NANO, MESO, MICRO : SCIENCES ET INNOVATIONS POUR LA RADIO ET LA PHOTONIQUE.

scikit-rf: une librairie open-source en Python pour la simulation, l'analyse et la calibration de dispositifs micro-ondes

J.Hillairet[1], A.Arsenovic[2], J.Anderson[3], H.Forstén[4], V.Rieß[5], W.Barnhart[6], F.Forstmayr[7] and GitHub contributors[8]

[1] *CEA, IRFM, F-13108, St-Paul-Lez-Durance, France*
[2] *810 Labs LLC, Stanardsville, Virginia, 22973, USA.*
[3] *Purdue University, West Lafayette, Indiana, 47907, USA*
[4] *VTT Technical Research Centre of Finland, Espoo, 02044, Finland*
[5] *AES GmbH, Bremen, 28199, Germany*
[6] *HawkEye 360, Herndon, VA, 20170, USA.*
[7] *Rosenberger Hochfrequenztechnik GmbH &Co. KG, Fridolfing, 83413, Germany*
[8] *https://github.com/scikit-rf/scikit-rf/graphs/contributors*

Python, open-source, modélisation RF, calibration

Résumé

scikit-rf est une librairie Python gratuite et open-source conçue pour simplifier et fiabiliser l'ingénierie RF/micro-ondes. C'est une boite à outils moderne pour l'analyse, la simulation de circuits passifs et la calibration de dispositifs RF. La librairie facilite l'analyse de dispositifs RF avec la lecture/écriture de fichiers Touchstone (fichiers .sNp), de fichiers GMDIF ou CITI, ou l'extraction et la représentation graphique des données (dB, amplitude, phase, abaques de Smith, représentations d'incertitudes, etc.). Les paramètres S des circuits peuvent être facilement transformés (en paramètres Z, Y, ABCD, etc.) ou manipulés : sélection d'intervalles de fréquences, sous-circuits, concaténations, interpolations ou transformations en représentation temporelle. La création de circuits équivalents pour SPICE est possible grâce aux méthodes d'ajustement vectoriel (vector fitting). De nombreuses méthodes de calibration (SOLT, SDDL, TRL, LRM, etc.) sont incluses pour la correction post-mesure ou en connexion directe avec un VNA. Lorsque des standards de calibrations ne sont pas disponibles, par exemple pour des dispositifs à très hautes fréquences au-delà de 100 GHz, des méthodes avancées de deembedding peuvent être employées.

Abstract

scikit-rf is a free and open-source Python package designed to make RF/Microwave engineering both robust and approachable. The package provides a modern library for RF network analysis, circuit building, calibration, and simulation. Besides offering standard microwave network operations, such as reading/writing Touchstone files (.sNp files), GMDIF or CITI files, connecting or deembedding N-port networks, frequency/port slicing, concatenation or interpolations, it is also capable of advanced operations such as Vector Network Analyzer (VNA) offline calibrations, advanced deembedding, time-gating, vector fitting, interpolating between an individual set of networks, deriving network statistical properties and support of Virtual Instruments for direct communication to VNAs. The package also allows straightforward plotting of rectangular plots (dB, magnitude, phase, group delay, etc.), Smith Charts or automated uncertainty bounds.

1 Introduction

La prolifération rapide des applications de télécommunication et de radiofréquence (RF) nécessite des outils efficaces et pratiques pour concevoir et caractériser ces dispositifs. scikit-rf [1], [2] est une librairie Python gratuite et open-source conçue pour rendre l'ingénierie RF/micro-ondes à la fois fiable et accessible. Créée en 2009 et continuellement améliorée depuis, la librairie est distribuée sous licence open-source (BSD) et activement développée par plus de 50 volontaires sur la plateforme GitHub. scikit-rf est utilisée dans de nombreuses universités et instituts de recherche dans le monde ainsi que par des industriels (Keysight, Rohde & Schwarz, National Instruments, Nvidia, 3M, etc). En 2022, la librairie a été téléchargée plus de 280 000 fois depuis son ouverture et a été utilisée dans plus d'une trentaine de publications [3].

La librairie étant écrite en Python, elle est naturellement compatible avec l'ensemble de l'écosystème scientifique gratuit de ce langage. Parce que le code est open-source, ses utilisateurs peuvent voir exactement ce que le code fait et, si une fonctionnalité doit être améliorée ou n'existe pas, ils peuvent y contribuer pour la corriger ou l'enrichir. Et bien sûr, elle est gratuite.

scikit-rf facilite l'analyse de dispositifs RF avec la lecture/écriture de fichiers Touchstone (fichiers .sNp), GMDIF/CITI, ou l'extraction et la représentation graphique des données (dB, amplitude, phase, abaques de Smith ou représentations d'incertitudes). Les paramètres S des circuits peuvent être facilement transformés (paramètres Z, Y, ABCD, etc.) ou manipulés : sélection d'intervalles de fréquences, de sous-circuits, concaténation ou interpolations fréquentielles ou entre un ensemble de mesures.

Si les analyseurs de réseaux permettent de réaliser des mesures fréquentielles, ils peuvent également par transformée de Fourier inverse calculer les réponses impulsionnelles et indicielles [4]. scikit-rf permet également de réaliser ces transformations, permettant ainsi de filtrer des réflexions ou des éléments parasites [5], [6], de réduire le bruit de mesure ou encore les effets de dispersions [7]. De plus, les méthodes d'ajustement vectoriel (*vector fitting*) [8] incluses dans scikit-rf permettent de générer des circuits équivalents pour SPICE.

scikit-rf permet de concevoir des circuits passifs complexes élaborés à partir de sous-circuits, en connectant ensemble des circuits multi-ports, ainsi que de réaliser des opérations de *deembedding* de réseaux N-port. Différentes méthodes de calibration [9] sont disponibles pour la correction *post*-mesure ou en connexion directe avec un VNA. Lorsque des standards de calibrations ne sont pas disponibles, par exemple pour des dispositifs à très haute fréquences (au-delà de 100 GHz), plusieurs méthodes de *deembedding* sont disponibles pour corriger les mesures [10], [11].

Dans cette contribution, les bases et quelques-unes des fonctionnalités de la librairie scikit-rf sont présentées. Dans la section 2, nous décrivons comment créer un dispositif RF N-port et en obtenir les caractéristiques principales. Dans la section 3, des circuits RF sont combinés entre eux pour créer de nouveaux circuits. Les transformations dans le domaine temporel, souvent utilisées pour les dispositifs à très hautes-fréquences, sont introduites dans la section 4 avec un exemple. La section 5 aborde les outils statistiques pour traiter de la quantification des erreurs de mesure et leur reproductibilité. Enfin, la section 6 discute des méthodes de calibration *offline* et *deembedding*.

2 `Network` : la brique de base de scikit-rf

Un dispositif hyperfréquence à N ports (Figure 1), décrit par ses paramètres S, est représenté dans scikit-rf par un objet appelé `Network`. Un tel objet peut être créé en important un fichier Touchstone (.sNp), un format *de-facto* standard utilisé par les appareils de mesure et les logiciels de simulation RF pour exporter des paramètres réseaux en fonction de la fréquence.

```
>>> import skrf # documentation : http://scikit-rf.org/
>>> ring_slot = skrf.Network('data/ring slot.s2p')
```

Il est également possible de définir un `Network` directement à partir de ses paramètres réseaux (S ou Z/Y/ABCD/T). La description courte des propriétés de l'objet ainsi créé donne le nombre de port(s), les fréquences et le nombre de points pour lesquelles les paramètres S sont définis ainsi que les impédances caractéristiques de ses ports (qui peuvent être fonction de la fréquence) :

```
>>> print(ring_slot)
2-Port Network: 'ring slot',  75.0-110.0 GHz, 201 pts, z0=[50.+0.j 50.+0.j]
```

Les caractéristiques du dispositif peuvent être directement obtenues à partir des propriétés et des méthodes disponibles, par exemple pour les plus utilisées :

```
>>> ring_slot.frequency  # objet décrivant la gamme de fréquences utilisée
>>> ring_slot.z0  # impédance caractéristiques des ports
>>> ring_slot.s  # paramètres S, matrice de dimension (nb_f, nb_port, nb_port)
>>> ring_slot.z  # paramètres Z. Aussi disponible: .y, .a (ABCD), .t (transfer), .h (hybride)
```

L'ensemble des paramètres réseaux sont disponibles selon plusieurs projections possibles : valeurs complexes, réelles ou imaginaires, amplitudes en valeur absolue ou en dB, phases en radian ou degré, etc :

```
>>> ring_slot.s_mag  # magnitude des paramètres S. Aussi : .s_deg (phase en degré), .s_db ...
```

Tous ces paramètres peuvent être obtenus pour des sous-bandes de fréquences, définies à partir des indices des points associés aux fréquences ou bien de façon « naturelle » :

```
>>> ring_slot[100:200]
2-Port Network: 'ring slot_subset',  92.5-109.825 GHz, 100 pts, z0=[50.+0.j 50.+0.j]
>>> ring_slot['80-100GHz']
2-Port Network: 'ring slot',  80.075-100.025 GHz, 115 pts, z0=[50.+0.j 50.+0.j]
```

Des méthodes associées à l'objet Network permettent de représenter graphiquement rapidement et facilement les paramètres réseaux :

```
>>> ring_slot.plot_s_db()  # -> Figure 2. Egalement : .plot_s_smith(), .plot_s_vswr(), ...
```

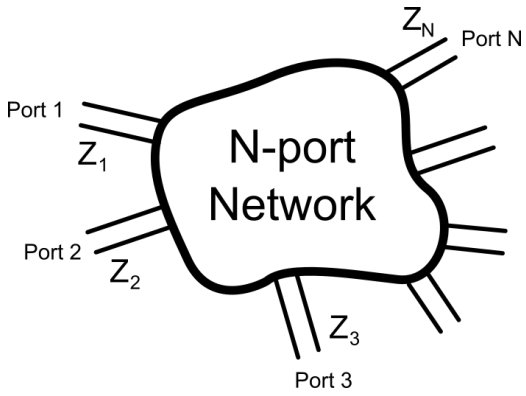

Figure 1. Illustration d'un dispositif RF à N ports.

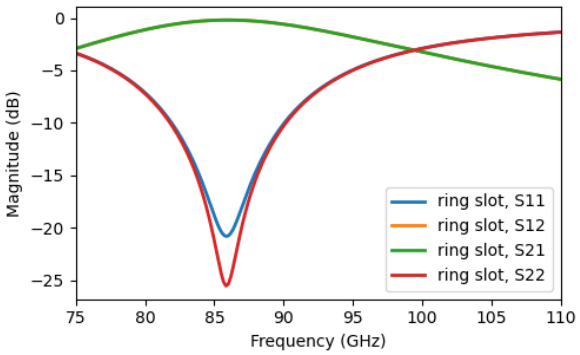

Figure 2. Paramètres S de l'exemple.

3 Manipulation de circuits RF

scikit-rf permet de connecter entre eux des circuits N-ports. La mise en cascade de deux `Networks` s'effectue grâce à l'opérateur « `**` ». La Figure 3 illustre quelques situations classiques : a) cascade d'un 2-port avec un 1-port ; b) cascade de deux 2-port entre eux ; c) cascade de deux 2N-port entre eux.

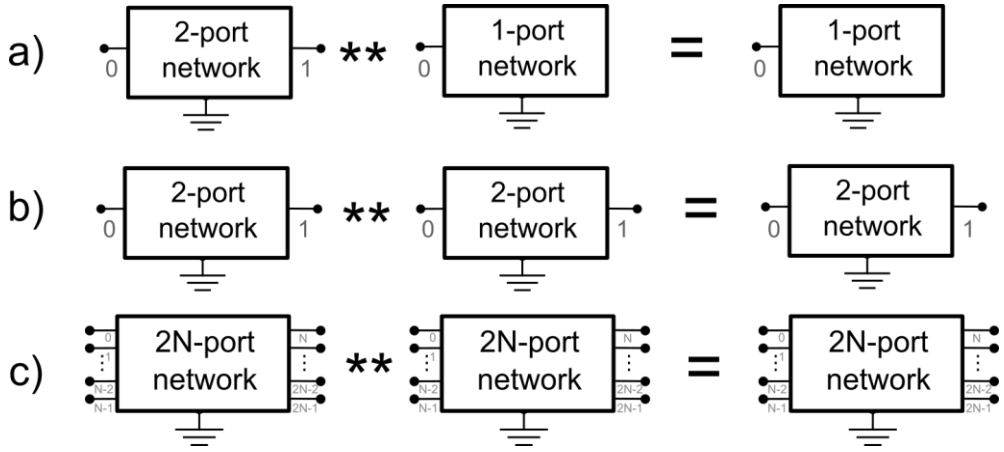

Figure 3. Illustration de quelques connexions de Network entre eux : a) connexion d'un 1-port à 2-port, générant un 1-port. b) connexion d'un 1-port à 2-port, générant un 1-port ; c) connexion de deux 2N-port entre eux, générant un 2N-port.

L'exemple suivant, illustré par la Figure 4 (a), est un exemple de cascade d'un tronçon de ligne à un court-circuit:

```
>>> short = skrf.data.wr2p2_short  # court-circuit en guide WR2.2 (1-port, 330-500 GHz)
>>> line = skrf.data.wr2p2_line  # ligne en guide WR2.2 (2-port, 330-500 GHz)
>>> delayshort = line ** short  # cf. Figure 4 (a)
```

Le *deembedding* d'un composant dont les paramètres-S du sont connus est réalisé en « inversant » le `Network` (Figure 4, b), ce qui correspond aux paramètres S de l'inverse de la matrice de transfert du `Network` originel. L'opérateur « == » permet de tester si deux objets Network sont égaux (paramètres S) :

```
>>> short_2 = line.inv ** delayshort  # deembedding, cf. Figure 4 (b)
>>> short_2 == short
True
```

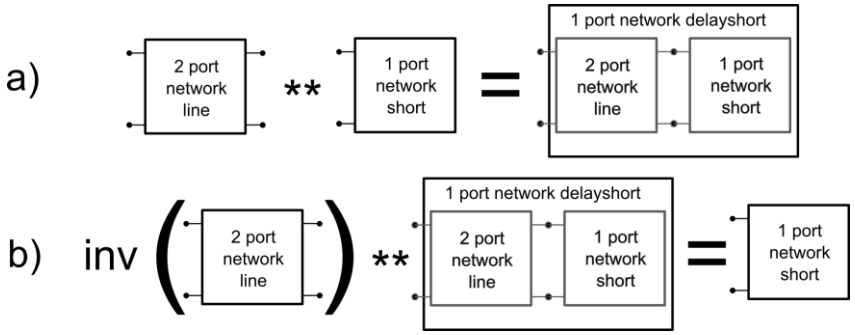

Figure 4. Exemple d'une mise en cascade (a) et de deembedding (b)

La réalisation de circuits passifs plus complexes constitués de multiples N-port connectés entre eux est également facilitée par scikit-rf. Une fois les connexions identifiées entre sous-circuits (Figure 5), il est possible de calculer les paramètres-S du N-ports résultant, ainsi que les paramètres-S, courants et tensions, aux nœuds « internes » du circuit à partir de l'algorithme décrit dans [12]. La Figure 6 est un exemple d'un diviseur de type Wilkinson, un circuit 3-ports réalisé à partir d'éléments de lignes de transmission et d'une résistance.

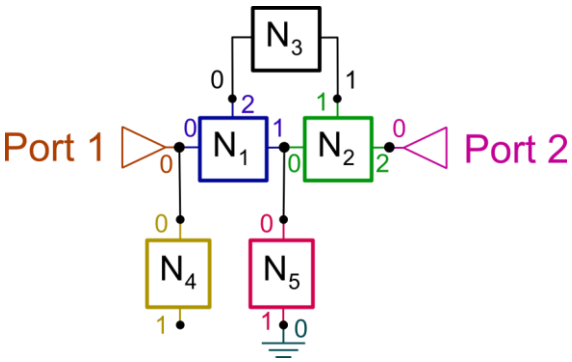

Figure 5. Illustration générale d'un circuit RF réalisé à partir de plusieurs sous-circuits connectés entre eux.

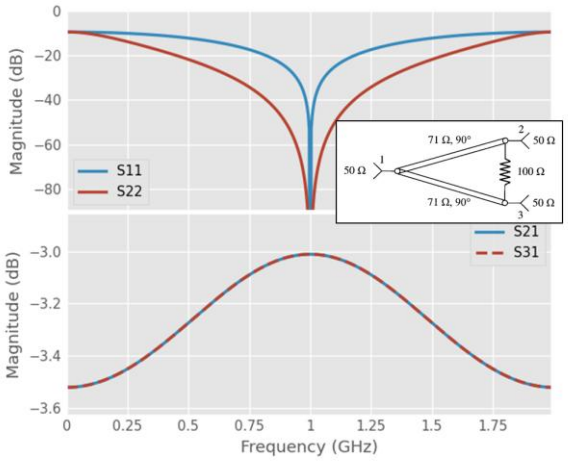

Figure 6. Réponse en fréquence d'un diviseur de type Wilkinson réalisé avec scikit-rf [2].

4 Domaine temporel

Dans le domaine des très hautes-fréquences, typiquement au-delà de 100 GHz, les problèmes d'alignement des guides d'ondes dus aux tolérances mécaniques génèrent des réflexions parasites pour chacune des connexions et provoquent des erreurs de mesures et de reproductibilité [13]. Ces réflexions parasites peuvent être supprimées en filtrant la réponse temporelle (*time-gating*), au prix d'une réduction de la précision. Le *time-gating* est une technique de traitement des signaux RF qui est couramment utilisée pour identifier les effets dus à une ou plusieurs réflexions dans un circuit et se fait généralement directement sur les analyseurs de réseau modernes. Il est possible de réaliser dans scikit-rf ces traitements à partir des mesures brutes réalisées dans le domaine fréquentiel. De cette manière, les mesures brutes sont conservées et l'analyse peut se faire hors-ligne. L'exemple suivant, issu de la documentation officielle du projet [2], illustre une utilisation classique du *time-gating* pour isoler une réponse particulière dans un signal affecté par multi-réflexions (Figure 7) :

```
# load data for the waveguide to CPW probe
>>> probe = skrf.Network('probe.s2p')
>>> s11 = probe.s11
# time-gate the first largest reflection
>>> s11_gated = s11.time_gate(center=0, span=.2)
>>> s11_gated.name='gated probe'
```

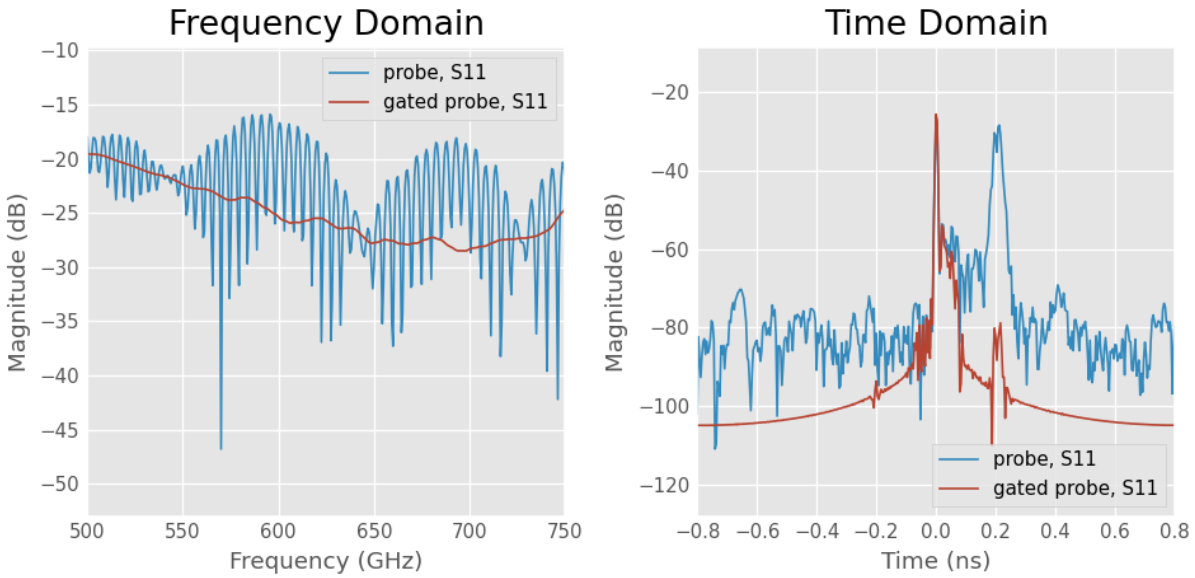

Figure 7. Exemple d'utilisation de *time-gating* pour filtrer une réponse temporelle [2]. À gauche, les réponses en fréquence du signal mesuré (bleu) et du signal filtré (rouge). À droite, les réponses temporelles du signal mesuré (bleu) et du signal filtré par une porte de largeur 0.2ns centrée sur t=0 (rouge).

5 Statistiques

scikit-rf facilite la lecture et l'analyse de séries de mesures, grâce à la classe `NetworkSet`. Elle permet d'ouvrir rapidement de multiple fichiers .sNp et d'en déduire rapidement des propriétés statistiques comme le calcul d'incertitudes. L'exemple suivant illustre comment tracer l'incertitude de mesure (en dB) associée à la répétition de 3 mesures consécutives (Figure 8).

```
from skrf.data import ro_1, ro_2, ro_3
ntwk_list = skrf.NetworkSet([ro_1, ro_2, ro_3])  # ouvre trois Network d'exemple
ntwk_list.plot_s_db()  # trace les trois S11 en dB
ntwk_list.plot_uncertainty_bounds_s_db(label='ro mean with uncertainty, S11')
```

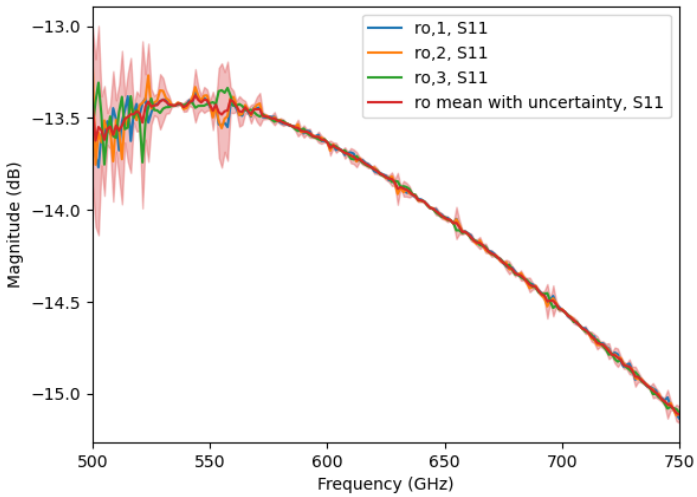

Figure 8. Tracé rapide des incertitudes d'une série de mesures.

La classe NetworkSet contient les outils de base (moyennes, écart-type, covariance, etc) pour déterminer la variabilité des mesures et la propagation d'incertitudes [14], [15]. Ces calculs statistiques peuvent être appliqués à tous les types de paramètres réseaux supportés. Dans l'exemple suivant, la moyennes et les bornes inférieures

et supérieures sont déterminées pour toutes les fréquences à partir de la valeur absolue des paramètres S pour trois écarts types :

```
ntwk_mean, ntwk_lb, ntwk_ub = ntwk_list.uncertainty_ntwk_triplet('s_mag', 3)
```

scikit-rf supporte la lecture des fichiers de données CITI et GMDIF, utilisés dans les logiciels de simulation et les appareils de mesures. Le contenu de ces fichiers est transformé en `NetworkSet` et contient tous les paramètres additionnels des séries de mesures qui sont définis dans ces fichiers. Il est possible, pour en ensemble de paramètres de mesure donné, d'extraire des mesures particulières ou d'interpoler les paramètres réseaux d'un dispositif pour un set de paramètre. Par exemple, si un dispositif actif est mesuré pour plusieurs tensions, il est possible d'interpoler ses paramètres réseaux pour des valeurs de tensions intermédiaires.

6 Calibration et *deembedding*

scikit-rf permet de réaliser des calibrations dites « hors-ligne » (*offline*), c'est-à-dire d'étalonner la mesure d'un VNA pour un dispositif RF à partir d'un ensemble de mesures préalables des références adéquates. Cette méthode de calibration à l'avantage d'être flexible et de préserver toutes les données brutes. De nombreux algorithmes de calibrations sont disponibles pour des dispositifs 1-port (SOL, SDDL, PHN, etc.) ou 2-port (SOLT, TRL, Unknown-Thru, LRM, LRRM, ou des modèles génériques à 8, 12 ou 16-termes) [9].

Pour les dispositifs hautes-fréquences (Figure 9), il n'est pas toujours possible réaliser des standards de calibration ou d'utiliser des techniques de calibrations et de caractérisations conventionnelles [16],[17]. Les mesures sont affectées par des éléments parasites (résistifs, capacitifs, etc). À partir d'une combinaison de mesures avec et sans le dispositif à tester, différentes méthodes disponibles dans la librairie permettent de réaliser le *deembedding* de ces composants : *Open, Short, Open/Short, Split Pi/Tee, Admittance/Impedance Cancel*, etc. [2].

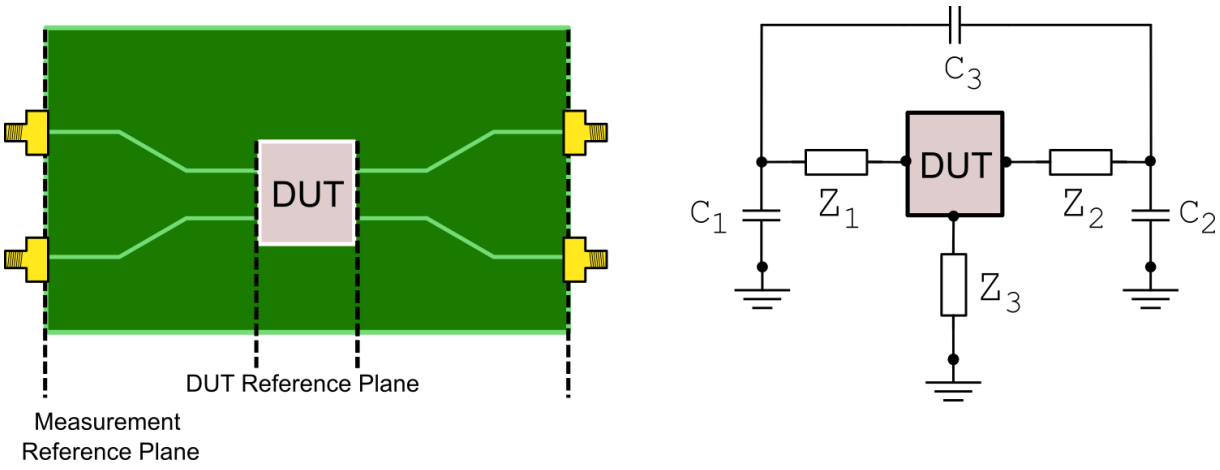

Figure 9. Après la calibration, le plan de référence des mesures ne débute par nécessairement aux bornes du dispositif à mesurer (DUT). Des éléments parasites associés aux éléments de circuits intermédiaires (pads, lignes, etc) sont également inclus dans les mesures (exemple de circuit équivalent sur la figure de droite). Ils peuvent être soustrait de la mesure à partir de techniques de *deembedding* appropriées [2].

7 Conclusion

scikit-rf est une librairie open-source gratuite écrite en Python pour l'ingénierie micro-onde et RF. Elle propose un ensemble d'outils pour la création et l'analyse de circuits RF ainsi que pour la calibration et le *deembedding*. Cet article illustre seulement une petite partie des fonctionnalités proposées par cette librairie. Beaucoup d'autres exemples sont disponibles sur le site web du projet : http://scikit-rf.org/. Le code étant open-source, il est régulièrement amélioré et de nouvelles fonctionnalités sont souvent rajoutées.

Remerciements

La librairie scikit-rf est un projet open-source et elle n'existerait pas sans l'aide et les retours des utilisateurs. Les corrections de bugs et les nouvelles fonctionnalités sont réalisées par les contributeurs au projet, tous bénévoles. La liste complète des contributeurs est visible en suivant le lien suivant : https://github.com/scikit-rf/scikit-rf/graphs/contributors.

Références bibliographiques

[1] A. Arsenovic *et al.*, 'scikit-rf: An Open Source Python Package for Microwave Network Creation, Analysis, and Calibration [Speaker's Corner]', *IEEE Microwave Magazine*, vol. 23, no. 1, pp. 98–105, Jan. 2022, doi: 10/gnrr9k.

[2] 'scikit-rf project'. http://scikit-rf.org/

[3] 'List of papers citing scikit-rf'. https://scholar.google.com/scholar?q=%22scikit-rf%22

[4] H. M. Cronson and P. G. Mitchell, 'Time-Domain Measurements of Microwave Components', *IEEE Transactions on Instrumentation and Measurement*, vol. 22, no. 4, pp. 320–325, Dec. 1973, doi: 10.1109/tim.1973.4314181.

[5] L. A. Hayden and V. K. Tripathi, 'Calibration methods for time domain network analysis', *IEEE Transactions on Microwave Theory and Techniques*, vol. 41, no. 3, pp. 415–420, Mar. 1993, doi: 10/fb5q42.

[6] D. Abessolo-Bidzo, P. Poirier, P. Descamps, and O. Hubert, 'Removing S-parameters on-wafer measurements parasitic elements using time domain gating: Application to transmission lines', in *2006 Asia-Pacific Microwave Conference*, Yokohama, Japan, Dec. 2006, pp. 575–578. doi: 10.1109/APMC.2006.4429491.

[7] J. D. Garrett and E. Tong, 'A Dispersion-Compensated Algorithm for the Analysis of Electromagnetic Waveguides', *IEEE Signal Processing Letters*, pp. 1–1, 2021, doi: 10/gkgg8f.

[8] B. Gustavsen and A. Semlyen, 'Rational approximation of frequency domain responses by vector fitting', *IEEE Trans. Power Delivery*, vol. 14, no. 3, pp. 1052–1061, Jul. 1999, doi: 10/ffvxkg.

[9] A. Rumiantsev and N. Ridler, 'VNA calibration', *IEEE Microwave Magazine*, vol. 9, no. 3, pp. 86–99, Jun. 2008, doi: 10/fczv4z.

[10] R. F. Bauer and P. Penfield, 'De-Embedding and Unterminating', *IEEE Transactions on Microwave Theory and Techniques*, vol. 22, no. 3, pp. 282–288, Mar. 1974, doi: 10.1109/tmtt.1974.1128212.

[11] A. M. Mangan, S. P. Voinigescu, M.-T. Yang, and M. Tazlauanu, 'De-embedding transmission line measurements for accurate modeling of IC designs', *IEEE Transactions on Electron Devices*, vol. 53, no. 2, pp. 235–241, Feb. 2006, doi: 10.1109/TED.2005.861726.

[12] P. Hallbjörner, 'Method for calculating the scattering matrix of arbitrary microwave networks giving both internal and external scattering', *Microw. Opt. Technol. Lett.*, vol. 38, no. 2, pp. 99–102, Jul. 2003, doi: 10/d27t7m.

[13] H. Li, A. Arsenovic, J. L. Hesler, A. R. Kerr, and R. M. Weikle, 'Repeatability and Mismatch of Waveguide Flanges in the 500–750 GHz Band', *IEEE Transactions on Terahertz Science and Technology*, vol. 4, no. 1, pp. 39–48, Jan. 2014, doi: 10.1109/TTHZ.2013.2283540.

[14] 'Evaluation of measurement data — Guide to the expression of uncertainty in measurement', 2008. Accessed: Feb. 14, 2022. [Online]. Available: https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf/cb0ef43f-baa5-11cf-3f85-4dcd86f77bd6

[15] EURAMET, ‘Guidelines on the Evaluation of Vector Network Analysers (VNA)’, Calibration Guide No. 12, 2018. [Online]. Available: https://www.euramet.org/Media/news/I-CAL-GUI-012_Calibration_Guide_No._12.web.pdf

[16] M. Kellermeier, F. Lemery, K. Floettmann, W. Hillert, and R. Aßmann, ‘Self-calibration technique for characterization of integrated THz waveguides’, *Phys. Rev. Accel. Beams*, vol. 24, no. 12, p. 122001, Dec. 2021, doi: 10.1103/PhysRevAccelBeams.24.122001.

[17] C. Cappellin, P. H. Nielsen, R. Appleby, R. Wylde, and E. Saenz, ‘Detailed design and RF analysis of a scatterometer for material characterization in the 50–750 GHz range’, in *12th European Conference on Antennas and Propagation (EuCAP 2018)*, Apr. 2018, pp. 1–5. doi: 10.1049/cp.2018.0646.