6-2-1998

Method of Discrete Orthogonal Basis Restoration

Edward B. Moody
University of Kentucky

Click here to let us know how access to this document benefits you.

Follow this and additional works at: https://uknowledge.uky.edu/radiology_patents

Part of the Radiology Commons

Recommended Citation
https://uknowledge.uky.edu/radiology_patents/1

This Patent is brought to you for free and open access by the Radiology at UKnowledge. It has been accepted for inclusion in Radiology Faculty Patents by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.
A method is described for utilizing discrete orthogonal basis to restore signal system, such as radio or sound waves and/or image system such as photographs or medical images that become distorted while being acquired, transmitted and/or received. The signal or image system are of the linear type and may be represented by the equation \([B] \hat{[o]}= \hat{[i]}\) wherein \([o]\) is an original signal or image, \([i]\) is a degraded signal or image and \([B]\) is a system transfer function matrix. The method involves estimating a signal-to-noise ratio for a restored signal or image. Next, is the selection of a set of orthogonal basis set functions to provide a stable inverse solution based upon the estimated signal-to-noise ratio. This is followed by removing time and/or spatially varying distortions in the restored system and obtaining an appropriate inverse solution vector.

17 Claims, 7 Drawing Sheets
CONSTRUCT AN ORTHONORMAL BASIS SET \([p]_m\) WITH \(m=1,\ldots,M\) MEMBERS OF LENGTH 1U.

PROJECT THE ADJOINT OF THE SYSTEM MATRIX ONTO EACH \([p]_m\) TO FORM THE VECTOR SET \([b]_m\):
\[
[B][p]_m = [b]_m
\]

PERFORM GRAM–SCHMIDT ORTHOGONALIZATION ON THE VECTOR SET \([b]_m\) TO YIELD \([c]_m\), WHICH IS ORTHONORMAL.

REGROUP THE GRAM–SCHMIDT COEFFICIENTS INTO A SET OF CONSTANTS \(\tau\) SO THAT
\[
c_{mk} = \sum_{i=1}^{m} \tau_{mi} b_{ik}
\]
IS SATISFIED.

USING THE SAME SET OF CONSTANTS \(\tau\), THE VECTOR SET \([d]_m\) IS CALCULATED BY
\[
d_{mk} = \sum_{i=1}^{m} \tau_{mi} P_{ik}
\]

THE INVERSE SOLUTION ESTIMATE \(O_k\) IS THEN CALCULATED BY
\[
O_k = \sum_{m=1}^{M} c_{mk} \sum_{k=1}^{IU} d_{mk} i_k
\]
WHERE \(i_k\) IS THE FORWARD SOLUTION VECTOR.

FIG. 8
1

METHOD OF DISCRETE ORTHOGONAL BASIS RESTORATION

This is a continuation-in-part of U.S. patent application Ser. No. 08/337,592, filed on Nov. 10, 1994, entitled "Method of Discrete Orthogonal Basis Restoration", now abandoned.

TECHNICAL FIELD

The present invention relates generally to the field of signal and image restoration and, more particularly, to a method of restoring a signal and/or image degraded by time and/or spatially varying transfer functions.

BACKGROUND OF THE INVENTION

It is commonplace for signals, such as radio or sound waves and images such as photographs or medical images to become distorted while being acquired, transmitted and/or received. This phenomenon occurs in various types of radar, sonar, optic imaging and electronic systems.

As an example, blurring of a photographic image may result from camera and/or object motion at the time of acquisition or may even be produced by the nature of the photographic equipment (e.g. "fish-eye" lens). In medical imaging, the type of equipment used and the way the images are acquired can have remarkable effects on the level of distortion or blurring present in the final images that are interpreted by physicians.

It should further be appreciated that the characteristics of the disturbing process may change with time during the acquisition of a signal or may vary with location over different areas of an image. These time and/or spatially varying distortions in a signal and/or image must be removed to restore a signal and/or image to its undistorted form and enhance clarity.

In practical application there are imperfections in the signal or image acquisition process that make it impossible for any method to perfectly recover the original signal or image. Special mathematical techniques may, however, be utilized to closely estimate what the signal or image was before it was degraded. The time and/or spatially-varying nature of some systems makes it particularly difficult to perform a fully accurate restoration. Thus, when properly applied such techniques may be utilized to substantially improve the quality of a signal or image so that it more closely approximates the true or undistorted original signal or image.

In order to further understand this process it must be appreciated that signals or images degraded by a linear system may be cast in the operator notation $[B] [o] = [i]$; where $[o]$ is the original signal or image, $[i]$ is the degraded signal or image, and $[B]$ is the system transfer function matrix. Signal or image restoration is the determination of an approximation $[o']$ to the original signal $[o]$, given a priori knowledge of the transfer function matrix $[B]$ and the forward solution $[i]$.

The most straightforward means of determining the inverse solution is by application of the transfer function matrix inverse to the forward solution, that is, $[B]^{-1} [i] = [o']$. However, determining $[o']$ by this approach frequently represents an ill-posed problem as the inverse of the transfer function may not exist (singular matrix) or $[B]^{-1}$ may be near-singular. In either case, the inverse solution cannot be determined. Further, even if $[B]$ is invertible, $[B]^{-1}$ will frequently be ill-conditioned, meaning that small perturba-

2

tions in $[i]$ will lead to large perturbations in $[o']$ when the inverse solution is computed. This leads to unacceptable results. This is because all practical systems have inherent uncertainty in the measurement of $[i]$, as well as added noise, and accordingly, adequate estimation of the inverse solution $[o']$ is not possible through application of an ill-conditioned transfer function inverse.

To date, many methods have been developed to solve inverse problems arising in image processing, optics, geophysics, astronomy, spectroscopy, and other engineering and scientific disciplines. The existence of multiple solutions is primarily due to the fact that no single prior art method provides the best estimate of inverse solution in all practical applications. In fact, most prior art methods have only very specific, limited applications within specialized technical fields.

As $[B] [o] = [i]$ constitutes a linear system, solution by linear methods is an intuitively attractive approach. However, while the solution may be attempted by linear transform methods, such as Fourier transforms, the ill-conditioned nature is not circumvented by these techniques. Furthermore transform techniques are not directly applicable when the transfer function is shift-variant.

Application of transform methods to shift-variant systems have been limited to those cases where the signal or image can be sectioned into regions over which the system may be considered to be stationary. The inverse solutions for these regions are computed by transform techniques, and then spliced back together to form the overall solution. Similar sectioning into assumed stationary regions with inversion by the maximum a posteriori method has also been proposed. This sectioning and reassembly approach ("mosaicking") is, however, highly dependent on the validity of the stationary assumption, the method of reassembly, and on sampling of the forward solution and these considerations all adversely effect restoration results.

Various non-transform methods of linear inverse solution have also been developed. These include Weiner filtering, constrained Weiner filtering, maximum entropy, and pseudo-inversion techniques. These methods are usually applicable to the shift-variant case and they address the ill-conditioned nature of the problem. One drawback of such methods is, however, that they tend to not perform well in the presence of low signal-to-noise ratio (SNR) or on systems with moderate to severely degrading transfer functions. Thus they fail when they are most needed. These linear methods also do not provide super-resolution capability, and the linear iterative methods (e.g. pseudo-inversion, van Cittert's method, maximum entropy) do not have well defined termination points and can have very high memory and computational demands if a large number of iterations are performed. Thus, hardware requirements and processing times are disadvantageously increased.

The shortcomings of existing linear techniques has spawned great interest in non-linear approaches. The non-linear approaches are based on various regularization techniques that incorporate a priori knowledge of various parameters to yield an inverse solution that stabilizes and constrains the inverse solution. The parameter variables (hyperparameters) may include constraints on the form of the solution (such as non-negativity) goodness of fit parameters, statistical parameters, and assumptions of the character of added noise. Non-linear methods are usually applicable to shift-variant systems and may have super-resolution properties. The performance of these approaches is highly dependent on proper choice of the hyperparameters.
needed for the particular method. Furthermore, these approaches are usually iterative with poorly defined criteria for termination. For systems with well defined hyperparameters and termination criteria, and when the computational burden is not an obstacle, these are usually the preferred method of inverse solution. However, when the a priori knowledge of the system is inadequate, or when optimal termination of the iterative process is problematic, a linear method of solution is likely to produce better restoration results.

Another linear method of interest provides deconvolution for stationary systems based on the properties of the system adjoint operator. Referred to as deconvolution by the method of orthogonal polynomials (Stritzke IEEE Trans Med Imaging vol. 9, 1990, pp. 11-23), the crux of this method is the inner-product property of adjoint operator on vectors. The method requires a discrete orthogonal basis set. The original author, however, failed to define the origins of instability or the criteria for insuring a stable solution. Accordingly, the scope of practical applications of this approach is very limited.

From the above it should be appreciated that a need exists for a more versatile and effective method of signal and image restoration suited for a wide range of applications in various fields.

### SUMMARY OF THE INVENTION

Accordingly, it is a primary object of the present invention to provide an efficient and dependable method for signal and/or image restoration adapted for a number of specific applications crossing a broad number of technical fields.

Another object of the invention is to provide an improved method for quickly restoring a signal and/or image system degraded by time and/or spatially varying transfer functions. Such a system reduces processing time without comprising the quality of the final or restored image.

Yet another object of the invention is to provide a discrete orthogonal basis method for quickly restoring a signal or image to an undistorted form. Advantageously, the method utilizes a mathematical processing technique requiring relatively small computer memory capacity such as found in a personal computer, so as to allow ready application by individuals in many, differing fields utilizing readily available computer hardware. Further, the method also provides uncompromising speed of operation and very effective results.

A still further object of this invention is to provide a discrete orthogonal basis restoration method particularly suited to reconstruct and restore nuclear medicine SPECT images.

Additional objects, advantages and other novel features of the invention will be set forth in part in the description that follows and in part will become apparent to those skilled in the art upon examination of the following or may be learned with the practice of the invention. The objects and advantages of the invention may be realized and obtained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

To achieve the foregoing and other objects, and in accordance with the purposes of the present invention as described herein, an improved method is provided wherein discrete orthogonal basis is utilized to restore a signal and/or image system that is degraded by time and/or spatially varying transfer functions. Advantageously, the present method represents a relatively simple inverse solution that quickly and efficiently restores the system to an undistorted form. Accordingly, a clearer and more focused signal or image system results.

The method includes the step of estimating a SNR for a restored system. Additionally, there is the step of selecting of a set of orthogonal basis set functions \( p_{\text{est}} \) to provide a stable inverse solution based upon the estimated forward solution SNR.

More specifically, the estimated SNR of the restored system is provided by applying a given forward solution SNR and selected set of orthogonal basis set functions \( p_{\text{est}} \) to a realistic simulation model of the system. The set of orthogonal basis set functions \( p_{\text{est}} \) may be any orthogonal basis that spans the forward and inverse solution vector spaces. Such basis set functions include but are not limited to a group consisting of Hartley, Walsh, Haar, Legendre, Jacobi, Chebyshev, Gegenbauer, Hermite and Laguerre functions.

Next is the step of removing the time and/or spatially varying distortions in the restored system by obtaining an inverse solution vector \( a_k \) for a one dimensional restoration wherein:

\[
\hat{a}_k = \sum_{m=1}^{M} c_m \left[ \sum_{l=1}^{L} d_{ml}a_l \right]
\]

wherein:

\[
(b_m^* \| B \|^* \| p_m \|)
\]

where \( |B|^* \) is the transpose-complex conjugate of the matrix \( B \) and \( p_m \) is an \( M \) member orthogonal basis set and \( a \) are the standard Gram-Schmidt orthogonalization coefficients.

\[
\hat{a}_k = \frac{a_k}{\sum_{m=1}^{M} c_m \sum_{l=1}^{L} d_{ml}a_l}
\]

\[
\hat{a}_k = \frac{a_k}{\sum_{m=1}^{M} c_m \sum_{l=1}^{L} d_{ml}a_l}
\]

\[
d_{ml} = c_m d_{ml}
\]

Alternatively, the inverse solution vector \( \hat{a}_k \) for a two dimensional system with separable spatially variant PSF the inverse solution may be obtained by successive row-column operations:

\[
\hat{a}_{jk} = \sum_{m=1}^{M} c_m \sum_{l=1}^{L} d_{ml}a_{jl}
\]

wherein:

\[
\sum_{m=1}^{M} \sum_{l=1}^{L} d_{ml}a_{jl}
\]

More specifically describing the invention, the estimating of the signal-to-noise ratio SNR \( \text{SNR}_{\text{est}} \) is provided by the formula
wherein $|O_{\text{total}}|^2$ is the signal power in the original (undegraded) signal or image, $|O_{\text{inverse}}|^2$: the inverse solution noise power due to the approximation nature of the inverse solution and $|O_{\text{noise}}|^2$ is the noise power in the inverse solution power due to added noise in the forward solution.

Advantageously, the present method functions to define the origins of instability and behavior in the presence of noise. By applying the method to time-varying systems and using a technique for a priori determination of the SNR inverse solution, it is possible to insure stability and optimal selection of the basis set. Further, the method of inverse solution and SNR estimation may be successfully extended to the restoration of two-dimensional images degraded by spatially variant point spread functions.

The discovery of the origins of instability along with the development of an approach for selection of the optimal basis set to maximize inverse solution SNR, makes the present method a viable linear approach to inverse solution. Advantageously, the method is applicable to both stationary and shift-variant systems, is non-iterative, and is computationally efficient. Thus, the speed of processing and the size of the computer necessary to complete that processing are both reduced. Further, it should be appreciated that the only a priori information required to estimate the SNR of the inverse solution is an estimate of the forward solution noise characteristics and estimate of the inverse solution noise due to a limited basis set. Of course, in some cases the type of instrumentation or acquisition parameters may guide the optimal basis set selection (e.g. Nuclear Medicine SPECT imaging with reconstruction from projections).

As the present method is advantageously applicable to both stationary and shift-variant linear systems in one or more dimensions, potential applications for the present method include medical imaging (e.g. emission tomographic, MRI, ultrasound), image processing (lens deblurring, motion artifacts), optics and spectroscopy (light and NMR), geophysics, radar/sonar, and general electronics and electrical engineering problems. Thus, the method is extremely versatile, having application in broad ranging technical fields.

Still other objects of the present invention will become apparent to those skilled in this art from the following description wherein there is shown and described a preferred embodiment of the invention, simply by way of illustration of one of the modes best suited to carry out the invention. As it will be realized, the invention is capable of other different embodiments and its several details are capable of modification in various, obvious aspects all without departing from the invention. Accordingly, the drawings and descriptions will be regarded as illustrative in nature and not as restrictive.

**BRIEF DESCRIPTION OF THE DRAWING**

The accompanying drawing incorporated in and forming a part of the specification, illustrates several aspects of the present invention and together with the description serves to explain the principles of the invention. In the drawings:

FIG. 1 graphically shows the original signal (a) is the sum of three unity amplitude sinusoids $(f_1=3$ cycles/2π, $\phi_1=0.1$ radian; $f_2=7$ cycles/2π, $\phi_2=1.0$ radian; $f_3=10$ cycles/2π, $\phi_3=0.6$ radian). The forward solution (b) results when the

**DETAILED DESCRIPTION OF THE INVENTION**

The method of the present invention for using discrete orthogonal basis to restore a signal or image system created by time and spatially varying transfer functions will now be described in detail. The method may be applied to restore a signal or image system in a wide variety of technical fields. Stability of the inverse solution may be achieved if the characteristics of the noise in the forward solution may be estimated. For time-varying linear systems having a region of basis function support approximately congruent to the support region of the transfer function, and for which there is sufficient a priori knowledge of the system, the present method provides an efficient and noise tolerant approach to achieve inverse solution.

As previously described, the method of the present invention may be utilized to obtain an inverse solution vector for either one or two dimensional restorations. For purposes of presentation, $m=1,2,3,\ldots,M$ is the index for the vector sets.
with $M$ members. The row and column vectors are designated by lowercase letters, and square matrices by uppercase letters. The system matrix transfer function is denoted by $B$. Given the linear operation
\[ \{B\} \{v\} = \{l\} \] (1)

the purpose of this invention is to recover the length $I$ vector $\{v\}$, given the forward solution vector $\{l\}$ and a priori knowledge of the forward operator $B$. The $I \times JU$ matrix $B$ is constructed using the time (or spatially) varying system transfer function $b_{mn}$ so that the forward solution $l_k$ is defined by
\[ l_k = \sum_{m=1}^{IU} b_{km}v_m \]
(2)

$k = 1, 2, 3, \ldots JU$.

Recovery of the inverse solution requires two orthogonal function sets related to the adjoint PSF operator. The construction of these function sets (equations (4—6b)) requires a set of $M$ orthogonal basis set functions $p_{mn}$ of length $I$. The Hartley basis set, defined by
\[ p_{mn} = \sin \left( \frac{2\pi(m-1)y}{I} \right) + \cos \left( \frac{2\pi(m-1)y}{I} \right) \]
\[ k = 1, 2, 3, \ldots JU \]
\[ m = 1, 3, 5, \ldots M - 1 \]

is the preferred basis for real-valued systems and will be utilized to illustrate the present method. It should be appreciated, however, that any orthogonal basis that spans the forward and inverse solution vector spaces may be utilized. These include for example, Hartley, Walsh, Haar, Legendre, Jacobi, Chebychev, Gegenbauer, Hermite and Laguerre functions.

As can be appreciated in viewing FIG. 8, the method begins with the application of the adjoint of the forward operator to each member of the basis set
\[ \{b_{mn}\}^T \{p\} = \{c_{mn}\} \] (4)

where $\{B\}^T$ is the transpose-complex conjugate of the matrix $B$.

The Gram-Schmidt orthogonalization procedure followed by normalization may be used to construct an orthonormal function set $c_{mn}$ from $b_{mn}$. Defining a set of constants $\tau_{mn}$ (5), where the constants $\tau$ are the Gram-Schmidt orthogonalization constants, $c_{mn}$ may be defined as a linear combination of $b_{mn}$ (6a). A second orthogonal function set $d_{mn}$ can then be constructed as a linear combination of the basis set functions $p_{mn}$ using the same set of constants $\tau_{mn}$ (6b).

\[ \tau_{mn} = \frac{c_{mn}}{\sqrt{\sum_{k=1}^{M} c_{km}}^2} \]
(5a)

\[ \tau_{mn} = \frac{1}{\sqrt{\sum_{k=1}^{M} c_{km}}^2} \]
(5b)

\[ c_{mn} = \sum_{k=1}^{M} \tau_{mn}b_{kn} \]
(6a)

Recovery of the inverse solution begins with the unit operator for orthonormal functions (7). Reference to (6a) and (4) yields equation (O). The property of adjoint operators on finite-dimension inner product space allows transition to (9) and equation (1) leads to (10). Equation (10) is the operational equation for the inverse solution vector for a one dimensional restoration utilizing the present method.

\[ a_k = \sum_{m=1}^{M} \tau_{mn}b_{kn} \]
(7)

\[ a_k = \sum_{m=1}^{M} \tau_{mn}b_{kn} \]
(8)

\[ a_k = \sum_{m=1}^{M} \tau_{mn}b_{kn} \]
(9)

\[ a_k = \sum_{m=1}^{M} \tau_{mn}b_{kn} \]
(10)

Of course, this inverse solution $\{a\}$, like all method of ill-posed problem solution, is an approximation of the true solution. In the noiseless case, the quality of the inverse solution depends primarily on the quality of the representation of the true signal afforded by the chosen finite set of basis functions $p_{mn}$, where $m=1, 2, 3 \ldots M$ is the index for the chosen orthogonal basis set with $M$ members of length $k = 1, 2, 3 \ldots JU$. The recovered vector $\{a\}$ may be expressed as a sum of the inverse solutions from the noiseless forward solution and from added noise components (see equation 11).

\[ a_k = \sum_{m=1}^{M} \tau_{mn}b_{kn} \]
(11)

In the Fourier domain, the noise component term in (11) may be expressed by
\[ \delta(O_{noise}) = \sum_{m=1}^{M} C_{mn} \]
(12)

\[ i = 1, 2, 3 \ldots N \]

which allows an approximation of the inverse solution noise power to be made if the characteristics of the noise are known. The predicted SNR of the inverse solution may be estimated by
\[ \text{SNR}_{\text{pred}} = 10 \log \frac{\text{I}_{\text{original}}^2}{\text{I}_{\text{original}}^2 + \text{I}_{\text{noise}}^2} \]
(13)

where $\text{I}_{\text{original}}^2$ is the signal power in the original (undegraded) signal or image, $\text{I}_{\text{original}}^2$ is the inverse solution noise power due to the approximate nature of the inverse solution, and $\text{I}_{\text{noise}}^2$ is the noise power in the inverse solution due to added noise in the forward solution.

For a given system, an a priori estimate of the inverse solution SNR can be made for various values of $\text{SNR}_{\text{forward}}$ using equations (12) and (13). Noise power in the inverse solution due to added noise may be estimated by assigning values to
\[ \text{I}_{\text{noise}}^2 \]
(12) based on assumptions of the character and magnitude of added noise. Simulation studies with well modeled noiseless signals allow estimation of intrinsic noise. $\text{I}_{\text{original}}^2$ in the
An example of a linearly time-varying transfer function with a zero-mean pseudo-random noise added to the forward solution is as follows:

\[ h_{\text{in}}(t) = 0.5 + 0.5 \frac{n}{N} + \rho \]

The input function (Fig 1(a)) is a summation of three unity amplitude sinusoids of arbitrarily chosen frequency and phase, \( f_1 = 3 \text{ cycles}/2\pi, \phi_1 = 0.1 \text{ radian}; f_2 = 7 \text{ cycles}/2\pi, \phi_2 = 1.0 \text{ radian}; f_3 = 10 \text{ cycles}/2\pi, \phi_3 = 0.6 \text{ radian}. \] Zero-mean pseudo-random noise was added to the forward solution (Fig 1(b)) to achieve SNRs (SNR_{\text{forward}} = 10 \log (\sigma_{\text{forward}}^2/\sigma_{\text{added noise}}^2)) of 10.0 and 20.0 dB (Fig 1(c) and (d)). The method was performed using the Hartley basis set (equation 3) extending from 0 to 10 cycles/2\pi. The inverse solutions show good recovery of the original input function, with inverse solution SNRs (SNR_{\text{inverse}} = 10 \log (\sigma_{\text{inversion}}^2/\sigma_{\text{recovered noise}}^2)) of 19.26 dB (Fig 1(d)) for the 20 dB forward solution, and 10.33 dB for the 10 dB forward solution (Fig 1(f)).

It should also be appreciated that using an assumption of zero-mean white noise, and a predetermined value of \( \sigma_{\text{inversion}}^2 \), a plot of SNR_{\text{inverse}} vs. SNR_{\text{forward}} (see Fig. 2) may be constructed for the system described for Hartley basis function bandwidths of 10 and 14 cycles/2\pi. For the 0–10 cycles/2\pi Hartley basis set, the SNR_{\text{inverse}} values of 11.39 dB and 17.91 dB (for the 10 dB and 20 dB forward solutions respectively) correspond reasonably well with the experimental SNR_{\text{inverse}} values of 10.33 dB and 19.26 dB. Fig. 2 illustrates that increasing the recovery bandwidth from 10 to 14 cycles/2\pi results in improved SNR_{\text{inverse}} when the SNR_{\text{forward}} is high, due to the improvement in representation of the inverse solution afforded by a more complete basis set. However, with lower SNR_{\text{forward}} the effects of increased noise recovery accompanying expansion of the basis set offsets this advantage and results in lower SNR_{\text{inverse}} of the inverse solution. This method of SNR_{\text{inverse}} estimation can be performed using training sets of large numbers of simulated signals and noise levels to determine the best selection of basis set for a given application.

The forward solution \( I_{\text{pk}} \) for two-dimensional separable spatially variant point spread function (SSVPSF) systems may be obtained by successive application of column and row degradation operators to the original image \( C_{\text{pk}} \). Obtaining the inverse solution for the SSVPSF system by the present method follows the general approach for matrix operators on separable systems. For each column \( \rho = 1, 2, \ldots, IU \), the blurring matrix across the columns, \( B_{\text{pk}} \), is constructed, allowing calculation of the corresponding orthogonal sets \( c_{\text{pk}} \) and \( d_{\text{pk}} \) (6). Successive application of the inverse operational equation (10) to all of the rows yields the IU\times IU intermediary matrix \( I_{\text{pk}} \), which has been corrected for the blurring across columns.
is likely to provide the best restoration. Large numbers of simulations with training sets of images and noise levels appropriate to the application can be used to determine the optimal basis set for a particular application.

The following example is presented for purposes of further illustrating the present invention, but it is not to be considered as limited thereto.

**EXAMPLE 2**

A 128×128 pixel four quadrant checkerboard pattern with quadrant check sizes of 4.5,6, and 7 pixels (see photographic FIG. 3) was chosen for illustration of the present method in restoring a SSVPFS system. The light squares were assigned a value of 1.0 and the dark squares 0.3. The multiple high-contrast discontinuities in this image were designed to be particularly challenging for a restoration method using a bandlimited basis set. The spatially varying gaussian transfer function had exponentially-radially varying FWHM of 6 pixels in the image center and FWHM of 3 pixels at the image corners. The PSF located at the center of the image space was assigned an amplitude so that the center PSF was lossless. PSF amplitude declined radially in exponential fashion so that the PSF amplitude in the corners was half of the center PSF amplitude.

Based on the methods described above, the Hartley basis set utilized was restricted to +/- 18 cycles/2π to assure a stable restoration for an anticipated forward solution SNR of 20 dB. Zero-mean pseudorandom noise was added to the forward solution to achieve a SNR of 20 dB (see photographic FIG. 4). The restored image (see photographic FIG. 5) from this noisy forward solution showed good contrast recovery with resolution of all image elements. Due to the presence of method recovered noise the subjective quality is inferior to the noiseless restoration shown in photographic FIG. 6, which is subject only to intrinsic noise, but is a dramatic improvement from the degraded image shown in photographic FIG. 4.

The model basis embodiment described extends directly to two or more dimensions provided that it is cast in "stacked lexicographic" format e.g. A 2×2 image is acted on by a degrading operator to yield a forward solution 2×2 image;

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

in stacked lexicographic notation this may be cast as the one dimensional problem

\[
\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix} \rightarrow \begin{bmatrix}
A & B \\
C & D
\end{bmatrix}
\]

This is a common approach in image processing to reduce multidimensional problems to 1-D problems.

As should be appreciated, the above described method for discrete orthogonal basis restoration (DOBR) is a time-domain approach. In an alternative embodiment of the present invention, discrete orthogonal basis restoration is presented as a frequency domain approach for the estimation of the inverse solution vector for linear systems defined by the matrix operation

\[ [B] [\omega] = [i] \]

where \([B]\) is an \(N\timesN\) non-singular transfer function matrix and \([\omega]\) and \([i]\) are length \(N\) column and row vectors. In this alternative embodiment, upper case letters denote frequency domain variables while lower case letters denote time-domain variables. For vector sets with two subscripts, such as \(C_{m,n}\) the first denotes the set position and the second the discrete time or frequency index.

The normalized Hartley basis set is preferred for this alternative approach of discrete orthogonal basis restoration. The Fourier representation of a Hartley basis vector of frequency \(f_n\) cycles/2π is non zero only at the \(\pm f_n\) discrete frequencies. All positive frequency Hartley basis vectors have identical complex amplitude

\[ \frac{\sqrt{N}}{2} \frac{\sqrt{N}}{2} \]

in their non-zero positive frequency bin, and the complex conjugate of this value in the negative frequency bin. Negative frequency Hartley basis vectors are the complex conjugates of the positive frequency spectrum. These relationships facilitate rapid computation and efficient storage of the basis set. The members of the frequency domain basis set, \(F_{m,n}\), are ordered such that the DC component is assigned index \(m=1\), even values of the basis set index \(m\) correspond to positive Hartley frequency \(f=2\) and odd values of \(m\) correspond to \(f=-(m-1)/2\) Hartley frequency. For a selected DOBR bandlimit of \(0-f_{\text{max}}\) cycles/2π, there are \(M=(2f_{\text{max}}+1)\) basis and vectors, so that the composite basis set spectrum is non-zero for \(m=1, \ldots, (f_{\text{max}}+1)\) and \((N-M+1), \ldots, N\). The relative compactness of the frequency domain representation of the basis set is instrumental in the development of an efficient frequency domain approach.

The initial step in the time-domain approach is the application of the adjoint (complex conjugate transpose) of the transfer function \([B]\) to each member of the orthogonal basis set \(F_{m,n}\) to yield the vector set \(b_{m,n}\). For the time varying case, the M frequency domain row vectors \(A_{m,n}\) may be determined by \([F]^{-1} [P]_{n} = [A]_{n}\), where the \(N\timesN\) matrix \([F]^{-1} = [DFT][B]^{-1}[DFT]^{*}\), and \([DFT]\) and \([DFT]^{*}\) are the discrete and inverse discrete Fourier transform matrices. Using the properties \([F]=([F]^{*})^{*}\) and \([DFT]=k[DFT]^{*}\), it follows that \([F]=([F]^{*})^{*} [DFT][B])\), and noted that the spectrum of the frequency domain solution is given by

\[ \sum_{n} F_{m,n} G_{n} = I_{k} \]  

(18)

For non-singular \([B]\) the vector set \(A_{m,n}\) is linearly independent, but usually not orthogonal. Gram-Schmidt orthogonalization can be performed on the frequency domain vector set \(A_{m,n}\) to yield an orthogonal complex set \(C_{m,n}\). Regrouping the complex Gram-Schmidt coefficients \(a_{m,n}\)

\[ a_{m,n} = \sum_{k=1}^{N} C_{m,k}^{*} C_{n,k} \]  

(19)

\[ i = 1, \ldots, m - 1 \]  

\[ m = 1, \ldots, M_{k} \]  

into a set of constants \(c\) defined by

\[ c_{m} = \sum_{n=1}^{N} C_{m,n}^{*} C_{m,n} \]  

(20)
allows the orthonormal vector set $C_{mk}$ to be expressed as a superposition of prior $A_{ik}$ by

$$C_{mk} = \sum_{i=1}^{M} \sum_{n=1}^{N} c_{i,m} a_{i,n}$$

A second complex vector set $D_{mk}$ is obtained by applying the same operation to the orthogonal basis set $F_{ik}$.

$$D_{mk} = \sum_{i=1}^{M} \sum_{n=1}^{N} d_{i,m} a_{i,n}$$

The vector sets $C_{mk}$ and $D_{mk}$ define the characteristics of the system $[B]$ for frequency domain DOBR.

In the frequency domain, the completeness relationship [24] for orthonormal vectors may be written as:

$$O_k = \frac{1}{N} \sum_{m=1}^{M} C_{mk} N^{1/2} a_{i,n}$$

Transferring the complex conjugation to $C_{mk}$, substituting (22) and using the relationship $[F]^{T} [F]_{m,n} = [A]_{m,n}$ yields

$$O_k = \frac{1}{N} \sum_{m=1}^{M} C_{mk} N^{1/2} a_{i,n}$$

which by the property of adjoint operators on inner product spaces [25] is equivalent to

$$O_k = \frac{1}{N} \sum_{m=1}^{M} C_{mk} N^{1/2} a_{i,n}$$

Reference to (18) and (23) yields the frequency domain DOBR operational equation

$$O_k = \frac{1}{N} \sum_{m=1}^{M} C_{mk} N^{1/2} a_{i,n}$$

$O_k$ is the spectral estimate of the inverse solution. The time domain inverse solution $O_k$ may be obtained by inverse FFT of $O_k$.

The steps where major computational differences exist between the time and frequency domain approaches involve summations that may be limited to regions of time or frequency domain support. In the frequency domain approach the range of summation is restricted to the $M$ discrete frequencies where the composite basis set spectrum in non-zero. The frequency domain DOBR approach saves $(N-M)$ multiplications and additions for each of the many inner products required by the approach. The reduction in the number of computations is at the expense of substituting complex for real operations. This is not particularly disadvantageous for additions, as current generation microprocessors perform complex and real additions with an equivalent number of clock cycles. Complex multiplications are more time consuming than real multiplications, but for practical DOBR applications the reduction in computations offsets the increased processor time.

The initial step in the frequency domain approach is the determination of $A_{mk}$. When $[B]$ is time varying, $A_{mk}$ may be estimated by $|DFT| [B]^{*} |IDFT| [P]_{mk}$, but this is less efficient than calculating the time domain $[b]_{mk} = [B]^{*} [p]_{mk}$ and subsequently performing an $N$ point FFT on each vector $[b]_{mk}$. Using this approach, an additional $M(N \log_2 N)$ complex additions and $M(N \log_2 N)$ complex multiplications are required for the frequency domain approach. Determining $C_{mk}$ by Gram-Schmidt orthogonalization requires calculation of $(M^2 - M)$ coefficients, for which the computational advantage of the frequency domain approach is $2NC(N-N)$ multiplications and additions. Performing the linear combinations of vectors weighted by these coefficients yields computational savings of $(M \log_2 N)$ multiplications and $(N-M) \log_2 N$ additions for the frequency domain approach. Normalization of the $C_{mk}$ has savings of $M(N-M)$ multiplications compared to the time-domain approach. The sparsity of the vectors $P_{mk}$, which are non-zero at only two discrete frequencies for $m=1$, and at one frequency (DC) for $m=1$, allow rapid frequency domain computation of the set $D_{mk}$. The computational advantage of the frequency domain approach is $(M+N)(N-N)$ additions and multiplications for the determination of $D_{mk}$.

Practical situations usually involve applying the DOBR operational equation for a stationary system $[B]$ to multiple forward solutions. The vector sets $C_{mk}$ and $D_{mk}$ define the DOBR inverse system for $[B]$, and may be computed, stored, and recalled for each implementation of the operational equation. Storage requirements are $2M^2$ complex numbers for the frequency domain approach and $2(MN)$ real numbers for the time-domain approach, so that a reduction in storage requirements is realized when $M<N/2$. The total operational equation computational advantage for the frequency domain approach is $2M(N-M)$ multiplications and additions, which is reduced by $N \log_2 N$ additions and $N/2 \log_2 N$ multiplications if the FFT of $i_k$ to yield $I_k$ is required. If the spectral estimate of the inverse solution is desired, the approach may be terminated at this point. Obtaining the time-domain solution by inverse FFT of $O_k$ requires an additional $N \log_2 N$ complex additions $N/2 \log_2 N$ complex multiplications, and $N$ real multiplications.

An additional advantage of the frequency domain approach is that it is more robust when there are significant perturbations in $[B]$. Errors in transfer function estimation are normally transmitted to the $b_{mk}$ and ultimately have adverse effects on the inverse solution [21]. In the frequency domain approach, the noise components in $[B]$ transmitted to $b_{mk}$ lying outside of the DOBR bandwidth are not included in the calculations of $C_{mk}$, and therefore are not propagated to the final solution. The inverse solution obtained from the frequency domain approach may be expected to be of higher quality than would be obtained from the time domain approach, especially when large errors in the estimation of $[B]$ are present.

**EXAMPLE 3**

Consider an application with a time duration signal of $N=128$ samples and a DOBR basis set bandwidth of $0-10$ cycles/2π, requiring $M=21$ Hartley basis vectors. For a previously defined stationary system, DOBR is performed by executing the operational equation (27) with stored values of $C_{mk}$ and $D_{mk}$. The time domain operational equation requires $5376$ real multiplications and $5334$ real additions. The frequency domain operational equation, including the FFT of $i_k$ and IFFT of $O_k$, requires $1778$ complex multiplications, $1736$ complex additions, and $128$ real multiplications, a reduction by a factor of $2.9$ in the number of operational equation computations.

Dynamic systems, and initial applications of the frequency domain approach require computation of $C_{mk}$ and
D_{\text{out}} and achieve all of the computational savings described above. Despite the computational debt established by (M+2) N-point FFTs, significant computational savings are realized by the frequency domain approach. For the example case of N=128 and M=21, the net savings are 82542 additions and 92852 multiplications. This reduces the number of operations performed after computation of the vector set b_{\text{out}} by a factor of 3.9 when compared to the time-domain approach. The computation of b_{\text{out}} common to both the time and frequency domain approach, is very computationally demanding, requiring MN^2 multiplications and M(N^2-N) additions.

It should be clear that DOBR is a robust method of inverse solution for time-varying or time-invariant linear systems expressed as a square matrix operator. The frequency domain approach exploits the compactness of frequency domain support exhibited by the DOBR vector sets to reduce the range of most summations from the N points of the time-domain signal to the M complex values of the frequency domain DOBR bandwidth. Frequency domain DOBR is used to greatest advantage when system stationarity allows repeated implementations of the operational equation with predetermined vector sets C_{\text{out}} and D_{\text{out}}, but also offers improvements in computational efficiency when the entire approach must be executed. The frequency domain DOBR approach significantly reduces the storage requirements and the number of arithmetic computations for DOBR, as well as lessening the deleterious effects of transfer function perturbations of the inverse solution.

Whether the original or alternative embodiment of discrete orthogonal basis restoration is utilized, it has been found that the present method possesses advantages over other prior art methods when there is noise in the system transfer function [B]. More specifically, the present discrete orthogonal basis restoration method may be utilized to assess the effects that observed noise in the transfer function and the forward solution have on the errors in the inverse solution estimate. It is assumed that there is no transmission of noise by the system and that the observed noise in the forward solution and transfer function are mutually independent.

In the presence of perturbations, the linear system [B][o] = [i], becomes

\[ [B+N][i] = [i] + [h] \]

where [N] is additive transfer function noise, [i] is additive forward solution noise, and [o] is the estimated inverse solution for the perturbed system.

For the system perturbed only by noise in the forward solution [B][o] = [i], the DOBR inverse solution is the superposition of DOBR solutions for the signal and noise components, which in the frequency domain is

\[ O_{i} = \sum_{m=1}^{M} C_{m} \frac{1}{NP} D_{m} + \sum_{m=1}^{M} C_{m} \frac{1}{NP} D_{m} \]

where NP is the number of discrete frequencies used in frequency domain DOBR.

The signal to noise ratio of the inverse solution in the presence of forward solution noise may be estimated by

\[ \text{SNR}_{\text{inverse}} = 10 \log \left( \frac{\|O_{\text{true}}\|^2}{\|O_{\text{noise}}\|^2 + \|O_{\text{true}}\|^2} \right) \]

where \( \|O_{\text{true}}\|^2 \) is the true inverse solution power, \( \|O_{\text{noise}}\|^2 \) is intrinsic noise resulting from error in estimation of the inverse solution with a limited basis set, and \( \|O_{\text{true}}\|^2 \) is the DOBR inverse solution of the forward solution noise component (right hand term in (11)).

Unlike forward solution noise, transfer function perturbations change the characteristics of the vector sets C_{\text{out}} and d_{\text{out}}, which define the behavior of the DOBR operational equation. Application of the perturbed transfer function [B+N] to each member of the basis set p_{\text{m}}, yields

\[ [B+N][p_{\text{m}}] = [p_{\text{m}}] + [h] \]

As was the case with forward solution noise, it is beneficial to analyze the effects of transfer function noise in the frequency domain. For the time-invariant (evolutionary) case the calculation of A_{m} reduces to the point by point multiplication of the Fourier transform of the transfer function and P_{m}. When [B] is a time-varying system, the spectral estimation of A_{m} is considerably more complex. In the noiseless case, the spectra A_{m} may be represented by

\[ A_{m} = \text{DFT}[B] \hat{A}_{m} \text{DFT}[P] \]

where \( \text{DFT} \) and \( \hat{A}_{m} \) are the NPXNP discrete Fourier transform and inverse discrete Fourier transform matrices, and \( [P] \) is the 1xNP column vector frequency domain representation of the mth basis function [8]. In the presence of transfer function noise, A_{m} may be estimated by:

\[ A_{m} = \text{DFT}[B+N] \hat{A}_{m} \text{DFT}[P] \]

\[ A_{m} = \text{DFT}[B] \hat{A}_{m} \text{DFT}[P] + [N] \text{DFT}[P] \]

The resultant noise in A_{m} is therefore the superposition of A_{m} and A_{m}.

The worst-case scenario will be considered in evaluating the propagation of noise in the frequency domain Gram-Schmidt orthogonalization process. Maximal noise transmission occurs when each A_{m} is orthogonal to C_{m-1} (FIG. T), and therefore has complete projection onto the vector C_{m}. The Gram-Schmidt coefficients involving the inner products of A_{m} and prior C_{m} are zero by virtue of orthogonality, greatly simplifying the computational process. Each C_{m} is then formed from A_{m}+A_{m} minus the projections of A_{m} on prior C_{m}.

\[ C_{1} = A_{1} = A_{1} + A_{1} \]

\[ C_{2} = A_{2} + A_{2} - \frac{A_{1} \cdot A_{2}}{|C_{2}|^2} \]

\[ C_{m} = A_{m} + A_{m} - \frac{1}{2} \sum_{i=1}^{m-1} \frac{A_{m} \cdot A_{i}}{|C_{i}|^2} \]

Each C_{m} is subsequently normalized, which preserves the relative noise contribution to C_{m} but depending on the amplification properties of the transfer function [B], the normalization may either increase or decrease the magnitude of noise transmitted to C_{m}. For the worst case scenario the total transmitted noise to C_{m} is

\[ \|w_{i}\| = A_{m} + \frac{M}{\|C_{m}\|^2} \frac{A_{m}}{|C_{m}|^2} \]

The frequency domain Gram-Schmidt orthogonalization constants

\[ \omega_{m} = \frac{\text{DFT}[w_{m}]}{|C_{m}|} \]

can be regrouped and normalized into the set of constants \( \phi_{m} \), so that each C_{m} is expressed as a linear combination of
A', (39). The D_m is then calculated using the same set of constants, g_{mi} (40).

\[ C_m = \sum_{i=1}^{m} g_{mi} A_k \]  
\[ D_m = \sum_{i=1}^{m} g_{mi} P_k \]  (39)  (40)

The frequency domain operational equation in the presence of transfer function noise is

\[ O_m = \frac{M}{n} C_m \frac{1}{NP} \sum_{k=1}^{NP} D_m P_k \]  (41)

The vector set D_m is constructed to be compatible with the set C_m and does not contribute additional noise to the inverse solution. The worst case inverse solution SNR due to transfer function perturbations may be estimated by

\[ SNR_{inv} = 10 \log \frac{\|O_m\|^2}{\|w_{nomega} + w_p\|^2} \]  (42)

In summary, the method of the present invention is equally applicable and extends to both one dimensional and two dimensional signal and image systems. The chief advantage of the present method compared to other SSVPSF restoration techniques are that preprocessing, matrix inversion, multiple iterations, or assumptions of local PSF variance are not required. As a result, processing times and computer power required for processing are both substantially reduced.

The primary limitation of the present method is that the basis function support of the inverse solution must be nearly congruent to the region of basis support of the transfer function to avoid an unstable inverse solution. Provided that a realistic simulation model exists, however, estimation of the restored image SNR may be made for a given forward solution SNR and chosen basis set. This allows a priori determination of the optimal basis set for a given application and provides an estimate of the anticipated quality of the restored image.

The foregoing description of a preferred embodiment of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Obvious modifications or variations are possible in light of the above teachings. The embodiment was chosen and described to provide the best illustration of the principles of the invention and its practical application to thereby enable one of ordinary skill in the art to utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. All such modifications and variations are within the scope of the invention as determined by the appended claims when interpreted in accordance with the breadth to which they are fairly, legally and equitably entitled.
APPENDIX A

c harpat.for
c .performs discrete orthogonal basis restoration for invariant
c .systems
c .using Hartley basis set
c .calls: mat hart inprod

::variables:::

izz is the maximum number of basis functions
i is the length of the fwd soln vector (ri) and inv. soln vector (h)
the basis set is determined and placed in poly
the basis set used for the inv soln is ordered transferred
to xleg
gau is the matrix formulation of the transfer function
vector g,
and gt is it's transpose. Note that gau and gt are cast for
row*square matrix=row vector fwd soln
The gram-schmidt coef are stored in matrix a
The tau coef are in tau
The vector sets b,c,d with m members correspond to the literature
descriptions. f(m) are <d(m),i> and are intermediate step
the inverse soln is h
npol is the number of basis vectors used
the array cnorm holds the normalization factors for each member of the vector set c

implicit integer(i-o)
parameter(izz=65,iu=128,twopi=6.283185)
common/c1/delt,rip
common/c2/rpl(iu),rp2(iu)
common/c4/poly(0:izz,iu)
common/c5/iz
common/c6/gau(iu),gt(iu,iu)
common/c6a/g(iu,((2*iu)-1))
common/c11/rphas
real b(izz,iu),c(izz,iu),d(izz,iu)
real a(izz,izz),tau(izz,izz)
real rleg(0:izz,iu)
real f(izz),h(iu),cnorm(izz)
real ri(iu),sumc(iu)
exteger ifini
integer npol
character*13 pre
character*4 suf3,suf2
character*10 file,filen,filg,filh,taunm,obas
character*30 taunmx,obax
character*30 filex,filegx,filein,fileh
character*2 ntrial
character*1 cxans,cyans
pre='c:\fort\data\'
  suf2='dec'
  suf3='hdc'
c......compute Hartley Basis Set
  iz=64
    write(6,'(a)\% calling hart\%\%\%','
    call hart

c....normalization of basis
  c
  do 820 m=0,iz
  do 812 kkk=1,iz
    rp1(kkk)=poly(m,kkk)
    rp2(kkk)=poly(m,kkk)
    call inprod
  do 815 kkk=1,iz
    poly(m,kkk)=(poly(m,kkk))/(sqrt(rp1))
  continue
  820 continue

c....read in the ideal image and the PSP
  write(6,'(a)\% name for header file xxxxxx.xxx','
  read(5,933)flh
  fileh=pre//flh
  open(25,file=fileh,status='new')
    write(6,'(a)\% input filename of PSP xxxxxx.xxx','
    read(5,933)fileg
    filegx=pre//fileg
    igau=0
    open(20,file=filegx,status='old')
  do 43 i=1,iz
    read(20,*,end=433)gau(i)
    igau=igau+1
  continue
  43 close(20)
  933 format(a10)
  write(6,'(a)\% input trial number for this input fn','
  read(5,964)ntrial
  964 format(a2)
  write(6,'(a)\% input filename of fwd soln xxxxxx.xxx','
  read(5,933)filen

c....set the restoration bandwidth
  write(6,'(a)\% high freq c/o','
  read(5,*t)fini
  npol=(2*fini)+1

c!!!!!!!!!!!!!! build the selected orthogonal set!!!!!!!!
c....always include the dc component
  do 24 ii=1,iz
  24 rleg(1,ii)=poly(0,ii)
iptr=0
do 25 j=1,ifini
    iptr=iptr+2
    jind=(j*2)-1
do 22 ii=1,iu
    rleg(iptr,ii)=poly(jind,ii)
    rleg(iptr+1,ii)=poly(jind+1,ii)
    continue
22 write(6,'$\$\$\$ orthonomol setup done $\$\$\$')
c.............................
c..............................
c..compensate for extra loop trip and truncate to odd #
    igau=igau-1
    ifig2=igau/2.
    ifig2=igau/2.
    if(fig2.eq.fif) then
        igau=igau-1
    endif
    c...call mat to cast the convolution operator in matrix form
    c... note that this is in row vector format
    call mat(igau)

  c>>>>>>>>>>calc b(k) using matrix
  approach>>>>>>>>>>>>>>>>>>>>>>>>>
  write(6,'.....calculating b(k).....')
c*****use transpose matrix
    iz=(2*ifini)+1
    do 9 ic=1,iz
        do 8 ir=1,iz
            sum=sum+(rleg(k,ir)*gt(ir,ic))
        end do
        continue
    b(k,ic)=sum
  9 continue
  10 continue
  c>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
  c?????????calc c(k)?????????????????????????????????????????????????????
  write(6,'+++++calculating c(k)+++++')
do 21 kkk=1,iu
    c(1,kkk)=b(1,kkk)
  23 a(m,m)=1.
c......calculate Gram-Schmidt coefficients
do 50 m=2,iz
do 40 j=1,m-1
    do 30 k=1,iu
      rp1(kkk)=c(j,kkk)
      rp2(kkk)=rp1(kkk)
      call inprod
      rnorm=rip
    end do
  end do
  
  do 29 kkk=1,iu
    rp1(kkk)=c(j,kkk)
    rp2(kkk)=b(m,kkk)
    call inprod
    a(m,j)=1.*(rip/rnorm)
  end do
  continue
  
c........now calculate c(m) by G-S orthogonalization
  do 46 ii=1,iu
    sumc(ii)=0.
  end do
  
  do 49 j=1,m-1
    do 48 kkk=1,iu
      sumc(kkk)=sumc(kkk)+(a(m,j)*c(j,kkk))
    end do
  end do
  continue
  
  do 720 m=1,iz
    do 712 kkk=1,iu
      rp1(kkk)=c(m,kkk)
      rp2(kkk)=c(m,kkk)
      call inprod
      cnorm(m)=sqrt(rip)
    end do
    715 kkk=1,iu
    c(m,kkk)=(c(m,kkk))/sqrt(rip))
  end do
  continue
  
c~~~~~~~~~~~~~~calculate tau~~~~~~~~~~~~~~
  write(6,*)'=====calculating tau=====
  do 75 k=1,iz
    tau(k,k)=a(k,k)
  end do
  
dos 83 i=1,m-1
    if(i.ne.m)then
      rsum=0.
    end if
    83 rsum=rsum+(a(m,j)*tau(j,i))
  end do
  continue
  tau(m,i)=rsum
else
eendif
85 continue
90 continue
c normalize tau by dividing by cnorms
do 94 m=1,iz
do 92 i=1,iz
    tau(m,i)=(tau(m,i)/cnorm(m))
92 continue
94 continue
c++++++++++++++++++++++++++++++++++++++++++++++++++++++
c!!!!!!!!!!!!!!!!!!!!!!!!!!!!calculate d(iz,iu)!!!!!!!!!!!
write(6,*)'!!!!!!calculating d(iz,iu)!!!!!!'
do 110 m=1,iz
    do 105 i=1,m
    do 100 it=1,iu
d(m, it) = d(m, it) + (tau(m, i) * rleg(i, it))
100 continue
105 continue
110 continue
c:::::::::::::::::::::::::::::::::::::::::::::::::::::::::
cfilex=pre//d'//ntrial//filen(5:6)
* //fileg(5:6)//suf2
fileh=pre//ntrial//filen(5:6)
* //fileg(5:6)//suf3
filein=pre//filen(1:10)
filegx=pre//fileg(1:10)
c:::-------------------------------------:
c:::-------------------------------------:
open(19, file=filein, status='old')
do 32 i=1,iu
    read(19,*,end=519) r1(i)
32 continue
519 close(19)

izk=npol
c....clear arrays
do 190 iy=1,izk
    f(iy)=0.
do 191 iy=1,iu
h(iy)=0.
c????????????????????compute f................................
do 120 m=1,izk
    rsum=0.
do 115 i=1,iu
    rsum=rsum+{d(m, i)*ri(i)}
115 continue
f(m)=rsum
write(6,*)m,'c(m)='',f(m)
120 continue
c ****************************************
calculate h

do 140 i=1,lu
rsum=0.
do 130 m=1,izk
rsum=rsum+(c(m,i)*f(m))
130 continue
h(i)=rsum
140 continue
c
open(18,file=filex,status='new')
do 150 i=1,lu
150 write(18,*)h(i)
close(18)

777 continue
write(6,*)filex
c
write(25,*)"iu= \',lu
close(25)
c!!!!!!!!!
c: : : this section allows various process vectors and
matrices to

: : : : be saved to disk for later analysis
write(6,*)' write taufile (y/n)? '
read(5,976)cxans
976 format(a1)
if(cxans.eq.'y')then
write(6,*)'enter name of taufile'
read(5,977)taumn
977 format(a10)
taumnx-pre//taumn
open(27,file=taumnx,status='new')
write(6,*)'iz= \',iz
do 833 m=1,iz
do 833 i=1,iz
write(27,*)tau(m,i)
833 continue
close(27)
endif

888 write(6,*)' save to disk d(m) array ? '
read(5,916)cyns
916 format(a1)
if(cyns.eq.'y')then
write(6,*)' # rows= \',iz
write(6,*)' input name of otpt file xxxxxx.xxx '
read(5,933)obas
obasx-pre//obas
open(30,file=obasx,status='new')
do 501 ibas=1,iz
do 501 ku=1,1u
501 write(30,*)d(ibas,ku)
close(30)
endif
write(6,*)' save to disk c(m) array ? '
read(5,916)cyans
if(cyans.eq.'y')then
write(6,*)' input name of opt file xxxxxx.xxx '
read(5,933)obas
obasx=pree/obas
open(30, file=obasx,status='new')
do 503 ibas=1,iz
do 503 ku=1,iu
503 write(30,*)c(ibas,ku)
close(30)
endif
write(6,*)' save to disk b(m) array ? '
read(5,916)cyans
if(cyans.eq.'y')then
write(6,*)'iz= ',iz
write(6,*)' input name of opt file xxxxxx.xxx '
read(5,933)obas
obasx=pree/obas
open(31, file=obasx,status='new')
do 504 ibas=1,iz
do 504 ku=1,iu
504 write(31,*)b(ibas,ku)
close(31)
endif
write(6,*)' save g-T ? '
read(5,916)cyans
if(cyans.eq.'y')then
write(6,*)' input name of opt file xxxxxx.xxx '
read(5,933)obas
obasx=pree/obas
open(30, file=obasx,status='new')
do 506 ibas=1,1u
506 write(30,*)gt(ibas,ku),ku=1,iu)
close(30)
endif
C>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
stop
end
APPENDIX B

c tvpat.for
.c performs discrete orthogonal basis restoration for invariant
c systems
c... using Hartley basis set
.c... calls: hart inprod

:::variable:::

c izz is the maximum number of basis functions
.c iu is the length of the fwd soln vector (ri) and inv. soln vector (h)
c the basis set is determined and placed in poly
c the basis set used for the inv soln is ordered transferred
c to rleg
c tvtfm is the imported matrix transfer function
.c The gram-schmidt coef are stored in matrix a
.c The tau coef are in tau
.c The vector sets b,c,d with m members correspond to the
.c literature
.c descriptions, f(m) are <d(m),i> and are an
.c intermediate step
c the inverse soln is h
.c npol is the number of basis vectors used
c the array cnorm holds the normalization factors for each
c member of the vector set c

implicit integer(i-o)
parameter(izz=65,iu=128,twopi=6.283185)
common/c1/delt,rip
common/c2/rpl(iu),rp2(iu)
common/c4/poly(0:izz,iu)
common/c5/iz
.common/c6a/g(iu,((2*iu)-1))
.common/c11/rphas
real b(izz,iu),c(izz,iu),d(izz,iu)
real a(izz,izz),tau(izz,izz)
real rleg(0:izz,iu)
real f(izz),h(iu),cnorm(izz)
.real ri(iu),sumc(iu)
.real tvtfm(iu,iu)
integer ifini
.integer npol
.character*13 pre
.character*4 suf3,suf2
.character*10 filen,fileg,flh,taumn,obas
.character*30 taumnx,obasx
.character*30 filex,filegx,filein,fileh
.character*2 ntrial
.character*1 cxans,c yans
pre='c:\fort\data\'
suf2='\data'
C........compute Hartley Basis Set
    iz=64
    write(6,*),'% calling hart
    call hart

C.... normalization of basis
C
    do 820 m=0,iz
    do 812 kkk=1,iu
       rp1(kkk)=poly(m,kkk)
    812
    rp2(kkk)=poly(m,kkk)
    call inprod
    do 815 kkk=1,iu
       poly(m,kkk)=(poly(m,kkk))/(sqrt(rip))
     815 exit
     820 continue

C.... read in the ideal image and the PSF
    write(6,*),' input name for header file xxxxxx.xxx
    read(5,913)filh
    fileh=pre//flh
    open(25,filname=fileh,status='new')
  933
    format(a10)
    write(6,*),' input trial number for this input fn'
    read(5,964)ntrial
  964
    format(a2)
    write(6,*),' input filename of fwd soln xxxxxx.xxx
    read(5,933)filen
C.... set the restoration bandwidth
    write(6,*),' high freq c/o'
    read(5,*)ifini
    npol=(2*ifini)+1

C!!!!!!!!!!!build the selected orthogonal set!!!!!!!!!
C......always include the dc component
  24   do 22 ii=1,iu
        rleg(ii,ii)=poly(0,ii)
        iptr=0
        do 25 j=1,ifini
            iptr=iptr+2
            jind=(j*2)-1
            do 22 ii=1,iu
                rleg(iptr,ii)=poly(jind,ii)
        22 continue
        rleg(iptr+1,ii)=poly(jind+1,ii)
  25 continue
write(6,'$S$$$ orthonomal setup done $$S$$

C............................
C...import the transfer function matrix
C............................
Cxxxxx.xxx
read(5,933)file
filex*pre//fileg
open(20,file=filex,status='old')
do 43 i=1,iu
   read(20,*,end=433) (tvtfm(i,j),j=1,iu)
43 continue
433 close(20)

C........calc b(k) using matrix
C........approach
C...........
write(6,'$S$'........calculating b(k).......'
C******use transpose matrix
   iz=(2*infini)+1
   do 10 k=1,iz
      do 9 ic=1,iu
         sum=0.
         do 8 ir=1,iu
            sum=sum+(rleg(k,iz)*tvtfm(ir,ic))
   8 continue
   b(k,ic)=sum
   9 continue
10 continue
C........calculating c(k)+++++
write(6,'$S$'+++++calculating c(k)+++++
   do 21 kkk=1,iu
21 c(1,kkk)=b(1,kkk)
   do 23 m=1,iz
23 a(m,m)=1.
C......calculate Gram-Schmidt coefficients
do 50 m=2,iz
   do 40 j=1,m-1
50 continue
   do 30 kkk=1,iu
      rp1(kkk)=c(j,kkk)
      rp2(kkk)=rp1(kkk)
      call inprod
      rn=norm-rip
   30 continue
   do 29 kkk=1,iu
      rp1(kkk)=c(j,kkk)
29 continue
   do 29 kkk=1,iu
      rp1(kkk)=c(j,kkk)
      call inprod
a(m,j)=1.*(rip/rnorm)

continue

c....now calculate c(m) by G-S orthogonalization
do 46 li=1,lu
46 sumc(li)=0.
   do 49 j=1,m-1
   do 48 kkk=1,li
      sumc(kkk)=sumc(kkk)+a(m,j)*c(j,kkk)
   continue
49 continue
   do 47 kkk=1,li
      c(m,kkk)=b(m,kkk)+sumc(kkk)
50 continue

c????????????????????????????????????????????????????????????????????????????

continue

c....normalized
do 720 m=1,iz
    do 712 kkk=1,lu
       rp1(kkk)=c(m,kkk)
712 call inprod
    cnorm(m)=sqrt(rip)
    do 715 kkk=1,lu
       c(m,kkk)=(c(m,kkk))/sqrt(rip)
715 continue

continue

write(6,'(***calculate tau=***')
   write(6,'(***=calcuting tau=***)
   do 75 k=1,iz
    tau(k,k)=a(k,k)
75 continue
   continue
90 continue

continue

c normalize taus by dividing by cnorms
   do 94 m=1,iz
      do 92 i=1,iz
         tau(m,i)=(tau(m,i)/cnorm(m))
94 continue
94 continue
c........................................................................................................
c!!!!!!!!!!!!!!!!!!!!!!!!calculate d(iz,iu)!!!!!!!!!!!!
   write(6,*)'!!!!!!!!calculating d(iz,iu)!!!!!'
   do 110 m=1,iz
do 105 i=1,m
do 100 it=1,iu
   d(m,it)=d(m,it)+(tau(m,i)*rleg(i,it))
100 continue
105 continue
110 continue

c........................................................................................................
filen=pre//d//ntrial//filen(5:6)
   * /fileg(5:6)//suf2
filen=pre//ntrial//filen(5:6)
   * /fileg(5:6)//suf3
filein=pre//filen(1:10)
fileg=pre//fileg(1:10)
c.................................................................

c.................................................................
   open(19,file=filein,status='old')
do 32 i=1,iu
read(19,*,end=519)ri(i)
32 continue
519 close(19)

izk=npol
c...clear arrays
do 190 iy=1,izk
190 f(iy)=0.
do 191 iy=1,izk
191 h(iy)=0.
c........................................................................................
compute f........................................................................................
do 120 m=1,izk
rsum=0.
do 115 i=1,iu
   rsum=rsum+(d(m,i)*ri(i))
115 continue
f(m)=rsum
   write(6,*)m,'c(m)='f(m)
120 continue
c........................................................................................
c.................................................................
c.................................................................
do 140 i=1,iu
rsum=0.
do 130 m=1,izk
   rsum=rsum+(c(m,i)*f(m))
130 continue
h(i)=x(i)
140 continue
$\ldots$
open(18,file=filex,status='new')
do 150 i=1,liu
  write(18,*)h(i)
close(18)

777 continue
write(6,*)filex
$\ldots$
write(25,*)'iu= ',iu
close(25)
c......
c: this section allows various process vectors and matrices to
be saved to disk for later analysis
$\ldots$
write(6,*)'write taufile (y/n)? '
read(5,976)cxans
976 format(a10)
  if(cxans.eq.'y')then
  write(6,*)'enter name of taufile '
read(5,977)taunn
977 format(a10)
  taunnx=pre//taunn
open(27,file=taunnx,status='new')
write(6,*)'iz= ',iz
do 833 m=1,iz
do 833 i=1,iz
write(27,*)tau(m,i)
833 continue
close(27)
endif
888 write(6,*)'save to disk d(m) array ? '
read(5,916)cyans
916 format(a10)
  if(cyans.eq.'y')then
556 write(6,*)'input name of otpt file xxxxxx.xxx '
read(5,933)obas
obasx=pre//obas
open(30,file=obasx,status='new')
do 501 ibas=1,iz
do 501 ku=1,iu
501 write(30,*)d(ibas,ku)
close(30)
endif
write(6,*)'save to disk c(m) array ? '
read(5,916)cyans
if(cyans.eq.'y') then
    write(6,'(a,e16.6)') input name of otpt file xxxxx.xxx
    read(5,933) obas
    obasx=pre//obas
    open(30,file=obasx,status='new')
    do 503 ibas=1,iz
      do 503 ku=1,iu

503  write(30,*) c(ibas,ku)
    close(30)
  endif
  write(6,'(a)') ' save to disk b(m) array ? '
  read(5,916) cyans
  if(cyans.eq.'y') then
    write(6,'(a,e16.6)') 'iz=
    write(6,'(a,e16.6)') input name of otpt file xxxxx.xxx
    read(5,933) obas
    obasx=pre//obas
    open(31,file=obasx,status='new')
    do 504 ibas=1,iz
      do 504 ku=1,iu

504  write(31,*) b(ibas,ku)
    close(31)
  endif

C>-------------------------------------------------------------------------
  stop
end
APPENDIX C

c TVFPAT.FOR
c FREQ DOMAIN DOBR THAT TAKES IDFT FOR TIME DOMAIN INV SOLN

c ....as most calculations involve complex operations, default implicit complex(c)

parameter(np=128,izz=128,dcval=11.3137,xval=5.6568,np2=128)
c ....np is the number of time and frequency domain signal points
c ....izz is the maximum number of basis functions
c ....dcval and xval are used to assign the frequency domain

c .... attributes of the Hartley Basis Set

cbn(izz,np),ccn(izz,np),csumc(np),ctruel(np)
complex cdn(izz,np),cp(izz,np),ca(izz,izz),ctau(izz,izz)
  complex cnorm(izz),cf(izz),ch(np),cfwd(np),cfwdn(np)
  real atrue(np2),arec(np2),xo(np)
c....Freq. domain variable names correspond to those used in
c ....time domain algorithm

cbn(m,k)-->b(m,k), ccn(m,k)--->c(m,k),
cdn(m,k)-->d(m,k)
c....the hartley basis is in cp(m,k)
c....gram-schmidt coeff are in array ca, tau in ctau
c....fwd soln-->cfwd. Freq domain inv soln-->ch,
c....time domain inv soln-->xo

character*10 tfnm,fnm,tnam,onnm
character*13 pre
character*30 tfmnx,fnmx,tnamx,onnmx

pre="c:\fort\data\"
900 format(a10)

c ....select the DOBR bandwidth
write(6,*)' input fmax '
read(5,*)ifmax

c ....and compute the number of Hartley Basis vectors to be used
npol=2*ifmax+1

c .... input the filename for the Fourier transforms of the vector

c ....set b(m,k). These are most easily obtained by dumping the
c ....vector set b(m,k) computed using time-domain DOBR and doing

c ....FFT of each member
write(6,*)' input TV B(m,k) file (xxxxxx.fcx) '
read(5,900)tfnm
tfnmx=pre//tfnm

c ....The FFT of the forward solution is also needed
write(6,*)' input FWD soln file (xxxxxx.fcx) '
read(5,900)tfnm

1
fnmx=pre//fnm
open(19,file=fnmx,status='old')
c...cfpwcr computes running sum of fwd soln pwr
cfpwcr=[0.,0.]
do 3 i=1,npj
read(19,*)cfdl(id)
cfpwcr(cfpwcr(cfdl(i))*conjg(cfdl(i)))
3 continue
close(19)
open(20,file=fnmx,status='old')
do 6 m=1,npd
do 5 i=1,npj
read(20,*)cbn(m,i)
5 continue
6 continue
close(20)

c...assign Freq. Domain Hartley basis set P(m,k) based on 128 pt
signal
c....assign dc val
cp(1,1)=(dcval,0.)
c....assign spectra for even-indexed basis
do 10 i=2,npd
ibin1=(i/2)+1
ibin2=(npd-1)-(i/2)
cp(i,ibin1)=(xval,-xval)
cp(i,ibin2)=(xval,xval)
10 continue
c...and for odd indexed basis
do 15 i=3,npd,2
ibin1=((i-1)/2)+1
ibin2=(npd-1)-((i-1)/2)
cp(i,ibin1)=(xval,xval)
cp(i,ibin2)=(xval,-xval)
15 continue
c...note that this assignment assumed that the basis ordering of
haropd

c...was: 1=dc, 2=+1hz,3=-1hz,4=+2hz,5=-2hz....
c?????????????????????calc c(k)s????????????????????????????????????
npxx=ifmax
'write6,*'+++++calculating c(k)+++++'+ccn(1,1)=cbn(1,1)
do 21 kkk=1,npxx
ccn(1,(npd-kkk+1))=cbn(1,(npd-kkk+1))
21 ccn(1,(kkk+1))=cbn(1,(kkk+1))
do 23 m=1,npd
c(m,m)=(1.,0.)
c......calculate ca coef (complex gram-schmidt coef.)
do 50 m=2,npd
do 40 j=1,m-1
   csum2=(0.,0.)
   csum2=(ccn(j,1)*( conjg(ccn(j,1))))
do 30 kkk=1,np

   csum2=csum2+(ccn(j, (np2-kkk+1))*( conjg(ccn(j, (np2-kkk+1))))
   csum2=csum2+(ccn(j, kkk+1)*( conjg(ccn(j, kkk+1))))

30 continue
   cnormx=csum2
   c........save the normalization factor for each vector
   c(m)........
   cnorm(j)=csqrt(cnormx)
   c.....................
   csum2=(0.,0.)
   csum2=(ccn(j,1)*conjg(cbn(m,1))
do 29 kkk=1,np
   csum2=csum2+(ccn(j, kkk+1)*conjg(cbn(m, kkk+1)))

   csum2=csum2+(ccn(j, (np2-kkk+1))*conjg(cbn(m, np2-kkk+1)))
29 continue
   cip=csum2
   ca(m,j)=-1.*(cip/cnormx)

40 continue

46 do 46 ii=1,np
   csumc(ii)=(0.,0.)
do 49 j=1,m-1
   csumc(1)=csumc(1)+(ca(m,j)*ccn(j,1))
do 48 kkk=1, np
   csumc(kkk+1)=csumc(kkk+1)+(ca(m,j)*ccn(j, kkk+1))

48 csumc(np2-kkk+1)=csumc(np2-kkk+1)+(ca(m,j)*ccn(j, np2-kkk+1)))
49 continue

47 continue
   ccn(m,1)=cbn(m,1)+csumc(1)
do 47 kkk=1,np
   ccn(m, kkk+1)=cbn(m, kkk+1)+csumc(kkk+1)
47 continue

47 continue

50 continue

c...now normalize

c...need the final norm
   crip=(0.,0.)
   crip=crip+(ccn(npol,1)*(conjg(ccn(npol,1))))
do 712 kkk=1,np

15
crip=crip+(ccn(npol,kkk+1)*conjg(ccn(npol,kkk+1))))

crip=crip+(ccn(npol,(np2-kkk+1))*conjg(ccn(npol,(np2-kkk+1))))
712 continue
    cnorm(npol)=csgrt(crip)
    do 720 m=1,npol
    ccn(m,1)=(ccn(m,1))/(cnorm(m))
    do 715 kkk=1,npnx
    ccn(m,kkk+1)=(ccn(m,kkk+1))/(cnorm(m))
    ccn(m,(np2-kkk+1))=ccn(m,(np2-kkk+1))/cnorm(m)
715 continue
    c***************calculate taus***************
    write(6,'(4x,1X,5A)')
    '************calculating tau******'
    do 75 k=1,npol
    ctau(k,k)=ca(k,k)
    75 ctau(k,k)=ca(k,k)
    do 85 i=1,m-1
    if(i.ne.m)then
        crsum=(0.,0.)
    do 83 j=i,m-1
        crsum=crsum+(ca(m,j)*ctau(j,i))
    83 crsum=crsum+(ca(m,j)*ctau(j,i))
    continue
    ctau(m,i)=crsum
    else
        endif
    85 continue
90 continue
720 c normalize taus by dividing by cnorms
    do 94 m=1,npol
    do 92 i=1,npol
    ctau(m,i)=(ctau(m,i)/cnorm(m))
92 continue
94 continue
7***********calculate d(iz,np)***********
    write(6,'(4x,1X,5A)')
    '************calculating d(iz,np)******'
    do 110 m=1,npol
    do 105 i=1,m
105 fi=float(i)
    itest1=aint(fi/2.)
    itest2=anint(fi/2.)
110 c....test for even or odd and assign index
    if(itest1.eq.itest2)then
        ind1=(i/2)+1
        ind2=(np+1)-(i/2)
        cdn(m,ind1)=cdn(m,ind1)+(ctau(m,i)*cp(i,ind1))
        cdn(m,ind2)=cdn(m,ind2)+(ctau(m,i)*cp(i,ind2))
    else
        ind1=((i-1)/2)+1
        cdn(m,ind1)=cdn(m,ind1)+(ctau(m,i)*cp(i,ind1))
if(i.gt.1) then
  ind2=(np+1)-((i-1)/2)
  cdn(m,ind2)=cdn(m,ind2)+(ctau(m,i)*cp(i,ind2))
else
  endif
100 continue
105 continue
110 continue

! --------------------------------------------------------------!
! this section applies to lab experiments where the true        !
! inverse son is known. when unknown, a dummy file can be       !
! used and snr values computed may be ignored.                  !
! write(6,'(E15.6)' input 128 pt spectrum file of true signal  !
! read(5,900)tnam                                            !
! nmax=pre/tnam                                              !
! open(21,file=tnamx,status='old')                           !
! sigp=0.                                                    !
! do 160 i=1,np                                              !
!   read(21,*)cttrue(i)                                       !
!   atrue(i)=real(cttrue(i))*conjg(cttrue(i))                !
!   sigp=sigp+attrue(i)                                       !
! 160 continue                                               !
! ! --------------------------------------------------------------!
! do 260 i=1,np                                              !
!   cfwdn(i)=cfwd(i)                                          !
! do 120 m=1,npol                                            !
!   crsum=(0.,0.)                                            !
!   crsum=crsum+(cdn(m,1)*conjg(cfwdn(1)))                    !
! do 115 i=1,npxx                                            !
!     crsum=crsum+(cdn(m,i+1)*conjg(cfwdn[i+1]))               !
!     crsum=crsum+(cdn(m,(np2-i+1))*conjg(cfwdn(np2-i+1)))     !
! 115 continue                                               !
! cf(m)=crsum                                                !
! continue                                                  !
! do 140 i=1,npxx+1                                          !
!   crsum=(0.,0.)                                            !
! do 130 m=1,npol                                            !
!   crsum=crsum+(ccn(m,i)*cf(m))                             !
! 130 continue                                               !
! ch(i)=crsum                                                !
! continue                                                  !
! do 141 i=1,npxx                                            !
!   crsum=(0.,0.)                                            !
! do 131 m=1,npol                                            !
!   crsum=crsum+(ccn(m,(np2-i+1))*cf(m))
continue
ch(np2-i+1)=crsum
continue

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
crsnum=(0.,0.)
crsnum=crsum+(ch(1)*conjg(ch(1)))
arec(1)=real(ch(1)*conjg(ch(1)))
do 150 i=1,nppx
  crsum=crsum+(ch(i+1)*conjg(ch(i+1)))
  arec(i+1)=real(ch(i+1)*conjg(ch(i+1)))
  crsum=crsum+(ch(np2-i+1)*conjg(ch(np2-i+1)))
  arec(np2-i+1)=real(ch(np2-i+1)*conjg(ch(np2-i+1)))
150 continue
write(6,'*(') recovered total pwr=',crsum

snse=0.
cfred=cttrue(1)-ch(1)
snse=snse+real(cfred*conjg(cfred))
do 180 i=1,nppx
  cfred=cttrue(i+1)-ch(i+1)
  snse=snse+real(cfred*conjg(cfred))
  cfred=cttrue(np2-i+1)-ch(np2-i+1)
  snse=snse+real(cfred*conjg(cfred))
180 continue
snrout2=10.*alog10(sigg/snse)
write(6,'*(') snr2=' ',snrout2

..................
...obtain time-domain inverse soln by IDFT
write(6,'*(') perform idft? 1=Y *
read(5,'*')iift
if (iift.eq.1) then
write(6,'*(') do 248 i=nppx+1,np2=npxx

248 ch(i)=(0.,0.)
call fft(ch,128,7,1.0)
do 249 li=1,np
  xo(li)=real(ch(li))
write(6,'*(') input outname xxxxxx.xxx ,
  read(5,900)onm
  omx=pre//onm
  open(27,ofile=onm,status='new')
do 250 li=1,np
  write(27,'*')xo(li)
250 continue
else
  close(27)
endif
777   continue

    stop
     end
I claim:

1. A method of using discrete orthogonal basis to restore a signal and/or image system degraded by time and/or spatially varying transfer functions said system being of linear type and a representation by the equation \([ B \cdot o = i ]\) wherein \([ o ]\) is an original signal or image and \([ i ]\) is a degraded signal or image and \([ B ]\) is a system transfer function matrix, comprising:

   estimating in estimating means a signal-to-noise ratio for a restored system;

   selecting in selecting means a set of orthogonal basis set functions \( p_{m} \) where \( m = 1, 2, 3 \ldots M \) is the index for the chosen orthogonal basis set with \( M \) members of length \( k = 1, 2, 3 \ldots ) \) to provide a stable inverse solution based upon the estimated signal-to-noise ratio; and

   removing in removing means time and/or spatially varying distortions in the restored system by obtaining an inverse solution vector \( O_{k} \) for a one dimensional restoration wherein:

\[
\begin{align*}
\alpha_{k} = & \sum_{m=1}^{M} \sigma_{m} \left[ \frac{H_{m}}{\sum_{k=1}^{K} d_{m}(s)} \right] \\
\end{align*}
\]

wherein \( d_{m} \) is a vector set created by linear combinations of \( p_{m} \) a set of constants formed by linear combinations of \( a \), the standard Gram-Schmidt orthogonalization coefficients, and wherein:

\[
[b \cdot o = (B)^T * p_{m}]
\]

where \([ B ]^T \) is the transpose-complex conjugate of the matrix \([ B ]\) and \([ p_{m} ]\) is an \( M \) member orthogonal basis set;

\[
\begin{align*}
\tau_{m} = & \sqrt{\sum_{k=1}^{K} d_{m}(s)^2} m = i; \\
\tau_{m} = & 1.0 \sqrt{\sum_{k=1}^{K} \sigma_{m}(s)^2} m = i; m = 1, \ldots , m = 1; \\
C_{m} = & \sum_{i=1}^{m} \tau_{m}(s) \\
d_{m} = & \sum_{i=1}^{m} \tau_{m}(s) m = 1, 2, 3 \ldots M
\end{align*}
\]

and an inverse solution vector \( O_{pk} \) for a two dimensional restoration wherein:

\[
\begin{align*}
o_{pk} = & \sum_{m=1}^{M} c_{m} \left[ \frac{H_{m}}{\sum_{k=1}^{K} d_{m}(s)} \right] \\
\end{align*}
\]

wherein \( J_{pk} \) is an intermediary matrix of the form:

\[
J_{pk} = \sum_{m=1}^{M} c_{m} \left[ \frac{H_{m}}{\sum_{k=1}^{K} d_{m}(s)} \right] \\
p = 1, 2, 3, \ldots , K
\]

2. The method set forth in claim 1, wherein said estimating of the signal-to-noise ratio of the restored system is provided by applying a given forward solution signal-to-noise ratio and selected set of orthogonal basis set functions \( p_{m} \) to a realistic simulation model of said system.

3. The method set forth in claim 1, wherein said set of orthogonal basis set functions \( p_{m} \) is selected from a group consisting of Hartley, Walsh, Haar, Legendre, Jacobi, Chebyshev, Gegenbauer, Hermite and Laguerre functions.

4. The method set forth in claim 2, wherein said set of orthogonal basis set functions \( p_{m} \) is selected from a group consisting of Hartley, Walsh, Haar, Legendre, Jacobi, Chebyshev, Gegenbauer, Hermite and Laguerre functions.

5. A method of using discrete orthogonal basis to restore a signal system degraded by a time varying transfer function, comprising:

   estimating in estimating means a signal-to-noise ratio for a restored system;

   selecting in selecting means a set of orthogonal basis set functions \( p_{m} \) where \( m = 1, 2, 3 \ldots M \) is the index for the chosen orthogonal basis set with \( M \) members of length \( k = 1, 2, 3 \ldots ) \) to provide a stable inverse solution based upon the estimated signal-to-noise ratio; and

   removing in removing means time and/or spatially varying distortions in the restored system by obtaining an inverse solution vector \( O_{k} \) wherein:

\[
\begin{align*}
\alpha_{k} = & \sum_{m=1}^{M} \sigma_{m} \left[ \frac{H_{m}}{\sum_{k=1}^{K} d_{m}(s)} \right] \\
\end{align*}
\]

wherein \( d_{m} \) is a vector set created by linear combinations of \( p_{m} \) a set of constants formed by linear combinations of \( a \), the standard Gram-Schmidt orthogonalization coefficients, and wherein:

\[
[b \cdot o = (B)^T * p_{m}]
\]

where \([ B ]^T \) is the transpose-complex conjugate of the matrix \([ B ]\) and \([ p_{m} ]\) is an \( M \) member orthogonal basis set;

\[
\begin{align*}
\tau_{m} = & \sqrt{\sum_{k=1}^{K} d_{m}(s)^2} m = i; \\
\tau_{m} = & 1.0 \sqrt{\sum_{k=1}^{K} \sigma_{m}(s)^2} m = i; m = 1, \ldots , m = 1; \\
C_{m} = & \sum_{i=1}^{m} \tau_{m}(s) \\
d_{m} = & \sum_{i=1}^{m} \tau_{m}(s) m = 1, 2, 3 \ldots M
\end{align*}
\]

6. The method set forth in claim 5 wherein said estimating of the signal-to-noise ratio \( \text{SNR}_{pred} \) is provided by:

\[
\text{SNR}_{pred} = 10 \log \frac{L_{o,original}^2}{L_{o,original}^2 + L_{o,noise}^2}
\]

wherein \( L_{o,original}^2 \) is the signal power in the original, undegraded signal/image, \( L_{o,noise}^2 \) is the inverse solution noise power due to the approximate nature of the inverse solution and \( L_{o,noise}^2 \) is the noise power in the inverse solution power due to added noise in the forward solution.

7. The method set forth in claim 6, wherein said set of orthogonal basis set functions \( p_{m} \) is selected from a group consisting of Hartley, Walsh, Haar, Legendre, Jacobi, Chebyshev, Gegenbauer, Hermite and Laguerre functions.

8. The method set forth in claim 5, wherein said set of orthogonal basis set functions \( p_{m} \) is selected from a group consisting of Hartley, Walsh, Haar, Legendre, Jacobi, Chebyshev, Gegenbauer, Hermite and Laguerre functions.
9. A method of using discrete orthogonal basis to restore an image system degraded by time and spatially varying transfer functions, comprising:

estimating in estimating means a signal-to-noise ratio for a restored system;

selecting in selecting means a set of orthogonal basis set functions \( p_{n\kappa} \) to provide a stable inverse solution based upon the estimated signal-to-noise ratio; and

removing in removing means time and spatially varying distortions in the restored system by obtaining an inverse solution vector \( O_{p\kappa} \) wherein:

\[
O_{p\kappa} = \sum_{m=1}^{M} c_{p\kappa m} \sum_{p=1}^{U} d_{p\kappa p} j_{p}\kappa
\]

\( \kappa = 1, 2, 3, \ldots, U \)

wherein \( d_{p\kappa} \) is a vector set created by linear combinations of \( p_{n\kappa} \) weighted by \( \tau_{n\kappa} \), a set of constants formed by linear combinations of \( a \), the standard Gram-Schmidt orthogonalization coefficients, and \( J_{p\kappa} \) is an intermediary matrix of the form:

\[
J_{p\kappa} = \sum_{m=1}^{M} c_{p\kappa m} \begin{bmatrix}
\sum_{k=1}^{U} d_{p\kappa p} j_{k}\kappa
\end{bmatrix}
\]

\( \rho = 1, 2, 3, \ldots, U \).

10. The method set forth in claim 9, wherein said estimating of the signal-to-noise ratio \( SNR_{p\kappa} \) is provided by simulating a noiseless forward solution and determining the intrinsic noise shown as the difference between the restored image and the original image and further considering recovered added noise.

11. The method set forth in claim 10, wherein said estimated signal-to-noise ratio \( SNR_{p\kappa} \) is

\[
SNR_{p\kappa} = 10 \log \frac{V_{signal}^2}{V_{noise}^2 + V_{noise}^2}
\]

wherein \( V_{signal}^2 \) is the signal power in the original, undegraded signal/image, \( V_{noise}^2 \) is the inverse solution noise power due to the approximate nature of the inverse solution and \( V_{noise}^2 \) is the noise power in the inverse solution due to added noise in the forward solution.

12. The method set forth in claim 9, wherein said estimated signal-to-noise ratio \( SNR_{p\kappa} \) is

\[
SNR_{p\kappa} = 10 \log \frac{V_{signal}^2}{V_{signal}^2 + V_{noise}^2}
\]

wherein \( V_{signal}^2 \) is the signal power in the original, undegraded signal/image, \( V_{noise}^2 \) is the inverse solution noise power due to the approximate nature of the inverse solution and \( V_{noise}^2 \) is the noise power in the inverse solution due to added noise in the forward solution.

13. A programmable apparatus, comprising:

means for computing; and

readable memory defining a process for estimating a signal-to-noise ratio for a restored system; selecting a set of orthogonal basis set functions \( P_{\kappa m} \) wherein \( m = 1, 2, 3, \ldots, M \) is the index for the chosen orthogonal basis set with \( M \) members of length \( k = 1, 2, 3, \ldots, U \) to provide a stable inverse solution based upon the estimated signal-to-noise ratio; and

removing time varying distortions in the restored system by obtaining an inverse solution vector \( O_{p\kappa} \) for a one dimensional restoration wherein:

\[
o_{\kappa} = \sum_{m=1}^{M} c_{\kappa m} \begin{bmatrix}
\frac{V_{U}}{k_{1}} \cdot d_{\kappa k_{1}}
\end{bmatrix}
\]

wherein \( d_{\kappa k_{1}} \) is a vector set created by linear combinations of \( P_{\kappa m} \) weighted by \( \tau_{\kappa m} \), a set of constants formed by linear combinations of \( a \), the standard Gram-Schmidt orthogonalization coefficients, and

\[
[B]^{*} = \text{transpose-complex conjugate of matrix}[B] \quad [p_{\kappa}]_{m}
\]

where \( [B]^{*} \) is the transpose-complex conjugate of the matrix \( [B] \) and \( [p_{\kappa}]_{m} \) is an \( M \) member orthogonal basis set;

\[
\tau_{\kappa m} = \begin{bmatrix}
\omega_{\kappa m} \cdot m = i;
\sum_{k=1}^{U} \omega_{\kappa m} \cdot k = 1, \ldots, m
\end{bmatrix}
\]

\[
\omega_{\kappa m} = \begin{bmatrix}
\frac{1}{m} \cdot \sum_{k=1}^{m-1} \omega_{\kappa m} \cdot k = i = 1, \ldots, m - 1;
\sum_{k=1}^{U} \omega_{\kappa m} \cdot k = 1, \ldots, m
\end{bmatrix}
\]

\[
c_{\kappa m} = \sum_{n=1}^{m} \omega_{\kappa m} \cdot n
\]

\[
d_{\kappa m} = \sum_{n=1}^{m} \omega_{\kappa m} \cdot n
\]

and an inverse solution vector \( O_{p\kappa} \) for a two dimensional restoration wherein:

\[
O_{p\kappa} = \sum_{m=1}^{M} c_{p\kappa m} \sum_{p=1}^{U} d_{p\kappa p} j_{p}\kappa
\]

\( \kappa = 1, 2, 3, \ldots, U \).

14. A programmable apparatus, comprising:

means for computing; and

readable memory defining a process for estimating a signal-to-noise ratio for a restored system; selecting a set of orthogonal basis set functions \( P_{\kappa m} \) wherein \( m = 1, 2, 3, \ldots, M \) is the index for the chosen orthogonal basis set with \( M \) members of length \( k = 1, 2, 3, \ldots, U \) to provide a stable inverse solution based upon the estimated signal-to-noise ratio; and

removing time varying distortions in the restored system by obtaining an inverse solution vector \( o_{\kappa} \) wherein:

\[
o_{\kappa} = \sum_{m=1}^{M} c_{\kappa m} \begin{bmatrix}
\frac{V_{U}}{k_{1}} \cdot d_{\kappa k_{1}}
\end{bmatrix}
\]

wherein \( d \) is a vector set created by linear combinations of \( P_{\kappa m} \) weighted by \( \tau_{\kappa m} \), a set of constants formed by linear combinations of \( a \), the standard Gram-Schmidt orthogonalization coefficients, and

\[
[B]^{*} = \text{transpose-complex conjugate of matrix}[B] \quad [p_{\kappa}]_{m}
\]
where \([B]^T\) is the transpose-complex conjugate of the matrix \([B]\) and \([p]_{lm}\) is an \(M\) member orthogonal basis set:

\[
t_m = \frac{a_{mk}}{\sqrt{\frac{1}{M} \sum_{k=1}^{M} |a_{mk}|^2}} \quad m = i;
\]

\[
t_m = \frac{1.0}{\sqrt{\frac{1}{M} \sum_{k=1}^{M} |a_{mk}|^2}} \quad m \neq i; \quad i = 1, \ldots, m - 1;
\]

\[
c_{mk} = \frac{1}{M} \sum_{i=1}^{M} a_{mk} p_{mi};
\]

\[
d_{mk} = \frac{1}{M} \sum_{i=1}^{M} a_{mk} p_{mi}; \quad m = 1, 2, 3 \ldots M.
\]

15. A programmable apparatus, comprising:

means for comparing; and

readable memory defining a process for estimating means a signal-to-noise ratio for a restored system;

selecting a set of orthogonal basis set functions \(p_{mk}\) to provide a stable inverse solution based upon the estimated signal-to-noise ratio; and

removing time and spatially varying distortions in the restored system by obtaining an inverse solution vector \(O_k\) wherein:

\[
O_k = \frac{\sum_{m=1}^{M} C_{mk}^{-1} \sum_{i=1}^{N} D_{mk}^{-1} s_i}{N}
\]

wherein \(s_i\) is a fourier transform of the forward solution and \(O_k\) is a fourier transform of the inverse solution and a time domain solution may be obtained by

\[
F^{-1}(O_k) = o_k
\]

where

\[
[\theta]_{lm} = [F]^T \cdot [P]_{lm}
\]

and \([F]=DFT\) \([IDFT]=\)

inverse fourier transform matrices.

\[
a_{mk} = \frac{\sum_{k=1}^{N} C_{mk} A_{mk}}{\sum_{k=1}^{N} C_{mk} C_{mk}^*}
\]

\(i = 1, \ldots, m - 1\)

\(m = 1, \ldots, M\)

wherein \(a\) is the complex Gram-Schmidt coefficient and vector sets \(C\) and \(A\) define the characteristics of the system \([B]\) for frequency domain discrete orthogonal basis restoration;

\[
I_{mk} = \frac{1.0}{\sqrt{\frac{1}{N} \sum_{k=1}^{N} C_{mk} C_{mk}^*}}
\]

\(m = i\)

\[
t_m = \frac{1.0}{\sqrt{\frac{1}{N} \sum_{k=1}^{N} C_{mk} C_{mk}^*}} \quad m = 1, \ldots, M
\]

\(i = 1, \ldots, m - 1\)

\(m = 1, \ldots, M\)

where \(\tau\) is a constant formed by linear combinations of \(a\);

\[
C_{mk} = \frac{1}{\sum_{i=1}^{M} t_{mk}} \quad t_{mk}
\]

\(m = 1, 2, 3 \ldots M\)

16. A method of using discrete orthogonal basis to restore a signal and/or image system degraded by time and/or spatially varying transfer functions said system being of linear type and represented by the equation \([B] [o] = [i]\) wherein \([o]\) and \([i]\) are length \(N\) column and row vectors and \([B]\) is an \(N \times N\) non-singular transfer function matrix, comprising:

estimating in estimating means a signal-to-noise ratio for a restored system;

selecting in selecting means a set of orthogonal basis set functions \(P_{mk}\) where \(m = 1, 2, 3 \ldots M\) is the index for the chosen orthogonal basis set with \(M\) members of length \(k = 1, 2, 3 \ldots N\) to provide a stable inverse solution based upon the estimated signal-to-noise ratio; and

removing in removing means time and/or spatially varying distortions in the restored system by obtaining an inverse solution vector \(O_k\) for a one dimensional restoration wherein:
removing time and/or spatially varying distortions in the restored system by obtaining an inverse solution vector $O_k$ for a one dimensional restoration wherein:

$$O_k = \sum_{m=1}^{M} C_{ab} \frac{1}{N} \sum_{k=1}^{N} D_{ab}^{*}$$

wherein $O_k$ is a Fourier transform of the forward solution and $O_k$ is a Fourier transform of the inverse solution and a time domain solution may be obtained by

$$\mathcal{F}^{-1}(O_k) = o_k$$

where

$$[B] = [F]^T [F]$$

where

$$[F] = \text{DFT}([\theta]) \text{IDFT}$$

and [DFT]-discrete Fourier transform and [IDFT]-inverse Fourier transform matrices.

$$a. \quad \frac{1}{N} \sum_{k=1}^{N} C_{ab} A_{i,k}$$

$$i = 1, \ldots m - 1$$

$$b. \quad \tau_{ab} = \frac{1.0}{\sum_{k=1}^{N} C_{ab} C_{ab}^{*}}$$

$$m = i$$

$$\tau_{ab} = \frac{1.0}{\sum_{k=1}^{N} C_{ab} C_{ab}^{*}}$$

$$i = 1, \ldots m - 1$$

$$m = 1, \ldots M$$

where $\tau$ is a constant formed by linear combinations of $a$;

$$c. \quad C_{ab} = \sum_{i=1}^{m} \tau_{ab} a_i$$

and

$$d. \quad D_{ab} = \sum_{i=1}^{m} \tau_{ab} a_{i}$$

$$\ast \ast \ast \ast$$