# Regression and cWB

February 11, 2014

Regression is naturally implemented in the cWB algorithm as a conditioning tool to subtract the persistent lines that affects the detectors data and affects the extraction of triggers disturbing the search of gravitational waves. The idea is to take informations from auxiliary channels not connected with the gravitational one to predict the effects of detector disturbances on the target channel and to subtract them.

## 1 Regression theory

Regression uses a Wiener filter to find the correlation between a target channel and one or more auxiliary channel.

### 1.1 One auxiliary channel

We consider as the target channel a discretized time series $h_i$ $(i = 1, ..., N)$, while the auxiliary channel is a time series $x_i$. For simplicity we assume they have the same number of samples.
The idea is to find a filter $a_j$ $(j = -L, ..., L)$ which allows to construct a prediction channel $s_i$ which describes the disturbances. The prediction is defined as:

$$s_i = \left( \sum_{j=-L}^{L} a_j x_{i+j} \right) \tag{1}$$

To find the coefficient of the filter we minimize the total residual:

$$\sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} \left[ h_i - \left( \sum_{j=-L}^{L} a_j x_{i+j} \right) \right]^2 \tag{2}$$

After some calculation, minimization of Eq. 2 ($\frac{\delta \chi^2}{\delta a_k} = 0$) leads to the system of 2L+1 equations (see Appendix A.1):

$$\sum_{j=-L}^{L} a_j \left( \sum_{i=1}^{N} x_{i+j} x_{i+k} \right) = \left( \sum_{i=1}^{N} x_{i+k} h_i \right) \tag{3}$$

which we simplify introducing a matricial notation:

$$R^{xx}\mathbf{a} = C^{hx} \tag{4}$$

where we have defined:

- $C^{hx}$ (vector) the correlation between $h$ and $x$: $C_k^{hx} = \sum_{i=1}^{N} h_i x_{i+k}$;
- $\mathbf{a}$ (vector) the filter: $\mathbf{a} = \{a_{-L}, ..., a_L\}$;
- $R^{xx}$ (matrix) the autocorrelation of x: $R_{jk}^{xx} = \sum_{i=1}^{N} x_{i+j} x_{i+k}$

## 1.2 Adding more auxiliary channels

The prediction can be constructed using $M$ auxiliary channels, like:

$$s_i = \left( \sum_{j=-L}^{L} a_j x_{i+j} \right) + \left( \sum_{j=-L}^{L} b_j y_{i+j} \right) + ... \tag{5}$$

In this case the filter coefficient become (2L+1 * M) and the minimization of the residual gives up (2L+1 * M) equations. However, using the matricial notations, it is possible to demonstrate that we can write all the equations in a compact form (see Appendix A.2):

$$\begin{pmatrix} R^{xx} & R^{yx} & ... \\ R^{xy} & R^{yy} & ... \\ ... & ... & ... \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ ... \end{pmatrix} = \begin{pmatrix} C^{hx} \\ C^{hy} \\ ... \end{pmatrix} \tag{6}$$

this means that considering one or more channels produces the same final equation, which is always of the type:

$$R\mathbf{a} = C \tag{7}$$

From this matricial equation we will start to find the solution.

## 1.3 Solutions

The matrix $R$ is symmetric and positive defined, with dimension $M(2L+1) \times M(2L+1) = n \times n$. So it is always possible to calculate its eigen-values ($\{\lambda_1, ..., \lambda_n\}$) and eigen-vectors ($\{\mathbf{v_1}, ..., \mathbf{v_n}\}$). For simplicity, we define eigen-values in a decreasing order ($\lambda_1 \geq ... \geq \lambda_n$).
Adopting the usual matricial notation, we define:

- $\Lambda$: eigen-values matrix ($n \times n$ dimension).

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & ... & 0 \\ 0 & \lambda_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \lambda_N \end{pmatrix}$$

- P: eigen-vector matrix ($n \times n$ dimension), each column is one of the eigen-vector.

$$P = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & ... & \mathbf{v}_N \end{pmatrix}$$

and we write:

$$P^{-1}RP = \Lambda \tag{8}$$

With simple calculations we can calculate the solution of the filter coefficient as:

$$a = P\Lambda^{-1}P^{-1}C \tag{9}$$

## 1.4 Regulators

The introduction of eigen-values is made for the application of the so-called regulators.

In most of the cases we are interested for, not all the eigen-values are significant for our purpose. This means that we do not loose information if we select only the most important eigen-values (i.e. the bigger ones).
Moreover this allows us to avoid the possibility to over-fitting the construction of prediction, i.e. the prediction could be too similar to the target channel.

In this contest we introduce a new eigen-values matrix $\Lambda'$ where we select only bigger eigen-values:

$$\Lambda_r^{-1} = \left(\begin{array}{ccccccc} 1/\lambda_1 & ... & ... & ... & ... & ... & ... \\ ... & 1/\lambda_2 & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & 1/\lambda_{th} & ... & ... & ... \\ \hline ... & ... & ... & ... & \lambda' & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & \lambda' \end{array}\right)$$

The value of $\lambda'$ could be various, we consider three cases (**regulators**):

- **hard**: $\lambda' = 0$

$$\Lambda^{-1}_{hard} = \left(\begin{array}{ccccccc} 1/\lambda_1 & ... & ... & ... & ... & ... & ... \\ ... & 1/\lambda_2 & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & 1/\lambda_{th} & ... & ... & ... \\ \hline ... & ... & ... & ... & 0 & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & 0 \end{array}\right)$$

- **soft**: $\lambda' = 1/\lambda_{th}$

$$\Lambda^{-1}_{soft} = \left(\begin{array}{ccccccc} 1/\lambda_1 & ... & ... & ... & ... & ... & ... \\ ... & 1/\lambda_2 & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & 1/\lambda_{th} & ... & ... & ... \\ \hline ... & ... & ... & ... & 1/\lambda_{th} & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & 1/\lambda_{th} \end{array}\right)$$

- **mild**: $\lambda' = 1/\lambda_1$

$$\Lambda^{-1}_{mild} = \left(\begin{array}{ccccccc} 1/\lambda_1 & ... & ... & ... & ... & ... & ... \\ ... & 1/\lambda_2 & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & 1/\lambda_{th} & ... & ... & ... \\ \hline ... & ... & ... & ... & 1/\lambda_1 & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & 1/\lambda_1 \end{array}\right)$$

# 2 How regression is implemented in cWB

The cWB algorithm is a coherent algorithm which extract from Time-Frequency (TF) decomposition the excess power in the data, which are collected in GW-like events. The TF decomposition is very useful for the application of regression, because it allows to split the calculation of the filter **a** in small sub-bands. This has the natural consequence that the filter length we need to characterize the disturbances can be reduced and conseguently the calculations are simpler.

In this way, we split the total frequency band $[min, max]$ in $K$ sub-bands of resolution $\Delta f = \frac{max-min}{K}$. For each sub-band $k$ we will calculate its proper filter $\mathbf{a}_k = P_k \Lambda_k^{-1} P_k^{-1}$.

One more advantage is the introduction of 90 degree phase data ($\tilde{x}$). This is done in CWB to have a more complete description of the signal. Regression include these information defining the time series $h_i$ and $x_i$ as linear combination of 0 and 90 phases:

$$H_i = h_i + i\tilde{h}_i$$
$$X_i = x_i + i\tilde{x}_i$$

so the residual is now defined as:

$$F_i = H_i - \sum_j (A_j X_{i+j}) \tag{10}$$

where now $A_j$ is a complex number $A_i = a_i + i\tilde{a}_i$.

The minimization is calculated on the norm of the residual. The prediction becomes:

$$S_k = \sum_j (A_j X_{j+k}) = \sum_j (a_j + i\tilde{a}_j)(x_{j+k} + i\tilde{x}_{j+k}) = (a_j x_{j+k} - \tilde{a}_j \tilde{x}_{j+k}) + i(a_j \tilde{x}_{j+k} + \tilde{a}_j x_{j+k}) \tag{11}$$

Anyway, as shown in Appendix B also in this formulation we arrive at a solution of the type:

$$R\mathbf{a} = C$$

## 2.1 cWB parameters for regression

Considering a small sub-band $k$, the regression algorithm accepts the following parameters for the calculation of the filter $\mathbf{a}_k$:

- FILTER_LENGTH
  value of L in the formula above (effective filter lenght is 2L+1)
- There are two ways to define the index $th$ of the eigenvalues
  - EIGEN_THR
    $\lambda_{th} = min\{\lambda_i > \text{EIGEN\_THR}\}$
  - EIGEN_INDEX
    $\lambda_{th} = \lambda_{\text{EIGEN\_INDEX}}$

  the variables can be used together, in this case the $\lambda_{th}$ is chosen according to the smallest one which satisfies both conditions.
- REGULATOR
  A character which identifies the regulator to use:

– 'h': hard
– 's': soft
– 'm': mild

- RANK_THR
  Each channel is labeled with a **rank** value (from 0 to 1) which refers to the channel contribution to the definition of prediction. For the calculation of the prediction are used only the channels which have rank greater than RANK_THR

# A Derivation of R matrix

## A.1 One auxiliary channel

Reminding that we want to minimize the residual:

$$\sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} \left[ h_i - \left( \sum_{j=-L}^{L} a_j x_{i+j} \right) \right]^2 \tag{12}$$

Calculate the first derivative respect to the generic coefficient $a_k$:

$$\begin{aligned}
\frac{\delta \chi^2}{\delta a_k} &= \frac{\delta}{\delta a_k} \left[ \sum_{i=1}^{N} \left( h_i - \sum_{j=-L}^{L} a_j x_{i+j} \right)^2 \right] = \\
&= \sum_{i=1}^{N} \left[ \frac{\delta}{\delta a_k} \left( h_i - \sum_{j=-L}^{L} a_j x_{i+j} \right)^2 \right] = \\
&= \sum_{i=1}^{N} \left[ \left( h_i - \sum_{j=-L}^{L} a_j x_{i+j} \right) \left( -2 \frac{\delta}{\delta a_k} \sum_{j=-L}^{L} a_j x_{i+j} \right) \right] = \\
&= \sum_{i=1}^{N} \left[ \left( h_i - \sum_{j=-L}^{L} a_j x_{i+j} \right) \left( -2 \sum_{j=-L}^{L} x_{i+j} \frac{\delta a_j}{\delta a_k} \right) \right] = \\
&= -2 \sum_{i=1}^{N} \left[ x_{i+k} h_i - \sum_{j=-L}^{L} a_j x_{i+j} x_{i+k} \right] = \\
&= -2 \left[ \left( \sum_{i=1}^{N} x_{i+k} h_i \right) - \sum_{j=-L}^{L} a_j \left( \sum_{i=1}^{N} x_{i+j} x_{i+k} \right) \right]
\end{aligned} \tag{13}$$

We put the $2L+1$ first derivative equal to zero:

$$\sum_{j=-L}^{L} a_j \left( \sum_{i=1}^{N} x_{i+j} x_{i+k} \right) = \left( \sum_{i=1}^{N} x_{i+k} h_i \right) \tag{14}$$

Remimding the matrix notation:

- $C_k^{hx} = \sum_{i=1}^{N} h_i x_{i+k}$;
- $\mathbf{a} = \{a_{-L}, ..., a_L\}$;
- $R_{jk}^{xx} = \sum_{i=1}^{N} x_{i+j} x_{i+k}$

we can write the previous equation as:

$$R^{xx}\mathbf{a} = C^{hx} \tag{15}$$

## A.2 More auxiliary channels

Suppose to have M auxiliary channels $x$, $y$, ... So the residual is defined:

$$\sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} \left[ h_i - \left( \sum_{j=-L}^{L} a_j x_{i+j} \right) - \left( \sum_{j=-L}^{L} b_j y_{i+j} \right) - ... \right]^2 \tag{16}$$

To minimize we should resolve the following equations:

$$\begin{cases} \frac{\delta \chi^2}{\delta a_k} = 0 \\ \frac{\delta \chi^2}{\delta b_k} = 0 \\ \quad ... \end{cases} \tag{17}$$

Calculating the derivative respect to $a_k$ leads to:

$$\begin{aligned} \frac{\delta \chi^2}{\delta a_k} &= \frac{\delta}{\delta a_k} \left[ \sum_{i=1}^{N} \left( h_i - \sum_{j=-L}^{L} a_j x_{i+j} - \sum_{j=-L}^{L} b_j y_{i+j} - ... \right)^2 \right] = \\ &= \sum_{i=1}^{N} \left[ \left( h_i - \sum_{j=-L}^{L} a_j x_{i+j} - \sum_{j=-L}^{L} b_j y_{i+j} - ... \right) \left( -2 \frac{\delta}{\delta a_k} \sum_{j=-L}^{L} a_j x_{i+j} \right) \right] = \\ &= -2 \sum_{i=1}^{N} \left[ x_{i+k} h_i - \sum_{j=-L}^{L} a_j x_{i+j} x_{i+k} - \sum_{j=-L}^{L} b_j y_{i+j} x_{i+k} - ... \right] = \\ &= -2 \left[ \left( \sum_{i=1}^{N} x_{i+k} h_i \right) - \sum_{j=-L}^{L} a_j \left( \sum_{i=1}^{N} x_{i+j} x_{i+k} \right) - \sum_{j=-L}^{L} b_j \left( \sum_{i=1}^{N} y_{i+j} x_{i+k} \right) - ... \right] \end{aligned} \tag{18}$$

It is easy to see that for other derivatives the equation is similar. Combining all the equations and using matrix notations we obtain:

$$\begin{cases} R^{xx}\mathbf{a} + R^{yx}\mathbf{b} + ... = C^{hx} \\ R^{xy}\mathbf{a} + R^{yy}\mathbf{b} + ... = C^{hy} \\ \qquad\qquad ... \end{cases} \tag{19}$$

where we have defined the correlation between two channels as: $R^{xy}_{jk} = \sum_{i=1}^{N} x_{i+j} y_{i+k}$

We resume all the previous equation in the following way:

$$\begin{pmatrix} R^{xx} & R^{yx} & ... \\ R^{xy} & R^{yy} & ... \\ ... & ... & ... \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ ... \end{pmatrix} = \begin{pmatrix} C^{hx} \\ C^{hy} \\ ... \end{pmatrix} \tag{20}$$

Which is equivalent to consider the M channels as an unique one with M times size.

# B Imaginary filters

## B.1 One channel

Let's consider a generic channel x composed of 0 ($x$) and 90 ($\tilde{x}$) phase data. We can describe a single channel as composed of a real and imaginary part:

$$X_i = x_i + i\tilde{x}_i \tag{21}$$

the same for filter:

$$A_i = a_i + i\tilde{a}_i \tag{22}$$

So if we define:

$$F_i = \left[ H_i - \sum_j (A_j X_{i+j}) \right] \tag{23}$$

we can write the residual:

$$\sum_{i=1}^{N} \left[ F_i \bar{F}_i \right] = \sum_{i=1}^{N} \left[ \Re^2(F_i) + \Im^2(F_i) \right] \tag{24}$$

where $\bar{}$ is the complex conjugate, $\Re$ and $\Im$ are real and imaginary parts.

Let's rewrite the residual and then calculate the first derivative respect to $a_k$ and $\tilde{a}_j$

$$A_j X_{i+j} = (a_j + i\tilde{a}_j)(x_{i+j} + i\tilde{x}_{i+j}) = (a_j x_{i+j} - \tilde{a}_j \tilde{x}_{i+j}) + i(a_j \tilde{x}_{i+j} + \tilde{a}_j x_{i+j}) \tag{25}$$

so

$$\begin{array}{l}\Re(F_i) = h_i - \sum_j (a_j x_{i+j} - \tilde{a}_j \tilde{x}_{i+j}) \\ \Im(F_i) = \tilde{h}_i - \sum_j (a_j \tilde{x}_{i+j} + \tilde{a}_j x_{i+j})\end{array} \tag{26}$$

Calculating the first derivative respect to $a_k$

$$\begin{aligned}\frac{\delta}{\delta a_k}\Re^2(F_i) &= 2\Re(F_i)\frac{\delta}{\delta a_k}\Re(F_i) = \\ &= 2\Re(F_i)\frac{\delta}{\delta a_k}\left[-\sum_j (a_j x_{i+j})\right] = -2\Re(F_i) x_{i+k} = \\ &= -2\left[h_i - \sum_j (a_j x_{i+j} - \tilde{a}_j \tilde{x}_{i+j})\right] x_{i+k}\end{aligned} \tag{27}$$

and:

$$\begin{aligned}\frac{\delta}{\delta a_k}\Im^2(F_i) &= 2\Im(F_i)\frac{\delta}{\delta a_k}\Im(F_i) = \\ &= 2\Im(F_i)\frac{\delta}{\delta a_k}\left[-\sum_j (a_j \tilde{x}_{i+j})\right] = -2\Im(F_i) \tilde{x}_{i+k} = \\ &= -2\left[\tilde{h}_i - \sum_j (a_j \tilde{x}_{i+j} + \tilde{a}_j x_{i+j})\right] \tilde{x}_{i+k}\end{aligned} \tag{28}$$

Combining the two equations we have:

$$\sum_{i=1}^{N}(h_i x_{i+k} + \tilde{h}_i \tilde{x}_{i+k}) = \sum_j a_j \sum_{i=1}^{N}(x_{i+j}x_{i+k} + \tilde{x}_{i+j}\tilde{x}_{i+k}) + \sum_j \tilde{a}_j \sum_{i=1}^{N}(x_{i+j}\tilde{x}_{i+k} - \tilde{x}_{i+j}x_{i+k}) \tag{29}$$

or, written more simply:

$$C_k^{hx} + C_k^{\tilde{h}\tilde{x}} = \sum_j a_j (R_{jk}^{xx} + R_{jk}^{\tilde{x}\tilde{x}}) + \sum_j \tilde{a}_j (R_{jk}^{x\tilde{x}} - R_{jk}^{\tilde{x}x}) \tag{30}$$

where:

$$\begin{cases} C_k^{ab} = \sum_{i=1}^{N} a_i b_{i+k} \\ R_{jk}^{ab} = \sum_{i=1}^{N} a_{i+j} b_{i+k} \end{cases} \tag{31}$$

Calculate derivative respect to $\tilde{a}_k$

$$\begin{aligned}\frac{\delta}{\delta \tilde{a}_k}\Re^2(F_i) &= 2\Re(F_i)\frac{\delta}{\delta \tilde{a}_k}\Re(F_i) = \\ &= 2\Re(F_i)\frac{\delta}{\delta \tilde{a}_k}\left[\tilde{a}_j \tilde{x}_{i+j}\right] = 2\Re(F_i) \tilde{x}_{i+k} \\ &= 2\left[h_i - \sum_j (a_j x_{i+j} - \tilde{a}_j \tilde{x}_{i+j})\right] \tilde{x}_{i+k}\end{aligned} \tag{32}$$

and:

$$\begin{aligned}\frac{\delta}{\delta\tilde{a}_k}\Im^2(F_i) &= 2\Im(F_i)\frac{\delta}{\delta\tilde{a}_k}\Im(F_i) = \\ &= 2\Im(F_i)\frac{\delta}{\delta\tilde{a}_k}\left[-\tilde{a}_j x_{i+j}\right] = -2\Im(F_i)x_{i+k} = \\ &= -2\left[\tilde{h}_i - \sum_j(a_j\tilde{x}_{i+j} + \tilde{a}_j x_{i+j})\right]x_{i+k}\end{aligned} \tag{33}$$

Combining and using same notations as before:

$$C_k^{h\tilde{x}} - C_k^{\tilde{h}x} = \sum_j a_j(R_{jk}^{x\tilde{x}} - R_{jk}^{\tilde{x}x}) - \sum_j \tilde{a}_j(R_{jk}^{\tilde{x}\tilde{x}} + R_{jk}^{xx}) \tag{34}$$

Using matrix notations:

$$\begin{cases}\mathbf{C}^{hx} + \mathbf{C}^{\tilde{h}\tilde{x}} = (R^{xx} + R^{\tilde{x}\tilde{x}})\mathbf{a} + (R^{x\tilde{x}} - R^{\tilde{x}x})\tilde{\mathbf{a}} \\ \mathbf{C}^{h\tilde{x}} - \mathbf{C}^{\tilde{h}x} = (R^{x\tilde{x}} - R^{\tilde{x}x})\mathbf{a} - (R^{\tilde{x}\tilde{x}} + R^{xx})\tilde{\mathbf{a}}\end{cases} \tag{35}$$

We can put all the system in a unique matricial equation (inverting the signs of second equation):

$$\begin{pmatrix} R^{xx} + R^{\tilde{x}\tilde{x}} & R^{x\tilde{x}} - R^{\tilde{x}x} \\ R^{\tilde{x}x} - R^{x\tilde{x}} & R^{\tilde{x}\tilde{x}} + R^{xx}\end{pmatrix}\begin{pmatrix}\mathbf{a} \\ \tilde{\mathbf{a}}\end{pmatrix} = \begin{pmatrix}\mathbf{C}^{hx} + \mathbf{C}^{\tilde{h}\tilde{x}} \\ \mathbf{C}^{\tilde{h}x} - \mathbf{C}^{h\tilde{x}}\end{pmatrix} \tag{36}$$

## B.2 More auxiliary channels

If we consider M channels we have:

$$F_i = \left[H_i - \sum_j(A_jX_{i+j}) - \sum_j(B_jY_{i+j}) - ...\right] \tag{37}$$

and conseguently:

$$\begin{aligned}\Re(F_i) = h_i - \textstyle\sum_j(a_jx_{i+j} - \tilde{a}_j\tilde{x}_{i+j}) - \sum_j(b_jy_{i+j} - \tilde{b}_j\tilde{y}_{i+j}) - ... \\ \Im(F_i) = \tilde{h}_i - \textstyle\sum_j(a_j\tilde{x}_{i+j} + \tilde{a}_jx_{i+j}) - \sum_j(b_j\tilde{y}_{i+j} + \tilde{b}_jy_{i+j}) - ...\end{aligned} \tag{38}$$

The first derivatives respect to $a_k$ and $\tilde{a}_k$ are similar to the case of one channel:

$$\begin{cases}\frac{\delta}{\delta a_k}\Re^2(F_i) = -2\Re(F_i)x_{i+k} \\ \frac{\delta}{\delta a_k}\Im^2(F_i) = -2\Im(F_i)\tilde{x}_{i+k} \\ \frac{\delta}{\delta\tilde{a}_k}\Re^2(F_i) = 2\Re(F_i)\tilde{x}_{i+k} \\ \frac{\delta}{\delta\tilde{a}_k}\Im^2(F_i) = -2\Im(F_i)x_{i+k}\end{cases} \tag{39}$$

So we have:

$$\begin{cases} \mathbf{C}^{hx} + \mathbf{C}^{\tilde{h}\tilde{x}} = (R^{xx} + R^{\tilde{x}\tilde{x}})\mathbf{a} + (R^{x\tilde{x}} - R^{\tilde{x}x})\tilde{\mathbf{a}} + (R^{yx} + R^{\tilde{y}\tilde{x}})\mathbf{b} + (R^{y\tilde{x}} - R^{\tilde{y}x})\tilde{\mathbf{b}} \\ \mathbf{C}_{h\tilde{x}} - \mathbf{C}_{\tilde{h}x} = (R^{x\tilde{x}} - R^{\tilde{x}x})\mathbf{a} - (R^{\tilde{x}\tilde{x}} - R^{xx})\tilde{\mathbf{a}} + (R^{y\tilde{x}} - R^{\tilde{y}x})\mathbf{b} - (R^{\tilde{y}\tilde{x}} - R^{yx})\tilde{\mathbf{b}} \end{cases} \tag{40}$$

and similar for other derivatives. Combining:

$$\begin{pmatrix} R^{xx} + R^{\tilde{x}\tilde{x}} & R^{x\tilde{x}} - R^{\tilde{x}x} & R^{yx} + R^{\tilde{y}\tilde{x}} & R^{y\tilde{x}} - R^{\tilde{y}x} & ... \\ R^{\tilde{x}x} - R^{x\tilde{x}} & R^{\tilde{x}\tilde{x}} + R^{xx} & R^{\tilde{y}x} - R^{y\tilde{x}} & R^{\tilde{y}\tilde{x}} + R^{yx} & ... \\ R^{xy} + R^{\tilde{x}\tilde{y}} & R^{x\tilde{y}} - R^{\tilde{x}y} & R^{yy} + R^{\tilde{y}\tilde{y}} & R^{y\tilde{y}} - R^{\tilde{y}y} & ... \\ R^{\tilde{x}y} - R^{x\tilde{y}} & R^{\tilde{x}\tilde{y}} + R^{xy} & R^{\tilde{y}y} - R^{y\tilde{y}} & R^{\tilde{y}\tilde{y}} + R^{yy} & ... \\ ... & ... & ... & ... & ... \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \tilde{\mathbf{a}} \\ \mathbf{b} \\ \tilde{\mathbf{b}} \\ ... \end{pmatrix} = \begin{pmatrix} \mathbf{C}^{hx} + \mathbf{C}^{\tilde{h}\tilde{x}} \\ \mathbf{C}^{\tilde{h}x} - \mathbf{C}^{h\tilde{x}} \\ \mathbf{C}^{hy} + \mathbf{C}^{\tilde{h}\tilde{y}} \\ \mathbf{C}^{\tilde{h}y} - \mathbf{C}^{h\tilde{y}} \\ ... \end{pmatrix} \tag{41}$$

# C Properties of R matrix

The auto-correlation matrix is symmetric in all the previous cases ($R^{jk} = R^{kj}$). This is a natural consequence of the fact we are minimizing a quadratic form. Let see in all the cases.
We remind that generally $R$ is a block matrix, where the diagonal matrices are auto-correlation of the same channels ($R_{xx}$) and off-diagonal matrices are cross-correlation of different channels ($R_{xy}$).

- **Real filter**
  The diagonal matrices are naturally symmetric:
  $R_{xx}^{jk} = \sum_{i=1}^{N} x_{i+j}x_{i+k} = \sum_{i=1}^{N} x_{i+k}x_{i+j} = R_{xx}^{kj}$
  The off-diagonal part is symmetric if $R_{xy}^{jk} = R_{yx}^{kj}$:
  $R_{xy}^{jk} = \sum_{i=1}^{N} x_{i+j}y_{i+k} = \sum_{i=1}^{N} y_{i+k}x_{i+j} = R_{yx}^{kj}$
- **Imaginary filter**
  This is already demonstrated from the two previous cases: $R$ matrix is composed from a sum of symmetric block matrices, so it is symmetric.