# MR Receive Coil Matching Networks

ISMRM 2024 MR Engineering Study Group Open-Source MRI Challenge Submission

Team Name: DTU RF Coils

Team Members: Rasmus Jepsen (rasale@dtu.dk), Vitaliy Zhurbenko (vizh@dtu.dk), Jan Henrik Ardenkjær-Larsen (jhar@dtu.dk) [All members are affiliated with the Technical University of Denmark (DTU)]

Repository: https://github.com/dtudk/mr-recv-coil-match-networks (also available at https://doi.org/10.5281/zenodo.10940156)

## How the tool can be used to advance MRI research

The tool is a calculator for MR receive coil matching networks, which is a supplement to one of our presentations at this year's ISMRM, "Studying Bandwidths of Balancing Preamplifier Decoupling Networks". It will also be a supplement to a paper regarding these balancing preamplifier decoupling networks, which is currently in the process of being revised.

These novel matching network topologies extend matching networks that provide noise matching and optimal preamplifier decoupling [1] such that they also provide common-mode rejection. Preamplifier decoupling is a technique that reduces coupling between coil elements, which can be implemented while simultaneously achieving optimal SNR through noise matching. The novel networks also provide common-mode rejection such that unwanted radiation from common-mode currents along transmission lines is mitigated. Integrating common-mode rejection with the matching network is anticipated to reduce the need for separate balun stages and cable traps.

There are currently freely available tools to design traditional power matching networks, such as the website at https://rftoolbox.dtu.dk/rfcalc/PowerMatch.html. However, there does not seem to be any open-source tools that implement calculations for preamplifier decoupling receive coil matching networks. This tool aims to simplify the procedure for designing receive coil matching networks by automatically calculating component values for receive coil matching networks, displaying circuit diagrams for the networks and providing estimates of their performance in terms of noise figure, decoupling and common-mode rejection.

This tool supports both high and low impedance coils. It can also be used to design matching networks for coils for arbitrary Larmor frequencies, hence it is not constrained to specific nuclei or field strengths. However, at high field, greater parasitic effects will need to be accounted for when implementing these networks. This tool is anticipated to further enable the development of X-nuclei by simplifying the procedure of creating coils for different Larmor frequencies.

Future directions for the development of this tool include extending the modelled matching networks to improve their bandwidths and to enable multifrequency X-nuclei coils, and to improve interoperability by supporting other input formats such as Touchstone files.

# The tool's functionality and capabilities

The repository contains:

- A main Jupyter notebook for designing novel common-mode rejecting matching networks and traditional matching networks for a given coil/preamplifier configuration;
- A Python package for programmatically interacting with these matching networks, which has also been published to PyPI at [https://pypi.org/project/mr-recv-coil-match-networks](https://pypi.org/project/mr-recv-coil-match-networks); and
- Jupyter notebooks showing the derivations of novel preamplifier decoupling networks that provide common-mode rejection.

The installation instructions for the tool are described in the README file in the root of the repository, which consists of installing the Python package for the tool and installing extra dependencies to enable circuit diagram rendering.

```
# inputs

# design frequency (Hz)
f0 = 127.73e6

# coil resistance (Ohm)
Rc = 1.875

# coil reactance (Ohm)
Xc = 190.788

# optimal preamplifier noise resistance (Ohm)
Rout = 45.740

# optimal preamplifier noise reactance (Ohm)
Xout = 29.525

# preamplifier input resistance (Ohm)
Ra = 45.373

# preamplifier input reactance (Ohm)
Xa = -135.560

# true if matching network input impedance should be maximized, false if it should be minimized
highZin=True

# The Q-factors below are used to provide performance estimates at the design frequency.
# These performance estimates are only rough and do not account for non-ideal effects other than the Q-factor.
# Capacitors are modelled as series RC circuits and inductors as parallel RL circuits.
# These estimates are not shown in the ideal case where both Q-factors are set to infinity.

# capacitor Q-factor at the design frequency
Qc = 200

# inductor Q-factor at the design frequency
Ql = 15

# minimum noise figure for the preamplifier (dB) (needed for noise figure estimates)
Fmin = 0.408

# noise resistance for the preamplifier (Ohm) (needed for noise figure estimates)
Rn = 2.255

# whether to display TikZ code for circuit diagrams
printTikZ = False
```

*Figure 1: Example inputs for the main Jupyter notebook*

The primary interface for the tool is the main Jupyter notebook, which provides a semi-graphical user interface for the matching network calculator. This notebook can be used by specifying inputs for the coil and preamplifier to calculate component values. Optional noise parameters and component Q-factors can be specified to enable performance estimates. Example input values are presented in Figure 1.

```
# calculate reactance matrices from [1]
# [1] W. Wang, V. Zhurbenko, J. D. Sánchez-Heredia, and J. H. Ardenkjær-Larsen, "Trade-off between preamplifier noise figure and decoupling in MRI detectors,"
wang_reactance1 = mr_recv_coil_match_networks.wang2023.calculate_reactance_matrix(Rc, Xc, Rout, Xout, Ra, Xa, highZin)
wang_reactance2 = (wang_reactance1[0], -wang_reactance1[1], wang_reactance1[2])
wang_reactances = [wang_reactance1, wang_reactance2]
for i, wang_reactance in enumerate(wang_reactances):
    print(f"Solution {i + 1}: (X11, X12, X22) = {wang_reactance} Ohm")
```

```
Solution 1: (X11, X12, X22) = (-185.75567221713882, 26.52434572331389, 152.2869588203045) Ohm
Solution 2: (X11, X12, X22) = (-185.75567221713882, -26.52434572331389, 152.2869588203045) Ohm
```

```
# ideal input impedance for matching network (for verification)
print(mr_recv_coil_match_networks.wang2023.calculate_ideal_zin(Rc, Xc, Rout, Xout, Ra, Xa, highZin))
```

```
(13.650527328051627-190.788j)
```

```
# ideal output impedance for matching network (for verification)
print(complex(Rout, Xout))
```

```
(45.74+29.525j)
```

```
# minimum and maximum values for preamplifier decoupling assuming a lossless matching network (for verification)
minimum_decoupling = mr_recv_coil_match_networks.wang2023.calculate_minimum_preamplifier_decoupling(Rout, Xout, Ra, Xa)
maximum_decoupling = mr_recv_coil_match_networks.wang2023.calculate_maximum_preamplifier_decoupling(Rout, Xout, Ra, Xa)
print(minimum_decoupling)
print(maximum_decoupling)
```

```
-4.9026612882274305
12.340301845636844
```

*Figure 2: Example intermediate results in Jupyter notebook*

The notebook prints initial intermediate results, including the impedance parameters from Wang et al. [1], the ideal input and output impedances of the matching networks and the ideal bounds for the preamplifier decoupling provided by lossless matching networks, as shown in Figure 2.

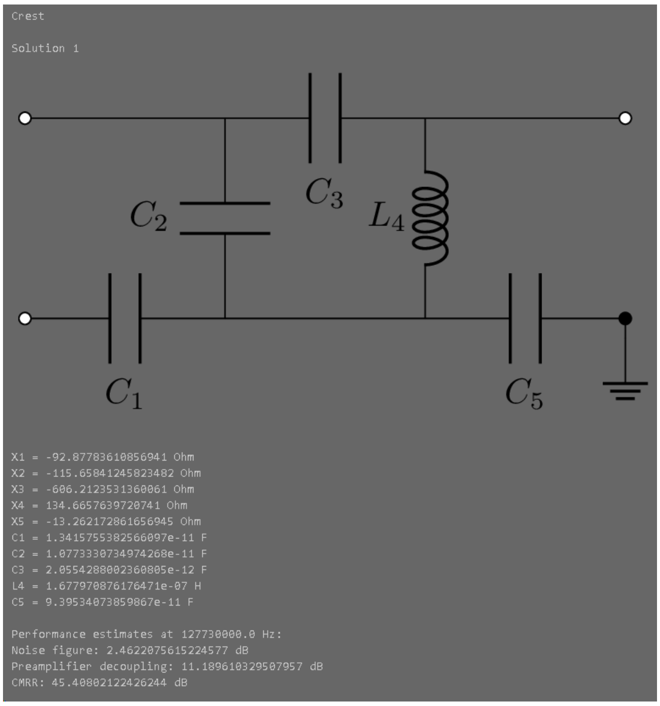


*Figure 3: Example displayed circuit design*

Circuit diagrams and ideal reactances and component values are also displayed for balancing and single-ended matching networks. An example of this output for one matching network solution is displayed in Figure 3. If component Q-factors are provided, performance estimates for the noise figure, preamplifier decoupling and common-mode rejection ratio are printed. These performance estimates are provided to help inform the user which matching network solutions will provide better performance compared to other solutions. To determine the actual noise figure, preamplifier decoupling and common-mode rejection ratio of a network, circuit simulations or EM simulations would need to be performed. Circuit diagrams are rendered by Lcapy [2] and performance estimates use Lcapy, scikit-rf [3] and an auxiliary tool for other lumped-element baluns [4] to analyse electrical networks.

The repository also exposes a Python API that allows for other programs to use the calculations for designing the matching networks. The API consists of two modules, a “wang2023” module which provides an implementation of the formulae in [1], and a “topology” module that models various matching network topologies. The “wang2023” module exposes functions for calculating the ideal impedance parameters for a preamplifier decoupling network and associated quantities.

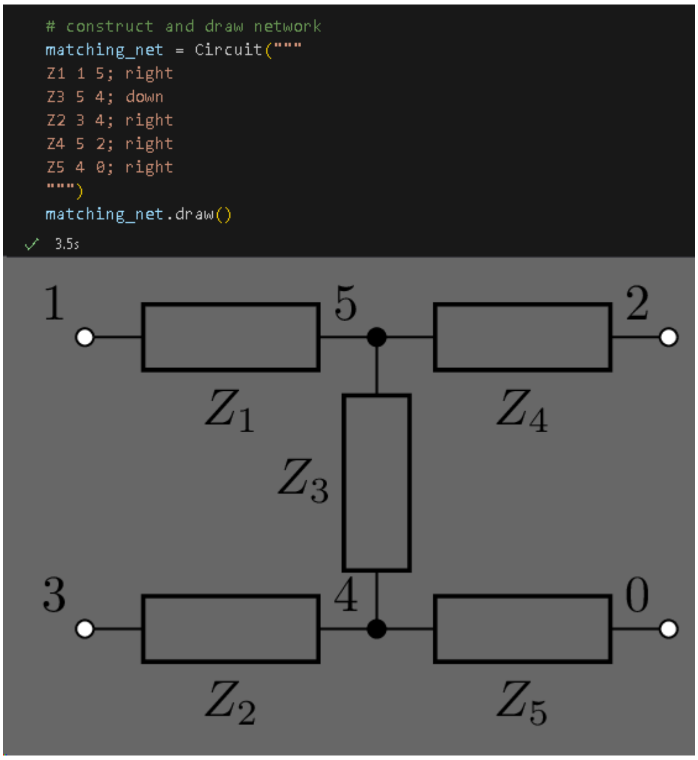


*Figure 4: Example of a circuit being modelled by its SPICE netlist*

Jupyter notebooks have also been included to show how the design equations for the balancing preamplifier decoupling networks were derived. These notebooks model matching network topologies using SPICE netlists with Lcapy [1] as presented in Figure 4. Three-port and differential-mode two-port representations are then determined to find symbolic expressions for their common-mode rejection ratios with the methods described by Kurokawa [5] and Bockelman and Eisenstadt [6], and differential-mode impedance parameters. This enables design equations for the element reactances to be found such that noise matching, optimal preamplifier decoupling and common-mode rejection are simultaneously provided.

# References


[1] W. Wang, V. Zhurbenko, J. D. Sánchez-Heredia, and J. H. Ardenkjær-Larsen, "Trade-off between preamplifier noise figure and decoupling in MRI detectors," Magnetic Resonance in Medicine, vol. 89, no. 2, pp. 859–871, 2023. doi:10.1002/mrm.29489

[2] M. Hayes, "Lcapy: symbolic linear circuit analysis with Python," PeerJ Computer Science, p. e875, Feb. 2022. [Online]. Available: https://doi.org/10.7717/peerj-cs.875

[3] A. Arsenovic, J. Hillairet, J. Anderson, H. Forst én, V. Rieß, M. Eller, N. Sauber, R. Weikle, W. Barnhart, and F. Forstmayr, "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, 2022.

[4] R. A. Jepsen, "LC Power-Matching Baluns". Zenodo, Feb. 28, 2024. doi: 10.5281/zenodo.10723786.

[5] K. Kurokawa, "Power waves and the scattering matrix," IEEE Transactions on Microwave Theory and Techniques, vol. 13, no. 2, pp. 194–202, 1965.

[6] D. Bockelman and W. Eisenstadt, "Combined differential and common-mode analysis of power splitters and combiners," IEEE Transactions on Microwave Theory and Techniques, vol. 43, no. 11, pp. 2627–2632, 1995.