

# DMUSIC Algorithm

Pedro Pons-Villalonga, 2021

## 1 Singular Value Decomposition

A matrix  $M \in \mathbb{C}^{m \times n}$  can be factorized as:

$$M = USV^* \quad (1)$$

Where:

- $U \in \mathbb{C}^{m \times m}$  is unitary.
- $S \in \mathbb{R}^{m \times n}$  is a rectangular diagonal matrix of positive real numbers.
- $V \in \mathbb{C}^{n \times n}$  is unitary.
- $V^*$  denotes the complex conjugate of  $V$ .

The elements of the diagonal of  $S$  are called *singular values*, and are usually sorted in descending order (important). The columns of  $U$  and  $V$  are orthonormal bases, and we can say that the matrix  $M$  maps the vectors  $\mathbf{v}_i$  to the stretched unit vector  $\sigma_i \mathbf{u}_i$ , where  $\sigma_i$  is the  $i$ -th singular value ( $i$ -th element of the diagonal of  $S$ ), as:

$$MV = USV^*V = US \quad (2)$$

## 2 DMUSIC

Consider a signal  $y(t)$ , that we assume can be written as a sum of  $K$  (damped) harmonics and white noise  $W(t)$ :

$$y(t) = \sum_{k=1}^K A_k e^{-a_k t + i\omega_k t} + W(t) \equiv \sum_{k=1}^K A_k e^{s_k t} + W(t) \quad (3)$$

Or, for a digitized signal:

$$y(n) = \sum_{k=1}^K A_k e^{s_k \tau n} + W(n) \quad (4)$$

Where  $\tau = 1/F_s$ , the inverse of the sampling frequency, is the time interval between digitizations.

We want to recover from the signal the frequencies  $\omega_k$  and damping rates  $a_k$ . For that, if the signal is digitized, we make the assumption that we can predict the signal in an interval of length  $N$  using the preceding  $J$  ( $> K$ ) values:

$$y(n) = \sum_{i=1}^J c_{(J-i)} y(n-i) \quad (5)$$

We can build a *prediction matrix*  $A \in \mathbb{R}^{(N-J) \times J}$ , so that:

$$\begin{pmatrix} y(1) & y(2) & \cdots & y(J) \\ y(2) & y(3) & \cdots & \\ \vdots & & \ddots & \\ y(N-J) & \cdots & & y(N) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_J \end{pmatrix} = \begin{pmatrix} y(J+1) \\ y(J+2) \\ \vdots \\ y(N) \end{pmatrix} \quad (6)$$

The matrix  $A$  is a Hankel matrix. If the noise was zero, the matrix  $A$  can be expressed as a Vandermonde decomposition:

$$A = S_L C S_R^* \quad (7)$$

Where  $C \in \mathbb{C}^{K \times K}$  is a diagonal matrix, and the  $k$ -th columns of the matrices  $S_L \in \mathbb{C}^{(N-J) \times K}$  and  $S_R \in \mathbb{C}^{J \times K}$  are given by the so-called *left* and *right signal vectors*:

$$\vec{r}_L(s_k) = \begin{pmatrix} 1 & e^{s_k \tau} & e^{2s_k \tau} & \dots & e^{(N-J-1)s_k \tau} \end{pmatrix}^T \quad (8)$$

$$\vec{r}_R(s_k) = \begin{pmatrix} 1 & e^{s_k \tau} & e^{2s_k \tau} & \dots & e^{(J-1)s_k \tau} \end{pmatrix}^T \quad (9)$$

From the right signal vectors we can define the *signal space*:

$$\Omega^S \equiv \text{span}(\{\vec{r}_R(s_k)\}) \quad (10)$$

and the *null space*:

$$\Omega^N \equiv \ker(A) \quad (11)$$

We can perform a singular value decomposition on matrix  $A$ , obtaining:

$$A = U S V^* \quad (12)$$

In this case,  $U \in \mathbb{C}^{(N-J) \times (N-J)}$ ,  $V \in \mathbb{C}^{J \times J}$  are square matrices and  $S \in \mathbb{R}^{(N-J) \times J}$  is a diagonal rectangular matrix of real, positive numbers.

In the absence of noise, the rank of  $S$  is  $K$ , and only the first  $K$  singular values (elements of the diagonal of  $S$ ) are non-zero. The first  $K$  columns of  $U$  and  $V$ , then, contain all the information from the signal, and the first  $K$  columns of  $V$  span the signal space  $\Omega^S$ . Consequently, as the columns  $\{K+1, \dots, J\}$  of  $V$  correspond with null singular values, they belong to the null space.

If there is noise, the separation between signal space and null space (or, more appropriately, *noise space*) becomes less clear. However, if the signal-to-noise ratio is sufficiently high, we can safely assume that the information of the signal remains in the first  $K$  columns of  $U$  and  $V$ .

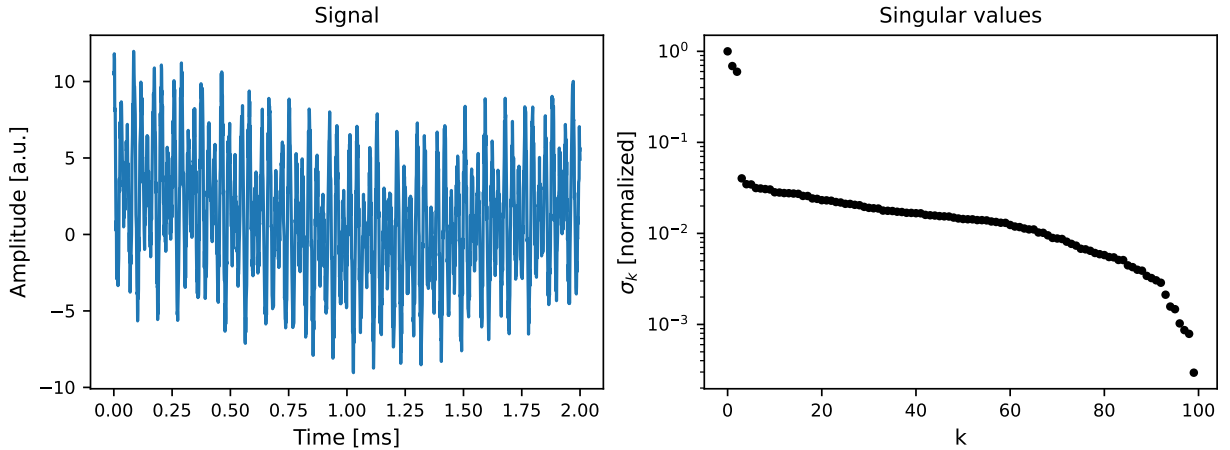


Figure 1: Synthetic (noisy) signal and singular values of the SVD decomposition of its  $A$  matrix.  $N=200$ ,  $K=3$ .

We want, given a real signal (of which we do not know the number of harmonics  $K$ ) to estimate its dominant frequencies and corresponding damping rates. For that, we need a good estimate of  $K$ , which can be obtained by looking at the SVD decomposition of the signal. Lower values of  $K$  can misidentify signal components as noise, and viceversa for higher values of  $K$ .

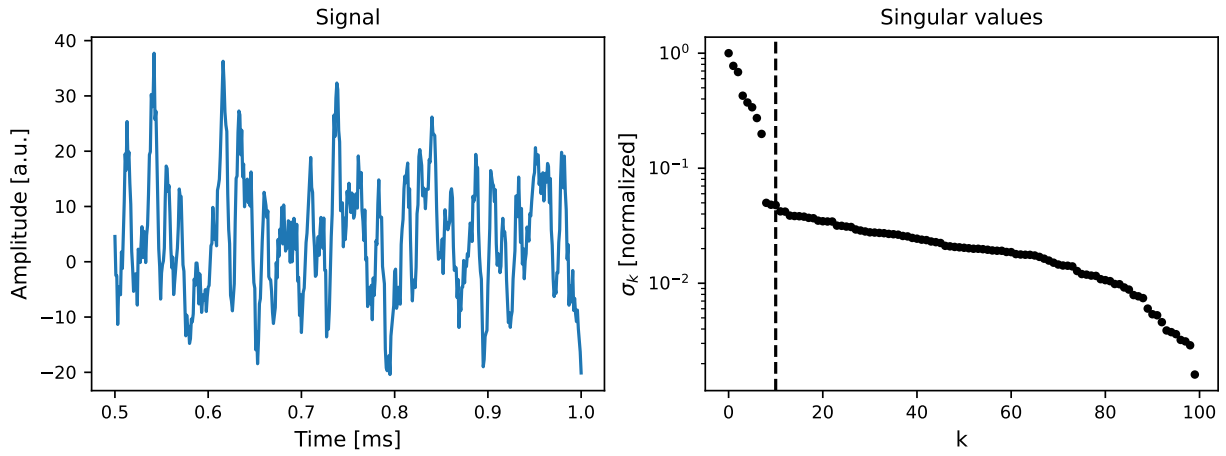


Figure 2: Synthetic (noisy) signal and singular values of its SVD decomposition.  $N=200$ ,  $K=10$ . The separation between signal and noise space is not that clear, maybe in part because the amplitudes of some harmonics are small.

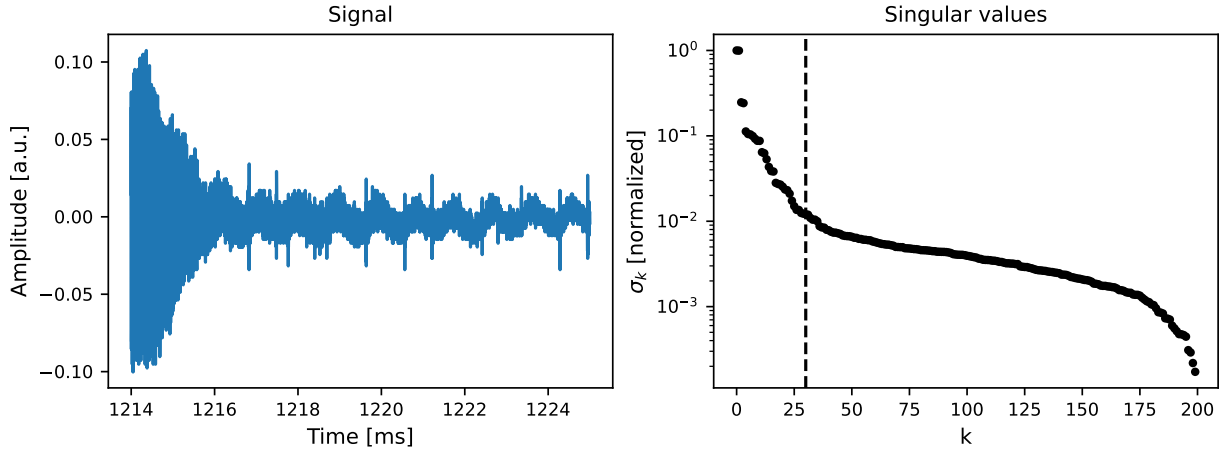


Figure 3: Real signal and singular values. Vertical line at  $K=30$  shows the justification for a commonly used value of  $K$ .  $N=400$ .

We can define a new matrix,  $V_N$ , given by the columns  $(K + 1, \dots, J)$  of  $V$ . Then, all vectors  $\vec{r}(s) \in \mathbb{C}^J$ :

$$\vec{r}(s) = (1, e^{s\tau}, e^{2s\tau}, \dots, e^{(J-1)s\tau}) \quad (13)$$

that verify:

$$\|V_N \vec{r}(s)\| = 0 \quad (14)$$

Will be orthogonal to the noise space and thus belong to the signal space. In the presence of noise, the true equality is not necessarily satisfied. However, the function:

$$P(s) \equiv \frac{1}{\|V_N \vec{r}(s)\|} \quad (15)$$

will still present maxima when  $s \simeq s_k$ , so we can use it to estimate the spectrum of the signal. To do that, we must perform a scan over a suitable grid. As we are interested only in the frequencies of the maxima, we integrate over the real part of the grid to get the frequency estimator:

$$p(\omega) = \int_{a_0}^{a_1} P(-a + i\omega) da \quad (16)$$

### 3 Implementation

A simple, reasonably fast implementation has been made with Python and MATLAB. We must realize that the bulk of the computational effort will be spent in the matrix multiplication  $V_N \vec{r}(s)$ . Python is relatively slow with loops, but the matrix multiplication routines of the package numpy are optimized. Because of that, the optimal solution consists in the creation of a big matrix  $R$  of all the  $\vec{r}(s)$ , that will be multiplied by  $V_N$  only once. This multiplication is done in parallel automatically.

---

#### Algorithm 1 DMUSIC spectrogram

---

```

procedure DMUSIC_SPEC(t, y, N, K, overlap)                                ▷ t: time array, y: raw signal
    J ← N//2                                                                ▷ For performance
    f_arr ← linspace(0, Fnyq, nf)                                         ▷ nf: number of frequencies to scan (typ. ~ 500)
    a_arr ← linspace(0, -1, na)                                           ▷ na: number of damping factors to scan (typ. ~ 50)
    R ← Empty matrix ∈ ℂJ×(nfna)                                       ▷ This way it is only created once
    for fj ← w_arr do
        for aj ← a_arr do
            s ← ai + i fi
            Make  $\vec{r}(s)$  (normalized)
            Append  $\vec{r}(s)$  to R
        end for
    end for
    Find nsteps
    Pf ← Empty matrix ∈ ℝnsteps×nf
    i ← 0
    while i < len(y) do
        ym ← Hankel(y[i:i+N])
        U, S, VT ← svd(ym)
        VN ← VT[K:(N-J)]                                                ▷ Last rows of VT, starting at K
        Ps ← norm(VNR)                                                    ▷ Ps ∈ ℝnfna
        i ← i + (1 - overlap)·N
        Append to Pf ← Integral of Ps over aarr in intervals of length na
    end while
    Pf ← 10 · log10(Pf/max(Pf))
end procedure

```

---

### References

- [1] Y. Li, J. Razavilar, and K. R. Liu, “A high-resolution technique for multidimensional NMR spectroscopy,” *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 1, pp. 78–86, 1998.
- [2] R. Kleiber, M. Borchardt, A. Könies, and C. Slaby, “Modern methods of signal processing applied to gyrokinetic simulations,” *Plasma Physics and Controlled Fusion*, vol. 63, no. 3, p. 035017, 2021.
- [3] W. Liao and A. Fannjiang, “MUSIC for single-snapshot spectral estimation: Stability and super-resolution,” *Applied and Computational Harmonic Analysis*, vol. 40, no. 1, pp. 33–67, 2016.