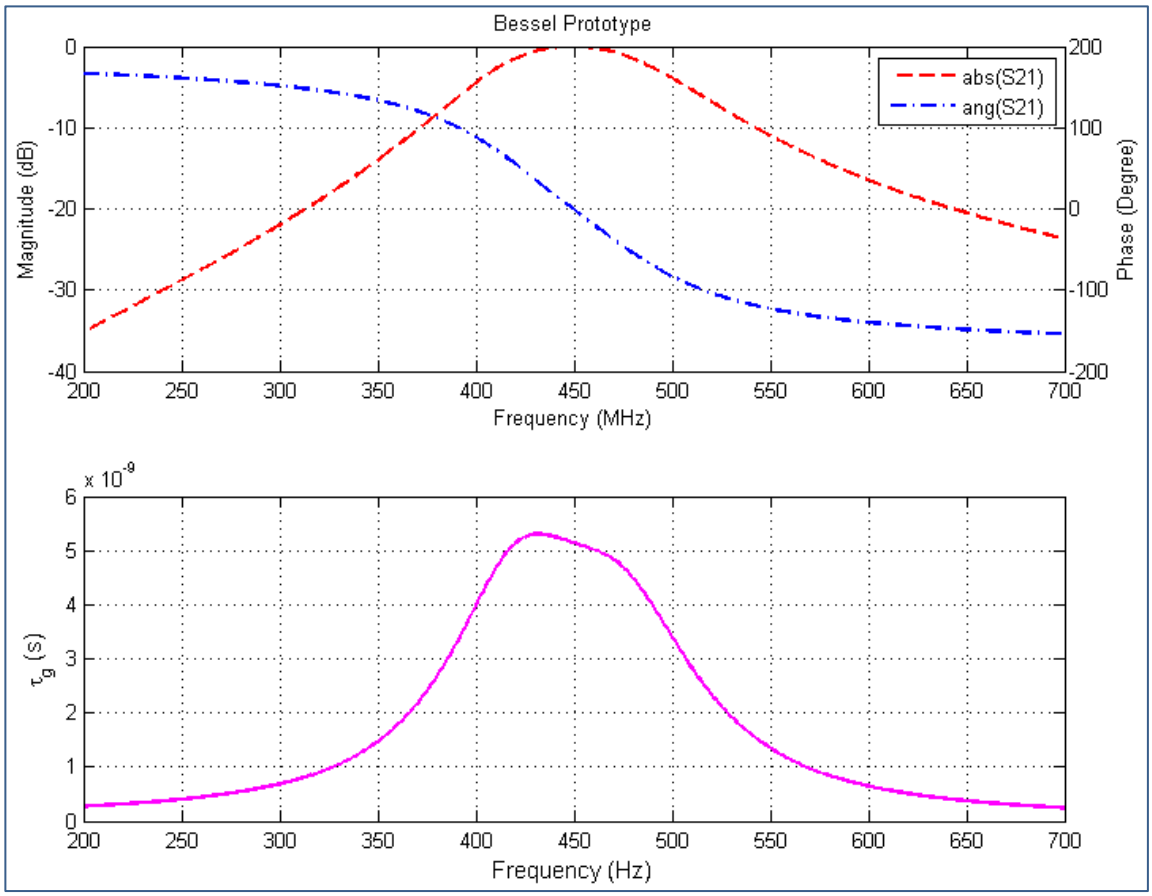


Figure 2.27 PIM analysis: Attenuation and group delay for Bessel prototype

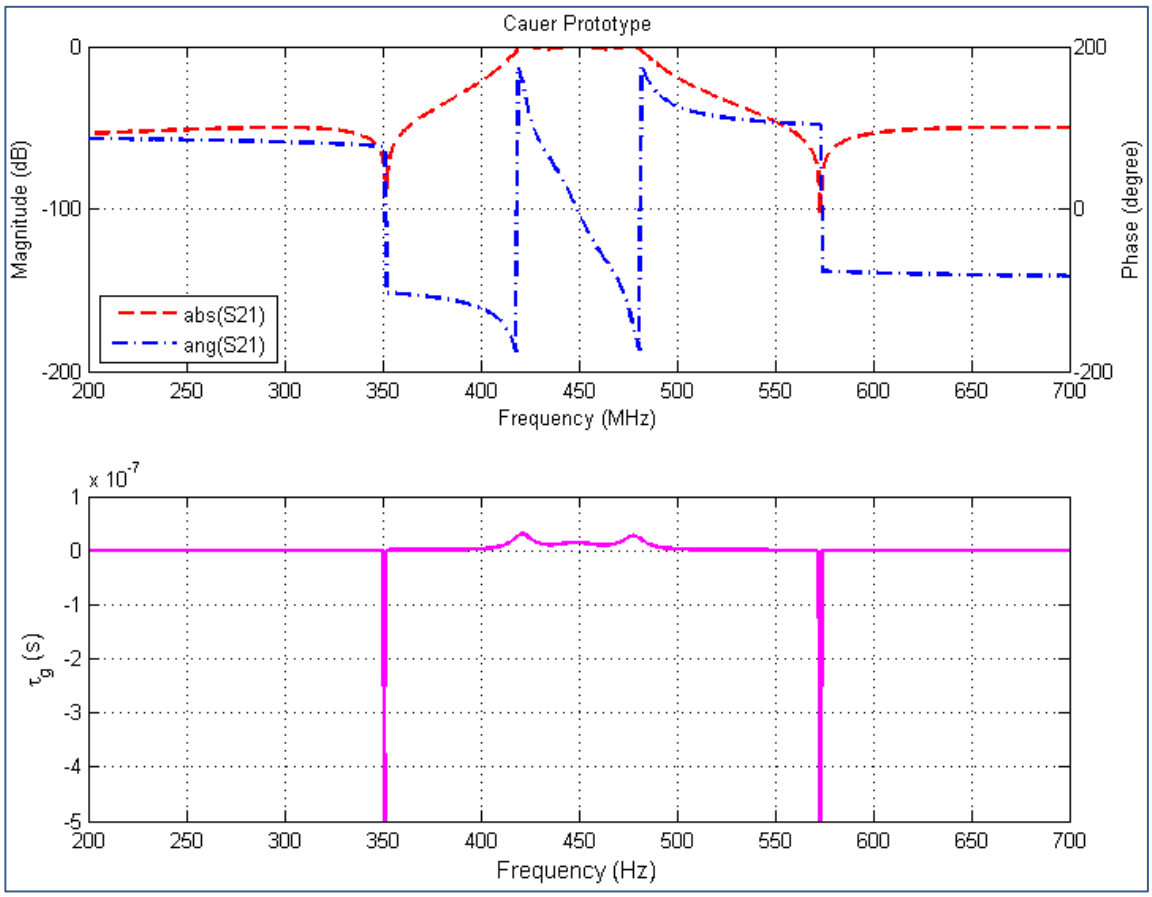


Figure 2.28 PIM analysis: Attenuation and group delay for Elliptic prototype

Table 2.8 Black-box model: performance summary of the four filter prototypes

		Butterworth	Chebyshev Type I	Bessel	Elliptic (Cauer)
	Filter Order (N)	5	3	6	3
Attenuation (dB)	**300 MHz**	-75.394	-55.569	-22.006	-50.004
	420 MHz	-1.501	-1.501	-1.5	-1.501
	450 MHz	$-385.2 \cdot 10^{-6}$	-0.018	-0.002	$-356.5 \cdot 10^{-6}$
	480 MHz	-1.501	-1.501	-1.5	-1.501
	600 MHz	-60.5	-46.46	-16.437	-53.586
Group Delay (s)	**300 MHz**	$805.4 \cdot 10^{-12}$	$195.3 \cdot 10^{-12}$	$6.868 \cdot 10^{-10}$	$194.1 \cdot 10^{-12}$
	420 MHz	$26.73 \cdot 10^{-9}$	$28.31 \cdot 10^{-9}$	$5.132 \cdot 10^{-9}$	$29.62 \cdot 10^{-9}$

		Butterworth	Chebyshev Type I	Bessel	Elliptic (Cauer)
	450 MHz	$15.67 \cdot 10^{-9}$	$14.82 \cdot 10^{-9}$	$5.137 \cdot 10^{-9}$	$14.35 \cdot 10^{-9}$
	480 MHz	$23.48 \cdot 10^{-9}$	$22.88 \cdot 10^{-9}$	$4.49 \cdot 10^{-9}$	$23.97 \cdot 10^{-9}$
	600 MHz	$769.4 \cdot 10^{-12}$	$190.4 \cdot 10^{-12}$	$6.457 \cdot 10^{-10}$	$189 \cdot 10^{-12}$
	Maximum group delay ripple over passband	$11.07 \cdot 10^{-9}$	$19.35 \cdot 10^{-9}$	$8.125 \cdot 10^{-10}$	$20.74 \cdot 10^{-9}$
Rejection (dB)	$\boldsymbol{R_L = A_s\vert_{F_{sL}} - A_p\vert_{F_{pL}}}$	73.893	54.068	20.506	48.503
	$\boldsymbol{R_H = A_s\vert_{F_{sH}} - A_p\vert_{F_{pH}}}$	58.999	44.959	14.937	52.085
Passband Bandwidth (MHz)	BW	60	60	60	60
Central Frequency (MHz)	F_0	450	450	450	450
Reference Impedance (Ω)	Z_{ref}	50	50	50	50
Shape Factor	$\boldsymbol{S_a^b = \frac{\Delta BW_{60dB}}{\Delta BW_{6dB}}}$	3.53	6.38	109.26	6.64

2.2.6 Analysis: PSM generation

After the selection of the best PIM-level design solution among the four filter prototypes (i.e., black-box model), we proceed to the transformation of that solution into a PSM-level solution.

Table 2.9 Constant weights and property constraints for each decision variable

	Value Property (X)	i	**Constant weights C_i**	**Property Constraint**
Filter order	N	1	4.0	Variable minimization (i.e., due to small form factor requirement)
Passband Attenuation (dB)	A_p @ 420 MHz	2	3.0	Relative specification
	A_p @ 480 MHz	3	3.0	Relative specification
Stopband Attenuation (dB)	A_s @ 300 MHz	4	5.0	Variable maximization
	A_s @ 600 MHz	5	5.0	Variable maximization
Group Delay (s)	g_d @ 300 MHz	6	1.5	Relative specification
	g_d @ 420 MHz	7	1.5	Relative specification
	g_d @ 450 MHz	8	1.5	Relative specification
	g_d @ 480 MHz	9	1.5	Relative specification
	g_d @ 600 MHz	10	1.5	Relative specification
	$r_{g_d} = \max(g_d)\|_{[F_{pL},F_{pH}]}$	11	2.0	Relative specification
Rejection (dB)	$R_L = A_s\|_{F_{sL}} - A_p\|_{F_{pL}}$	12	3.0	Variable minimization
	$R_H = A_s\|_{F_{sH}} - A_p\|_{F_{pH}}$	13	3.0	Variable maximization

	Value Property (X)	i	Constant weights C_i	Property Constraint
Passband Bandwidth (MHz)	BW	14	2.0	Absolute specification
Central Frequency (MHz)	F_0	15	2.0	Absolute specification
Reference Impedance (Ω)	Z_{ref}	16	2.0	Absolute specification
Shape Factor	$S_a^b = \frac{\Delta BW_{60dB}}{\Delta BW_{6dB}}$	17	1.0	Relative specification

- **PIM-to-PSM transformation**

To transform the PIM-level Butterworth filter prototype into a PSM-level lumped-component filter circuit, we use a model-to-model (i.e., PIM-to-PSM) transformation. For this purpose, Table 2.11 suggests a cross-view transformation that consists of three consecutive steps:

1. Step 1: Calculation of the normalized values for filter resonators

The first task is the calculation of the lowpass prototype normalized values. To do so, a frequency transformation should take place in order to determine the lowpass prototype frequency characteristics based on their bandpass filter counterparts. Then, the normalized values g_k for each resonator (i.e., N) are calculated using the equations (A-6, p. 115) and (A-7, p. 115) in APPENDIX A for Butterworth approximation. The resonators X_k are calculated using the equations (A-12, p. 117) and (A-13, p. 117).

2. Step 2: Calculation of lumped elements values

Given the values of resonators, the corresponding lumped components are computed after impedance and frequency scaling (using equations (A-14, p. 118) and (A-15, p. 118)). This results in a lowpass lumped-component filter that requires a lowpass to bandpass frequency transformation (as given in equation (A-16, p. 119) and Table A-1, p. 117). Thus, we obtain a first PSM that is an ideal lumped-component bandpass filter. Its circuit schematic is shown in Figure 2.29.a. The corresponding performance assessment (in terms of attenuation and group delay responses) is illustrated in Figure 2.29.b.

3. Step 3: Synthesis of transmission lines and circuit topology

The PSM generated in step 2 satisfies the requirements of the functional description. However, it is still not practically realizable because there are no physical connections between the lumped components. In step 3 of the PIM-to-PSM transformation, we alter the topology of each resonator (as illustrated in Figure 2.30.a and Figure 2.30.b) in order to physically connect each lumped component to the others. This takes place by adding several sections of planar transmission lines. The final PSM circuit is shown in Figure 2.31.

Table 2.10 Objective function calculations for each filter prototype

Decision variables	**Constant weights C_i**	**Variable weights X_i**				**Weights $C_i \cdot X_i$**			
		Butterworth	**Chebyshev Type I**	**Bessel**	**Elliptic**	**Butterworth**	**Chebyshev Type I**	**Bessel**	**Elliptic**
X_1	4.0	0.2	0.3334	0.1667	0.3334	0.8	1.3336	0.6668	1.3336
X_2	3.0	1.0007	1.0007	1	1.0007	3.0021	3.0021	3	3.0021
X_3	3.0	1.0007	1.0007	1	1.0007	3.0021	3.0021	3	3.0021
X_4	5.0	1.88485	1.389225	-0.39985	1.2501	9.42425	6.946125	-1.99925	6.2505
X_5	5.0	1.21	0.9292	-0.62126	1.07172	6.05	4.646	-3.1063	5.3586
X_6	1.5	1	1	1	1	1.5	1.5	1.5	1.5
X_7	1.5	1	1	1	1	1.5	1.5	1.5	1.5
X_8	1.5	1	1	1	1	1.5	1.5	1.5	1.5
X_9	1.5	1	1	1	1	1.5	1.5	1.5	1.5
X_{10}	1.5	1	1	1	1	1.5	1.5	1.5	1.5
X_{11}	2.0	0	0	0	0	0	0	0	0
X_{12}	3.0	1.9193	1.4044	-0.4174	1.2598	5.7579	4.2132	-1.2522	3.7794
X_{13}	3.0	1.2165	0.927	-0.642	1.0739	3.6495	2.781	-1.926	3.2217
X_{14}	2.0	1	1	1	1	2	2	2	2
X_{15}	2.0	1	1	1	1	2	2	2	2
X_{16}	2.0	1	1	1	1	2	2	2	2
X_{17}	1.0	0	0	0	0	0	0	0	0
Objective Function $F(n) = \sum C_i X_i$						**45.18585**	**39.424125**	**11.88305**	**39.448**

Table 2.11 PIM-to-PSM transformation (target platform: LC/distributed lines)

Step 1: Calculation of resonators normalized values

1.1. Calculate lowpass prototype (LPP) normalized values (g_k) for the response approximation
1.2. Calculate corresponding resonator values (X_k)

Step 2: Calculation of corresponding lumped elements' values

2.1. Perform impedance scaling
2.2. Perform frequency scaling
2.3. Perform frequency transformation

Step 3: Synthesis of transmission lines and circuit topology

3.1.a. For each series resonator:

Create topology as depicted in Figure 2.30.a

3.2.b. For each parallel resonator:

Create topology as depicted in Figure 2.30.b

3.3. For each transmission line:

Consider the properties given in the following:

$$\begin{cases} Z_{ci} = Z_{reference} \\ E_i = E_{default} \\ F_i = F_0 \\ A_i = A_{default} \end{cases}$$

where Z_{ci} is the characteristic impedance of the transmission line, E_i is its electrical length, F_i is the frequency at which the electrical length is calculated and A_i is the transmission line loss (in dB).

- **PSM performance assessment**

The step 3 of the PIM-to-PSM transformation generates a physically realizable PSM (see circuit schematic in Figure 2.31.a). After linear simulation, the PSM's frequency response is shown in Figure 2.32. It is obvious that this frequency response does not satisfy the functional description requirements. This is because of the transmission-line sections added to connect the lumped components together. The physical properties of these sections already not accounted at the PIM level while considered at the PSM level degraded the solution's overall performance. Despite the several optimization iterations carried out to enhance the obtained PSM's performance, we did not succeed to enhance significantly its frequency response.

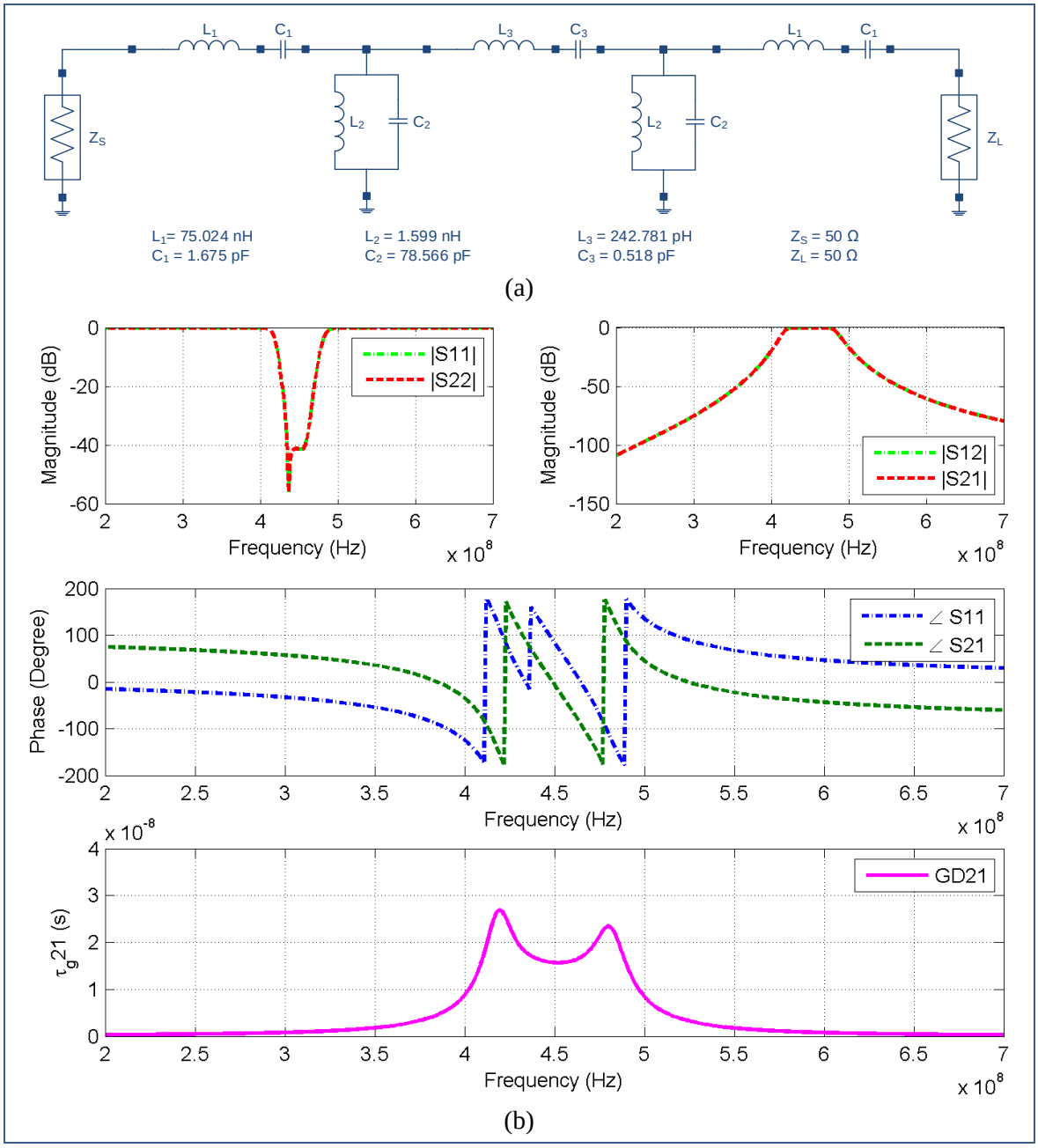


Figure 2.29 Transformation (step 2): (a) the LC model and its (b) frequency response

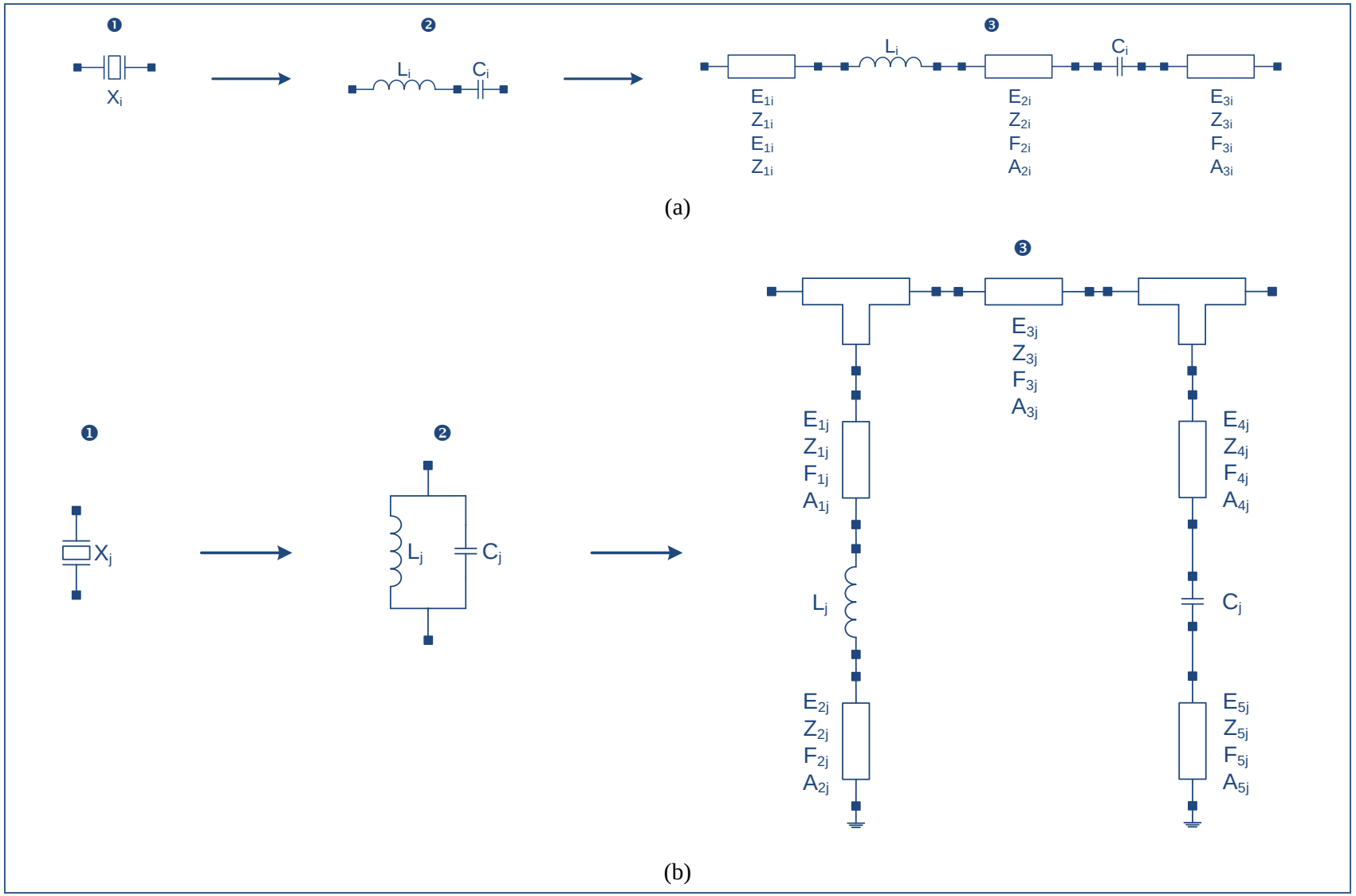


Figure 2.30 Generation of lumped-component filter circuit topology: (a) series and (b) parallel resonators

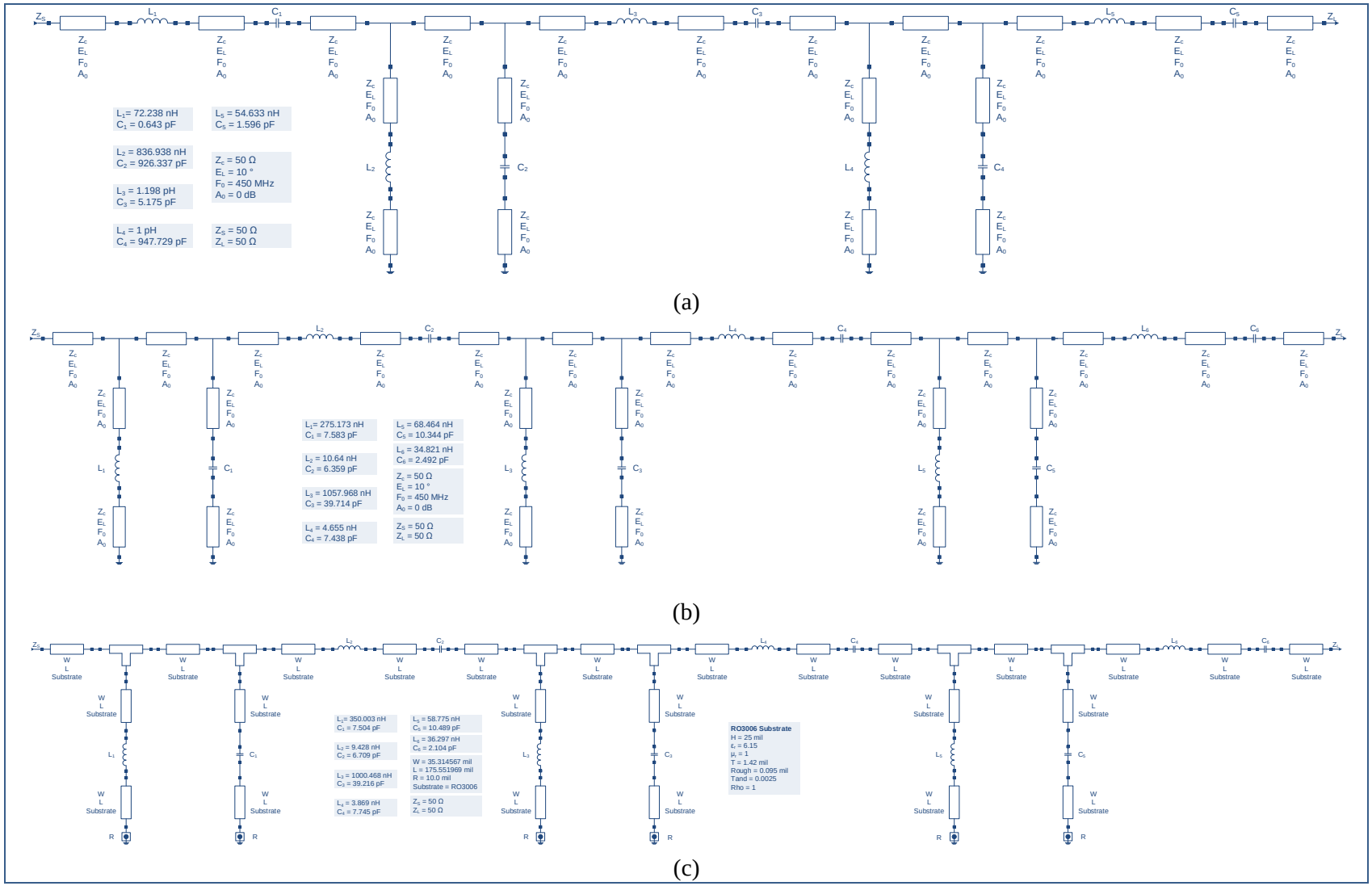


Figure 2.31 PSM evolution: (a) transformation output ($N = 5$) (b) refined ($N = 6$) and (c) technology-mapped ($N = 6$)

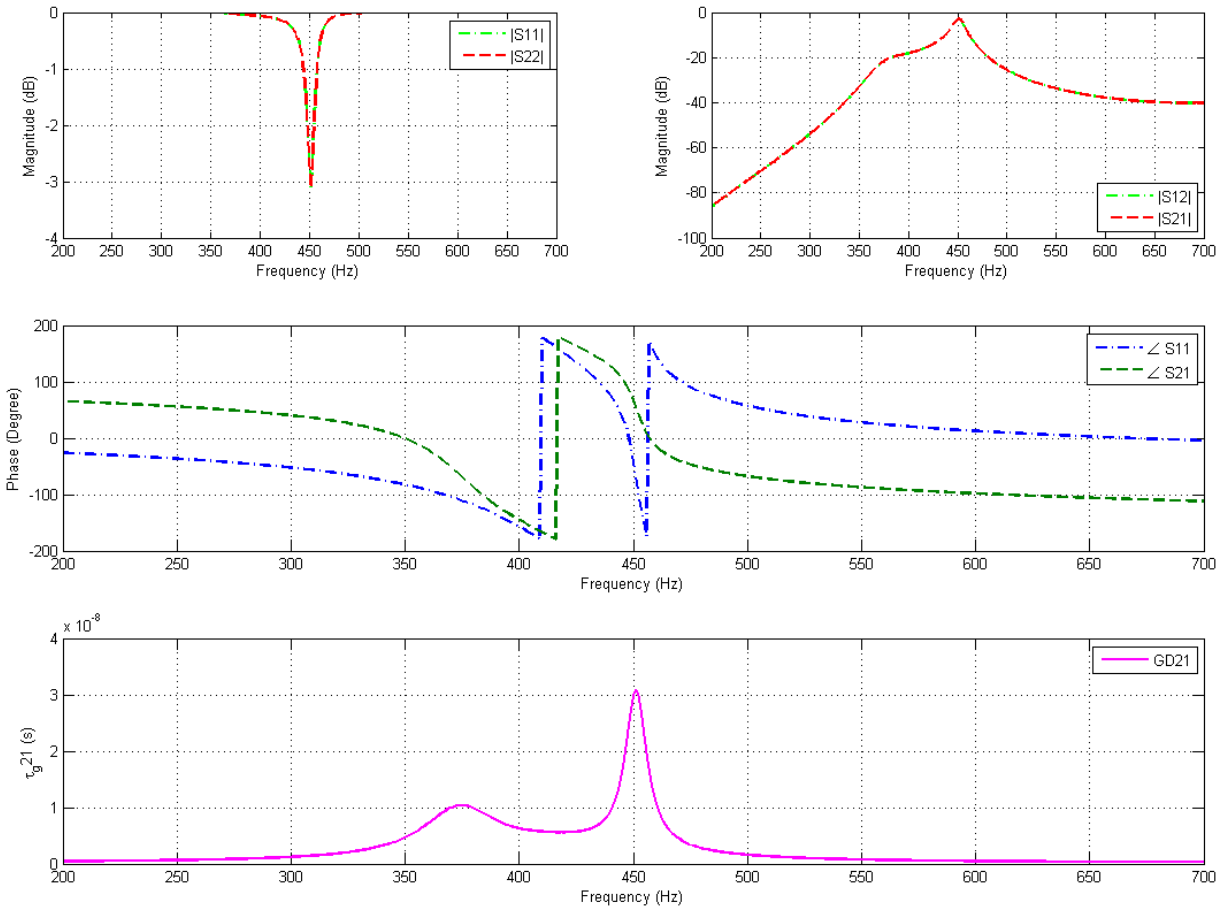


Figure 2.32 The frequency response of the PSM resulting from step 3 of the PIM-to-PSM transformation ($N = 5$)

2.2.7 Synthesis: PSM granularity refinement

Since the obtained PSM (even after several optimization iterations) does not satisfy the requirements, we need to use the granularity refinement process in order to enhance its quality. To change the PSM's granularity level, we must define an intra-view transformation that alters the number of resonators and relaxes the design constraints to satisfy the requirements.

- **Intra-view transformation**

The minimum number of resonators that meets the requirements at the PIM level is five (i.e., Butterworth prototype). After connecting the resonators using transmission-line sections, the PSM's performance is no longer satisfactory. To enhance its frequency response, we define the intra-view transformation of Table 2.12. Depending on the PSM's topology, this intra-view transformation adds a new resonator as depicted in Figure 2.33. That resonator corresponds to the immediate higher order (i.e., $N + 1$). Its properties are defined as given in step 2 of the granularity refinement transformation. Figure 2.31 illustrates the resulting PSM schematic.

The PIM-to-PSM and the granularity refinement transformations of Table 2.11 and Table 2.12 respectively illustrate the difference between intra- and cross-view transformations already discussed in (Lafi, 2015). The first implies a change of the abstraction level (in this case, from functional to circuit). However, the second changes only the design viewpoint within the same abstraction level (represented here by the change of the circuit's granularity level).

- **Refined PSM performance optimization**

The PSM circuit schematic of Figure 2.31 is submitted to many optimization iterations. In this case study, we rapidly come up with a design solution that satisfies the functional description requirements (see frequency response in Figure 2.34). In general, it is

possible to get no satisfactory solution, which suggests attempting more granularity refinement iterations.

Table 2.12 Overview of the intra-view transformation for granularity refinement

Step 1: Topology refinement

- In case of n[th] series resonator:

Add series resonator structure as depicted in Figure 2.33.a.

- In case of n[th] parallel resonator:

Add parallel resonator structure as depicted in Figure 2.33.b.

Step 2: Resonator properties

Consider the resonator properties given in the following:

$$\begin{cases} Z_{ci} = Z_{reference} \\ E_i = E_{default} \\ F_i = F_0 \\ A_i = A_{default} \\ L_i = L_N|_{N=N+1} \\ C_i = C_N|_{N=N+1} \end{cases}$$

where Z_{ci} is the characteristic impedance of the transmission line, E_i is its electrical length, F_i is the frequency at which the electrical length is calculated and A_i is the transmission line loss (in dB). L_i and C_i are the n[th] lumped components corresponding to the immediate higher order $N + 1$.

2.2.8 Synthesis: Technology mapping

The PIM-to-PSM transformation considered ideal circuit components at the PSM generation due to the absence of detailed technology data. That is why we used so far ideal lumped components and transmission lines. However, ideal components do not reflect the real circuit performance. To bridge this gap, the technology mapping stage allows augmenting the PSM with detailed information about the target technology platform for each component.

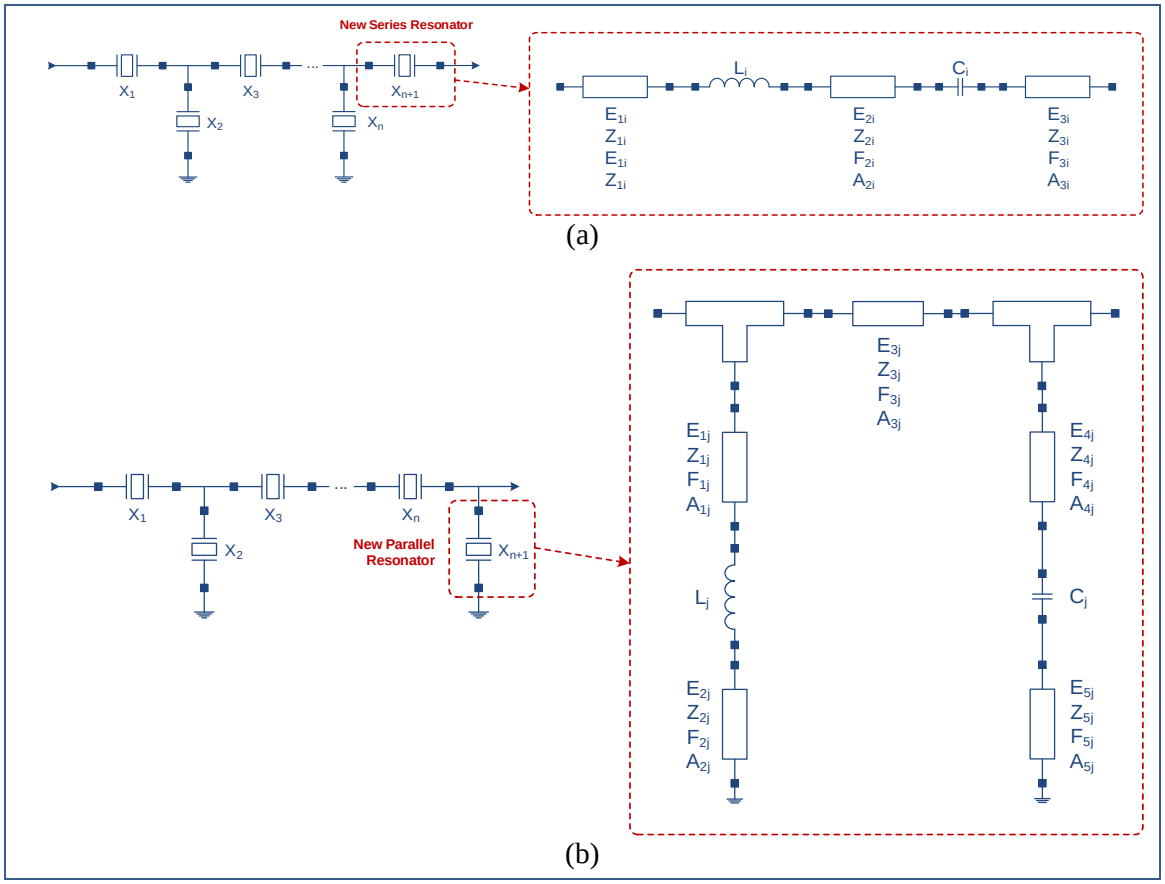


Figure 2.33 Granularity refinement: transformation of (a) series and (b) parallel resonators

- **Target technologies**

As previously mentioned, the PIM is implemented using lumped components (i.e., capacitors and inductors) to be connected together through distributed lines. Lumped components such as capacitors and inductors are non-ideal devices. In fact, a real capacitor (respectively inductor) is not fully capacitive (respectively inductive). Depending on their manufacturing technology, these components suffer from various parasitic effects (such as series and/or shunt resistances). For the characterization of each component, electrical models are developed in order to take into account these parasitics. For instance, Table 2.13 shows two electrical models for non-ideal inductors

and capacitors. In addition to capacitors and inductors, we use microstrip[2] transmission lines to route signals between resonators. This type of transmission lines consists of a conducting strip separated from a ground plane by a dielectric layer (i.e., the substrate).

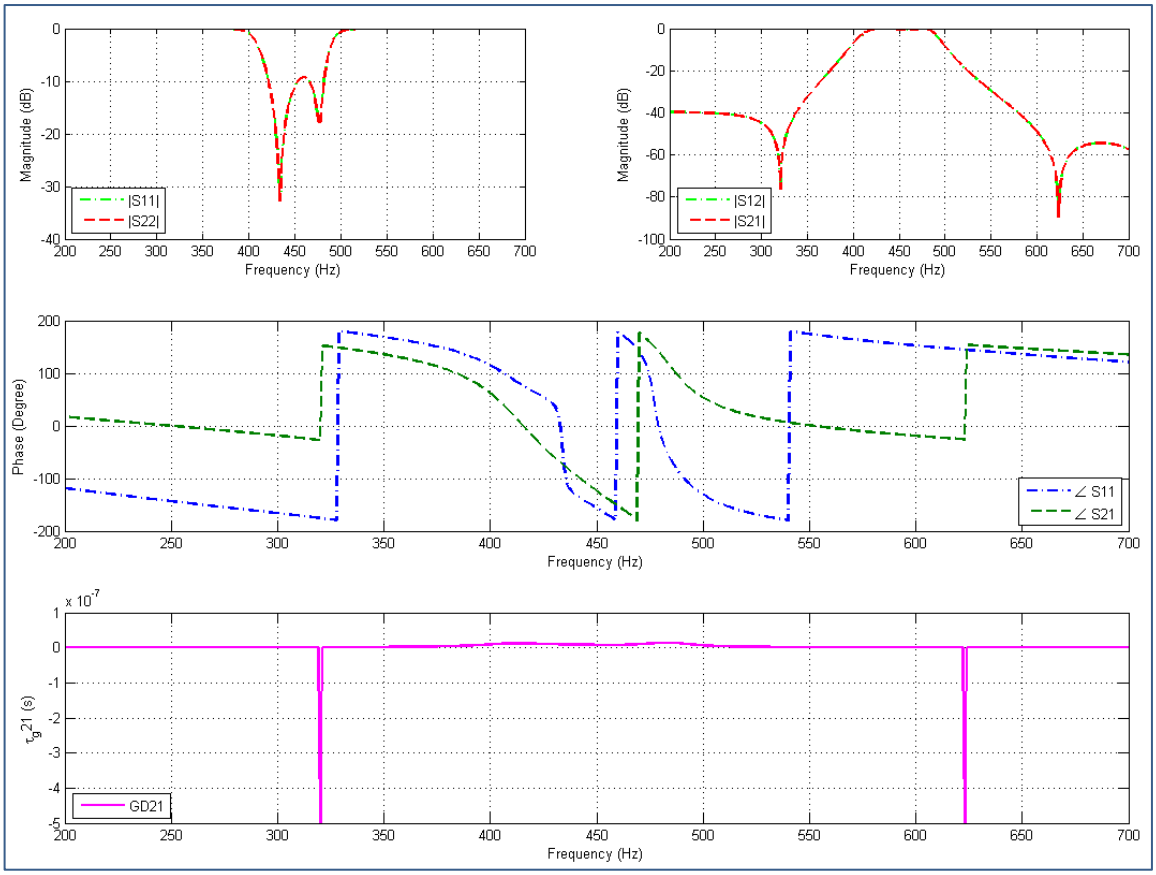
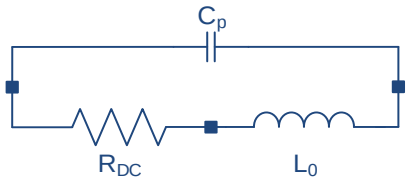


Figure 2.34 The frequency response of the optimized design solution after PSM granularity refinement

[2] For more information about microstrip transmission lines, the reader might refer to Hong, J.S., et M.J. Lancaster. 2001. *Microstrip Filters for RF/Microwave Applications*. John Wiley and Sons., Pozar, David M. (732). 2012. *Microwave Engineering*, Fourth. John Wiley and Sons.

Table 2.13 Electrical models for non-ideal inductors and capacitors

Ideal Device	Real Electrical Model	Remarks
		**L_0**: nominal inductance value (measured in Henri) **R_{DC}**: DC Resistance (measured in ohm) **C_p**: parallel capacitance value (measured in Farad), estimated using the formula: $$C_p = \frac{1}{(2\pi F_0)^2 L_0}$$ where F_0 is the inductance's self-resonant frequency
		**C_0**: nominal capacitance value (measured in Farad) **ESR**: Equivalent Series Resistance (measured in Ohm)

The technology mapping process requires a technology library that provides the electrical characteristics of real components and transmission lines. It searches in this library the components that approaches the most the ideal ones already included in the PSM. Various metrics might be considered for the selection of real components.

```
<!--
    Lumped-Component Database - version 1.0 @ 2012-02-17 21:52:54 - Author: Sabeur Laf
-->
<database>
  <tech type="RLC">
    <kit provider="Digikey">
      <Resistors></Resistors>
      <Capacitors></Capacitors>
      <Inductors>
        <Inductor unit="nH">30</Inductor>
        <Inductor unit="nH">32</Inductor>
        <Inductor unit="nH">37</Inductor>
        <Inductor unit="nH">38</Inductor>
        <Inductor unit="nH">40</Inductor>
        <Inductor unit="nH">41</Inductor>
        <Inductor unit="nH">48</Inductor>
        <Inductor unit="nH">50</Inductor>
        <Inductor unit="nH">59</Inductor>
        <Inductor unit="nH">60</Inductor>
        <Inductor unit="nH">67</Inductor>
        <Inductor unit="nH">68</Inductor>
        <Inductor unit="nH">70</Inductor>
        <Inductor unit="nH">72</Inductor>
```

(a)

```
<tech type="Microstrip">
  <package>
    <substrate name="RO3006">
      <er>6.15</er>
      <mur>1.0</mur>
      <h unit="mil">25</h>
      <ht unit="mil">1e20</ht>
      <t unit="mil">0.7</t>
      <cond>4.1e7</cond>
      <tand>0.002</tand>
      <rough>0</rough>
    </substrate>
    <constraints>
      <manufacturingUpperLimit unit="mil">400</manufacturingUpperLimit>
      <manufacturingLowerLimit unit="mil">8</manufacturingLowerLimit>
    </constraints>
  </package>
</tech>
```

(b)

Figure 2.35 Technology files: (a) Lumped components (b) Microstrip substrate

Depending on the available technology information, these metrics might include (but not limited to) the nominal value, parasitic effects, physical dimensions, operating temperature, power consumption and packaging type. In this case study, we could not use proprietary technology libraries. Therefore, we built our own one that is composed of two XML files: the first gathers few thousands of lumped devices (i.e., resistors, capacitors and inductors) while the second provides the technological properties of common microstrip substrates. Figure 2.35.a shows a snippet of Digi-Key[3] inductors included in the lumped components technology file while Figure 2.35.b shows the properties of Rogers 3006 (aka RO3006) substrate, which is a part of the microstrip technology file.

- **PSM augmentation with technology data**

We developed a script to search the devices in the technology library corresponding to ideal PSM lumped components (already listed in Figure 2.31.c). We search inductors in three kits (i.e., fixed, common-mode choke and adjustable). Figure 2.36.a summarizes the findings in each kit as well as their number. The list of selected inductors is detailed in Table 2.14.

Then, we search capacitors in five kits already included in the technology library (i.e., ceramic, aluminum, film, tantalum and polymer). The results are illustrated in Figure 2.36.b and detailed information about the selected capacitors is given in Table 2.15. Finally, we choose the RO3006 substrate for the implementation of the microstrip transmission lines. We synthesized the required sections as illustrated in Figure 2.37.

For the purpose of simplification, we use in this case study only the nominal value of the lumped component as a selection criterion. In general, many selection criteria can be considered. Using an objective function, it becomes possible to automatically select the best devices corresponding to the PSM components.

[3] Digi-Key is one of the largest electronic-component distributors in North America and worldwide.

```
Searching in available inductors libraries...
    ->  Fixed Inductors Library: (910 components)
    ->  Common-Mode-Choke Inductors Library: (367 components)
    ->  Adjustable Inductors Library: (62 components)

Findings Summary:
```

No.	Ideal (H)	Fixed Lib.	CMC Lib.	Adjustable Lib.	Selected (#)
L1	3.500030e-7	3.50e-7	3.00e-7	3.50e-7	3.50e-7 (3)
L2	9.428000e-9	9.40e-9	NaN	NaN	9.40e-9 (2)
L3	1.000468e-6	1.00e-6	1.0e-6	6.80e-7	1.00e-6 (3)
L4	3.869000e-9	3.90e-9	NaN	NaN	3.90e-9 (3)
L5	5.877500e-8	5.90e-8	NaN	5.90e-8	5.90e-8 (1)
L6	3.629700e-8	3.60e-8	NaN	3.70e-8	3.60e-8 (3)

```
NaN: no coherent value found
```

(a)

```
Searching in available capacitors libraries...
    ->  Ceramic Capacitors Library: (387 components)
    ->  Aluminium Capacitors Library: (399 components)
    ->  Film Capacitors Library: (309 components)
    ->  Tantalum Capacitors Library: (112 components)
    ->  Polymer Capacitors Library: (47 components)

Findings Summary:
```

No.	Ideal (F)	Ceramic	Aluminium	Film	Tantalum	Polymer	Selected (#)
C1	7.5040e-12	7.50e-12	NaN	NaN	NaN	NaN	7.5e-12 (2)
C2	6.7090e-12	6.70e-12	NaN	NaN	NaN	NaN	6.7e-12 (2)
C3	3.9216e-11	3.90e-11	NaN	NaN	NaN	NaN	3.9e-11 (2)
C4	7.7450e-12	7.70e-12	NaN	NaN	NaN	NaN	7.7e-12 (2)
C5	1.0489e-11	1.00e-11	NaN	NaN	NaN	NaN	1.0e-11 (2)
C6	2.1040e-12	2.10e-12	NaN	NaN	NaN	NaN	2.1e-12 (2)

```
NaN: no coherent value found
```

(b)

Figure 2.36 Technology mapping report: (a) Inductors (b) Capacitors

```
Synthesizing transmission lines...

TLine properties:
    Characteristic Impedance (Zc): 50.0 ohm
    Electrical Length (L): 5.0 deg.
    Frequency (F): 450.0 MHz

Found technologies: (1)
    (1) Microstrip - RO3006
        Substrate Name: Rogers 3006
        Dielectric Constant (Er): 6.15
        Magnetic Constant (Ur): 1
        Loss Tangent (Tand): 0.0025
        Resistivity (Rho): 1
        Substrate Height (H): 25.0 mil
        Metal Thickness (T): 1.42 mil
        Metal Roughness (Sr): 0.095 mil

TLines' properties:
    Width: 35.314567 mil
    Length: 175.551969 mil
    Effective dielectric constant: 4.305
    Total attenuation of the structure: 0.004 dB
    Skin Depth: 0.145

Tees' properties:
    Width (W1): 35.314567 mil
    Width (W2): 35.314567 mil
    Width (W3): 35.314567 mil

Ground connections: via ground (default)
    Radius: 10.0 mil
    Height: (Substrate.H) 25.0 mil
    Thickness: (Substrate.T) 1.42 mil
```

Figure 2.37 Technology mapping report: Transmission lines synthesis

If quite detailed and complete technology libraries are available, the technology mapping process contributes significantly to the automation of the design scheme. Concretely, this process allows to:

– Bridge a gap in existent design tools*:* most popular commercial design packages do not provide the automated selection of technology devices. Some design environments allow the automated synthesis of distributed lines for limited number of circuits (e.g., planar filters in Genesys and Microwave Office);

- Enable fully automated multi-criteria devices selection: given detailed characterization of technology elements, objective functions can be used to select the best components fitting to the PSM circuit;
- Operate in multi-technology design environment: the selection process does not only work with lumped components. It is also possible to automatically synthesize complex planar transmission lines, coaxial cables and multi-layer structures. Specialized tools can be used to handle multi-technology issues (e.g., packaging, connections, discontinuities, etc.);
- Open the door for new types of analyses: the availability of detailed technology data covering not only the electrical but also mechanical, environmental aspects, etc., which enables to carry out different analyses uncommon at this abstraction level. In addition to area estimation, other analyses such as power consumption, carbon fingerprint, and heat dissipation can take place.

In addition, the use of standard formats (such as XML) for technology libraries instead of the prevalent proprietary file formats is another provision that enhances data exchange and communication between concurrent design environments.

2.2.9 Synthesis: PSM performance assessment and optimization

During technology mapping, we choose the real capacitors and inductors to replace ideal PSM lumped components (see the row entitled "selected?" in Table 2.14 and Table 2.15). We have also synthesized the microstrip sections to connect the lumped devices as well as the suitable via holes to ensure ground connectivity. The resulting circuit PSM schematic is shown in Figure 2.31.c.

After linear simulation, we illustrate its frequency response in Figure 2.38. It is obvious that the PSM's overall performance has decreased if compared to the ideal PSM frequency response of Figure 2.34. The reason behind this performance degradation is not only due to the small difference in nominal values between ideal and non-ideal

lumped devices but also caused by their parasitic effects as well as the imperfections of microstrip sections (e.g., the effect of discontinuities, fringing fields, via holes, substrate, etc.).

To meet the requirements, we carried out several optimization iterations in order to minimize the effects due to the imperfections of lumped devices and microstrip sections. We succeed to find out two PSM configurations that meet the requirements. The first is characterized by a relatively constant group delay over a wide range of frequencies but it suffers from a disturbed roll-off in the upper transition band (see Figure 2.39). The second design solution shows interesting reflection and transmission characteristics (see Figure 2.40). We use both candidate solutions to generate two distinct PMs in the next design stage.

It is worth noting that if the optimization effort fails at this step in the search of a PSM that meets the initial requirements, a design re-spin becomes inevitable. Thus, the design will restart at the step of PSM granularity refinement in the purpose of figuring out a new design solution. The new PSM is submitted again to technology mapping and performance assessment respectively.

Table 2.14 List of inductors found in the technology library during automated technology mapping

Ideal Value (nH)	Real Value (nH)	L_0 (nH)	R_{DC} (Ω)	F_0 (MHz)	C_p (pF)	Tolerance	Manufacturer Part Number	Size Dimensions (mm)			Selec-ted?
								Length	Width	Thickness	
350.003	350	350	0.04	120	5.03	±20%	GLFR1608TR35M-LR◈	1.6	0.8	1	
		350	0.039	98	7.54	±20%	SPM5030T-R35M◈	5.2	5	3	✓
		350	0.08	300	0.804	±20%	BRC1608TR35M♥	1.3	0.8	1	
9.428	9.4	9.4	0.081	6000	0.0749	±2%	LQW15AN9N4G80♠	1	0.6	0.6	✓
		9.4	0.081	6000	0.0749	±5%	LQW15AN9N4J80♠	1	0.6	0.6	
1000.468	1000	1000	0.078	150	1.13	±20%	LB3218T1R0M♥	3.2	1.8	2	
		1000	0.00295	147	1.17	±20%	7443340100✦	8.4	7.9	7.5	✓
		1000	0.34	120	1.76	±20%	NLCV25T-1R0M◈	2.5	2	1.9	
3.869	3.9	3.9	0.18	6000	0.18	±0.3nH	LQG15HS3N9S02♠	1	0.5	0.55	
		3.9	0.07	10000	0.0649	±0.1nH	LQW15AN3N9B00♠	1	0.5	0.6	✓
		3.9	0.25	9800	0.0676	±0.3nH	MLK0603L3N9ST000◈	0.6	0.3	0.33	
58.775	59	59	0.2	150	0.191	±10%	WW1008R-59NK✳	2.92	2.79	2.03	✓
36.297	36	36	0.26	2900	0.0837	±2%	LQW18AN36NG00♠	1.6	0.8	1	✓
		36	1.6	1800	0.217	±5%	ELJ-QF36NGF♣	1	0.5	0.55	
		36	0.2	2080	0.163	±5%	744761136C✦	1.65	1.15	1.1	

◈ TDK Corporation ✦ Wurth Electronics
♥ Taiyo Yuden ✳ API Delevan Inc.
♠ Murata Electronics ♣ Panasonic Electronic Components

Table 2.15 List of capacitors found in the technology library during automated technology mapping

Ideal Value (pF)	Real Value (pF)	C_0 (pF)	ESR (Ω)	Tolerance	Manufacturer Part Number	Size Dimensions (mm)			Selected?
						Length	Width	Thickness	
7.504	7.5	7.5	0.1	±0.25pF	GRM0335C1H7R5CA01*	0.6	0.3	0.33	
		7.5	0.1	±0.25pF	GRM0225C1C7R5WA02*	0.4	0.2	0.22	✓
6.709	6.7	6.7	0.1	±0.25pF	GRM0335C1H6R7CA01*	0.6	0.3	0.33	
		6.7	0.1	±0.1pF	GRM0332C1H6R7BA01*	0.6	0.3	0.33	✓
39.216	39.0	39.0	0.16	±2%	GRM0332C1H390GA01*	0.6	0.3	0.33	✓
		39.0	0.15	±5%	GRM0335C1H390JA01*	0.6	0.3	0.33	
7.745	7.7	7.7	0.2	±0.1pF	GRM0332C1H7R7BA01*	0.6	0.3	0.33	
		7.7	0.2	±0.05pF	GRM0332C2A7R7WA01*	0.6	0.3	0.33	✓
10.489	10.0	10.0	0.2	±2%	GRM0332C1H100GA01*	0.6	0.3	0.33	
		10.0	0.15	±5%	GRM1552C1H100JA01*	0.6	0.3	0.33	✓
2.104	2.1	2.1	0.28	±0.1pF	GRM0333C1H2R1BA01*	0.6	0.3	0.33	✓
		2.1	0.29	±0.05pF	GRM0335C1H2R1WA01*	0.6	0.3	0.33	

* Murata Electronics

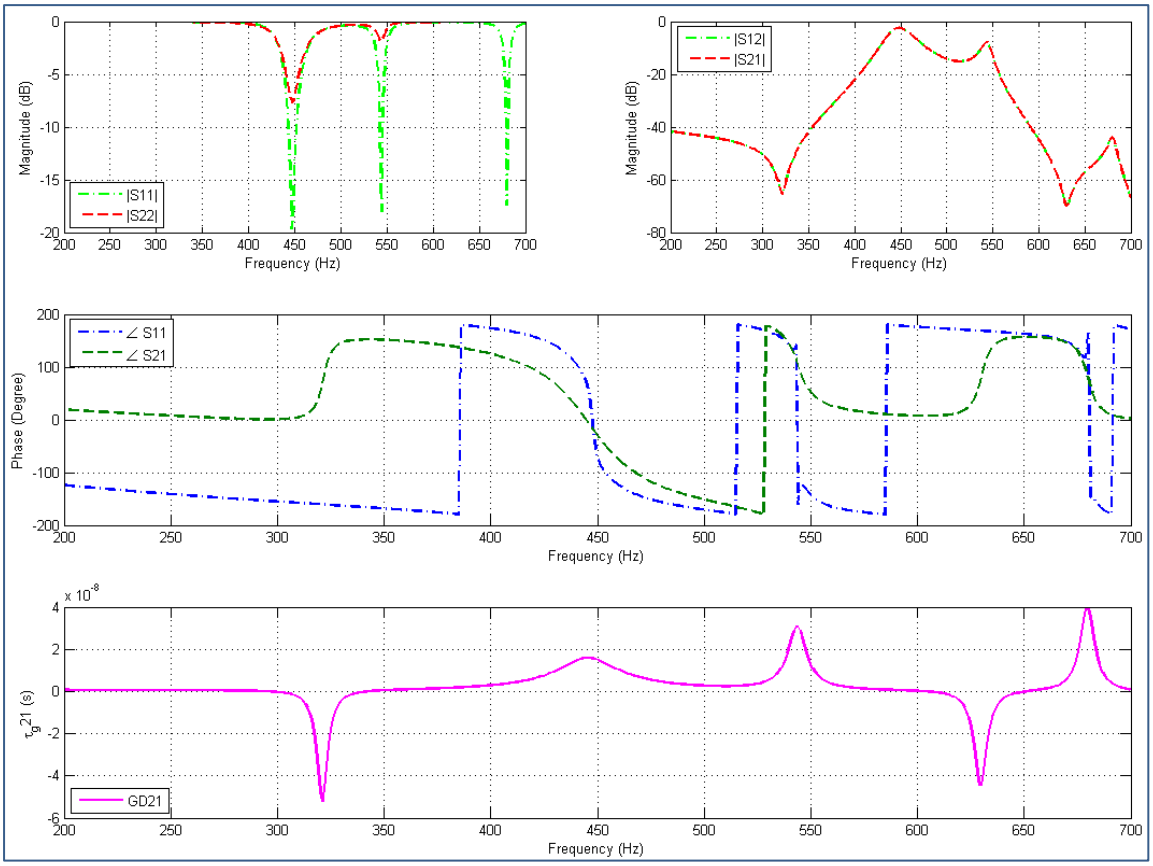


Figure 2.38 The PSM's frequency response after technology mapping

2.2.10 Synthesis: Platform model generation

Having two PSM candidate solutions (with enough technology data) that relatively meet most of the requirements captured in the initial functional description, we proceed henceforth to the generation of the corresponding platform model in the purpose of the physical implementation of the target filter.

- **Platform models**

To generate the platform models corresponding to both PSMs, we need a PSM-to-PM transformation that converts the PSM circuit-level artifacts (e.g., schematic, technology models and constraints, etc.) into a PM physical-level layout. So, the PM should reflect

at physical level (e.g., layers, shapes, routes, etc.) what the PSM captures at circuit level (e.g., types and order of elements, connectivity, etc.).

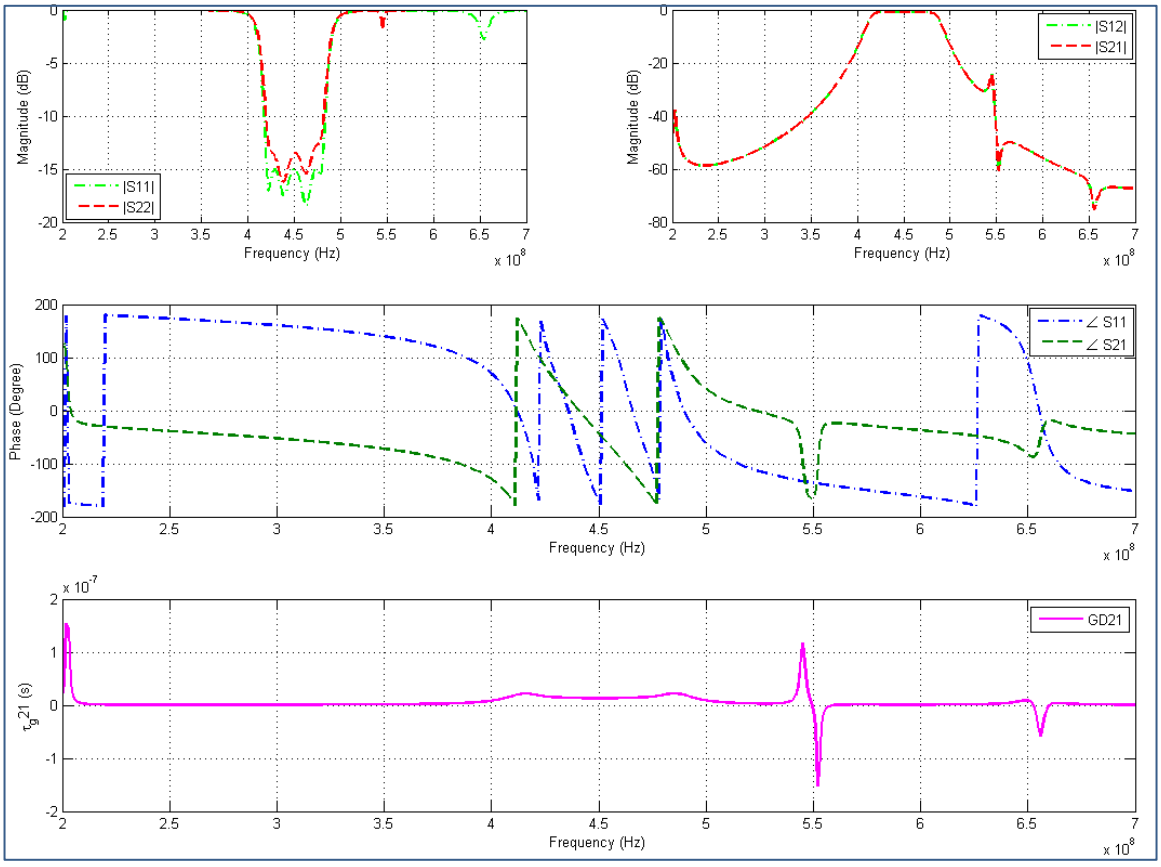


Figure 2.39 First candidate solution (optimization round 1)

Given the detailed PSM components and technology information, the next step is the generation of the corresponding layouts (i.e., PM). For simplicity, we use a commercial layout generation tool (e.g., Genesys) as PSM-to-PM transformation. The platform models corresponding to the first and second PSMs are shown in Figure 2.41 and Figure 2.42 respectively. These figures show a single 2D and three 3D views of each layout. Microstrip sections and via holes are fully captured in 3D view. Nevertheless, the lumped components visible on 2D layouts are not shown on 3D ones. The reason behind this is that the software tool has no information about their three-dimensional

representations because these components are off-the-shelf (i.e., available technology data is limited to 2D aspects). To be considered in the 3D layout view, the designer should manually import each device's layout. This said, the spacing between the microstrip sections takes into account the dimensions of each lumped device.

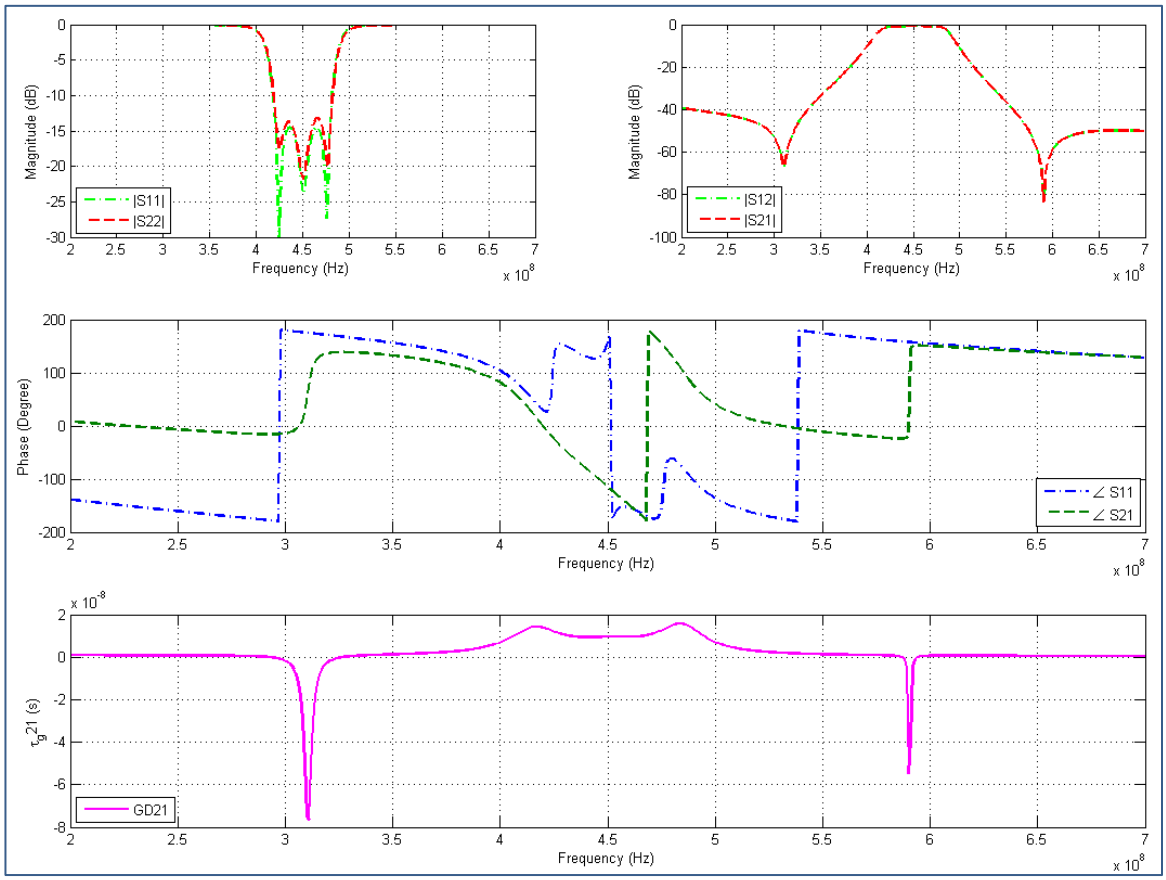


Figure 2.40 First candidate solution (optimization round 2)

- **PM selection and performance assessment**

After PSM-to-PM generation, we should validate both PMs against the initial functional description not only in terms of electrical performance (e.g., frequency response) but also considering the other requirements captured in the diagram of Figure 2.11 (e.g., form factor, lightening protection, EMC requirements and humidity resistance).

In this case study, the available technology data are limited and provided us only with thorough information about the filter's form factor. Knowing the physical dimensions and the size of each PM component, the area and volume of the PM can be estimated with good accuracy. Hence, Table 2.16 presents the maximum length, width and PCB area required for the manufacturing of each platform model. The first PM needs more than twice the area occupied by the second one. For a comparable frequency response, the first PM can be discarded since the filter is required to be as small as much. The advantage of making such estimation early enough before prototype manufacturing is preventing additional development costs.

Unfortunately, we are not able at this step to evaluate the PM against the other requirements (i.e., lightening protection, EMC and humidity resistance requirements). So, it becomes mandatory to postpone these assessment tasks to the test and measurements stage. This lack is not inherent to the design framework but is related to:

- The quality of technology data: the provision of comprehensive physical, electrical and mechanical data as input to the technology mapping step enables thorough characterization of both PSM and PM at circuit and device levels. If some characterization aspects are missing, it is generally not possible to evaluate PSMs and PMs regarding that specific aspect;
- The availability of highly specialized tools: to validate particular behavior of both PSMs and PMs, specialized tools are required. For example, an EM solver uses the PM's structure information (e.g., mechanical data, materials' properties, etc.) in order to assess the PM's response against EM excitation (e.g., electrical and magnetic fields distribution and strength, signal attenuation levels, etc.).

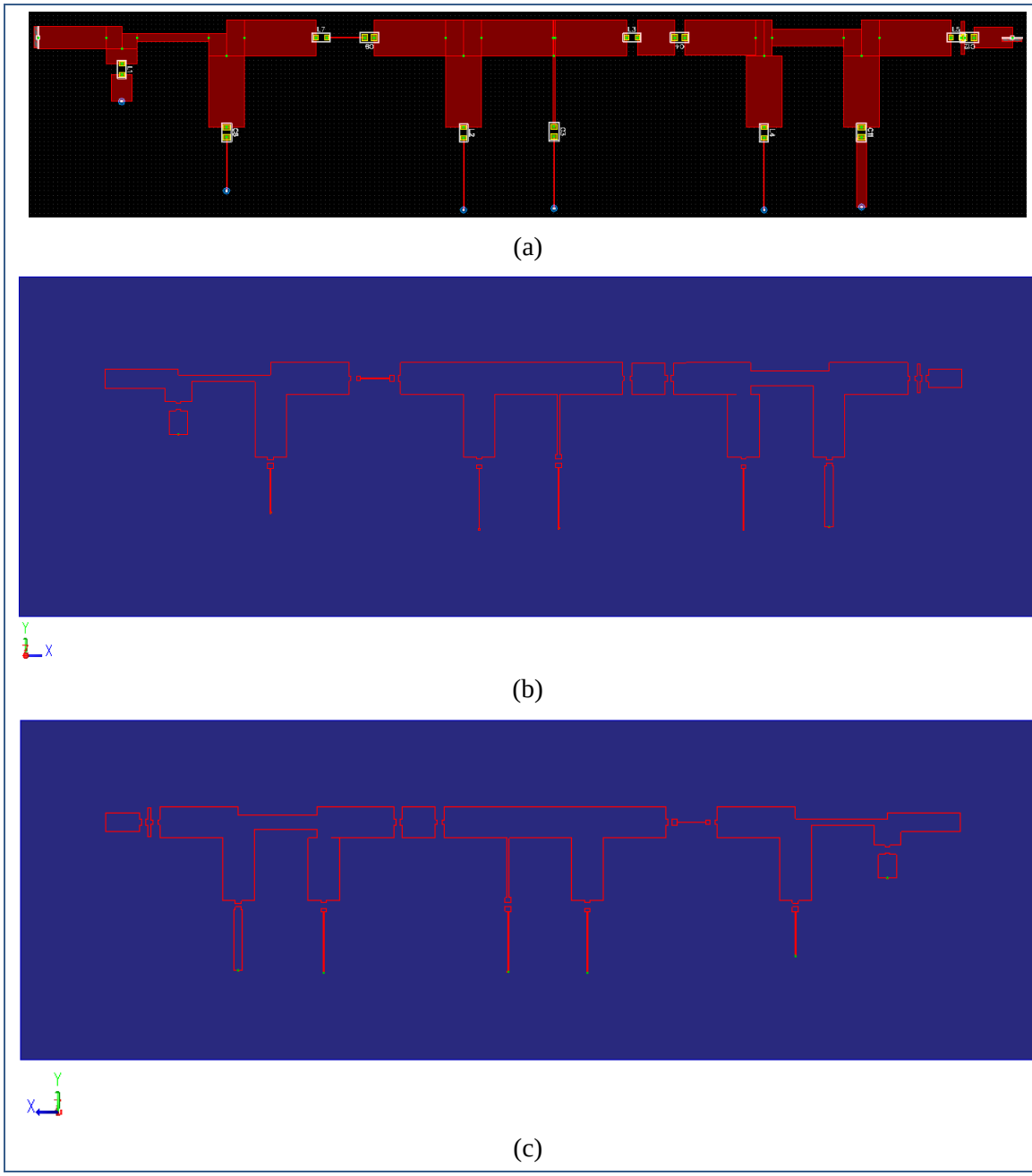
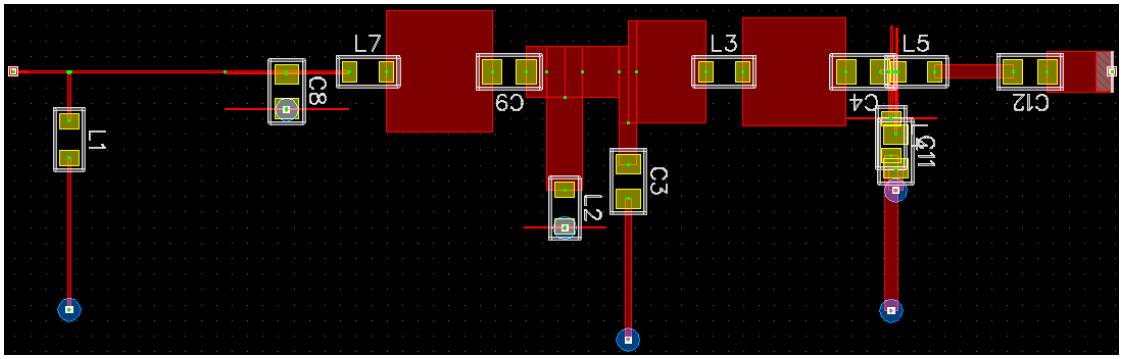
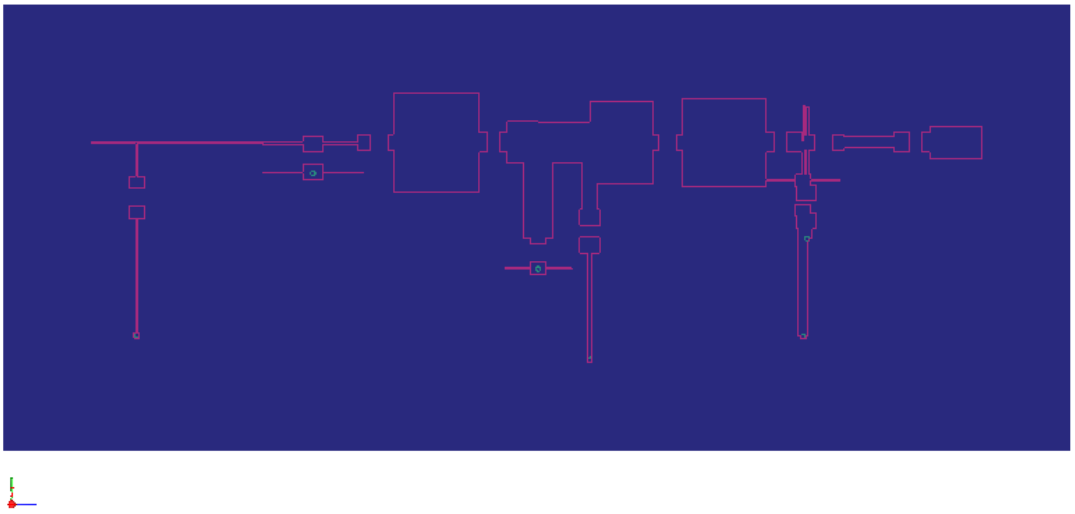


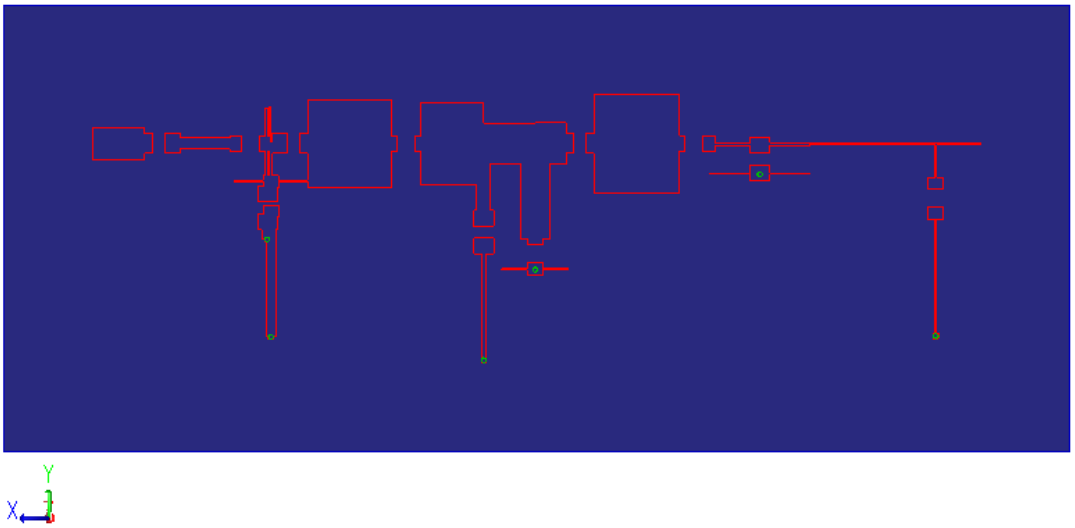
Figure 2.41 Platform model (first optimized PSM): (a) 2D layout (b) 3D layout top and (c) bottom views



(a)



(b)



(c)

Figure 2.42 Platform model (second optimized PSM): (a) 2D layout (b) 3D layout top and (c) bottom views

Table 2.16 Comparison of PCB areas required to hold the generated platform models

	Maximum Length (cm)	**Maximum Width (cm)**	**Maximum Area (cm^2)**
PM 1 (PSM opt. 1)	26.5710328	5.1156499	135.9281
PM 2 (PSM opt. 2)	8.0287038	7.5807504	60.8636

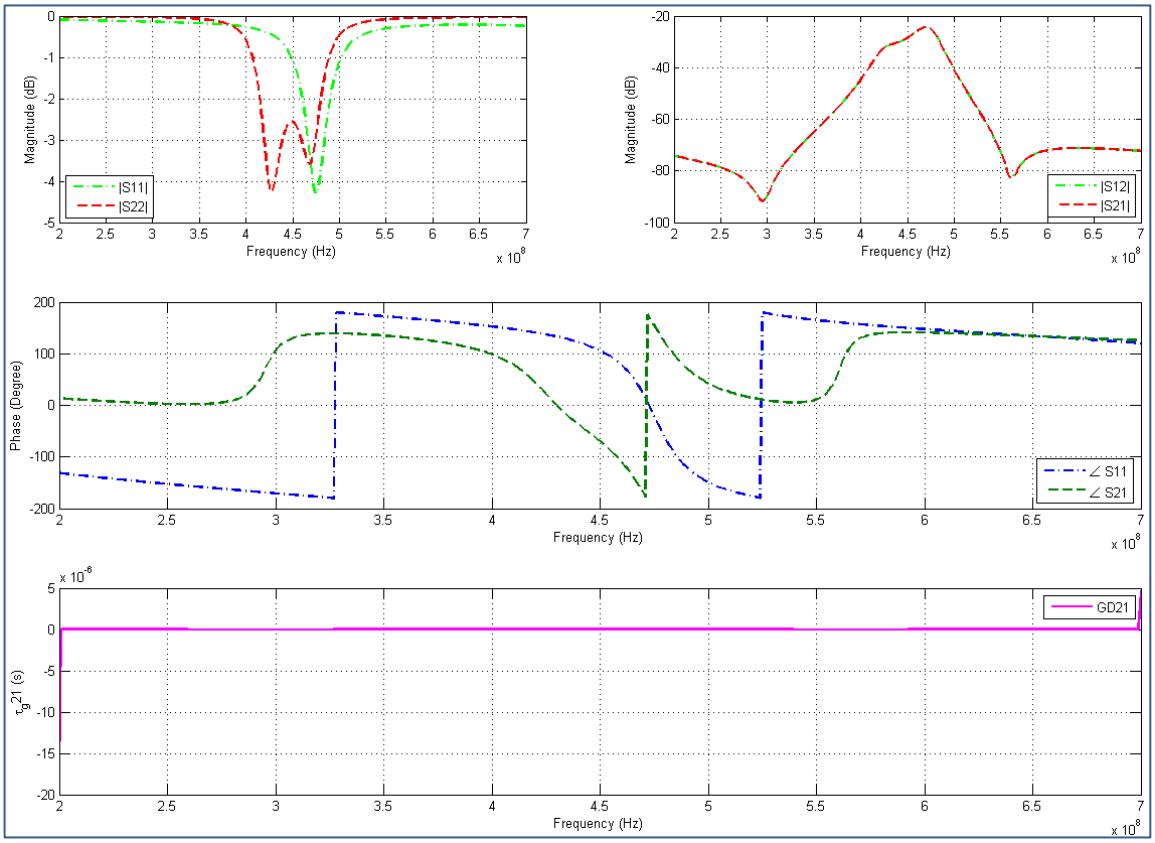


Figure 2.43 Frequency response of the PM model (PSM-to-PM transformation output)

Before the PM manufacturing, we carry out a hybrid simulation (EM/circuit) to evaluate its frequency response. As illustrated in Figure 2.43, the attenuation levels in the

passband are poor. This performance decrease is mainly due to the tools accuracy at both circuit- and device-levels. The PSM performance is evaluated using circuit-level tools based on electrical models while the PM frequency response is obtained after EM simulation. On the contrary to the former, the latter takes into account physical-level artefacts such as the layout structure properties. The EM solver simulating the physical structure of the PM is more accurate than the electrical models used at circuit-level for the evaluation of the PSM. Before fabrication, this PM requires layout-level optimizations. However, manufacturing is not the main purpose of this case study. Therefore, we give up the design process at this step.

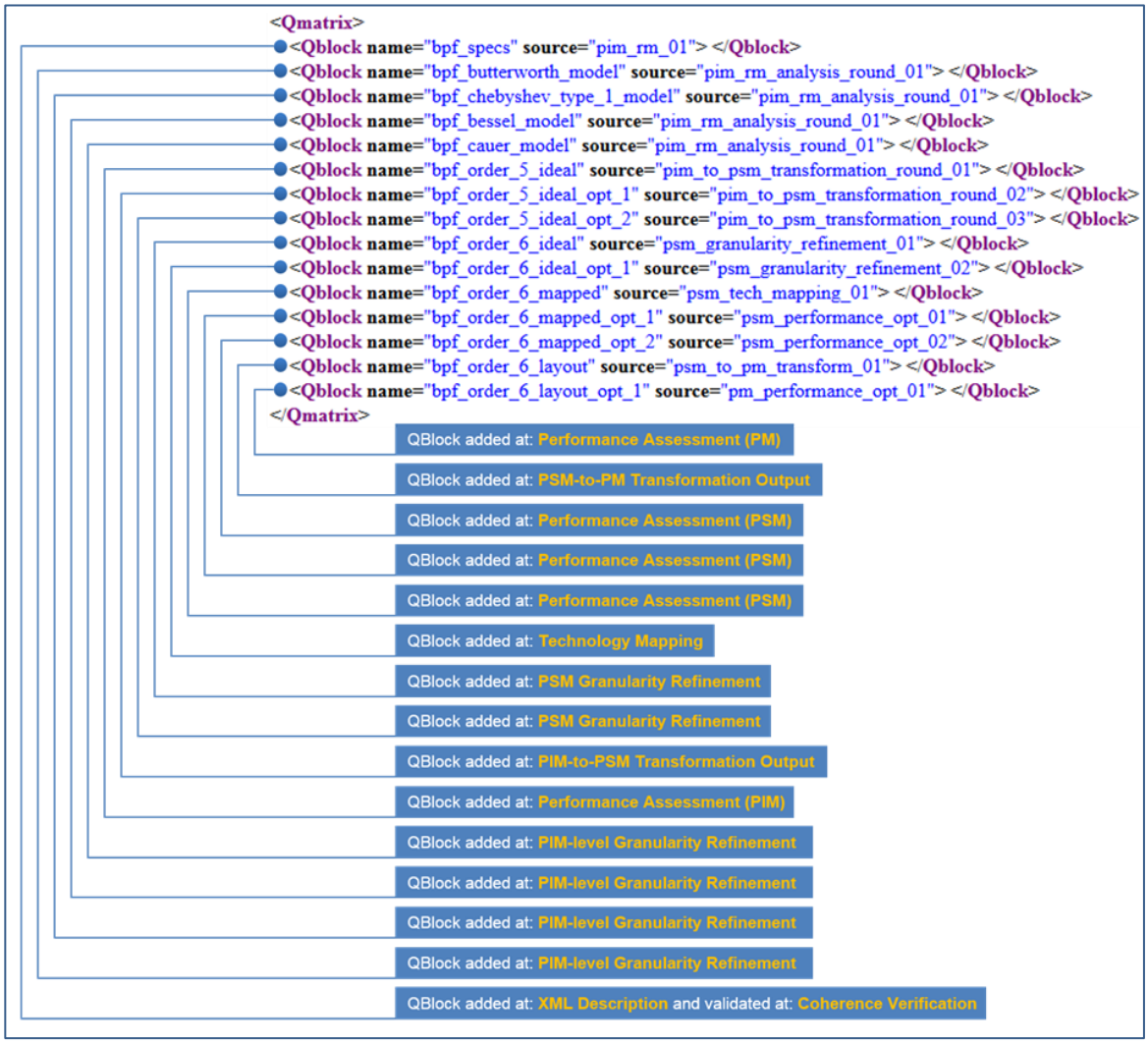


Figure 2.44 Overview of the QBlocks added to the Q-matrix throughout the filter's design process

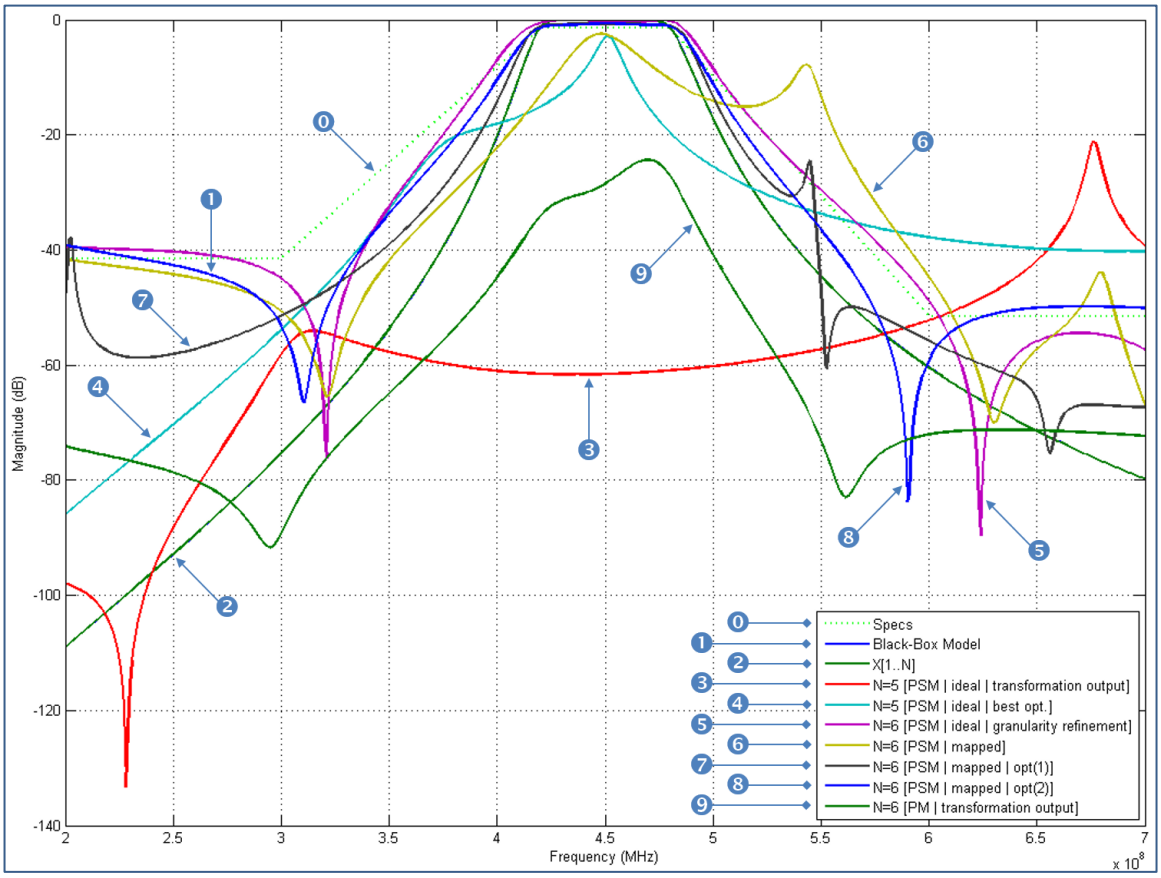


Figure 2.45 The Q-matrix centralizes design data, which allows for example to make performance comparisons throughout the design process

2.2.11 Q-matrix evolution

In this first case study, we illustrated in Figure 2.22 the Q-matrix created at the end of the functional description stage (i.e., XML description step). However, we did not so far comment its role and practical use in the design scheme. This choice is deliberate to prevent ambiguity and confusion between the design scheme stages and Q-matrix. In this section, we focus on the construction and evolution of the Q-matrix used for the previous bandpass filter design.

The design cycle of the bandpass filter required three model-to-model transformations, several analyses and optimization iterations. Each of these mechanisms added new design data that were held in new QBlocks of the Q-matrix. At this regard, Figure 2.44 shows the most important QBlocks added throughout the design process. Each of these QBlocks has its own name and data source. It gathers the electrical data that resulted from a given design step. In this Q-matrix, all design data originated from either functional description or simulations.

As discussed in section 1.3.1.c of (Lafi, 2015), the Q-matrix centralizes and enables concurrent sharing of design data between tools and designers. For instance, this feature is useful when comparisons between multiple design solutions are required. As illustrated in Figure 2.45, we used the Q-matrix to compare the performance of the bandpass filter at different design steps. Since the data is available in the same XML file, it is easy to select and plot the attenuation response of the filter in some key design steps. Using a popular commercial design package such as ADS, the Figure 2.45 requires at least ten different datasets (or/and data files) to be plotted. To export such data using the same tool, ten different touchstone files are generated. With the proposed Q-matrix structure, this effort is minimized since all the data is available in the same file.

Another interesting feature of the Q-matrix is the possibility of storing fragmented data. For example, no data is stored for frequency points having no attached information. This is not possible in most commercial tools. To do so, the designer should carry out complicated operations in order to discard all the frequency points where no data is available. For illustration, Figure 2.46 shows a single frequency point (i.e., 420 MHz) in different QBlocks added at different design stages. In the QBlock (no. 2) resulting from the coherence verification, there is no data about the reflection levels needed at 420 MHz. This is expected because the specifications and the functional description do not provide that information. In the following step (i.e., analysis), the filter approximations (i.e., Butterworth, Chebyshev Type I, Elliptic and Bessel) associated to the black-box

model provided both transmission and reflection coefficients at 420 MHz. The associated QBlocks (no. 3) stored this information. Another example is the temperature considered for simulations. The black-box model does not consider the temperature parameter. That is why the temperature value (in QBlock no. 3) is set to "F" (i.e., "Forget" which literally means "do not consider"). On the contrary, the LC model (in QBlock no. 4) computed in the second step of the PIM-to-PSM transformation takes into account the standard temperature (i.e., 16.85°C). Thus, the temperature information becomes available in the QBlocks of following design steps.

2.2.12 Additional remarks

Throughout the design cycle, we developed various filter prototypes and used different assessment techniques to validate each of them against the initial requirements.

- **Models evolution throughout the design cycle**

This case study started with the functional description that consisted of developing high-level models for the description of the filter properties, structure and requirements. Then, we associated a black-box model to this functional description in the purpose of deriving a platform-independent model. Using appropriate model-to-model (cross- and intra-view) transformations, we generated circuit- and device-level models that implement the bandpass filter. An overview of the main models developed at each design stage is already shown in Figure 2.47.

It is worth noting the transition between the high-level functional description and the physical-level layout artwork that resulted from an abstraction level raise. The gap between both types of models is larger than what is actually available in the current design practice. This is due to the abstraction strategy that defined concepts and mechanisms allowing to:

a. Develop high-level platform independent models for functional description,
b. Associate appropriate black-box models to each functional description, and

c. Derive platform-specific and implementation models from their high-level counterparts using relevant model-to-model transformations.

- **Assessment techniques**

Raising the abstraction level and subdividing the design space into three contiguous domains (i.e., RM/PIM, PSM and PM) arranged assessment and simulation techniques into three categories:

1. PIM-level simulation techniques: In RM/PIM domain, models are evaluated using coherence verification rules for the functional description and mathematical equations for black-box models. These formalisms do not take into account any platform-specific details;
2. PSM-level simulation techniques: In PSM domain, more accurate simulation techniques (e.g., circuit) are used because some platform specifications are known;
3. PM-level simulation techniques: In PM domain, the physical-level validation requires more specialized techniques that take into account the physical aspects at device level.

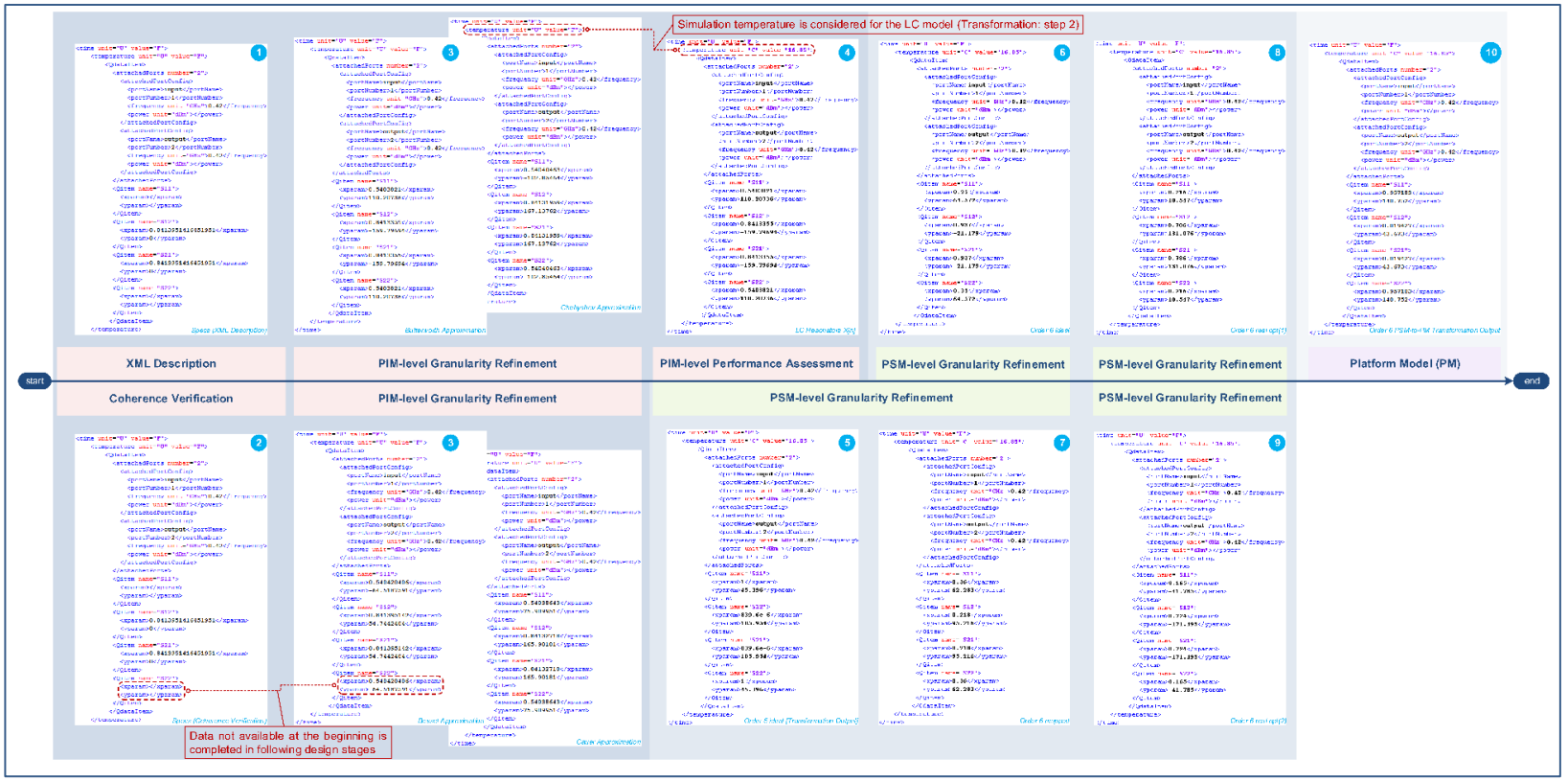


Figure 2.46 An overview of Q-matrix QBlocks evolution throughout the design cycle

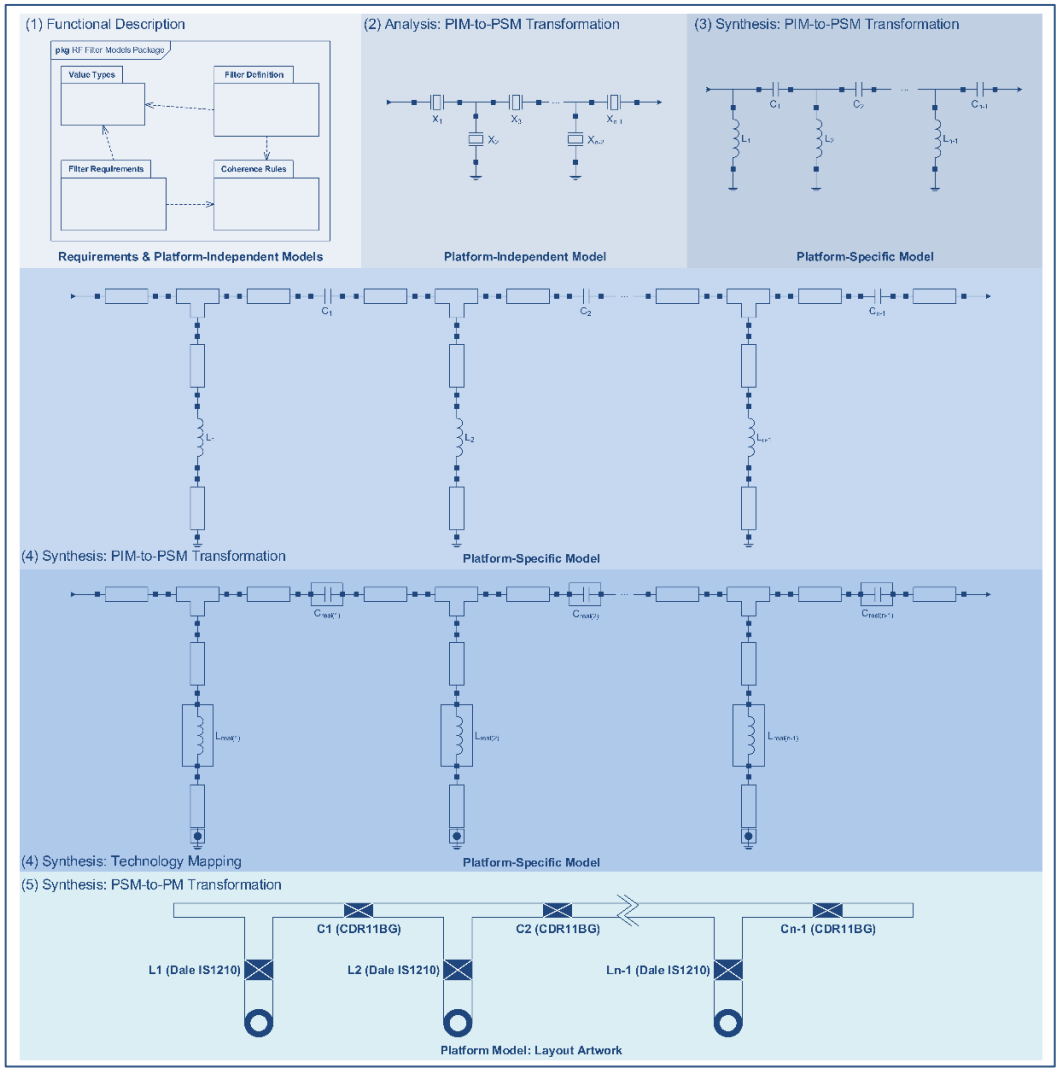


Figure 2.47 The evolution of the filter models throughout the design cycle

To illustrate this observation, Figure 2.48 maps the simulation techniques we used in this case study to the different filter models and design viewpoints. However, this mapping remains relative because some simulation and validation techniques of an upper-level domain might be used in an underlying one. For example, the frequency response of the filter's PM (see Figure 2.43) was obtained after using a hybrid simulation: an EM solver computed the frequency response of the microstrip sections while the lumped components one was determined using a circuit-level simulation. The EM simulation could not compute the entire structure frequency response because the physical layout of lumped components is missing.

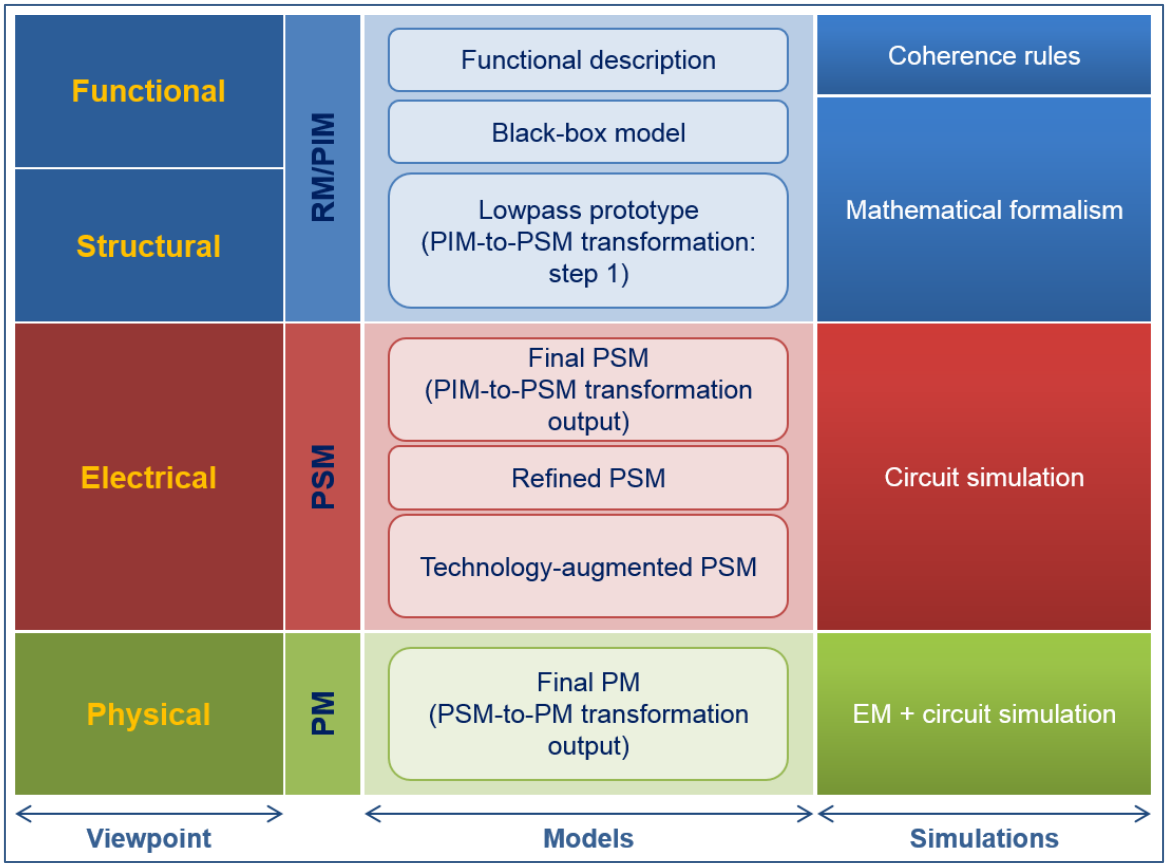


Figure 2.48 Simulation techniques versus filter models and design viewpoints

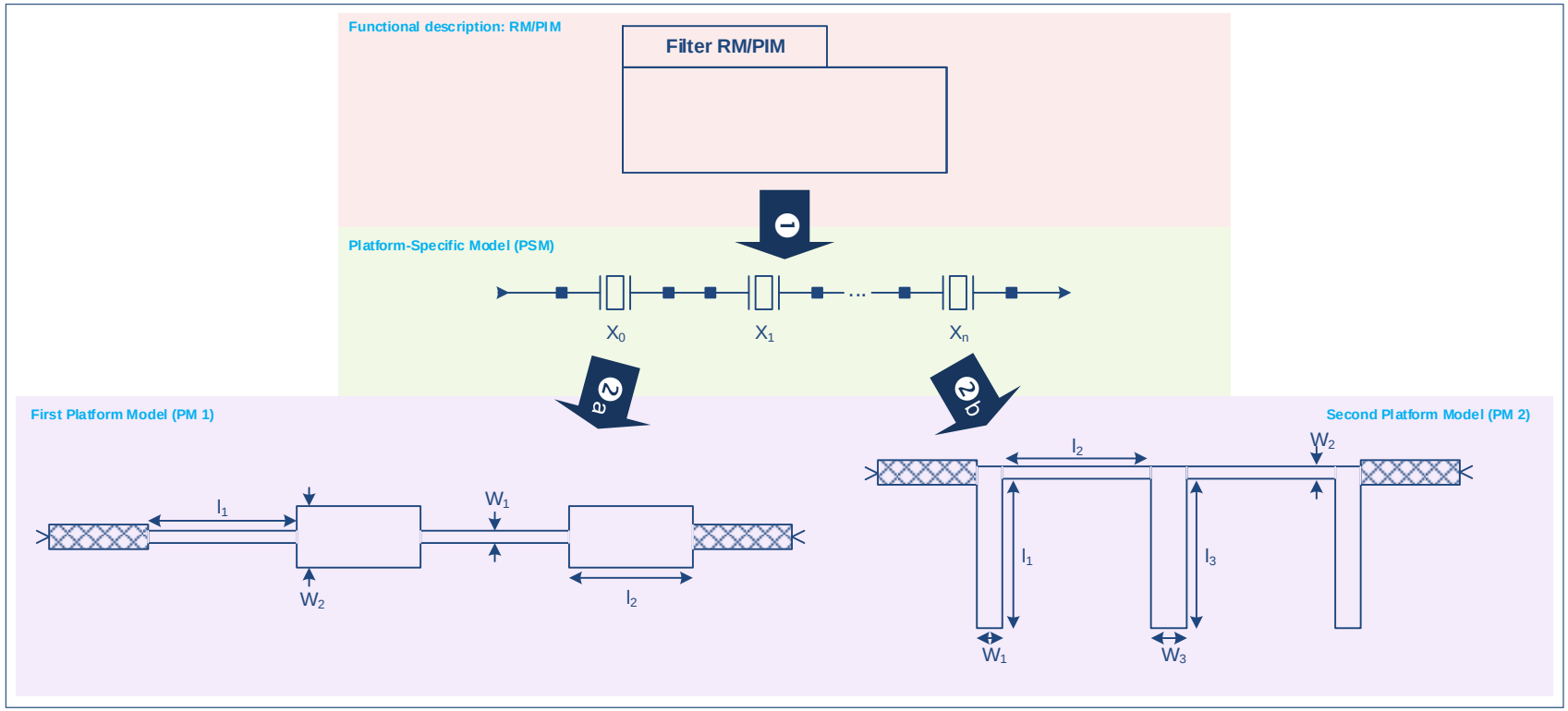


Figure 2.49 Using two cross-view transformations, the same lowpass filter PSM is derived into two separate platform models

2.3 A 1-GHz microstrip lowpass filter

In this section, we attempt to illustrate how the proposed framework can be used for the design of a lowpass filter where multiple transformations are involved. As shown in Figure 2.49, we will use two PSM-to-PM transformations in order to derive two physical implementations of the lowpass filter. Unlike the previous case study, we will not detail all the design steps in the following. Only the most outstanding design stages are presented.

2.3.1 Specifications

We aim at designing a lowpass filter that meets the specifications enumerated in Table 2.17. A lowpass filter is a frequency selection device that passes the signals lower than a certain cutoff frequency (F_c) while it attenuates those higher than F_c. Figure 2.50 illustrates the main parameters of a typical lowpass filter.

Table 2.17 Lowpass filter specifications

Filter Type	Lowpass
Passband Attenuation (dB)	< 0.1
Stopband Attenuation (dB)	> 25.0 @ 1800 MHz
Cutoff Frequency (MHz)	1000
Termination Impedance (ohm)	50
Other Requirements	Small form factor

2.3.2 Functional Description: SysML requirements and platform-independent models

Using the same SysML models developed for the functional description of the bandpass filter and presented in the previous section, the lowpass filter functional description is developed alike. The resulting RM/PIM models are elaborated as given in the package diagram of Figure 2.6.

The RM/PIM consists of four main package:

- Filter definition: similarly to the bandpass, the lowpass filter is a two-port network that is hierarchically structured as illustrated in Figure 2.7 (see detailed bdd in Figure 2.8). On the contrary to the bandpass, the lowpass device is a single-side filter. Thus, it inherits the properties of a single-side instead of a mid-band filter (see Figure 2.9). This difference is illustrated in Table 2.18 where the properties "*centerFrequency*" and "*bandwidth*" are replaced by their counterparts "*cutoffFrequency*" and "*stopbandEdgeFrequency*" (see gray-colored rows in Table 2.18). In addition, the value property belonging to the bandpass bdd, namely "*fractionalBandwidth*" is ignored in the lowpass case because it is has no definition;
- Filter requirements: the lowpass requirements include performance and form factor specifications (see requirement diagram of Figure 2.51);
- Filter coherence rules: the lowpass coherent rules consist of the PIM-level design constraints, the electrical consistence rules, and the integrity control rules. The first rules are the same as for the bandpass filter while the second and the third ones are slightly different.

2.3.3 Analysis: Coherence verification

As previously mentioned, the lowpass filter functional description include three types of coherence rules:

1. PIM-level design constraints: are the same as in Table 2.6;
2. Electrical consistence rules: include two sets of rules. The first is represented in the parameters relationships graph of Figure 2.52 and the corresponding equations are enumerated in Table 2.19. The second set consists of equations E.1 and E.2 of Table 2.5;
3. Integrity control rules: include the rules I.1 through I.32 of Table 2.7. The rules I.34 and I.35 are used only with mid-band (such as bandpass filters).

The functional description elaborated in the previous steps is checked using the coherence verification rules. The resulting coherence verification report is given in Figure 2.53.

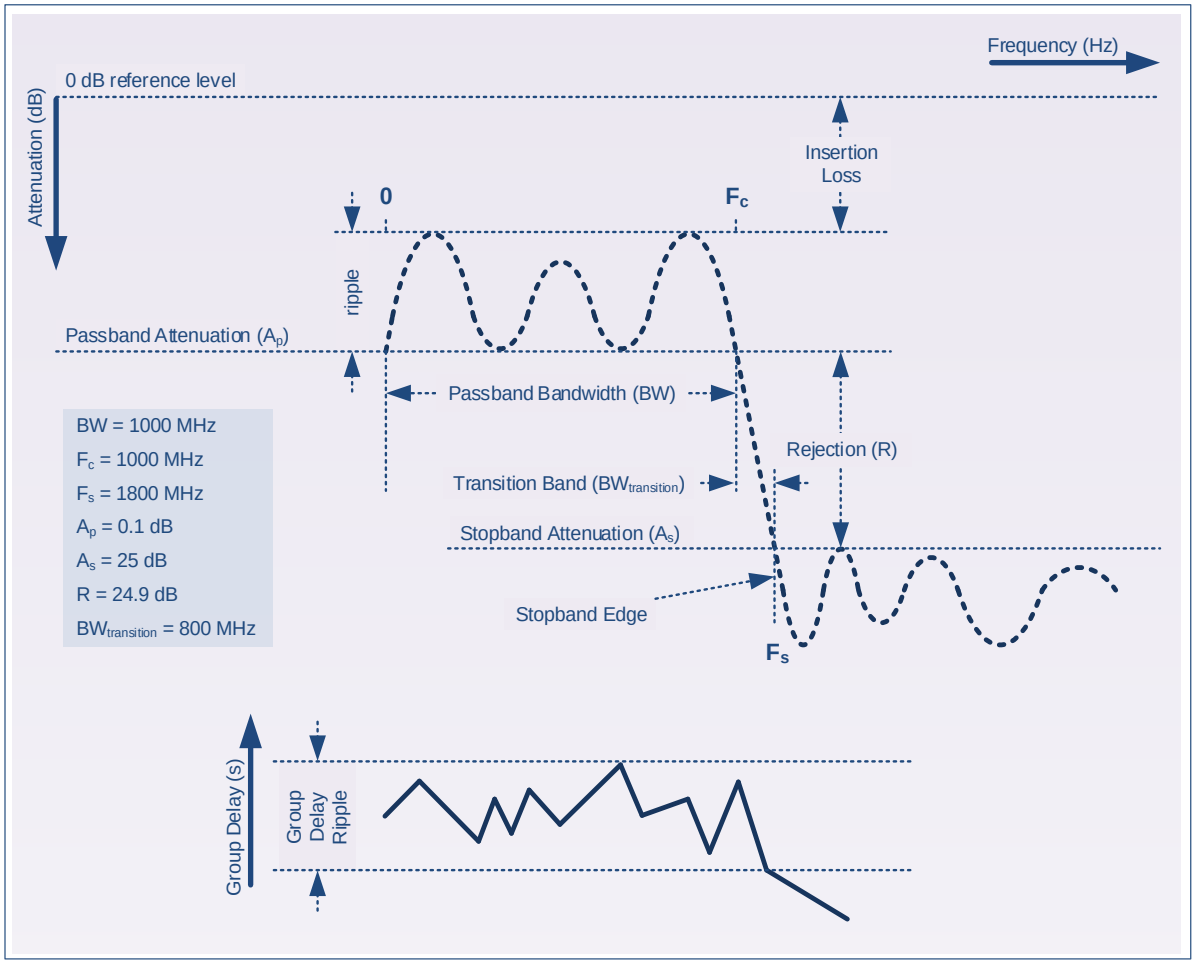


Figure 2.50 Typical parameters of a lowpass filter

2.3.4 Analysis: Design space exploration

After submitting the functional description to the coherence verification, it is associated to a black-box model. In this case, we choose a Chebyshev Type I filter approximation as a black-box model (as given in equations A-8, A-9 and A-10 in APPENDIX A, pp. 115-116). The design space exploration finds out an initial candidate design solution that

satisfies the filter requirements. The black-box model has a frequency response as illustrated in Figure 2.54.

2.3.5 Analysis: PIM-to-PSM transformations

As initially specified, we aim at using two separate PIM-to-PSM transformations in order to derive two different filter PSMs. This exercise is intended to verify the capacity of the design framework to enable concurrent design.

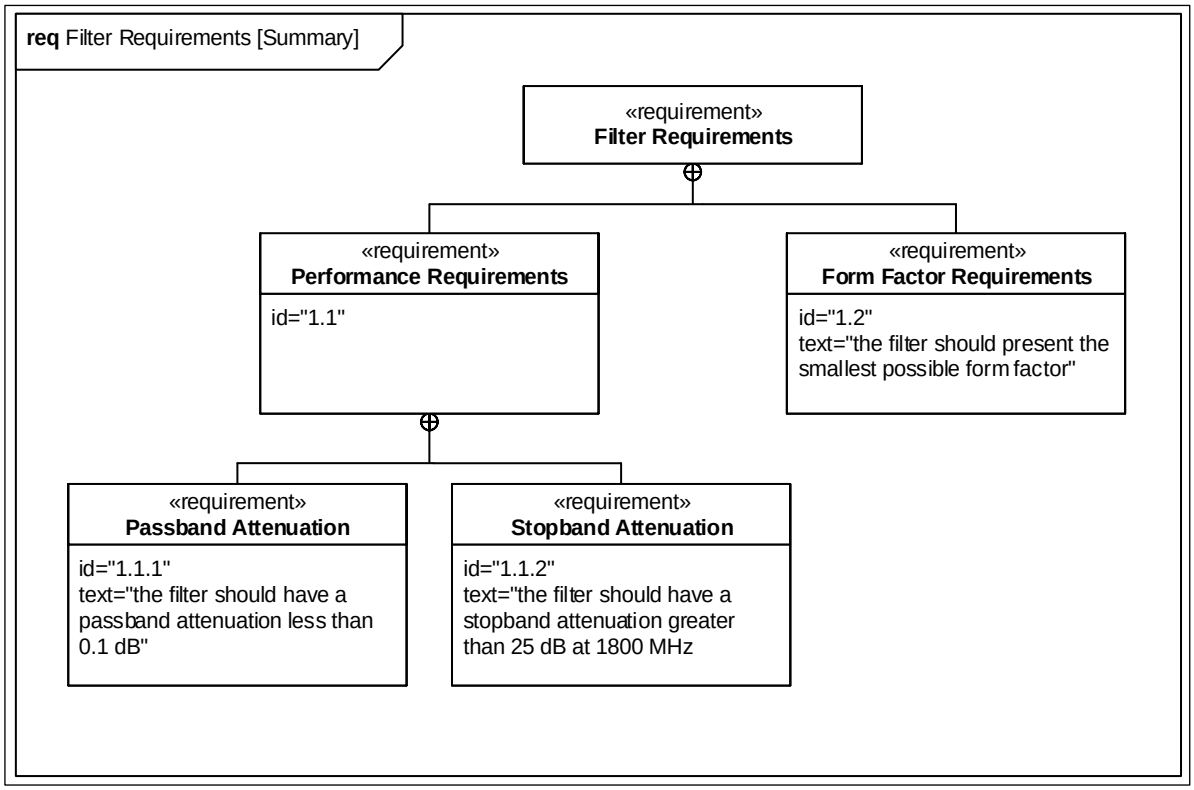


Figure 2.51 The requirements diagram of the lowpass filter

Table 2.18 List of lowpass filter value properties (port value properties are not considered)

Parameter Value	Type	Default Value	Remarks	Specifications Value
temperatureRange	interval (Celsius)	[-40, 80]		

Parameter Value	Type	Default Value	Remarks	Specifications Value
ratingTemperature	Boolean	TRUE	TRUE FALSE	
portList	Object (Port)			
portsNo	integer	2		
fractionalBandwidth	double	20%		ignored
gain	array (dB)			Overridden by *passbandAttenuation* and *stopbandAttenuation*
qFactor	double			
isolation	dB			Overridden by *filterSelectivity*
type	filterType	Butterworth	Bessel Butterworth Chebyshev Type I Chebyshev Type II Custom Elliptic	
ripple	array (dB)	0.25		
passbandAttenuation	array (dB)	3.0		
stopbandAttenuation	array (dB)	60.0		25.0 @1800 MHz
groupDelayRange	interval (seconds)	[0, 5E-6]		
ratingGroupDelay	Boolean		TRUE FALSE	
groupDelayRipple	interval (seconds)			
ratingGroupDelayRipple	Boolean			
order	integer			
shapeFactor	double			
filterSelectivity	dB			

Parameter Value	Type	Default Value	Remarks	Specifications Value
cutoffFrequency	MHz			1000.0
stopbandFrequency	MHz			1800.0

We choose that the first PIM-to-PSM transformation generates a stepped-impedance-based lowpass filter PSM while the second derives another PSM that is based on open-end distributed lines:

1. Stepped-impedance lines transformation: it uses very high and very low characteristic impedance distributed lines to derive the inductive and capacitive sections of the filter PSM (see Table A-2 in APPENDIX A, p. 119). The characteristic impedances depend on the type of distributed lines. In this case study, we use the mathematical formalism described in (Pozar, 2012, p. 470) to calculate each section properties but other formalisms can also be considered (e.g., Hong et Lancaster, 2001, pp. 112-115). Table 2.20 presents the three main steps of the transformation;
2. Open-end stub-based transformation: it uses large open-end circuit stubs to approximate capacitive elements while it considers thin distributed-line sections for inductive ones. To do so, it takes part of Richard's transformation that allows converting a capacitive (respectively inductive) lumped element into a distributed-line section approximating the same capacitive (respectively inductive) effect. In addition, it uses Kuroda's identities to separate the resulting filter elements using distributed-line sections (see Table A-4 and Figure A-1 in APPENDIX A, pp. 120-121). For simplicity, we use in this transformation steps the calculation method given in (Pozar, 2012, pp. 462-469). Other computation techniques are also available (e.g., Hong et Lancaster, 2001, pp. 115-119). The transformation consists of four key steps as illustrated in Table 2.21.

In practice, the framework tools allow the parallel use of both transformations to derivate target PSMs from a unique source PIM. For clarity, we present hereafter each transformation separately.

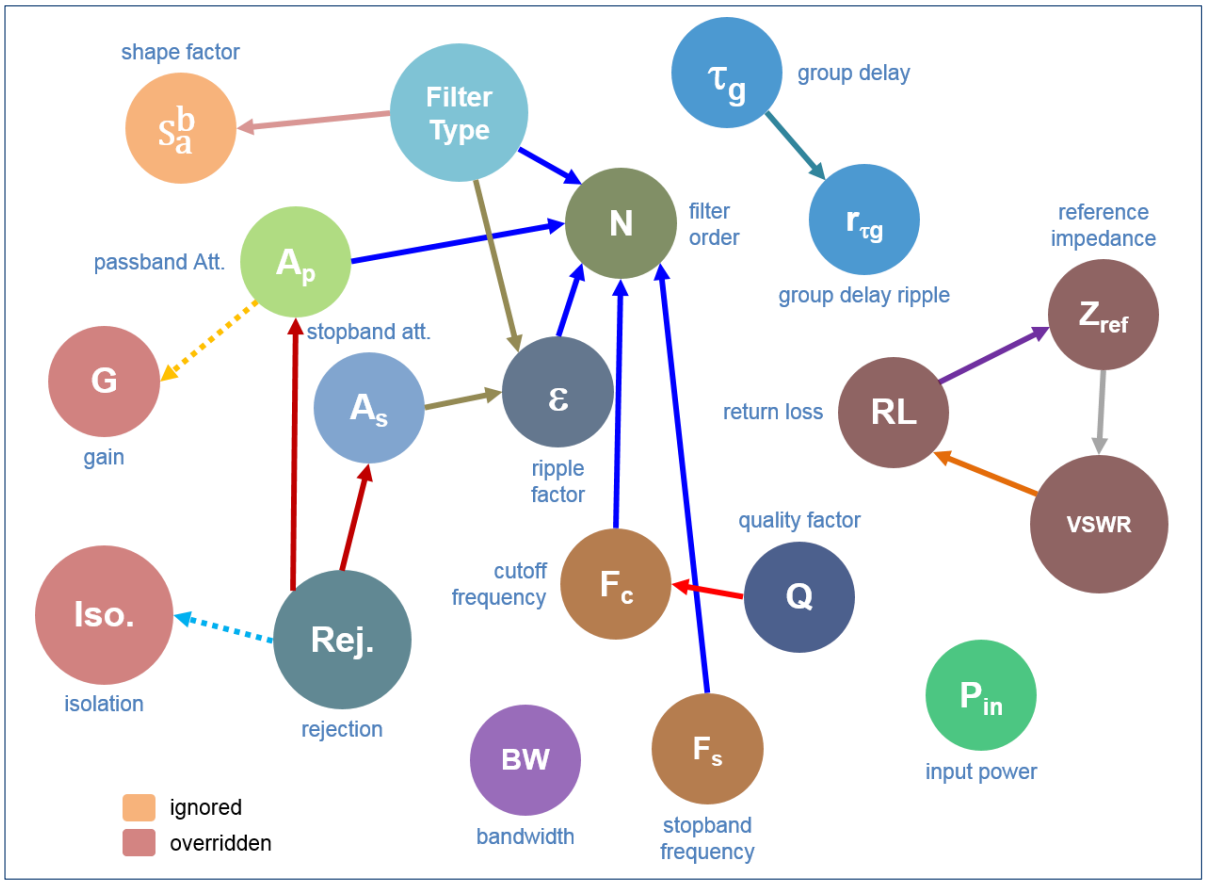


Figure 2.52 Lowpass filter parameters relationships graph

Table 2.19 Relationships between lowpass filter value properties given in Figure 2.52

Arrow Color	Mathematical Relationship	Parameter
●	$R = A_s\|_{F_s} - A_p\|_{F_p}$	Rejection (filter selectivity)

●	$N = \frac{log\left(\frac{10^{\frac{A_s}{10}} - 1}{10^{\frac{A_p}{10}} - 1}\right)}{2log\left(\frac{\omega_s}{\omega_p}\right)}$	Filter Order (Butterworth)
●	$\varepsilon = \frac{1}{\sqrt{10^{\frac{A_s}{10}} - 1}}$	Ripple Factor (Chebyshev)
●	$r_{\tau g} = \max(\tau_g)\big\|_{[0,F_p]} - \min(\tau_g)\big\|_{[0,F_p]}$	Group Delay Ripple
●	$S_a^b = \left[\frac{10^{A_s\|_{F_s}/10} - 1}{10^{A_p\|_{F_p}/10} - 1}\right]^{1/2N}$	Shape factor (Butterworth)
●	$Q = \frac{\omega_c}{2\cos\theta_k}$	Quality Factor (Butterworth)
●	$VSWR = \frac{10^{\frac{RL}{20}} + 1}{10^{\frac{RL}{20}} - 1}$	Voltage Standing Wave Ratio
●	$RL = \frac{Z_L - Z_S}{Z_L + Z_S}$	Return Loss
●	$A_p = isolation$	Passband Attenuation overrides Isolation property
●	$R = G$	Rejection (filter selectivity) overrides Gain property

```
******** Coherence Verification Test ********
TARGET PIM: 1-GHz LOWPASS FILTER (C:\Users\slafi\Documents\Design\lpf1GHz_pim.xml)

-> Electrical Consistence Rules (12 found)
    ..................................... PASS
-> PIM-level Design Constraints Rules (1 found)
    ..................................... PASS
-> Integrity Control Rules (33 found)
    ..................................... PASS
    Summary: (0) Warnings (0) Errors
```

Figure 2.53 Lowpass filter coherence verification report

a) Stepped-impedance lines transformation

2.3.6 Analysis: PSM generation

The first transformation composed of three steps derives distributed-line sections to approximate capacitive and inductive components.

1. Step 1: Determination of high and low characteristic impedances depending on target distributed lines

The low and high characteristic impedances for capacitive and inductive line sections are chosen at this step with respect to the following considerations (already enumerated in Hong et Lancaster, 2001, p. 113):

- System reference impedance: The reference impedance (commonly $Z_0 = 50\Omega$) is higher (respectively lower) than the low (respectively high) characteristic impedance (i.e., $Z_{Low} < Z_0 < Z_{High}$).
- Feasible dimensions of capacitive sections: The lowest Z_{Low} is, the better the lumped-element capacitor approximation is. The lowest Z_{Low} is, the largest the corresponding line width W_C is. However, W_C is limited by W_{Cmax} causing transverse resonance to occur at operation frequencies.
- Feasible dimensions of inductive sections: The highest Z_{High} is, the better the lumped-element inductor approximation is. The highest Z_{High} is, the thinnest the corresponding line width W_L is. The practical implementation of the line section becomes dependent on the smallest feasible line width W_{Lmin}.

The first step of PIM-to-PSM transformation is carried out in "Analysis" stage (i.e., RM/PIM domain). The target platform is known (i.e., distributed lines). Nevertheless, it is possible that there are little information about the implementation technology that are available at this stage. If so, default values may be used and adjusted later.

For this case study, we choose low and high characteristic impedances as following:

$$\begin{cases} Z_{Low} = 10\Omega \\ Z_{High} = 93\Omega \end{cases}$$

2. Step 2: Calculation of resonators normalized values

The normalized element values of the Chebyshev Type I lowpass filter prototype are calculated using the equations A-8, A-9 and A-10 (APPENDIX A, pp. 115-116). The resulting filter is an order 5 lowpass prototype whose normalized element values are listed in the second column of Table 2.22. Using the equations of Table A-3 (p. 120), the electrical length of each section can be computed (see column 2 in Table 2.22).

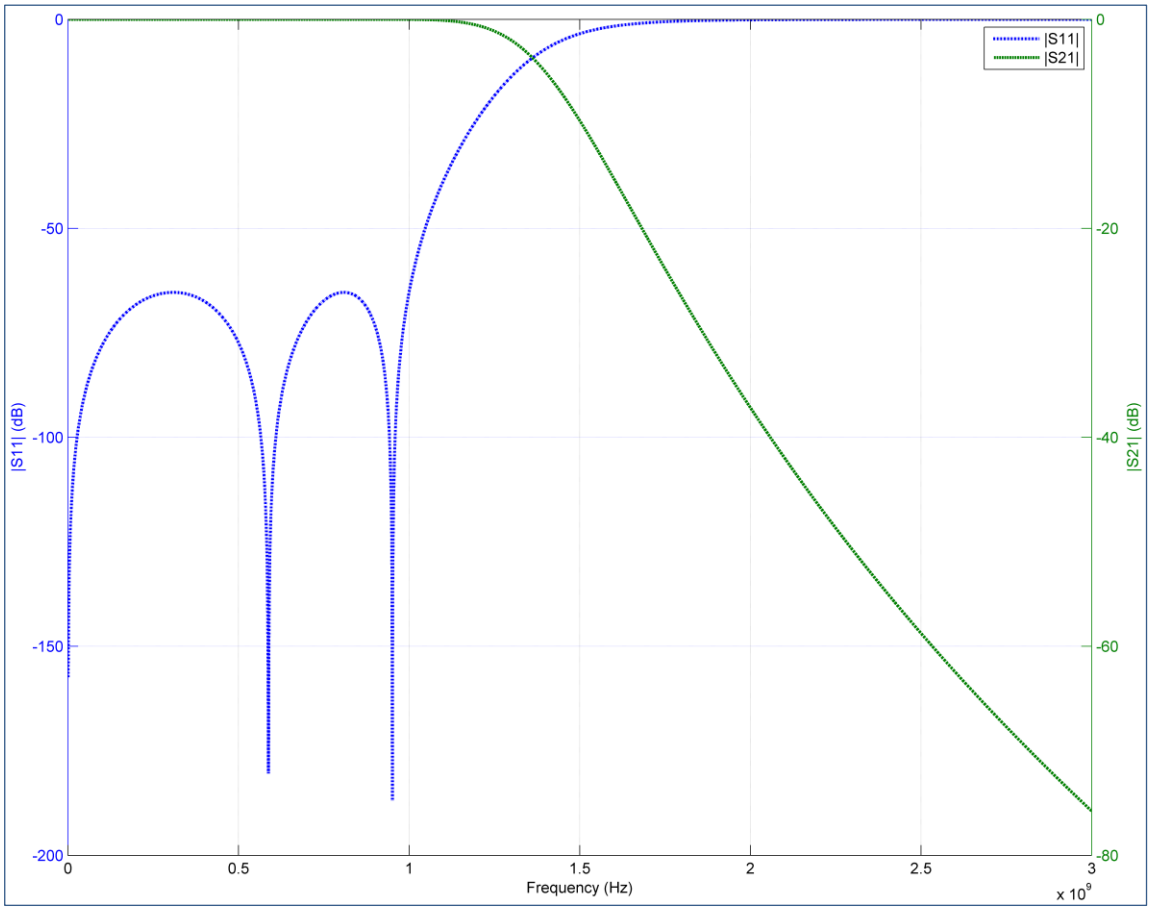


Figure 2.54 Black-box model frequency response: Chebyshev Type I approximation

3. Step 3: Synthesis of transmission lines and circuit topology

At this step, line sections are ideal. For each capacitive (respectively inductive) one, the assigned characteristic impedance is Z_{Low} (Z_{High} respectively). The electrical length is assigned as given in Table 2.22. The cutoff frequency F_c is assigned to the operation frequency. The resulting PSM is depicted in Figure 2.55.

Table 2.20 A PIM to PSM transformation for stepped-impedance lowpass filters

Step 1: Determination of high and low characteristic impedances depending on target distributed lines (see considerations in Hong et Lancaster, 2001, p. 113)

1.1. Specify low characteristic impedance (Z_{Low}) for capacitive sections
1.2. Specify high characteristic impedance (Z_{High}) for inductive sections

Step 2: Calculation of resonators normalized values

2.1. Calculate lowpass prototype (LPP) normalized values (g_k) for Chebyshev Type I response approximation (see equations A-8, A-9 and A-10, pp. 115-116)
2.2. Calculate corresponding electrical lengths (βl_k) (see Table A-3 in APPENDIX A, p. 120)

Step 3: Synthesis of transmission lines and circuit topology

3.1. For each capacitive resonator ($\forall p \leq \frac{N}{2}$):

Consider the properties given in the following:

$$\begin{cases} Z_{ci} = Z_{Low} \\ E_i = \beta l_{2p} \\ F_i = F_c \\ A_i = A_{default} \end{cases}$$

3.2. For each inductive resonator ($\forall p \leq \frac{N+1}{2}$):

Consider the properties given in the following:

$$\begin{cases} Z_{ci} = Z_{Low} \\ E_i = \beta l_{2p+1} \\ F_i = F_c \\ A_i = A_{default} \end{cases}$$

where Z_{ci} is the characteristic impedance of the distributed line, E_i is its electrical length, F_i is the frequency at which the electrical length is calculated and A_i is the transmission line loss (in dB).

2.3.7 Synthesis: Technology mapping

After performance assessment, the obtained PSM goes through PSM granularity refinement if required. Then, the PSM is submitted to technology. In this case study, we choose microstrip transmission lines as implementation technology. We also use Rogers 3006 as a substrate (see its properties in Table 2.23). Having no off-the-shelf components to consider, technology mapping consists in this case study of synthesizing a microstrip line for each filter section. Using a dedicated tool (e.g., LineCalc) and the calculations resulting from the PIM-to-PSM transformation as well as the technology input given in Table 2.23, we determine the physical dimensions (i.e., width and length) of each filter section. The results are reported in the last two columns of Table 2.22 and the resulting filter's PSM is illustrated in Figure 2.56.

Table 2.21 A PIM-to-PSM transformation based on open-end circuit stubs

Step 1: Calculation of resonators normalized values

1.1. Calculate lowpass prototype (LPP) normalized values (g_k) for Chebyshev Type I response approximation (see equations A-8, A-9 and a-10, pp. 115-116)

Step 2: Conversion of capacitive and inductive resonators using Richard's transformation (see Table A-4, p. 120)

2.1. Consider electrical length $\beta l = \frac{\lambda}{8}$) for all filter prototype sections

2.2. Convert shunt capacitive elements (g_{2p}) to shunt stubs

2.3. Convert series inductive elements (g_{2p+1}) to series stubs

2.4. Add unit elements ($Z_c = 1\Omega$)

Step 3: Transformation of the filter sections using Kuroda identities

3.1. Convert iteratively shunt into series stubs (and vice-versa) until getting a practically realizable filter sections (see Figure A-1 in APPENDIX A, p. 121)

3.2. Frequency and impedance scaling of all sections

Step 4: Synthesis of transmission lines and circuit topology

4.1. For each capacitive resonator ($\forall p \leq \frac{N}{2}$):

Consider the properties given in the following:

$$\begin{cases} Z_{ci} = Z_{2p} \\ E_i = \dfrac{\lambda}{8} \\ F_i = F_c \\ A_i = A_{default} \end{cases}$$

4.2. For each inductive resonator ($\forall p \leq \frac{N+1}{2}$):

Consider the properties given in the following:

$$\begin{cases} Z_{ci} = Z_{2p+1} \\ E_i = \dfrac{\lambda}{8} \\ F_i = F_c \\ A_i = A_{default} \end{cases}$$

where Z_{ci} is the characteristic impedance of the distributed line, E_i is its electrical length, F_i is the frequency at which the electrical length is calculated and A_i is the transmission line loss (in dB).

Table 2.22 Calculations resulting from the stepped-impedance-based (PIM-to-PSM) transformation

Step Impedance (Ω)	**Normalized Values (g_i)**	**Electrical Length (°)**	**Physical Dimensions (mil)**	
			Width	**Length**
Z_{high} = 93.0	1.146837827800645	13.141793	8.117520	218.300394
Z_{low} = 10.0	1.371209987510143	42.239003	324.938976	592.850394
Z_{high} = 93.0	1.975027577855714	22.632149	8.117559	375.946457
Z_{low} = 10.0	1.371209987510143	42.239003	324.938976	592.850394
Z_{high} = 93.0	1.146837827800645	13.141793	8.117520	218.300394

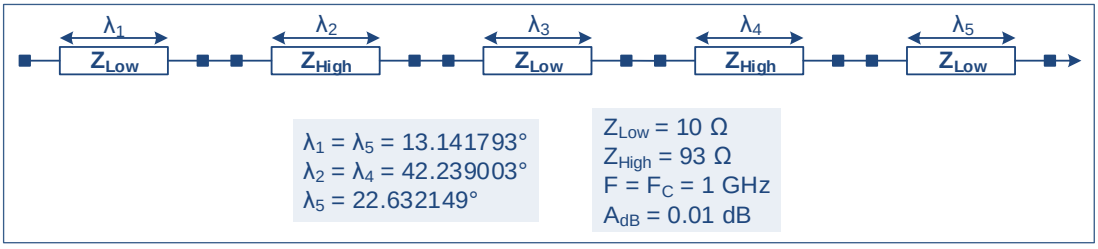
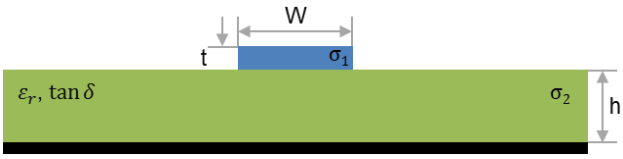


Figure 2.55 First PSM: The stepped-impedance lowpass filter

If the default platform considerations were used in the first step of the PIM-to-PSM transformation (i.e., low and high characteristic impedances), it is possible to adjust values in the technology mapping step and make the PSM more accurate.

Table 2.23 Main properties of the RO3006 substrate

Dielectric Constant (ε_r)	6.15
Magnetic Constant (μ_r)	1
Loss Tangent ($\tan\delta$)	0.0025
Resistivity (σ_1)	1
Metal Thickness (t)	0.71 mil
Metal Roughness (Sr)	0.075 mil
Substrate Height (h)	25 mil



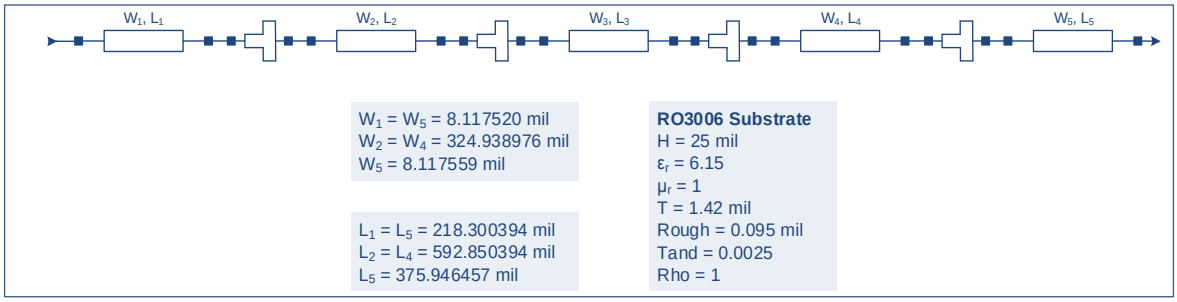


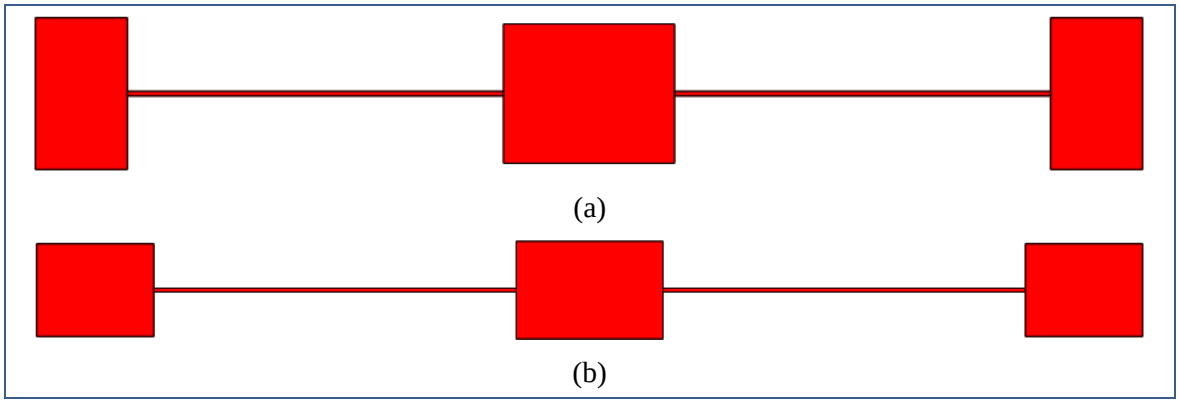
Figure 2.56 First PSM after technology mapping

Table 2.24 Lowpass stepped-impedance filter PM form factor

PM	Width (mm)	Length (mm)	Average Area (mm^2)
Transformation output	54.9783	6.84878	376.53427
Optimization (round 1)	58.16422	5.08	295.47425
Optimization (round 2)	42.45915	22.86	970.61613

2.3.8 Synthesis: PM generation

After eventual PSM optimization (including through granularity refinement) and performance assessment, the final PSM is transformed into a platform model. For this purpose, we use a commercial design package to derive the filter PM (i.e., layout) directly from the PSM. Using Keysight ADS, we obtain the platform model illustrated in Figure 2.57.a. Then, the physical structure of the filter is simulated using an EM solver to evaluate its performance. We carried out several optimization iterations for the purpose of enhancing the frequency response of the obtained PM. In each iteration, the physical properties of the PM (such as the length and width of the microstrip sections) were altered. Figure 2.57.b and 2.57.c depict the best optimizations of the obtained PM. The form factor of each PM was also estimated in the Table 2.24. In addition, Figure 2.58 illustrates the evolution of the lowpass filter frequency response throughout the design cycle. In addition to the black-box model (PIM level) and layout (PM level) performance, the PSM frequency response after PIM-to-PSM transformation, technology and optimization is also illustrated.



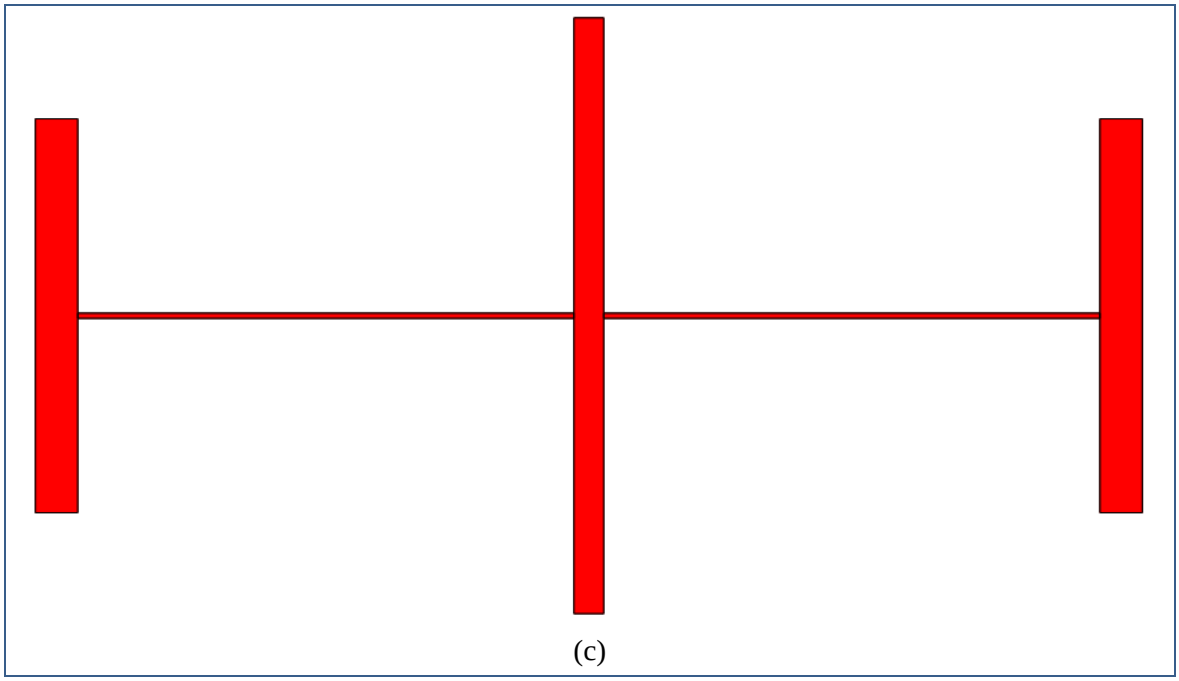


Figure 2.57 Lowpass filter stepped-impedance platform model: (a) transformed, (b) first- and (c) second-round optimizations

b) Open-end stub-based transformation

- **Analysis: PSM generation**

The second transformation generates open-end stubs sections for the lowpass filter. It consists of four steps:

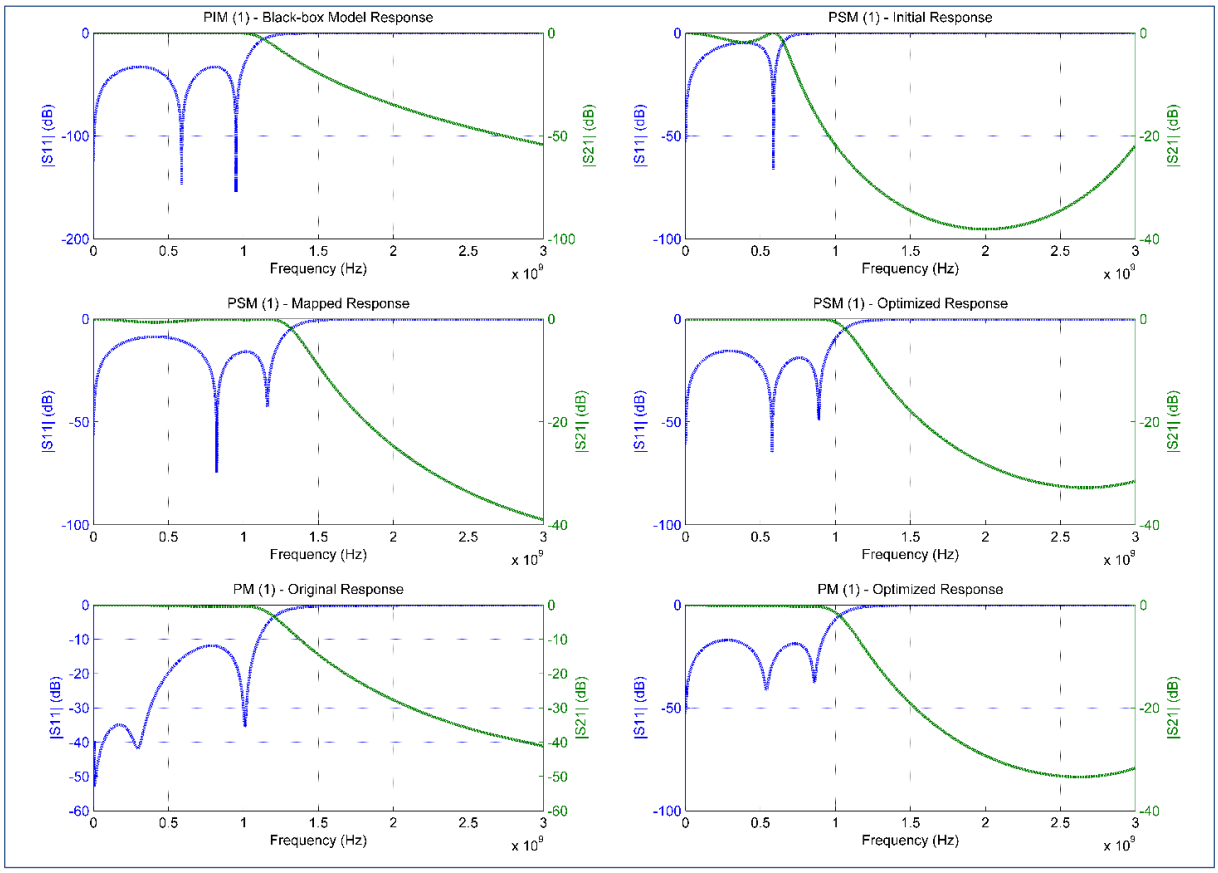


Figure 2.58 Frequency response of the stepped-impedance lowpass prototype in key design steps

1. Step 1: Calculation of resonators normalized values

In the first step, we use the equations A-8, a-9 and A-10 (in APPENDIX A, pp. 115-116) to calculate the normalized element values of the lowpass filter prototype (see column one in Table 2.25).

2. Step 2: Conversion of capacitive and inductive resonators using Richard's transformation (see Table A-4, p. 120)

The prototype normalized element values correspond to the capacitive and inductive filter components. Using the Richard's transformation given in Table A-4 (APPENDIX A, p. 120), the series inductors (respectively shunt capacitors) are converted to series subs (respectively shunt stubs).

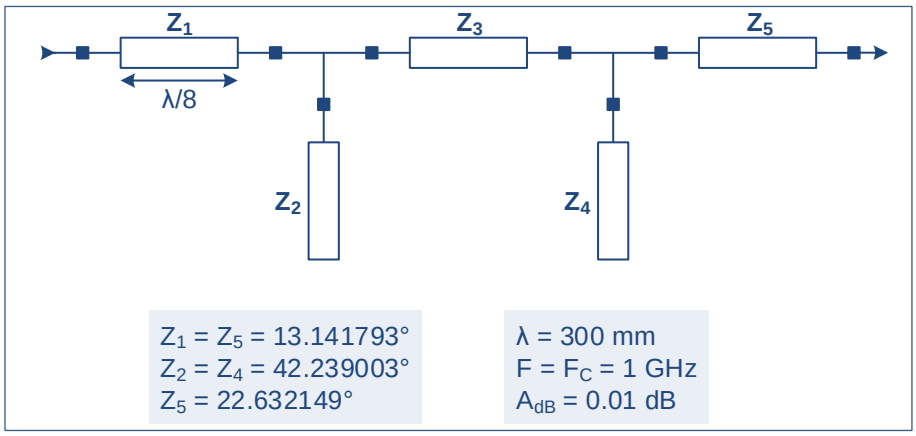


Figure 2.59 Second PSM: The open-end stub lowpass filter

3. Step 3: Transformation of the filter sections using Kuroda identities

Since the series stubs are very difficult to realize in practice, we should convert them into feasible structures. We use Kuroda identities given in Figure A-1 (p. 121) to iteratively derive new transmission line sections from these stubs. We get the normalized characteristic impedances listed in the second column of Table 2.25. After frequency and impedance scaling, we get the true characteristic impedance of each stub (see third column of Table 2.25).

4. Step 4: Synthesis of transmission lines and circuit topology

In the last step, we create the PSM circuit topology. We assign an operation frequency, an electrical length and a characteristic impedance to each line section. The resulting PSM is illustrated in Figure 2.59.

2.3.9 Synthesis: Technology mapping

After performance assessment and the eventual granularity refinement, the technology mapping is carried out. The resulting PSM (composed so far of ideal elements) is enriched with the appropriate technology information. In this case, the technology input to consider is the type of technology (i.e., microstrip) and associated substrate (i.e., RO3006). Using a commercial line synthesis tool (i.e., LineCalc), we synthesized the

physical dimensions that fit with the characteristic impedance and electrical length at the operation frequency (i.e., $\beta l = \frac{\lambda}{8} = 22.5°$). The length and width of each line section is given in the last two columns of Table 2.25.

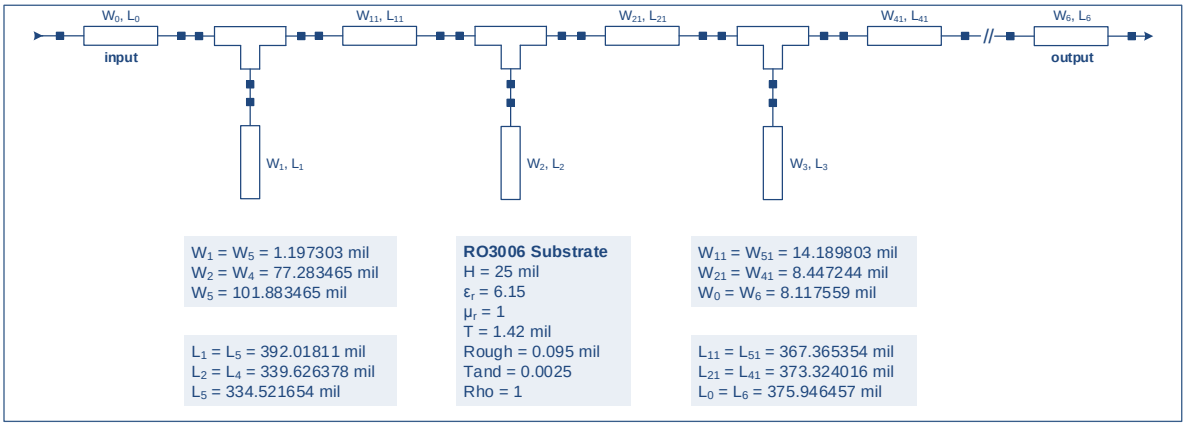


Figure 2.60 Second PSM after technology mapping

Table 2.25 Calculations resulting from the open-end stub-based transformation

Normalized Values (g_i)	**Normalized Line Impedance (Ω)**	**Line Impedance (Ω)**	**Physical Dimensions (mil)***	
			Width	**Length**
1.146837827800645	2.864	143.2	1.197303	392.01811
	1.5342	76.71	14.189803	367.365354
1.371209987510143	0.624	31.2	77.283465	339.626378
	1.837	91.85	8.447244	373.324016
1.975027577855714	0.5128	25.64	101.883465	334.521654
1.371209987510143	1.837	91.85	8.447244	373.324016
	0.624	31.2	77.283465	339.626378
1.146837827800645	1.5342	76.71	14.189803	367.365354
	2.864	143.2	1.197303	392.01811

(*) Associated Electrical Length: 22.5° ($\lambda/8$)

2.3.10 Synthesis: PM generation

To derive the final layout of the open-end stub-based lowpass filter, we use a PSM-to-PM transformation that consists of a commercial design package. Using Keysight ADS, we generate the filter's layout depicted in Figure 2.61.a. After many optimization iterations, we obtained the final layout illustrated in Figure 2.61.b. Regarding the form factor requirement, Table 2.26 gives an estimation of the final dimensions of both the original and optimized layouts.

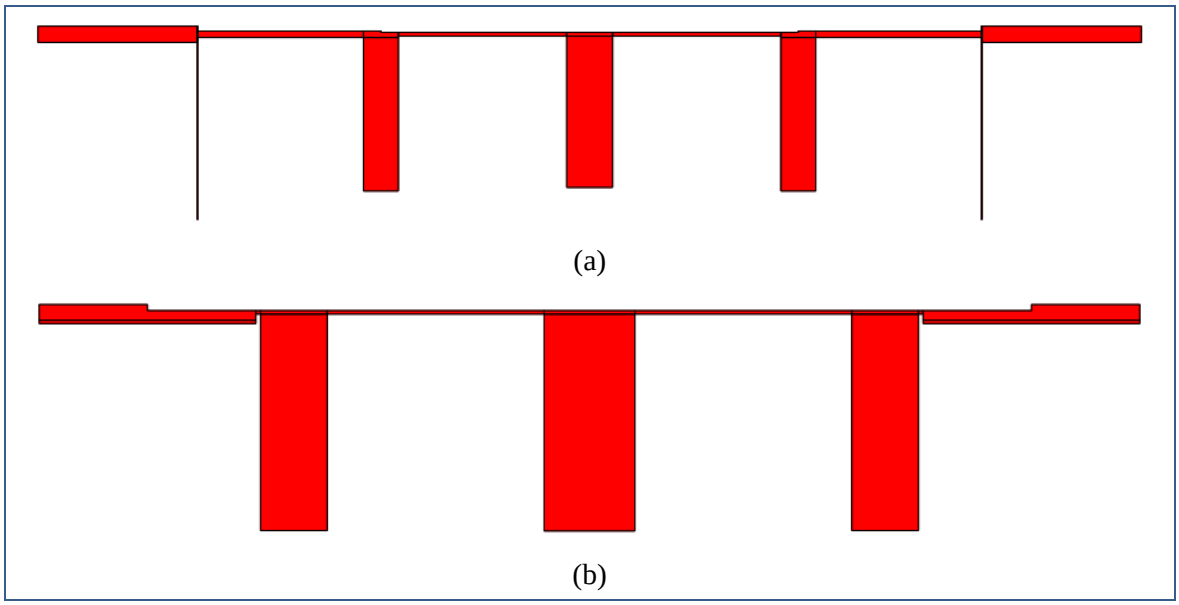


Figure 2.61 Lowpass filter open-end stub-based platform model: (a) transformed, (b) best optimization round

Table 2.26 Lowpass open-end stub-based filter PM form factor

PM	Width (mm)	Length (mm)	Average Area (mm^2)
Transformation output	13.335	56.5404	753.966234
Best optimization	13.208	64.4906	851.7918448

The lowpass filter was submitted to several rounds of performance assessment and optimization from the RM/PIM through PM domains. Figure 2.62 illustrates the filter's

frequency response at the PIM level (i.e., the black-box model). In the PSM domain, the frequency response of the filter are illustrated after PIM-to-PSM transformation, technology mapping and optimization. Then, it is illustrated after PSM-to-PM transformation and best optimization iteration.

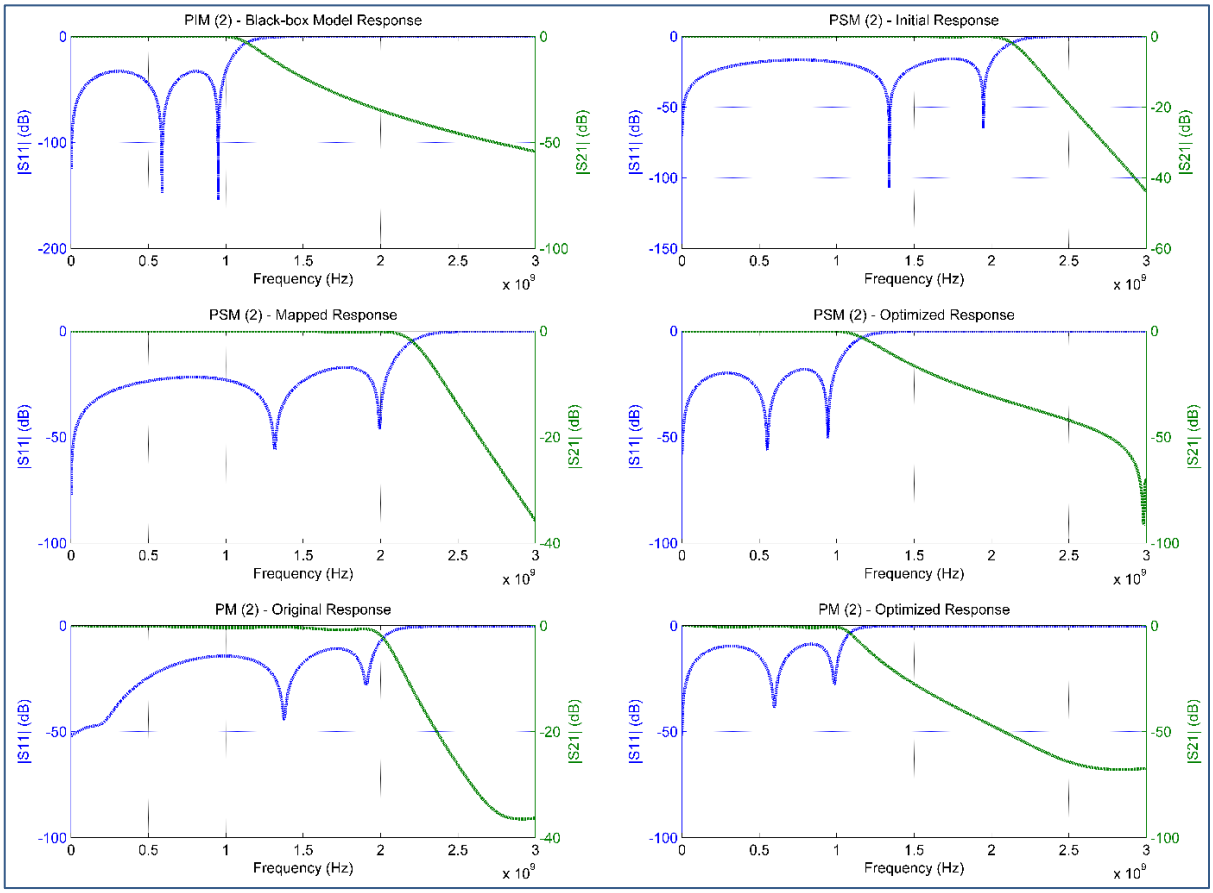


Figure 2.62 Frequency response of the open-end stubs lowpass prototype in key design steps

Finally, the Q-matrix of Figure 2.63 depicts the design data stored in each design step for both techniques (i.e., stepped-impedance lines, open-end stubs). Each Qblock includes the data for a design iteration. Having all this data in the same file, it becomes easier to make performance comparisons between design techniques. For instance, Figure 2.64 shows a comparison between the stepped-impedance and open-end stub-

based filter prototypes. The frequency response (particularly within the transition band) of the latter is better than the former. This result is expected because the coarse approximations involved in the stepped-impedance filter degrade the overall performance (Pozar, 2012, p. 470).

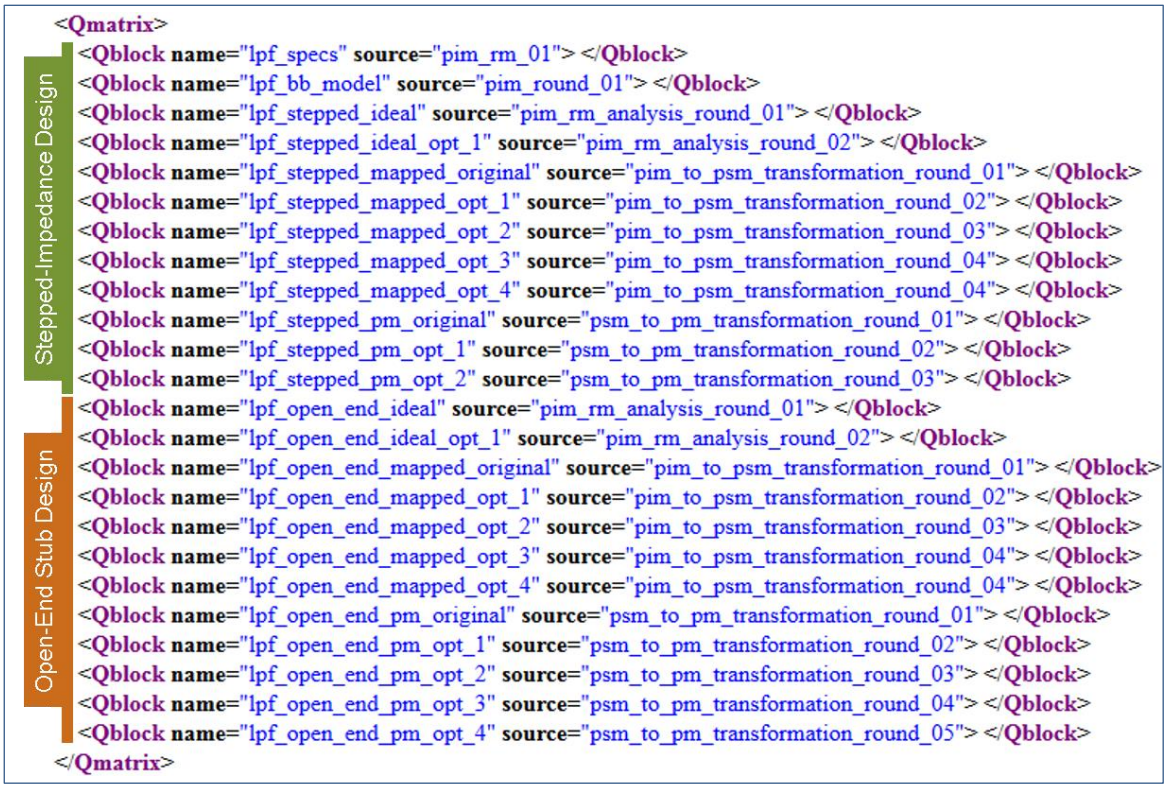
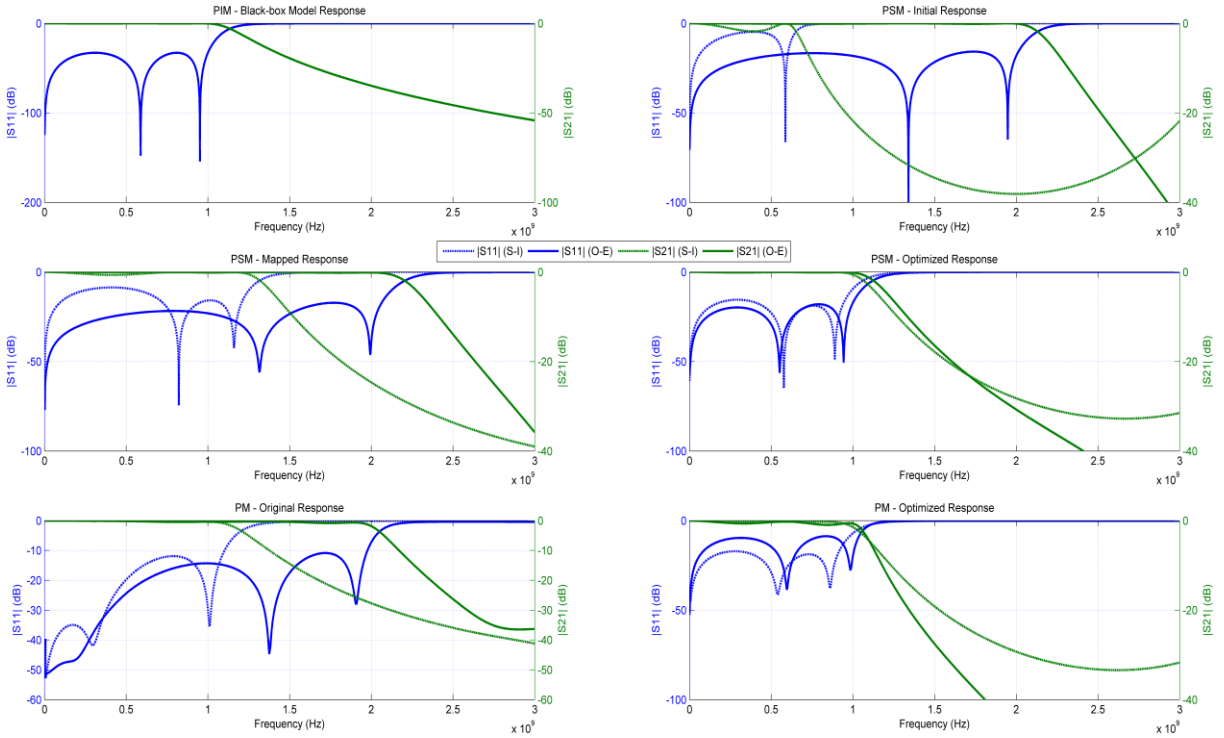


Figure 2.63 Overview of the Qblocks added to the filter's Q-matrix through the design cycle



S-I: Stepped-impedance

O-E: Open-end

Figure 2.64 Comparison between the performance of the stepped-impedance and open-end stub-based filter prototypes in different design stages

REFERENCES

Hong, Jia-Sheng, and M.J. Lancaster. 2001. *Microstrip Filters for RF/Microwave Applications*. Hoboken, New Jersey: John Wiley and Sons.

Lafi, Sabeur, Ammar Kouki, and Jean Belzile. 2016. *SysML Profile for RF Devices*, Technical report, École de technologie supérieure.

Lafi, Sabeur, Ammar Kouki and Jean Belzile. 2015. "Enhancing Automation in RF Design Using Hardware Abstraction". In *Computational Intelligence in Analog and Mixed-Signal (AMS) and Radio-Frequency (RF) Circuit Design*, Mourad Fakhfakh, Esteban Tlelo-Cuautle, and Patrick Siarry (eds), pp. 439-469. Cham: Springer International Publishing.

Pozar, David M. 2012. *Microwave Engineering*, 4th ed. Hoboken, New Jersey: John Wiley and Sons.

Wanhammar, Lars. 2009. *Analog Filters Using MATLAB*. Springer US.

APPENDIXES

APPENDIX A

SUMMARY OF EQUATIONS AND FORMULAS USED IN CASE STUDIES

A.1 Case Study 1: Frequency Selection Device

A.1.1 General Definitions

- General linear analog filter transfer function:

$$H(s)|_{s=\sigma+j\omega} = \frac{E(s)}{P(s)} = \frac{\sum_{i=0}^{m} a_i\, s^i}{\sum_{j=0}^{n} b_j\, s^j} \tag{A-1}$$

where a_i and b_j are real coefficients.

- Analog filter phase shift:

$$\phi(\omega) = arg\big(H(j\omega)\big) \tag{A-2}$$

where H is the filter's transfer function and arg is the complex argument function.

- Analog filter phase delay:

$$\tau_\phi(\omega) = -\frac{\phi(\omega)}{\omega} \tag{A-3}$$

where ϕ is the filter's phase shift and ω is the angular frequency.

- Analog filter group delay:

$$\tau_g(\omega) = -\frac{d\phi(\omega)}{d\omega} \tag{A-4}$$

where ϕ is the filter's phase shift and ω is the angular frequency.

A.1.2 Examples of Filter Approximations

A.1.2.1 Butterworth Filter Approximation

- Butterworth model transfer function:

$$G_T = |H(j\omega)|^2 = \frac{H_0}{1 + \left(\frac{\omega}{\omega_s}\right)^{2N}} \tag{A-5}$$

where ω_s is the stopband edge angular frequency and N is the filter order.

- Butterworth filter order:

$$N = \frac{\log\left(\frac{10^{\frac{A_p}{10}} - 1}{10^{\frac{A_s}{10}} - 1}\right)}{2\log\left(\frac{\omega_s}{\omega_c}\right)} \tag{A-6}$$

where A_p is the maximum attenuation (in dB) in the passband ($0 \leq \omega \leq \omega_c$), A_s is the minimum attenuation (in dB) in the stopband ($\omega_s \leq \omega \leq +\infty$), ω_c is the passband edge angular frequency and ω_s is the stopband edge angular frequency.

- Butterworth lowpass prototype normalized elements:

$$\begin{cases} g_0 = g_{N+1} = 1 \\ g_k = 2\sin\left(\frac{(2k-1)\pi}{2N}\right), k = 1,2,\dots,N \end{cases} \tag{A-7}$$

where N is the filter order.

A.1.2.2 Chebyshev Type I Filter Approximation

- Chebyshev Type I model transfer function:

$$G_T = |H(j\omega)|^2 = \frac{H_0}{1 + \varepsilon^2 T_n^2\left(\frac{\omega}{\omega_s}\right)} \tag{A-8}$$

where ω_s is the stopband edge angular frequency, ε is a measure of the passband ripple and $T_n(x)$ is the Chebyshev function defined as follows:

$$\begin{cases} T_n(x) = \cos[n \cdot \cos^{-1}(x)], & 0 \leq x \leq 1 \\ T_n(x) = \cosh[n \cdot \operatorname{arccosh}(x)], & x > 1 \end{cases} \tag{A-9}$$

- Chebyshev Type I filter order:

$$N = \frac{\operatorname{arccosh}\left(\frac{1}{\varepsilon}\left(10^{\frac{A_p}{10}} - 1\right)^{-1/2}\right)}{\operatorname{arccosh}\left(\frac{\omega_s}{\omega_c}\right)} \tag{A-10}$$

where A_p is the maximum attenuation (in dB) in the passband ($0 \leq \omega \leq \omega_c$), ε is a measure of the passband ripple (in dB), ω_c is the passband edge angular frequency and ω_s is the stopband edge angular frequency.

- Chebyshev Type I lowpass prototype normalized elements:

$$\begin{gathered} g_0 = 1 \\ g_{N+1} = \begin{cases} 1, & N\ odd \\ \tanh^2\left(\frac{\beta}{4}\right), & N\ even \end{cases} \\ g_1 = 2\frac{a_1}{\gamma} \\ g_k = \frac{4a_{k-1}a_k}{b_{k-1}g_{k-1}}, \qquad k = 2,3,\dots,N \\ a_k = \sin\left(\frac{(2k-1)\pi}{2N}\right), \qquad k = 1,2,\dots,N \\ b_k = \gamma^2 + \sin^2\left(\frac{k\pi}{N}\right), \qquad k = 1,2,\dots,N \\ \beta = \ln\left[\coth\left(\frac{A_m}{17.32}\right)\right] \\ A_m = 10\log(\varepsilon^2 + 1) \\ \gamma = \sinh\left(\frac{\beta}{2N}\right) \end{gathered} \tag{A-11}$$

where N is the filter order.

A.1.2.3 Resonators selection

Given that N is the filter order, then:

- if is N even:

$$\begin{cases} X_i = g_{2p}, & p \in [0,1,..,\frac{N}{2}] \\ X_j = g_{2p+1}, & p \in [0,1,..,\frac{N-2}{2}] \end{cases} \quad \text{(A-12)}$$

where X_i and X_j are respectively series and parallel resonators.

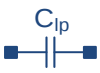
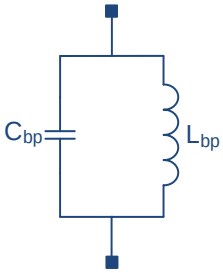
- if is N odd:

$$\begin{cases} X_i = g_{2p}, & p \in [0,1,..,\frac{N-1}{2}] \\ X_j = g_{2p+1}, & p \in [0,1,..,\frac{N-1}{2}] \end{cases} \quad \text{(A-13)}$$

where X_i and X_j are respectively series and parallel resonators.

Given ω_0 and BW the center frequency and the bandwidth of the bandpass filter and considering that $\Delta = \frac{BW}{\omega_0}$, the bandpass lumped elements can be derived from the lowpass prototype ones as follows :

Table A-1 Lowpass to bandpass lumped-element transformations
Adapted from Pozar (2012, pp. 414)

Lowpass Prototype Element	**Bandpass Elements**	**Transformation Formulas**
		$\begin{cases} L_{bp} = \frac{L_{lp}}{\Delta\omega_0} \\ C_{bp} = \frac{\Delta}{\omega_0 L_{lp}} \end{cases}$
		$\begin{cases} L_{bp} = \frac{\Delta}{\omega_0 C_{lp}} \\ C_{bp} = \frac{C_{lp}}{\Delta\omega_0} \end{cases}$

Lowpass Prototype Element	Bandpass Elements	Transformation Formulas
 R_{lp}	 R_{bp}	$R_{bp} = R_{lp}$

A.1.2.4 Impedance Scaling

- Impedance scaling of lowpass prototype elements:

Given that N is the filter order, then:

$$\begin{cases} L = R_L L_p, & p \in [0,1,..,N] \\ C = \dfrac{C_p}{R_L}, & p \in [0,1,..,N] \\ R = R_L R_p \end{cases} \tag{A-14}$$

where C_p and L_p are respectively the lowpass prototype capacitance and inductance values, R_p is the LPP output impedance and R_L is the load impedance.

A.1.2.5 Frequency Scaling

- Frequency scaling of lowpass prototype elements:

$$\begin{cases} L' = \dfrac{L}{\omega_c} \\ C' = \dfrac{C}{\omega_c} \\ R' = R \end{cases} \tag{A-15}$$

where ω_c is the angular cutoff frequency.

A.1.2.5 Frequency Scaling

- Lowpass to bandpass frequency transformation:

$$\omega' = \frac{\omega_0}{\Delta\omega}\left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega}\right) \tag{A-16}$$

where ω_0 is the angular bandpass filter center frequency and $\Delta\omega$ is its angular bandwidth.

A.2 Case Study 2: Transmission Line Lowpass Filter

A.2.1 Transmission Line Electrical Models

Table A-2 High- and low-impedance transmission line approximations

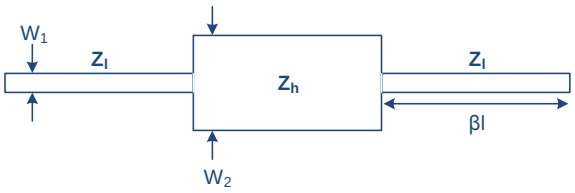
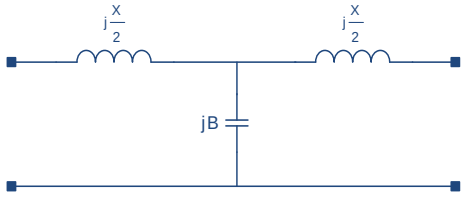
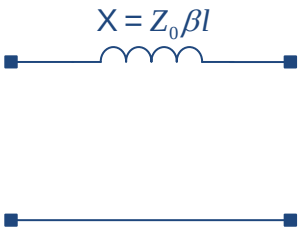
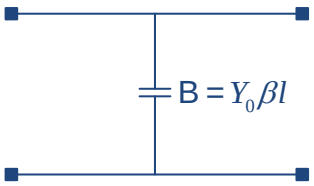
<table>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>Transmission line</td><td>T-equivalent circuit for a transmission line section having $\beta l \ll \frac{\pi}{2}$</td></tr>
<tr><td> 
</td><td>if $\beta l < \frac{\pi}{2}$:
$\frac{X}{2} = Z_0 \tan\left(\frac{\beta l}{2}\right)$
if $\beta l < \frac{\pi}{4}$:
$X \cong Z_0 \beta l$</td></tr>
<tr><td colspan="2">Equivalent circuit for small βl and large characteristic impedance Z_0</td></tr>
<tr><td> 
</td><td>if $\beta l < \frac{\pi}{2}$:
$B = \frac{1}{Z_0}\sin(\beta l)$
if $\beta l < \frac{\pi}{4}$:
$B \cong 0$</td></tr>
<tr><td colspan="2">Equivalent circuit for small βl and small characteristic impedance Z_0</td></tr>
</table>

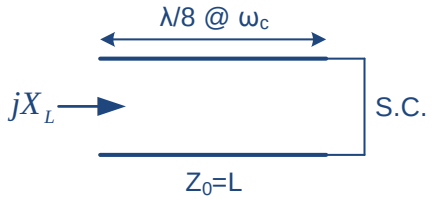
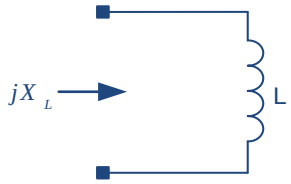
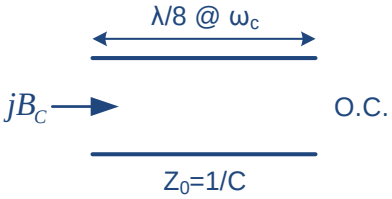
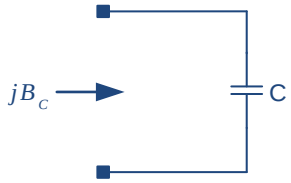
Table A-3 Equivalent LC components for high- and low-impedance transmission lines

$Z_0 = Z_{High}$	$Z_0 = Z_{Low}$
if $\beta l < \frac{\pi}{4}$ and $\frac{Z_{High}}{Z_{Low}} \gg 1$ at $\omega = \omega_c$: $$\beta l = \frac{LR_0}{Z_{High}}$$	if $\beta l < \frac{\pi}{4}$ and $\frac{Z_{High}}{Z_{Low}} \gg 1$ at $\omega = \omega_c$: $$\beta l = \frac{CZ_{Low}}{R_0}$$
Series inductor: $$L = \frac{Z_{High}\beta l}{R_0}$$	Shunt capacitor: $$C = \frac{R_0\beta l}{Z_{Low}}$$

A.2.1 Stub Electrical Models

a- Richard's Transformation

Table A-4 Richard's transformations for shunt- and open-circuit stubs

Stub Configuration	Electrical Model	Mathematical Model
		$jX_L = jL\tan(\beta l)$
		$jB_C = jC\tan(\beta l)$

S.C.: Shunt circuit

O.C.: Open circuit

b- Kuroda Identities

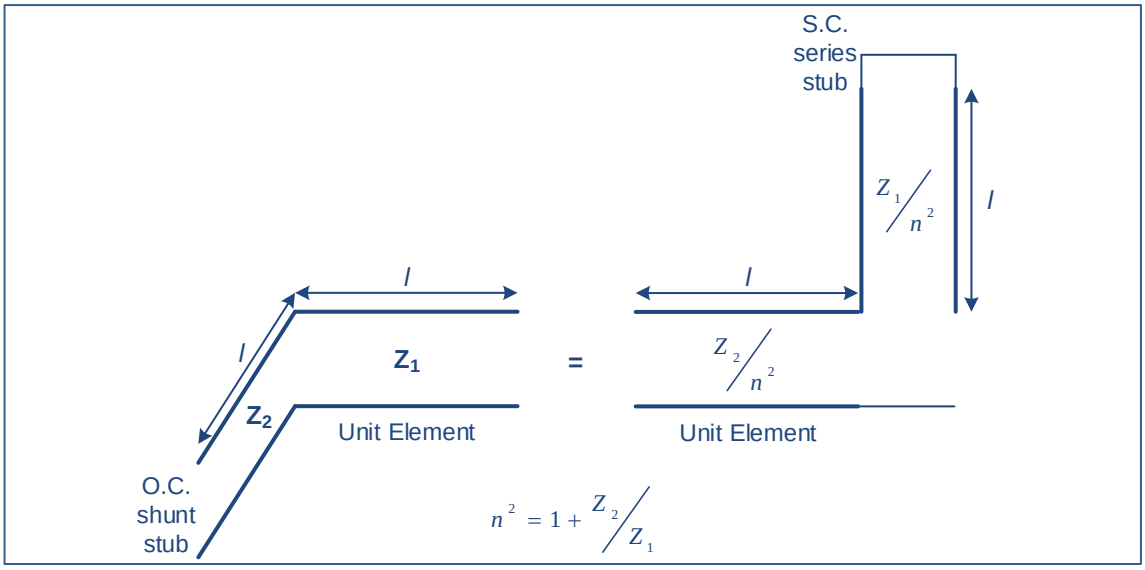


Figure A-1 Transformation of transmission lines using Kuroda identities