

Multiple Signal Classification (MUSIC)

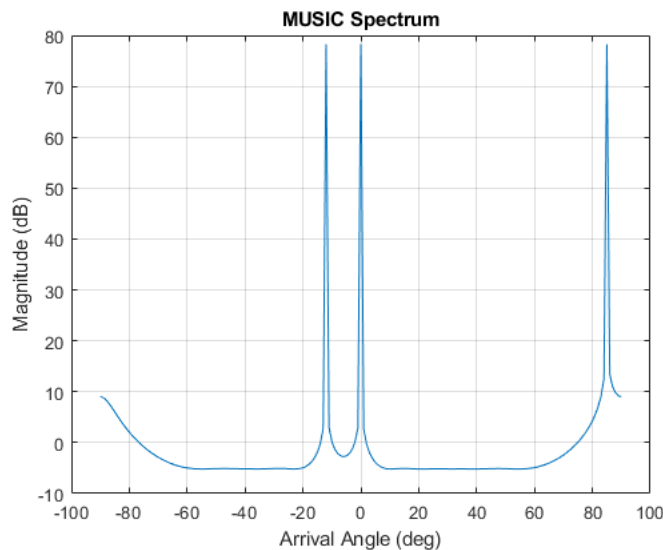
Functions: linear algebra

Inputs: complex-valued covariance matrices

Metrics: processing time, angle estimation error

1. Overview

The MULTiple Signal Classification (MUSIC) algorithm is a method for estimating the direction of arrival of signals received by an antenna array. The algorithm inputs the covariance matrix of the signals as received by each antenna element, performs an eigen decomposition, separates the result into signal and noise subspaces, derives an angular spectrum from this information, and finally searches that spectrum for peaks that represent signal arrival angles. For further reading, see <https://pdfs.semanticscholar.org/5ff7/806b44e60d41c21429e1ad2755d72bba41d7.pdf> or <https://ieeexplore.ieee.org/document/7322593>.



Example: plot showing MUSIC spectrum peaks (source: <https://www.mathworks.com/help/phased/ref/musicdoa.html>)

MilSpec’s MUSIC benchmark is based on an open-source implementation¹ manually translated from Python to C++ and integrated with an open-source linear algebra library.² In addition to the C++ translation, the MilSpec version removes external dependencies, adds assorted data input options, and incorporates benchmarking instrumentation.

2. System requirements

Platform: Ubuntu 18.04 LTS with g++ 7.4.0. Code may build and run successfully on other versions/platforms, but has not been tested with them.

Storage: ~10MB for code and sample inputs.

Dependencies: None.

¹ <https://github.com/dengjunquan/DoA-Estimation-MUSIC-ESPRIT>

² <https://gitlab.com/libeigen/eigen>

3. Build and run

To benchmark:

- Download and extract the zipfile from www.adacenter.org/milspec
- From the music/ directory, make clean && make
- Choose a testcase: covmat.in (small sample with simulated data for one signal source and two antennas), big_covmat.in (larger testcase with six signal sources and 400 antennas), or sample.in (mid-size testcase with six signal sources and 100 antennas that uses dynamically generated input data).
- From the music/ directory, ./music [testcase]
- Results are displayed in the terminal, as below:

```
Running MUSIC algorithm for DoA with 6 sources and 100 antennas
Estimated angles:
10.028 15.0419 50.1393 75.2089 90.2507 99.7771
Actual angles:
10 15 50 75 90 100
Mean squared error: 0.0296854
Average angle difference: 0.148607 degrees
Largest angle difference: 0.250696 degrees
Total time taken: 0.548075s
```

- Note that if an estimated angle differs too much from the expected value, result misalignment may occur (which the program is not designed to detect), increasing the calculated error metrics. Below, an actual 90-degree angle was missed, and an erroneous signal was detected at 12.0335 degrees:

```
Running MUSIC algorithm for DoA with 6 sources and 400 antennas
Estimated angles:
10.028 12.0335 15.0419 50.1393 75.2089 99.7771
Actual angles:
10 15 50 75 90 100
Mean squared error: 344.625
Average angle difference: 12.9712 degrees
Largest angle difference: 34.9581 degrees
Total time taken: 16.3429s
```

4. Code structure

Coming soon!

5. MilSpec development notes, errata, changelog

v0.9:

- Built baseline MUSIC implementation:
 - Translated original implementation from Python to C++

- Integrated libeigen for linear algebra computation
- Added file-based input options for (1) parameterized signal auto-generation, and (2) pregenerated covariance matrix input
- Added performance instrumentation (execution time and angle estimation error)

6. Acknowledgements

MilSpec is a project under development at the University of Michigan – Ann Arbor by Pete Ehrett, Bing Schaefer, Adrian Berding, Nathan Block, John Paul Koenig, Pranav Srinivasan, Todd Austin, and Valeria Bertacco. This project is supported by the Applications Driving Architectures Center, one of six centers of JUMP, a Semiconductor Research Corporation program co-sponsored by DARPA.

The original MUSIC implementation upon which the MilSpec version is based is available at <https://github.com/dengjunquan/DoA-Estimation-MUSIC-ESPRIT>, and the Eigen library used for linear algebra computations is available at <https://gitlab.com/libeigen/eigen>.