2.3 Physical Meaning of Matrix $P$

The Recursive Least Squares (RLS) algorithm updates the parameter vector $\hat{\theta}(t-1)$ based on new data $\varphi^T(t), y(t)$ in such a way that the overall squared error may be minimal. This is done by multiplying the prediction error $\varphi^T(t)\hat{\theta}(t-1) - y(t)$ with the gain matrix which contains matrix $P_{t-1}$. To better understand the RLS algorithm, let us examine the physical meaning of matrix $P_{t-1}$.

Recall the definition of the matrix:

$$P_{t}^{-1} = \sum_{i=1}^{t} \varphi(i)\varphi^T(i) = \Phi \Phi^T$$

(17)

$$\Phi = \begin{bmatrix} \varphi(1) & . & \varphi(t) \end{bmatrix} \in \mathbb{R}^{m \times t}, \Phi^T \in \mathbb{R}^{t \times m}, \Phi \Phi^T \in \mathbb{R}^{m \times m}$$

Note that matrix $\Phi \Phi^T$ varies depending on how the set of vectors $\{\varphi(i)\}$ span the $m$-dimensional space. See the figure below.

![Geometric Interpretation of matrix $P_{t-1}$](image)

Geometric Interpretation of matrix $P_{t-1}$. 
Since $\Phi\Phi^T \in \mathbb{R}^{m \times m}$ is a symmetric matrix of real numbers, it has all \textit{real} eigenvalues. The eigen vectors associated with the individual eigenvalues are also real. Therefore, the matrix $\Phi\Phi^T$ can be reduced to a diagonal matrix using a coordinate transformation, i.e. using the eigen vectors as the bases.

$$
\Phi\Phi^T \Rightarrow D = \begin{pmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & \\
\vdots & \vdots & \ddots & \\
0 & \cdots & 0 & \lambda_m \\
\end{pmatrix} \in \mathbb{R}^{m \times m}
$$

(19)

$$
\lambda_{\max} = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m = \lambda_{\min}
$$

$$
P = (\Phi\Phi^T)^{-1} \Rightarrow D^{-1} = \begin{pmatrix}
1/\lambda_1 & 0 & \cdots & 0 \\
0 & 1/\lambda_2 & \cdots & \\
\vdots & \vdots & \ddots & \\
0 & \cdots & 0 & 1/\lambda_m \\
\end{pmatrix} \in \mathbb{R}^{m \times m}
$$

(20)

The direction of $\lambda_{\max}(\Phi\Phi^T)$ = The direction of $\lambda_{\min}(P)$.

If $\lambda_{\min} = 0$, then $\det(\Phi\Phi^T) = 0$, and the ellipsoid collapses. This implies that there is no input data $\phi(i)$ in the direction of $\lambda_{\min}$, i.e. the input data set does not contain any information in that direction. In consequence, the $m$-dimensional parameter vector $\theta$ cannot be fully determined by the data set.

In the direction of $\lambda_{\max}$, there are plenty of input data: $\phi(i)\cdots$. This direction has been well explored, well excited. Although new data are obtained, the correction to the parameter vector $\hat{\theta}(t-1)$ is small, if the new input data $\phi(t)$ is in the same direction as that of $\lambda_{\max}$. See the second figure above.

The above observations are summarized as follows:

1) Matrix $P$ determines the gain of the prediction error feedback

$$
\hat{\theta}(t) = \hat{\theta}(t-1) + K_r e(t)
$$

(17)

where $K_r$ is a varying gain matrix:

$$
K_r = \frac{P_{t-1}\phi(t)}{1 + \phi^T(t)P_{t-1}\phi(t)}
$$

2) If a new-data point $\phi(t)$ is aligned with the direction of $\lambda_{\max}(\Phi\Phi^T)$ or $\lambda_{\min}(P_{t-1})$,

$$
\phi^T(t)P_{t-1}\phi(t) \ll 1,
$$

then $K_r \approx P_{t-1}\phi(t)$ which is small. Therefore the correction is small.
3) Matrix $P_t$ represents how much data we already have in each direction in the $m$-dimensional space. The more we already know, the less the error correction gain $K_t$ becomes. Correction $\Delta \theta$ gets smaller and smaller as $t$ tends infinity.

### 2.4 Initial Conditions and Properties of RLS

**a) Initial conditions for $P_o$**

$P_o$ does not have to be accurate (close to its correct value), since it is recursively modified. But $P_o$ must be good enough to make the RLS algorithm executable. For this,

$P_o$ must be a **positive definite** matrix, such as the identity matrix $I$. (21)

Depending on initial values of $\hat{\theta}(0)$ and $P_o$, the (best) estimation thereafter will be different.

Question: How do the initial conditions influence the estimate? The following theorem shows exactly how the RLS algorithm works, given initial conditions.

**Theorem**

The Recursive Least Squares (RLS) algorithm minimizes the following cost function:

$$J_t(\theta) = \frac{1}{2} \sum_{i=1}^{t} (y(i) - \varphi^T(i)\theta)^2 + \frac{1}{2} (\theta - \hat{\theta}(0))^T P_0^{-1}(\theta - \hat{\theta}(0))$$

(22)

where $P_o$ is an arbitrary positive definite matrix ($m$ by $m$) and $\hat{\theta}(0) \in R^n$ is arbitrary.

**Proof** Differentiating $J_t(\theta)$

$$\frac{dJ_t(\theta)}{d\theta} = 0 \quad \rightarrow \quad -\sum_{i=1}^{t} (y(i) - \varphi^T(i)\theta)\varphi(i) + P_0^{-1}(\theta - \hat{\theta}(0)) = 0$$

(23)

Collecting terms

$$\left[ \sum_{i=1}^{t} \varphi(i)\varphi^T(i) + P_0^{-1} \right] \theta = \sum_{i=1}^{t} y(i)\varphi(i) + P_0^{-1}\hat{\theta}(0)$$

(24)

The parameter vector minimizing (22) is then given by

$$\hat{\theta}(t) = P_t \left[ \sum_{i=1}^{t} y(i)\varphi(i)\varphi^T(i) + P_0^{-1}\hat{\theta}(0) \right]$$

$$= P_t \left[ y(t)\varphi(t) + \sum_{i=1}^{t-1} y(i)\varphi(i)\varphi^T(i) + P_0^{-1}\hat{\theta}(0) \right]$$
\[
P_{t-1}^{-1}\hat{\theta}(t-1)
\]
Recall \( P_{t-1}^{-1} = \varphi(t)\varphi^T(t) + P_{t-1}^{-1} \)

\[
\hat{\theta}(t) = P_t\left[P_{t-1}^{-1}\hat{\theta}(t-1) + y(t)\varphi(t) - \varphi(t)\varphi^T(t)\hat{\theta}(t-1)\right]
\]
\[
= \hat{\theta}(t-1) + P_t\varphi(t)\left[y(t) - \varphi^T(t)\hat{\theta}(t-1)\right]
\]  
(25)

Postmultiplying \( \varphi(t) \) to both sides of (14)

\[
P_t\varphi(t) = P_{t-1}\varphi(t) - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}\varphi(t)}{(1 + \varphi^T(t)P_{t-1}\varphi(t))}
\]
\[
= \frac{P_{t-1}\varphi(t)}{(1 + \varphi^T(t)P_{t-1}\varphi(t))}
\]  
(26)

using (26) in (25) yields (18), the RLS algorithm,

\[
\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P_{t-1}\varphi(t)}{(1 + \varphi^T(t)P_{t-1}\varphi(t))}\left[y(t) - \varphi^T(t)\hat{\theta}(t-1)\right]
\]  
(18)

Q.E.D.

Discussion on the Theorem of RLS

\[
\hat{\theta}(t) = \operatorname{arg \, min}_{\theta} \left[ \frac{1}{2} \sum_{i=1}^{t} \left(y(i) - \varphi^T(i)\hat{\theta}\right)^2 + \frac{1}{2} \left(\theta - \hat{\theta}(0)\right)^T P_0^{-1} \left(\theta - \hat{\theta}(0)\right) \right]
\]

\[
\begin{bmatrix}
\text{Squared estimation error} \\
\text{Weighted squared distance from } \hat{\theta}(0)
\end{bmatrix}
\]

\( A \)

\( B \)

(27)

1) As \( t \) gets larger, more data are obtained and term \( A \) gets overwhelmingly larger than term \( B \). As a result, the influence of initial conditions fades out.

2) In an early stage, i.e. small time index \( t \), \( \theta \) is pulled towards \( \hat{\theta}(0) \), particularly when the eigenvalues of matrix \( P_0^{-1} \) are large.

3) In contrast, if the eigenvalues of \( P_0^{-1} \) are small, \( \theta \) tends to change more quickly in response to the prediction error, \( y(t) - \varphi^T(t)\hat{\theta} \).

4) The initial matrix \( P_0 \) represents the level of confidence for the initial parameter value \( \hat{\theta}(0) \).

Note: The \( P \) matrix involved in RLS with an initial condition \( P_o \) has been extended in the RLS theorem from the batch processing case of \( P_t^{-1} = \sum_{i=1}^{t} \varphi(i)\varphi^T(i) \) to:

\[
P_t^{-1} = \sum_{i=1}^{t} \varphi(i)\varphi^T(i) + P_o^{-1}
\]

(28)
Other important properties of RLS include:

- Convergence of $\theta(t)$. It can be shown that
  \[
  \lim_{t \to \infty} \left\| \hat{\theta}(t) - \hat{\theta}(t-1) \right\| = 0
  \]  
  (29)
  See Goodwin and Sin’s book, Ch.3, for proof.

- The change to the $P$ matrix: $\Delta P = P_t - P_{t-1}$ is negative semi-definite, i.e.
  \[
  \Delta P = - \frac{P_{t-1} \varphi(t) \varphi^T(t) P_{t-1}}{(1 + \varphi^T(t) P_{t-1} \varphi(t))} \leq 0
  \]  
  (30)
  for an arbitrary $\varphi(t) \in \mathbb{R}^m$ and positive definite $P_{t-1}$.

Exercise Prove this property.

### 2.5 Estimation of Time-varying Parameters
Least Squares with Exponential Data Weighting

Forgetting factor: \( \alpha \)

\[
0 < \alpha \leq 1
\]  
(31)

Large \( \alpha \), for slowly changing processes

Small \( \alpha \), for rapidly changing processes

Weighted Squared Error

\[
J_t(\theta) = \sum_{i=1}^{t} \alpha^{t-i} e^2(i)
\]  
(32)

\[
\hat{\theta}(t) = \arg \min_{\theta} J_t(\theta)
\]  
(33)

\( \hat{\theta}(t) \) is given by the following recursive algorithm.

\[
\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P_{t-1}\varphi(t)}{\alpha + \varphi^T(t)P_{t-1}\varphi(t)}(y(t) - \varphi^T(t)\hat{\theta}(t-1))
\]  
(34)

\[
P_t = \frac{1}{\alpha} \left[ P_{t-1} - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{\alpha + \varphi^T(t)P_{t-1}\varphi(t)} \right]
\]  
(35)

Exercise: Obtain (34) and (35) from (32) and (33).

A drawback of the forgetting factor approach

When the system under consideration enters “steady state”, the matrix \( P_{t-1}\varphi(t)\varphi^T(t)P_{t-1} \) tends to the null matrix. This implies

\[
P_t \approx \frac{1}{\alpha} P_{t-1}
\]  
(36)

As \( \alpha < 1 \), \( 1/\alpha \) makes \( P_t \) larger than \( P_{t-1} \). Therefore \( \{ P_t \} \) begins to increase exponentially. The “Blow-Up” problem

Remedy:

Covariance Re-setting Approach
• The forgetting factor approach has the “Blow-Up” problem
• The ordinary RLS
  The P matrix gets small after some iterations (typically 10-20 iterations). Then the
gain dramatically reduces, and \( \hat{\theta} \) is no longer varying.

The Covariance Re-Setting method is to solve these shortcomings by occasionally
re-setting the P matrix to:
\[
P_t = kI \quad 0 < k < \infty
\]  (37)
This re-vitalizes the algorithm.

### 2.6 Orthogonal Projection

The RLS algorithm provides an iterative procedure to converge to its final parameter
value. This may take more than \( m \) (dimension of \( \theta \)) steps.
The Orthogonal Projection algorithm provides the least squares solution exactly in \( m \)
recursive steps

Assume \( \Phi_m = \begin{bmatrix} \varphi(1) & \varphi(2) & \ldots & \varphi(m) \end{bmatrix} \)  (38)

Spanning the whole \( m \)-dim space

Set \( P_\theta=I \) (the \( mxm \) identity matrix) and \( \hat{\theta}(0) \) arbitrary

Compute
\[
\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P_{t-1}\varphi(t)}{\varphi^T(t)P_{t-1}\varphi(t)} \left( y(t) - \varphi^T(t)\hat{\theta}(t-1) \right)
\]  (39)

where matrix \( P_{t-1} \) is updated with the same recursive formula as RLS
Note that +1 involved in the denominator of RLS is eliminated in (39)
This causes a numerical problem when \( \varphi^T(t)P_{t-1}\varphi(t) \) is small, the gain is large.

2-parameter example

This orthogonal projection algorithm is more efficient, but is very sensitive to noisy data.
Ill-conditioned when \( \varphi^T(t)P_{t-1}\varphi(t) \approx 0 \). RLS is more robust.
2.6 Multi-Output, Weighted Least Squares Estimation

For each output $\hat{\theta}_i(t) = \varphi_i^T(t)\theta$

$$\hat{y}(t) = \begin{bmatrix} \varphi_1^T \\ \vdots \\ \varphi_l^T \end{bmatrix} \theta = \Psi^T(t)\theta \quad \Psi \in \mathbb{R}^{l \times m} \quad (40)$$

Error $\bar{e}(t) = \begin{bmatrix} e_1 \\ \vdots \\ e_l \end{bmatrix} = \bar{y}(t) - \Psi^T(t)\theta \quad (41)$

Consider that each squared error is weighted differently, or Weighted Multi-Output Squared Error:

$$J_i(\theta) = \sum_{i=1}^{l} \bar{e}^T(i) W(i) \bar{e}(i) = \sum_{i=1}^{l} (\bar{y}(i) - \Psi^T(t)\theta)^T W(i) (\bar{y}(i) - \Psi^T(t)\theta) \quad (42)$$

$$\hat{\theta}(t) = \arg \min_{\theta} J_i(\theta) \quad \Rightarrow \quad \hat{\theta}(t) = P_t B_t$$

$$P_t = \left[ \sum_{i=1}^{l} \Psi^T(i) W(i) \right]^{-1}$$

$$B_t = \sum_{i=1}^{l} \Psi^T(i) W(i) \bar{y}(i) \quad (43)$$

The recursive algorithm

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P_t \Psi(t) W(i) (\bar{y}(t) - \Psi^T(t)\hat{\theta}(t-1))$$

$$P_t = P_{t-1} - P_{t-1} \Psi(t) [W^{-1} + \Psi^T(t) P_{t-1} \Psi(t)]^{-1} \Psi^T(t) P_{t-1}$$