VFTool: Frequency-domain Vector Fitting Toolbox in MATLAB

Petr Kadlec, Martin Štumpf
Brno University of Technology
Department of Radio Electronics

25. června 2019

Abstrakt

This report describes a toolbox called VFTool prepared by scientists at Department of Radio Electronics at Brno University of Technology. The tool is a standalone toolbox in MATLAB with its own GUI (Graphical User Interface). The toolbox works in two regimes: user-guided and automatic search for residues and poles of the specified system.

1 Introduction

The code is based on papers published by B. Gustavsen in [1] and [2]. Goal for the VF (Vector Fitting) is to find a rational approximation of given function $g(s)$ in form of:

$$f(s) = \sum_{n=1}^{N} \frac{r_n}{s - p_n} + d + se \tag{1}$$

where s is the complex frequency $s = jw$ (w denotes the angular frequency $w = 2\pi f$), N stands for order of the approximation - number of residues r_n or poles p_n, and d and e are amplitude coefficients.

2 Code Description

First of all, the code works so that the order N is always even number so that all the complex poles occur in complex-conjugate pairs. THe pseudocode of the Vector Fitting process is as follows:

```
generate initial poles p
for t = 1:T
    relocate poles p
    if t == T
```

```
        compute residues r and coef. d and e
    end
end
```

2.1 Initial Poles Generation

The initial poles are either specified by the user or can be specified by an automatic procedure. In the latter case, the initial poles are produced with a linear or logarithmic distribution. In case of logarithmic distribution, the minimal and maximal order o of the given s limits are computed:

$$o = \left\lfloor \frac{\ln |\frac{\Im(s)}{2\pi}|}{\ln 10} \right\rfloor \tag{2}$$

In case of smooth system response $g(s)$, real poles are used:

$$p_r = -2\pi f \tag{3}$$

where f is a frequency point from a linear or logarithmic distribution from the interval $\langle f_{\min},\, f_{\max}\rangle$. In case of system response with very sharp peaks, N complex-conjugate poles are initiated with:

$$\begin{aligned} p_{2k-1} &= -\frac{\beta_k}{100} + j\beta_k, \\ p_{2k} &= -\frac{\beta_k}{100} - j\beta_k \\ k &= 1, 2, ..., \frac{N}{2} \end{aligned} \tag{4}$$

where β are frequency points with a linear or logarithmic distribution.

2.2 Iterative Pole Relocation

For a given frequency point s_k we solve the Least squares problem

$$\begin{aligned} \boldsymbol{A_k x} &= b_k, \\ \boldsymbol{A_k} &= \begin{bmatrix} \frac{1}{s_k - p_1} & ... & \frac{1}{s_k - p_N} & 1 & s_k & \frac{-g(s_k)}{s_k - p_1} & ... & \frac{-g(s_k)}{s_k - p_N} \end{bmatrix}, \\ b_k &= f_k, \\ \boldsymbol{x} &= \begin{bmatrix} m_1 & ... & m_N & d & e & c_1 & ... & c_N \end{bmatrix}^\intercal \end{aligned} \tag{5}$$

Here, c_n is the new n-th pole computed using all $k = 1,\, 2,\, ...,\, K$ frequency samples. In case of complex-conjugate pairs of initial poles, the corresponding elements of $\boldsymbol{A}$ are:

$$A_{k,\, i} = \frac{1}{s_k - p_i} + \frac{1}{s_k - p_i^*},\; A_{k,\, i+1} = \frac{j}{s_k - p_i} + \frac{j}{s_k - p_i^*} \tag{6}$$

The scaling factor of frequency samples h reads:

$$h = \frac{\|\boldsymbol{g}\|}{K} \tag{7}$$

where $\|\boldsymbol{g}\|$ denotes a norm of the samples vector $\boldsymbol{g}$.

In order to avoid trivial (null) solution, one equation is added to the system (5) Next, we want to solve the LS (Least Squares) problem with real values only. Therefore, the system (5) is augmented to build the final one:

$$\begin{bmatrix} \Re(\boldsymbol{A}) \\ \Im(\boldsymbol{A}) \\ \boldsymbol{0}_{1\times(N+2)},\ h\sum_{k=1}^{K} A_{k,\,1},\ \ldots,\ h\sum_{k=1}^{K} A_{k,\,N} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \Re(\boldsymbol{b}) \\ \Im(\boldsymbol{b}) \\ hK \end{bmatrix} \tag{8}$$

Equation (8) is solved using the QR decomposition with MATLAB function:

```
[Q, R] = qr(A,0);
```

where A is built so that the right-hand side of (8) is added to it's left-hand side (except of $\boldsymbol{x}$). The final system of linear equations is build from $\boldsymbol{Q}$ and $\boldsymbol{R}$. The right-hand side $\boldsymbol{b}$ is the last row of the $\boldsymbol{Q}$, columns $K+3$ to end:

$$\boldsymbol{b} = hK\boldsymbol{Q}(end, K+3 : end) \tag{9}$$

From $\boldsymbol{R}$, we take only the part $\boldsymbol{R}(K+3 : end, K+3 : end)$. Every column of the $\boldsymbol{R}$ matrix is multiplied by its norm scale factor e_k:

$$e_k = \frac{1}{\|\boldsymbol{R_k}\|} \tag{10}$$

Finally, we solve the SLE (System of Linear Equations):

$$\begin{aligned} \boldsymbol{R}\boldsymbol{x} &= \boldsymbol{b} \\ x_k &= x_k e_k \end{aligned} \tag{11}$$

First K solutions from x are the new poles $\boldsymbol{p}$ and the last one is a coefficient d. The results corresponding to initially complex poles, are rearranged in the following way:

$$c_i = x_i + jx_{i+1},\ c_{i+1} = x_i - jx_{i+1} \tag{12}$$

The model poles are eigenvalues of the following problem:

$$\boldsymbol{p} = eig(\boldsymbol{A} - \boldsymbol{b}\boldsymbol{c}^{\mathsf{T}}/d) \tag{13}$$

where matrix A has the initial poles (formed by pole generation 2.1 or taken from the previous iteration) on its diagonal. In case of complex poles when the i-th pole and $i+1$-th poles are complex-conjugate, the corresponding diagonal block of A is formed as follows:

$$\boldsymbol{A} = \begin{bmatrix} \Re(p_i) & \Im(p_i) \\ -\Im(p_i) & \Re(p_i) \end{bmatrix} \tag{14}$$

Vector $\boldsymbol{b}$ is formed by ones for real initial poles and blocks of $\begin{bmatrix} 2 \\ 0 \end{bmatrix}$ for the complex-conjugate pair. Vector c is formed by solving (11) and (12) as well as coefficient d. Coefficient d is first checked if it is higher than tolerance $\mathrm{tol} = 1 \times 10^{-8}$. In case of $d < \mathrm{tol}$, new d value is computed:

$$d = \mathrm{tol} \frac{d}{|d|} \tag{15}$$

All unstable poles having $\Re(p) > 0$ are changed to become stable:

$$p = p - 2\Im(p) \tag{16}$$

2.3 Final residues computation

At the last iteration of the process $t = T$, the final residues $\boldsymbol{r}$ are computed. AS input, we use the poles $\boldsymbol{p}$ determined in previous step. We form a matrix $\boldsymbol{A}$ so that:

$$\boldsymbol{A_k} = \begin{bmatrix} \frac{1}{s_k - p_1} & ... & \frac{1}{s_k - p_N} & 1 & s_k \end{bmatrix} \tag{17}$$

In case of complex-conjugate pair of poles the corresponding entries of $\boldsymbol{A}$ read:

$$A_{k,\,i} = \frac{1}{s_k - p_i} + \frac{1}{s_k - p_i^*},\, A_{k,\,i+1} = \frac{j}{s_k - p_i} + \frac{j}{s_k - p_i^*} \tag{18}$$

After that we form a real-valued matrix system:

$$\begin{bmatrix} \Re(\boldsymbol{A}) \\ \Im(\boldsymbol{A}) \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \Re(\boldsymbol{g}) \\ \Im(\boldsymbol{g}) \end{bmatrix} \tag{19}$$

Before we solve the system (19) it is recommended to normalize the $\boldsymbol{A}$ matrix by dividing every column of $\boldsymbol{A}$ by its norm:

$$e_k = \|\boldsymbol{A_k}\| \; \boldsymbol{A_k} = \boldsymbol{A_k}/e_k \tag{20}$$

Every member of the solution has to be unnormalized:

$$x_k = \frac{num}{e_k} \tag{21}$$

The solution x is composed in the following way:

$$\boldsymbol{x} = \begin{bmatrix} r_1 & ... & r_N & d & e \end{bmatrix} \tag{22}$$

As a final step, residues corresponding to the initially complex-conjugate pairs of poles are converted to a complex form:

$$r_i = x_i + jx_{i+1},\, r_{i+1} = x_i - jx_{i+1} \tag{23}$$

Reference

[1] Bjørn Gustavsen. Improving the pole relocating properties of vector fitting. *IEEE Transactions on Power Delivery*, 21(3):1587–1592, 2006.

[2] Bjorn Gustavsen and Adam Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on power delivery*, 14(3):1052–1061, 1999.