# 常见 SSVEP 信号处理算法（空间滤波器）

更新进展：终于把 ms- 写完了，累死我了

近期计划：谁开完组会还有计划啊？

**建议各位同僚读完硕士赶紧去就业，千万不要盲目读博、投身火海。**

---

## 公式变量符号及说明

| 符号名称 | 物理含义 |
| --- | --- |
| $N_e$ | 刺激类别数 |
| $N_t$ | 训练样本数 |
| $N_c$ | 导联数 |
| $N_p$ | 单试次采样点数 |
| $N_h$ | 正余弦信号谐波个数 |
| $N_k$ | 保留子空间个数 |
| $f_s$ | EEG 信号采样率 |
| $Cov(\boldsymbol{x},\boldsymbol{y}), Cov(\boldsymbol{X},\boldsymbol{Y})$ | 向量（矩阵）$\boldsymbol{x}$（$\boldsymbol{X}$）和 $\boldsymbol{y}$（$\boldsymbol{Y}$）的协方差（阵） |
| $Var(\boldsymbol{x}), Var(\boldsymbol{X})$ | 向量（矩阵）$\boldsymbol{x}$（$\boldsymbol{X}$）的方差（自协方差）（阵） |
| $corr(\boldsymbol{x},\boldsymbol{y}), corr2(\boldsymbol{X},\boldsymbol{Y})$ | 向量（矩阵）$\boldsymbol{x}$（$\boldsymbol{X}$）和 $\boldsymbol{y}$（$\boldsymbol{Y}$）的 *Pearson* 相关系数 |
| $\boldsymbol{I}_N$ | $N$ 阶单位阵 |
| $\boldsymbol{\mathcal{I}}_{M,N} \in \mathbb{R}^{N\times(MN)}$ | $M$ 个 $\boldsymbol{I}_N$ 的横向拼接，$[\boldsymbol{I}_N,\cdots,\boldsymbol{I}_N]$ |
| $\boldsymbol{\chi}$ | EEG 测试数据矩阵 |
| $\boldsymbol{X}$ | EEG 训练数据矩阵 |
| $\boldsymbol{x}$ | EEG 训练数据序列 |
| $\boldsymbol{Y}$ | 人工构建正余弦模板 |
| $\boldsymbol{X}^i, \boldsymbol{x}^i$ | 第 $i$ 试次或第 $i$ 导联数据，详见各部分具体说明 |
| $\boldsymbol{X}_k, \boldsymbol{x}_k$ | 第 $k$ 类别数据 |

| 符号名称 | 物理含义 |
| --- | --- |
| $\bar{\boldsymbol{X}}_k$, $\bar{\boldsymbol{x}}_k$ | 类别样本中心，由 $\boldsymbol{X}_k$ 或 $\boldsymbol{x}_k$ 按试次叠加平均获得 |
| $\bar{\bar{\boldsymbol{X}}}$, $\bar{\bar{\boldsymbol{x}}}$ | 总体样本中心，由 $\bar{\boldsymbol{X}}_k$ 或 $\bar{\boldsymbol{x}}_k$ 按类别叠加平均获得 |
| $\boldsymbol{\omega}, \boldsymbol{U}, \boldsymbol{V}$ ... | 低维空间滤波器 |
| $\boldsymbol{W}$ | 高维空间滤波器，由数个低维空间滤波器集成获得 |

（在无特殊说明的情况下，所有训练数据默认经过了零均值化处理）

---

# 1. 典型相关性分析

**Canonical correlation analysis, CCA**

## 1.1 标准 CCA：CCA

论文链接 | 代码：cca.cca()

对于具有 $N_e$ 个不同频率刺激的 SSVEP-BCI 系统，频率索引 $k$ 的人工构建正余弦模板 $\boldsymbol{Y}_k$ 可表示为：

$$\boldsymbol{Y}_k = \begin{bmatrix} \sin\left(2\pi f_k n\right) \\ \cos\left(2\pi f_k n\right) \\ \sin\left(4\pi f_k n\right) \\ \cos\left(4\pi f_k n\right) \\ \cdots \\ \sin\left(2N_h\pi f_k n\right) \\ \cos\left(2N_h\pi f_k n\right) \end{bmatrix} \in \mathbb{R}^{(2N_h)\times N_p},\ n = \left[\frac{1}{f_s}, \frac{2}{f_s}, ..., \frac{N_p}{f_s}\right] \tag{1-1-1}$$

对于单试次多导联 EEG 数据 $\boldsymbol{X} \in \mathbb{R}^{N_c\times N_p}$ 以及假定的所属类别 $k$，CCA 的优化目标为一组投影向量 $\hat{\boldsymbol{U}}_k$ 和 $\hat{\boldsymbol{V}}_k$，使得一维信号 $\hat{\boldsymbol{U}}_k\boldsymbol{X}$ 与 $\hat{\boldsymbol{V}}_k\boldsymbol{Y}_k$ 之间相关性最大化，其目标函数为：

$$\hat{\boldsymbol{U}}_k, \hat{\boldsymbol{V}}_k = \underset{\boldsymbol{U}_k,\boldsymbol{V}_k}{\arg\max} \frac{Cov\left(\boldsymbol{U}_k\boldsymbol{X}, \boldsymbol{V}_k\boldsymbol{Y}_k\right)}{\sqrt{Var\left(\boldsymbol{U}_k\boldsymbol{X}\right)}\sqrt{Var\left(\boldsymbol{V}_k\boldsymbol{Y}_k\right)}} = \underset{\boldsymbol{U}_k,\boldsymbol{V}_k}{\arg\max} \frac{\boldsymbol{U}_k\boldsymbol{C}_{\boldsymbol{X}\boldsymbol{Y}_k}{\boldsymbol{V}_k}^T}{\sqrt{\boldsymbol{U}_k\boldsymbol{C}_{\boldsymbol{X}\boldsymbol{X}}{\boldsymbol{U}_k}^T}\sqrt{\boldsymbol{V}_k\boldsymbol{C}_{\boldsymbol{Y}_k\boldsymbol{Y}_k}{\boldsymbol{V}_k}^T}} \tag{1-1-2}$$

$$
\begin{cases}
\boldsymbol{C_{XX}} = \dfrac{1}{N_p - 1}\boldsymbol{X}\boldsymbol{X}^T \in \mathbb{R}^{N_c \times N_c} \\
\boldsymbol{C_{Y_kY_k}} = \dfrac{1}{N_p - 1}\boldsymbol{Y}_k{\boldsymbol{Y}_k}^T \in \mathbb{R}^{(2N_h) \times (2N_h)} \\
\boldsymbol{C_{XY_k}} = \dfrac{1}{N_p - 1}\boldsymbol{X}{\boldsymbol{Y}_k}^T \in \mathbb{R}^{N_c \times (2N_h)} \\
\boldsymbol{C_{Y_kX}} = \dfrac{1}{N_p - 1}\boldsymbol{Y}_k\boldsymbol{X}^T \in \mathbb{R}^{(2N_h) \times N_c}
\end{cases}
\tag{1-1-3}
$$

根据最优化理论，函数 (1-2) 的等效形式为：

$$
\begin{cases}
\max\limits_{\boldsymbol{U}_k, \boldsymbol{V}_k} \boldsymbol{U}_k\boldsymbol{C_{XY_k}}{\boldsymbol{V}_k}^T \\
s.t.\ \boldsymbol{U}_k\boldsymbol{C_{XX}}{\boldsymbol{U}_k}^T = \boldsymbol{V}_k\boldsymbol{C_{Y_kY_k}}{\boldsymbol{V}_k}^T = 1
\end{cases}
\tag{1-1-4}
$$

利用 *Lagrandian* 乘子法构建多元函数 $J(\boldsymbol{U}_k, \boldsymbol{V}_k, \lambda, \theta)$：

$$
J = \boldsymbol{U}_k\boldsymbol{C_{XY_k}}{\boldsymbol{V}_k}^T - \frac{1}{2}\lambda\left(\boldsymbol{U}_k\boldsymbol{C_{XX}}{\boldsymbol{U}_k}^T - 1\right) - \frac{1}{2}\theta\left(\boldsymbol{V}_k\boldsymbol{C_{Y_kY_k}}{\boldsymbol{V}_k}^T - 1\right)
\tag{1-1-5}
$$

对函数 $J$ 求偏导数并置零：

$$
\begin{cases}
\dfrac{\partial J}{\partial \boldsymbol{U}_k} = \boldsymbol{C_{XY_k}}{\boldsymbol{V}_k}^T - \lambda\boldsymbol{C_{XX}}{\boldsymbol{U}_k}^T = 0 \quad (I) \\
\dfrac{\partial J}{\partial \boldsymbol{V}_k} = \boldsymbol{C_{Y_kX}}{\boldsymbol{U}_k}^T - \theta\boldsymbol{C_{Y_kY_k}}{\boldsymbol{V}_k}^T = 0 \quad (II)
\end{cases}
\tag{1-1-6}
$$

消元化简后可知 $\lambda = \theta$。送佛送到西，咱们来看看到底怎么消元：

$$
\begin{cases}
\boldsymbol{U}_k * (I) \rightarrow \boldsymbol{U}_k\boldsymbol{C_{XY_k}}{\boldsymbol{V}_k}^T - \lambda\boldsymbol{U}_k\boldsymbol{C_{XX}}{\boldsymbol{U}_k}^T = 0 \\
\boldsymbol{V}_k * (II) \rightarrow \boldsymbol{V}_k\boldsymbol{C_{Y_kX}}{\boldsymbol{U}_k}^T - \theta\boldsymbol{V}_k\boldsymbol{C_{Y_kY_k}}{\boldsymbol{V}_k}^T = 0
\end{cases}
\tag{1-1-7}
$$

根据约束条件 (1-1-4) 可知：

$$
\lambda = \boldsymbol{U}_k\boldsymbol{C_{XY_k}}{\boldsymbol{V}_k}^T,\ \theta = \boldsymbol{V}_k\boldsymbol{C_{Y_kX}}{\boldsymbol{U}_k}^T
\tag{1-1-8}
$$

大家注意 $\lambda = \theta^T$，而当我们明确要求优化目标是**一维向量**的时候，这两位大哥其实都是实数，所以它们相等。之后就是大家在解二元一次方程组时常用的代换消元过程（ $\boldsymbol{U}_k$ 与 $\boldsymbol{V}_k$ 互相替换），我就不再演示了。最终应得到两个特征值方程：

$$\begin{cases} {\boldsymbol{C}_{\boldsymbol{XX}}}^{-1}\boldsymbol{C}_{\boldsymbol{XY}_k}{\boldsymbol{C}_{\boldsymbol{Y}_k\boldsymbol{Y}_k}}^{-1}\boldsymbol{C}_{\boldsymbol{Y}_k\boldsymbol{X}}{\boldsymbol{U}_k}^T = \lambda^2{\boldsymbol{U}_k}^T \\ {\boldsymbol{C}_{\boldsymbol{Y}_k\boldsymbol{Y}_k}}^{-1}\boldsymbol{C}_{\boldsymbol{Y}_k\boldsymbol{X}}{\boldsymbol{C}_{\boldsymbol{XX}}}^{-1}\boldsymbol{C}_{\boldsymbol{XY}_k}{\boldsymbol{V}_k}^T = \theta^2{\boldsymbol{V}_k}^T \end{cases} \tag{1-1-9}$$

对式 (1-1-9) 中的两个 *Hermitte* 矩阵分别进行特征值分解，取最大特征值对应的特征向量作为投影向量，即为所求。对所有的假定类别遍历上述过程，基于一维 *Pearson* 相关系数分别计算判别系数并比较大小，确定最终的结果输出 $\hat{k}$：

$$\rho_k = corr\left(\hat{\boldsymbol{U}}_k\boldsymbol{X}, \hat{\boldsymbol{V}}_k\boldsymbol{Y}_k\right),\ \hat{k} = \arg\max_k\{\rho_k\} \tag{1-1-10}$$

式 (1-1-2) 与 (1-1-9) 请各位务必熟悉，之后部分算法的推导过程中我可能直接从类似前者的形式跳至后者，不再进行中间步骤的详细解说。

# 1.2 扩展 CCA：eCCA

**(Extended CCA)**

论文链接 | 代码：cca.ecca()

# 1.3 多重刺激 CCA：msCCA

**(Multi-stimulus CCA)**

论文链接 | 代码：cca.mscca()

**（大量数学前置知识警告）**

朋友们我们今天来膜拜 (~~gank~~) 澳门大学的内卷发动机 *Chi Man Wong* 和 *Feng Wan* 老师团队。之所以对他们“赞誉有加”，主要有三方面原因：

（1）**算法有用，但只有一点用**：他们提出的一系列 SSVEP 算法在公开数据集与竞赛数据集中具有极大优势（即样本量不足的情况）。不过在数据样本量充足的情况下，与传统的 (e)TRCA 算法难以拉开差距；

（2）**强悍如斯，地都快耕坏了**：他们的每篇论文都充分（~~往死里~~）挖掘了公开数据集的可用潜力，从 **Benchmark**、**UCSD** 再到 **BETA** 都能看到他的身影，从 CCA 到 ms-(e)TRCA 各种花里胡哨的算法都测了个遍（~~根本不给人活路~~），低频 SSVEP-BCI 系统的解码被他卷得翻江倒海，再无探索空间。

（3）**故弄玄虚，堆砌数学壁垒**：该团队 2020 年发表的一篇关于**空间滤波器构建框架**的综述性论文就是万恶之源。在其文章中，经常使用怪僻的希腊字母、花体英文字母作为变量名称，为了形式简约而把简单的实际操作过程复杂化。例如明明是人畜无害的试次叠加平均：

$$\bar{\boldsymbol{X}} = \frac{1}{N_t}\sum_{n=1}^{N_t}\boldsymbol{X}_i,\ \boldsymbol{X}_i \in \mathbb{R}^{Nc\times Np} \tag{1-3-1}$$

为了凑上自己提出的框架，硬要给你表演一套天书：

$$\oplus_{i=1}^{N_t}\boldsymbol{X}_i = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \cdots & \boldsymbol{X}_{N_t} \end{bmatrix},\ \bar{\boldsymbol{X}} = \frac{1}{N_t}\boldsymbol{\mathcal{I}}_{N_t,N_c}\cdot\left[\oplus_{i=1}^{N_t}\boldsymbol{X}_i\right]\cdot\boldsymbol{\mathcal{I}}_{N_t,N_p}^T \tag{1-3-2}$$

组里有些萌新，一板一眼地照着论文公式复现算法，结果训练样本一多，程序不仅运行慢，还动不动就内存溢出。从原始数据结构中创建 $\oplus_{i=1}^{N_t}\boldsymbol{X}_i$、$\boldsymbol{\mathcal{I}}_{N_t,N_p}$ 这样的大矩阵送去后端运算，相当于先做一把电锯去再杀鸡炖汤，能不慢吗？家庭厨房里接上工业用电，能不溢出吗？

不可否认的是，*Chi Man Wong* 及其团队对于 SSVEP 信号解码的研究是成体系的、步骤严谨的，他们提出的空间滤波器框架适配了自 CCA 以来的各种算法，为后续研究工作打开了思路。更重要的是，他们团队以及清华大学 *Yijun Wang*、*Xiaogang Chen* 等老师带领的团队，都不搞弯道超车，不搞非对称竞争，每一个研究思路都是建立在已有研究基础上，每一步新的收获都会切实体现为文章成果。这样的团队对于学生培养是利好的，学生不用担心梭哈一个老板异想天开的课题而愁于毕业困境。因此再让我跑题一次：**但凡遇到老板鼓吹自己手握多少项目、每年经费多少万、带领多少优秀青年教师团队、手下多少研究生之类的话术，一定要慎之又慎**。你要知道，牛逼吹得不够大是不能吸引上边的人投资的，牛逼吹起来了就是在梭哈你自己的学术生涯与宝贵光阴。老板项目结不了题顶多延期，少赚一点经费，少评一些名声，日子一分都不会难受。你延期延掉的是什么还请自己掂量清楚。

言归正传，我们首先有必要介绍一下 *Wong* 提出的统一框架。*Wong* 以及部分研究者喜欢按列展示向量，而本文中向量统一按行排布（~~我偏不，老子就喜欢按行~~），因此部分公式可能在形式上与原文有所出入，但本质是一样的：

$$\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{P}}^T\boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{Z}}^T\boldsymbol{W} = \begin{cases} \boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{Z}}^T\boldsymbol{W}\boldsymbol{\Lambda},\ TypeI \\ \\ \boldsymbol{W}\boldsymbol{\Lambda},\ TypeII \end{cases} \tag{1-3-3}$$

这里 $\boldsymbol{W}$ 与 $\boldsymbol{\Lambda}$ 之所以写成矩阵而不是“向量+标量”形式，是因为空间滤波器并不总是将多通道信号压缩至一维，对于需要进行压缩的一般情况，只需取方阵 $\boldsymbol{\mathcal{Z}\mathcal{D}\mathcal{P}\mathcal{P}^T\mathcal{D}^T\mathcal{Z}^T}$ 的最大（小）特征值对应的特征向量即可；而当需要保留多个维度（投影子空间）时，$\boldsymbol{W}$ 的最优解为多个特征向量的拼接，拼接顺序以对应特征向量的大小顺序为准。

接下来我不想再去复述他们文章中对各种算法的具体匹配方式，仅在此对式 (1-3-3) 中的主要成分进行简单介绍：

$\boldsymbol{\mathcal{Z}}$ 是数据（默认按列排布）的集合矩阵，可能是（1）单个数据矩阵；（2）形如 $\bigoplus_{i=1}^{N_t}\boldsymbol{X}_i$ 的多种数据块对角拼接组成的联合矩阵。一般来说（2）中的对角联合矩阵，在整体公式中需要经过 $\boldsymbol{\mathcal{I}}$ 矩阵的变形处理,将其转换为由多个数据块**横向**或**纵向**拼接而成的大矩阵，如式 (1-3-11) ；

$\boldsymbol{\mathcal{D}}$ 是时域滤波矩阵，除了滤波器组技术以外，通常预处理（带通滤波）结束后的数据无需再进行时域滤波，即 $\boldsymbol{\mathcal{D}=I}$；

$\boldsymbol{\mathcal{P}}$ 为正交投影矩阵，通常满足 $\boldsymbol{\mathcal{P}=\mathcal{P}^T=\mathcal{P}\mathcal{P}^T=\mathcal{P}^T\mathcal{P}}$。根据给定的投影方向（$\boldsymbol{T}$），可表示为:

$$\begin{cases}\boldsymbol{\mathcal{P}=T^T\left(TT^T\right)^{-1}T=Q_TQ_T}^T\\\boldsymbol{T=Q_TR_T},\ Reduced\ QR\ decomposition\end{cases}\tag{1-3-4}$$

不难发现，该框架的数学本质是一系列**广义特征值**（*Generalized eigenvalue problems, GEPs*）方程，而空间滤波器构建过程中常见的**广义瑞利商**（*Generalized Rayleigh quotient*）问题通常又可以转化为 *GEP* 方程加以求解，因此该框架几乎能够契合现有各种 SSVEP-BCI 系统中常见的空间滤波器算法。尽管如此，除了 *Wong* 设计的算法，我基本上不会使用这个框架来展示公式。原因除了之前吐槽过的“数学墙”以外，还有很重要的一点，即**凭空创造**、**设计**更好的时域滤波矩阵 $\boldsymbol{\mathcal{D}}$ 或正交投影矩阵 $\boldsymbol{\mathcal{P}}$ 都是不现实的，想要从数学上证明某个投影空间具有某些数学特性或优势都极具挑战性。我个人更倾向于通过**直观物理含义**的途径来阐释算法原理，希望通过我的讲解，能够让大家实现数学原理、编程实践与性能优化的三者合一，从而更好地掌握算法的精髓、洞察未来发展方向。

在本节前置数学知识的最后，给大家简单介绍一下广义瑞利商及其与 *GEP* 问题的关系。形如式 (1-3-5) 所示的函数称为瑞利商（*Rayleigh quotient*），其中 $\boldsymbol{A}$ 为 *Hermitte* 矩阵：

$$f(\boldsymbol{\omega})=\frac{\boldsymbol{\omega A\omega}^T}{\boldsymbol{\omega\omega}^T},\ \boldsymbol{A}\in\mathbb{R}^{N\times N},\ \boldsymbol{\omega}\in\mathbb{R}^{1\times N}\tag{1-3-5}$$

一般最优化问题需要求解的是瑞利商的最值，参考式 (1-1-4) 的方式，将式 (1-3-5) 转化为最优化问题的标准描述形式，之后利用 *Lagrandian* 乘子法构建函数 $\boldsymbol{J}$：

$$\begin{cases} \max\limits_{\boldsymbol{\omega}} \boldsymbol{\omega}\boldsymbol{A}\boldsymbol{\omega}^T \\ s.t.\ \boldsymbol{\omega}\boldsymbol{\omega}^T = 1 \end{cases} \Longrightarrow J(\boldsymbol{\omega}) = \boldsymbol{\omega}\boldsymbol{A}\boldsymbol{\omega}^T - \lambda\left(\boldsymbol{\omega}\boldsymbol{\omega}^T - 1\right) \tag{1-3-6}$$

对 $J$ 求导并置零，最终可得特征值方程：

$$\frac{dJ(\boldsymbol{\omega})}{d\boldsymbol{\omega}} = 2\boldsymbol{A}\boldsymbol{\omega}^T - 2\lambda\boldsymbol{\omega}^T = 0 \rightarrow \boldsymbol{A}\boldsymbol{\omega}^T = \lambda\boldsymbol{\omega}^T \tag{1-3-7}$$

至此可以看出，瑞利商的最值即为方阵 $\boldsymbol{A}$ 最大（小）特征值对应的特征向量。至于广义瑞利商，其形如式 (1-3-8) 所示的函数，$\boldsymbol{B}$ 同样也是 *Hermitte* 矩阵：

$$f(\boldsymbol{\omega}) = \frac{\boldsymbol{\omega}\boldsymbol{A}\boldsymbol{\omega}^T}{\boldsymbol{\omega}\boldsymbol{B}\boldsymbol{\omega}^T},\ \boldsymbol{A},\boldsymbol{B} \in \mathbb{R}^{N\times N},\ \boldsymbol{\omega} \in \mathbb{R}^{1\times N} \tag{1-3-8}$$

同上进行类似操作，可以得到式 (1-3-9) 所示的 *GEP* 方程。由此可知广义瑞利商的最值即为方阵 $\boldsymbol{B}^{-1}\boldsymbol{A}$ 最大（小）特征值对应的特征向量：

$$\boldsymbol{A}\boldsymbol{\omega}^T = \lambda\boldsymbol{B}\boldsymbol{\omega}^T \Longrightarrow \left(\boldsymbol{B}^{-1}\boldsymbol{A}\right)\boldsymbol{\omega}^T = \lambda\boldsymbol{\omega}^T \tag{1-3-9}$$

接下来我们来看 msCCA 算法。首先给出统一框架（**Type I**）下的各部分组成：

$$\begin{cases} \boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{I}}_{N_e,N_c}\left(\oplus_{k=1}^{N_e}\bar{\boldsymbol{X}}_k\right) \in \mathbb{R}^{N_c\times(N_eN_p)} \\ \boldsymbol{\mathcal{D}} = \boldsymbol{I}_{N_eN_p} \in \mathbb{R}^{(N_eN_p)\times(N_eN_p)} \\ \boldsymbol{\mathcal{P}} = \boldsymbol{Q}_{\boldsymbol{\mathcal{Y}}}\boldsymbol{Q}_{\boldsymbol{\mathcal{Y}}}{}^T =\in \mathbb{R}^{(N_eN_p)\times(N_eN_p)} \end{cases} \tag{1-3-10}$$

其中：

$$\boldsymbol{\mathcal{Y}} = \begin{bmatrix} \boldsymbol{Y}_1 & \boldsymbol{Y}_2 & \cdots & \boldsymbol{Y}_{N_e} \end{bmatrix} \in \mathbb{R}^{(2N_h)\times(N_eN_p)} \tag{1-3-11}$$

$$\boldsymbol{Q}_{\boldsymbol{\mathcal{Y}}}\boldsymbol{Q}_{\boldsymbol{\mathcal{Y}}}{}^T = \begin{bmatrix} \boldsymbol{Q}_{\boldsymbol{Y}_1}\boldsymbol{Q}_{\boldsymbol{Y}_1}{}^T & \boldsymbol{Q}_{\boldsymbol{Y}_1}\boldsymbol{Q}_{\boldsymbol{Y}_2}{}^T & \cdots & \boldsymbol{Q}_{\boldsymbol{Y}_1}\boldsymbol{Q}_{\boldsymbol{Y}_{N_e}}{}^T \\ \boldsymbol{Q}_{\boldsymbol{Y}_2}\boldsymbol{Q}_{\boldsymbol{Y}_1}{}^T & \boldsymbol{Q}_{\boldsymbol{Y}_2}\boldsymbol{Q}_{\boldsymbol{Y}_2}{}^T & \cdots & \boldsymbol{Q}_{\boldsymbol{Y}_2}\boldsymbol{Q}_{\boldsymbol{Y}_{N_e}}{}^T \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Q}_{\boldsymbol{Y}_{N_e}}\boldsymbol{Q}_{\boldsymbol{Y}_1}{}^T & \boldsymbol{Q}_{\boldsymbol{Y}_{N_e}}\boldsymbol{Q}_{\boldsymbol{Y}_2}{}^T & \cdots & \boldsymbol{Q}_{\boldsymbol{Y}_{N_e}}\boldsymbol{Q}_{\boldsymbol{Y}_{N_e}}{}^T \end{bmatrix} \tag{1-3-12}$$

不要被这些花里胡哨的公式迷乱了双眼，我们来看看每一步都发生了什么：

$$\boldsymbol{\mathcal{Z}} = \underbrace{\begin{bmatrix} \boldsymbol{I}_{N_c} & \boldsymbol{I}_{N_c} & \cdots & \boldsymbol{I}_{N_c} \end{bmatrix}}_{\mathbb{R}^{N_c \times (N_e N_c)}} \underbrace{\begin{bmatrix} \bar{\boldsymbol{X}}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{X}}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \bar{\boldsymbol{X}}_{N_e} \end{bmatrix}}_{\mathbb{R}^{(N_e N_c) \times (N_e N_p)}} = \underbrace{\begin{bmatrix} \bar{\boldsymbol{X}}_1 & \bar{\boldsymbol{X}}_2 & \cdots & \bar{\boldsymbol{X}}_{N_e} \end{bmatrix}}_{\mathbb{R}^{N_c \times (N_e N_p)}} \tag{1-3-13}$$

$$\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}} = \underbrace{\begin{bmatrix} \bar{\boldsymbol{X}}_1 & \bar{\boldsymbol{X}}_2 & \cdots & \bar{\boldsymbol{X}}_{N_e} \end{bmatrix}}_{\mathbb{R}^{N_c \times (N_e N_p)}} \underbrace{\begin{bmatrix} \boldsymbol{I}_{N_p} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I}_{N_p} \end{bmatrix}}_{\mathbb{R}^{(N_e N_p) \times (N_e N_p)}} = \boldsymbol{\mathcal{Z}} \tag{1-3-14}$$

$$\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{P}} = \underbrace{\begin{bmatrix} \sum_{k=1}^{N_e} \bar{\boldsymbol{X}}_k \boldsymbol{Q}_{\boldsymbol{Y}_k} \boldsymbol{Q}_{\boldsymbol{Y}_1}{}^T & \cdots & \sum_{k=1}^{N_e} \bar{\boldsymbol{X}}_k \boldsymbol{Q}_{\boldsymbol{Y}_k} \boldsymbol{Q}_{\boldsymbol{Y}_{N_e}}{}^T \end{bmatrix}}_{\mathbb{R}^{N_c \times (N_e N_p)}} \tag{1-3-15}$$

$$\begin{aligned} \boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{P}}^T\boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{Z}}^T &= \sum_{c=1}^{N_e} \left[ \sum_{a=1}^{N_e} \bar{\boldsymbol{X}}_a \boldsymbol{Q}_{\boldsymbol{Y}_a} \boldsymbol{Q}_{\boldsymbol{Y}_c}{}^T \left( \sum_{b=1}^{N_e} \bar{\boldsymbol{X}}_b \boldsymbol{Q}_{\boldsymbol{Y}_b} \boldsymbol{Q}_{\boldsymbol{Y}_c}{}^T \right)^T \right] \\ &\xrightarrow{\boldsymbol{Q}_{\boldsymbol{Y}_c}{}^T \boldsymbol{Q}_{\boldsymbol{Y}_c} = \boldsymbol{I}} N_e \sum_{b=1}^{N_e} \sum_{a=1}^{N_e} \bar{\boldsymbol{X}}_a \boldsymbol{Q}_{\boldsymbol{Y}_a} \boldsymbol{Q}_{\boldsymbol{Y}_b}{}^T \bar{\boldsymbol{X}}_b \end{aligned} \tag{1-3-16}$$

$$\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{Z}}^T = \boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{Z}}^T = \underbrace{\begin{bmatrix} \bar{\boldsymbol{X}}_1 & \bar{\boldsymbol{X}}_2 & \cdots & \bar{\boldsymbol{X}}_{N_e} \end{bmatrix}}_{\mathbb{R}^{N_c \times (N_e N_p)}} \underbrace{\begin{bmatrix} \bar{\boldsymbol{X}}_1 \\ \bar{\boldsymbol{X}}_2 \\ \vdots \\ \bar{\boldsymbol{X}}_{N_e} \end{bmatrix}}_{\mathbb{R}^{(N_e N_p) \times N_c}} = \sum_{k=1}^{N_e} \bar{\boldsymbol{X}}_k \bar{\boldsymbol{X}}_k{}^T \tag{1-3-17}$$

可真是费了好一番力气才完成。最终 *GEP* 方程为（仅需一维投影向量）：

$$\left(N_e\sum_{b=1}^{N_e}\sum_{a=1}^{N_e}\bar{\boldsymbol{X}}_a\boldsymbol{Q}_{\boldsymbol{Y}_a}{\boldsymbol{Q}_{\boldsymbol{Y}_b}}^T{\bar{\boldsymbol{X}}_b}^T\right)\boldsymbol{\omega}^T=\lambda\left(\sum_{k=1}^{N_e}\bar{\boldsymbol{X}}_k{\bar{\boldsymbol{X}}_k}^T\right)\boldsymbol{\omega}^T \tag{1-3-18}$$

相信各位很快就会发现，这个公式没有为我们提供任何直白的、一般人能阅读的有效信息。坦白地说，仅靠式 (1-3-18) 设计的滤波器以及相应的模板匹配方法是不完整的，具体原因请各位移步下一节 ms-eCCA，我们将从另一个更合理的角度审视这个算法。

不得不说 *Wong* 这一手阉割刀法堪比老黄，先在 *JNE* 上发表精心推导设计的两大 ms- 算法，再在 *IEEE TBME* 上发表统一框架，顺带蜻蜓点水一般地，用框架小增小改就套出了这个丐版 msCCA，还比老旧的 itCCA 强上一截，以此彰显框架的“易用性”，其实根本没这么浅显，套模型套框架也并非研学之道。

# 1.4 多重刺激扩展 CCA：ms-eCCA

**(Multi-stimulus CCA)**

**论文链接 | 代码：cca.msecca()**

顾名思义，ms-eCCA 是 msCCA 的扩展，在模板匹配上类似 eCCA 之于 CCA，此外在扩增的刺激目标选择上有所约束。阅读本节内容之前，建议阅读 3.2 节以了解部分背景知识，因为本节将从另外一个角度解释ms-eCCA 公式的由来。ms-eCCA 与 ms-eTRCA 是 *Wong* 在同一篇论文中提出的方法，在发表时间上也先于 msCCA。与 3.2 节所述一致，ms- 技术假定一套滤波器同时适用于目标频率以及周边少数频率的信号，通过合并一定范围内的多类别信号，强行扩增可用训练样本数目。

“合并样本”的操作，既可以理解为**不同类别样本模板在时间顺序上的拼接**，见式(1-4-2)；也可以理解为**不同类别样本各自协方差矩阵的叠加**，见式 (1-4-4)。结合 3.2 节 ms-(e)TRCA 与 1.1 节 CCA 的相关内容，ms-eCCA 的目标函数可表示为：

$$\hat{\boldsymbol{U}}_k,\hat{\boldsymbol{V}}_k=\underset{\boldsymbol{U}_k,\boldsymbol{V}_k}{\arg\max}\frac{Cov\left(\boldsymbol{U}_k\boldsymbol{\mathcal{Z}}_k,\boldsymbol{\mathcal{Y}}_k\boldsymbol{V}_k\right)}{\sqrt{Var\left(\boldsymbol{U}_k\boldsymbol{\mathcal{Z}}_k\right)}\sqrt{Var\left(\boldsymbol{V}_k\boldsymbol{\mathcal{Y}}_k\right)}}=\underset{\boldsymbol{U}_k,\boldsymbol{V}_k}{\arg\max}\frac{\boldsymbol{U}_k\boldsymbol{C}_{\boldsymbol{\mathcal{Z}}_k\boldsymbol{\mathcal{Y}}_k}{\boldsymbol{V}_k}^T}{\sqrt{\left(\boldsymbol{U}_k\boldsymbol{C}_{\boldsymbol{\mathcal{Z}}_k\boldsymbol{\mathcal{Z}}_k}{\boldsymbol{U}_k}^T\right)}\sqrt{\left(\boldsymbol{V}_k\boldsymbol{C}_{\boldsymbol{\mathcal{Y}}_k\boldsymbol{\mathcal{Y}}_k}{\boldsymbol{V}_k}^T\right)}} \tag{1-4-1}$$

$$\begin{cases}\boldsymbol{\mathcal{Z}}_k=\begin{bmatrix}\bar{\boldsymbol{X}}_{k-m} & \bar{\boldsymbol{X}}_{k-m+1} & \cdots & \bar{\boldsymbol{X}}_{k+n}\end{bmatrix}\in\mathbb{R}^{N_c\times[(m+n+1)N_p]}\\ \boldsymbol{\mathcal{Y}}_k=\begin{bmatrix}\boldsymbol{Y}_{k-m} & \boldsymbol{Y}_{k-m+1} & \cdots & \boldsymbol{Y}_{k+n}\end{bmatrix}\in\mathbb{R}^{N_c\times[(m+n+1)N_tN_p]}\end{cases} \tag{1-4-2}$$

$$\boldsymbol{Y}_k=\begin{bmatrix}\sin(2\pi f_k\boldsymbol{t}+\phi_k)\\\cos(2\pi f_k\boldsymbol{t}+\phi_k)\\\vdots\\\sin(2\pi N_h f_k\boldsymbol{t}+N_h\phi_k)\\\cos(2\pi N_h f_k\boldsymbol{t}+N_h\phi_k)\end{bmatrix}\in\mathbb{R}^{(2N_h)\times N_p},\ \boldsymbol{t}=\begin{bmatrix}\frac{1}{f_s}&\frac{2}{f_s}&\cdots&\frac{N_p}{f_s}\end{bmatrix}\tag{1-4-3}$$

$$\begin{cases}\boldsymbol{C}_{\boldsymbol{Z}_k\boldsymbol{Z}_k}=\dfrac{1}{N_p-1}\sum_{i=-n}^{m}\bar{\boldsymbol{X}}_{k+i}{\bar{\boldsymbol{X}}_{k+i}}^T\in\mathbb{R}^{N_c\times N_c}\\\boldsymbol{C}_{\boldsymbol{y}_k\boldsymbol{y}_k}=\dfrac{1}{N_p-1}\sum_{i=-n}^{m}\boldsymbol{Y}_{k+i}{\boldsymbol{Y}_{k+i}}^T\in\mathbb{R}^{(2N_h)\times(2N_h)}\\\boldsymbol{C}_{\boldsymbol{Z}_k\boldsymbol{y}_k}=\dfrac{1}{N_p-1}\sum_{i=-n}^{m}\bar{\boldsymbol{X}}_{k+i}{\boldsymbol{Y}_{k+i}}^T\in\mathbb{R}^{N_c\times(2N_h)}\\\boldsymbol{C}_{\boldsymbol{y}_k\boldsymbol{Z}_k}=\dfrac{1}{N_p-1}\sum_{i=-n}^{m}\boldsymbol{Y}_{k+i}{\bar{\boldsymbol{X}}_{k+i}}^T\in\mathbb{R}^{(2N_h)\times N_c}\end{cases}\tag{1-4-4}$$

简而言之，在空间滤波器构建上，ms-eCCA 与 ms-(e)TRCA 的思路一致，即把不同类别信号按时间维度顺次拼接，以起到数据扩增的作用。类比 1.1 节推导过程可知，式 (1-4-1) 所对应的两个 *GEP* 方程为：

$$\begin{cases}{\boldsymbol{C}_{\boldsymbol{Z}_k\boldsymbol{Z}_k}}^{-1}\boldsymbol{C}_{\boldsymbol{Z}_k\boldsymbol{y}_k}{\boldsymbol{C}_{\boldsymbol{y}_k\boldsymbol{y}_k}}^{-1}\boldsymbol{C}_{\boldsymbol{y}_k\boldsymbol{Z}_k}\boldsymbol{U}_k=\lambda^2{\boldsymbol{U}_k}^T\quad(I)\\{\boldsymbol{C}_{\boldsymbol{y}_k\boldsymbol{y}_k}}^{-1}\boldsymbol{C}_{\boldsymbol{y}_k\boldsymbol{Z}_k}{\boldsymbol{C}_{\boldsymbol{Z}_k\boldsymbol{Z}_k}}^{-1}\boldsymbol{C}_{\boldsymbol{Z}_k\boldsymbol{y}_k}\boldsymbol{V}_k=\theta^2{\boldsymbol{V}_k}^T\quad(II)\end{cases}\tag{1-4-5}$$

首先来看方程组中的 $(I)$：

$$\left(\sum_k\bar{\boldsymbol{X}}_k{\bar{\boldsymbol{X}}_k}^T\right)^{-1}\left(\sum_a\bar{\boldsymbol{X}}_a{\boldsymbol{Y}_a}^T\right)\left(\sum_c\boldsymbol{Y}_c{\boldsymbol{Y}_c}^T\right)^{-1}\left(\sum_b\boldsymbol{Y}_b{\bar{\boldsymbol{X}}_b}^T\right){\boldsymbol{U}_k}^T=\lambda^2{\boldsymbol{U}_k}^T$$

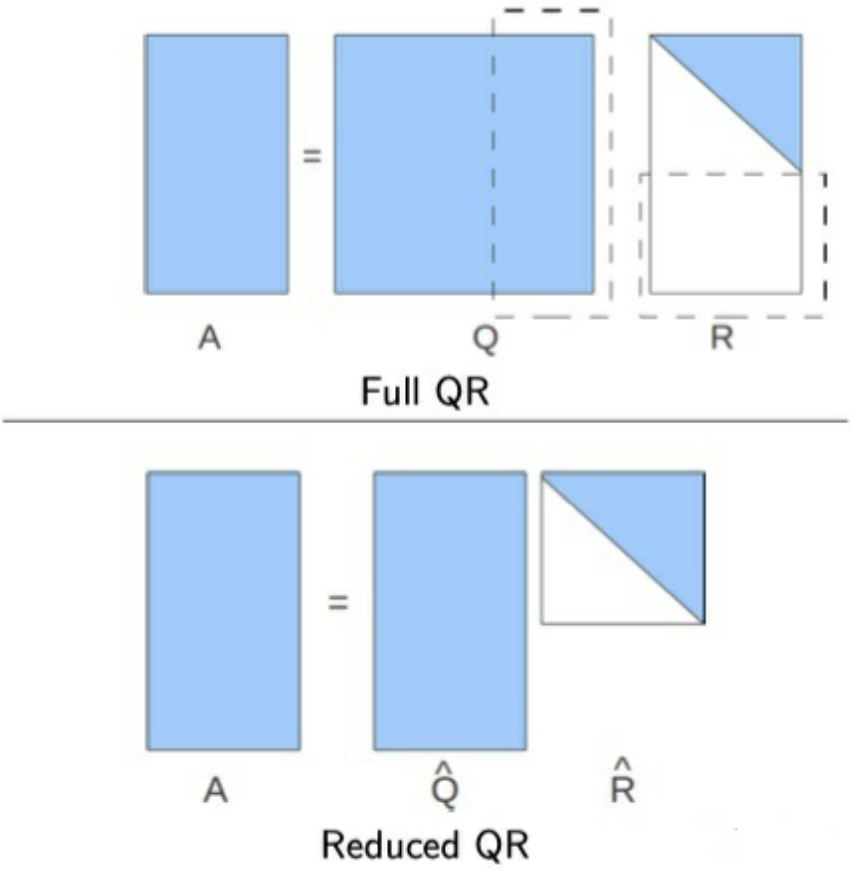
$$\Longrightarrow\left(\sum_a\bar{\boldsymbol{X}}_a{\boldsymbol{Y}_a}^T\right)\left(\sum_c\boldsymbol{Y}_c{\boldsymbol{Y}_c}^T\right)^{-1}\left(\sum_b\boldsymbol{Y}_b{\bar{\boldsymbol{X}}_b}^T\right){\boldsymbol{U}_k}^T=\lambda^2\left(\sum_k\bar{\boldsymbol{X}}_k{\bar{\boldsymbol{X}}_k}^T\right){\boldsymbol{U}_k}^T$$

$$\Longrightarrow\left[\sum_b\sum_a\bar{\boldsymbol{X}}_a{\boldsymbol{Y}_a}^T\left(\sum_c\boldsymbol{Y}_c{\boldsymbol{Y}_c}^T\right)^{-1}\boldsymbol{Y}_b{\bar{\boldsymbol{X}}_b}^T\right]{\boldsymbol{U}_k}^T=\lambda^2\left(\sum_k\bar{\boldsymbol{X}}_k{\bar{\boldsymbol{X}}_k}^T\right){\boldsymbol{U}_k}^T\tag{1-4-6}$$

到这一步我们不难发现，其结果已经与上一节末尾的式 (1-3-18) 形式非常相似，仅剩的问题在于正交投影

$\boldsymbol{Q}_k\boldsymbol{Q}_k^T$ 与正余弦矩阵 $\boldsymbol{Y}_k$ 之间的关系。

因此我们有必要讲解（~~插播~~）一下何为 QR 分解（*QR decomposition*）。QR 分解又称正交三角分解，通常用于求解矩阵的特征值与特征向量。QR 分解的作用是将实（复）非奇异矩阵 $\boldsymbol{A}$ 转化为正交（酉）矩阵 $\boldsymbol{Q}$ 与实（复）非奇异上三角矩阵 $\boldsymbol{R}$ 的乘积。从操作结果来看，QR 矩阵分为全分解（*full decomposition*）与约化分解（*reduced decomposition*）两种，二者的差异见下图：




一般情况下，我们需要的是约化 QR 分解结果。计算方法有很多，这里给出其中一种方便理解与实操的方法——*Gram-Schmidt* 正交化。首先我们获取矩阵 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ 的列向量组 $\boldsymbol{A}(:,j)$，之后单位化 $\boldsymbol{A}(:,0)$，并将其作为第一个正交基，对后续列向量依次进行投影分解：

$$\begin{cases}\boldsymbol{A}(:,1)=r_{1,1}\boldsymbol{q}_1\\ \boldsymbol{A}(:,2)=r_{2,1}\boldsymbol{q}_1+r_{2,2}\boldsymbol{q}_2\\ \cdots\\ \boldsymbol{A}(:,n)=r_{n,1}\boldsymbol{q}_1+r_{n,2}\boldsymbol{q}_2+\cdots+r_{n,n}\boldsymbol{q}_n\end{cases}\tag{1-4-7}$$

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_n\end{bmatrix}\begin{bmatrix}r_{1,1} & r_{1,2} & \cdots & r_{1,n}\\ 0 & r_{2,2} & \cdots & r_{2,n}\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & r_{n,n}\end{bmatrix}=\boldsymbol{Q_A}\boldsymbol{R_A}\tag{1-4-8}$$

我们在 -R 技术、ms- 技术乃至未来将更新的 TDCA 算法中都经常见到正交投影 $\boldsymbol{\mathcal{P}}_k = \boldsymbol{Q}_k \boldsymbol{Q}_k{}^T$ 的身影，我们来仔细研究一下这个矩阵。已知 $\boldsymbol{Y}_k{}^T = \boldsymbol{Q}_k \boldsymbol{R}_k$，对照式 (1-4-8) 不难看出：

（1）$\boldsymbol{Q}_k$ 本质上就是 $\boldsymbol{Y}_k{}^T$，二者在数值上存在一定的系数比例。因为对于正余弦模板矩阵而言，其列向量本来就是正交的，只是不满足单位化而已；

（2）$\boldsymbol{R}_k$ 是个对角阵，主对角线上的系数绝对值均相等，且为 $\boldsymbol{Y}_k{}^T$ 中任一列向量的**内积平方根**，$\boldsymbol{R}_k$ 的唯一作用就是将 $\boldsymbol{Q}_k$ 压缩至单位化水平。这一点很容易证明：

$$\boldsymbol{Y}_k{}^T \boldsymbol{Y}_k = \boldsymbol{Q}_k \left( \boldsymbol{R}_k \boldsymbol{R}_k{}^T \right) \boldsymbol{Q}_k{}^T = {r_{1,1}}^2 \boldsymbol{Q}_k \boldsymbol{Q}_k{}^T \tag{1-4-9}$$

又因为式 (1-3-4) 指出，$\boldsymbol{\mathcal{P}} = \boldsymbol{T}^T \left( \boldsymbol{T} \boldsymbol{T}^T \right)^{-1} \boldsymbol{T} = \boldsymbol{Q_T} \boldsymbol{Q_T}{}^T$，我们知道 $\boldsymbol{Q}_k \boldsymbol{Q}_k{}^T = \boldsymbol{Y}_k{}^T \left( \boldsymbol{Y}_k \boldsymbol{Y}_k{}^T \right)^{-1} \boldsymbol{Y}_k$，接下来我们证明这个矩阵能够起到正交投影的作用。

关于正交投影，从泛函的角度而言可以给出如下定义：令 $\boldsymbol{H}$ 为向量空间，$\boldsymbol{M}$ 是 $\boldsymbol{H}$ 内的 $n$ 维子空间。若对于 $\boldsymbol{H}$ 中的向量 $\boldsymbol{x}$：

$$\exists\ \hat{\boldsymbol{x}} \in \boldsymbol{M},\quad s.t. \forall\ \boldsymbol{y} \in \boldsymbol{M},\ \langle \boldsymbol{x} - \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle = 0 \tag{1-4-10}$$

则称 $\hat{\boldsymbol{x}}$ 是 $\boldsymbol{x}$ 在子空间 $\boldsymbol{M}$ 上的投影。对于某一频率、各次谐波的正余弦信号张成的向量空间 $\boldsymbol{Y}_k$，矩阵中每一个行向量都是该空间内的一个向量。我们来看投影过程：

$$\begin{gathered} \boldsymbol{X} \boldsymbol{Y}_k{}^T \left( \boldsymbol{Y}_k \boldsymbol{Y}_k{}^T \right)^{-1} \boldsymbol{Y}_k = \hat{\boldsymbol{X}} \implies \boldsymbol{X} \boldsymbol{Y}_k{}^T = \hat{\boldsymbol{X}} \boldsymbol{Y}_k{}^T \\ \therefore \left( \boldsymbol{X} - \hat{\boldsymbol{X}} \right) \boldsymbol{Y}_k{}^T = 0 \implies \left\langle \boldsymbol{X} - \hat{\boldsymbol{X}}, \boldsymbol{Y}_k \right\rangle = 0 \end{gathered} \tag{1-4-11}$$

由上可知，经过投影后的信号 $\hat{\boldsymbol{X}}$ 位于向量空间 $\boldsymbol{Y}_k$ 内，即纯化了 EEG 信号中与正余弦刺激相关的成分，尽管这个成分也许并不能完全覆盖刺激诱发信号的共性或个性特征。结合此前关于 QR 分解的描述，我们知道 $\boldsymbol{Q}_k \boldsymbol{Q}_k{}^T$ 与 $\boldsymbol{Y}_k{}^T \boldsymbol{Y}_k$ 的差别仅在于一个数值平衡，起到该效果的部分显然就是 $\left( \boldsymbol{Y}_k \boldsymbol{Y}_k{}^T \right)^{-1}$ 了：

$$\langle \boldsymbol{Y}_k(i,:), \boldsymbol{Y}_k(i,:) \rangle = \| \boldsymbol{Y}_k(i,:) \|_2^2$$

$$\| \boldsymbol{Y}_k(i,:) \|_2^2 = \| \boldsymbol{Y}_k(j,:) \|_2^2,\ \forall i, j \in [1, 2N_h]$$

$$\left( \boldsymbol{Y}_k \boldsymbol{Y}_k{}^T \right)^{-1} = \begin{bmatrix} \dfrac{1}{\| \boldsymbol{Y}_k(1,:) \|_2^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \dfrac{1}{\| \boldsymbol{Y}_k(2N_h,:) \|_2^2} \end{bmatrix} = \frac{\boldsymbol{I}_{2N_h}}{\| \boldsymbol{Y}_k(i,:) \|_2^2} \tag{1-4-12}$$

这里需要额外指出，$\left\|\boldsymbol{Y}_k(i,:)\right\|_2^2$ 的数值与**正余弦信号的频率或相位**是无关的（这一点可以自行编程测试，不再证明），且人工构建正余弦信号的峰峰值均为 2（即幅值相等），因此该范数只与**数据长度**有关。换句话说在同一批次数据中，对任意 $k$ 该范数均相等：

$$\left(\sum_c \boldsymbol{Y}_c {\boldsymbol{Y}_c}^T\right)^{-1} = \frac{\boldsymbol{I}_{2N_h}}{N_e \left\|\boldsymbol{Y}_k(i,:)\right\|_2^2} \tag{1-4-13}$$

因此，式 (1-4-6) 可简化为：

$$\begin{aligned} &\frac{1}{N_e}\left[\sum_b\sum_a \bar{\boldsymbol{X}}_a \frac{{\boldsymbol{Y}_a}^T}{\left\|\boldsymbol{Y}_k(i,:)\right\|_2}\frac{\boldsymbol{Y}_b}{\left\|\boldsymbol{Y}_k(i,:)\right\|_2}{\bar{\boldsymbol{X}}_b}^T\right]{\boldsymbol{U}_k}^T = \lambda^2\left(\sum_k \bar{\boldsymbol{X}}_k{\bar{\boldsymbol{X}}_k}^T\right){\boldsymbol{U}_k}^T \\ \Longrightarrow\ &\left(\frac{1}{N_e}\sum_b\sum_a \bar{\boldsymbol{X}}_a \boldsymbol{Q}_{\boldsymbol{Y}_a}{\boldsymbol{Q}_{\boldsymbol{Y}_b}}^T{\bar{\boldsymbol{X}}_b}^T\right){\boldsymbol{U}_k}^T = \lambda^2\left(\sum_k \bar{\boldsymbol{X}}_k{\bar{\boldsymbol{X}}_k}^T\right){\boldsymbol{U}_k}^T \end{aligned} \tag{1-4-14}$$

这样我们就来到了式 (1-3-18)。条条大路通罗马，不是吗？但比起式 (1-4-14) 这种惊险刺激的过山车，我个人还是更喜欢式 (1-4-5) 这种平静祥和的旅途。

我们可以发现，上一节中提到的 msCCA 只是式 (1-4-5) 中的 $(I)$。虽然满足以下两个条件（1）有滤波器（降维手段）（2）有训练数据，理论上就已经可以进行模板匹配了，但是这种设计思维背后的目标函数并不止于此。这也是通过统一框架无法了解的信息，很容易被遗漏。直白一点地说，msCCA 的训练目标应当是为训练数据与正余弦信号（二者均为合并样本）各自寻找一组投影向量以满足后续需求，而不是单方面优化训练数据。结合上述关于投影矩阵相关的说明，我们可以剔除 ${\boldsymbol{C}_{\boldsymbol{y}_k\boldsymbol{y}_k}}^{-1}$ 以进一步简化 $(I)$：

$${\boldsymbol{C}_{\boldsymbol{z}_k\boldsymbol{z}_k}}^{-1}\boldsymbol{C}_{\boldsymbol{z}_k\boldsymbol{y}_k}\boldsymbol{C}_{\boldsymbol{y}_k\boldsymbol{z}_k}\boldsymbol{U}_k = \lambda^2{\boldsymbol{U}_k}^T \tag{1-4-15}$$

我们来看看这个 *GEP* 方程对应的广义瑞利商形式：

$$\hat{\boldsymbol{U}}_k = \arg\max_{\boldsymbol{U}_k}\frac{\boldsymbol{U}_k\boldsymbol{C}_{\boldsymbol{z}_k\boldsymbol{y}_k}\boldsymbol{C}_{\boldsymbol{y}_k\boldsymbol{z}_k}{\boldsymbol{U}_k}^T}{\boldsymbol{U}_k\left(\sum_k \bar{\boldsymbol{X}}_k{\bar{\boldsymbol{X}}_k}^T\right){\boldsymbol{U}_k}^T} \tag{1-4-16}$$

这个方程的分母是**滤波后合并样本的能量**，但是分子就很有意思了，显然 $\boldsymbol{U}_k\boldsymbol{C}_{\boldsymbol{z}_k\boldsymbol{y}_k}$ 并不能表示某种信号，因为它的长度已经与采样信号不一样了，这一点在所有 CCA 系列算法中其实都存在。目前我还无法给它一个合理的物理解释，如果各位观众姥爷有高见，希望邮件联系我指点迷津。$(I)$ 的问题到此为止了，可 $(II)$ 真不是一位善茬，不论如何都无法缩减运算量，只得就此作罢。最后总结一下 ms-eCCA 的空间滤波器构建函数（*GEP* 方程）以及模板匹配所需的步骤：

$$\begin{cases} \boldsymbol{C_{z_k z_k}}^{-1} \boldsymbol{C_{z_k y_k}} \boldsymbol{C_{y_k z_k}} \boldsymbol{U}_k = \lambda^2 \boldsymbol{U_k}^T \quad (I) \\ \boldsymbol{C_{y_k y_k}}^{-1} \boldsymbol{C_{y_k z_k}} \boldsymbol{C_{z_k z_k}}^{-1} \boldsymbol{C_{z_k y_k}} \boldsymbol{V}_k = \theta^2 \boldsymbol{V_k}^T \quad (II) \end{cases} \tag{1-4-17}$$

$$\rho_k = \sum_{i=1}^{2} sign\left(\rho_{k,i}\right) \times \rho_{k,i}^2, \begin{cases} \rho_{k,1} = corr\left(\boldsymbol{U}_k \boldsymbol{\chi}, \boldsymbol{V}_k \boldsymbol{Y}_k\right) \\ \rho_{k,2} = corr\left(\boldsymbol{U}_k \boldsymbol{\chi}, \boldsymbol{U}_k \bar{\boldsymbol{X}}_k\right) \end{cases} \tag{1-4-18}$$

# 1.x 跨个体空间滤波器迁移：CSSFT

**(Cross-subject spatial filter transfer method)**

论文链接 | 代码：**cca**.cssft()

---

# 2. 多变量同步化系数

**Multivariate synchronization index, MSI**

## 2.1 标准 MSI：MSI

论文链接 | 代码：**msi**.msi()

## 2.2 时域局部 MSI：tMSI

**(Temporally MSI)**

论文链接 | 代码：**msi**.tmsi()

## 2.3 扩展 MSI：eMSI

**(Extended MSI)**

论文链接 | 代码：**msi**.emsi()

# 3. 任务相关成分分析

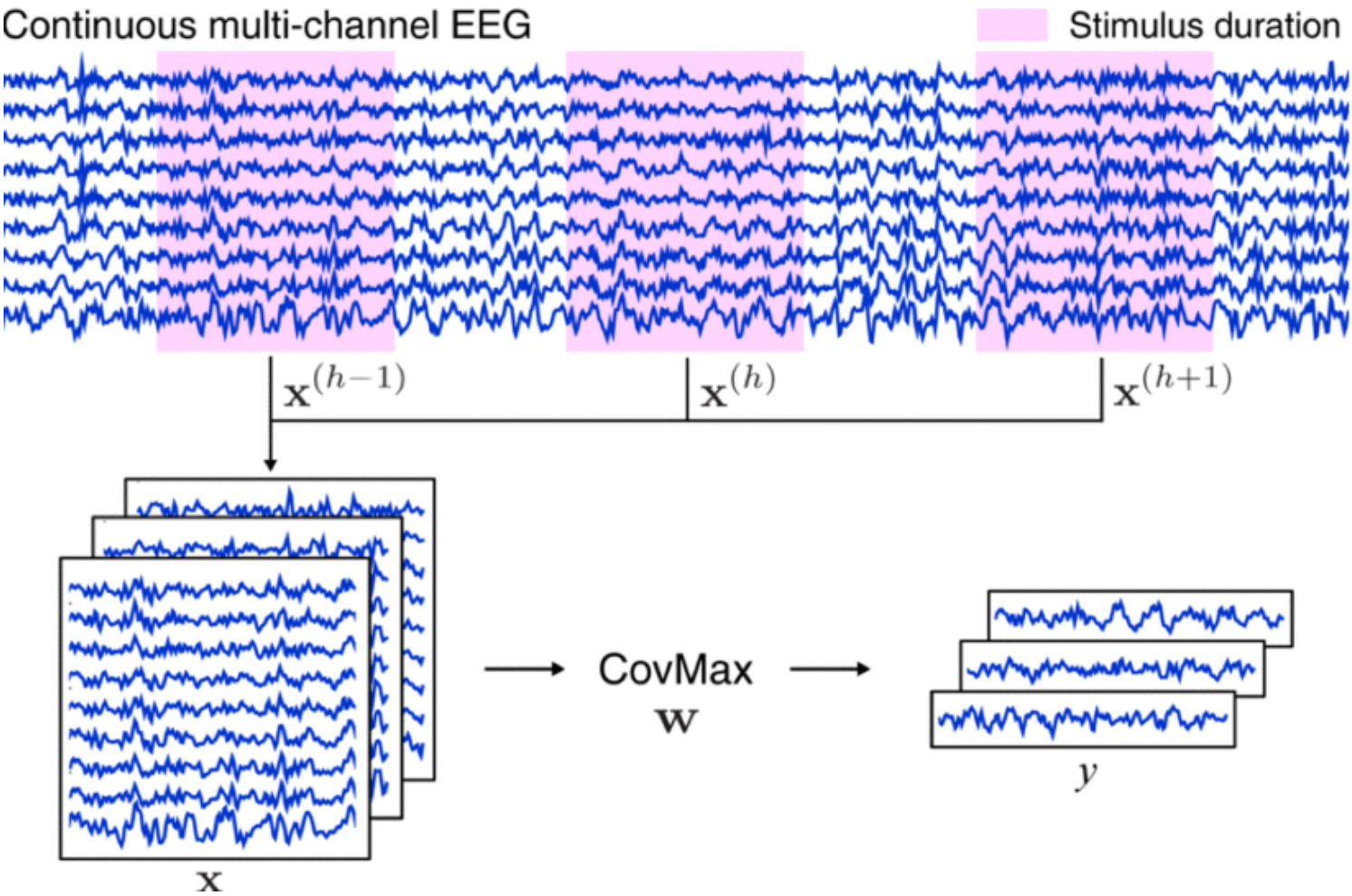
**Task-related component analysis, TRCA**

## 3.1 普通/集成 TRCA：(e)TRCA

**( (Ensemble) TRCA, (e)TRCA )**

论文链接 | 代码：trca.etrca()

与此前基于 CCA 改进的 SSVEP 算法相比，TRCA 在构建思路上存在较大差别，具体表现在其关注对象（即信号模板）不再限定为具有正余弦波动性质的传统模型，而是充分包含了个体信息的 “任务相关成分”（*Task-related components, TRCs* ）。关于TRC可以简单理解为：当受试者在多次接受相同任务时，其 EEG 信号中应当包含具有相同性质的诱发成分。由此可见，TRCA 在理论上适用于任何诱发信号具有稳定波形特征的 BCI 范式特征信号解码。



*Nakanishi* 等人首次将 TRCA 应用至 SSVEP 信号解码上时，在公式推导部分使用了一个非常讨巧的办法：**跨试次信号相关性最大化**。之所以称其“讨巧”，是因为原版 TRCA 公式回避了很多关键物理意义问题，用几近感性认识一般的操作去掩盖了实际上非常重要的物理过程，这就导致理论与实践过程的脱节：例如分子中强调的**跨试次协方差计算**操作，在编程过程中会产生大量冗余计算步骤；分母的**矩阵拼接**操作也缺乏明确的物理意义对应说明。而上述“瑕疵”在后续算法改进工作中被不断研究透彻。因此本文不再按照原文思路推导算法，仅给出相对成熟的阐释：

对于第 $k$ 类别、第 $i$、$j$ 试次数据 $\boldsymbol{X}_k^i, \boldsymbol{X}_k^j \in \mathbb{R}^{N_c \times N_p}$ (假定 $i \neq j$)，其跨试次样本协方差以及单试次样本方差（自协方差）分别为：

$$Cov\left(\boldsymbol{\omega}_k \boldsymbol{X}_k^i, \boldsymbol{\omega}_k \boldsymbol{X}_k^j\right) = \frac{1}{N_p - 1} \boldsymbol{\omega}_k \boldsymbol{X}_k^i {\boldsymbol{X}_k^j}^T {\boldsymbol{\omega}_k}^T, i \neq j \tag{3-1-1}$$

$$Var\left(\boldsymbol{\omega}_k \boldsymbol{X}_k^i\right) = Cov\left(\boldsymbol{\omega}_k \boldsymbol{X}_k^i, \boldsymbol{\omega}_k \boldsymbol{X}_k^j\right) = \frac{1}{N_p - 1} \boldsymbol{\omega}_k \boldsymbol{X}_k^i {\boldsymbol{X}_k^i}^T {\boldsymbol{\omega}_k}^T \tag{3-1-2}$$

因此，TRCA 的目标函数可写为：

$$\hat{\boldsymbol{\omega}}_k = \arg\max_{\boldsymbol{\omega}_k} \frac{\sum_{j=1, j\neq i}^{N_t} \sum_{i=1}^{N_t} Cov\left(\boldsymbol{\omega}_k \boldsymbol{X}_k^i, \boldsymbol{\omega}_k \boldsymbol{X}_k^j\right)}{\sum_{i=1}^{N_t} Var\left(\boldsymbol{\omega}_k \boldsymbol{X}_k^i\right)} = \arg\max_{\boldsymbol{\omega}_k} \frac{\boldsymbol{\omega}_k \boldsymbol{S}_k {\boldsymbol{\omega}_k}^T}{\boldsymbol{\omega}_k \boldsymbol{Q}_k {\boldsymbol{\omega}_k}^T} \tag{3-1-3}$$

$$\boldsymbol{S}_k = \sum_{j=1, j\neq i}^{N_t} \sum_{i=1}^{N_t} \boldsymbol{X}_k^i {\boldsymbol{X}_k^j}^T, \ \boldsymbol{Q}_k = \sum_{i=1}^{N_t} \boldsymbol{X}_k^i {\boldsymbol{X}_k^i}^T \tag{3-1-4}$$

根据广义瑞利商的结论，上述目标函数的单维度最优解即为方阵 ${\boldsymbol{Q}_k}^{-1} \boldsymbol{S}_k$ 的最大特征值对应的特征向量。接下来对TRCA的目标函数作进一步分析：

$$\boldsymbol{S}_k = \sum_{j=1}^{N_t} \sum_{i=1}^{N_t} \boldsymbol{X}_k^i {\boldsymbol{X}_k^j}^T - \boldsymbol{Q}_k = {N_t}^2 \bar{X}_k {\bar{X}_k}^T - \boldsymbol{Q}_k = \boldsymbol{S}_k^{'} - \boldsymbol{Q}_k \tag{3-1-5}$$

$$\frac{\boldsymbol{\omega}_k \boldsymbol{S}_k {\boldsymbol{\omega}_k}^T}{\boldsymbol{\omega}_k \boldsymbol{Q}_k {\boldsymbol{\omega}_k}^T} = \frac{\boldsymbol{\omega}_k \boldsymbol{S}_k^{'} {\boldsymbol{\omega}_k}^T}{\boldsymbol{\omega}_k \boldsymbol{Q}_k {\boldsymbol{\omega}_k}^T} - 1 \tag{3-1-6}$$

相比于直接计算 $\boldsymbol{S}_k$，经由 $\boldsymbol{S}_k^{'}$ 替换或计算得到 $\boldsymbol{S}_k$ 能够大幅提升运算速度。其原因如下：将单次浮点数相乘与

相加设为两种单位操作，其耗时分别为 $T_{\times}$ 和 $T_{+}$，对应时间复杂度分别为 $O_{\times}$ 与 $O_{+}$。则针对 $\boldsymbol{X}_k^i$ 执行一次矩阵乘法 $\boldsymbol{X}_k^i \boldsymbol{X}_k^{i\,T}$ 或矩阵加法 $\boldsymbol{X}_k^i + \boldsymbol{X}_k^j$ 所需的理论运行时间 $T_{M\times}$、$T_{M+}$ 分别为：

$$T_{M\times} = \left({N_c}^2 N_p\right) T_{+} + \left[{N_c}^2 (N_p - 1)\right] T_{\times} \tag{3-1-7}$$

$$T_{M+} = (N_c N_p) T_{+} \tag{3-1-8}$$

对于具有 $\mathbb{R}^{N_t \times N_c \times N_p}$ 维度的训练数据张量 $\boldsymbol{X}_k$，求解 $\boldsymbol{S}_k$ 的总计理论时间 $T_1$ 与时间复杂度 $O_1$ 分别为：

$$T_1 = N_t \left(N_t - 1\right) T_{M\times} + \left[N_t (N_t - 1) - 1\right] T_{M+} \tag{3-1-9}$$

$$O_1 = O_{\times} \left({N_t}^2 {N_c}^2 N_p\right) + O_{+} \left({N_t}^2 {N_c}^2 N_p\right) \tag{3-1-10}$$

而使用 $\boldsymbol{S}_k^{'}$ 时，首先计算按试次平均后的个体模板 $\bar{\boldsymbol{X}}_k$，其理论运行时间 $T_0$ 为：

$$T_0 = \left(N_c N_p\right) T_{\times} + \left(N_t - 1\right) T_{M+} \tag{3-1-11}$$

$\boldsymbol{S}_k^{'}$ 的总计理论计算时间 $T_2$ 与时间复杂度 $O_2$ 分别为：

$$T_2 = T_0 + T_{M\times} \tag{3-1-12}$$

$$O_2 = O_{\times} \left({N_c}^2 N_p\right) + O_{+} \left[\max\left(N_t N_c N_p, {N_c}^2 N_p\right)\right] \tag{3-1-13}$$

对比 $O_1$ 与 $O_2$ 可见，样本数量越多，采用该种替换方法与原始情况所产生的偏差越小、速度提升越大。

综上所述，通过训练数据获取当前类别专属的空间滤波器 $\hat{\boldsymbol{\omega}}_k$ 以及信号模板 $\hat{\boldsymbol{\omega}}_k \bar{\boldsymbol{X}}_k$，基于一维 *Pearson* 相关系数公式，对单试次测试数据 $\boldsymbol{\chi}$ 应用空间滤波后与模板信号计算判别系数：

$$\rho_k = corr\left(\hat{\boldsymbol{\omega}}_k \bar{\boldsymbol{X}}_k, \hat{\boldsymbol{\omega}}_k \boldsymbol{\chi}\right) \tag{3-1-14}$$

eTRCA 是基于 TRCA 的集成学习版本，它把各类别 $\hat{\boldsymbol{\omega}}_k \in \mathbb{R}^{1 \times N_c}$ 按行拼接在一起组成高维滤波器 $\hat{\boldsymbol{W}}$，通过计算二维 *Pearson* 相关系数完成信号的模式识别：

$$\begin{cases} \hat{\boldsymbol{W}} = \begin{bmatrix} \hat{\boldsymbol{\omega}}_1 \\ \hat{\boldsymbol{\omega}}_2 \\ \vdots \\ \hat{\boldsymbol{\omega}}_{N_e} \end{bmatrix} \in \mathbb{R}^{N_e \times N_c} \\ \rho_k = corr2\left(\hat{\boldsymbol{W}}\bar{\boldsymbol{X}}_k, \hat{\boldsymbol{W}}\boldsymbol{\chi}\right) \end{cases} \tag{3-1-15}$$

其中 $corr2()$ 函数本质上就是先把一个二维矩阵按行拉平变成一维序列，之后再计算一维 *Pearson* 系数。关于这个过程，Matlab 的 **corr2()** 函数给出了一种更适合编程的高效运算方法。对于同维度二维矩阵 $\boldsymbol{A} = [a_{ij}]_{m\times n}$，$\boldsymbol{B} = [b_{ij}]_{m\times n}$，计算矩阵中心 $\bar{\bar{\boldsymbol{A}}}$、$\bar{\bar{\boldsymbol{B}}}$：

$$\bar{\bar{\boldsymbol{A}}} = \sum_{j=1}^{m}\sum_{i=1}^{n} a_{ij} = \sum\sum a_{ij},\ \bar{\bar{\boldsymbol{B}}} = \sum\sum b_{ij} \tag{3-1-16}$$

$$corr2(\boldsymbol{A},\boldsymbol{B}) = \frac{\sum\sum\left(a_{ij}-\bar{\bar{\boldsymbol{A}}}\right)\left(b_{ij}-\bar{\bar{\boldsymbol{B}}}\right)}{\sqrt{\sum\sum\left(a_{ij}-\bar{\bar{\boldsymbol{A}}}\right)^2}\sqrt{\sum\sum\left(b_{ij}-\bar{\bar{\boldsymbol{B}}}\right)^2}} \tag{3-1-17}$$

别被式 (3-1-17) 一层又一层的叠加操作唬住了，在 *ndarray* 的基础上很容易就能将其复现出来。由于避免了大矩阵变量的维度变换，运算速度会有显著提升：

```python
import numpy as np
def corr2_coef(X, Y):
    """2-D Pearson correlation coefficient
    Args:
        X (ndarray): (m,n)
        Y (ndarray): (m,n)
    Returns:
        coef (float)
    """
    mean_X, mean_Y = X.mean(), Y.mean()  # matrix center
    decen_X, decen_Y = X-mean_X, Y-mean_Y  # decentralized matrix
    numerator = np.einsum('ij->', decen_X*decen_Y)
    denominator_X = np.einsum('ij->', decen_X**2)
    denominator_Y = np.einsum('ij->', decen_Y**2)
```

```
    coef = numerator/np.sqrt(denominator_X*denominator_Y)
    return coef
```

论文中关于所谓的集成思想有这样一段描述：

> ***Since there are $N_f$ individual calibration data corresponding to all visual stimuli, $N_f$ different spatial filters can be obtained. Ideally, they should be similar to each other because the mixing coefficients from SSVEP source signals to scalp recordings could be considered similar within the used frequency range, which indicates the possibility of further improvements by intergrating all spatial filters.***

翻译一下，*Nakanishi* 认为 8 - 15.8 Hz 刺激诱发的共计 40 类 SSVEP，彼此之间对应的空间滤波器应当是相似的，因为刺激频段比较窄，诱发脑电的头皮分布模式不至于产生过大的变化。所以根据 8 Hz SSVEP 训练得到的空间滤波器（注意这个“**适用**”应当与 TRCA 的目标函数结合理解），将其应用至其它频率信号时，理论上其目标函数，即式 (3-2) 也能达到比较高的水平，当然这样滤波后的信号质量显然不如“一个萝卜一个坑”，但多多少少能保留相当程度的特征成分。所以将其它类别专属的滤波器用在当前类别上，是变相地扩增了信号模板的空间维度信息。

依我愚见，eTRCA 虽然性能更为强劲，但该算法可能存在原理性缺陷：容易产生冗余成分。在刺激目标较多时，全类别集成似乎并无必要。这一点在 2021 年清华大学 *Liu Binchuan* 发表的 **TDCA** 算法文章中也有所指出，当然人家是大佬，成果已经产出了。鄙人的具体研究工作仍在进行（~~构思~~）中。

至此我们有必要再回顾一下 TRCA 的目标函数：

（1）分子中 $\boldsymbol{\omega}_k \bar{X}_k \bar{X}_k{}^T \boldsymbol{\omega}_k{}^T$ 的本质为“**滤波后特征信号的能量**”。训练样本数目越多，叠加平均操作获取的信号模板质量越高，即随机信号成分削减越充分。而且分子能够决定目标函数的最终优化上限。

（2）分母 $\boldsymbol{\omega}_k \left( \sum_{i=1}^{N_t} \boldsymbol{X}_k^i \boldsymbol{X}_k^{i\,T} \right) \boldsymbol{\omega}_k{}^T$ 的本质为“**滤波后各试次信号能量之和**”。

（3）结合上述两点可见，TRCA 的性能优越是原理性的，其结构相当完善。唯一的缺陷在于训练样本数目：当 $N_t$ 较小时，由（1）可知优化目标将产生无法弥补的偏差。因此后续关于 TRCA 的改进，大多针对少样本下获取更稳健的信号模板估计入手，我们将在 (e)TRCA-R、sc-(e)TRCA 等算法中观察到这一倾向。

# 3.2 多重刺激 TRCA：ms-(e)TRCA

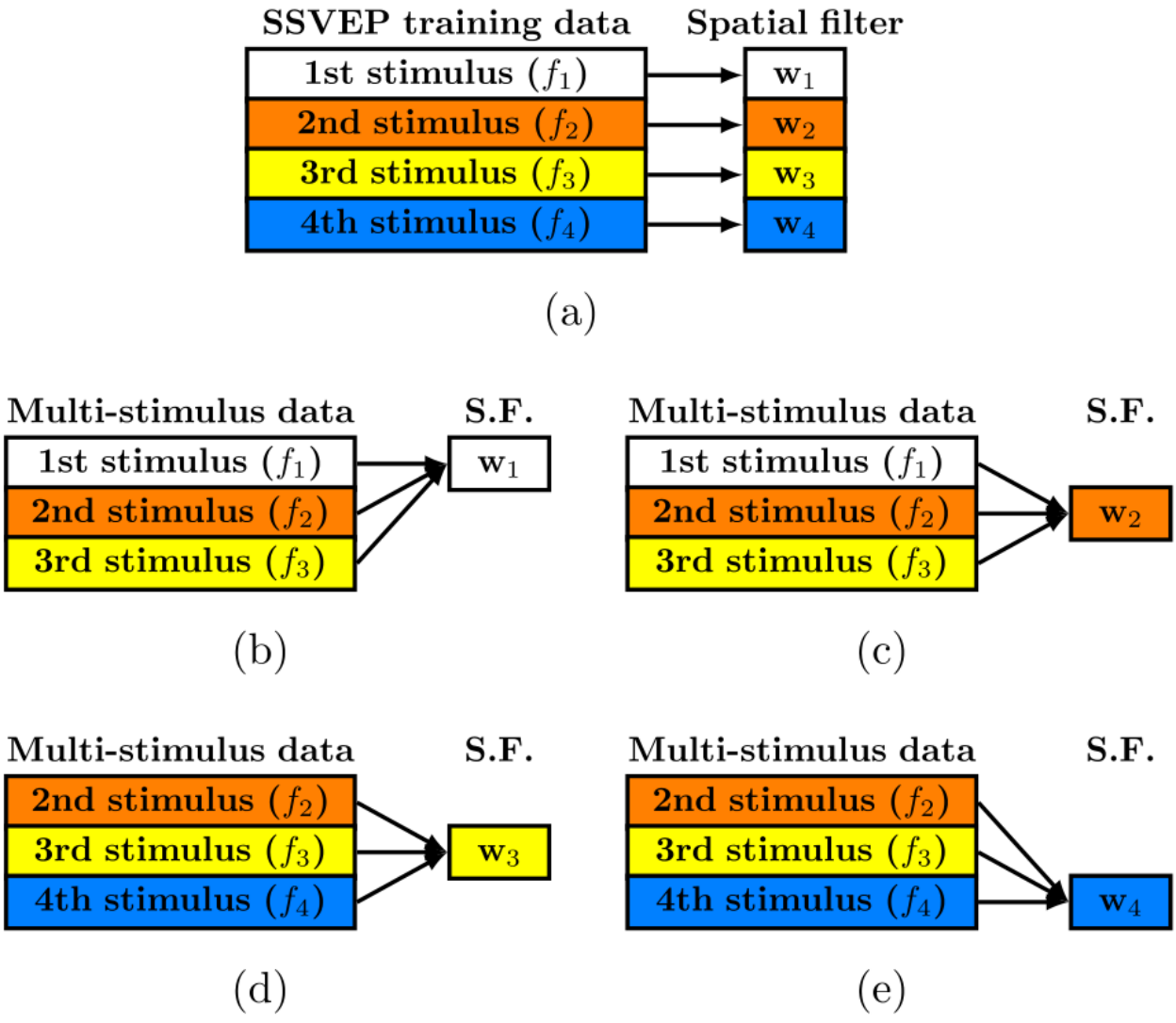
**(Multi-stimulus (e)TRCA)**

**论文链接 | 代码：trca.ms_etrca()**

该算法是我们提到的第一种关于TRCA的改进算法，与 msCCA、ms-eCCA 一样出自 *Wong* 的之手，行文风格也一以贯之。由 3.1 节末尾分析可知，TRCA 类型的算法需要较多的训练数据以保证协方差矩阵估计的准确性，在原文中亦有相关描述：

> ***To guarantee good performance, the number of calibration trials for each visual stimulus cannot be small for the eCCA method and the eTRCA method; otherwise, their recognition accuracies would decrease dramatically . . . A major reason is that the estimate of covariance matrices in the CCA-based (or the TRCA-based) methods could become unreliable in the case of small training data so that the resulting spatial filters may not be accurate. As a matter of fact, the algorithms relying on the estimator of covariance matrix also suffer from insufficient training data problem . . .***

然而对于具有较多指令集（32、40 或更多）的 SSVEP-BCI 系统而言，为每个指令获取足量训练试次（$N_t > 10$）在时间上通常是不允许的。而且刺激时间如果过长，诱发信号的质量会由于受试者累积的视觉疲劳而急剧下降，不利于系统实际应用。因此，*Wong* 提出了一种学习方法，从目标 $A$ 的相邻刺激（$B$、$C$ 等）的诱发信号中学习 $A$ 的特征。以一个简单的 4 指令集系统为例，ms- 技术的学习流程见下图（原文 fig.1）：



原文对于这一套流程的可行性做了连篇累牍的证明，在此仅提炼其最终结论：

**去除 0.14 s 的人眼视觉通路延迟之后，SSVEP 采集信号与闪烁刺激相位基本一致、不同刺激间的相位差也与闪烁刺激相位差基本一致。**

听君一席话，如听一席话是吧？这不是 SSVEP 信号处理的常规步骤吗？此外关于 Benchmark 数据集的刺激参数设定（频率、相位等），他们也特地写了三个**方程**来说明，总之一通操作着实看得人头昏脑胀。鉴于本人有幸曾经听过 *Wong* 的线上报告，视频里他只消三言两语便把 ms-eTRCA 的原理描述得鞭辟入里，所以我相信这种反人类的数学语言其实出自 *Feng Wan* 老师的手笔。

接下来我们先看看原文中给出的公式。ms-eTRCA 与 ms-TRCA 的关系类似 eTRCA 之于 TRCA，因此前两者的目标函数是共通的：

$$\hat{\boldsymbol{\omega}}_k = \arg\max_{\boldsymbol{\omega}_k} \frac{\boldsymbol{\omega}_k \boldsymbol{A}_k {\boldsymbol{A}_k}^T {\boldsymbol{\omega}_k}^T}{\boldsymbol{\omega}_k \boldsymbol{B}_k {\boldsymbol{B}_k}^T {\boldsymbol{\omega}_k}^T} \tag{3-2-1}$$

分子分母中的协方差矩阵与 (e)TRCA 略有差异，具体如下：

$$\begin{cases} \boldsymbol{A}_k = \begin{bmatrix} \bar{\boldsymbol{X}}_{k-m} & \bar{\boldsymbol{X}}_{k-m+1} & \cdots & \bar{\boldsymbol{X}}_{k+n} \end{bmatrix} \in \mathbb{R}^{N_c \times [(m+n+1)N_p]} \\ \boldsymbol{B}_k = \begin{bmatrix} \boldsymbol{\chi}_{k-m} & \boldsymbol{\chi}_{k-m+1} & \cdots & \boldsymbol{\chi}_{k+n} \end{bmatrix} \in \mathbb{R}^{N_c \times [(m+n+1)N_t N_p]} \end{cases} \tag{3-2-2}$$

$$\boldsymbol{\chi}_k = \begin{bmatrix} \boldsymbol{X}_k^1 & \boldsymbol{X}_k^2 & \cdots & \boldsymbol{X}_k^{N_t} \end{bmatrix} \in \mathbb{R}^{N_c \times (N_t N_p)} \tag{3-2-3}$$

可以看出，*multi-stimulus* 技术的本质就是把目标刺激前 $m$、后 $n$ 个不同频率的刺激信号**顺次拼接**起来，在时间维度上对训练数据进行扩增，其范围为 $d$（含自身），具体拼接个数（$m$、$n$）依不同情况各有一番规定。

（1）关于 $m$ 与 $n$ 的大小分配（一般情况）：若 $d$ 为奇数，则前向与后向扩增范围相等；若 $d$ 为偶数，则前向扩增比后向多一位，即有：

$$\begin{cases} m = n = \dfrac{1}{2}(d-1), \quad d = 2n+1 | n \in \mathbb{N}^+ \\ m = \dfrac{1}{2}d, \ n = \dfrac{1}{2}d - 1, \quad d = 2n | n \in \mathbb{N}^+ \end{cases} \tag{3-2-4}$$

（2）假设刺激目标 $k$ 处于相对**靠前**的位置，由于前向没有足够类别的信号用于拼接，因此需向后顺延扩增位数。例如 $d = 5, k = 2$，应向后顺延一位（$m = 1, n = 3$）；若 $d = 6, k = 2$，则向后顺延两位（$m = 1, n = 4$）。综上可总结出式 (3-2-7)：

$$\begin{cases} m = k-1 \\ \\ n = d-k \end{cases}, \ k \in \left[1, \ \frac{1}{2}d\right] \tag{3-2-5}$$

（3）假设刺激目标 $k$ 处于**中部**位置，即 (1) 中所述的“一般情况”，则有式 (3-2-8)：

$$\begin{cases} m = \left[\frac{1}{2}d\right] \\ \\ n = d - \left[\frac{1}{2}d\right] - 1 \end{cases}, \ k \in \left(\left[\frac{1}{2}d\right], \ N_e - \left(d - \left[\frac{1}{2}d\right]\right)\right) \tag{3-2-6}$$

（4）假设刺激目标 $k$ 位于**尾部**位置，此时与（2）相反，需向前顺延扩增位数，即有式 (3-2-9)：

$$\begin{cases} m = d - 1 - (N_e - k) \\ \\ n = N_e - k \end{cases}, \ k \in \left[N_e - \left(d - \left[\frac{1}{2}d\right] - 1\right), N_e\right] \tag{3-2-7}$$

好了，我们再回过头去看式 (3-2-3)，该式分子、分母中的协方差矩阵可以通过与式 (3-1-5) 联动，进一步改写为如下形式：

$$\begin{cases} \boldsymbol{A}_k {\boldsymbol{A}_k}^T = \sum_{i=-m}^{n+1} \bar{\boldsymbol{X}}_{k+i} {\bar{\boldsymbol{X}}_{k+i}}^T = \frac{1}{{N_t}^2} \sum_{i=-m}^{n+1} \boldsymbol{S}'_{k+i} \\ \\ \boldsymbol{B}_k {\boldsymbol{B}_k}^T = \sum_{i=-m}^{n+1} \sum_{j=1}^{N_t} \boldsymbol{X}_i^j {\boldsymbol{X}_i^j}^T = \sum_{i=-m}^{n+1} \boldsymbol{Q}_{k+i} \end{cases} \tag{3-2-8}$$

一般来说，非零常系数是不影响矩阵特征值分解结果的。所以我们看 ms-(e)TRCA 的目标函数式，它就是把不同频率信号对应的 (e)TRCA 目标函数的分子、分母各自相加组成新的分式。再直白一点，就是“**把多个频率的信号当一个频率去训练**”，强行增加了可用样本数目。

我们有一点需要注意，根据文章里网格筛选的结果（原文 Fig.3），$d$ 的范围并非是越大越好，在 (e)TRCA 算法上体现得尤为明显。根据本人测试经验，扩增至全类别时（$d = N_e$），ms-TRCA（此时大家共用一套滤波器也就谈不上集成了）往往还不如 eTRCA 效果好，当然显著优于 TRCA。换句话说，在 TRCA 的内核（目标函数）上简单粗暴地扩增协方差矩阵估计，虽然可能起到一定的效果，但是扩增方案以及参数选择仍然需要仔细调整。

说来也令人感慨，ms- 的思路不可谓不简单，但是 *Wong* 等人之所以成功，一方面是因为敢想敢做，另一方面也要归功于砌墙的数学功底，能够把简单的内核包装成高大上的复杂操作，让人一眼完全看不透其内在关联。

# 3.3 正余弦扩展 TRCA：(e)TRCA-R

论文链接 | 代码：trca.etrca_r()

该算法依旧出自 *Wong* 的手笔。按照其提出的设计框架，-R 技术就是将原本为单位阵的空间投影矩阵替换为正余弦信号张成的投影空间 $\boldsymbol{\mathcal{P}}$，与之类似的算法还有 MsetCCA1-R（未来更新）。在讲解 (e)TRCA-R 之前，我们先来观察 (e)TRCA 的目标函数在统一框架（ 1.3 节式 (1-3-4)，**Type I**）下的各部分组成：

$$\begin{cases} \boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{I}}_{N_t,N_c}\left(\oplus_{i=1}^{N_t}\boldsymbol{X}_k^i\right) \in \mathbb{R}^{N_c\times(N_tN_p)} \\ \boldsymbol{\mathcal{D}} = \boldsymbol{I}_{N_tN_p} \in \mathbb{R}^{(N_tN_p)\times(N_tN_p)} \\ \boldsymbol{\mathcal{P}} = {\boldsymbol{\mathcal{I}}_{N_t,N_p}}^T\boldsymbol{\mathcal{I}}_{N_t,N_p} \in \mathbb{R}^{(N_tN_p)\times(N_tN_p)} \end{cases} \tag{3-3-1}$$

为了更清楚地让大家明白这个框架到底干了什么事，我们来依次画一下各步骤的展开形态：

$$\boldsymbol{\mathcal{Z}} = \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_c} & \boldsymbol{I}_{N_c} & \cdots & \boldsymbol{I}_{N_c}\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_c)}} \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_k^2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{(N_tN_c)\times(N_tN_p)}} = \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 & \boldsymbol{X}_k^2 & \cdots & \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}} \tag{3-3-2}$$

$$\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}} = \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 & \boldsymbol{X}_k^2 & \cdots & \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}} \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_p} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I}_{N_p}\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times(N_tN_p)}} = \boldsymbol{\mathcal{Z}} \tag{3-3-3}$$

$$\boldsymbol{\mathcal{P}} = \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_p} \\ \boldsymbol{I}_{N_p} \\ \vdots \\ \boldsymbol{I}_{N_p}\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times N_p}} \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_p} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p}\end{bmatrix}}_{\mathbb{R}^{N_p\times(N_tN_p)}} = \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_p} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p} \\ \boldsymbol{I}_{N_p} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I}_{N_p} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p}\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times(N_tN_p)}} \tag{3-3-4}$$

$$\boldsymbol{\mathcal{Z}\mathcal{D}\mathcal{P}} = \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 & \cdots & \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}} \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p}\\ \vdots & \ddots & \vdots\\ \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p}\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times(N_tN_p)}} = \underbrace{\begin{bmatrix}\sum_{i=1}^{N_t}\boldsymbol{X}_k^i & \cdots & \sum_{i=1}^{N_t}\boldsymbol{X}_k^i\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}} \tag{3-3-5}$$

$$\boldsymbol{\mathcal{Z}\mathcal{D}\mathcal{P}\mathcal{P}}^T\boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{Z}}^T = \underbrace{\begin{bmatrix}\sum_{i=1}^{N_t}\boldsymbol{X}_k^i & \cdots & \sum_{i=1}^{N_t}\boldsymbol{X}_k^i\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}} \underbrace{\begin{bmatrix}\sum_{i=1}^{N_t}\boldsymbol{X}_k^i\\ \vdots\\ \sum_{i=1}^{N_t}\boldsymbol{X}_k^i\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times N_c}} = N_t\sum_{j=1}^{N_t}\sum_{i=1}^{N_t}\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T \tag{3-3-6}$$

$$\boldsymbol{\mathcal{Z}\mathcal{D}\mathcal{D}}^T\boldsymbol{\mathcal{Z}}^T = \boldsymbol{\mathcal{Z}\mathcal{Z}}^T = \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 & \boldsymbol{X}_k^2 & \cdots & \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}} \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1\\ \boldsymbol{X}_k^2\\ \vdots\\ \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times N_c}} = \sum_{i=1}^{N_t}\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T \tag{3-3-7}$$

综上所述，仅需一维投影向量的情况下，*GEP* 方程可表示为式 (3-3-8)，忽略常系数影响后可发现该式与 (e)TRCA 的目标函数完全吻合。

$$\left(N_t\sum_{j=1}^{N_t}\sum_{i=1}^{N_t}\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T\right)\boldsymbol{\omega} = \lambda\left(\sum_{i=1}^{N_t}\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T\right)\boldsymbol{\omega} \tag{3-3-8}$$

在常规 (e)TRCA 中，正交投影矩阵 $\boldsymbol{\mathcal{P}}$ 的本质作用仅是叠加。而在 (e)TRCA-R 中，*Wong* 将其改为了正余弦信号张成的投影空间，其余均与 (e)TRCA 保持一致：

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix}\boldsymbol{Q}_{\boldsymbol{Y}_k}\\ \vdots\\ \boldsymbol{Q}_{\boldsymbol{Y}_k}\end{bmatrix}\begin{bmatrix}\boldsymbol{Q}_{\boldsymbol{Y}_k} & \cdots & \boldsymbol{Q}_{\boldsymbol{Y}_k}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T & \cdots & \boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T\\ \vdots & \ddots & \vdots\\ \boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T & \cdots & \boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T\end{bmatrix} \tag{3-3-9}$$

注意有 $\boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{P}}^T$，所以 (e)TRCA-R 的 *GEP* 方程可表示为：

$$\left[N_t\sum_{j=1}^{N_t}\sum_{i=1}^{N_t}\left(\boldsymbol{X}_k^i\boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T\right)\left(\boldsymbol{X}_k^j\boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T\right)^T\right]\boldsymbol{\omega}^T = \lambda\left(\sum_{i=1}^{N_t}\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T\right)\boldsymbol{\omega}^T \tag{3-3-10}$$

关于投影矩阵的作用，在 ms-eCCA 章节中已有介绍，此处不再赘述。由于 -R 技术在 $\boldsymbol{X}_k^i \boldsymbol{X}_k^{i^T}$ 的过程中插入了额外的矩阵乘法，导致 TRCA 复现过程中的一个重要 trick（通过叠加平均信号的矩阵乘法替代跨试次循环）不再生效，而 **np.einsum()** 函数在面对跨试次矩阵乘法时似乎又存在一些内在的逻辑问题，其运算效率出奇地低，因此目前我推荐的复现方法只有循环。

## 3.4 相似度约束 TRCA：sc-(e)TRCA

**(Similarity-constrained (e)TRCA)**

论文链接 | 代码：trca.sc_etrca()

## 3.5 组 TRCA：gTRCA

**(Group TRCA)**

论文链接 | 代码：trca.gtrca()

## 3.6 交叉相关性 TRCA：xTRCA

**(Cross-correlation TRCA)**

论文链接 | 代码：trca.xtrca()

---

# x. 其它早期算法

## x.1 最小能量组合：MEC

**(Minimun energy combination)**

论文链接 | 代码：other.mec()

# x.2 最大对比度组合：MCC

**Maximun contrast combination, MCC**

论文链接 | 代码：other.mcc()

---