Decomposition Methods for Least-Squares Parameter Estimation

STEVE S. NIU, LENNART LJUNG and AKE BJORCK

First Draft: September 15, 1994
Last Modified: December 21, 1994, 4:34 P.M.

Abstract

In this paper, least-squares method with matrix decomposition is revisited and a multiple model formulation is proposed. The proposed formulation takes advantage of the well-established decomposition methods but possesses a multiple model structure which leads to simpler and more flexible implementations, and produces more information than the original least squares methods. Several application examples in signal processing and system identification are included. As a basic numerical analysis tool, the proposed methods can be used in many different application areas.

1 Introduction

Least-squares method can be traced back to almost two hundred years ago when Gauss used this simple method to calculate the orbits of the planets (Gauss 1809, Barnard 1963). The least squares method has ever since become one of the most widely used and most useful numerical tools in almost all fields of science and engineering due to its simplicity in concept and convenience in implementation.

Many of the real world problems would finally reduce to the problem of solving a set of overdetermined linear simultaneous equations of the form

\[
y_1 = X_1 \theta_1 \\
y_2 = X_2 \theta_2 \\
y_3 = X_3 \theta_3 \\
\vdots \\
y_m = X_m \theta_m
\]

or in a compact form

\[
y = X \theta
\]

where \( X \in \mathbb{R}^{m \times n} \) is the coefficient matrix and \( y \in \mathbb{R}^{m \times 1} \) is the observation vector. \( \theta \in \mathbb{R}^{n \times 1} \) is the vector of the unknowns to be solved. This is one of the most commonly
encountered least-squares problems, though the method of least-squares can apply to a much greater range of problems.

The least-squares method aims to find the solution $\hat{\theta} = [\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_n]$ that minimizes the sum of squares of the difference between the observed data (the observation vector) and their estimated values, which is

$$\hat{\theta} = \arg \min_\theta ||y - X\theta||_2$$  \hspace{1cm} (3)

where $\| \cdot \|_2$ denotes the Euclidian vector norm. According to Gauss’s Theorem of Least-Squares (Barnard 1963), the least-squares solution $\hat{\theta}$ to equation (2) is given by the following normal equation

$$[X^\top X] \theta = X^\top y$$  \hspace{1cm} (4)

Conceptually, the least-squares solution to (2) can be trivially obtained by solving the above normal equation, then the solution can be obtained by symmetric Gaussian elimination if $X^\top X$ is invertible

$$\hat{\theta} = [X^\top X]^{-1}X^\top y$$  \hspace{1cm} (5)

and the loss function (sum of squared residuals) is given by

$$J = [y - X\hat{\theta}]^\top [y - X\hat{\theta}]$$  \hspace{1cm} (6)

In practice, however, the least-squares problem is almost never solved in this way due to the poor numerical performance when the covariance matrix involved is ill-conditioned (Lawson & Hanson 1974, Golub & Loan 1989). Much effort has been devoted to the improvement of the numerical properties of the least-squares method, one of which matrix decomposition is one of the most successful endeavors. The Cholesky decomposition, QR decomposition and singular value decomposition (SVD) (Björck 1994, Golub & Loan 1989, Forsythe et al. 1977, Dahlquist & Björck 1974, Stewart 1973) are some of the commonly used decomposition techniques, among which the QR and SVD decompositions are the most celebrated. For instance, the data matrix $X$ in equation (2) can be decomposed with the QR decomposition as

$$X = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \quad \text{and} \quad Q^\top y = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$  \hspace{1cm} (7)

and the least-squares estimate is then given by solving

$$R \theta = z_1$$

and the loss function is given by

$$J = ||y - X\hat{\theta}||_2 = ||z_2||_2$$

Obviously, the decomposition methods are used solely as a numerical enhancement to the least-squares method. In this paper, however, the matrix decomposition techniques used for solving least-squares problems are investigated from a different standpoint. By working on the augmented least-squares problem, deeper insight into the least-squares principle is obtained which shows that the decomposition least-squares method can produce much more information than conventional least-squares, and also with a simpler and more compact implementation structure.
2 Multiple Model Decomposition Least-Squares

The least-squares problem defined in (1) or (2) can be represented in a slightly different form

$$[X, -y] \begin{bmatrix} \theta \\ 1 \end{bmatrix} = e$$

or

$$\bar{X} \bar{\theta} = e$$

This is called the augmented system. $\bar{X}$ is called the augmented data matrix and $\bar{\theta}$ is called the augmented solution vector. In this paper, we will be focusing on the augmented data matrix $\bar{X}$ and show that the least-squares problem defined in (2) actually contains much more information than just the solution vector $\theta$. By appropriate decomposition of the augmented data matrix, all this information can be extracted and made explicit in useful forms. Many of the well-established decomposition methods such as LU/LDL$^T$, Cholesky, QR, UDU$^T$ can all be used to extract the information. However, for simplicity of notation and convenience of presentation, we will in this section use the LDL$^T$ (symmetrical matrix) and LDU decomposition (non-symmetrical matrix) methods to illustrate the concept of this type of decomposition least-squares methods. For practical implementation, which is discussed in Section 2.4, the QR decomposition is recommended because of its superior numerical reliability.

2.1 Forward Decomposition Least-Squares

Rewrite the the augmented data matrix $\bar{X}$ into columns

$$\bar{X} = [x_1, x_2, \cdots, x_n, -y]$$

and consider the following $n$ sets of simultaneous equations

$$
\begin{align*}
0 \\
-12x_1 \\
-13x_1 + u_{23}x_2 \\
\vdots \\
-1nx_1 + u_{1n}x_2 + \cdots + u_{n-1,n}x_{n-1} = -x_n + e_n \\
\theta_1x_1 + \theta_2x_2 + \cdots + \theta_{n-1}x_{n-1} + \theta_nx_n = y + e_n
\end{align*}
$$

Obviously, the last equation is the original least-squares problem defined in (2) or (1). The other rows define some additional equations. The equations can be re-written into a matrix form as

$$
\begin{bmatrix}
x_1 & x_2 & x_3 & \cdots & x_n & -y
\end{bmatrix}
\begin{bmatrix}
1 & u_{12} & u_{13} & \cdots & u_{1n} & \theta_1 \\
1 & u_{23} & u_{24} & \cdots & u_{2n} & \theta_2 \\
1 & u_{34} & u_{35} & \cdots & u_{3n} & \theta_3 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
1 & \theta_n \\
1 & \theta_n
\end{bmatrix} = E
$$

or in the compact form

$$\bar{X} \bar{u} = E$$
The least-squares solutions for all the above \( n \) sets of linear equations can be produced, *simultaneously*, by a simple decomposition. This is stated as the following theorem.

**Theorem 1 (Forward Decomposition Least-Squares)**

Assuming that the augmented coefficient matrix \( \tilde{X} \) is defined in (9) and is non-singular, define following \( \text{LDL}^T \) decomposition

\[
S = \tilde{X}^T \tilde{X} = LDL^T
\]

(13)

where \( L \) is unit lower triangular and \( D \) is diagonal. Then \( L \) and \( D \) simultaneously provide all the least-squares solutions and loss functions for the \( n \) sets of linear equations defined by (10) or (11) in the form

\[
\begin{align*}
\mathcal{U} &= L^{-\tau} \quad \text{(unit upper triangular)} \\
\mathcal{D} &= D \quad \text{(diagonal)}
\end{align*}
\]

(14)

where "\( \tau \)" stands for matrix transpose and inverse, \( \mathcal{U} \) and \( \mathcal{D} \) are called the solution and loss function matrices respectively.

Obviously, the conventional least-squares (3) problem is only a subset of the augmented least-squares problem. Proof of this theorem can be found in Appendix A.

The solution matrix \( \mathcal{U} \) in (14) is analog to the solution vector (5) in conventional least-squares, and contains all the least-squares solutions to all the equations in (10). The loss function matrix \( \mathcal{D} \) is in analog to the loss function (scalar) in (6). It contains the loss functions (sum of the squared residuals) for the least-squares solutions to all the equations in (10). The \( n \) additional sets of linear simultaneous equations are produced *without any additional computational cost*. Although here the extra \( n \) sets of equations do not seem to have very specific practical significance, in parameter estimation, however, with an appropriate formulation of the data vector, they constitute all the lower order models. This is discussed in details in Section 3.

**Example 1 (Forward Decomposition Least-Squares)**

Consider the following linear simultaneous equations

\[
X \theta = y
\]

(15)

with

\[
X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix} = [x_1, x_2, x_3], \quad y = \begin{bmatrix} 10 \\ 28 \\ 37 \end{bmatrix}
\]

where \( \text{rank}(X) = 3 \) and \( \text{cond}(X) = 35 \). This is a well-conditioned matrix. Apply the standard \( \text{LDL}^T \) decomposition to the augmented data matrix

\[
S = [X, -y]^T [X, -y] = LDL^T
\]

then the solution and loss function matrices are given by

\[
\begin{align*}
\mathcal{U} &= L^{-\tau} = \begin{bmatrix} 1 & -1.1818 & 5.5000 & 3.0000 \\ 1 & -5.0000 & 2.0000 \\ 1 & 1.0000 \\ 1 & 1 \end{bmatrix} \\
\mathcal{D} &= D = \text{diag}(66.000, 0.8182, 13.500, 0.0000)
\end{align*}
\]
which can be written into 3 sets of equations as
\[
\begin{bmatrix}
-1.1818x_1 \\
+5.5000x_1 - 5.0000x_2 \\
+3.0000x_1 + 2.0000x_2 + 1.0000x_3
\end{bmatrix}
= \begin{bmatrix}
x_2 \\
x_3 \\
y
\end{bmatrix}
= \begin{bmatrix}
0.8182 \\
13.5000 \\
0.0000
\end{bmatrix}
\] (16)

The numbers in the brackets are the corresponding loss functions. The last set of equations is obviously the original linear equations that is defined by (15). The least-squares solution provided in the last column of \( \mathcal{U} \), namely \( \hat{\theta} = [3.0000, 2.0000, 1.0000]^T \), is the best linear least-squares fit of \( y \) by \( x_1, x_2, x_3 \). The least-squares fitting error (the loss function) is given by the last element of \( \mathcal{D} \) as zero since the solution \( \hat{\theta} \) agrees to the true solution \([3, 2, 1]^T\).

Note that the third column of matrix \( \mathcal{U} \) gives the best linear least-squares fit of \( x_3 \) with \( x_1 \) and \( x_2 \), which is the second set of equations in (16). The loss function is given by the third element in \( \mathcal{D} \). Similarly, the second column of \( \mathcal{U} \) gives the fit of \( x_2 \) with \( x_1 \) and the corresponding loss function is given by the second element in \( \mathcal{D} \).

Conventional methods can work equally well. However, with approximately the same amount of computational effort, the conventional implementation of the least-squares method does not produce any information other than the least-squares solution \( \hat{\theta} \).

2.2 Backward Decomposition Least-Squares Method

If the the augmented data matrix \( \mathbf{X} \) is formulated as follows
\[
\mathbf{X} = [-y, x_1, x_2, \ldots, x_n]
\]
and consider the following \( n \) sets of equations
\[
\begin{align*}
y + e_0 & = 0 \\
y + e_1 & = u_{12} x_1 \\
y + e_2 & = u_{13} x_1 + u_{23} x_2 \\
& \vdots \\
y + e_{n-1} & = u_{1n} x_1 + u_{2n} x_2 + \cdots + u_{n-1,n} x_{n-1} \\
y + e_n & = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_{n-1} x_{n-1} + \theta_n x_n
\end{align*}
\] (17)

that is, each set of equations uses the \( x \) vectors to fit the observation vector \( y \), which obviously makes more sense than those in (10). The different set of equations differ in the number of \( x \) vectors (ranging from 1 to \( n \)) used for fitting \( y \). Again, the last equation is the original least-squares problem defined in (2) or (1).

Rewrite Equation (17) into a matrix form as
\[
\begin{bmatrix}
-y & x_1 & x_2 & \cdots & x_{n-1} & x_n
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 1 & \cdots & 1 & 1 \\
u_{12} & u_{13} & \cdots & u_{1n} & \theta_1 \\
u_{23} & \cdots & u_{2n} & \theta_2 \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
u_{n-1,n} & \theta_{n-1} & \cdots & \cdots & \theta_n
\end{bmatrix}
= \mathbf{E}
\] (18)

where \( \mathbf{E} \) is the error matrix of appropriate dimension. In a more compact form
\[
\mathbf{X} \ U = \mathbf{E}
\] (19)

The solution matrix \( \mathcal{U} \) is then given by the following theorem.
Theorem 2 (Backward Decomposition Least-Squares)

For the least-squares problem defined in (2), define the (non-symmetric) augmented data product matrix $S$ as

$$S = [-y, X]^T [X, -y]$$

and decompose $S$ with LDU decomposition into $S = LDU$, the least-squares solution matrix $U$ and the corresponding loss function matrix $D$ for the $n$ sets of linear equations in (17) or (19) are then given by

$$\begin{align*}
\{ U &= \text{unilead}(L^{-1}) \\
D &= \text{diag}(E^T E)
\end{align*}$$

(20)

where the residual matrix $E$ is calculated by (19).

Here ‘unilead($A$)’ means that all the leading element of each column is unity. This can be readily achieved by dividing every column of the matrix $A$ by the first element of that column. A better implementation is provided in Section 2.4. Proof of this theorem is in Appendix B.

Example 2 (Backward Decomposition Least-Squares)

Consider the same example as Example 1. Apply the LDU decomposition to the augmented data matrix

$$S = [-y, X]^T [X, -y] = LDU$$

then the solution and loss function matrices $U$ and $D$ are given by

$$U = \begin{bmatrix}
1 & 1 & 1 & 1 \\
5.7727 & -2.5000 & 3.0000 \\
7.0000 & 2.0000 \\
1.0000
\end{bmatrix}
$$

$$D = \text{diag}(2253.0, 53.591, 13.500, 0.0000)$$

which provides the least-squares solutions as the following three sets of equations

$$\begin{align*}
y &= +5.7727 x_1 \\
y &= -2.5000 x_1 + 7.0000 x_2 \\
y &= +3.0000 x_1 + 2.0000 x_2 + 1.0000 x_3
\end{align*}$$

(21)

That is, the last column of $U$ gives the least-squares solution to the original equations, which is the third set of equations in (21). The third column of matrix $U$ gives the observation vector $y$ as the best linear combination, in least-squares sense, of the two vectors $x_1$ and $x_2$, which is the second set of equations in (21). The second column of $U$ gives the least-squares fit of $y$ with a single vector $x_1$. The corresponding loss functions for the above three equations are given by the 2nd, 3rd and 4th elements in matrix $D$ respectively.

A comparison of Example 2 with Example 1 reveals that the results here are more natural and meaningful.

2.3 Generalization: Multiple Model Least-Squares

With the augmented equations, the least-squares problem in (3) can be redefined as follows

$$\hat{\theta} = \arg \min_{\theta} \| \hat{X} \hat{\theta} \|_2 \quad \text{subject to} \quad \hat{\theta}_{n+1} \equiv 1$$

(22)
This is a more general form for least-squares problems. The least-squares method with QR decomposition then becomes

\[ \hat{X} = QR = QDU \]  

(23)

where \( D \) and \( U \) is the further decomposition of the triangular matrix \( R \) and has a diagonal and unit upper triangular form respectively. The least-squares solution can then be directly read off from the last column of \( U^{-1} \) and the corresponding loss function is read off from the last diagonal element of \( D \). In addition, matrix \( Q \) is not explicitly needed so both computation and storage can be saved or reduced. For example, in the QR decomposition,

\[ \hat{X} = [Q_1, Q_2]\begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} = Q_1 R \]

then only the \( Q_1 \) part of the orthogonal matrix \( Q \) needs to be calculated.

As a generalization, the following three least-squares problems can be defined with the same cost function (22). That is,

1. **Forward Prediction Least-Squares.**
   \[ \hat{\theta} = \arg\min_{\theta} \|\hat{X}\theta\|_2 \quad \text{subject to} \quad \theta_{n+1} \equiv 1 \]  
   (24)

2. **Backward Prediction Least-Squares.**
   \[ \hat{\theta} = \arg\min_{\theta} \|\hat{X}\theta\|_2 \quad \text{subject to} \quad \theta_1 \equiv 1 \]  
   (25)

3. **Total Least-Squares**
   \[ \hat{\theta} = \arg\min_{\theta} \|\hat{X}\theta\|_2 \quad \text{subject to} \quad \|\theta\|_2 \equiv 1 \]  
   (26)

In fact, we can go one step further, that is, to extend the conventional single model least-squares method as defined in (22) into a multiple model form, without increasing the computational effort.

Consider the following extension of the least-squares problem (2)

\[ \hat{X} \Theta = E \]  

(27)

where \( \hat{X} \in \mathbb{R}^{m \times (n+1)} \) is defined in (8) and \( \Theta \) is a square solution matrix of dimension \( (n + 1) \times (n + 1) \). Compared with the conventional least-squares method defined in (8) which solves for one set of equations, equation (27) defines \( n+1 \) sets of equations and is referred to as multiple models. The structure of the conventional least-squares and multiple model least-squares can be depicted as in Figure 1, where it is clear that conventional least-squares can be treated as a subset of the multiple model least-squares.

The multiple model least-squares is defined as follows

**Definition 1 (Multiple Model Least-Squares)**

For the multiple equations defined in (27), the least-squares estimate of the solution matrix is defined as

\[ \hat{U} = \arg\min_{\Theta} \|\hat{X}\Theta\|_2, \]  

(28)

where \( \hat{X} \in \mathbb{R}^{m \times n} \) is the augmented data matrix and can take any form. \( \Theta \) is the solution matrix with a user specified special structure.
Orthogonal transformations are the natural choices for the multiple model least-squares problem. Here is this paper, we consider the QR and singular value decomposition methods. For the QR decomposition shown in (23), the MMLS criterion (28) becomes

$$U = \arg \min_{\Theta} ||\tilde{X}\Theta||_2 = \arg \min_{\Theta} ||R\Theta||_2$$

which indicates that the $R$ matrix contains the same amount of information as $\tilde{X}$ in terms of meeting the MMLS criterion (28). The following are the two special structures for $\Theta$ that are discussed in this Sections 2.1 and 2.2:

1. $\Theta$ is restricted to an upper triangular matrix with all its diagonal elements being unity

$$U = \begin{bmatrix}
1 & u_{12} & u_{13} & \cdots & u_{1n} & \theta_1 \\
1 & u_{23} & \cdots & u_{2n} & \theta_2 \\
1 & \cdots & u_{3n} & \theta_3 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \cdots & 1 & \theta_n \\
1 & \cdots & 1 & 1
\end{bmatrix}$$

(29)

This is the structure that is discussed in Section 2.1 and will be referred to as the *forward MMLS* method.

2. $\Theta$ is assumed to be an upper triangular matrix with its first row being unity

$$U = \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 & 1 & 1 \\
1 & u_{12} & u_{13} & \cdots & u_{1n} & \theta_1 \\
1 & u_{23} & \cdots & u_{2n} & \theta_2 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
1 & \cdots & 1 & \theta_{n-1} & \theta_n \\
1 & \cdots & 1 & 1 & 1 & 1
\end{bmatrix}$$

(30)

This is the structure that is discussed in Section 2.2 and will be referred to as the *backward MMLS* method.
Consequently, with QR decomposition, the multiple model least-squares problem becomes that of extracting the useful information from $R$ in an efficient and explicit form such as (29) and (30). The leftover of the $R$ matrix after the extraction, which is a diagonal matrix, is the corresponding loss function matrix.

Obviously, an appropriate combination of the data matrix structure and the decomposition method can lead to several useful structures. For example

1. If the augmented data matrix is arranged in the form of

$$\tilde{X} = [x_1, x_2, \ldots, x_n, -y]$$

the applying the forward and backward decompositions (Theorem 1 and Theorem 2) respectively leads to the results shown in Figures 2a and 2b, where each arrow represents a set of equations. For instance, the bottom arrow in Figures 2a implies fitting $y$ with $x_1, x_2, \ldots, x_n$.

2. If the order of the columns in the data matrix is reversed as

$$\tilde{X} = [-y, x_n, \ldots, x_2, x_1]$$

then applying Theorem 1 and Theorem 2 produces the solutions and loss functions for the equations represented by Figures 2c and 2d.

![Figure 2: Different Model Structures](image)

Another obvious choice of orthogonal decomposition is the singular value decomposition (Golub & Loan 1989), where the augmented data matrix $\tilde{X}$ is decomposed into

$$\tilde{X} = USV^\tau$$

(31)
\( U \) and \( V \) are two orthogonal matrices and \( S \) is a diagonal matrix that contains all the singular values of matrix \( \tilde{X} \) as its diagonal elements. If the parameter matrix \( \Theta \) in (27) is assumed to be orthogonal, then the best parameter matrix \( \Upsilon \) for \( \Theta \) is given by

\[
\Upsilon = V^{-\top} = V
\]

(32)

From Huffel & Vandewalle (1991) it is known that the last column of \( \Upsilon \) gives the Total Least-Squares solution to (27). In parameter estimation, total least-squares problem is also called the Error in Variable problem. In least-squares method the variable to be fitted (corresponding to the 1 in the augmented parameter vector) is assumed to be noise-corrupted and other variables are noise free. The total least-squares method, on the other hand, assumes that all the variables are contaminated by noise. Details on total least-squares can be found in Huffel & Vandewalle (1991) and Golub & Loan (1989).

Notice that there is not any restrictions posed on the structure of the data matrix. In another words, given an arbitrary matrix \( \tilde{X} \), the forward and backward MMLS provides the information on various forms of linear dependence among its columns as shown in Figure 2. The linear dependence is uniquely and conveniently determined and interpreted by equation (27).

### 2.4 Implementation

In previous sections, the LDU/LDL\(^T\) decomposition methods are used to explain the principle of the multiple model least-squares method just for ease of presentation. However, forming the data product matrix may lose precision due to finite word length and roundoff errors. As a result, the QR decomposition, which works directly on the data matrix, is always recommended for practical application. This section presents the implementation of the MMLS with QR decomposition.

First of all, the QR decomposition is related to the LU/LDL\(^T\) and Cholesky decompositions as follows. Given the symmetrical matrix \( S = \tilde{X}^\top \tilde{X} \), the following decompositions

- LDL\(^T\) Decomposition: \( S = LDL^\top \)
- LU Decomposition: \( S = LU \)
- Cholesky Decomposition: \( S = GG^\top \)
- QR Decomposition: \( \tilde{X} = QR \)

are related to each other by

\[
\begin{align*}
U & = DL^\top, & G & = LD^{1/2}, & R & = G^\top.
\end{align*}
\]

Theoretically, all these decompositions can produce exactly the same results. Choosing QR as our decomposition method is solely for its superior numerical properties over others.

The general procedure for MMLS with QR is thus as follows. First do a standard QR decomposition (Björcck 1994, for instance) to the augmented data matrix. Then extract the parameter and loss function matrix according to the special structure of the parameter matrix. For the forward MMLS, the extraction is trivially done by

\[
\mathcal{D} = \text{diag}(R)^2, \quad \Upsilon = R^{-1} \text{diag}(R)
\]

For the backward MMLS, the stepwise procedures shown in Table 1 can be used.
\begin{align*}
d(n) &= R(n, n)^2 \\
z &= R(1 : n - 1, n) \\
&\text{for } k = n - 1 \text{ down to } 1 \\
d(k) &= d(k + 1) + z^2(k) \\
y &= z(1 : k) \\
y(k) &= z(k)/R(k, k) \\
&\text{for } i = k - 1 \text{ down to } 1 \\
y(i) &= (z(i) - R(i, i + 1 : k)y(i + 1 : k))/R(i, i) \\
U(1 : k, k) &= y
\end{align*}

Table 1: Backward MMLS Procedure

Up to now, full rank of the coefficient matrix $X$ has been assumed. In practice, however, it is often required to handle rank-deficient least-squares problems, in which the equations in (1) are correlated with each other. In this case, the decomposition methods discussed above will run into numerical problems and special caution should be taken. However, with a small modification, the above decomposition methods can always give a solution. This is stated as follows

**Property 1** Assuming that the augmented data matrix $\tilde{X}$ is singular with a mathematical rank $\text{rank}(\tilde{X}) = n_1 < n$, if the $\tilde{R}$ matrix from the QR decomposition $\tilde{X}$ is given by

$$
R = \begin{bmatrix}
R_{11} & R_{12} \\
0 & 0
\end{bmatrix}, \quad \text{with } R_{11, n_1 \times n_1 \text{ non-singular}}
$$

then

1. matrix $R$ can be uniquely decomposed into the partitioned form

$$
R = \begin{bmatrix}
R_{11} & R_{12} \\
0 & 0
\end{bmatrix} = \begin{bmatrix}
D_{11} & 0 \\
0 & I
\end{bmatrix} \begin{bmatrix}
U_{11} \\
0
\end{bmatrix}
$$

2. the principal submatrix $R_{11}$ satisfies

$$
R_{11} = D_{11}U_{11}
$$

here $R_{11} \in \mathbb{R}^{n_1 \times n_1}$, $R_{12} \in \mathbb{R}^{n_1 \times n_2}$ and $n_1 + n_2 = n$. $U_{11}$ is unit-upper triangular, $D_{11}$ is diagonal, all with appropriate dimensions determined by the partitioning of $R$.

The following remarks can be made regarding this property

1. If $R$ is non-singular, then the above decomposition is the same as the normal QR decomposition. The interpretation of the decomposition result follows Theorems 1 and 2.

2. For singular $\tilde{X}$ matrix, the solution to the original equations (2) is not unique. However, the last column of $U$ gives one of the many possible solutions, which is one of those so-called the basic solution (Golub & Loan 1989) that has the minimum number of non-zeros.
3. The $R_{11}$ matrix represents the non-singular part of the full matrix $R$. For many applications such as parameter estimation, this is the only part that is of interest to us and $R_{12}$ can be simply ignored. In addition, since $R_{11} = D_{11} U_{11}$, it is not difficult to prove that $U_{11} = U_{11}^{-1}$ gives all the solutions of the non-singular equations (i.e., the first $n_1$ sets of equations) and $D = D_{11}$ gives all the corresponding loss functions. In terms of implementation, since the decomposition of the $R$ matrix starts from the top left corner, the decomposition can just stop when the first zero diagonal element of $D$ is encountered. This provides a simple and practical method for handling singularity of the linear equations.

4. In many of the signal processing applications (e.g., parameter estimation), the first zero diagonal element in $D$, i.e., $d_{n_1+1}$, indicates the starting of rank deficiency. Its index $n_1$ gives the rank of the coefficient matrix $R$ as rank($R$) = $n_1$.

5. The diagonal element of $D$ being zero only indicates that the corresponding least-squares solution in $U$ is exact, but does not guarantee that it is unique.

### 3 Application in Parameter Estimation

The examples in Sections 2.1 and 2.2 shows how the decomposition methods can be used for solving linear simultaneous equations. As basic numerical tools, however, the decomposition least-squares methods proposed in this paper can be used in many different application areas. This section gives some application examples of the decomposition methods to signal processing, particularly parameter estimation. For a general background on system identification, see Ljung (1987) or Söderström & Stoica (1989). For an introduction on signal processing see Haykin (1991).

Assume that a set of input/output data are collected from the system under investigation as

\[
\cdots, \{u(1), z(1)\}, \{u(2), z(2)\}, \cdots, \{u(N), z(N)\}
\]

where $u(\cdot)$ and $z(\cdot)$ are the system input and output respectively, $N$ is the total number of the input/output data pairs, or the data length. The following sections provide some examples of the application of the MMLS method.

#### 3.1 Forward and Backward Prediction

Model based prediction is widely used in many fields such as signal processing (Haykin 1991) and control (Åström & Wittenmark 1989). Here some formulations and interpretations based on the multiple model least-squares are presented with the main focus on the multiple model structures of the prediction models.

Assuming that the data vector is constructed as

\[
\varphi(t) = [-z(t-n), u(t-n), \cdots, -z(t-1), u(t-1), -z(t)]^T
\]

The corresponding data matrix is, obviously,

\[
\Phi(N) = \begin{bmatrix} -z(N-n), u(N-n), \cdots, -z(N-1), u(N-1), -z(N) \end{bmatrix}
\]

where $z(N-i), i = 0, 1, \cdots, n$ is a stack of output $z$ from $z(n-i+1)$ up to $z(N-i)$, $u(N-i)$ is a column of input $u$ from $u(n-i+1)$ to $u(N-i)$. Then we have the following two situations
1. **Forward Prediction Models.** If the forward multiple model least-squares (Theorem 1) is used, the resulting parameter matrix $\mathbf{U}$ and loss function matrix $\mathbf{D}$ give the parameter estimates and corresponding loss functions for all the following prediction models

$$
\begin{align*}
&z(t-n) \Rightarrow u(t-n) \\
z(t-n), u(t-n) \Rightarrow z(t-n+1) \\
&\vdots \\
z(t-n), u(t-n), \ldots, z(t-1) \Rightarrow u(t-1) \\
z(t-n), u(t-n), \ldots, z(t-1), u(t-1) \Rightarrow z(t)
\end{align*}
$$

where ‘$\Rightarrow$’ means to use the variables on the left hand side to predict/fit the right hand side. Parameter estimation of this type of models are thoroughly discussed in, e.g., Niu & Fisher (1994a), Niu et al. (1992) or Niu & Fisher (1994b). Note that this is the case with Figure 2a.

2. **Backward Prediction Models.** If the backward multiple model least-squares (Theorem 2) is used, then the resulting parameter and loss function matrices provide all the parameter estimates and corresponding loss functions for the following models

$$
\begin{align*}
&z(t-n) \Leftarrow u(t-n) \\
z(t-n) \Leftarrow u(t-n), z(t-n+1) \\
&\vdots \\
z(t-n) \Leftarrow u(t-n), z(t-n+1), \ldots, u(t-1) \\
z(t-n) \Leftarrow u(t-n), z(t-n+1), \ldots, u(t-1), z(t)
\end{align*}
$$

These are the so-called backward prediction models and are used in signal processing, see Haykin (1991). This corresponds to Figure 2b.

Now if the order of the elements in the data vector $\varphi(t)$ are reversed, that is

$$
\varphi(t) = [-z(t), u(t-1), -z(t-1), \ldots, u(t-n), -z(t-n)]^\top
$$

and the corresponding data matrix is given by

$$
\Phi(N) = [-z(N), u(t-1), -z(t-1), \ldots, u(t-n), -z(t-n)]
$$

then the following results follow

1. **Backward Prediction Models.** Now the forward multiple model least-squares method applying to the data matrix provides the parameter matrix $\mathbf{U}$ and loss function matrix $\mathbf{D}$ for the following back prediction models

$$
\begin{align*}
&z(t) \Rightarrow u(t-1) \\
z(t), u(t-1) \Rightarrow z(t-1) \\
&\vdots \\
z(t), u(t-1), \ldots, z(t-n+1) \Rightarrow u(t-n) \\
z(t), u(t-1), \ldots, z(t-n+1), u(t-n) \Rightarrow z(t-n)
\end{align*}
$$

This structure corresponds to Figure 2c.
2. **Forward Prediction Models.** If the backward decomposition least-squares (Theorem 2) is used, then the resulting parameter and loss function matrices provide all the parameter estimates and corresponding loss functions for the forward prediction models of the form

\[
\begin{align*}
    z(t) &\triangleq u(t-1) \\
    z(t) &\triangleq u(t-1), z(t-1) \\
    &\vdots \\
    z(t) &\triangleq u(t-1), z(t-1), \ldots, u(t-n) \\
    z(t) &\triangleq u(t-1), z(t-1), \ldots, u(t-n), z(t-n)
\end{align*}
\]

This is the case shown in Figure 2d.

Keeping in mind the multiple model structure as our goal, by choosing the right combination of the decomposition method and data vector structure, we can basically meet almost all our need in signal processing and system identification. The FIR coefficient identification and time delay estimation discussed in the next sections can be treated as special cases of the above combinations.

### 3.2 Estimation of FIR Parameters

Although identifying FIR coefficients can be treated as a special case of the general parameter estimation problem, however, due to its special importance and widespread use, it deserves some special discussion.

If the data vector \( \varphi(t) \) is defined as follows

\[
\varphi(t) = [-z(t), u(t-1), u(t-2), \ldots, u(t-n+1), u(t-n)]^T
\]

the corresponding data matrix is then given by

\[
\Phi(N) = [-z(N), u(N-1), u(N-2), \ldots, u(N-n+1), z(N-n)]
\]

Then the backward multiple model least-squares provides, in the matrices \( \mathbb{U} \) and \( \mathbb{D} \) respectively, all the parameter estimates and loss functions for the following FIR models

\[
\begin{align*}
    z(t) &\triangleq u(t-1) \\
    z(t) &\triangleq u(t-1), u(t-2) \\
    &\vdots \\
    z(t) &\triangleq u(t-1), u(t-2), \ldots, u(t-n+1) \\
    z(t) &\triangleq u(t-1), u(t-2), \ldots, u(t-n+1), u(t-n)
\end{align*}
\]

The multiple model structure can be very useful for certain applications. For instance, the Laguerre model identification (Dumont et al. 1991, Wahlberg & Hannan 1990) is covered by this structure. By pretreating the system inputs with a series of filters into some orthonormal variables and taking these variables as the inputs to the FIR identification scheme, the Laguerre model structure substantially reduces the number of FIR coefficients needed for describing a system. However, an important question remains on how many of these orthonormal variable should be used. With the above decomposition method to identify the Laguerre FIR model, not only are the parameter estimates of all orders simultaneously provided, the determining of the number of orthonormal variables (or the number of input filters) also becomes very straightforward and convenient.
3.3 Identification of Time Delay

By selecting a special structure for the data vector, the system time delay can also be easily determined. The basic strategy is still an exhaustive search among all the possible time delay, however, the special structure of the decomposition least-squares method makes the search much more efficient.

The main idea is to use a second (or first) model to try out all the possible time delay in the range specified by the user. The time delay that leads to the smallest loss function is then the estimated time delay.

Assuming that the time delay is in the range \([0, n_k]\), define the data vector as

\[
\varphi(t) = [-z(t), -z(t - 1), -z(t - 2), \\
u(t - n_k - 2), u(t - n_k - 1), \ldots, u(t - 2), u(t - 1)]^T
\]

and decomposing the resulting data product matrix with Theorem 2, it is found that the loss functions, provided in \(D\) decreases as order increases, and stays flat after the order is higher that a certain value. A closer look into the structure of the data vector readily reveals that the number of the trailing constant loss functions in the loss function matrix \(D\) actually equals the number of time delays for this system.

4 Conclusions

This paper presents a multiple model formulation for least-squares problems. Matrix decomposition methods such as LU, LDL\(^T\), Cholesky and QR decompositions can all be used for solving multiple model least-squares problems though the preferred/recommended is the QR decomposition method. The new formulations provides more flexibility, produces more information and leads to more convenient implementation and interpretation.

As basic numerical tools, the proposed methods can be used in many different fields. Among them, solving linear equations and parameter estimation are presented by examples.

References


### A Proof of the Forward Decomposition

Assume that the data product matrix is given by $S = \bar{X}^T \bar{X}$ in Lemma 1. Since it is symmetrical, it can be decomposed into LDLᵀ decomposed form as

$$S = LDLᵀ$$  \hspace{1cm} (34)

Assuming that the parameter and loss function matrices are given by

$$\mathbf{U} = L⁻ᵀ, \quad \mathbf{D} = \text{diag}(\mathbf{U})$$

respectively, then the error matrix for $\mathbf{U}$ is given by

$$\mathbf{E} = \bar{X} \mathbf{U}$$

and the loss function matrix $\mathbf{D}$ can be rewritten as

$$\mathbf{D} = \mathbf{E}^T \mathbf{E} = \mathbf{U}^T \bar{X}^T \bar{X} \mathbf{U}$$

16
Now assume that there exists another parameter matrix $V$, with exactly the same unit upper triangular structure as $U$, but gives a smaller loss function matrix, that is

$$Assumption: \quad U^\top \tilde{X}^\top \tilde{X} U - V^\top \tilde{X}^\top \tilde{X} V > 0 \quad (\text{positive definite})$$ (35)

From (34), bearing in mind that $D = D$, we have

$$D - V^\top L D L^\top V > 0$$ (36)

Define $G = V^\top L$, then clearly $G$ is a unit lower triangular matrix, with all its diagonal elements $g_{ii} = 1, i \in [1, m]$. With this in mind, the diagonal elements of the left hand side of (36) becomes

$$(D - G^\top G)_{ij}$$

$$= d_i - \sum_{j=1}^{i} g_{ij} d_j g_{ij}$$

$$= d_i - d_i \sum_{j=1}^{i} g_{ij}^2$$

$$= d_i - d_i \left[ \sum_{j=1}^{i-1} g_{ij}^2 + g_{ii}^2 \right]$$

$$= d_i - d_i \left[ \sum_{j=1}^{i-1} g_{ij}^2 + 1 \right]$$

$$= -d_i \sum_{j=1}^{i-1} g_{ij}^2$$

The data product matrix $S$ is assumed to be positive definite, which implies $d_i > 0$ for $i \in [1, m]$, therefore, to satisfy our assumption (equation 35), we can only have

$$g_{ij} = 0, \quad j = 1, \ldots, i - 1, \quad i = 1, \ldots, m$$

which implies that $G$ is an identity matrix

$$G = V^\top L = I$$

or equivalently

$$V = L^{-\tau} = U$$

That is, $U$ is the parameter matrix that leads to the smallest least squares errors, therefore $U$ is the least-squares parameter matrix, and $D$ is the corresponding loss function matrix.

### B Proof of the Backward Decomposition

First of all, we define some more notations which will be used solely for this section. From (18)

$$\tilde{X} = \begin{bmatrix} -y, x_1, x_2, \ldots, x_{n-1}, x_n \end{bmatrix}$$
\[
\begin{align*}
\mathbf{y} & = [-y, \mathbf{H}, x_n] \\
& = [-y, \mathbf{X}] \\
\mathbf{\hat{\theta}} & = [1, \theta_1, \theta_2, \ldots, \theta_{n-1}, \theta_n]^T \\
\mathbf{\hat{\alpha}} & = [\beta, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, 1]^T \\
& = [\beta, \mathbf{\alpha}, 1]^T
\end{align*}
\]  

(37)

(38)

(39)

The least-squares solution for (2) is given by minimizing the following criterion

\[
J = \min_{\theta} \|y - X\theta\|_2
\]

(40)

which leads to the normal equation in (45). In order to solve the multiple equations in (18), we need to consider the following criterion instead

\[
J = \min_{\alpha, \beta} \left[ \frac{1}{\beta^2} \|x_n - (\mathbf{-H\alpha} + \beta y)\|_2 \right]
\]

(41)

Assuming \(\theta = \begin{bmatrix} \mathbf{\alpha} \\ 1 \end{bmatrix} / \beta\), it is found that \(J\) in (41) is equivalent to that in (40) since

\[
J = \min_{\alpha, \beta} \left[ \|\frac{1}{\beta} x_n + \mathbf{H\alpha} - y\|_2 \right]
\]

The cost function (41) can be written into

\[
J = \frac{1}{\beta^2} (x_n + \mathbf{H\alpha} - \beta y)^T (x_n + \mathbf{H\alpha} - \beta y)
\]

(42)

Setting the derivative of \(J\) with respect to \(\mathbf{\alpha}\) to zero, i.e.,

\[
\frac{\partial J}{\partial \mathbf{\alpha}} = 0
\]

and through some manipulation, we have

\[
\mathbf{H}^\top [-y, \mathbf{H}] \begin{bmatrix} \beta \\ \mathbf{\alpha} \end{bmatrix} = \mathbf{H}^\top x_n
\]

(43)

Similarly, taking the derivative of \(J\) w.r.t. \(\beta\) as zero and by some rearrangement, we have

\[
x_n^\top [-y, \mathbf{H}] \begin{bmatrix} \beta \\ \mathbf{\alpha} \end{bmatrix} = x_n^\top x_n
\]

(44)

Combine equations (43) and (44), we have

\[
\begin{bmatrix} \mathbf{H}^\top \\ x_n \end{bmatrix} [-y, \mathbf{H}] \begin{bmatrix} \beta \\ \mathbf{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{H}^\top \\ x_n \end{bmatrix} x_n
\]

or

\[
\mathbf{X}^\top [-y, \mathbf{H}] \begin{bmatrix} \beta \\ \mathbf{\alpha} \end{bmatrix} = \mathbf{X}^\top x_n
\]

(45)

which is the normal equation for solving \(\beta\) and \(\mathbf{\alpha}\). Consequently, the desired least-squares solution \(\mathbf{\hat{\theta}}\) can be obtained by

\[
\mathbf{\hat{\theta}} = \begin{bmatrix} \mathbf{\alpha} \\ \beta \end{bmatrix}
\]

(46)
Following the same concept as the forward decomposition (Section 2.1), the augmented information matrix can be formulated as

$$ S = [-y, X]^{\top} [X, -y] $$

and LU factorization provides all the estimated $\hat{\alpha}$s and $\hat{\beta}$s that can be easily converted with (46) into the solution vector $\hat{\theta}$ for all the equations listed in (18).