Machine Learning

A Bayesian and Optimization Perspective

Academic Press, 2015

Sergios Theodoridis[1]

[1]Dept. of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece.

Spring 2015, Version I

Chapter 4
Mean-Square Error Linear Estimation

Mean-Square Error Linear Estimation: The Normal Equations

- The general estimation task is introduced in Chapter 3. There, it is stated that given two dependent random vectors, $\mathbf{y}$ $\mathbf{x}$, the goal of the estimation task is to obtain a function, g, so as given a value $\boldsymbol{x}$ of $\mathbf{x}$, to be able to predict (estimate), in some optimal sense, the corresponding value $\boldsymbol{y}$ of $\mathbf{y}$, i.e., $\hat{\boldsymbol{y}} = g(\boldsymbol{x})$.
- The Mean-Square Error (MSE) estimation is also discussed in Chapter 3. The optimal MSE estimate of $\mathbf{y}$ given the value $\mathbf{x} = \boldsymbol{x}$ is

$$\hat{\boldsymbol{y}} = \mathbb{E}[\mathbf{y}|\boldsymbol{x}].$$

- In general, this is a nonlinear function. We now turn our attention to the case where g is **constrained** to be a linear function. For simplicity and in order to pay more attention to the concepts, we will restrict our discussion to the case of scalar dependent variables.
- Let $(\mathrm{y}, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^l$ be two jointly distributed random entities of zero mean values. If they are not zero, they are subtracted. Our goal is to obtain an estimate of $\boldsymbol{\theta} \in \mathbb{R}^l$ in the **linear** estimator model,

$$\hat{\mathrm{y}} = \boldsymbol{\theta}^T \mathbf{x},$$

so that to minimize the Mean-Square Error (MSE) cost function,

$$J(\boldsymbol{\theta}) = \mathbb{E}\left[(\mathrm{y} - \hat{\mathrm{y}})^2\right].$$

Mean-Square Error Linear Estimation: The Normal Equations

- The general estimation task is introduced in Chapter 3. There, it is stated that given two dependent random vectors, $\mathbf{y}$ $\mathbf{x}$, the goal of the estimation task is to obtain a function, g, so as given a value $\boldsymbol{x}$ of $\mathbf{x}$, to be able to predict (estimate), in some optimal sense, the corresponding value $\boldsymbol{y}$ of $\mathbf{y}$, i.e., $\hat{\boldsymbol{y}} = g(\boldsymbol{x})$.
- The Mean-Square Error (MSE) estimation is also discussed in Chapter 3. The optimal MSE estimate of $\mathbf{y}$ given the value $\mathbf{x} = \boldsymbol{x}$ is

$$\hat{\boldsymbol{y}} = \mathbb{E}[\mathbf{y}|\boldsymbol{x}].$$

- In general, this is a nonlinear function. We now turn our attention to the case where g is **constrained** to be a linear function. For simplicity and in order to pay more attention to the concepts, we will restrict our discussion to the case of scalar dependent variables.
- Let $(\mathrm{y}, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^l$ be two jointly distributed random entities of zero mean values. If they are not zero, they are subtracted. Our goal is to obtain an estimate of $\boldsymbol{\theta} \in \mathbb{R}^l$ in the **linear** estimator model,

$$\hat{\mathrm{y}} = \boldsymbol{\theta}^T \mathbf{x},$$

so that to minimize the Mean-Square Error (MSE) cost function,

$$J(\boldsymbol{\theta}) = \mathbb{E}\left[(\mathrm{y} - \hat{\mathrm{y}})^2\right].$$

Mean-Square Error Linear Estimation: The Normal Equations

- The general estimation task is introduced in Chapter 3. There, it is stated that given two dependent random vectors, $\mathbf{y}$ $\mathbf{x}$, the goal of the estimation task is to obtain a function, g, so as given a value $\boldsymbol{x}$ of $\mathbf{x}$, to be able to predict (estimate), in some optimal sense, the corresponding value $\boldsymbol{y}$ of $\mathbf{y}$, i.e., $\hat{\boldsymbol{y}} = g(\boldsymbol{x})$.
- The Mean-Square Error (MSE) estimation is also discussed in Chapter 3. The optimal MSE estimate of $\mathbf{y}$ given the value $\mathbf{x} = \boldsymbol{x}$ is

$$\hat{\boldsymbol{y}} = \mathbb{E}[\mathbf{y}|\boldsymbol{x}].$$

- In general, this is a nonlinear function. We now turn our attention to the case where g is **constrained** to be a linear function. For simplicity and in order to pay more attention to the concepts, we will restrict our discussion to the case of scalar dependent variables.
- Let $(\mathrm{y}, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^l$ be two jointly distributed random entities of zero mean values. If they are not zero, they are subtracted. Our goal is to obtain an estimate of $\boldsymbol{\theta} \in \mathbb{R}^l$ in the **linear** estimator model,

$$\hat{\mathrm{y}} = \boldsymbol{\theta}^T \mathbf{x},$$

so that to minimize the Mean-Square Error (MSE) cost function,

$$J(\boldsymbol{\theta}) = \mathbb{E}\left[(\mathrm{y} - \hat{\mathrm{y}})^2\right].$$

Mean-Square Error Linear Estimation: The Normal Equations

- The general estimation task is introduced in Chapter 3. There, it is stated that given two dependent random vectors, $\mathbf{y}$ $\mathbf{x}$, the goal of the estimation task is to obtain a function, g, so as given a value $\boldsymbol{x}$ of $\mathbf{x}$, to be able to predict (estimate), in some optimal sense, the corresponding value $\boldsymbol{y}$ of $\mathbf{y}$, i.e., $\hat{\boldsymbol{y}} = g(\boldsymbol{x})$.
- The Mean-Square Error (MSE) estimation is also discussed in Chapter 3. The optimal MSE estimate of $\mathbf{y}$ given the value $\mathbf{x} = \boldsymbol{x}$ is

$$\hat{\boldsymbol{y}} = \mathbb{E}[\mathbf{y}|\boldsymbol{x}].$$

- In general, this is a nonlinear function. We now turn our attention to the case where g is **constrained** to be a linear function. For simplicity and in order to pay more attention to the concepts, we will restrict our discussion to the case of scalar dependent variables.
- Let $(\mathrm{y}, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^l$ be two jointly distributed random entities of zero mean values. If they are not zero, they are subtracted. Our goal is to obtain an estimate of $\boldsymbol{\theta} \in \mathbb{R}^l$ in the **linear** estimator model,

$$\hat{\mathrm{y}} = \boldsymbol{\theta}^T \mathbf{x},$$

so that to minimize the Mean-Square Error (MSE) cost function,

$$J(\boldsymbol{\theta}) = \mathbb{E}\left[(\mathrm{y} - \hat{\mathrm{y}})^2\right].$$

Mean-Square Error Linear Estimation: The Normal Equations

- In other words, the optimal estimator is chosen so as to minimize the variance of the error random variable

$$\mathrm{e} = \mathrm{y} - \hat{\mathrm{y}}.$$

- Minimizing the cost function, $J(\boldsymbol{\theta})$, is equivalent with setting its gradient with respect to $\boldsymbol{\theta}$ equal to zero,

$$\begin{aligned}\nabla J(\boldsymbol{\theta}) &= \nabla \mathbb{E}\Big[(\mathrm{y} - \boldsymbol{\theta}^T \mathbf{x})(\mathrm{y} - \mathbf{x}^T \boldsymbol{\theta})\Big] \\ &= \nabla \Big\{\mathbb{E}[\mathrm{y}^2] - 2\boldsymbol{\theta}^T \mathbb{E}[\mathbf{x}\mathrm{y}] + \boldsymbol{\theta}^T \mathbb{E}[\mathbf{x}\mathbf{x}^T]\boldsymbol{\theta}\Big\} = -2\boldsymbol{p} + 2\Sigma_x \boldsymbol{\theta} = 0.\end{aligned}$$

- Solving the above leads to

$$\Sigma_x \boldsymbol{\theta}_* = \boldsymbol{p}$$

where, the input-output cross-correlation vector, $\boldsymbol{p}$, and the respective covariance matrix are given by given by

$$\boldsymbol{p} = \Big[\mathbb{E}[\mathrm{x}_1\mathrm{y}], ..., \mathbb{E}[\mathrm{x}_l\mathrm{y}]\Big]^T = \mathbb{E}[\mathbf{x}\mathrm{y}], \quad \Sigma_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$$

- Thus, the weights of the optimal linear estimator are obtained via a linear system of equations, provided that the covariance matrix is positive definite. Moreover, in this case, the solution is unique.

Mean-Square Error Linear Estimation: The Normal Equations

- In other words, the optimal estimator is chosen so as to minimize the variance of the error random variable

$$\mathrm{e} = \mathrm{y} - \hat{\mathrm{y}}.$$

- Minimizing the cost function, $J(\boldsymbol{\theta})$, is equivalent with setting its gradient with respect to $\boldsymbol{\theta}$ equal to zero,

$$\begin{aligned}\nabla J(\boldsymbol{\theta}) &= \nabla \mathbb{E}\Big[\big(\mathrm{y} - \boldsymbol{\theta}^T \mathbf{x}\big)\big(\mathrm{y} - \mathbf{x}^T \boldsymbol{\theta}\big)\Big] \\ &= \nabla\Big\{\mathbb{E}[\mathrm{y}^2] - 2\boldsymbol{\theta}^T \mathbb{E}[\mathbf{x}\mathrm{y}] + \boldsymbol{\theta}^T \mathbb{E}[\mathbf{x}\mathbf{x}^T]\boldsymbol{\theta}\Big\} = -2\boldsymbol{p} + 2\Sigma_x \boldsymbol{\theta} = \mathbf{0}.\end{aligned}$$

- Solving the above leads to

$$\Sigma_x \boldsymbol{\theta}_* = \boldsymbol{p}$$

where, the input-output cross-correlation vector, $\boldsymbol{p}$, and the respective covariance matrix are given by given by

$$\boldsymbol{p} = \big[\mathbb{E}[\mathrm{x}_1\mathrm{y}], ..., \mathbb{E}[\mathrm{x}_l\mathrm{y}]\big]^T = \mathbb{E}[\mathbf{x}\mathrm{y}], \quad \Sigma_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$$

- Thus, the weights of the optimal linear estimator are obtained via a linear system of equations, provided that the covariance matrix is positive definite. Moreover, in this case, the solution is unique.

Mean-Square Error Linear Estimation: The Normal Equations

- In other words, the optimal estimator is chosen so as to minimize the variance of the error random variable

$$\mathrm{e} = \mathrm{y} - \hat{\mathrm{y}}.$$

- Minimizing the cost function, $J(\boldsymbol{\theta})$, is equivalent with setting its gradient with respect to $\boldsymbol{\theta}$ equal to zero,

$$\begin{aligned}\nabla J(\boldsymbol{\theta}) &= \nabla \mathbb{E}\Big[\big(\mathrm{y} - \boldsymbol{\theta}^T\mathbf{x}\big)\big(\mathrm{y} - \mathbf{x}^T\boldsymbol{\theta}\big)\Big] \\ &= \nabla\Big\{\mathbb{E}[\mathrm{y}^2] - 2\boldsymbol{\theta}^T\mathbb{E}[\mathbf{x}\mathrm{y}] + \boldsymbol{\theta}^T\mathbb{E}[\mathbf{x}\mathbf{x}^T]\boldsymbol{\theta}\Big\} = -2\boldsymbol{p} + 2\Sigma_x\boldsymbol{\theta} = \mathbf{0}.\end{aligned}$$

- Solving the above leads to

$$\Sigma_x\boldsymbol{\theta}_* = \boldsymbol{p}$$

where, the input-output cross-correlation vector, $\boldsymbol{p}$, and the respective covariance matrix are given by given by

$$\boldsymbol{p} = \Big[\mathbb{E}[\mathrm{x}_1\mathrm{y}], ..., \mathbb{E}[\mathrm{x}_l\mathrm{y}]\Big]^T = \mathbb{E}[\mathbf{x}\mathrm{y}], \quad \Sigma_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$$

- Thus, the weights of the optimal linear estimator are obtained via a linear system of equations, provided that the covariance matrix is positive definite. Moreover, in this case, the solution is unique.

Mean-Square Error Linear Estimation: The Normal Equations

- In other words, the optimal estimator is chosen so as to minimize the variance of the error random variable

$$\mathrm{e} = \mathrm{y} - \hat{\mathrm{y}}.$$

- Minimizing the cost function, $J(\boldsymbol{\theta})$, is equivalent with setting its gradient with respect to $\boldsymbol{\theta}$ equal to zero,

$$\begin{aligned}\nabla J(\boldsymbol{\theta}) &= \nabla \mathbb{E}\Big[\big(\mathrm{y} - \boldsymbol{\theta}^T\mathbf{x}\big)\big(\mathrm{y} - \mathbf{x}^T\boldsymbol{\theta}\big)\Big] \\ &= \nabla\Big\{\mathbb{E}[\mathrm{y}^2] - 2\boldsymbol{\theta}^T\mathbb{E}[\mathbf{x}\mathrm{y}] + \boldsymbol{\theta}^T\mathbb{E}[\mathbf{x}\mathbf{x}^T]\boldsymbol{\theta}\Big\} = -2\boldsymbol{p} + 2\Sigma_x\boldsymbol{\theta} = \mathbf{0}.\end{aligned}$$

- Solving the above leads to

$$\Sigma_x\boldsymbol{\theta}_* = \boldsymbol{p}$$

where, the input-output cross-correlation vector, $\boldsymbol{p}$, and the respective covariance matrix are given by given by

$$\boldsymbol{p} = \Big[\mathbb{E}[\mathrm{x}_1\mathrm{y}], ..., \mathbb{E}[\mathrm{x}_l\mathrm{y}]\Big]^T = \mathbb{E}[\mathbf{x}\mathrm{y}], \quad \Sigma_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$$

- Thus, the weights of the optimal linear estimator are obtained via a linear system of equations, provided that the covariance matrix is positive definite. Moreover, in this case, the solution is unique.

The MSE Cost Function

- Elaborating on the MSE cost function, $J(\boldsymbol{\theta})$, we get that

$$J(\boldsymbol{\theta}) = \sigma_y^2 - 2\boldsymbol{\theta}^T \boldsymbol{p} + \boldsymbol{\theta}^T \Sigma_x \boldsymbol{\theta}.$$

Adding and subtracting the term $\boldsymbol{\theta}_*^T \Sigma_x \boldsymbol{\theta}_*$ and taking into account the definition of $\boldsymbol{\theta}_*$ ($\Sigma_x \boldsymbol{\theta}_* = \boldsymbol{p}$), it is readily seen that

$$J(\boldsymbol{\theta}) = J(\boldsymbol{\theta}_*) + (\boldsymbol{\theta} - \boldsymbol{\theta}_*)^T \Sigma_x (\boldsymbol{\theta} - \boldsymbol{\theta}_*), \tag{1}$$

where

$$J(\boldsymbol{\theta}_*) = \sigma_y^2 - \boldsymbol{p}^T \Sigma_x^{-1} \boldsymbol{p} = \sigma_y^2 - \boldsymbol{\theta}_*^T \Sigma_x \boldsymbol{\theta}_* = \sigma_y^2 - \boldsymbol{p}^T \boldsymbol{\theta}_*, \tag{2}$$

is the minimum achieved at the optimal solution.

- The following remarks are in order:
  - The cost at the optimal value θ_* is always less than the variance $\mathbb{E}[\mathrm{y}^2]$ of the output variable. This is guaranteed by the positive definite nature of Σ_x or Σ_x^{-1}, unless $\boldsymbol{p} = \mathbf{0}$; however, the latter is zero if $\mathbf{x}$ and y, are uncorrelated. On the contrary, if the input-output variables are correlated, then observing $\mathbf{x}$ **removes** part of the **uncertainty** associated with y.
  - For any value θ, other than the optimal θ_*, the error variance increases as (1) suggests, due to the positive definite nature of Σ_x.

The MSE Cost Function

- Elaborating on the MSE cost function, $J(\boldsymbol{\theta})$, we get that

$$J(\boldsymbol{\theta}) = \sigma_y^2 - 2\boldsymbol{\theta}^T \boldsymbol{p} + \boldsymbol{\theta}^T \Sigma_x \boldsymbol{\theta}.$$

  Adding and subtracting the term $\boldsymbol{\theta}_*^T \Sigma_x \boldsymbol{\theta}_*$ and taking into account the definition of $\boldsymbol{\theta}_*$ ($\Sigma_x \boldsymbol{\theta}_* = \boldsymbol{p}$), it is readily seen that

$$J(\boldsymbol{\theta}) = J(\boldsymbol{\theta}_*) + (\boldsymbol{\theta} - \boldsymbol{\theta}_*)^T \Sigma_x (\boldsymbol{\theta} - \boldsymbol{\theta}_*), \tag{1}$$

  where

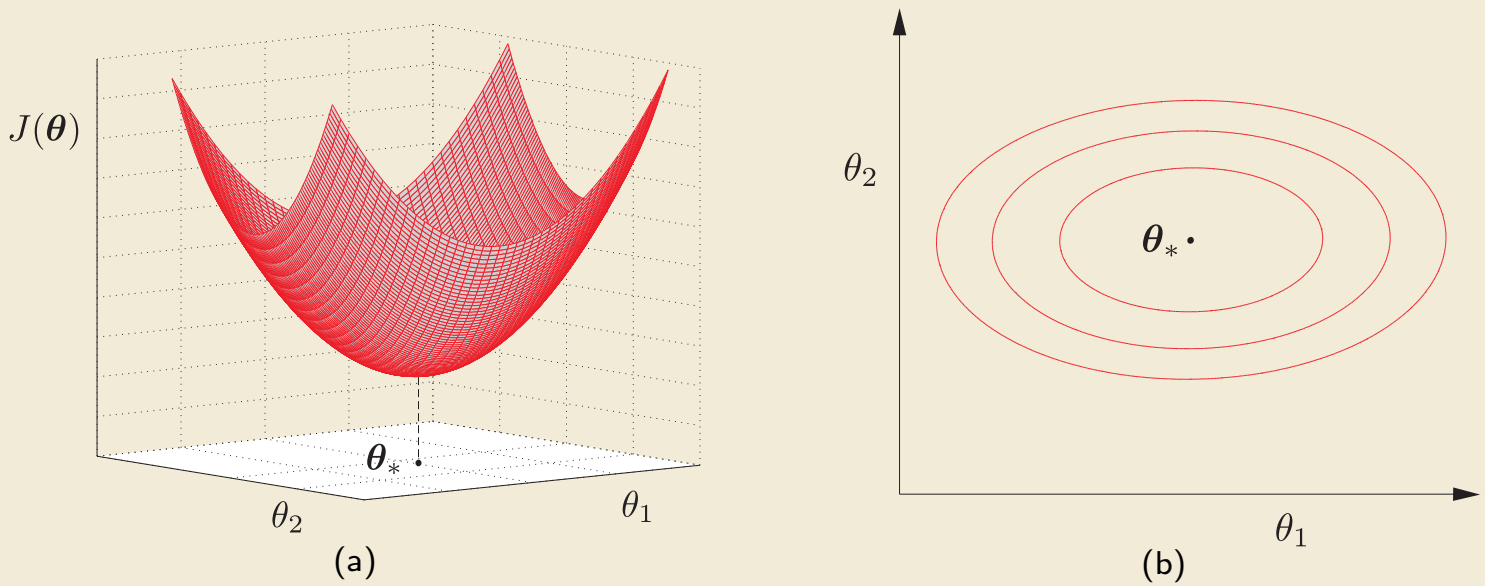
$$J(\boldsymbol{\theta}_*) = \sigma_y^2 - \boldsymbol{p}^T \Sigma_x^{-1} \boldsymbol{p} = \sigma_y^2 - \boldsymbol{\theta}_*^T \Sigma_x \boldsymbol{\theta}_* = \sigma_y^2 - \boldsymbol{p}^T \boldsymbol{\theta}_*, \tag{2}$$

  is the minimum achieved at the optimal solution.

- The following remarks are in order:
  - The cost at the optimal value $\boldsymbol{\theta}_*$ is always less than the variance $\mathbb{E}[\mathrm{y}^2]$ of the output variable. This is guaranteed by the positive definite nature of Σ_x or Σ_x^{-1}, unless $\boldsymbol{p} = \mathbf{0}$; however, the latter is zero if $\mathbf{x}$ and y, are uncorrelated. On the contrary, if the input-output variables are correlated, then observing $\mathbf{x}$ **removes** part of the **uncertainty** associated with y.
  - For any value $\boldsymbol{\theta}$, other than the optimal $\boldsymbol{\theta}_*$, the error variance increases as (1) suggests, due to the positive definite nature of Σ_x.

The MSE Cost Function Surface

- Figure (a) shows the MSE cost function surface and Figure (b) the corresponding isovalue contours. The latter are in general ellipses, whose axes are determined by the eigenstructure of Σ_x. For $\Sigma_x = \sigma^2 I$, where all eigenvalues are equal to σ^2, the contours are circles.



(a) The MSE cost has the form of a (hyper)paraboloid. b) The isovalue contours for the MSE cost function surface are ellipses; the major axis of each ellipse is determined by the maximum eigenvalue λ_{max} and the minor one by the smaller, λ_{min} of the Σ of the input random variables. The largest the ratio $\frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}$ is the more elongated the ellipses are. The ellipses become circles, if the covariance matrix has the special form of $\sigma^2 I$. That is, all variables are mutually uncorrelated and they have the same variance. By varying Σ, different shapes of the ellipses and different orientations result.

A Geometric Viewpoint: Orthogonality Condition

- A very intuitive view of what we have said so far comes from the geometric interpretation of the random variables. The reader can easily check out that the set of random variables is a vector space over the field of real (and complex) numbers. Indeed, if x and y are any two random variables then $\mathrm{x}+\mathrm{y}$, as well as $\alpha\mathrm{x}$, are also random variables for every $\alpha \in \mathbb{R}$.
- We can now equip this vector space with an inner product operation, which also implies a norm and make it a Euclidean space. The mean value operation has all the properties required for an operation to be called an inner product. Indeed, for any subset of random variables,
  - $\mathbb{E}[\mathrm{xy}] = \mathbb{E}[\mathrm{yx}]$
  - $\mathbb{E}[(\alpha_1\mathrm{x}_1 + \alpha_2\mathrm{x}_2)\mathrm{y}] = \alpha_1\mathbb{E}[\mathrm{x}_1\mathrm{y}] + \alpha_2\mathbb{E}[\mathrm{x}_2\mathrm{y}]$
  - $\mathbb{E}[\mathrm{x}^2] \geq 0$, with equality if and only if $\mathrm{x} = 0$.
- Thus, the norm induced by this inner product,

$$\|\mathrm{x}\| := \sqrt{\mathbb{E}[\mathrm{x}^2]},$$

coincides with the respective standard deviation (assuming $\mathbb{E}[\mathrm{x}] = 0$). From now on, given two uncorrelated random variables, x, y, i.e., $\mathbb{E}[\mathrm{xy}] = 0$, we can call them orthogonal, since their inner product is zero.

A Geometric Viewpoint: Orthogonality Condition

- A very intuitive view of what we have said so far comes from the geometric interpretation of the random variables. The reader can easily check out that the set of random variables is a vector space over the field of real (and complex) numbers. Indeed, if x and y are any two random variables then $\mathrm{x}+\mathrm{y}$, as well as $\alpha\mathrm{x}$, are also random variables for every $\alpha \in \mathbb{R}$.
- We can now equip this vector space with an inner product operation, which also implies a norm and make it a Euclidean space. The mean value operation has all the properties required for an operation to be called an inner product. Indeed, for any subset of random variables,
  - $\mathbb{E}[\mathrm{xy}] = \mathbb{E}[\mathrm{yx}]$
  - $\mathbb{E}[(\alpha_1\mathrm{x}_1 + \alpha_2\mathrm{x}_2)\mathrm{y}] = \alpha_1\mathbb{E}[\mathrm{x}_1\mathrm{y}] + \alpha_2\mathbb{E}[\mathrm{x}_2\mathrm{y}]$
  - $\mathbb{E}[\mathrm{x}^2] \geq 0$, with equality if and only if $\mathrm{x} = 0$.
- Thus, the norm induced by this inner product,

$$\|\mathrm{x}\| := \sqrt{\mathbb{E}[\mathrm{x}^2]},$$

coincides with the respective standard deviation (assuming $\mathbb{E}[\mathrm{x}] = 0$). From now on, given two uncorrelated random variables, x, y, i.e., $\mathbb{E}[\mathrm{xy}] = 0$, we can call them orthogonal, since their inner product is zero.

A Geometric Viewpoint: Orthogonality Condition

- A very intuitive view of what we have said so far comes from the geometric interpretation of the random variables. The reader can easily check out that the set of random variables is a vector space over the field of real (and complex) numbers. Indeed, if x and y are any two random variables then $\mathrm{x}+\mathrm{y}$, as well as $\alpha\mathrm{x}$, are also random variables for every $\alpha \in \mathbb{R}$.
- We can now equip this vector space with an inner product operation, which also implies a norm and make it a Euclidean space. The mean value operation has all the properties required for an operation to be called an inner product. Indeed, for any subset of random variables,
  - $\mathbb{E}[\mathrm{xy}] = \mathbb{E}[\mathrm{yx}]$
  - $\mathbb{E}[(\alpha_1\mathrm{x}_1 + \alpha_2\mathrm{x}_2)\mathrm{y}] = \alpha_1\mathbb{E}[\mathrm{x}_1\mathrm{y}] + \alpha_2\mathbb{E}[\mathrm{x}_2\mathrm{y}]$
  - $\mathbb{E}[\mathrm{x}^2] \geq 0$, with equality if and only if $\mathrm{x} = 0$.
- Thus, the norm induced by this inner product,

$$\|\mathrm{x}\| := \sqrt{\mathbb{E}[\mathrm{x}^2]},$$

coincides with the respective standard deviation (assuming $\mathbb{E}[\mathrm{x}] = 0$). From now on, given two uncorrelated random variables, x, y, i.e., $\mathbb{E}[\mathrm{xy}] = 0$, we can call them orthogonal, since their inner product is zero.

A Geometric Viewpoint: Orthogonality Condition

- Let us now rewrite the linear estimator as

$$\hat{\mathrm{y}} = \theta_1 \mathrm{x}_1 + \ldots + \theta_l \mathrm{x}_l.$$

  Thus, the random variable, $\hat{\mathrm{y}}$, which is now interpreted as a point in a vector space, results as a linear combination of l vectors in this space. Thus, the estimate, $\hat{\mathrm{y}}$, will necessarily lie in the subspace spanned by these points. In contrast, the true variable, y, will not lie, in general, in this subspace.

- Since our goal is to obtain $\hat{\mathrm{y}}$ so that to be a good approximation of y, we have to seek for the specific linear combination that makes the norm of the error vector, $\mathrm{e} = \mathrm{y} - \hat{\mathrm{y}}$, minimum.
- This specific linear combination corresponds to the orthogonal projection of y onto the subspace spanned by the points $\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_l$. This is equivalent with requiring,

$$\mathbb{E}[\mathrm{e}\mathrm{x}_k] = 0, \ \ k = 1, ..., l.$$

- The error vector being orthogonal to every point $\mathrm{x}_k, \ k = 1, 2, \ldots, l$, will be orthogonal to the respective subspace. This is geometrically illustrated in the following figure.

A Geometric Viewpoint: Orthogonality Condition

- Let us now rewrite the linear estimator as

$$\hat{\mathrm{y}} = \theta_1 \mathrm{x}_1 + \ldots + \theta_l \mathrm{x}_l.$$

  Thus, the random variable, $\hat{\mathrm{y}}$, which is now interpreted as a point in a vector space, results as a linear combination of l vectors in this space. Thus, the estimate, $\hat{\mathrm{y}}$, will necessarily lie in the subspace spanned by these points. In contrast, the true variable, y, will not lie, in general, in this subspace.
- Since our goal is to obtain $\hat{\mathrm{y}}$ so that to be a good approximation of y, we have to seek for the specific linear combination that makes the norm of the error vector, $\mathrm{e} = \mathrm{y} - \hat{\mathrm{y}}$, minimum.
- This specific linear combination corresponds to the orthogonal projection of y onto the subspace spanned by the points $\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_l$. This is equivalent with requiring,

$$\mathbb{E}[\mathrm{e}\mathrm{x}_k] = 0, \ k = 1, ..., l.$$

- The error vector being orthogonal to every point $\mathrm{x}_k, \ k = 1, 2, \ldots, l$, will be orthogonal to the respective subspace. This is geometrically illustrated in the following figure.

A Geometric Viewpoint: Orthogonality Condition

- Let us now rewrite the linear estimator as

$$\hat{\mathrm{y}} = \theta_1 \mathrm{x}_1 + \ldots + \theta_l \mathrm{x}_l.$$

  Thus, the random variable, $\hat{\mathrm{y}}$, which is now interpreted as a point in a vector space, results as a linear combination of l vectors in this space. Thus, the estimate, $\hat{\mathrm{y}}$, will necessarily lie in the subspace spanned by these points. In contrast, the true variable, y, will not lie, in general, in this subspace.
- Since our goal is to obtain $\hat{\mathrm{y}}$ so that to be a good approximation of y, we have to seek for the specific linear combination that makes the norm of the error vector, $\mathrm{e} = \mathrm{y} - \hat{\mathrm{y}}$, minimum.
- This specific linear combination corresponds to the orthogonal projection of y onto the subspace spanned by the points $\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_l$. This is equivalent with requiring,

$$\mathbb{E}[\mathrm{e}\mathrm{x}_k] = 0, \ k = 1, ..., l.$$

- The error vector being orthogonal to every point $\mathrm{x}_k, \ k = 1, 2, \ldots, l$, will be orthogonal to the respective subspace. This is geometrically illustrated in the following figure.

A Geometric Viewpoint: Orthogonality Condition

- Let us now rewrite the linear estimator as

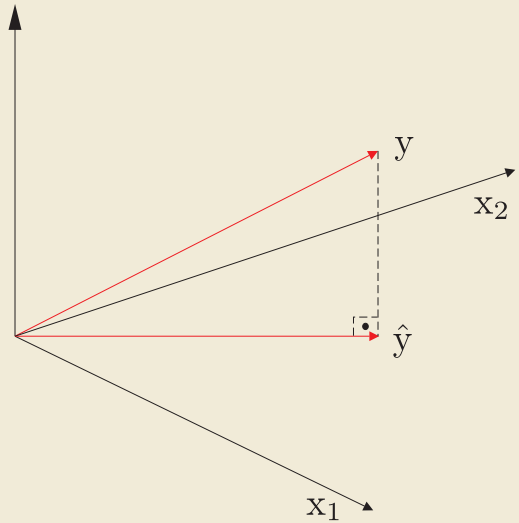
$$\hat{\mathrm{y}} = \theta_1 \mathrm{x}_1 + \ldots + \theta_l \mathrm{x}_l.$$

  Thus, the random variable, $\hat{\mathrm{y}}$, which is now interpreted as a point in a vector space, results as a linear combination of l vectors in this space. Thus, the estimate, $\hat{\mathrm{y}}$, will necessarily lie in the subspace spanned by these points. In contrast, the true variable, y, will not lie, in general, in this subspace.
- Since our goal is to obtain $\hat{\mathrm{y}}$ so that to be a good approximation of y, we have to seek for the specific linear combination that makes the norm of the error vector, $\mathrm{e} = \mathrm{y} - \hat{\mathrm{y}}$, minimum.
- This specific linear combination corresponds to the orthogonal projection of y onto the subspace spanned by the points $\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_l$. This is equivalent with requiring,

$$\mathbb{E}[\mathrm{e}\mathrm{x}_k] = 0, \;\; k = 1, ..., l.$$

- The error vector being orthogonal to every point $\mathrm{x}_k, \;\; k = 1, 2, \ldots, l$, will be orthogonal to the respective subspace. This is geometrically illustrated in the following figure.

A Geometric Viewpoint: Orthogonality Condition



Projecting y on the subspace spanned by $\mathrm{x}_1, \mathrm{x}_2$ guarantees that the deviation between y and $\hat{\mathrm{y}}$ corresponds to the minimum MSE.

- The set of equations in the orthogonal conditions can now be written as

$$\mathbb{E}\big[(\mathrm{y} - \sum_{i=1}^{l} \theta_i \mathrm{x}_i)\mathrm{x}_k\big] = 0, \;\; k = 1, 2, \ldots, l,$$

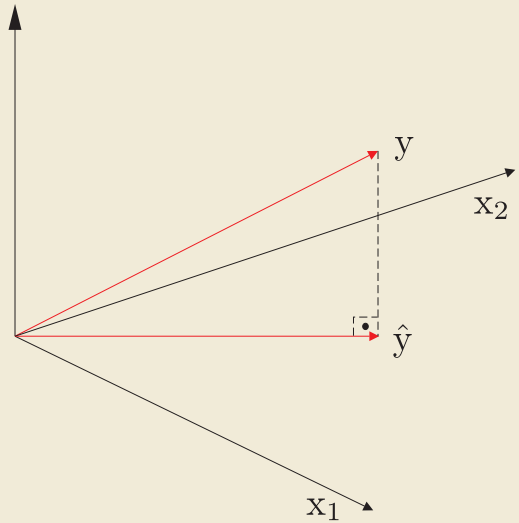
or

$$\sum_{i=1}^{l} \mathbb{E}[\mathrm{x}_i \mathrm{x}_k]\theta_i = \mathbb{E}[\mathrm{x}_k \mathrm{y}], \;\; k = 1, 2, \ldots, l,$$

which leads to the same set of linear equations, $\Sigma_x \boldsymbol{\theta} = \boldsymbol{p}$.

- The derivation via the orthogonality conditions is the reason that this elegant set of equations is known as normal equations. Another name is Wiener-Hopf equations.

A Geometric Viewpoint: Orthogonality Condition



Projecting y on the subspace spanned by $\mathrm{x}_1, \mathrm{x}_2$ guarantees that the deviation between y and $\hat{\mathrm{y}}$ corresponds to the minimum MSE.

- The set of equations in the orthogonal conditions can now be written as

$$\mathbb{E}\big[\big(\mathrm{y} - \sum_{i=1}^{l} \theta_i \mathrm{x}_i\big)\mathrm{x}_k\big] = 0, \;\; k = 1, 2, \ldots, l,$$

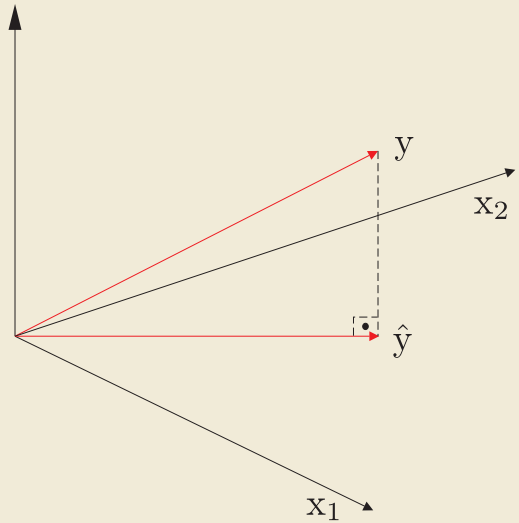
or

$$\sum_{i=1}^{l} \mathbb{E}[\mathrm{x}_i \mathrm{x}_k]\theta_i = \mathbb{E}[\mathrm{x}_k \mathrm{y}], \;\; k = 1, 2, \ldots, l,$$

which leads to the same set of linear equations, $\Sigma_x \boldsymbol{\theta} = \boldsymbol{p}$.

- The derivation via the orthogonality conditions is the reason that this elegant set of equations is known as normal equations. Another name is Wiener-Hopf equations.

A Geometric Viewpoint: Orthogonality Condition



Projecting y on the subspace spanned by $\mathrm{x}_1, \mathrm{x}_2$ guarantees that the deviation between y and $\hat{\mathrm{y}}$ corresponds to the minimum MSE.

- The set of equations in the orthogonal conditions can now be written as

$$\mathbb{E}\big[\big(\mathrm{y} - \sum_{i=1}^{l} \theta_i \mathrm{x}_i\big)\mathrm{x}_k\big] = 0, \;\; k = 1, 2, \ldots, l,$$

or

$$\sum_{i=1}^{l} \mathbb{E}[\mathrm{x}_i \mathrm{x}_k]\theta_i = \mathbb{E}[\mathrm{x}_k \mathrm{y}], \;\; k = 1, 2, \ldots, l,$$

which leads to the same set of linear equations, $\Sigma_x \boldsymbol{\theta} = \boldsymbol{p}$.

- The derivation via the orthogonality conditions is the reason that this elegant set of equations is known as normal equations. Another name is Wiener-Hopf equations.

A Geometric Viewpoint: Orthogonality Condition

- Some remarks:
  - So far, we have only assumed that $\mathbf{x}$ and y are jointly distributed (correlated) variables. If, in addition, we assume that they are linearly related according to the linear regression model,

$$y = \boldsymbol{\theta}_o^T \mathbf{x} + \eta, \ \boldsymbol{\theta}_o \in \mathbb{R}^k,$$

    where η is a zero mean noise variable independent of $\mathbf{x}$, then, if the dimension, k, of the true system, $\boldsymbol{\theta_o}$ is equal to the number of parameters, l, adopted for the model, i.e., the $k = l$, it turns out that

$$\boldsymbol{\theta}_* = \boldsymbol{\theta}_o,$$

    and the optimal MSE is equal to the variance of the noise, σ_η^2.
  - Undermodeling. If $k > l$, then the order of the model is less than that of the true system; this is known as undermodeling. It is easy to show that if the variables comprising $\mathbf{x}$ are uncorrelated, then,

$$\boldsymbol{\theta}_* = \boldsymbol{\theta}_o^1, \quad \text{where} \quad \boldsymbol{\theta}_o := \begin{bmatrix} \boldsymbol{\theta}_o^1 \\ \boldsymbol{\theta}_o^2 \end{bmatrix}, \ \boldsymbol{\theta}_o^1 \in \mathbb{R}^l, \ \boldsymbol{\theta}_o^2 \in \mathbb{R}^{k-l}.$$

    In other words, the MSE optimal estimator identifies the first l components of $\boldsymbol{\theta}_o$.

A Geometric Viewpoint: Orthogonality Condition

- Some remarks:
  - So far, we have only assumed that $\mathbf{x}$ and y are jointly distributed (correlated) variables. If, in addition, we assume that they are linearly related according to the linear regression model,

$$y = \boldsymbol{\theta}_o^T \mathbf{x} + \eta, \ \boldsymbol{\theta}_o \in \mathbb{R}^k,$$

    where η is a zero mean noise variable independent of $\mathbf{x}$, then, if the dimension, k, of the true system, $\boldsymbol{\theta}_o$ is equal to the number of parameters, l, adopted for the model, i.e., the $k = l$, it turns out that

$$\boldsymbol{\theta}_* = \boldsymbol{\theta}_o,$$

    and the optimal MSE is equal to the variance of the noise, σ_η^2.
  - Undermodeling. If $k > l$, then the order of the model is less than that of the true system; this is known as undermodeling. It is easy to show that if the variables comprising $\mathbf{x}$ are uncorrelated, then,

$$\boldsymbol{\theta}_* = \boldsymbol{\theta}_o^1, \quad \text{where} \quad \boldsymbol{\theta}_o := \begin{bmatrix} \boldsymbol{\theta}_o^1 \\ \boldsymbol{\theta}_o^2 \end{bmatrix}, \ \boldsymbol{\theta}_o^1 \in \mathbb{R}^l, \ \boldsymbol{\theta}_o^2 \in \mathbb{R}^{k-l}.$$

    In other words, the MSE optimal estimator identifies the first l components of $\boldsymbol{\theta}_o$.

Linear Filtering

- Linear statistical filtering is an instance of the general estimation task, when the notion of (time) evolution needs to be taken into consideration and estimates are obtained at each time instant. There are three major types of problems that emerge:
  - Filtering, where the estimate at time instant n is based on all previously received (measured) input information up to and including the current time index, n.
  - Smoothing, where data over a time interval, $[0, N]$, are first collected and an estimate is obtained at each time instant $n \leq N$, using all the available information in the interval $[0, N]$.
  - Prediction, where estimates at times $n + \tau,\ \tau > 0$ are to be obtained based on the information up to and including time instant n.
- Take for example a time-varying case, where the output variable, at time instant n, is y_n and its value depends on observations included in the corresponding input vector $\mathbf{x}_n$. In filtering, the latter can include measurements received only at time instants, $n, n-1, \ldots, 0$. This restriction in the index set, is directly related to **causality**. In contrast, in smoothing, we can also include future time instants, e.g., $n+2, n+1, n, n-1$.

Linear Filtering

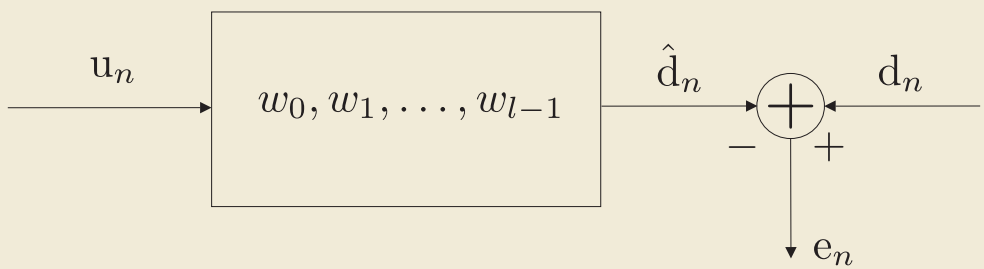
- Linear statistical filtering is an instance of the general estimation task, when the notion of (time) evolution needs to be taken into consideration and estimates are obtained at each time instant. There are three major types of problems that emerge:
  - Filtering, where the estimate at time instant n is based on all previously received (measured) input information up to and including the current time index, n.
  - Smoothing, where data over a time interval, $[0, N]$, are first collected and an estimate is obtained at each time instant $n \leq N$, using all the available information in the interval $[0, N]$.
  - Prediction, where estimates at times $n + \tau,\ \tau > 0$ are to be obtained based on the information up to and including time instant n.
- Take for example a time-varying case, where the output variable, at time instant n, is y_n and its value depends on observations included in the corresponding input vector $\mathbf{x}_n$. In filtering, the latter can include measurements received only at time instants, $n, n-1, \ldots, 0$. This restriction in the index set, is directly related to **causality**. In contrast, in smoothing, we can also include future time instants, e.g., $n+2, n+1, n, n-1$.

Linear Filtering

- In Signal Processing, the term filtering is usually used in a more specific context, and it refers to the operation of a filter, which acts on an input random process/signal (u_n), to transform it into another one (d_n). Note that we have changed the notation, to stress out that we talk about processes and not random variables, in general.
- The task in statistical linear filtering is to compute the coefficients (impulse response) of the filter so that the output process of the filter, $\hat{\mathrm{d}}_n$, when the filter is excited by the input random process, u_n, to be as close as possible, to a desired response process, d_n. In other words, the goal is to minimize, in some sense, the corresponding error process. This is illustrated in the figure below.

Linear Filtering

- In Signal Processing, the term filtering is usually used in a more specific context, and it refers to the operation of a filter, which acts on an input random process/signal (u_n), to transform it into another one (d_n). Note that we have changed the notation, to stress out that we talk about processes and not random variables, in general.
- The task in statistical linear filtering is to compute the coefficients (impulse response) of the filter so that the output process of the filter, $\hat{\mathrm{d}}_n$, when the filter is excited by the input random process, u_n, to be as close as possible, to a desired response process, d_n. In other words, the goal is to minimize, in some sense, the corresponding error process. This is illustrated in the figure below.



In statistical filtering, the impulse response coefficients are estimated so as the error between the output and the desired response process to be minimised. In MSE linear filtering, the cost function is $\mathbb{E}[\mathrm{e}_n^2]$.

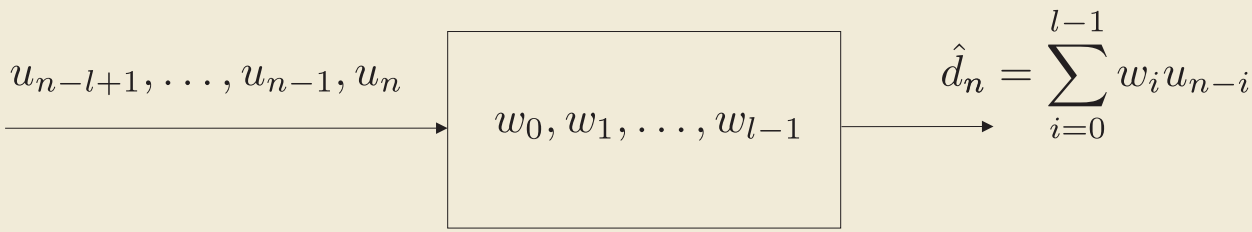
Linear Filtering

- Assuming that the unknown filter is of a finite impulse response (FIR) , denoted as $w_0, w_1, \ldots, w_{l-1}$, the output $\hat{\mathrm{d}}_n$ of the filter is given as

$$\hat{\mathrm{d}}_n = \sum_{i=0}^{l-1} w_i \mathrm{u}_{n-i} = \boldsymbol{w}^T \mathbf{u}_n \tag{3}$$

where,

$$\boldsymbol{w} = [w_0, w_1, ..., w_{l-1}]^T, \text{ and } \mathbf{u}_n = [\mathrm{u}_n, \mathrm{u}_{n-1}, ..., \mathrm{u}_{n-l+1}]^T.$$



The linear filter is excited by a realization of an input process. The output signal is the convolution between the input sequence and the filter's impulse response.

- Alternatively, Eq. (3) can be viewed as the linear estimator function; given the jointly distributed variables, at time instant n, $(\mathrm{d}_n, \mathbf{u}_n)$, Eq. (3) provides the estimator, $\hat{\mathrm{d}}_n$, given the values of $\mathbf{u}_n$. In order to obtain the coefficients, $\boldsymbol{w}$, the mean-square error criterion will be adopted.

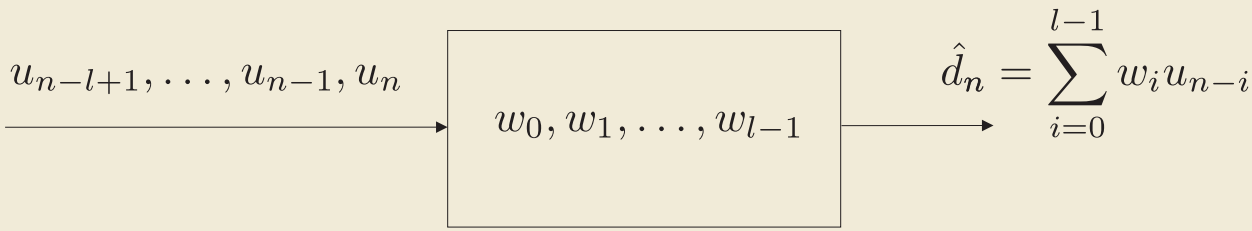
Linear Filtering

- Assuming that the unknown filter is of a finite impulse response (FIR) , denoted as $w_0, w_1, \ldots, w_{l-1}$, the output $\hat{\mathrm{d}}_n$ of the filter is given as

$$\hat{\mathrm{d}}_n = \sum_{i=0}^{l-1} w_i \mathrm{u}_{n-i} = \boldsymbol{w}^T \mathbf{u}_n \tag{3}$$

where,

$$\boldsymbol{w} = [w_0, w_1, ..., w_{l-1}]^T, \text{ and } \mathbf{u}_n = [\mathrm{u}_n, \mathrm{u}_{n-1}, ..., \mathrm{u}_{n-l+1}]^T.$$



The linear filter is excited by a realization of an input process. The output signal is the convolution between the input sequence and the filter's impulse response.

- Alternatively, Eq. (3) can be viewed as the linear estimator function; given the jointly distributed variables, at time instant n, $(\mathrm{d}_n, \mathbf{u}_n)$, Eq. (3) provides the estimator, $\hat{\mathrm{d}}_n$, given the values of $\mathbf{u}_n$. In order to obtain the coefficients, $\boldsymbol{w}$, the mean-square error criterion will be adopted.

Linear Filtering

- Let us now assume that:
  - The processes, $\mathrm{u}_n,\ \mathrm{d}_n$ are wide-sense stationary real random processes.
  - Their mean values are equal to zero, i.e., $\mathbb{E}[\mathrm{u}_n] = \mathbb{E}[\mathrm{d}_n] = 0,\ \forall n$. If this is not the case, we can subtract the respective mean values from the processes, u_n and d_n, during a preprocessing stage. Due to this assumption, the autocorrelation and covariance matrices of $\mathbf{u}_n$ coincide, i.e., $R_u = \Sigma_u$.
- The normal equations now take the form

$$\Sigma_u \boldsymbol{w} = \boldsymbol{p}, \quad \boldsymbol{p} = \left[\mathbb{E}[\mathrm{u}_n \mathrm{d}_n], ..., \mathbb{E}[\mathrm{u}_{n-l+1} \mathrm{d}_n]\right]^T,$$

and the respective covariance/autocorrelation matrix, of order l, of the input process is given by,

$$\Sigma_u := \mathbb{E}[\mathbf{u}_n \mathbf{u}_n^T] = \begin{bmatrix} r(0) & r(1) & \dots & r(l-1) \\ r(1) & r(0) & \dots & r(l-2) \\ \vdots & & \ddots & \\ r(l-1) & r(l-2) & \dots & r(0) \end{bmatrix},$$

where $r(k)$ is the autocorrelation sequence of the input process.

- Let us now assume that:
  - The processes, $\mathrm{u}_n,\ \mathrm{d}_n$ are wide-sense stationary real random processes.
  - Their mean values are equal to zero, i.e., $\mathbb{E}[\mathrm{u}_n] = \mathbb{E}[\mathrm{d}_n] = 0,\ \forall n$. If this is not the case, we can subtract the respective mean values from the processes, u_n and d_n, during a preprocessing stage. Due to this assumption, the autocorrelation and covariance matrices of $\mathbf{u}_n$ coincide, i.e., $R_u = \Sigma_u$.
- The normal equations now take the form

$$\Sigma_u \boldsymbol{w} = \boldsymbol{p}, \quad \boldsymbol{p} = \left[\mathbb{E}[\mathrm{u}_n \mathrm{d}_n], ..., \mathbb{E}[\mathrm{u}_{n-l+1} \mathrm{d}_n]\right]^T,$$

and the respective covariance/autocorrelation matrix, of order l, of the input process is given by,

$$\Sigma_u := \mathbb{E}[\mathbf{u}_n \mathbf{u}_n^T] = \begin{bmatrix} r(0) & r(1) & \dots & r(l-1) \\ r(1) & r(0) & \dots & r(l-2) \\ \vdots & & \ddots & \\ r(l-1) & r(l-2) & \dots & r(0) \end{bmatrix},$$

where $r(k)$ is the autocorrelation sequence of the input process.

MSE Linear Filtering: A Frequency-Domain Point of View

- Let us now turn our attention to the more general case, and assume that our filter is of infinite impulse response (IIR). Then, the input-output relation becomes,

$$\hat{\mathrm{d}}_n = \sum_{i=-\infty}^{+\infty} w_i \mathrm{u}_{n-i}.$$

Moreover, we have allowed the filter to be non-causal. Recall that a system is called causal if the output, $\hat{d}_n$, **does** depend **only** on input values $u_m,\ m \leq n$. A necessary and sufficient condition for causality is that, the impulse response is zero for negative time instants, i.e., $w_n = 0,\ n < 0$..

- Following similar arguments as those used to prove the MSE optimality of $\mathbb{E}[\mathrm{y}|\boldsymbol{x}]$ (Chapter 3), it turns out that the optimal filter coefficients must satisfy the following condition,

$$\mathbb{E}\big[(\mathrm{d}_n - \sum_{i=-\infty}^{+\infty} w_i \mathrm{u}_{n-i})\mathrm{u}_{n-j}\big] = 0,\ j \in \mathbb{Z}.$$

Observe that this is a generalization of the orthogonality condition stated before.

MSE Linear Filtering: A Frequency-Domain Point of View

- Let us now turn our attention to the more general case, and assume that our filter is of infinite impulse response (IIR). Then, the input-output relation becomes,

$$\hat{\mathrm{d}}_n = \sum_{i=-\infty}^{+\infty} w_i \mathrm{u}_{n-i}.$$

Moreover, we have allowed the filter to be non-causal. Recall that a system is called causal if the output, $\hat{d}_n$, **does** depend **only** on input values $u_m,\ m \leq n$. A necessary and sufficient condition for causality is that, the impulse response is zero for negative time instants, i.e., $w_n = 0,\ n < 0$..

- Following similar arguments as those used to prove the MSE optimality of $\mathbb{E}[\mathrm{y}|\boldsymbol{x}]$ (Chapter 3), it turns out that the optimal filter coefficients must satisfy the following condition,

$$\mathbb{E}\Big[(\mathrm{d}_n - \sum_{i=-\infty}^{+\infty} w_i \mathrm{u}_{n-i})\mathrm{u}_{n-j}\Big] = 0,\ j \in \mathbb{Z}.$$

Observe that this is a generalization of the orthogonality condition stated before.

MSE Linear Filtering: A Frequency-Domain Point of View

- A rearrangement of the terms in the previous equation results in

$$\sum_{i=-\infty}^{+\infty} w_i \mathbb{E}[\mathrm{u}_{n-i}\mathrm{u}_{n-j}] = \mathbb{E}[\mathrm{d}_n \mathrm{u}_{n-j}], \ j \in \mathbb{Z},$$

and finally to,

$$\sum_{i=-\infty}^{+\infty} w_i r(j-i) = r_{du}(j), \ j \in \mathbb{Z}.$$

- The above can be considered as the generalization of the normal equations, $\Sigma_x \theta = p$, involving an infinite set of parameters. The way to solve it is to cross into the frequency domain. Indeed, this can be seen as the convolution of the unknown sequence (w_i) with the autocorrelation sequence of the input process, which gives rise to the cross-correlation sequence. However, we know that convolution of two sequences corresponds to the **product** of the respective Fourier transforms. Thus, we can now write that

$$W(\omega) S_u(\omega) = S_{du}(\omega), \tag{4}$$

where $W(\omega)$ is the Fourier transform of w_i and $S_u(\omega)$ is the power spectral density of the input process. In analogy, the Fourier transform, $S_{du}(\omega)$, of the cross-correlation sequence is known as the cross-spectral density.

- If the latter two quantities are available, then once $W(\omega)$ has been computed, the unknown parameters can be obtained via the inverse Fourier transform.

MSE Linear Filtering: A Frequency-Domain Point of View

- A rearrangement of the terms in the previous equation results in

$$\sum_{i=-\infty}^{+\infty} w_i \mathbb{E}[\mathrm{u}_{n-i}\mathrm{u}_{n-j}] = \mathbb{E}[\mathrm{d}_n \mathrm{u}_{n-j}], \ j \in \mathbb{Z},$$

  and finally to,

$$\sum_{i=-\infty}^{+\infty} w_i r(j-i) = r_{du}(j), \ j \in \mathbb{Z}.$$

- The above can be considered as the generalization of the normal equations, $\Sigma_x \boldsymbol{\theta} = \boldsymbol{p}$, involving an infinite set of parameters. The way to solve it is to cross into the frequency domain. Indeed, this can be seen as the convolution of the unknown sequence (w_i) with the autocorrelation sequence of the input process, which gives rise to the cross-correlation sequence. However, we know that convolution of two sequences corresponds to the **product** of the respective Fourier transforms. Thus, we can now write that

$$W(\omega) S_u(\omega) = S_{du}(\omega), \tag{4}$$

  where $W(\omega)$ is the Fourier transform of w_i and $S_u(\omega)$ is the power spectral density of the input process. In analogy, the Fourier transform, $S_{du}(\omega)$, of the cross-correlation sequence is known as the cross-spectral density.

- If the latter two quantities are available, then once $W(\omega)$ has been computed, the unknown parameters can be obtained via the inverse Fourier transform.

MSE Linear Filtering: A Frequency-Domain Point of View

- A rearrangement of the terms in the previous equation results in

$$\sum_{i=-\infty}^{+\infty} w_i \mathbb{E}[\mathrm{u}_{n-i}\mathrm{u}_{n-j}] = \mathbb{E}[\mathrm{d}_n \mathrm{u}_{n-j}], \ j \in \mathbb{Z},$$

  and finally to,

$$\sum_{i=-\infty}^{+\infty} w_i r(j-i) = r_{du}(j), \ j \in \mathbb{Z}.$$

- The above can be considered as the generalization of the normal equations, $\Sigma_x \boldsymbol{\theta} = \boldsymbol{p}$, involving an infinite set of parameters. The way to solve it is to cross into the frequency domain. Indeed, this can be seen as the convolution of the unknown sequence (w_i) with the autocorrelation sequence of the input process, which gives rise to the cross-correlation sequence. However, we know that convolution of two sequences corresponds to the **product** of the respective Fourier transforms. Thus, we can now write that

$$W(\omega) S_u(\omega) = S_{du}(\omega), \tag{4}$$

  where $W(\omega)$ is the Fourier transform of w_i and $S_u(\omega)$ is the power spectral density of the input process. In analogy, the Fourier transform, $S_{du}(\omega)$, of the cross-correlation sequence is known as the cross-spectral density.

- If the latter two quantities are available, then once $W(\omega)$ has been computed, the unknown parameters can be obtained via the inverse Fourier transform.

MSE Linear Filtering: A Frequency-Domain Point of View

- A rearrangement of the terms in the previous equation results in

$$\sum_{i=-\infty}^{+\infty} w_i \mathbb{E}[\mathrm{u}_{n-i}\mathrm{u}_{n-j}] = \mathbb{E}[\mathrm{d}_n \mathrm{u}_{n-j}], \ j \in \mathbb{Z},$$

  and finally to,

$$\sum_{i=-\infty}^{+\infty} w_i r(j-i) = r_{du}(j), \ j \in \mathbb{Z}.$$

- The above can be considered as the generalization of the normal equations, $\Sigma_x \boldsymbol{\theta} = \boldsymbol{p}$, involving an infinite set of parameters. The way to solve it is to cross into the frequency domain. Indeed, this can be seen as the convolution of the unknown sequence (w_i) with the autocorrelation sequence of the input process, which gives rise to the cross-correlation sequence. However, we know that convolution of two sequences corresponds to the **product** of the respective Fourier transforms. Thus, we can now write that

$$W(\omega) S_u(\omega) = S_{du}(\omega), \tag{4}$$

  where $W(\omega)$ is the Fourier transform of w_i and $S_u(\omega)$ is the power spectral density of the input process. In analogy, the Fourier transform, $S_{du}(\omega)$, of the cross-correlation sequence is known as the cross-spectral density.
- If the latter two quantities are available, then once $W(\omega)$ has been computed, the unknown parameters can be obtained via the inverse Fourier transform.

Deconvolution: Image Deblurring

- We will now consider an important application in order to demonstrate the power of MSE linear estimation. Image deblurring is a typical deconvolution task. An image is degraded due to its transmission via a nonideal system; the task of deconvolution is to optimally recover (in the MSE sense in our case), the original undegraded one. Figure (a) shows the original image and Figure (b) a blurred version (e.g., taken by a non-steady camera) with some small additive noise.

(a)

(b)

- Deconvolution is a process that our human brain performs all the time. The human vision system is one of the most complex and highly developed biological systems. Any raw image that falls on the retina of the eye is severely blurred. Thus, one of the main early processing activities of our visual system is to deblur it.

Deconvolution: Image Deblurring

- We will now consider an important application in order to demonstrate the power of MSE linear estimation. Image deblurring is a typical deconvolution task. An image is degraded due to its transmission via a nonideal system; the task of deconvolution is to optimally recover (in the MSE sense in our case), the original undegraded one. Figure (a) shows the original image and Figure (b) a blurred version (e.g., taken by a non-steady camera) with some small additive noise.

(a)

(b)

- Deconvolution is a process that our human brain performs all the time. The human vision system is one of the most complex and highly developed biological systems. Any raw image that falls on the retina of the eye is severely blurred. Thus, one of the main early processing activities of our visual system is to deblur it.

Deconvolution: Image Deblurring

- Before we proceed any further, the following assumptions are adopted:
  - The image is a wide-sense stationary two-dimensional random process. Two-dimensional random processes are also known as random fields, discussed in Chapter 15.
  - The image is of an infinite extent; this can be justified for the case of large images. This assumption will grant us the "permission" to use (4). The fact that an image is a two-dimensional process does not change anything in the theoretical analysis; the only difference is that now the Fourier transforms involve two frequency variables, ω_1, ω_2, one for each of the two dimensions.
- A gray image is represented as a two-dimensional array. To stay close to the notation used so far, let $d(n,m),\ n,m \in \mathbb{Z}$ be the original undegraded image (which for us is now the desired response) and $u(n,m),\ n,m \in \mathbb{Z}$ be the degraded one, obtained as

$$u(n,m) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i,j) d(n-i, m-j) + \eta(n,m),$$

where $\eta(n,m)$ is the realization of a noise field, which is assumed to be zero mean and independent of the input (undegraded) image. The sequence $h(i,j)$ is the point spread sequence (impulse response) of the system (e.g., camera). We will assume that this is known and it has, somehow, been measured.

Deconvolution: Image Deblurring

- Before we proceed any further, the following assumptions are adopted:
  - The image is a wide-sense stationary two-dimensional random process. Two-dimensional random processes are also known as random fields, discussed in Chapter 15.
  - The image is of an infinite extent; this can be justified for the case of large images. This assumption will grant us the "permission" to use (4). The fact that an image is a two-dimensional process does not change anything in the theoretical analysis; the only difference is that now the Fourier transforms involve two frequency variables, ω_1, ω_2, one for each of the two dimensions.
- A gray image is represented as a two-dimensional array. To stay close to the notation used so far, let $d(n,m),\ n,m \in \mathbb{Z}$ be the original undegraded image (which for us is now the desired response) and $u(n,m),\ n,m \in \mathbb{Z}$ be the degraded one, obtained as

$$u(n,m) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i,j) d(n-i, m-j) + \eta(n,m),$$

where $\eta(n,m)$ is the realization of a noise field, which is assumed to be zero mean and independent of the input (undegraded) image. The sequence $h(i,j)$ is the point spread sequence (impulse response) of the system (e.g., camera). We will assume that this is known and it has, somehow, been measured.

Deconvolution: Image Deblurring

- Our task now is to estimate a two-dimensional filter, $w(n, m)$, which is applied to the degraded image to optimally reconstruct (in the MSE sense) the original undegraded one. In the current context, Eq. (4) is written as,
$$W(\omega_1, \omega_2) S_u(\omega_1, \omega_2) = S_{du}(\omega_1, \omega_2).$$
Furthermore, it can be shown (as in respective section in Chapter 2) that,
$$S_{du}(\omega_1, \omega_2) = H^*(\omega_1, \omega_2) S_d(\omega_1, \omega_2),$$
and
$$S_u(\omega_1, \omega_2) = |H(\omega_1, \omega_2)|^2 S_d(\omega_1, \omega_2) + S_\eta(\omega_1, \omega_2),$$
where "*" denotes complex conjugation and S_η is the power spectral density of the noise field. Thus, we finally obtain that
$$W(\omega_1, \omega_2) = \frac{1}{H(\omega_1, \omega_2)} \frac{|H(\omega_1, \omega_2)|^2}{|H(\omega_1, \omega_2)|^2 + \frac{S_\eta(\omega_1, \omega_2)}{S_d(\omega_1, \omega_2)}}$$
- Once $W(\omega_1, \omega_2)$ has been computed, the unknown parameters could be obtained via an inverse (two-dimensional) Fourier transform.

Deconvolution: Image Deblurring

- Our task now is to estimate a two-dimensional filter, $w(n,m)$, which is applied to the degraded image to optimally reconstruct (in the MSE sense) the original undegraded one. In the current context, Eq. (4) is written as,
$$W(\omega_1,\omega_2)S_u(\omega_1,\omega_2) = S_{du}(\omega_1,\omega_2).$$
Furthermore, it can be shown (as in respective section in Chapter 2) that,
$$S_{du}(\omega_1,\omega_2) = H^*(\omega_1,\omega_2)S_d(\omega_1,\omega_2),$$
and
$$S_u(\omega_1,\omega_2) = |H(\omega_1,\omega_2)|^2 S_d(\omega_1,\omega_2) + S_\eta(\omega_1,\omega_2),$$
where "*" denotes complex conjugation and S_η is the power spectral density of the noise field. Thus, we finally obtain that
$$W(\omega_1,\omega_2) = \frac{1}{H(\omega_1,\omega_2)} \frac{|H(\omega_1,\omega_2)|^2}{|H(\omega_1,\omega_2)|^2 + \frac{S_\eta(\omega_1,\omega_2)}{S_d(\omega_1,\omega_2)}}$$
- Once $W(\omega_1,\omega_2)$ has been computed, the unknown parameters could be obtained via an inverse (two-dimensional) Fourier transform.

Deconvolution: Image Deblurring

- The deblurred image then results as

$$\hat{d}(n,m) = \sum_{i=-\infty}^{+\infty} w(i,j)u(n-i,m-j).$$

In practice, since we are not really interested in obtaining the weights of the deconvolution filter, the above convolution is implemented in the frequency domain, i.e.,

$$\hat{D}(\omega_1,\omega_2) = W(\omega_1,\omega_2)U(\omega_1,\omega_2),$$

and then obtain the inverse Fourier transform. Thus, all the processing is efficiently performed in the frequency domain. Software packages to perform Fourier transforms (via the Fast Fourier Transform, FFT) of an image array are "omnipresent" in the internet.

- Another important issue is that, in practice, we do not know $S_d(\omega_1,\omega_2)$. An approximation, which is usually adopted that renders sensible results, is to assume that $\frac{S_\eta(\omega_1,\omega_2)}{S_d(\omega_1,\omega_2)}$ is a constant, C. Then, one tries different values of it and selects the one that results in the best reconstructed image.

Deconvolution: Image Deblurring

- The deblurred image then results as

$$\hat{d}(n,m) = \sum_{i=-\infty}^{+\infty} w(i,j)u(n-i,m-j).$$

  In practice, since we are not really interested in obtaining the weights of the deconvolution filter, the above convolution is implemented in the frequency domain, i.e.,

$$\hat{D}(\omega_1,\omega_2) = W(\omega_1,\omega_2)U(\omega_1,\omega_2),$$

  and then obtain the inverse Fourier transform. Thus, all the processing is efficiently performed in the frequency domain. Software packages to perform Fourier transforms (via the Fast Fourier Transform, FFT) of an image array are "omnipresent" in the internet.

- Another important issue is that, in practice, we do not know $S_d(\omega_1,\omega_2)$. An approximation, which is usually adopted that renders sensible results, is to assume that $\frac{S_\eta(\omega_1,\omega_2)}{S_d(\omega_1,\omega_2)}$ is a constant, C. Then, one tries different values of it and selects the one that results in the best reconstructed image.

Deconvolution: Image Deblurring

- Figure (a) shows the deblurred image for $C = 2.3 \times 10^{-6}$, alongside the original one shown in Figure (b). The quality of the end result depends a lot on the choice of this value.

(a)

(b)

a) the deblurred image for $C = 2.3 \times 10^{-6}$, and b) the original one. Observe that in spite of the simplicity of the method, the reconstruction is pretty good. The differences become more obvious to the eye, when the images are enlarged.

- Needless to say that, other, more advanced techniques, have also been proposed. For example, one can get a better estimate of $S_d(\omega_1, \omega_2)$ by using information from $S_\eta(\omega_1, \omega_2)$ and $S_u(\omega_1, \omega_2)$.

Deconvolution: Image Deblurring

- Figure (a) shows the deblurred image for $C = 2.3 \times 10^{-6}$, alongside the original one shown in Figure (b). The quality of the end result depends a lot on the choice of this value.

(a)

(b)

a) the deblurred image for $C = 2.3 \times 10^{-6}$, and b) the original one. Observe that in spite of the simplicity of the method, the reconstruction is pretty good. The differences become more obvious to the eye, when the images are enlarged.

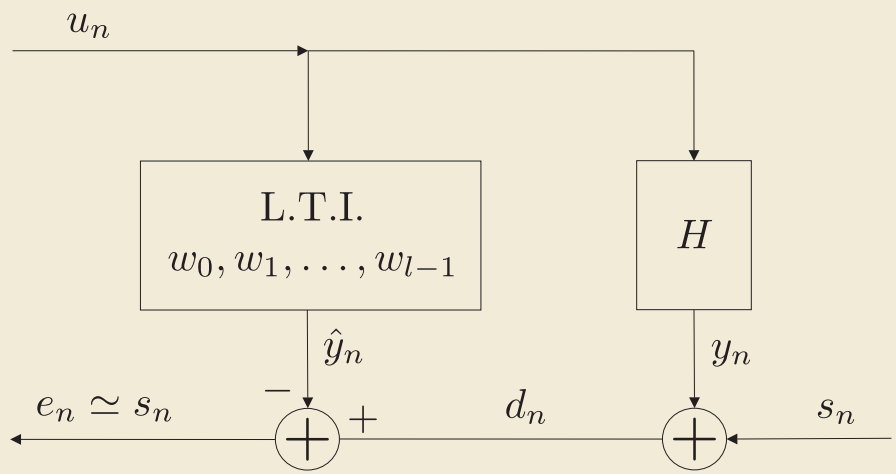
- Needless to say that, other, more advanced techniques, have also been proposed. For example, one can get a better estimate of $S_d(\omega_1, \omega_2)$ by using information from $S_\eta(\omega_1, \omega_2)$ and $S_u(\omega_1, \omega_2)$.

Interference Cancellation

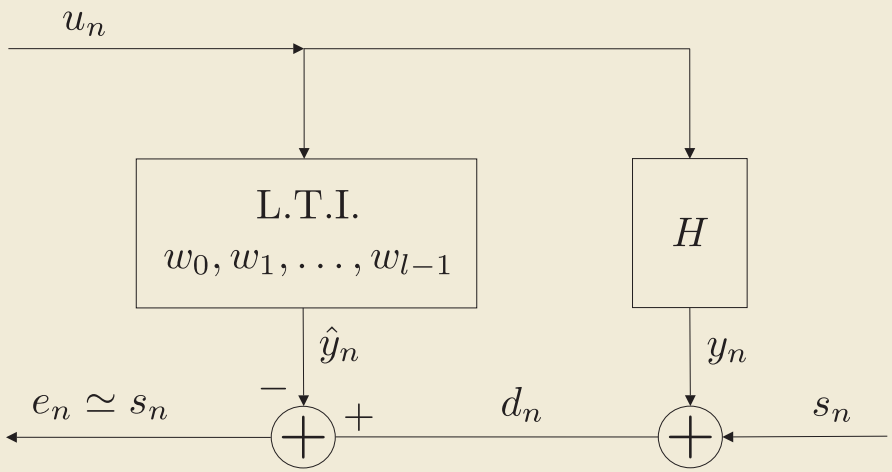
- In interference cancellation, we have access to a mixture of two signals, e.g., $d_n = y_n + s_n$. Ideally, we would like to remove one of them, say y_n. We will consider them as realizations of respective random processes/signals, i.e., d_n, y_n and s_n. To achieve this goal, the only available information is another signal, say u_n, which is statistically related to the unwanted signal, y_n. For example, y_n may be a filtered version of u_n.
- This is illustrated in the figure below, where the corresponding realizations of the involved processes are shown, on which a real system works on.

Interference Cancellation

- In interference cancellation, we have access to a mixture of two signals, e.g., $d_n = y_n + s_n$. Ideally, we would like to remove one of them, say y_n. We will consider them as realizations of respective random processes/signals, i.e., d_n, y_n and s_n. To achieve this goal, the only available information is another signal, say u_n, which is statistically related to the unwanted signal, y_n. For example, y_n may be a filtered version of u_n.
- This is illustrated in the figure below, where the corresponding realizations of the involved processes are shown, on which a real system works on.



Interference Cancellation

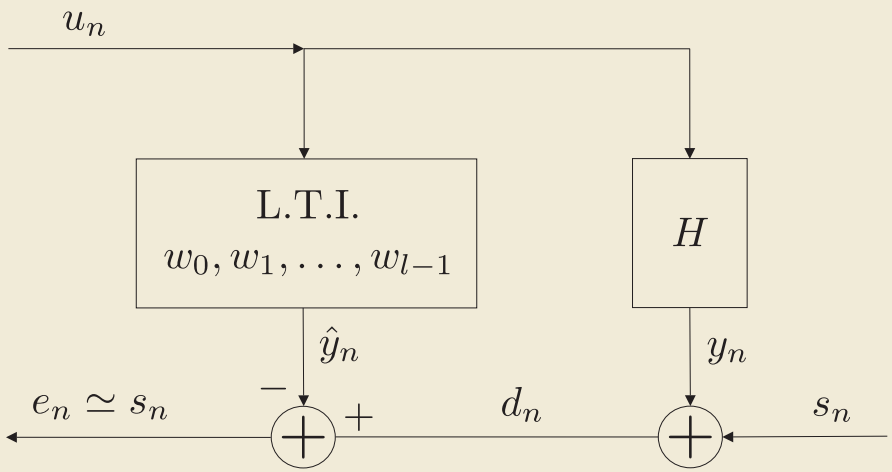


- Process y_n is the output of an unknown system H, whose input is excited by u_n. The task is to model H by obtaining estimates of its impulse response (assuming that it is LTI and of known order). Then, the output of the model will be an approximation of y_n, when this is activated by the same input, u_n. We will use d_n as the desired response process.

- The optimal estimates of $w_0, ..., w_{l-1}$ (assuming the order of the unknown system H to be l) are provided by the normal equations

$$\Sigma_u \boldsymbol{w}_* = \boldsymbol{p}.$$

Interference Cancellation



- Process y_n is the output of an unknown system H, whose input is excited by u_n. The task is to model H by obtaining estimates of its impulse response (assuming that it is LTI and of known order). Then, the output of the model will be an approximation of y_n, when this is activated by the same input, u_n. We will use d_n as the desired response process.

- The optimal estimates of $w_0, ..., w_{l-1}$ (assuming the order of the unknown system H to be l) are provided by the normal equations

$$\Sigma_u \boldsymbol{w}_* = \boldsymbol{p}.$$

Interference Cancellation

- However,
$$\boldsymbol{p} = \mathbb{E}\left[\mathbf{u}_n \mathrm{d}_n\right] = \mathbb{E}\left[\mathbf{u}_n\left(\mathrm{y}_n + \mathrm{s}_n\right)\right] = \mathbb{E}\left[\mathbf{u}_n \mathrm{y}_n\right],$$
since the respective input vector $\mathbf{u}_n$ and s_n are considered statistically independent.
- That is, the previous formulation of the problem leads to the same normal equations as if the desired response was the signal y_n, **which we want to remove**! Hence, the output of our model will be an approximation (in the MSE sense), $\hat{\mathrm{y}}_n$, of y_n and if subtracted from d_n the resulting error signal, e_n, will be an approximation to s_n.
- How good this approximation is depends on whether l is a good "estimate" of the true order of H. The cross-correlation in the right hand side of the normal equations can be approximated by computing the respective sample mean vales, in particular over periods where s_n is absent. In practical systems, adaptive/online versions of this implementation are usually employed, as those discussed in Chapter 5.
- Interference cancellation schemes are used in many systems such as noise cancellation, echo cancellation in telephone networks and video conferencing, and in biomedical applications; for example, in order to cancel the maternal interference in fetal electrocardiograph.

Interference Cancellation

- However,
$$\boldsymbol{p} = \mathbb{E}\left[\mathbf{u}_n \mathrm{d}_n\right] = \mathbb{E}\left[\mathbf{u}_n\left(\mathrm{y}_n + \mathrm{s}_n\right)\right] = \mathbb{E}\left[\mathbf{u}_n \mathrm{y}_n\right],$$
since the respective input vector $\mathbf{u}_n$ and s_n are considered statistically independent.
- That is, the previous formulation of the problem leads to the same normal equations as if the desired response was the signal y_n, **which we want to remove**! Hence, the output of our model will be an approximation (in the MSE sense), $\hat{\mathrm{y}}_n$, of y_n and if subtracted from d_n the resulting error signal, e_n, will be an approximation to s_n.
- How good this approximation is depends on whether l is a good "estimate" of the true order of H. The cross-correlation in the right hand side of the normal equations can be approximated by computing the respective sample mean vales, in particular over periods where s_n is absent. In practical systems, adaptive/online versions of this implementation are usually employed, as those discussed in Chapter 5.
- Interference cancellation schemes are used in many systems such as noise cancellation, echo cancellation in telephone networks and video conferencing, and in biomedical applications; for example, in order to cancel the maternal interference in fetal electrocardiograph.

Interference Cancellation

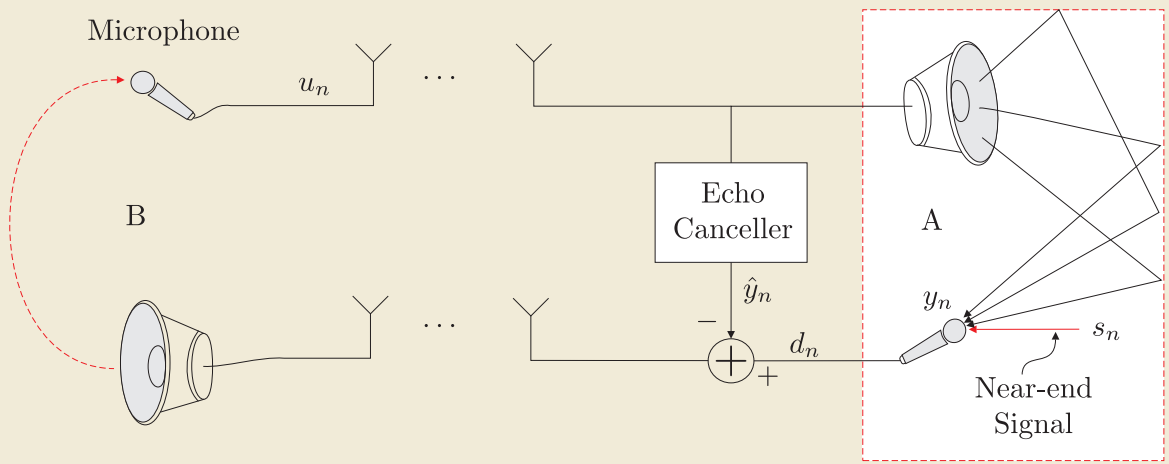
- However,
$$\boldsymbol{p} = \mathbb{E}\left[\mathbf{u}_n \mathrm{d}_n\right] = \mathbb{E}\left[\mathbf{u}_n\left(\mathrm{y}_n + \mathrm{s}_n\right)\right] = \mathbb{E}\left[\mathbf{u}_n \mathrm{y}_n\right],$$
since the respective input vector $\mathbf{u}_n$ and s_n are considered statistically independent.
- That is, the previous formulation of the problem leads to the same normal equations as if the desired response was the signal y_n, **which we want to remove**! Hence, the output of our model will be an approximation (in the MSE sense), $\hat{\mathrm{y}}_n$, of y_n and if subtracted from d_n the resulting error signal, e_n, will be an approximation to s_n.
- How good this approximation is depends on whether l is a good "estimate" of the true order of H. The cross-correlation in the right hand side of the normal equations can be approximated by computing the respective sample mean vales, in particular over periods where s_n is absent. In practical systems, adaptive/online versions of this implementation are usually employed, as those discussed in Chapter 5.
- Interference cancellation schemes are used in many systems such as noise cancellation, echo cancellation in telephone networks and video conferencing, and in biomedical applications; for example, in order to cancel the maternal interference in fetal electrocardiograph.

Interference Cancellation

- However,
$$\boldsymbol{p} = \mathbb{E}\left[\mathbf{u}_n \mathrm{d}_n\right] = \mathbb{E}\left[\mathbf{u}_n\left(\mathrm{y}_n + \mathrm{s}_n\right)\right] = \mathbb{E}\left[\mathbf{u}_n \mathrm{y}_n\right],$$
since the respective input vector $\mathbf{u}_n$ and s_n are considered statistically independent.
- That is, the previous formulation of the problem leads to the same normal equations as if the desired response was the signal y_n, **which we want to remove**! Hence, the output of our model will be an approximation (in the MSE sense), $\hat{\mathrm{y}}_n$, of y_n and if subtracted from d_n the resulting error signal, e_n, will be an approximation to s_n.
- How good this approximation is depends on whether l is a good "estimate" of the true order of H. The cross-correlation in the right hand side of the normal equations can be approximated by computing the respective sample mean vales, in particular over periods where s_n is absent. In practical systems, adaptive/online versions of this implementation are usually employed, as those discussed in Chapter 5.
- Interference cancellation schemes are used in many systems such as noise cancellation, echo cancellation in telephone networks and video conferencing, and in biomedical applications; for example, in order to cancel the maternal interference in fetal electrocardiograph.

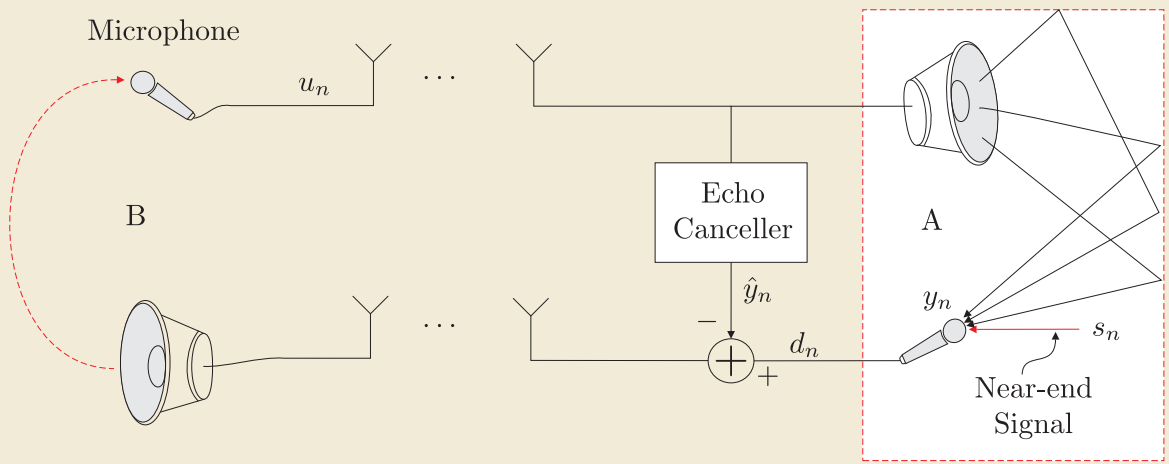
Echo Cancellation In Video Conferencing

- The task of echo cancelation in video conferencing is illustrated in the figure below. The same set up applies to the hands-free telephone service in a car.
- The far-end speech signal is considered to be a realization of a random process, u_n; through the loudspeakers, it is broadcasted in room A (car) and it is reflected in the interior of the room. Part of it is absorbed and part of it enters the microphone; this is denoted as y_n.
- The equivalent response of the room (reflections) on u_n can be represented by a filter, H, as in the interference cancelation task before. Signal y_n returns back and the speaker in location B listens her/his own voice, together with the near-end speech signal, s_n of the speaker in A. The goal of the echo canceller is to optimally remove y_n.



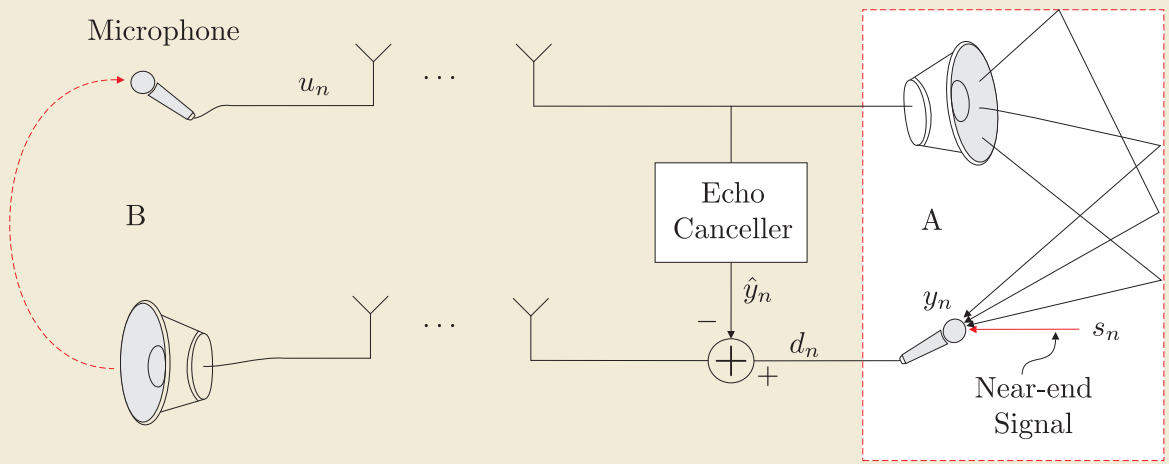
Echo Cancellation In Video Conferencing

- The task of echo cancelation in video conferencing is illustrated in the figure below. The same set up applies to the hands-free telephone service in a car.
- The far-end speech signal is considered to be a realization of a random process, u_n; through the loudspeakers, it is broadcasted in room A (car) and it is reflected in the interior of the room. Part of it is absorbed and part of it enters the microphone; this is denoted as y_n.
- The equivalent response of the room (reflections) on u_n can be represented by a filter, H, as in the interference cancelation task before. Signal y_n returns back and the speaker in location B listens her/his own voice, together with the near-end speech signal, s_n of the speaker in A. The goal of the echo canceller is to optimally remove y_n.



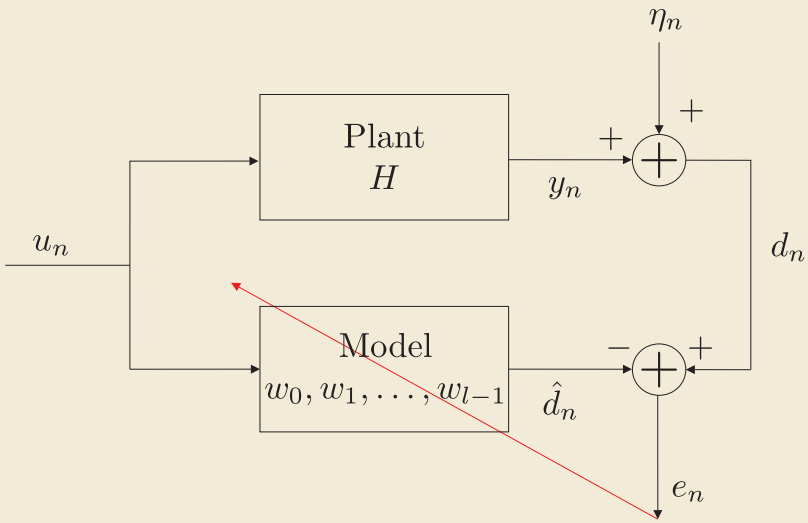
Echo Cancellation In Video Conferencing

- The task of echo cancelation in video conferencing is illustrated in the figure below. The same set up applies to the hands-free telephone service in a car.
- The far-end speech signal is considered to be a realization of a random process, u_n; through the loudspeakers, it is broadcasted in room A (car) and it is reflected in the interior of the room. Part of it is absorbed and part of it enters the microphone; this is denoted as y_n.
- The equivalent response of the room (reflections) on u_n can be represented by a filter, H, as in the interference cancelation task before. Signal y_n returns back and the speaker in location B listens her/his own voice, together with the near-end speech signal, s_n of the speaker in A. The goal of the echo canceller is to optimally remove y_n.



System Identification

- System identification is similar in nature to the interference cancellation task. However, in the latter task, the focus was on replicating the output y_n. In contrast, in the system identification the focus is on the system's impulse response.

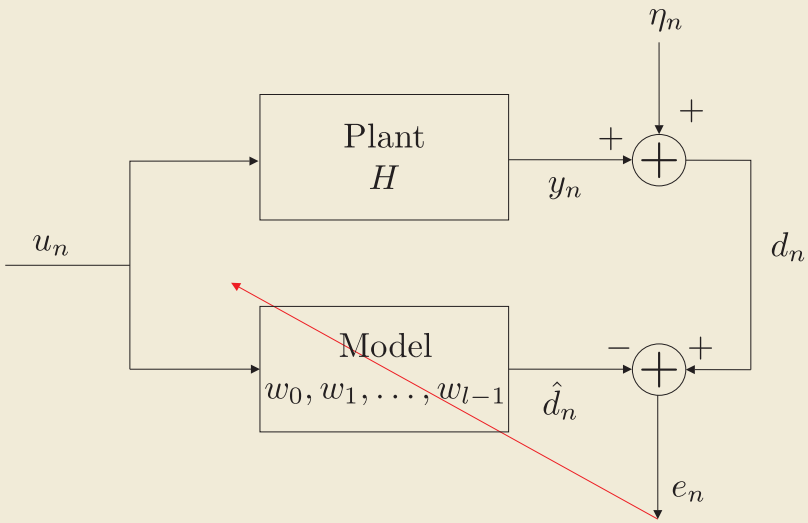


In system identification, the impulse response of the model is optimally estimated so that the output to be close, in the MSE, to that of the unknown plant. The red line indicates that the error is used for the optimal estimation of the unknown parameters of the filter.

- In system identification, the aim is to model the impulse response of an unknown plant. To this end, we have access to its input signal as well as to a noisy version of its output. The task is to design a model whose impulse response approximates that of the unknown plant. To achieve this, we optimally design a linear filter whose input is the same signal as the one that activates the plant and its desired response is the **noisy output** of the plant, as shown in the figure above.

System Identification

- System identification is similar in nature to the interference cancellation task. However, in the latter task, the focus was on replicating the output y_n. In contrast, in the system identification the focus is on the system's impulse response.



In system identification, the impulse response of the model is optimally estimated so that the output to be close, in the MSE, to that of the unknown plant. The red line indicates that the error is used for the optimal estimation of the unknown parameters of the filter.

- In system identification, the aim is to model the impulse response of an unknown plant. To this end, we have access to its input signal as well as to a noisy version of its output. The task is to design a model whose impulse response approximates that of the unknown plant. To achieve this, we optimally design a linear filter whose input is the same signal as the one that activates the plant and its desired response is the **noisy output** of the plant, as shown in the figure above.

System Identification

- The associated normal equations are,

$$\Sigma_u \boldsymbol{w}_* = \mathbb{E}[\mathbf{u}_n \mathrm{d}_n] = \mathbb{E}[\mathbf{u}_n \mathrm{y}_n] + 0,$$

  assuming that the noise η_n to be statistically independent of $\mathbf{u}_n$. Thus, once more, the resulting normal equations are the same as if we had provided the model with a desired response equal to the **noiseless output** of the unknown plant, i.e., $\mathrm{d}_n = \mathrm{y}_n$. Hence, the impulse response of the model is estimated so that its output to be close, in the MSE, to the true (noiseless) output of the unknown plant.

- System identification is of major importance in a number of applications. In control, it is used for driving the associated controllers. In data communications, for estimating the transmission channel in order to build up maximum likelihood estimators of the transmitted data. In many practical systems, adaptive/online versions of the System Identification scheme are implemented, as it is discussed in Chapter 5.

System Identification

- The associated normal equations are,

$$\Sigma_u \boldsymbol{w}_* = \mathbb{E}[\mathbf{u}_n \mathrm{d}_n] = \mathbb{E}[\mathbf{u}_n \mathrm{y}_n] + 0,$$

  assuming that the noise η_n to be statistically independent of $\mathbf{u}_n$. Thus, once more, the resulting normal equations are the same as if we had provided the model with a desired response equal to the **noiseless output** of the unknown plant, i.e., $\mathrm{d}_n = \mathrm{y}_n$. Hence, the impulse response of the model is estimated so that its output to be close, in the MSE, to the true (noiseless) output of the unknown plant.
- System identification is of major importance in a number of applications. In control, it is used for driving the associated controllers. In data communications, for estimating the transmission channel in order to build up maximum likelihood estimators of the transmitted data. In many practical systems, adaptive/online versions of the System Identification scheme are implemented, as it is discussed in Chapter 5.

Deconvolution: Channel Equalization

- Note that in the cancellation task the goal was to "remove" the output (filtered version of the input signal, u_n,) of the unknown system H. In system identification, the focus was on the unknown system itself. In deconvolution, the emphasis is on the input of the unknown system. That is, our goal now is to recover, in the MSE optimal sense, a (delayed) input signal, $\mathrm{d}_n = \mathrm{s}_{n-L+1}$, where L is the delay in units of the sampling period, T. The task is also called inverse system identification.
- The term equalization or channel equalization is used in communications. The deconvolution task was introduced in the context of image deblurring. There, the required information about the unknown input process was obtained via an approximation. In the current framework, this can be approached via the transmission of a **training** sequence.
- The goal of an equalizer is to recover the **transmitted** information symbols, by mitigating the so-called inter-symbol interference (ISI), that any (imperfect) dispersive communication channel imposes on the transmitted signal; besides ISI, additive noise is also present in the transmission information bits.

Deconvolution: Channel Equalization

- Note that in the cancellation task the goal was to "remove" the output (filtered version of the input signal, u_n,) of the unknown system H. In system identification, the focus was on the unknown system itself. In deconvolution, the emphasis is on the input of the unknown system. That is, our goal now is to recover, in the MSE optimal sense, a (delayed) input signal, $\mathrm{d}_n = \mathrm{s}_{n-L+1}$, where L is the delay in units of the sampling period, T. The task is also called inverse system identification.
- The term equalization or channel equalization is used in communications. The deconvolution task was introduced in the context of image deblurring. There, the required information about the unknown input process was obtained via an approximation. In the current framework, this can be approached via the transmission of a **training** sequence.
- The goal of an equalizer is to recover the **transmitted** information symbols, by mitigating the so-called inter-symbol interference (ISI), that any (imperfect) dispersive communication channel imposes on the transmitted signal; besides ISI, additive noise is also present in the transmission information bits.

Deconvolution: Channel Equalization

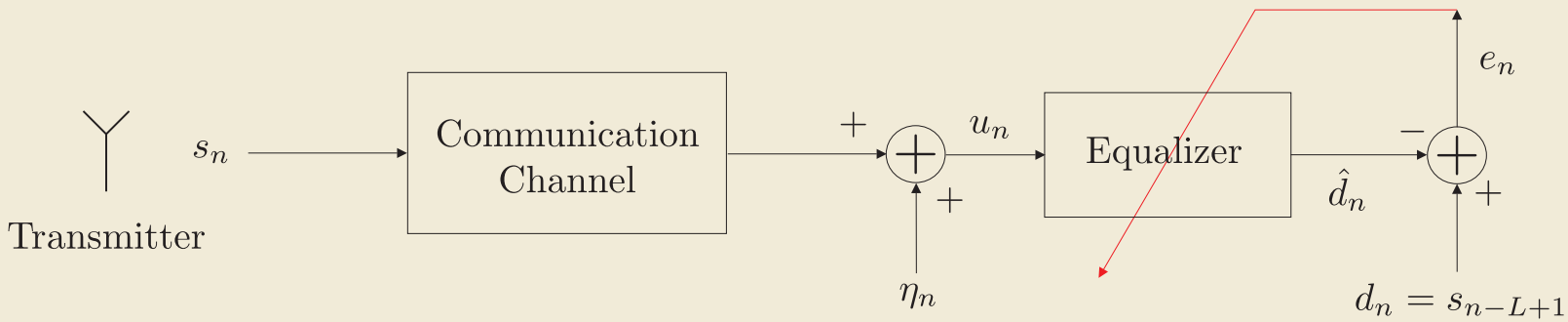
- Note that in the cancellation task the goal was to "remove" the output (filtered version of the input signal, u_n,) of the unknown system H. In system identification, the focus was on the unknown system itself. In deconvolution, the emphasis is on the input of the unknown system. That is, our goal now is to recover, in the MSE optimal sense, a (delayed) input signal, $\mathrm{d}_n = \mathrm{s}_{n-L+1}$, where L is the delay in units of the sampling period, T. The task is also called inverse system identification.
- The term equalization or channel equalization is used in communications. The deconvolution task was introduced in the context of image deblurring. There, the required information about the unknown input process was obtained via an approximation. In the current framework, this can be approached via the transmission of a **training** sequence.
- The goal of an equalizer is to recover the **transmitted** information symbols, by mitigating the so-called inter-symbol interference (ISI), that any (imperfect) dispersive communication channel imposes on the transmitted signal; besides ISI, additive noise is also present in the transmission information bits.

Deconvolution: Channel Equalization

- Equalizers are "omnipresent" in these days; in our mobile phones, in our modems, e.t.c. Deconvolution/channel equalization is at the heart of a number of applications besides communications, such as acoustics, optics, seismic signal processing, control.
- The figure below presents the basic scheme for an equalizer. The equalizer is trained so that its **output** to be as close as possible to the **transmitted data bits** delayed by some time lag L; the delay is used in order to account for the overall delayed imposed by the channel-equalizer system.

Deconvolution: Channel Equalization

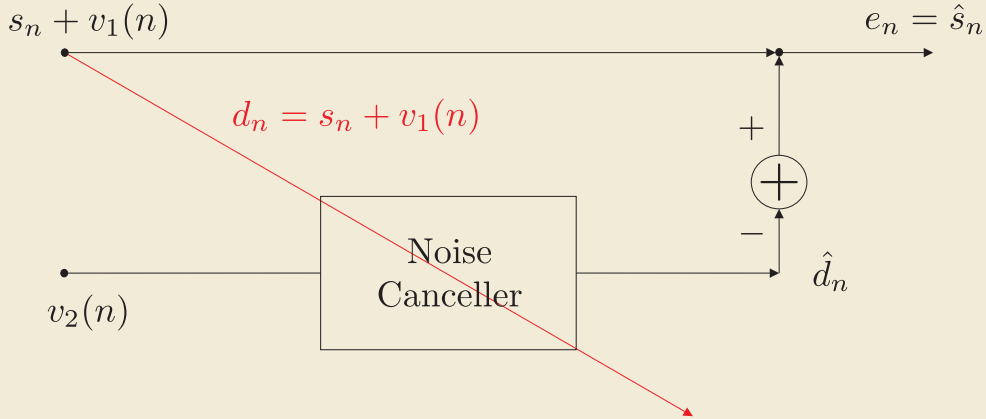
- Equalizers are "omnipresent" in these days; in our mobile phones, in our modems, e.t.c. Deconvolution/channel equalization is at the heart of a number of applications besides communications, such as acoustics, optics, seismic signal processing, control.
- The figure below presents the basic scheme for an equalizer. The equalizer is trained so that its **output** to be as close as possible to the **transmitted data bits** delayed by some time lag L; the delay is used in order to account for the overall delayed imposed by the channel-equalizer system.



The task of an equalizer is to optimally recover the originally transmitted information sequence, s_n, delayed by L time lags.

Example: Noise Cancellation

- The noise cancellation application is illustrated in the figure below. The signal of interest is a realization of a process, s_n, which is contaminated by the noise process $\mathrm{v}_1(n)$.



- For example, s_n may be the speech signal of the pilot in the cockpit and $\mathrm{v}_1(n)$ the aircraft noise at the location of the microphone. We assume that $\mathrm{v}_1(n)$ is an AR process of order one, i.e.,

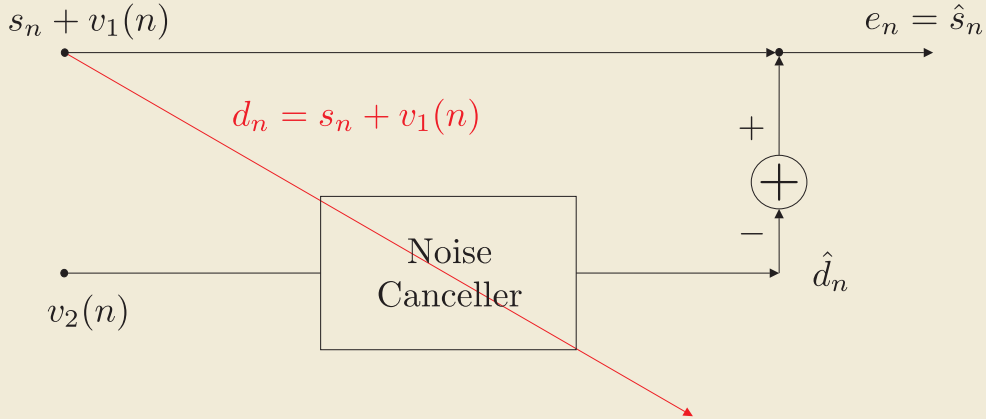
$$\mathrm{v}_1(n) = a_1 \mathrm{v}_1(n-1) + \eta_n.$$

- The random signal $\mathrm{v}_2(n)$ is a noise sequence, which is **related** to $\mathrm{v}_1(n)$, but it is statistically independent of s_n. For example, it may be the noise picked from another microphone positioned at a nearby location. This is also assumed to be an AR process of the first order,

$$\mathrm{v}_2(n) = a_2 \mathrm{v}_2(n-1) + \eta_n.$$

Example: Noise Cancellation

- The noise cancellation application is illustrated in the figure below. The signal of interest is a realization of a process, s_n, which is contaminated by the noise process $\mathrm{v}_1(n)$.



- For example, s_n may be the speech signal of the pilot in the cockpit and $\mathrm{v}_1(n)$ the aircraft noise at the location of the microphone. We assume that $\mathrm{v}_1(n)$ is an AR process of order one, i.e.,

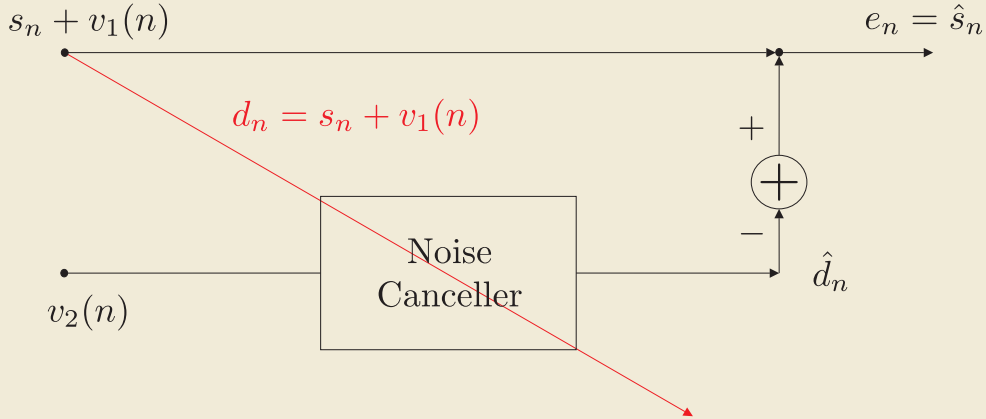
$$\mathrm{v}_1(n) = a_1 \mathrm{v}_1(n-1) + \eta_n.$$

- The random signal $\mathrm{v}_2(n)$ is a noise sequence, which is **related** to $\mathrm{v}_1(n)$, but it is statistically independent of s_n. For example, it may be the noise picked from another microphone positioned at a nearby location. This is also assumed to be an AR process of the first order,

$$\mathrm{v}_2(n) = a_2 \mathrm{v}_2(n-1) + \eta_n.$$

Example: Noise Cancellation

- The noise cancellation application is illustrated in the figure below. The signal of interest is a realization of a process, s_n, which is contaminated by the noise process $\mathrm{v}_1(n)$.



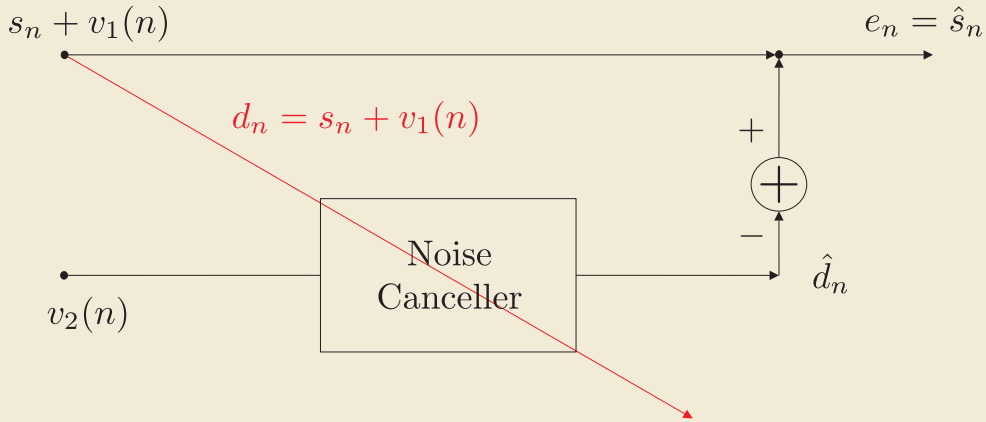
- For example, s_n may be the speech signal of the pilot in the cockpit and $\mathrm{v}_1(n)$ the aircraft noise at the location of the microphone. We assume that $\mathrm{v}_1(n)$ is an AR process of order one, i.e.,

$$\mathrm{v}_1(n) = a_1 \mathrm{v}_1(n-1) + \eta_n.$$

- The random signal $\mathrm{v}_2(n)$ is a noise sequence, which is **related** to $\mathrm{v}_1(n)$, but it is statistically independent of s_n. For example, it may be the noise picked from another microphone positioned at a nearby location. This is also assumed to be an AR process of the first order,

$$\mathrm{v}_2(n) = a_2 \mathrm{v}_2(n-1) + \eta_n.$$

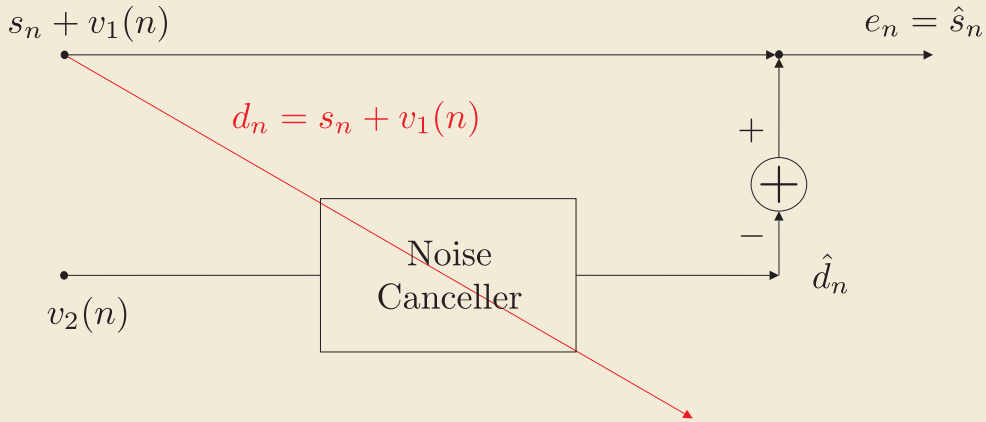
Example: Noise Cancellation



- The goal of the example is to compute estimates of the weights of the noise canceller, in order to optimally remove (in the MSE sense) the noise $\mathrm{v}_1(n)$ from the mixture $\mathrm{s}_n + \mathrm{v}_1(n)$. Assume the canceller to be of order two.
- The input to the canceller is $\mathrm{v}_2(n)$ and as desired response the mixture signal, $\mathrm{d}_n = \mathrm{s}_n + \mathrm{v}_1(n)$, will be used. To establish the normal equations, we need to compute the covariance matrix, Σ_2, of $\mathrm{v}_2(n)$ and the cross-correlation vector, $\boldsymbol{p}_2$, between the input random vector, $\mathbf{v}_2(n)$, and d_n.
- Since $\mathrm{v}_2(n)$ is an AR process of the first order, we know form Chapter 2 that, the autocorrelation coefficients are given by

$$r_2(k) = \frac{a_2^k \sigma_\eta^2}{1 - a_2^2}, \; k = 0, 1, \ldots$$

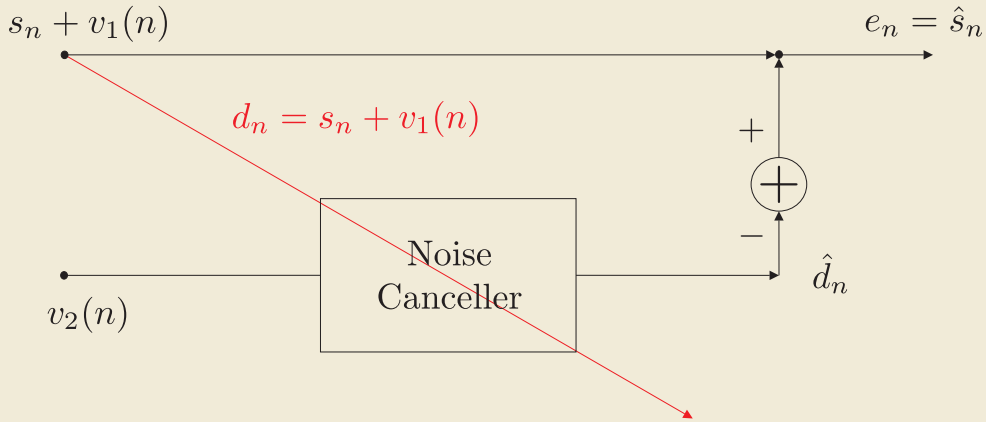
Example: Noise Cancellation



- The goal of the example is to compute estimates of the weights of the noise canceller, in order to optimally remove (in the MSE sense) the noise $\mathrm{v}_1(n)$ from the mixture $\mathrm{s}_n + \mathrm{v}_1(n)$. Assume the canceller to be of order two.
- The input to the canceller is $\mathrm{v}_2(n)$ and as desired response the mixture signal, $\mathrm{d}_n = \mathrm{s}_n + \mathrm{v}_1(n)$, will be used. To establish the normal equations, we need to compute the covariance matrix, Σ_2, of $\mathrm{v}_2(n)$ and the cross-correlation vector, $\boldsymbol{p}_2$, between the input random vector, $\mathbf{v}_2(n)$, and d_n.
- Since $\mathrm{v}_2(n)$ is an AR process of the first order, we know form Chapter 2 that, the autocorrelation coefficients are given by

$$r_2(k) = \frac{a_2^k \sigma_\eta^2}{1 - a_2^2}, \; k = 0, 1, \ldots$$

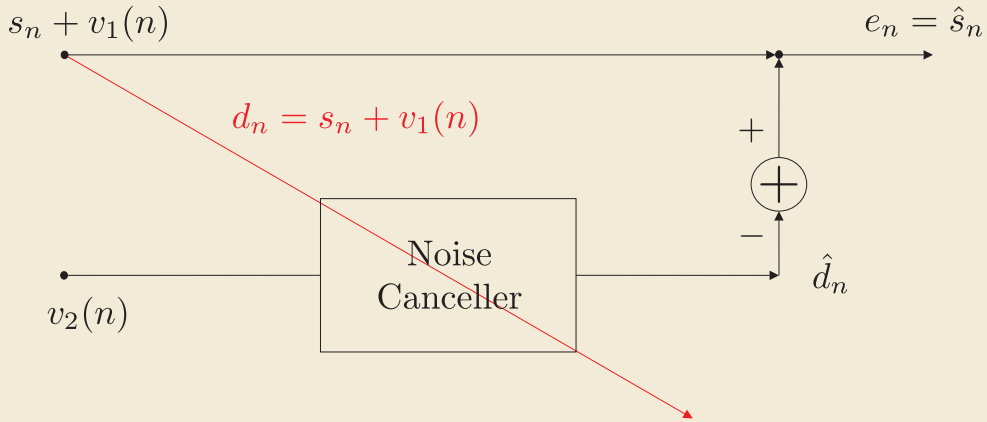
Example: Noise Cancellation



- The goal of the example is to compute estimates of the weights of the noise canceller, in order to optimally remove (in the MSE sense) the noise $\mathrm{v}_1(n)$ from the mixture $\mathrm{s}_n + \mathrm{v}_1(n)$. Assume the canceller to be of order two.
- The input to the canceller is $\mathrm{v}_2(n)$ and as desired response the mixture signal, $\mathrm{d}_n = \mathrm{s}_n + \mathrm{v}_1(n)$, will be used. To establish the normal equations, we need to compute the covariance matrix, Σ_2, of $\mathrm{v}_2(n)$ and the cross-correlation vector, $\boldsymbol{p}_2$, between the input random vector, $\mathbf{v}_2(n)$, and d_n.
- Since $\mathrm{v}_2(n)$ is an AR process of the first order, we know form Chapter 2 that, the autocorrelation coefficients are given by

$$r_2(k) = \frac{a_2^k \sigma_\eta^2}{1 - a_2^2}, \; k = 0, 1, ...$$

Example: Noise Cancellation



- Hence,

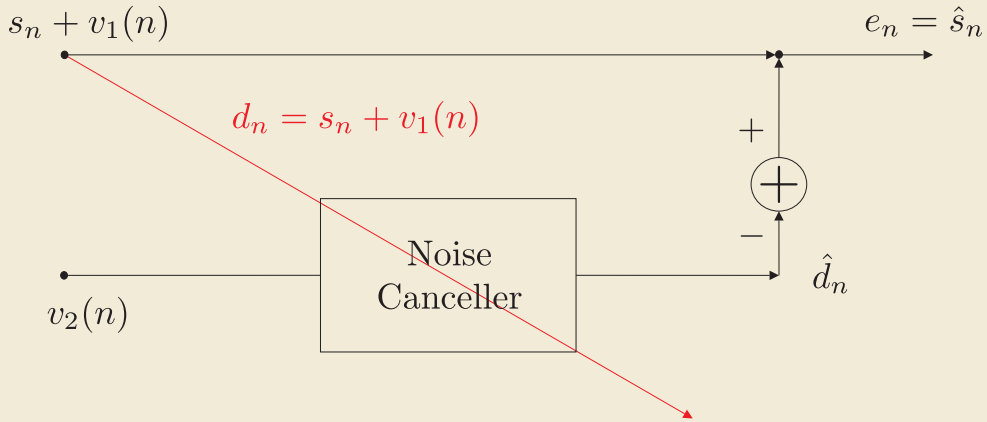
$$\Sigma_2 = \begin{bmatrix} r_2(0) & r_2(1) \\ r_2(1) & r_2(0) \end{bmatrix} = \begin{bmatrix} \frac{\sigma_\eta^2}{1-a_2^2} & \frac{a_2\sigma_\eta^2}{1-a_2^2} \\ \frac{a_2\sigma_\eta^2}{1-a_2^2} & \frac{\sigma_\eta^2}{1-a_2^2} \end{bmatrix}.$$

- Next, we are going to compute the cross-correlation vector,

$$\begin{aligned} p_2(0) &:= \mathbb{E}[\mathrm{v}_2(n)\mathrm{d}_n] = \mathbb{E}\big[\mathrm{v}_2(n)\big(\mathrm{s}_n + \mathrm{v}_1(n)\big)\big] = \mathbb{E}[\mathrm{v}_2(n)\mathrm{v}_1(n)] + 0 \\ &= \mathbb{E}\big[\big(a_2\mathrm{v}_2(n-1) + \eta_n\big)\big(a_1\mathrm{v}_1(n-1) + \eta_n\big)\big] \\ &= a_2a_1p_2(0) + \sigma_\eta^2 \Longrightarrow p_2(0) = \frac{\sigma_\eta^2}{1-a_2a_1}. \end{aligned}$$

- We used the fact that $\mathbb{E}[\mathrm{v}_2(n-1)\eta_n] = \mathbb{E}[\mathrm{v}_1(n-1)\eta_n] = 0$, since $\mathrm{v}_2(n-1)$ and $\mathrm{v}_1(n-1)$ depend recursively on previous values $\eta(n-1), \eta(n-2), \ldots$ and also η_n is a white noise sequence, hence the respective correlation values are zero.

Example: Noise Cancellation



- Hence,

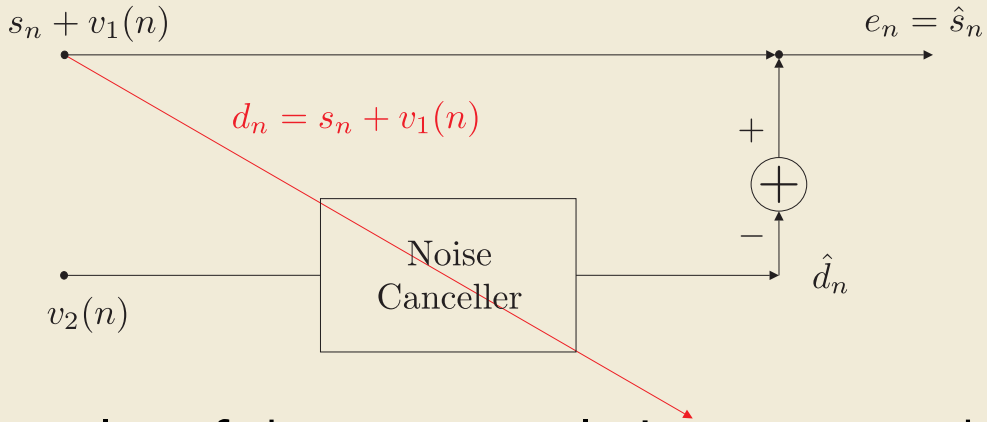
$$\Sigma_2 = \begin{bmatrix} r_2(0) & r_2(1) \\ r_2(1) & r_2(0) \end{bmatrix} = \begin{bmatrix} \frac{\sigma_\eta^2}{1-a_2^2} & \frac{a_2\sigma_\eta^2}{1-a_2^2} \\ \frac{a_2\sigma_\eta^2}{1-a_2^2} & \frac{\sigma_\eta^2}{1-a_2^2} \end{bmatrix}.$$

- Next, we are going to compute the cross-correlation vector,

$$\begin{aligned} p_2(0) &:= \mathbb{E}[\mathrm{v}_2(n)\mathrm{d}_n] = \mathbb{E}\big[\mathrm{v}_2(n)\big(\mathrm{s}_n + \mathrm{v}_1(n)\big)\big] = \mathbb{E}[\mathrm{v}_2(n)\mathrm{v}_1(n)] + 0 \\ &= \mathbb{E}\big[\big(a_2\mathrm{v}_2(n-1) + \eta_n\big)\big(a_1\mathrm{v}_1(n-1) + \eta_n\big)\big] \\ &= a_2a_1p_2(0) + \sigma_\eta^2 \Longrightarrow \; p_2(0) = \frac{\sigma_\eta^2}{1 - a_2a_1}. \end{aligned}$$

- We used the fact that $\mathbb{E}[\mathrm{v}_2(n-1)\eta_n] = \mathbb{E}[\mathrm{v}_1(n-1)\eta_n] = 0$, since $\mathrm{v}_2(n-1)$ and $\mathrm{v}_1(n-1)$ depend recursively on previous values $\eta(n-1), \eta(n-2), \ldots$ and also η_n is a white noise sequence, hence the respective correlation values are zero.

Example: Noise Cancellation



- For the other value of the cross-correlation vector we have,

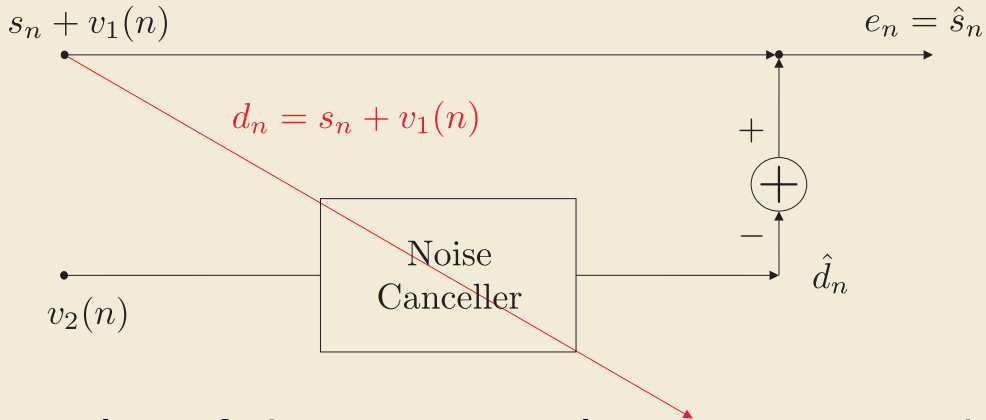
$$\begin{aligned} p_2(1) &= \mathbb{E}[\mathrm{v}_2(n-1)\mathrm{d}_n] = \mathbb{E}\big[\mathrm{v}_2(n-1)\big(\mathrm{s}_n + \mathrm{v}_1(n)\big)\big] \\ &= \mathbb{E}[\mathrm{v}_2(n-1)\mathrm{v}_1(n)] + 0 = \mathbb{E}\big[\mathrm{v}_2(n-1)\big(a_1\mathrm{v}_1(n-1) + \eta_n\big)\big] \\ &= a_1 p_2(0) = \frac{a_1\sigma_\eta^2}{1-a_1a_2}. \end{aligned}$$

- In general, it is easy to show that,

$$p_2(k) = \frac{a_1^k\sigma_\eta^2}{1-a_2a_1}, \; k = 0, 1, \dots .$$

Recall that since the processes are real-valued, the covariance matrix is symmetric, i.e., $r_2(k) = r_2(-k)$. Also, in order the models to make sense, (i.e., $r_2(0) > 0$), $|a_2| < 1$. The same holds true for $|a_1|$.

Example: Noise Cancellation



- For the other value of the cross-correlation vector we have,

$$\begin{aligned} p_2(1) &= \mathbb{E}[\mathrm{v}_2(n-1)\mathrm{d}_n] = \mathbb{E}\big[\mathrm{v}_2(n-1)\big(\mathrm{s}_n + \mathrm{v}_1(n)\big)\big] \\ &= \mathbb{E}[\mathrm{v}_2(n-1)\mathrm{v}_1(n)] + 0 = \mathbb{E}\big[\mathrm{v}_2(n-1)\big(a_1\mathrm{v}_1(n-1) + \eta_n\big)\big] \\ &= a_1 p_2(0) = \frac{a_1\sigma_\eta^2}{1 - a_1 a_2}. \end{aligned}$$

- In general, it is easy to show that,

$$p_2(k) = \frac{a_1^k \sigma_\eta^2}{1 - a_2 a_1}, \; k = 0, 1, \ldots .$$

Recall that since the processes are real-valued, the covariance matrix is symmetric, i.e., $r_2(k) = r_2(-k)$. Also, in order the models to make sense, (i.e., $r_2(0) > 0$), $|a_2| < 1$. The same holds true for $|a_1|$.

Example: Noise Cancellation

- Thus, the optimal weights of the noise canceller are given by the following set of normal equations,

$$\begin{bmatrix} \frac{\sigma_\eta^2}{1-a_2^2} & \frac{a_2\sigma_\eta^2}{1-a_2^2} \\ \frac{a_2\sigma_\eta^2}{1-a_2^2} & \frac{\sigma_\eta^2}{1-a_2^2} \end{bmatrix} \boldsymbol{w} = \begin{bmatrix} \frac{\sigma_\eta^2}{1-a_1a_2} \\ \frac{a_1\sigma_\eta^2}{1-a_1a_2} \end{bmatrix}.$$

  Note that the canceller optimally removes from the mixture, $\mathrm{s}_n + \mathrm{v}_1(n)$, the component which is correlated to the input, $\mathrm{v}_2(n)$.

- To demonstrate the validity of the above, we adopted as our information signal, the sinusoid, $s_n = \cos(\omega_0 n)$ with $\omega_0 = 2 * 10^{-3} * \pi$. Also, $d_n = s_n + v_1(n)$, with $a_1 = 0.8$ and $\sigma_\eta^2 = 0.05$. To generate $v_2(n)$, we used two different values, namely $a_2 = 0.75$ and $a_2 = 0.5$. The obtained results are shown in the next figures.

Example: Noise Cancellation

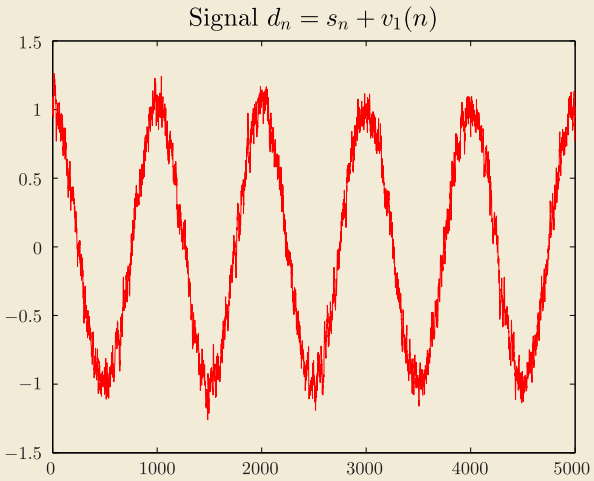
- Thus, the optimal weights of the noise canceller are given by the following set of normal equations,

$$\begin{bmatrix} \frac{\sigma_\eta^2}{1-a_2^2} & \frac{a_2\sigma_\eta^2}{1-a_2^2} \\ \frac{a_2\sigma_\eta^2}{1-a_2^2} & \frac{\sigma_\eta^2}{1-a_2^2} \end{bmatrix} \boldsymbol{w} = \begin{bmatrix} \frac{\sigma_\eta^2}{1-a_1a_2} \\ \frac{a_1\sigma_\eta^2}{1-a_1a_2} \end{bmatrix}.$$

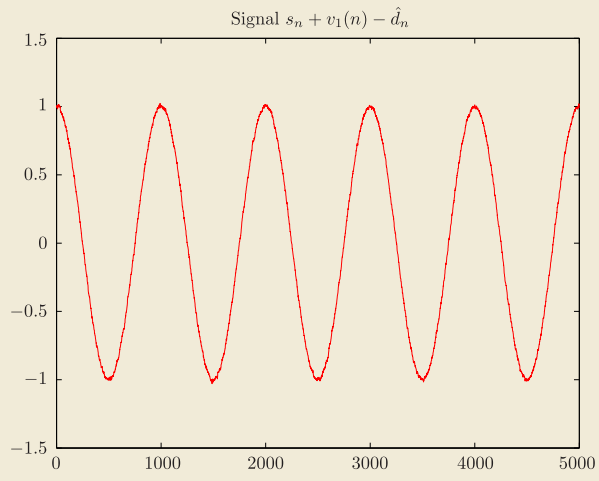
  Note that the canceller optimally removes from the mixture, $\mathrm{s}_n + \mathrm{v}_1(n)$, the component which is correlated to the input, $\mathrm{v}_2(n)$.

- To demonstrate the validity of the above, we adopted as our information signal, the sinusoid, $s_n = \cos(\omega_0 n)$ with $\omega_0 = 2 * 10^{-3} * \pi$. Also, $d_n = s_n + v_1(n)$, with $a_1 = 0.8$ and $\sigma_\eta^2 = 0.05$. To generate $v_2(n)$, we used two different values, namely $a_2 = 0.75$ and $a_2 = 0.5$. The obtained results are shown in the next figures.

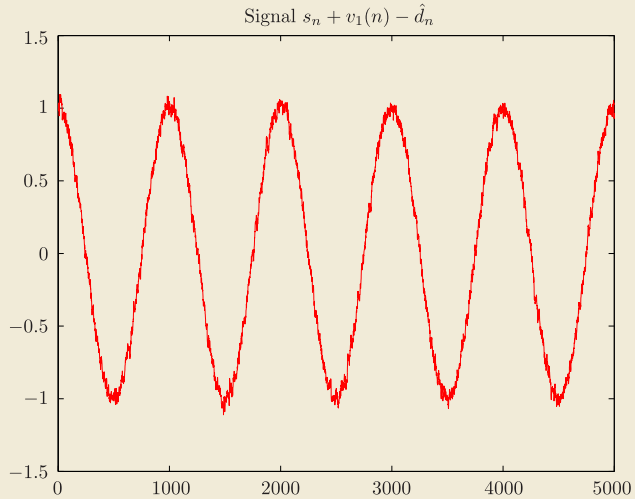
Example: Noise Cancellation



(a)



(b)

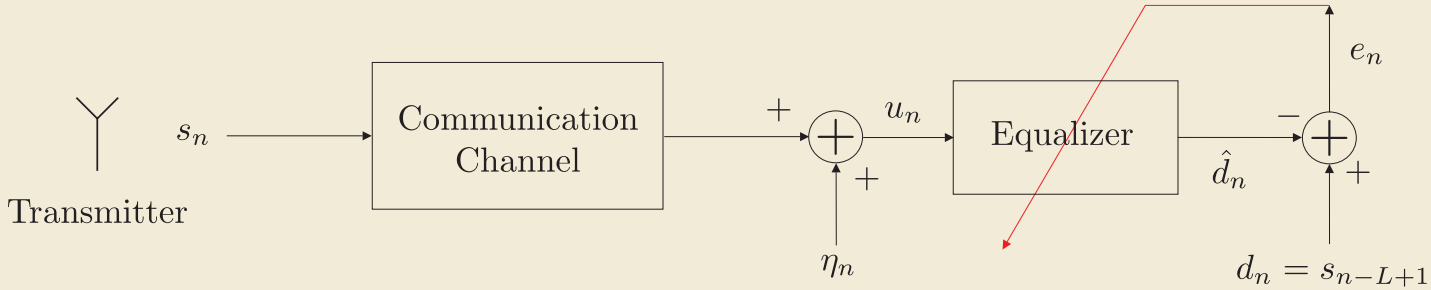


(c)

a) The noisy sinusoid signal. b) The de-noised signal for strongly correlated noise sources, v_1 and v_2, ($a_2 = 0.75$). b) The obtained de-noised signal for less correlated noise sources, ($a_2 = 0.5$).

Example: Channel Equalization

- Consider the channel equalization set up of the figure



- The output of the channel, which is sensed by the receiver, is assumed to be

$$\mathrm{u}_n = 0.5\mathrm{s}_n + \mathrm{s}_{n-1} + \eta_n.$$

- The goal is to design an equalizer comprising three taps, i.e., $\boldsymbol{w} = [w_0, w_1, w_2]^T$, so as

$$\hat{\mathrm{d}}_n = \boldsymbol{w}^T \mathbf{u}_n,$$

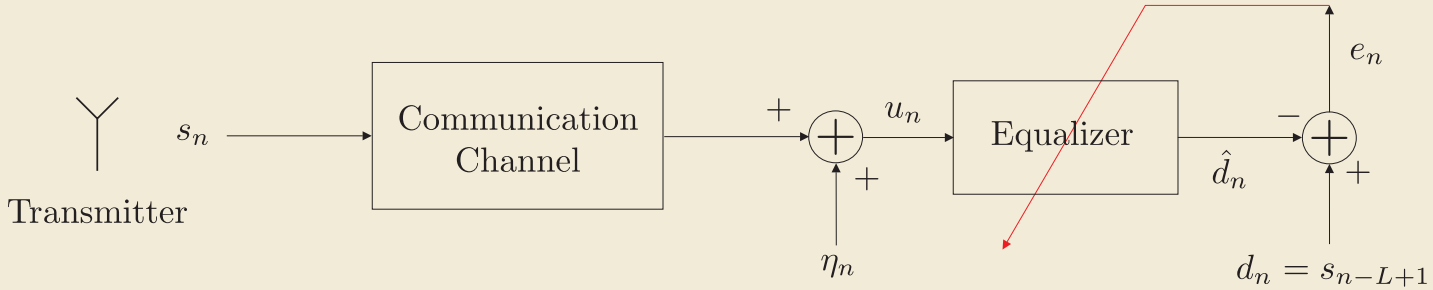
and estimate the unknown taps using as a desired response sequence $\mathrm{d}_n = \mathrm{s}_{n-1}$. We are given that $\mathbb{E}[\mathrm{s}_n] = \mathbb{E}[\eta_n] = 0$ and

$$\Sigma_s = \sigma_s^2 I, \ \Sigma_\eta = \sigma_\eta^2 I.$$

- Note that we have used a delay $L = 1$. In simple words, to explain why a delay is used, observe that at time n, most of the contribution to u_n comes from the symbol s_{n-1}; hence, it is intuitively natural, at time n, having received u_n, to try to obtain an estimate for s_{n-1}.

Example: Channel Equalization

- Consider the channel equalization set up of the figure



- The output of the channel, which is sensed by the receiver, is assumed to be

$$\mathrm{u}_n = 0.5\mathrm{s}_n + \mathrm{s}_{n-1} + \eta_n.$$

- The goal is to design an equalizer comprising three taps, i.e., $\boldsymbol{w} = [w_0, w_1, w_2]^T$, so as

$$\hat{\mathrm{d}}_n = \boldsymbol{w}^T \mathbf{u}_n,$$

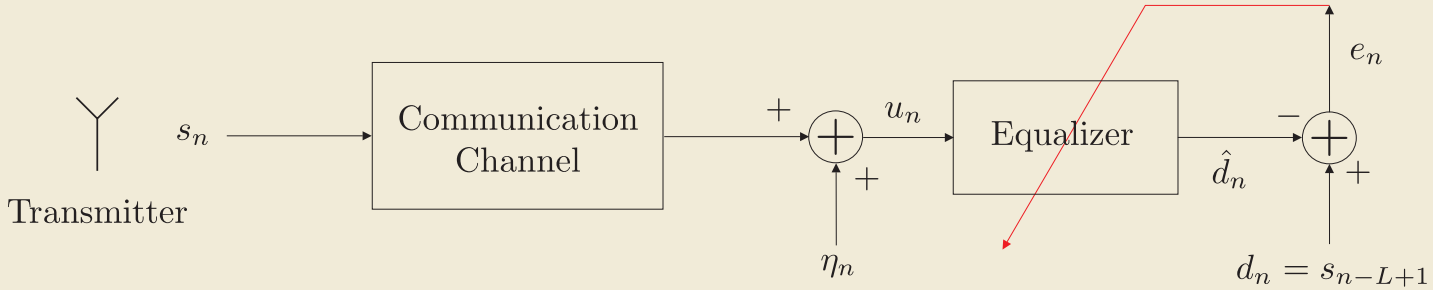
and estimate the unknown taps using as a desired response sequence $\mathrm{d}_n = \mathrm{s}_{n-1}$. We are given that $\mathbb{E}[\mathrm{s}_n] = \mathbb{E}[\eta_n] = 0$ and

$$\Sigma_s = \sigma_s^2 I, \ \Sigma_\eta = \sigma_\eta^2 I.$$

- Note that we have used a delay $L = 1$. In simple words, to explain why a delay is used, observe that at time n, most of the contribution to u_n comes from the symbol s_{n-1}; hence, it is intuitively natural, at time n, having received u_n, to try to obtain an estimate for s_{n-1}.

Example: Channel Equalization

- Consider the channel equalization set up of the figure



- The output of the channel, which is sensed by the receiver, is assumed to be

$$\mathrm{u}_n = 0.5\mathrm{s}_n + \mathrm{s}_{n-1} + \eta_n.$$

- The goal is to design an equalizer comprising three taps, i.e., $\boldsymbol{w} = [w_0, w_1, w_2]^T$, so as

$$\hat{\mathrm{d}}_n = \boldsymbol{w}^T \mathbf{u}_n,$$

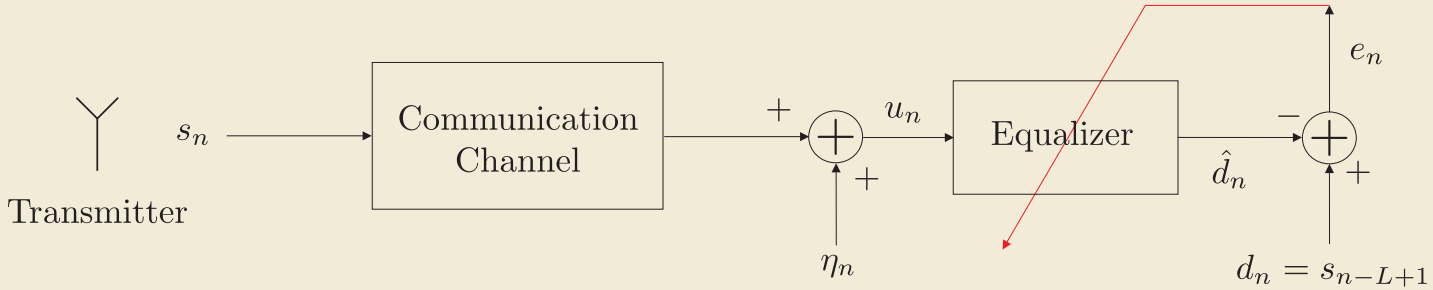
and estimate the unknown taps using as a desired response sequence $\mathrm{d}_n = \mathrm{s}_{n-1}$. We are given that $\mathbb{E}[\mathrm{s}_n] = \mathbb{E}[\eta_n] = 0$ and

$$\Sigma_s = \sigma_s^2 I, \ \Sigma_\eta = \sigma_\eta^2 I.$$

- Note that we have used a delay $L = 1$. In simple words, to explain why a delay is used, observe that at time n, most of the contribution to u_n comes from the symbol s_{n-1}; hence, it is intuitively natural, at time n, having received u_n, to try to obtain an estimate for s_{n-1}.

Example: Channel Equalization

- Consider the channel equalization set up of the figure



- The output of the channel, which is sensed by the receiver, is assumed to be

$$\mathrm{u}_n = 0.5\mathrm{s}_n + \mathrm{s}_{n-1} + \eta_n.$$

- The goal is to design an equalizer comprising three taps, i.e., $\boldsymbol{w} = [w_0, w_1, w_2]^T$, so as

$$\hat{\mathrm{d}}_n = \boldsymbol{w}^T \mathbf{u}_n,$$

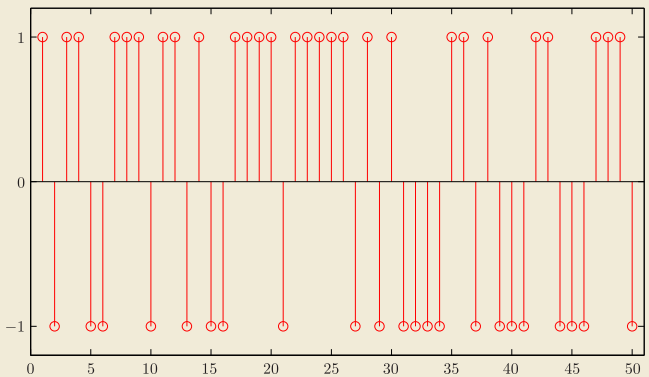
and estimate the unknown taps using as a desired response sequence $\mathrm{d}_n = \mathrm{s}_{n-1}$. We are given that $\mathbb{E}[\mathrm{s}_n] = \mathbb{E}[\eta_n] = 0$ and

$$\Sigma_s = \sigma_s^2 I, \ \Sigma_\eta = \sigma_\eta^2 I.$$

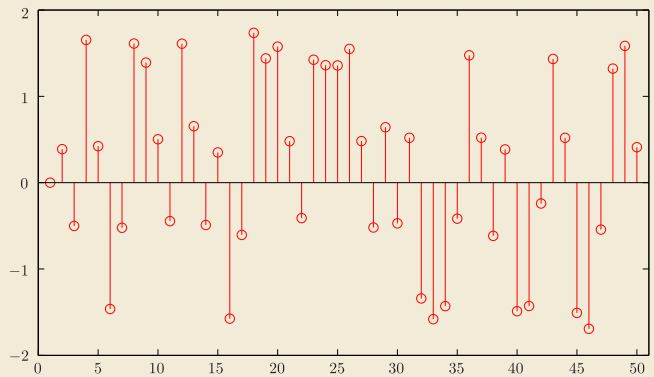
- Note that we have used a delay $L = 1$. In simple words, to explain why a delay is used, observe that at time n, most of the contribution to u_n comes from the symbol s_{n-1}; hence, it is intuitively natural, at time n, having received u_n, to try to obtain an estimate for s_{n-1}.

Example: Channel Equalization

- Figure (a) shows a realization of the input information sequence s_n. It consists of equiprobable ± 1 samples, randomly generated. The effect of the channel is a) to combine successive information samples together (ISI) and b) to add noise; the purpose of the equalizer is to optimally remove **both** of them. Figure (b) shows the respective realization sequence of u_n, which is received at the receiver's front end. Observe that, by looking at it, one cannot recognize in it the original sequence; the noise together with the ISI have really changed its "look".



(a)



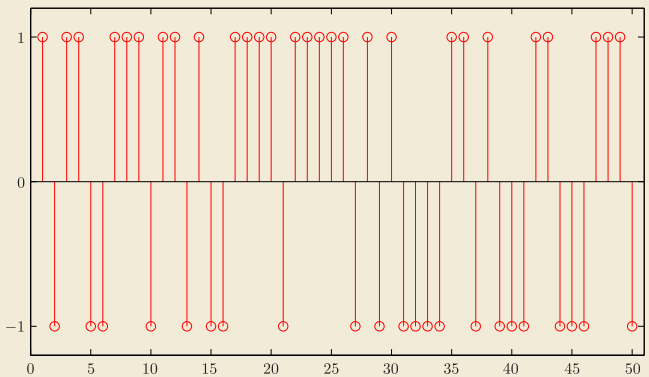
(b)

- Following a similar procedure as in the previous example, we obtain

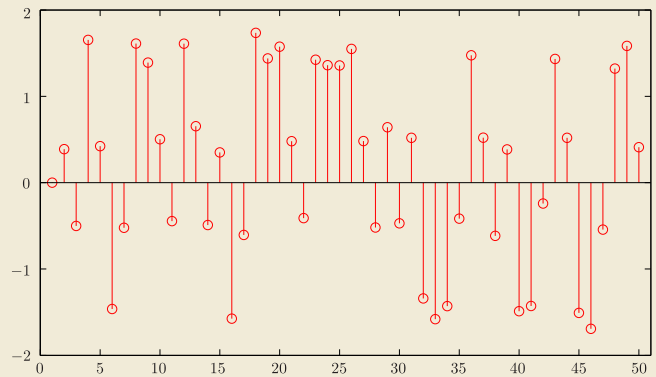
$$\Sigma_u = \begin{bmatrix} 1.25\sigma_s^2 + \sigma_\eta^2 & 0.5\sigma_s^2 & 0 \\ 0.5\sigma_s^2 & 1.25\sigma_s^2 + \sigma_\eta^2 & 0.5\sigma_s^2 \\ 0 & 0.5\sigma_s^2 & 1.25\sigma_s^2 + \sigma_\eta^2 \end{bmatrix}, \; \boldsymbol{p} = \begin{bmatrix} \sigma_s^2 \\ 0.5\sigma_s^2 \\ 0 \end{bmatrix}.$$

Example: Channel Equalization

- Figure (a) shows a realization of the input information sequence s_n. It consists of equiprobable ± 1 samples, randomly generated. The effect of the channel is a) to combine successive information samples together (ISI) and b) to add noise; the purpose of the equalizer is to optimally remove **both** of them. Figure (b) shows the respective realization sequence of u_n, which is received at the receiver's front end. Observe that, by looking at it, one cannot recognize in it the original sequence; the noise together with the ISI have really changed its "look".



(a)



(b)

- Following a similar procedure as in the previous example, we obtain

$$\Sigma_u = \begin{bmatrix} 1.25\sigma_s^2 + \sigma_\eta^2 & 0.5\sigma_s^2 & 0 \\ 0.5\sigma_s^2 & 1.25\sigma_s^2 + \sigma_\eta^2 & 0.5\sigma_s^2 \\ 0 & 0.5\sigma_s^2 & 1.25\sigma_s^2 + \sigma_\eta^2 \end{bmatrix}, \; \boldsymbol{p} = \begin{bmatrix} \sigma_s^2 \\ 0.5\sigma_s^2 \\ 0 \end{bmatrix}.$$

Example: Channel Equalization

- Solving the normal equations,

$$\Sigma_u \boldsymbol{w}_* = \boldsymbol{p},$$

for $\sigma_s^2 = 1$ and $\sigma_\eta^2 = 0.01$, results in

$$\boldsymbol{w}_* = [0.7462, 0.1195, -0.0474]^T.$$

- Figure (c) shows the recovered sequence by the equalizer ($\boldsymbol{w}_*^T \boldsymbol{u}_n$), after thresholding. It is exactly the same with the transmitted one; no errors. Figure (d) shows the recovered sequence, for increased noise variance, i.e., $\sigma_\eta^2 = 1$. The corresponding MSE optimal equalizer is equal to

$$\boldsymbol{w}_* = [0.4132, 0.1369, -0.0304]^T.$$

This time, the reconstructed by the equalizer sequence has errors, with respect to the transmitted one (gray lines).

Example: Channel Equalization

- Solving the normal equations,

$$\Sigma_u \boldsymbol{w}_* = \boldsymbol{p},$$

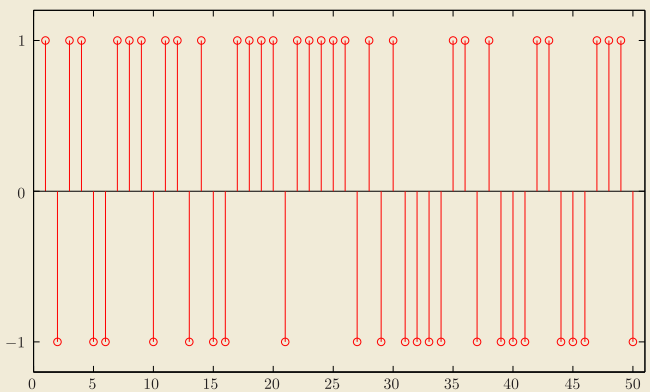
for $\sigma_s^2 = 1$ and $\sigma_\eta^2 = 0.01$, results in

$$\boldsymbol{w}_* = [0.7462, 0.1195, -0.0474]^T.$$

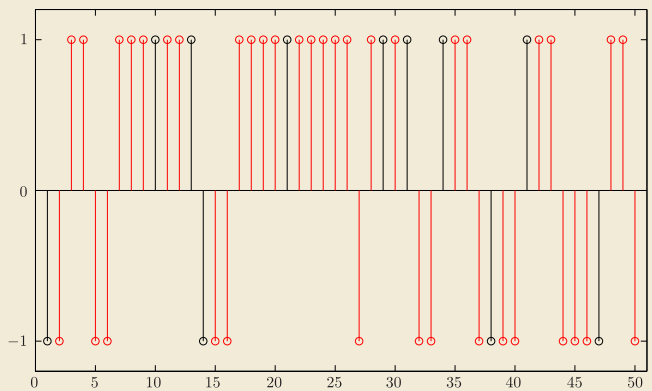
- Figure (c) shows the recovered sequence by the equalizer ($\boldsymbol{w}_*^T \boldsymbol{u}_n$), after thresholding. It is exactly the same with the transmitted one; no errors. Figure (d) shows the recovered sequence, for increased noise variance, i.e., $\sigma_\eta^2 = 1$. The corresponding MSE optimal equalizer is equal to

$$\boldsymbol{w}_* = [0.4132, 0.1369, -0.0304]^T.$$

This time, the reconstructed by the equalizer sequence has errors, with respect to the transmitted one (gray lines).



(c)



(d)

Extension to Complex-Valued Variables

- Everything that has been said so far can be extended to complex-valued signals. However, there are a few subtle points involved and this is the reason that we chose to treat this case separately. Complex-valued variables are very common in a number of applications, as for example in communications.
- Given two real-valued variables, (x, y), one can consider them either as a vector quantity in the two dimensional-space, $[x, y]^T$, or can describe them as a complex variable, $z = x + jy$, where $j^2 := -1$. Adopting the latter approach, offers the luxury of exploiting the operations available in the field $\mathbb{C}$ of complex numbers, i.e. multiplication and division. The existence of such operations greatly facilitates the algebraic manipulations. Recall that such operations are **not** defined in vector spaces.
- Let us assume that we are given a complex-valued (output) random variable,

$$\mathrm{y} := \mathrm{y}_r + j\mathrm{y}_i,$$

and a complex-valued (input) random vector

$$\mathbf{x} = \mathbf{x}_r + j\mathbf{x}_i.$$

Extension to Complex-Valued Variables

- Everything that has been said so far can be extended to complex-valued signals. However, there are a few subtle points involved and this is the reason that we chose to treat this case separately. Complex-valued variables are very common in a number of applications, as for example in communications.
- Given two real-valued variables, (x, y), one can consider them either as a vector quantity in the two dimensional-space, $[x, y]^T$, or can describe them as a complex variable, $z = x + jy$, where $j^2 := -1$. Adopting the latter approach, offers the luxury of exploiting the operations available in the field $\mathbb{C}$ of complex numbers, i.e. multiplication and division. The existence of such operations greatly facilitates the algebraic manipulations. Recall that such operations are **not** defined in vector spaces.
- Let us assume that we are given a complex-valued (output) random variable,

$$\mathrm{y} := \mathrm{y}_r + j\mathrm{y}_i,$$

and a complex-valued (input) random vector

$$\mathbf{x} = \mathbf{x}_r + j\mathbf{x}_i.$$

Extension to Complex-Valued Variables

- Everything that has been said so far can be extended to complex-valued signals. However, there are a few subtle points involved and this is the reason that we chose to treat this case separately. Complex-valued variables are very common in a number of applications, as for example in communications.
- Given two real-valued variables, (x, y), one can consider them either as a vector quantity in the two dimensional-space, $[x, y]^T$, or can describe them as a complex variable, $z = x + jy$, where $j^2 := -1$. Adopting the latter approach, offers the luxury of exploiting the operations available in the field $\mathbb{C}$ of complex numbers, i.e. multiplication and division. The existence of such operations greatly facilitates the algebraic manipulations. Recall that such operations are **not** defined in vector spaces.
- Let us assume that we are given a complex-valued (output) random variable,

$$\mathrm{y} := \mathrm{y}_r + j\mathrm{y}_i,$$

and a complex-valued (input) random vector

$$\mathbf{x} = \mathbf{x}_r + j\mathbf{x}_i.$$

Extension to Complex-Valued Variables

- The quantities $\mathrm{y}_r, \mathrm{y}_i, \mathbf{x}_r$ and $\mathbf{x}_i$ are real-valued random variables/vectors. The goal is to compute a linear estimator defined by a complex-valued parameter vector $\boldsymbol{\theta} = \boldsymbol{\theta}_r + j\boldsymbol{\theta}_i \in \mathbb{C}^l$, so as to minimize the respective mean-square error,

$$\mathbb{E}\left[|\mathrm{e}|^2\right] := \mathbb{E}\left[\mathrm{e}\mathrm{e}^*\right] = \mathbb{E}[|\mathrm{y} - \boldsymbol{\theta}^H \mathbf{x}|^2]\,.$$

- Looking at the above, it is readily observed that in the case of complex variables the inner product operation between two complex-valued random variables should be defined as $\mathbb{E}[\mathrm{x}\mathrm{y}^*]$, so as to guarantee that the implied norm by the inner product, i.e., $\|\mathrm{x}\| = \sqrt{\mathbb{E}[\mathrm{x}\mathrm{x}^*]}$, is a valid quantity. Applying the orthogonality condition as before, we rederive the normal equations, i.e.,

$$\Sigma_x \boldsymbol{\theta}_* = \boldsymbol{p},$$

where now the covariance matrix and cross-correlation vector are given by

$$\Sigma_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right], \quad \boldsymbol{p} = \mathbb{E}\left[\mathbf{x}\mathrm{y}^*\right].$$

- The equivalent cost function to be minimized is given by

$$J(\boldsymbol{\theta}) = \mathbb{E}[|\mathrm{e}|^2] = \mathbb{E}[|\mathrm{y} - \hat{\mathrm{y}}|^2] = \mathbb{E}[|\mathrm{y}_r - \hat{\mathrm{y}}_r|^2] + \mathbb{E}[|\mathrm{y}_i - \hat{\mathrm{y}}_i|^2]$$

where,

$$\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^H \mathbf{x}. \tag{5}$$

Extension to Complex-Valued Variables

- The quantities $\mathrm{y}_r, \mathrm{y}_i, \mathbf{x}_r$ and $\mathbf{x}_i$ are real-valued random variables/vectors. The goal is to compute a linear estimator defined by a complex-valued parameter vector $\boldsymbol{\theta} = \boldsymbol{\theta}_r + j\boldsymbol{\theta}_i \in \mathbb{C}^l$, so as to minimize the respective mean-square error,

$$\mathbb{E}\left[|\mathrm{e}|^2\right] := \mathbb{E}\left[\mathrm{e}\mathrm{e}^*\right] = \mathbb{E}[|\mathrm{y} - \boldsymbol{\theta}^H \mathbf{x}|^2] \,.$$

- Looking at the above, it is readily observed that in the case of complex variables the inner product operation between two complex-valued random variables should be defined as $\mathbb{E}[\mathrm{x}\mathrm{y}^*]$, so as to guarantee that the implied norm by the inner product, i.e., $\|\mathrm{x}\| = \sqrt{\mathbb{E}[\mathrm{x}\mathrm{x}^*]}$, is a valid quantity. Applying the orthogonality condition as before, we rederive the normal equations, i.e.,

$$\Sigma_x \boldsymbol{\theta}_* = \boldsymbol{p},$$

where now the covariance matrix and cross-correlation vector are given by

$$\Sigma_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right], \quad \boldsymbol{p} = \mathbb{E}\left[\mathbf{x}\mathrm{y}^*\right].$$

- The equivalent cost function to be minimized is given by

$$J(\boldsymbol{\theta}) = \mathbb{E}[|\mathrm{e}|^2] = \mathbb{E}[|\mathrm{y} - \hat{\mathrm{y}}|^2] = \mathbb{E}[|\mathrm{y}_r - \hat{\mathrm{y}}_r|^2] + \mathbb{E}[|\mathrm{y}_i - \hat{\mathrm{y}}_i|^2]$$

where,

$$\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^H \mathbf{x}. \tag{5}$$

Extension to Complex-Valued Variables

- The quantities $\mathrm{y}_r, \mathrm{y}_i, \mathbf{x}_r$ and $\mathbf{x}_i$ are real-valued random variables/vectors. The goal is to compute a linear estimator defined by a complex-valued parameter vector $\boldsymbol{\theta} = \boldsymbol{\theta}_r + j\boldsymbol{\theta}_i \in \mathbb{C}^l$, so as to minimize the respective mean-square error,

$$\mathbb{E}\left[|\mathrm{e}|^2\right] := \mathbb{E}\left[\mathrm{e}\mathrm{e}^*\right] = \mathbb{E}[|\mathrm{y} - \boldsymbol{\theta}^H \mathbf{x}|^2] \,.$$

- Looking at the above, it is readily observed that in the case of complex variables the inner product operation between two complex-valued random variables should be defined as $\mathbb{E}[\mathrm{x}\mathrm{y}^*]$, so as to guarantee that the implied norm by the inner product, i.e., $\|\mathrm{x}\| = \sqrt{\mathbb{E}[\mathrm{x}\mathrm{x}^*]}$, is a valid quantity. Applying the orthogonality condition as before, we rederive the normal equations, i.e.,

$$\Sigma_x \boldsymbol{\theta}_* = \boldsymbol{p},$$

where now the covariance matrix and cross-correlation vector are given by

$$\Sigma_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right], \quad \boldsymbol{p} = \mathbb{E}\left[\mathbf{x}\mathrm{y}^*\right].$$

- The equivalent cost function to be minimized is given by

$$J(\boldsymbol{\theta}) = \mathbb{E}[|\mathrm{e}|^2] = \mathbb{E}[|\mathrm{y} - \hat{\mathrm{y}}|^2] = \mathbb{E}[|\mathrm{y}_r - \hat{\mathrm{y}}_r|^2] + \mathbb{E}[|\mathrm{y}_i - \hat{\mathrm{y}}_i|^2]$$

where,

$$\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^H \mathbf{x}. \tag{5}$$

Extension to Complex-Valued Variables

- Complex linear estimator: The previous estimator, i.e.,

$$\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^H \mathbf{x}$$

  can also be written as,

$$\hat{\mathrm{y}} = (\boldsymbol{\theta}_r^T - j\boldsymbol{\theta}_i^T)(\mathbf{x}_r + j\mathbf{x}_i) = (\boldsymbol{\theta}_r^T \mathbf{x}_r + \boldsymbol{\theta}_i^T \mathbf{x}_i) + j(\boldsymbol{\theta}_r^T \mathbf{x}_i - \boldsymbol{\theta}_i^T \mathbf{x}_r).$$

- The above equation reveals the true flavor behind the complex notation; that is, its multichannel nature. In multichannel estimation, we are given more than one sets of input variables, e.g., $\mathbf{x}_r$ and $\mathbf{x}_i$, and we want to generate, **jointly**, more than one output variables, e.g., $\hat{\mathrm{y}}_r$ and $\hat{\mathrm{y}}_i$.
- The last equation can equivalently be written as,

$$\begin{bmatrix} \hat{\mathrm{y}}_r \\ \hat{\mathrm{y}}_i \end{bmatrix} = \Theta \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \end{bmatrix}, \tag{6}$$

  where

$$\Theta := \begin{bmatrix} \boldsymbol{\theta}_r^T & \boldsymbol{\theta}_i^T \\ -\boldsymbol{\theta}_i^T & \boldsymbol{\theta}_r^T \end{bmatrix}.$$

- Looking at (6), we observe that the complex linear estimation task, resulted in a matrix, Θ, of a very special structure.

Extension to Complex-Valued Variables

- Complex linear estimator: The previous estimator, i.e.,

$$\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^H \mathbf{x}$$

  can also be written as,

$$\hat{\mathrm{y}} = (\boldsymbol{\theta}_r^T - j\boldsymbol{\theta}_i^T)(\mathbf{x}_r + j\mathbf{x}_i) = (\boldsymbol{\theta}_r^T \mathbf{x}_r + \boldsymbol{\theta}_i^T \mathbf{x}_i) + j(\boldsymbol{\theta}_r^T \mathbf{x}_i - \boldsymbol{\theta}_i^T \mathbf{x}_r).$$

- The above equation reveals the true flavor behind the complex notation; that is, its multichannel nature. In multichannel estimation, we are given more than one sets of input variables, e.g., $\mathbf{x}_r$ and $\mathbf{x}_i$, and we want to generate, **jointly**, more than one output variables, e.g., $\hat{\mathrm{y}}_r$ and $\hat{\mathrm{y}}_i$.
- The last equation can equivalently be written as,

$$\begin{bmatrix} \hat{\mathrm{y}}_r \\ \hat{\mathrm{y}}_i \end{bmatrix} = \Theta \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \end{bmatrix}, \tag{6}$$

  where

$$\Theta := \begin{bmatrix} \boldsymbol{\theta}_r^T & \boldsymbol{\theta}_i^T \\ -\boldsymbol{\theta}_i^T & \boldsymbol{\theta}_r^T \end{bmatrix}.$$

- Looking at (6), we observe that the complex linear estimation task, resulted in a matrix, Θ, of a very special structure.

Extension to Complex-Valued Variables

- Complex linear estimator: The previous estimator, i.e.,

$$\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^H \mathbf{x}$$

  can also be written as,

$$\hat{\mathrm{y}} = (\boldsymbol{\theta}_r^T - j\boldsymbol{\theta}_i^T)(\mathbf{x}_r + j\mathbf{x}_i) = (\boldsymbol{\theta}_r^T \mathbf{x}_r + \boldsymbol{\theta}_i^T \mathbf{x}_i) + j(\boldsymbol{\theta}_r^T \mathbf{x}_i - \boldsymbol{\theta}_i^T \mathbf{x}_r).$$

- The above equation reveals the true flavor behind the complex notation; that is, its multichannel nature. In multichannel estimation, we are given more than one sets of input variables, e.g., $\mathbf{x}_r$ and $\mathbf{x}_i$, and we want to generate, **jointly**, more than one output variables, e.g., $\hat{\mathrm{y}}_r$ and $\hat{\mathrm{y}}_i$.
- The last equation can equivalently be written as,

$$\begin{bmatrix} \hat{\mathrm{y}}_r \\ \hat{\mathrm{y}}_i \end{bmatrix} = \Theta \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \end{bmatrix}, \tag{6}$$

  where

$$\Theta := \begin{bmatrix} \boldsymbol{\theta}_r^T & \boldsymbol{\theta}_i^T \\ -\boldsymbol{\theta}_i^T & \boldsymbol{\theta}_r^T \end{bmatrix}.$$

- Looking at (6), we observe that the complex linear estimation task, resulted in a matrix, Θ, of a very special structure.

Extension to Complex-Valued Variables

- Complex linear estimator: The previous estimator, i.e.,

$$\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^H \mathbf{x}$$

  can also be written as,

$$\hat{\mathrm{y}} = (\boldsymbol{\theta}_r^T - j\boldsymbol{\theta}_i^T)(\mathbf{x}_r + j\mathbf{x}_i) = (\boldsymbol{\theta}_r^T \mathbf{x}_r + \boldsymbol{\theta}_i^T \mathbf{x}_i) + j(\boldsymbol{\theta}_r^T \mathbf{x}_i - \boldsymbol{\theta}_i^T \mathbf{x}_r).$$

- The above equation reveals the true flavor behind the complex notation; that is, its multichannel nature. In multichannel estimation, we are given more than one sets of input variables, e.g., $\mathbf{x}_r$ and $\mathbf{x}_i$, and we want to generate, **jointly**, more than one output variables, e.g., $\hat{\mathrm{y}}_r$ and $\hat{\mathrm{y}}_i$.
- The last equation can equivalently be written as,

$$\begin{bmatrix} \hat{\mathrm{y}}_r \\ \hat{\mathrm{y}}_i \end{bmatrix} = \Theta \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \end{bmatrix}, \tag{6}$$

  where

$$\Theta := \begin{bmatrix} \boldsymbol{\theta}_r^T & \boldsymbol{\theta}_i^T \\ -\boldsymbol{\theta}_i^T & \boldsymbol{\theta}_r^T \end{bmatrix}.$$

- Looking at (6), we observe that the complex linear estimation task, resulted in a matrix, Θ, of a very special structure.

Extension to Complex-Valued Variables

- Widely linear Complex estimator: Let us define the linear two-channel estimation task starting from the definition of a **linear operation** in vector spaces. The task is to generate a vector output, $\hat{\mathbf{y}} = [\hat{\mathrm{y}}_r, \hat{\mathrm{y}}_i]^T$, $\boldsymbol{y} \in \mathbb{R}^2$ from the input vector variables, $\mathbf{x} = [\mathbf{x}_r, \mathbf{x}_i]^T \in \mathbb{R}^{2l}$, via the linear operation,

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{\mathrm{y}}_r \\ \hat{\mathrm{y}}_i \end{bmatrix} = \Theta \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \end{bmatrix}, \tag{7}$$

where,

$$\Theta := \begin{bmatrix} \boldsymbol{\theta}_{11}^T & \boldsymbol{\theta}_{12}^T \\ \boldsymbol{\theta}_{21}^T & \boldsymbol{\theta}_{22}^T \end{bmatrix},$$

and compute the matrix Θ so as to minimize the total error variance i.e,

$$\Theta_* := \arg\min_{\Theta} \Big\{ \mathbb{E}\big[(\mathrm{y}_r - \hat{\mathrm{y}}_r)^2\big] + \mathbb{E}\big[(\mathrm{y}_i - \hat{\mathrm{y}}_i)^2\big] \Big\}.$$

- It turns out that the general formulation of the linear operation in the two-dimensional space, given in (7) can be equivalently written as

$$\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^H \mathbf{x} + \boldsymbol{v}^H \mathbf{x}^* \tag{8}$$

where

$$\boldsymbol{\theta}_r := \frac{1}{2}(\boldsymbol{\theta}_{11} + \boldsymbol{\theta}_{22}), \quad \boldsymbol{\theta}_i := \frac{1}{2}(\boldsymbol{\theta}_{12} - \boldsymbol{\theta}_{21}),$$

and

$$\boldsymbol{v}_r := \frac{1}{2}(\boldsymbol{\theta}_{11} - \boldsymbol{\theta}_{22}), \quad \boldsymbol{v}_i := -\frac{1}{2}(\boldsymbol{\theta}_{12} + \boldsymbol{\theta}_{21}).$$

Extension to Complex-Valued Variables

- Widely linear Complex estimator: Let us define the linear two-channel estimation task starting from the definition of a **linear operation** in vector spaces. The task is to generate a vector output, $\hat{\mathbf{y}} = [\hat{\mathrm{y}}_r, \hat{\mathrm{y}}_i]^T$, $\boldsymbol{y} \in \mathbb{R}^2$ from the input vector variables, $\mathbf{x} = [\mathbf{x}_r, \mathbf{x}_i]^T \in \mathbb{R}^{2l}$, via the linear operation,

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{\mathrm{y}}_r \\ \hat{\mathrm{y}}_i \end{bmatrix} = \Theta \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \end{bmatrix}, \tag{7}$$

where,

$$\Theta := \begin{bmatrix} \boldsymbol{\theta}_{11}^T & \boldsymbol{\theta}_{12}^T \\ \boldsymbol{\theta}_{21}^T & \boldsymbol{\theta}_{22}^T \end{bmatrix},$$

and compute the matrix Θ so as to minimize the total error variance i.e,

$$\Theta_* := \arg\min_{\Theta} \left\{ \mathbb{E}\big[(\mathrm{y}_r - \hat{\mathrm{y}}_r)^2\big] + \mathbb{E}\big[(\mathrm{y}_i - \hat{\mathrm{y}}_i)^2\big] \right\}.$$

- It turns out that the general formulation of the linear operation in the two-dimensional space, given in (7) can be equivalently written as

$$\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^H \mathbf{x} + \boldsymbol{v}^H \mathbf{x}^* \tag{8}$$

where

$$\boldsymbol{\theta}_r := \frac{1}{2}(\boldsymbol{\theta}_{11} + \boldsymbol{\theta}_{22}), \quad \boldsymbol{\theta}_i := \frac{1}{2}(\boldsymbol{\theta}_{12} - \boldsymbol{\theta}_{21}),$$

and

$$\boldsymbol{v}_r := \frac{1}{2}(\boldsymbol{\theta}_{11} - \boldsymbol{\theta}_{22}), \quad \boldsymbol{v}_i := -\frac{1}{2}(\boldsymbol{\theta}_{12} + \boldsymbol{\theta}_{21}).$$

Extension to Complex-Valued Variables

- To distinguish from the complex linear estimation, Eq. (8) is known as widely linear complex-valued estimator. Note that in this case, both $\mathbf{x}$ as well as its complex conjugate, $\mathbf{x}^*$, are simultaneously used in order to cover all possible solutions, as those are dictated by the general linear formulation in a vector space.
- Let us now define,

$$\boldsymbol{\varphi} := \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{v} \end{bmatrix} \text{ and } \tilde{\mathbf{x}} := \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix}.$$

Then the widely linear estimator is written as,

$$\hat{\mathrm{y}} = \boldsymbol{\varphi}^H \tilde{\mathbf{x}}.$$

- Adopting the orthogonality condition in its complex formulation, i.e.,

$$\mathbb{E}\left[\tilde{\mathbf{x}}\mathrm{e}^*\right] = \mathbb{E}\left[\tilde{\mathbf{x}}\left(\mathrm{y} - \hat{\mathrm{y}}\right)^*\right] = \mathbf{0},$$

it turns out that the normal equations are equivalently written as

$$\begin{bmatrix} \Sigma_x & P_x \\ P_x^* & \Sigma_x^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_* \\ \boldsymbol{v}_* \end{bmatrix} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q}^* \end{bmatrix}, \text{ where } P_x := \mathbb{E}[\mathbf{x}\mathbf{x}^T],\ \boldsymbol{q} := \mathbb{E}[\mathbf{x}\mathrm{y}].$$

Extension to Complex-Valued Variables

- To distinguish from the complex linear estimation, Eq. (8) is known as widely linear complex-valued estimator. Note that in this case, both $\mathbf{x}$ as well as its complex conjugate, $\mathbf{x}^*$, are simultaneously used in order to cover all possible solutions, as those are dictated by the general linear formulation in a vector space.
- Let us now define,

$$\boldsymbol{\varphi} := \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{v} \end{bmatrix} \text{ and } \tilde{\mathbf{x}} := \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix}.$$

Then the widely linear estimator is written as,

$$\hat{\mathrm{y}} = \boldsymbol{\varphi}^H \tilde{\mathbf{x}}.$$

- Adopting the orthogonality condition in its complex formulation, i.e.,

$$\mathbb{E}\left[\tilde{\mathbf{x}}\mathrm{e}^*\right] = \mathbb{E}\left[\tilde{\mathbf{x}}\left(\mathrm{y} - \hat{\mathrm{y}}\right)^*\right] = \mathbf{0},$$

it turns out that the normal equations are equivalently written as

$$\begin{bmatrix} \Sigma_x & P_x \\ P_x^* & \Sigma_x^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_* \\ \boldsymbol{v}_* \end{bmatrix} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q}^* \end{bmatrix}, \text{ where } P_x := \mathbb{E}[\mathbf{x}\mathbf{x}^T], \ \boldsymbol{q} := \mathbb{E}[\mathbf{x}\mathrm{y}].$$

Extension to Complex-Valued Variables

- To distinguish from the complex linear estimation, Eq. (8) is known as widely linear complex-valued estimator. Note that in this case, both $\mathbf{x}$ as well as its complex conjugate, $\mathbf{x}^*$, are simultaneously used in order to cover all possible solutions, as those are dictated by the general linear formulation in a vector space.
- Let us now define,

$$\boldsymbol{\varphi} := \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{v} \end{bmatrix} \text{ and } \tilde{\mathbf{x}} := \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix}.$$

Then the widely linear estimator is written as,

$$\hat{\mathrm{y}} = \boldsymbol{\varphi}^H \tilde{\mathbf{x}}.$$

- Adopting the orthogonality condition in its complex formulation, i.e.,

$$\mathbb{E}\left[\tilde{\mathbf{x}}\mathrm{e}^*\right] = \mathbb{E}\left[\tilde{\mathbf{x}}\left(\mathrm{y} - \hat{\mathrm{y}}\right)^*\right] = \mathbf{0},$$

it turns out that the normal equations are equivalently written as

$$\begin{bmatrix} \Sigma_x & P_x \\ P_x^* & \Sigma_x^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_* \\ \boldsymbol{v}_* \end{bmatrix} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q}^* \end{bmatrix}, \text{ where } P_x := \mathbb{E}[\mathbf{x}\mathbf{x}^T],\ \boldsymbol{q} := \mathbb{E}[\mathbf{x}\mathrm{y}].$$

Extension to Complex-Valued Variables

- Circularity conditions: The matrix P_x is known as the pseudo covariance/autocorrelation matrix. If in the latter orthogonality condition, for the widely linear estimator, one sets

$$P_x = 0 \text{ and } \boldsymbol{q} = \mathbf{0}$$

it leads to $\boldsymbol{v}_* = \mathbf{0}$, and the task becomes equivalent to the complex linear estimator. In this case, we say that the input-output variables are jointly circular and the input variables in $\mathbf{x}$ obey the second order circular condition.

- A stronger condition for circularity is based on the pdf of a complex random variable: A random variable x is circular (or strictly circular) if x and $\mathrm{x}e^{j\phi}$ are distributed according to the same pdf; that is, the pdf is rotationally invariant. Strict circularity implies the second order circularity, but the converse is not always true. Figure (a) shows the scatter plot of points generated by a circularly distributed variable and Figure (b) corresponds to a non-circular one.

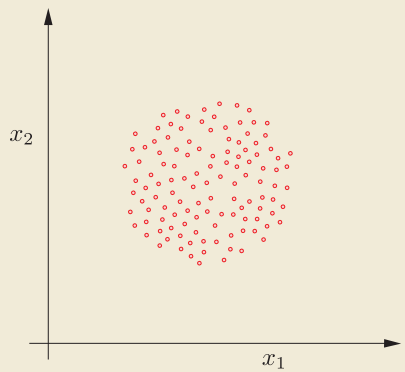
Extension to Complex-Valued Variables

- Circularity conditions: The matrix P_x is known as the pseudo covariance/autocorrelation matrix. If in the latter orthogonality condition, for the widely linear estimator, one sets

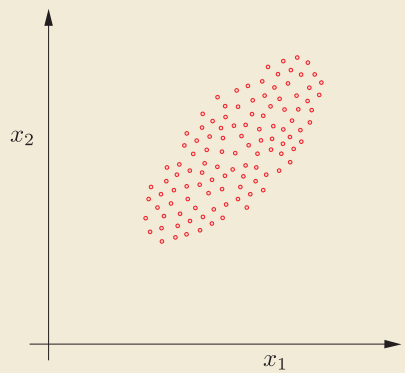
$$P_x = 0 \text{ and } \boldsymbol{q} = \mathbf{0}$$

it leads to $\boldsymbol{v}_* = \mathbf{0}$, and the task becomes equivalent to the complex linear estimator. In this case, we say that the input-output variables are jointly circular and the input variables in $\mathbf{x}$ obey the second order circular condition.

- A stronger condition for circularity is based on the pdf of a complex random variable: A random variable x is circular (or strictly circular) if x and $\mathrm{x}e^{j\phi}$ are distributed according to the same pdf; that is, the pdf is rotationally invariant. Strict circularity implies the second order circularity, but the converse is not always true. Figure (a) shows the scatter plot of points generated by a circularly distributed variable and Figure (b) corresponds to a non-circular one.



(a)



(b)

Mean-Square Error Estimation of Linear Models

- We now turn our attention to the case where the underlying model, that relates the input-output variables, is a linear one. So far, we have been concerned with the linear estimation task. At no point in the stage of our discussion, the generation model of the data was brought in. We just adopted a linear estimator and obtained the MSE solution for it. In contrast, the emphasis here is on cases where the input-output variables **are related** via a linear **data generation** model.
- Let us assume that we are given two jointly distributed random vectors, $\mathbf{y}$ and $\boldsymbol{\theta}$, which are related according to the following linear model,

$$\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\eta},$$

where $\boldsymbol{\eta}$ denotes the set of the involved noise variables. Note that such a model covers the case of our familiar regression task, where the unknown parameters $\boldsymbol{\theta}$ are considered random, which is in line with the Bayesian philosophy. Once more, we assume zero-mean vectors; otherwise the respective mean values are subtracted.

- The dimensions of $\mathbf{y}$ ($\boldsymbol{\eta}$) and $\boldsymbol{\theta}$ may not necessarily be the same; to be in line with the notation used in Chapter 3, let $\mathbf{y}, \boldsymbol{\eta} \in \mathbb{R}^N$ and $\boldsymbol{\theta} \in \mathbb{R}^l$. Hence X is a $N \times l$ matrix. Note that, matrix X is considered to be deterministic and not a random one.

Mean-Square Error Estimation of Linear Models

- We now turn our attention to the case where the underlying model, that relates the input-output variables, is a linear one. So far, we have been concerned with the linear estimation task. At no point in the stage of our discussion, the generation model of the data was brought in. We just adopted a linear estimator and obtained the MSE solution for it. In contrast, the emphasis here is on cases where the input-output variables **are related** via a linear **data generation** model.
- Let us assume that we are given two jointly distributed random vectors, $\mathbf{y}$ and $\boldsymbol{\theta}$, which are related according to the following linear model,

$$\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\eta},$$

where $\boldsymbol{\eta}$ denotes the set of the involved noise variables. Note that such a model covers the case of our familiar regression task, where the unknown parameters $\boldsymbol{\theta}$ are considered random, which is in line with the Bayesian philosophy. Once more, we assume zero-mean vectors; otherwise the respective mean values are subtracted.

- The dimensions of $\mathbf{y}$ ($\boldsymbol{\eta}$) and $\boldsymbol{\theta}$ may not necessarily be the same; to be in line with the notation used in Chapter 3, let $\mathbf{y}, \boldsymbol{\eta} \in \mathbb{R}^N$ and $\boldsymbol{\theta} \in \mathbb{R}^l$. Hence X is a $N \times l$ matrix. Note that, matrix X is considered to be deterministic and not a random one.

Mean-Square Error Estimation of Linear Models

- We now turn our attention to the case where the underlying model, that relates the input-output variables, is a linear one. So far, we have been concerned with the linear estimation task. At no point in the stage of our discussion, the generation model of the data was brought in. We just adopted a linear estimator and obtained the MSE solution for it. In contrast, the emphasis here is on cases where the input-output variables **are related** via a linear **data generation** model.
- Let us assume that we are given two jointly distributed random vectors, $\mathbf{y}$ and $\boldsymbol{\theta}$, which are related according to the following linear model,

$$\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\eta},$$

where $\boldsymbol{\eta}$ denotes the set of the involved noise variables. Note that such a model covers the case of our familiar regression task, where the unknown parameters $\boldsymbol{\theta}$ are considered random, which is in line with the Bayesian philosophy. Once more, we assume zero-mean vectors; otherwise the respective mean values are subtracted.

- The dimensions of $\mathbf{y}$ ($\boldsymbol{\eta}$) and $\boldsymbol{\theta}$ may not necessarily be the same; to be in line with the notation used in Chapter 3, let $\mathbf{y}, \boldsymbol{\eta} \in \mathbb{R}^N$ and $\boldsymbol{\theta} \in \mathbb{R}^l$. Hence X is a $N \times l$ matrix. Note that, matrix X is considered to be deterministic and not a random one.

Mean-Square Error Estimation of Linear Models

- Assume the covariance matrices of our zero-mean variables,

$$\Sigma_\theta = \mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^T],\ \Sigma_\eta = \mathbb{E}[\boldsymbol{\eta}\boldsymbol{\eta}^T],$$

to be known. The goal is to compute a matrix, H, of dimension $l \times N$, so that the linear estimator

$$\hat{\boldsymbol{\theta}} = H\mathbf{y},$$

minimizes the mean-square error cost,

$$J(H) := \mathbb{E}\left[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right] = \sum_{i=1}^{l} \mathbb{E}\left[|\theta_i - \hat{\theta}_i|^2\right].$$

- Note that this is a multichannel estimation task and it is equivalent with solving l optimization tasks, one for each component, θ_i, of $\boldsymbol{\theta}$.
- If we define the error vector as,

$$\boldsymbol{\varepsilon} := \boldsymbol{\theta} - \hat{\boldsymbol{\theta}},$$

then the cost function is equal to the trace of the corresponding error covariance matrix, i.e.,

$$J(H) := \text{trace}\left\{\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T]\right\}.$$

Mean-Square Error Estimation of Linear Models

- Assume the covariance matrices of our zero-mean variables,

$$\Sigma_\theta = \mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^T],\ \Sigma_\eta = \mathbb{E}[\boldsymbol{\eta}\boldsymbol{\eta}^T],$$

to be known. The goal is to compute a matrix, H, of dimension $l \times N$, so that the linear estimator

$$\hat{\boldsymbol{\theta}} = H\mathbf{y},$$

minimizes the mean-square error cost,

$$J(H) := \mathbb{E}\left[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right] = \sum_{i=1}^{l} \mathbb{E}\left[|\theta_i - \hat{\theta}_i|^2\right].$$

- Note that this is a multichannel estimation task and it is equivalent with solving l optimization tasks, one for each component, θ_i, of $\boldsymbol{\theta}$.
- If we define the error vector as,

$$\boldsymbol{\varepsilon} := \boldsymbol{\theta} - \hat{\boldsymbol{\theta}},$$

then the cost function is equal to the trace of the corresponding error covariance matrix, i.e.,

$$J(H) := \text{trace}\left\{\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T]\right\}.$$

Mean-Square Error Estimation of Linear Models

- Assume the covariance matrices of our zero-mean variables,

$$\Sigma_\theta = \mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^T],\ \Sigma_\eta = \mathbb{E}[\boldsymbol{\eta}\boldsymbol{\eta}^T],$$

to be known. The goal is to compute a matrix, H, of dimension $l \times N$, so that the linear estimator

$$\hat{\boldsymbol{\theta}} = H\mathbf{y},$$

minimizes the mean-square error cost,

$$J(H) := \mathbb{E}\left[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right] = \sum_{i=1}^{l} \mathbb{E}\left[|\theta_i - \hat{\theta}_i|^2\right].$$

- Note that this is a multichannel estimation task and it is equivalent with solving l optimization tasks, one for each component, θ_i, of $\boldsymbol{\theta}$.
- If we define the error vector as,

$$\boldsymbol{\varepsilon} := \boldsymbol{\theta} - \hat{\boldsymbol{\theta}},$$

then the cost function is equal to the trace of the corresponding error covariance matrix, i.e.,

$$J(H) := \text{trace}\left\{\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T]\right\}.$$

Mean-Square Error Estimation of Linear Models

- Performing minimization of the previous cost function, and following standard arguments, that we have already used before, we can show that (details in the text)

$$\hat{\boldsymbol{\theta}} = (\Sigma_\theta^{-1} + X^T \Sigma_\eta^{-1} X)^{-1} X^T \Sigma_\eta^{-1} \mathbf{y}. \tag{9}$$

- If we allow nonzero mean values for $\boldsymbol{\theta}$ and $\mathbf{y}$, it turns out that

$$\hat{\boldsymbol{\theta}} = \mathbb{E}[\hat{\boldsymbol{\theta}}] + \left(\Sigma_\theta^{-1} + X^T \Sigma_\eta^{-1} X\right)^{-1} X^T \Sigma_\eta^{-1} \left(\mathbf{y} - \mathbb{E}[\mathbf{y}]\right).$$

  The above is the same as the estimator resulting from the Bayesian inference approach (Chapter 3), provided that the covariance matrix of the prior (Gaussian) pdf is equal to Σ_θ and the corresponding mean $\boldsymbol{\theta}_0 = \mathbb{E}[\hat{\boldsymbol{\theta}}]$, for a zero-mean noise variable.

- We know from Chapter 3 that, the optimal MSE estimator of $\boldsymbol{\theta}$, given the values of $\mathbf{y}$, is given by

$$\mathbb{E}[\boldsymbol{\theta}|\mathbf{y}].$$

  Furthermore, if $\boldsymbol{\theta}$ and $\mathbf{y}$ are jointly Gaussian vectors, then the optimal estimator is linear (affine for nonzero mean variables) and it coincides with the above MSE linear estimator.

Mean-Square Error Estimation of Linear Models

- Performing minimization of the previous cost function, and following standard arguments, that we have already used before, we can show that (details in the text)

$$\hat{\boldsymbol{\theta}} = (\Sigma_\theta^{-1} + X^T \Sigma_\eta^{-1} X)^{-1} X^T \Sigma_\eta^{-1} \mathbf{y}. \tag{9}$$

- If we allow nonzero mean values for $\boldsymbol{\theta}$ and $\mathbf{y}$, it turns out that

$$\hat{\boldsymbol{\theta}} = \mathbb{E}[\hat{\boldsymbol{\theta}}] + \left(\Sigma_\theta^{-1} + X^T \Sigma_\eta^{-1} X\right)^{-1} X^T \Sigma_\eta^{-1} \left(\mathbf{y} - \mathbb{E}[\mathbf{y}]\right).$$

  The above is the same as the estimator resulting from the Bayesian inference approach (Chapter 3), provided that the covariance matrix of the prior (Gaussian) pdf is equal to Σ_θ and the corresponding mean $\boldsymbol{\theta}_0 = \mathbb{E}[\hat{\boldsymbol{\theta}}]$, for a zero-mean noise variable.

- We know from Chapter 3 that, the optimal MSE estimator of $\boldsymbol{\theta}$, given the values of $\mathbf{y}$, is given by

$$\mathbb{E}[\boldsymbol{\theta}|\mathbf{y}].$$

  Furthermore, if $\boldsymbol{\theta}$ and $\mathbf{y}$ are jointly Gaussian vectors, then the optimal estimator is linear (affine for nonzero mean variables) and it coincides with the above MSE linear estimator.

Mean-Square Error Estimation of Linear Models

- Performing minimization of the previous cost function, and following standard arguments, that we have already used before, we can show that (details in the text)

$$\hat{\boldsymbol{\theta}} = (\Sigma_\theta^{-1} + X^T \Sigma_\eta^{-1} X)^{-1} X^T \Sigma_\eta^{-1} \mathbf{y}. \tag{9}$$

- If we allow nonzero mean values for $\boldsymbol{\theta}$ and $\mathbf{y}$, it turns out that

$$\hat{\boldsymbol{\theta}} = \mathbb{E}[\hat{\boldsymbol{\theta}}] + \left(\Sigma_\theta^{-1} + X^T \Sigma_\eta^{-1} X\right)^{-1} X^T \Sigma_\eta^{-1} \left(\mathbf{y} - \mathbb{E}[\mathbf{y}]\right).$$

  The above is the same as the estimator resulting from the Bayesian inference approach (Chapter 3), provided that the covariance matrix of the prior (Gaussian) pdf is equal to Σ_θ and the corresponding mean $\boldsymbol{\theta}_0 = \mathbb{E}[\hat{\boldsymbol{\theta}}]$, for a zero-mean noise variable.

- We know from Chapter 3 that, the optimal MSE estimator of $\boldsymbol{\theta}$, given the values of $\mathbf{y}$, is given by

$$\mathbb{E}[\boldsymbol{\theta}|\mathbf{y}].$$

  Furthermore, if $\boldsymbol{\theta}$ and $\mathbf{y}$ are jointly Gaussian vectors, then the optimal estimator is linear (affine for nonzero mean variables) and it coincides with the above MSE linear estimator.

The Gauss-Markov Theorem

- We now turn our attention to the case where $\boldsymbol{\theta}$ in the regression model is considered to be an (unknown) constant, instead of a random variable. Thus, the linear model is now written as,

$$\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\eta}, \tag{10}$$

and the randomness of $\mathbf{y}$ is solely due to the noise $\boldsymbol{\eta}$, which is assumed to be zero-mean with covariance matrix Σ_η.

- The goal is to design an **unbiased** linear estimator, that minimizes the mean-square error, i.e.,

$$\hat{\theta} = H\mathbf{y}, \tag{11}$$

and select H such as

$$\begin{aligned} \text{minimize} \quad & \text{trace}\Big\{\mathbb{E}\big[(\boldsymbol{\theta}-\hat{\theta})(\boldsymbol{\theta}-\hat{\theta})^T\big]\Big\} \\ \text{s.t.} \quad & \mathbb{E}[\hat{\theta}] = \boldsymbol{\theta}. \end{aligned} \tag{12}$$

- From (10) and (11), we get that

$$\mathbb{E}[\hat{\theta}] = H\mathbb{E}[\mathbf{y}] = H\mathbb{E}\big[(X\boldsymbol{\theta}+\boldsymbol{\eta})\big] = HX\boldsymbol{\theta},$$

which implies that the unbiased constraint is equivalent to

$$HX = I.$$

The Gauss-Markov Theorem

- We now turn our attention to the case where $\boldsymbol{\theta}$ in the regression model is considered to be an (unknown) constant, instead of a random variable. Thus, the linear model is now written as,

$$\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\eta}, \tag{10}$$

and the randomness of $\mathbf{y}$ is solely due to the noise $\boldsymbol{\eta}$, which is assumed to be zero-mean with covariance matrix Σ_η.

- The goal is to design an **unbiased** linear estimator, that minimizes the mean-square error, i.e.,

$$\hat{\boldsymbol{\theta}} = H\mathbf{y}, \tag{11}$$

and select H such as

$$\begin{aligned} \text{minimize} \quad & \operatorname{trace}\Big\{\mathbb{E}\big[(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})^T\big]\Big\} \\ \text{s.t.} \quad & \mathbb{E}[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta}. \end{aligned} \tag{12}$$

- From (10) and (11), we get that

$$\mathbb{E}[\hat{\boldsymbol{\theta}}] = H\mathbb{E}[\mathbf{y}] = H\mathbb{E}\big[(X\boldsymbol{\theta}+\boldsymbol{\eta})\big] = HX\boldsymbol{\theta},$$

which implies that the unbiased constraint is equivalent to

$$HX = I.$$

The Gauss-Markov Theorem

- We now turn our attention to the case where $\boldsymbol{\theta}$ in the regression model is considered to be an (unknown) constant, instead of a random variable. Thus, the linear model is now written as,

$$\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\eta}, \tag{10}$$

and the randomness of $\mathbf{y}$ is solely due to the noise $\boldsymbol{\eta}$, which is assumed to be zero-mean with covariance matrix Σ_η.

- The goal is to design an **unbiased** linear estimator, that minimizes the mean-square error, i.e.,

$$\hat{\boldsymbol{\theta}} = H\mathbf{y}, \tag{11}$$

and select H such as

$$\begin{aligned} \text{minimize} \quad & \text{trace}\left\{\mathbb{E}\big[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\big]\right\} \\ \text{s.t.} \quad & \mathbb{E}[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta}. \end{aligned} \tag{12}$$

- From (10) and (11), we get that

$$\mathbb{E}[\hat{\boldsymbol{\theta}}] = H\mathbb{E}[\mathbf{y}] = H\mathbb{E}\big[(X\boldsymbol{\theta} + \boldsymbol{\eta})\big] = HX\boldsymbol{\theta},$$

which implies that the unbiased constraint is equivalent to

$$HX = I.$$

The Gauss-Markov Theorem

- Employing (11), the error vector becomes

$$\boldsymbol{\epsilon} = \boldsymbol{\theta} - \hat{\theta} = \boldsymbol{\theta} - H\mathbf{y} = \boldsymbol{\theta} - H(X\boldsymbol{\theta} + \boldsymbol{\eta}) = H\boldsymbol{\eta}.$$

  Hence, the constrained minimization in (12) can now be written as

$$\begin{aligned} H_* &= \arg\min_H \operatorname{trace}\{H\Sigma_\eta H^T\}, \\ \text{s.t.} &\quad HX = I. \end{aligned}$$

- Employing Lagrange multipliers, the optimal MSE linear **unbiased** estimator results as,

$$\hat{\theta} = (X^T\Sigma_\eta^{-1}X)^{-1}X^T\Sigma_\eta^{-1}\mathbf{y},$$

  and it is also known as the best linear unbiased estimator (BLUE) or the minimum variance unbiased linear estimator. For complex-valued variables, the transposition is simply replaced by the Hermitian one.
- The BLUE coincides with the maximum likelihood estimator, if $\boldsymbol{\eta}$ follows a multivariate Gaussian distribution; under this assumption, the Cramér-Rao bound is achieved. If this is not the case, there may be another unbiased estimator (nonlinear), which results in lower MSE. Moreover, there may be a biased estimator that results in lower MSE.

The Gauss-Markov Theorem

- Employing (11), the error vector becomes

$$\boldsymbol{\epsilon} = \boldsymbol{\theta} - \hat{\theta} = \boldsymbol{\theta} - H\mathbf{y} = \boldsymbol{\theta} - H(X\boldsymbol{\theta} + \boldsymbol{\eta}) = H\boldsymbol{\eta}.$$

  Hence, the constrained minimization in (12) can now be written as

$$\begin{aligned} H_* \quad &= \quad \arg\min_{H} \operatorname{trace}\{H\Sigma_\eta H^T\}, \\ \text{s.t.} \quad &\quad HX = I. \end{aligned}$$

- Employing Lagrange multipliers, the optimal MSE linear **unbiased** estimator results as,

$$\hat{\theta} = (X^T\Sigma_\eta^{-1}X)^{-1}X^T\Sigma_\eta^{-1}\mathbf{y},$$

  and it is also known as the best linear unbiased estimator (BLUE) or the minimum variance unbiased linear estimator. For complex-valued variables, the transposition is simply replaced by the Hermitian one.

- The BLUE coincides with the maximum likelihood estimator, if $\boldsymbol{\eta}$ follows a multivariate Gaussian distribution; under this assumption, the Cramér-Rao bound is achieved. If this is not the case, there may be another unbiased estimator (nonlinear), which results in lower MSE. Moreover, there may be a biased estimator that results in lower MSE.

The Gauss-Markov Theorem

- Employing (11), the error vector becomes

$$\boldsymbol{\epsilon} = \boldsymbol{\theta} - \hat{\theta} = \boldsymbol{\theta} - H\mathbf{y} = \boldsymbol{\theta} - H(X\boldsymbol{\theta} + \boldsymbol{\eta}) = H\boldsymbol{\eta}.$$

  Hence, the constrained minimization in (12) can now be written as

$$\begin{aligned} H_* \quad &= \quad \arg\min_{H} \operatorname{trace}\{H\Sigma_\eta H^T\}, \\ \text{s.t.} \quad &\quad HX = I. \end{aligned}$$

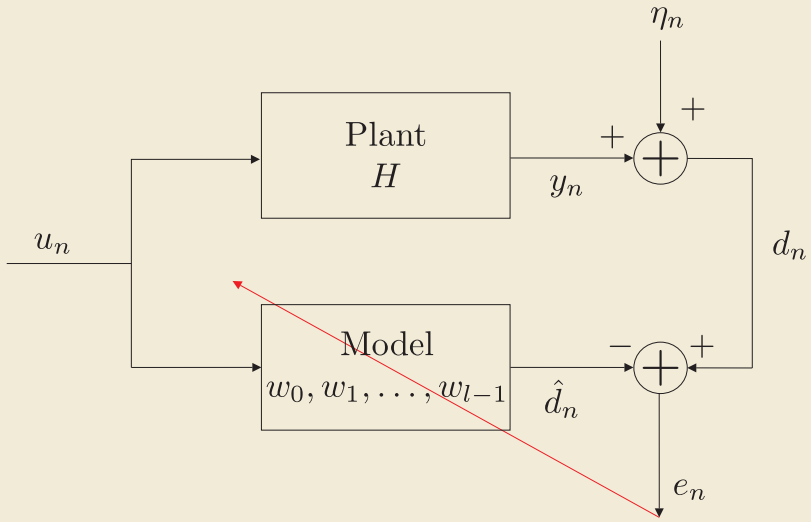
- Employing Lagrange multipliers, the optimal MSE linear **unbiased** estimator results as,

$$\hat{\theta} = (X^T\Sigma_\eta^{-1}X)^{-1}X^T\Sigma_\eta^{-1}\mathbf{y},$$

  and it is also known as the best linear unbiased estimator (BLUE) or the minimum variance unbiased linear estimator. For complex-valued variables, the transposition is simply replaced by the Hermitian one.
- The BLUE coincides with the maximum likelihood estimator, if $\boldsymbol{\eta}$ follows a multivariate Gaussian distribution; under this assumption, the Cramér-Rao bound is achieved. If this is not the case, there may be another unbiased estimator (nonlinear), which results in lower MSE. Moreover, there may be a biased estimator that results in lower MSE.

Example: Channel Identification

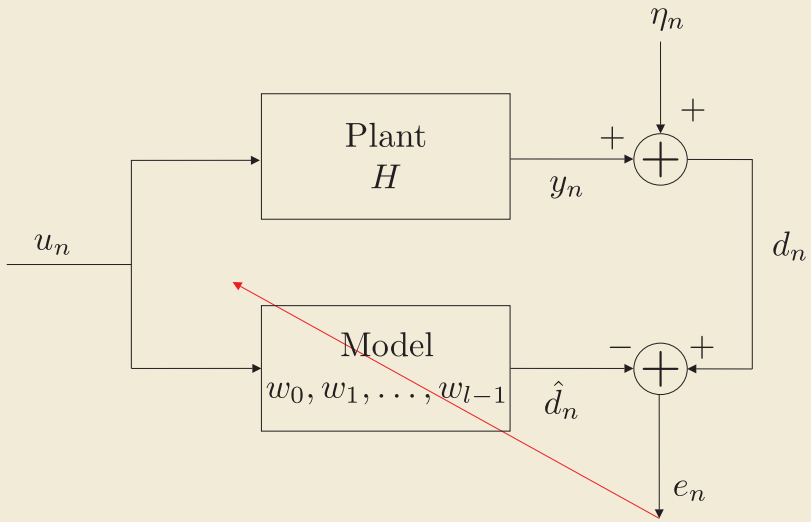
- The task is that of system identification and it is illustrated, once again for convenience, in the figure below



- Assume that we have access to a set of input-output observations, u_n and $d_n, n = 0, 1, 2, \ldots, N-1$. Moreover, we are given that the impulse response of the system/plant comprises l taps and it is zero-mean and its covariance matrix is Σ_w. Also, the second order statistics of the zero-mean noise are also known and we are given its covariance matrix, Σ_η.
- Then, assuming that the plant starts from zero initial conditions, we can adopt the following model relating the involved random variables, which is in line with the previously discussed model, i.e., $\mathbf{y} = H\boldsymbol{\theta} + \boldsymbol{\eta}$.

Example: Channel Identification

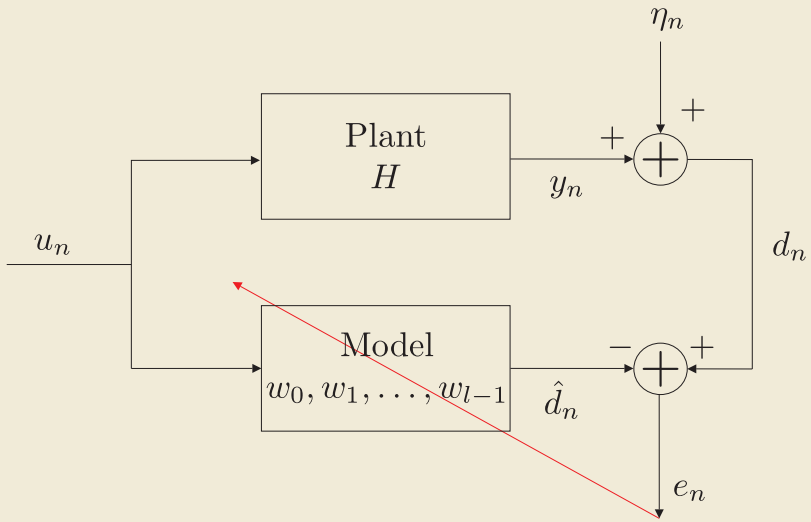
- The task is that of system identification and it is illustrated, once again for convenience, in the figure below



- Assume that we have access to a set of input-output observations, u_n and $d_n, n = 0, 1, 2, \ldots, N-1$. Moreover, we are given that the impulse response of the system/plant comprises l taps and it is zero-mean and its covariance matrix is Σ_w. Also, the second order statistics of the zero-mean noise are also known and we are given its covariance matrix, Σ_η.
- Then, assuming that the plant starts from zero initial conditions, we can adopt the following model relating the involved random variables, which is in line with the previously discussed model, i.e., $\mathbf{y} = H\boldsymbol{\theta} + \boldsymbol{\eta}$.

Example: Channel Identification

- The task is that of system identification and it is illustrated, once again for convenience, in the figure below



- Assume that we have access to a set of input-output observations, u_n and $d_n, n = 0, 1, 2, \ldots, N-1$. Moreover, we are given that the impulse response of the system/plant comprises l taps and it is zero-mean and its covariance matrix is Σ_w. Also, the second order statistics of the zero-mean noise are also known and we are given its covariance matrix, Σ_η.
- Then, assuming that the plant starts from zero initial conditions, we can adopt the following model relating the involved random variables, which is in line with the previously discussed model, i.e., $\mathbf{y} = H\boldsymbol{\theta} + \boldsymbol{\eta}$.

Example: Channel Identification

- The input-output relation of the task is written as,

$$\mathbf{d} := \begin{bmatrix} \mathrm{d}_0 \\ \mathrm{d}_1 \\ \vdots \\ \mathrm{d}_{l-1} \\ \vdots \\ \mathrm{d}_{N-1} \end{bmatrix} = U \begin{bmatrix} \mathrm{w}_0 \\ \mathrm{w}_1 \\ \vdots \\ \mathrm{w}_{l-1} \end{bmatrix} + \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_{l-1} \\ \vdots \\ \eta_{N-1} \end{bmatrix},$$

where

$$U := \begin{bmatrix} u_0 & 0 & 0 & \cdots & 0 \\ u_1 & u_0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{l-1} & u_{l-2} & \cdots & \cdots & u_0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{N-1} & \cdots & \cdots & \cdots & u_{N-l+1} \end{bmatrix}.$$

- Note that U is treated deterministically. Then, recalling (9) (i.e., $\hat{\theta} = (\Sigma_\theta^{-1} + X^T \Sigma_\eta^{-1} X)^{-1} X^T \Sigma_\eta^{-1} \mathbf{y}$) and plugging in the set of obtained measurements, the following estimate results

$$\hat{\boldsymbol{w}} = (\Sigma_w^{-1} + U^T \Sigma_\eta^{-1} U) U^T \Sigma_\eta^{-1} \boldsymbol{d},$$

where $\boldsymbol{d}$ is the vector of the desired response observations.

Example: Channel Identification

- The input-output relation of the task is written as,

$$\mathbf{d} := \begin{bmatrix} \mathrm{d}_0 \\ \mathrm{d}_1 \\ \vdots \\ \mathrm{d}_{l-1} \\ \vdots \\ \mathrm{d}_{N-1} \end{bmatrix} = U \begin{bmatrix} \mathrm{w}_0 \\ \mathrm{w}_1 \\ \vdots \\ \mathrm{w}_{l-1} \end{bmatrix} + \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_{l-1} \\ \vdots \\ \eta_{N-1} \end{bmatrix},$$

where

$$U := \begin{bmatrix} u_0 & 0 & 0 & \cdots & 0 \\ u_1 & u_0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{l-1} & u_{l-2} & \cdots & \cdots & u_0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{N-1} & \cdots & \cdots & \cdots & u_{N-l+1} \end{bmatrix}.$$

- Note that U is treated deterministically. Then, recalling (9) (i.e., $\hat{\theta} = (\Sigma_\theta^{-1} + X^T \Sigma_\eta^{-1} X)^{-1} X^T \Sigma_\eta^{-1} \mathbf{y}$) and plugging in the set of obtained measurements, the following estimate results

$$\hat{\boldsymbol{w}} = (\Sigma_w^{-1} + U^T \Sigma_\eta^{-1} U) U^T \Sigma_\eta^{-1} \boldsymbol{d},$$

where $\boldsymbol{d}$ is the vector of the desired response observations.

Constrained Linear Estimation: The Beamforming Case

- We have already dealt with a constrained linear estimation task, in our effort to obtain an unbiased estimator of a fixed-value parameter vector. In the current subsection, we will see that the procedure developed there is readily applicable for cases where the unknown parameter vector is required to respect more general linear constraints. The case will be demonstrated in the context of beamforming, one of the major application areas in modern communications.
- A beamformer comprises a set of antenna elements. We consider the case where the antenna elements are uniformly spaced along a straight line. The goal is to linearly combine the signals received by the individual antenna elements, so as:
  - to turn the main beam of the array to a specific direction in space,
  - to optimally reduce the noise.

  The first goal imposes a constraint to the designer, which will guarantee that the gain of the array is high for the specific desired direction; for the second goal, we will adopt MSE arguments.

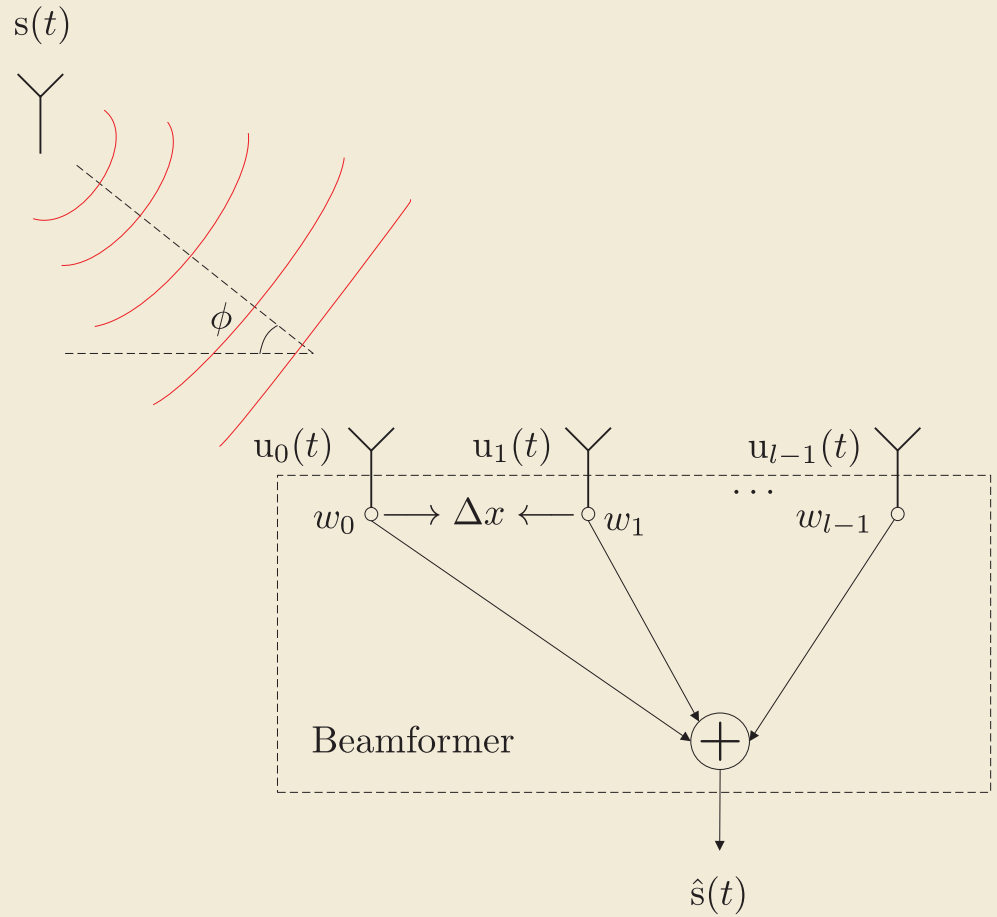
Constrained Linear Estimation: The Beamforming Case

- We have already dealt with a constrained linear estimation task, in our effort to obtain an unbiased estimator of a fixed-value parameter vector. In the current subsection, we will see that the procedure developed there is readily applicable for cases where the unknown parameter vector is required to respect more general linear constraints. The case will be demonstrated in the context of beamforming, one of the major application areas in modern communications.
- A beamformer comprises a set of antenna elements. We consider the case where the antenna elements are uniformly spaced along a straight line. The goal is to linearly combine the signals received by the individual antenna elements, so as:
  - to turn the main beam of the array to a specific direction in space,
  - to optimally reduce the noise.

  The first goal imposes a constraint to the designer, which will guarantee that the gain of the array is high for the specific desired direction; for the second goal, we will adopt MSE arguments.

Constrained Linear Estimation: The Beamforming Case

- The figure below illustrates the basic block diagram of the beamforming task.



The task of the beamformer is to obtain estimates of the weights $w_0, \ldots, w_{l-1}$, so that to minimize the effect of noise and at the same time to impose a constraint which, in the absence of noise, would leave signals impinging the array from the desired angle, ϕ, unaffected.

- In a more formal way, assume that the transmitter is far enough, so that to guarantee that the wavefronts that the array "sees" are planar. Let $\mathrm{s}(t)$ be the information random process transmitted at a carrier frequency, ω_c, hence the modulated signal is

$$\mathrm{r}(t) = \mathrm{s}(t)e^{j\omega_c t}.$$

If Δx is the distance between successive elements of the array, then a wavefront that arrives at time t_0 at the first element will reach the i-th element delayed by

$$\Delta_{t_i} = t_i - t_0 = i\frac{\Delta x \cos\phi}{c}, \ i = 0, 1, ..., l-1,$$

where c is the speed of propagation, ϕ is the angle formed by the array and the direction propagation of the wavefronts and l the number of array elements; we know from our basic electromagnetic courses that

$$c = \frac{\omega_c \lambda}{2\pi},$$

where λ is the respective wavelength.

- Taking a snapshot at time t, and assuming a relatively low time signal variation, the signal received from direction ϕ at the i-th element will be

$$\begin{aligned} \mathrm{r}_i(t) &= \mathrm{s}(t-\Delta t_i)e^{j\omega_c(t-i\frac{2\pi\Delta x\cos\phi}{\omega_c\lambda})} \\ &\simeq \mathrm{s}(t)e^{j\omega_c t}e^{-2\pi j\frac{i\Delta x\cos\phi}{\lambda}}, \ i = 0, 1, ..., l-1. \end{aligned}$$

- In a more formal way, assume that the transmitter is far enough, so that to guarantee that the wavefronts that the array "sees" are planar. Let $\mathrm{s}(t)$ be the information random process transmitted at a carrier frequency, ω_c, hence the modulated signal is
$$\mathrm{r}(t) = \mathrm{s}(t)e^{j\omega_c t}.$$
If Δx is the distance between successive elements of the array, then a wavefront that arrives at time t_0 at the first element will reach the i-th element delayed by
$$\Delta_{t_i} = t_i - t_0 = i\frac{\Delta x \cos\phi}{c}, \ i = 0, 1, ..., l-1,$$
where c is the speed of propagation, ϕ is the angle formed by the array and the direction propagation of the wavefronts and l the number of array elements; we know from our basic electromagnetic courses that
$$c = \frac{\omega_c \lambda}{2\pi},$$
where λ is the respective wavelength.
- Taking a snapshot at time t, and assuming a relatively low time signal variation, the signal received from direction ϕ at the i-th element will be
$$\begin{aligned} \mathrm{r}_i(t) &= \mathrm{s}(t-\Delta t_i)e^{j\omega_c(t-i\frac{2\pi\Delta x\cos\phi}{\omega_c\lambda})} \\ &\simeq \mathrm{s}(t)e^{j\omega_c t}e^{-2\pi j\frac{i\Delta x\cos\phi}{\lambda}}, \ i = 0, 1, ..., l-1. \end{aligned}$$

Constrained Linear Estimation: The Beamforming Case

- After converting the received signals in the baseband (multiplying by $e^{-j\omega_c t}$), the vector of the received signals (one per array element), at time t, can be written in the following linear regression-type formulation,

$$\mathbf{u}(t) := \begin{bmatrix} \mathrm{u}_0(t) \\ \mathrm{u}_1(t) \\ \vdots \\ \mathrm{u}_{l-1}(t) \end{bmatrix} = \boldsymbol{x}\mathrm{s}(t) + \boldsymbol{\eta}(t), \quad \text{where} \quad \boldsymbol{x} := \begin{bmatrix} 1 \\ e^{-2\pi j \frac{\Delta x \cos\phi}{\lambda}} \\ \vdots \\ e^{-2\pi j \frac{(l-1)\Delta x \cos\phi}{\lambda}} \end{bmatrix},$$

and the vector $\boldsymbol{\eta}(t)$ contains the additive noise plus any other interference due to signals coming from directions **other than** ϕ, i.e.,

$$\boldsymbol{\eta}(t) = [\eta_0(t), ..., \eta_{l-1}(t)]^T,$$

and it is assumed to be of zero mean; $\boldsymbol{x}$ is also known as the steering vector.

- The output of the beamformer, acting on the input vector signal, will be

$$\hat{\mathrm{s}}(t) = \boldsymbol{w}^H \mathbf{u}(t),$$

where the Hermitian operation has to be used, since now the involved signals are complex-valued.

Constrained Linear Estimation: The Beamforming Case

- After converting the received signals in the baseband (multiplying by $e^{-j\omega_c t}$), the vector of the received signals (one per array element), at time t, can be written in the following linear regression-type formulation,

$$\mathbf{u}(t) := \begin{bmatrix} \mathrm{u}_0(t) \\ \mathrm{u}_1(t) \\ \vdots \\ \mathrm{u}_{l-1}(t) \end{bmatrix} = \boldsymbol{x}\mathrm{s}(t) + \boldsymbol{\eta}(t), \quad \text{where} \quad \boldsymbol{x} := \begin{bmatrix} 1 \\ e^{-2\pi j \frac{\Delta x \cos\phi}{\lambda}} \\ \vdots \\ e^{-2\pi j \frac{(l-1)\Delta x \cos\phi}{\lambda}} \end{bmatrix},$$

  and the vector $\boldsymbol{\eta}(t)$ contains the additive noise plus any other interference due to signals coming from directions **other than** ϕ, i.e.,

$$\boldsymbol{\eta}(t) = [\eta_0(t), ..., \eta_{l-1}(t)]^T,$$

  and it is assumed to be of zero mean; $\boldsymbol{x}$ is also known as the steering vector.

- The output of the beamformer, acting on the input vector signal, will be

$$\hat{\mathrm{s}}(t) = \boldsymbol{w}^H \mathbf{u}(t),$$

  where the Hermitian operation has to be used, since now the involved signals are complex-valued.

Constrained Linear Estimation: The Beamforming Case

- We will first impose the constraint. Ideally, in the absence of noise, one would like to recover signals, that impinge on the array from the desired direction, ϕ, **exactly**. Thus, $\boldsymbol{w}$ should satisfy the following constraint

$$\boldsymbol{w}^H \boldsymbol{x} = 1,$$

which guarantees that $\hat{\mathrm{s}}(t) = \mathrm{s}(t)$ in the absence of noise. To account for the noise, we require the MSE,

$$\mathbb{E}\big[|\mathrm{s}(t) - \hat{\mathrm{s}}(t)|^2\big] = \mathbb{E}\big[|\mathrm{s}(t) - \boldsymbol{w}^H \mathbf{u}(t)|^2\big],$$

to be minimized. However,

$$\mathrm{s}(t) - \boldsymbol{w}^H \mathbf{u}(t) = \mathrm{s}(t) - \boldsymbol{w}^H \big(\boldsymbol{x}\mathrm{s}(t) + \boldsymbol{\eta}(t)\big) = -\boldsymbol{w}^H \boldsymbol{\eta}(t).$$

Hence, the optimal $\boldsymbol{w}_*$ results by the following constrained task

$$\begin{aligned} \boldsymbol{w}_* \quad &:= \quad \arg\min_{\boldsymbol{w}} (\boldsymbol{w}^H \Sigma_\eta \boldsymbol{w}), \\ \text{s.t.} \quad &\quad \boldsymbol{w}^H \boldsymbol{x} = 1. \end{aligned}$$

- Employing Lagrange multipliers, we finally obtain,

$$\boldsymbol{w}_*^H = \frac{\boldsymbol{x}^H \Sigma_\eta^{-1}}{\boldsymbol{x}^H \Sigma_\eta^{-1} \boldsymbol{x}}, \quad \text{and} \quad \hat{\mathrm{s}}(t) = \boldsymbol{w}_*^H \mathbf{u}(t) = \frac{\boldsymbol{x}^H \Sigma_\eta^{-1} \mathbf{u}(t)}{\boldsymbol{x}^H \Sigma_\eta^{-1} \boldsymbol{x}}.$$

Constrained Linear Estimation: The Beamforming Case

- We will first impose the constraint. Ideally, in the absence of noise, one would like to recover signals, that impinge on the array from the desired direction, ϕ, **exactly**. Thus, $\boldsymbol{w}$ should satisfy the following constraint

$$\boldsymbol{w}^H \boldsymbol{x} = 1,$$

which guarantees that $\hat{\mathrm{s}}(t) = \mathrm{s}(t)$ in the absence of noise. To account for the noise, we require the MSE,

$$\mathbb{E}\big[|\mathrm{s}(t) - \hat{\mathrm{s}}(t)|^2\big] = \mathbb{E}\big[|\mathrm{s}(t) - \boldsymbol{w}^H \mathbf{u}(t)|^2\big],$$

to be minimized. However,

$$\mathrm{s}(t) - \boldsymbol{w}^H \mathbf{u}(t) = \mathrm{s}(t) - \boldsymbol{w}^H \big(\boldsymbol{x}\mathrm{s}(t) + \boldsymbol{\eta}(t)\big) = -\boldsymbol{w}^H \boldsymbol{\eta}(t).$$

Hence, the optimal $\boldsymbol{w}_*$ results by the following constrained task

$$\begin{aligned} \boldsymbol{w}_* \quad &:= \quad \arg\min_{w} (\boldsymbol{w}^H \Sigma_\eta \boldsymbol{w}), \\ \text{s.t.} \quad &\quad \boldsymbol{w}^H \boldsymbol{x} = 1. \end{aligned}$$

- Employing Lagrange multipliers, we finally obtain,

$$\boldsymbol{w}_*^H = \frac{\boldsymbol{x}^H \Sigma_\eta^{-1}}{\boldsymbol{x}^H \Sigma_\eta^{-1} \boldsymbol{x}}, \quad \text{and} \quad \hat{\mathrm{s}}(t) = \boldsymbol{w}_*^H \mathbf{u}(t) = \frac{\boldsymbol{x}^H \Sigma_\eta^{-1} \mathbf{u}(t)}{\boldsymbol{x}^H \Sigma_\eta^{-1} \boldsymbol{x}}.$$

Constrained Linear Estimation: The Beamforming Case

- We will first impose the constraint. Ideally, in the absence of noise, one would like to recover signals, that impinge on the array from the desired direction, ϕ, **exactly**. Thus, $\boldsymbol{w}$ should satisfy the following constraint

$$\boldsymbol{w}^H \boldsymbol{x} = 1,$$

which guarantees that $\hat{\mathrm{s}}(t) = \mathrm{s}(t)$ in the absence of noise. To account for the noise, we require the MSE,

$$\mathbb{E}\big[|\mathrm{s}(t) - \hat{\mathrm{s}}(t)|^2\big] = \mathbb{E}\big[|\mathrm{s}(t) - \boldsymbol{w}^H \mathbf{u}(t)|^2\big],$$

to be minimized. However,

$$\mathrm{s}(t) - \boldsymbol{w}^H \mathbf{u}(t) = \mathrm{s}(t) - \boldsymbol{w}^H \big(\boldsymbol{x}\mathrm{s}(t) + \boldsymbol{\eta}(t)\big) = -\boldsymbol{w}^H \boldsymbol{\eta}(t).$$

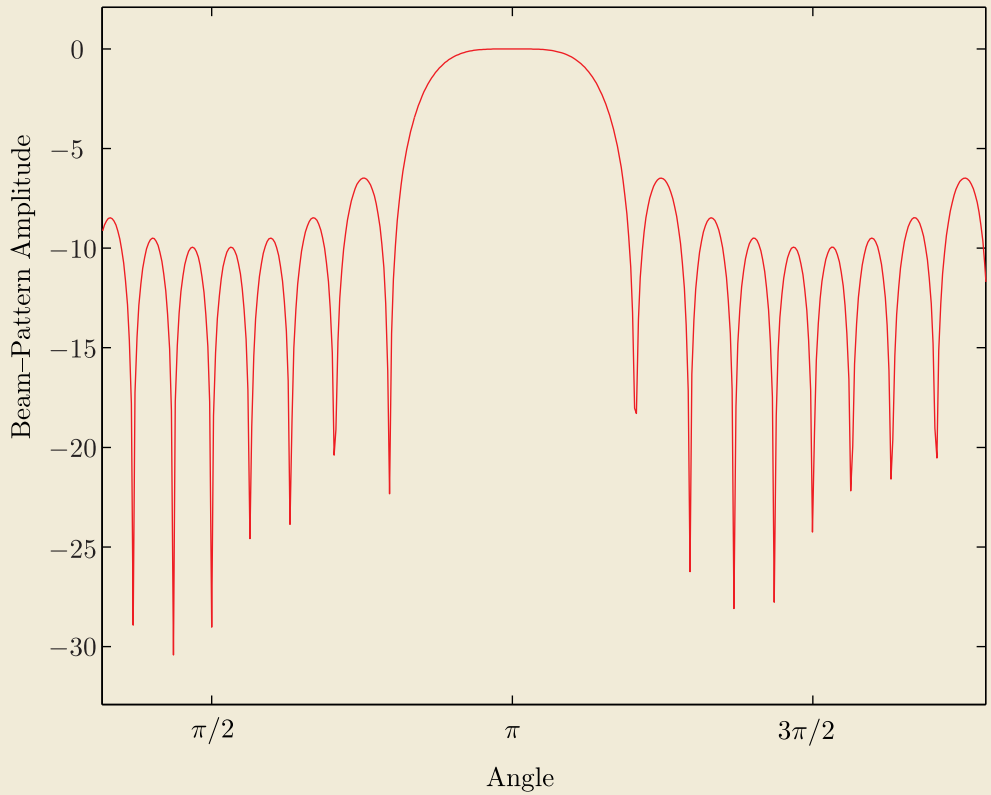
Hence, the optimal $\boldsymbol{w}_*$ results by the following constrained task

$$\begin{aligned} \boldsymbol{w}_* \quad &:= \quad \arg\min_{w} (\boldsymbol{w}^H \Sigma_\eta \boldsymbol{w}), \\ \text{s.t.} \quad &\quad \boldsymbol{w}^H \boldsymbol{x} = 1. \end{aligned}$$

- Employing Lagrange multipliers, we finally obtain,

$$\boldsymbol{w}_*^H = \frac{\boldsymbol{x}^H \Sigma_\eta^{-1}}{\boldsymbol{x}^H \Sigma_\eta^{-1} \boldsymbol{x}}, \quad \text{and} \quad \hat{\mathrm{s}}(t) = \boldsymbol{w}_*^H \mathbf{u}(t) = \frac{\boldsymbol{x}^H \Sigma_\eta^{-1} \mathbf{u}(t)}{\boldsymbol{x}^H \Sigma_\eta^{-1} \boldsymbol{x}}.$$

- The following figure shows the resulting beam-pattern as a function of the angle ϕ. The desired angle for designing the optimal set of weights is $\phi = \pi$. The number of antenna elements is $l = 10$, the spacing has been chosen as $\frac{\Delta x}{\lambda} = 0.5$ and the noise covariance matrix as $0.1I$. The beam-pattern amplitude is in dBs, i.e., the vertical axis shows $20\log_{10}(|\boldsymbol{w}_*^H \boldsymbol{x}(\phi)|)$. Thus, any signal arriving from directions, ϕ, not close to $\phi = \pi$, will be absorbed. The main beam can become sharper, if more elements are used.



Time Varying Statistics: Kalman Filtering

- So far, our discussion on the linear estimation task was limited to stationary environments, where the statistical properties of the involved random variables are assumed to be invariant with time. However, very often in practice, this is not the case and the statistical properties may be different at different time instants. As a matter of fact, a large part in the book is devoted to study the estimation task under time-varying environments.
- We will derive the basic recursions of the Kalman filter in the general context of two jointly distributed random vectors $\mathbf{y}$, $\mathbf{x}$. The task is to estimate the values of $\mathbf{x}$ given observations on $\mathbf{y}$. If $\boldsymbol{\eta}_n \in \mathbb{R}^l$, $\mathbf{v}_n, \in \mathbb{R}^k$ are noise sources, let $\mathbf{y} \in \mathbb{R}^k$ and $\mathbf{x} \in \mathbb{R}^l$ be linearly related via the following set of time-dependent recursions,

$$\mathbf{x}_n = F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n, \ n \geq 0, \quad \text{State Equation,} \tag{13}$$

$$\mathbf{y}_n = H_n \mathbf{x}_n + \mathbf{v}_n, \ n \geq 0, \quad \text{Output Equation.} \tag{14}$$

- The vector $\mathbf{x}_n$ is known as the state of the system at time n and $\mathbf{y}_n$ is the output, which is the vector which can be observed (measured); $\boldsymbol{\eta}_n$ and $\mathbf{v}_n$ are the noise vectors, known as process noise and measurement noise, respectively. Matrices F_n and H_n are of appropriate dimensions and they are assumed to be known.

Time Varying Statistics: Kalman Filtering

- So far, our discussion on the linear estimation task was limited to stationary environments, where the statistical properties of the involved random variables are assumed to be invariant with time. However, very often in practice, this is not the case and the statistical properties may be different at different time instants. As a matter of fact, a large part in the book is devoted to study the estimation task under time-varying environments.
- We will derive the basic recursions of the Kalman filter in the general context of two jointly distributed random vectors $\mathbf{y},\ \mathbf{x}$. The task is to estimate the values of $\mathbf{x}$ given observations on $\mathbf{y}$. If $\boldsymbol{\eta}_n \in \mathbb{R}^l$, $\mathbf{v}_n, \in \mathbb{R}^k$ are noise sources, let $\mathbf{y} \in \mathbb{R}^k$ and $\mathbf{x} \in \mathbb{R}^l$ be linearly related via the following set of time-dependent recursions,

$$\mathbf{x}_n = F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n,\ n \geq 0, \quad \text{State Equation,} \tag{13}$$

$$\mathbf{y}_n = H_n \mathbf{x}_n + \mathbf{v}_n,\ n \geq 0, \quad \text{Output Equation.} \tag{14}$$

- The vector $\mathbf{x}_n$ is known as the state of the system at time n and $\mathbf{y}_n$ is the output, which is the vector which can be observed (measured); $\boldsymbol{\eta}_n$ and $\mathbf{v}_n$ are the noise vectors, known as process noise and measurement noise, respectively. Matrices F_n and H_n are of appropriate dimensions and they are assumed to be known.

Time Varying Statistics: Kalman Filtering

- So far, our discussion on the linear estimation task was limited to stationary environments, where the statistical properties of the involved random variables are assumed to be invariant with time. However, very often in practice, this is not the case and the statistical properties may be different at different time instants. As a matter of fact, a large part in the book is devoted to study the estimation task under time-varying environments.
- We will derive the basic recursions of the Kalman filter in the general context of two jointly distributed random vectors $\mathbf{y}$, $\mathbf{x}$. The task is to estimate the values of $\mathbf{x}$ given observations on $\mathbf{y}$. If $\boldsymbol{\eta}_n \in \mathbb{R}^l$, $\mathbf{v}_n, \in \mathbb{R}^k$ are noise sources, let $\mathbf{y} \in \mathbb{R}^k$ and $\mathbf{x} \in \mathbb{R}^l$ be linearly related via the following set of time-dependent recursions,

$$\mathbf{x}_n = F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n, \ n \geq 0, \quad \text{State Equation,} \tag{13}$$

$$\mathbf{y}_n = H_n \mathbf{x}_n + \mathbf{v}_n, \ n \geq 0, \quad \text{Output Equation.} \tag{14}$$

- The vector $\mathbf{x}_n$ is known as the state of the system at time n and $\mathbf{y}_n$ is the output, which is the vector which can be observed (measured); $\boldsymbol{\eta}_n$ and $\mathbf{v}_n$ are the noise vectors, known as process noise and measurement noise, respectively. Matrices F_n and H_n are of appropriate dimensions and they are assumed to be known.

Time Varying Statistics: Kalman Filtering

- Observe that the so-called state equation provides the information related to the time-varying dynamics of the underlying system. It turns out that a large number of real world tasks can be brought into the form of (13), (14). The model is known as the state-space model for $\mathbf{y}_n$.
- In order to derive the time-varying estimator, $\hat{\mathbf{x}}_n$, given the measured values of $\mathbf{y}_n$, the following assumptions will be adopted:
  - $\mathbb{E}[\boldsymbol{\eta}_n \boldsymbol{\eta}_n^T] = Q_n, \ \mathbb{E}[\boldsymbol{\eta}_n \boldsymbol{\eta}_m^T] = O, \ n \neq m.$
  - $\mathbb{E}[\mathbf{v}_n \mathbf{v}_n^T] = R_n, \ \mathbb{E}[\mathbf{v}_n \mathbf{v}_m^T] = O, \ n \neq m.$
  - $\mathbb{E}[\boldsymbol{\eta}_n \mathbf{v}_m^T] = O, \ \forall n, m.$
  - $\mathbb{E}[\boldsymbol{\eta}_n] = \mathbb{E}[\mathbf{v}_n] = \mathbf{0}, \ \forall n,$

  where O denotes a matrix with zero elements. That is, $\boldsymbol{\eta}_n, \mathbf{v}_n$ are **uncorrelated**; moreover, noise vectors at **different time instants** are also considered **uncorrelated**. Versions where some of these conditions are relaxed are also available. The respective covariance matrices, Q_n, R_n, are assumed to be known.

Time Varying Statistics: Kalman Filtering

- Observe that the so-called state equation provides the information related to the time-varying dynamics of the underlying system. It turns out that a large number of real world tasks can be brought into the form of (13), (14). The model is known as the state-space model for $\mathbf{y}_n$.
- In order to derive the time-varying estimator, $\hat{\mathbf{x}}_n$, given the measured values of $\mathbf{y}_n$, the following assumptions will be adopted:
  - $\mathbb{E}[\boldsymbol{\eta}_n \boldsymbol{\eta}_n^T] = Q_n,\ \mathbb{E}[\boldsymbol{\eta}_n \boldsymbol{\eta}_m^T] = O,\ n \neq m.$
  - $\mathbb{E}[\mathbf{v}_n \mathbf{v}_n^T] = R_n,\ \mathbb{E}[\mathbf{v}_n \mathbf{v}_m^T] = O,\ n \neq m.$
  - $\mathbb{E}[\boldsymbol{\eta}_n \mathbf{v}_m^T] = O,\ \forall n, m.$
  - $\mathbb{E}[\boldsymbol{\eta}_n] = \mathbb{E}[\mathbf{v}_n] = \mathbf{0},\ \forall n,$

  where O denotes a matrix with zero elements. That is, $\boldsymbol{\eta}_n, \mathbf{v}_n$ are **uncorrelated**; moreover, noise vectors at **different time instants** are also considered **uncorrelated**. Versions where some of these conditions are relaxed are also available. The respective covariance matrices, Q_n, R_n, are assumed to be known.

Time Varying Statistics: Kalman Filtering

- The development of the time-varying estimation task evolves around two types of estimators for the state variables:
  - The first one is denoted as

    $$\hat{\mathbf{x}}_{n|n-1}.$$

    and it is based on all information that has been received up to and including time instant $n-1$; i.e., the observations of $\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{n-1}$. This is known as the a-priori or prior estimator.
  - The second estimator at time n is known as the posterior one, it is denoted as

    $$\hat{\mathbf{x}}_{n|n},$$

    and it is computed by updating $\hat{\mathbf{x}}_{n|n-1}$ after the observation of $\mathbf{y}_n$ has been received.
- For the development of the algorithm, assume that at time $n-1$ **all** required information is available; that is, the value of the posterior estimator and the respective error covariance matrix,

  $$\hat{\mathbf{x}}_{n-1|n-1}, \text{ and } P_{n-1|n-1} := \mathbb{E}[\mathbf{e}_{n-1|n-1}\mathbf{e}_{n-1|n-1}^T],$$

  where

  $$\mathbf{e}_{n-1|n-1} := \mathbf{x}_{n-1} - \hat{\mathbf{x}}_{n-1|n-1}.$$

Time Varying Statistics: Kalman Filtering

- The development of the time-varying estimation task evolves around two types of estimators for the state variables:
  - The first one is denoted as

    $$\hat{\mathbf{x}}_{n|n-1}.$$

    and it is based on all information that has been received up to and including time instant $n-1$; i.e., the observations of $\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{n-1}$. This is known as the a-priori or prior estimator.
  - The second estimator at time n is known as the posterior one, it is denoted as

    $$\hat{\mathbf{x}}_{n|n},$$

    and it is computed by updating $\hat{\mathbf{x}}_{n|n-1}$ after the observation of $\mathbf{y}_n$ has been received.
- For the development of the algorithm, assume that at time $n-1$ **all** required information is available; that is, the value of the posterior estimator and the respective error covariance matrix,

  $$\hat{\mathbf{x}}_{n-1|n-1}, \text{ and } P_{n-1|n-1} := \mathbb{E}[\mathbf{e}_{n-1|n-1}\mathbf{e}_{n-1|n-1}^T],$$

  where

  $$\mathbf{e}_{n-1|n-1} := \mathbf{x}_{n-1} - \hat{\mathbf{x}}_{n-1|n-1}.$$

- **The Kalman Filtering Algorithm**
  - Input: $F_n,\ H_n,\ Q_n,\ R_n,\ \boldsymbol{y}_n,\ n = 1, 2, ...$
  - Initialization:
    - $\hat{\boldsymbol{x}}_{1|0} = \mathbb{E}[\mathbf{x}_1]$
    - $P_{1|0} = \Pi_0$
  - **For** $n = 1, 2, ...,$ **Do**
    - $S_n = R_n + H_n P_{n|n-1} H_n^T$
    - $K_n = P_{n|n-1} H_n^T S_n^{-1}$
    - $\hat{\boldsymbol{x}}_{n|n} = \hat{\boldsymbol{x}}_{n|n-1} + K_n(\boldsymbol{y}_n - H_n \hat{\boldsymbol{x}}_{n|n-1})$
    - $P_{n|n} = P_{n|n-1} - K_n H_n P_{n|n-1}$
    - $\hat{\boldsymbol{x}}_{n+1|n} = F_{n+1} \hat{\boldsymbol{x}}_{n|n}$
    - $P_{n+1|n} = F_{n+1} P_{n|n} F_{n+1}^T + Q_{n+1}$
  - **End For**
- For complex-valued variables, transposition is replaced by the Hermitian one.
- Observe that $P_{n|n}$ is computed as the difference of two positive definite matrices; this may lead to a non-positive definite, $P_{n|n}$, due to numerical errors. This can cause the algorithm to diverge. A popular alternative is the information filtering scheme, which propagates the inverse covariance matrices, $P_{n|n}^{-1},\ P_{n|n-1}^{-1}$, usually via respective Cholesky factorization. In contrast, the above scheme is known as the covariance Kalman algorithm.

- **The Kalman Filtering Algorithm**
  - Input: $F_n,\ H_n,\ Q_n,\ R_n,\ \boldsymbol{y}_n,\ n = 1, 2, ...$
  - Initialization:
    - $\hat{\boldsymbol{x}}_{1|0} = \mathbb{E}[\mathbf{x}_1]$
    - $P_{1|0} = \Pi_0$
  - **For** $n = 1, 2, ...,$ **Do**
    - $S_n = R_n + H_n P_{n|n-1} H_n^T$
    - $K_n = P_{n|n-1} H_n^T S_n^{-1}$
    - $\hat{\boldsymbol{x}}_{n|n} = \hat{\boldsymbol{x}}_{n|n-1} + K_n(\boldsymbol{y}_n - H_n \hat{\boldsymbol{x}}_{n|n-1})$
    - $P_{n|n} = P_{n|n-1} - K_n H_n P_{n|n-1}$
    - $\hat{\boldsymbol{x}}_{n+1|n} = F_{n+1} \hat{\boldsymbol{x}}_{n|n}$
    - $P_{n+1|n} = F_{n+1} P_{n|n} F_{n+1}^T + Q_{n+1}$
  - **End For**
- For complex-valued variables, transposition is replaced by the Hermitian one.
- Observe that $P_{n|n}$ is computed as the difference of two positive definite matrices; this may lead to a non-positive definite, $P_{n|n}$, due to numerical errors. This can cause the algorithm to diverge. A popular alternative is the information filtering scheme, which propagates the inverse covariance matrices, $P_{n|n}^{-1},\ P_{n|n-1}^{-1}$, usually via respective Cholesky factorization. In contrast, the above scheme is known as the covariance Kalman algorithm.

- **Proof**: The proof involves four major steps.
  - Step 1: Using $\hat{\mathbf{x}}_{n-1|n-1}$, predict $\hat{\mathbf{x}}_{n|n-1}$ using the state equation; that is,
    $$\hat{\mathbf{x}}_{n|n-1} = F_n \hat{\mathbf{x}}_{n-1|n-1}.$$
    In other words, ignore the contribution from the noise. This is natural, since prediction cannot involve the unobserved variable.
  - Step 2: Obtain the respective error covariance matrix,
    $$P_{n|n-1} = \mathbb{E}[(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1})(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1})^T].$$
    However,
    $$\mathbf{e}_{n|n-1} := \mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1} = F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n - F_n \hat{\mathbf{x}}_{n-1|n-1} = F_n \mathbf{e}_{n-1|n-1} + \boldsymbol{\eta}_n.$$
    Combining the last two equations we get,
    $$P_{n|n-1} = F_n P_{n-1|n-1} F_n^T + Q_n.$$
  - Step 3: Update $\hat{\mathbf{x}}_{n|n-1}$. To this end, adopt the following recursion
    $$\hat{\mathbf{x}}_{n|n} = \hat{\mathbf{x}}_{n|n-1} + K_n \mathbf{e}_n, \tag{15}$$
    where
    $$\mathbf{e}_n := \mathbf{y}_n - H_n \hat{\mathbf{x}}_{n|n-1}. \tag{16}$$

- (proof continued: $\hat{\mathbf{x}}_{n|n} = \hat{\mathbf{x}}_{n|n-1} + K_n \mathbf{e}_n$),
  - (Step 3 Continued): Thus, the "new" (posterior) estimate is equal to the "old" (prior) one, based on **the past history** plus a correction term; the latter is proportional to the error $\mathbf{e}_n$ in predicting the newly arrived measurement and its prediction based on the "old" estimate. Matrix K_n, known as the Kalman gain, controls the amount of correction and its value is computed so that to minimize the mean square error, i.e.,

$$J(K_n) := \mathbb{E}[\mathbf{e}_{n|n}^T \mathbf{e}_{n|n}] = \text{trace}\{P_{n|n}\},$$

where

$$P_{n|n} = \mathbb{E}[\mathbf{e}_{n|n} \mathbf{e}_{n|n}^T], \tag{17}$$

and

$$\mathbf{e}_{n|n} := \mathbf{x}_n - \hat{\mathbf{x}}_{n|n}.$$

It can be shown that the optimal Kalman gain is equal to

$$K_n = P_{n|n-1} H_n^T S_n^{-1},$$

where

$$S_n = R_n + H_n P_{n|n-1} H_n^T.$$

- (proof continued: $\hat{\mathbf{x}}_{n|n} = \hat{\mathbf{x}}_{n|n-1} + K_n \mathbf{e}_n$),
  - (Step 3 Continued): Thus, the "new" (posterior) estimate is equal to the "old" (prior) one, based on **the past history** plus a correction term; the latter is proportional to the error $\mathbf{e}_n$ in predicting the newly arrived measurement and its prediction based on the "old" estimate. Matrix K_n, known as the Kalman gain, controls the amount of correction and its value is computed so that to minimize the mean square error, i.e.,

$$J(K_n) := \mathbb{E}[\mathbf{e}_{n|n}^T \mathbf{e}_{n|n}] = \text{trace}\{P_{n|n}\},$$

where

$$P_{n|n} = \mathbb{E}[\mathbf{e}_{n|n} \mathbf{e}_{n|n}^T], \tag{17}$$

and

$$\mathbf{e}_{n|n} := \mathbf{x}_n - \hat{\mathbf{x}}_{n|n}.$$

It can be shown that the optimal Kalman gain is equal to

$$K_n = P_{n|n-1} H_n^T S_n^{-1},$$

where

$$S_n = R_n + H_n P_{n|n-1} H_n^T.$$

- (proof continued)
  - Step 4: The final recursion that is now needed, in order to complete the scheme, is that for the update of $P_{n|n}$. Combining the definitions in (16) and (17) with (15), the following results,

$$P_{n|n} = P_{n|n-1} - K_n H_n P_{n|n-1}.$$

The algorithm has now been derived. All that is now needed is to select the initial conditions, which are chosen such as:

$$\begin{aligned} \hat{\boldsymbol{x}}_{1|0} &= \mathbb{E}[\mathbf{x}_1] \\ P_{1|0} &= \mathbb{E}\left[(\mathbf{x}_1 - \hat{\boldsymbol{x}}_{1|0})(\mathbf{x}_1 - \hat{\boldsymbol{x}}_{1|0})^T\right] = \Pi_0. \end{aligned}$$

- Extended Kalman Filters Kalman filtering, in a more general formulation, can be cast as

$$\begin{aligned} \mathbf{x}_n &= \boldsymbol{f}_n(\mathbf{x}_{n-1}) + \boldsymbol{\eta}_n \\ \mathbf{y}_n &= \boldsymbol{h}_n(\mathbf{x}_n) + \mathbf{v}_n \end{aligned}$$

- The vector-functions $\boldsymbol{f}_n$ and $\boldsymbol{h}_n$ are nonlinear. In the Extended Kalman Filtering (EKF), the idea is to linearize the functions $\boldsymbol{h}_n(\cdot)$ and $\boldsymbol{f}_n(\cdot)$, at each time instant, via their Taylor series expansions, keep the linear term only and then proceed with the linear Kalman filtering algorithm.

- (proof continued)
  - Step 4: The final recursion that is now needed, in order to complete the scheme, is that for the update of $P_{n|n}$. Combining the definitions in (16) and (17) with (15), the following results,

$$P_{n|n} = P_{n|n-1} - K_n H_n P_{n|n-1}.$$

The algorithm has now been derived. All that is now needed is to select the initial conditions, which are chosen such as:

$$\begin{aligned} \hat{\boldsymbol{x}}_{1|0} &= \mathbb{E}[\mathbf{x}_1] \\ P_{1|0} &= \mathbb{E}\left[(\mathbf{x}_1 - \hat{\boldsymbol{x}}_{1|0})(\mathbf{x}_1 - \hat{\boldsymbol{x}}_{1|0})^T\right] = \Pi_0. \end{aligned}$$

- Extended Kalman Filters Kalman filtering, in a more general formulation, can be cast as

$$\begin{aligned} \mathbf{x}_n &= \boldsymbol{f}_n(\mathbf{x}_{n-1}) + \boldsymbol{\eta}_n \\ \mathbf{y}_n &= \boldsymbol{h}_n(\mathbf{x}_n) + \mathbf{v}_n \end{aligned}$$

- The vector-functions $\boldsymbol{f}_n$ and $\boldsymbol{h}_n$ are nonlinear. In the Extended Kalman Filtering (EKF), the idea is to linearize the functions $\boldsymbol{h}_n(\cdot)$ and $\boldsymbol{f}_n(\cdot)$, at each time instant, via their Taylor series expansions, keep the linear term only and then proceed with the linear Kalman filtering algorithm.

Example: Autoregressive Process Estimation

- Let us consider an AR process of order l, i.e.,

$$\mathrm{x}_n = -\sum_{i=1}^{l} a_i \mathrm{x}_{n-i} + \eta_n, \tag{18}$$

where η_n is a white noise sequence of variance σ_η^2. Our task is to get an estimate $\hat{x}_n$ of x_n, having observed a noisy version of it, y_n. The corresponding random variables are related as,

$$\mathrm{y}_n = \mathrm{x}_n + \mathrm{v}_n. \tag{19}$$

- To this end, the Kalman filtering formulation will be used. Note that the MSE linear estimation, presented previously, cannot be used here. As it is discussed in Chapter 2, an AR process is asymptotically stationary; for finite time samples, the initial conditions at time $n = 0$ are "remembered" by the process and the respective (second) order statistics are time dependent, hence it is a non-stationary process. However, Kalman filtering is specially suited for such cases.

Example: Autoregressive Process Estimation

- Let us consider an AR process of order l, i.e.,

$$\mathrm{x}_n = -\sum_{i=1}^{l} a_i \mathrm{x}_{n-i} + \eta_n, \tag{18}$$

where η_n is a white noise sequence of variance σ_η^2. Our task is to get an estimate $\hat{x}_n$ of x_n, having observed a noisy version of it, y_n. The corresponding random variables are related as,

$$\mathrm{y}_n = \mathrm{x}_n + \mathrm{v}_n. \tag{19}$$

- To this end, the Kalman filtering formulation will be used. Note that the MSE linear estimation, presented previously, cannot be used here. As it is discussed in Chapter 2, an AR process is asymptotically stationary; for finite time samples, the initial conditions at time $n = 0$ are "remembered" by the process and the respective (second) order statistics are time dependent, hence it is a non-stationary process. However, Kalman filtering is specially suited for such cases.

Example: Autoregressive Process Estimation

- Let us rewrite (18) and (19) as

$$\begin{bmatrix} \mathrm{x}_n \\ \mathrm{x}_{n-1} \\ \mathrm{x}_{n-2} \\ \vdots \\ \mathrm{x}_{n-l+1} \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{l-1} & -a_l \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{x}_{n-1} \\ \mathrm{x}_{n-2} \\ \mathrm{x}_{n-3} \\ \vdots \\ \mathrm{x}_{n-l} \end{bmatrix} + \begin{bmatrix} \eta_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\mathrm{y}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \mathrm{x}_n \\ \vdots \\ \mathrm{x}_{n-l+1} \end{bmatrix} + \mathrm{v}_n$$

or

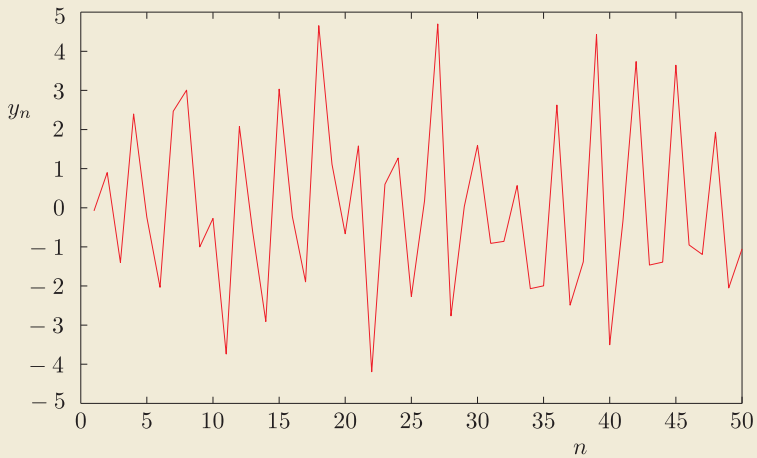
$$\begin{aligned} \mathbf{x}_n &= F\mathbf{x}_{n-1} + \boldsymbol{\eta} \\ \mathrm{y}_n &= H\mathbf{x}_n + \mathrm{v}_n \end{aligned}$$

where the definitions of $F_n \equiv F$ and $H_n \equiv H$ are obvious and

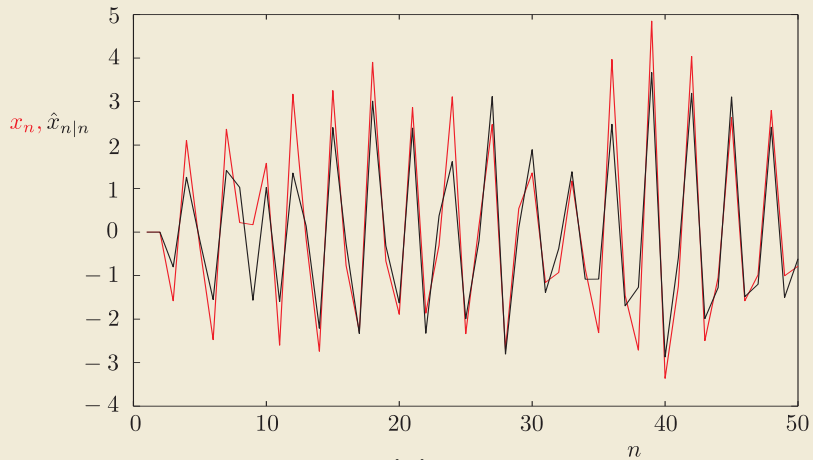
$$Q_n = \begin{bmatrix} \sigma_n^2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \; R_n = \sigma_{\mathrm{v}}^2 \;\; \text{(scalar)}.$$

Example: Autoregressive Process Estimation

- Figure (a) shows the values of a specific realization y_n, and Figure (b) the corresponding realization of the AR(2) (red) together with the predicted by the Kalman filter sequence $\hat{x}_n$. Observe that the match is very good. For the generation of AR process we used of $l = 2$, $\alpha_1 = 0.95$, $\alpha_2 = 0.9$, $\sigma_\eta^2 = 0.5$. For the Kalman filter output noise $\sigma_v^2 = 1$.



(a)



(b)

a) A realization of the observation sequence, y_n, which is used by the Kalman filter to obtain the predictions of the state variable. b) The AR process (state variable) in red together with the predicted by the Kalman filter sequence (gray). The Kalman filter has optimally removed the effect of the noise v_n and closely predicts the state variation.