On the Use of Lyapunov Criteria to Analyze the Convergence of Blind Deconvolution Algorithms

Deepa Kundur, *Student Member, IEEE*, and Dimitrios Hatzinakos, *Member, IEEE*

***Abstract*— We present an approach to determine sufficient conditions for the global convergence of iterative blind deconvolution algorithms using finite impulse response (FIR) deconvolution filters. The novel technique, which incorporates Lyapunov's direct method, is general, flexible, and can be easily adapted to analyze the behavior of many types of nonlinear iterative signal processing algorithms. Specifically, we find sufficient conditions to guarantee a unique solution for the NAS-RIF algorithm used for blind image restoration. We determine that in many cases, there exists a tradeoff between the quality of the deconvolution result and the uniqueness of the solution. A procedure to determine the length of the deconvolution filter to guarantee a unique solution is established.**

I. Introduction

IN APPLICATIONS such as image processing, digital communications, and seismic data analysis among others, there is a need for efficient blind deconvolution algorithms. Blind deconvolution refers to the problem of separating two signals f and h from their convolutional product $g = f * h$, when both signals are only partially known. In most applications, f is the signal to be restored, h is the unknown degradation, and g is the degraded signal.

Many existing iterative blind deconvolution methods make use of nonlinear transformations on the degraded signal to deconvolve the data. The existence of the nonlinearities makes analysis of the convergence and uniqueness properties of the algorithms difficult. In this paper, we present a general framework based on Lyapunov's direct method for studying the behavior of nonlinear blind deconvolution algorithms. The framework we use applies to many blind deconvolution schemes, such as the Bussgang class of blind equalization methods [1] and the nonnegativity and support constraints recursive inverse filtering (NAS-RIF) algorithm for blind image restoration [2]. These methods belong the to the class of zero memory nonlinearity deconvolution techniques and are popular for their low computational complexity and flexibility in performing blind deconvolution. Fig. 1 presents the general architecture, which we discuss later. Each technique of the class differs in the nonlinearity NL imposed during deconvolution. We specifically discuss and concentrate on the NAS-RIF algorithm [3] for blind image restoration. Convexity

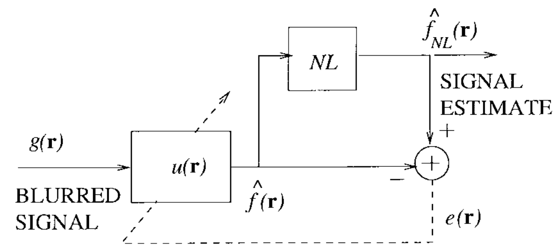

Fig. 1. Zero-memory nonlinearity deconvolution. Blind deconvolution is performed by recursively inverse FIR filtering the degraded signal g with the adaptive filter u to produce an estimate of the true signal $\hat{f}$. This estimate is passed through a zero-memory nonlinearity to produce a signal $\hat{f}_{\mathrm{NL}}$, which is considered to be an even better estimate of the true signal. The difference between $\hat{f}$ and $\hat{f}_{\mathrm{NL}}$ is used to update u for the next iteration.

of the cost function minimized in the NAS-RIF algorithm has been established in [3]. In this paper, we go beyond our previous results and provide an analysis framework to determine sufficient conditions to guarantee a unique solution.

In the next section, we introduce some fundamental concepts used in our analysis, give a brief discussion of Lyapunov's direct method, and discuss zero memory nonlinearity deconvolution. In Section III, we provide a method to analyze the convergence properties of blind deconvolution methods using finite extent [i.e., finite impulse response (FIR)] deconvolution filters; we use the framework to find sufficient conditions to ensure a unique solution for the NAS-RIF algorithm. Concluding remarks are provided in Section IV.

II. Background

A. Algorithm Convergence

In this section, we address some related concepts concerning the issue of algorithm convergence and uniqueness of solution for the blind deconvolution process. The general blind deconvolution problem is said to have a unique solution if there exists only one pair of signals f and h that produce the result $g = f * h$, where $*$ is the linear convolution operator. It is easy to see that the solution is unique if both components f and h are considered to be irreducible.[1] Ideally, the deconvolution process may be implemented by using an infinite extent deconvolution filter to perform the restoration of f given g.

[1] The term *irreducible* refers to a signal that cannot be expressed as the convolution of two or more component images of finite support on the understanding that the delta function is not a component image.

Manuscript received September 15, 1997; revised April 20. 1998. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The associate editor coordinating the review of this paper and approving it for publication was Prof Peter C. Doerschuk.

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ont., Canada M5S 3G4 (e-mail: deepa@comm.toronto.edu; dimitris@comm.toronto.edu).

Publisher Item Identifier S 1053-587X(98)07802-7.

1053–587X/98$10.00 © 1998 IEEE

However, in practice, FIR deconvolution filters are used for processing. Therefore, we must settle for obtaining a solution that is an approximation to the signal f. We call this approximation a *desired solution*. Given an FIR deconvolution filter, the objective of an iterative blind deconvolution method is to converge to a desired solution from an arbitrary initial condition. Two issues arise concerning the convergence of the algorithm: *global convergence* and *uniqueness of solution*.

An iterative blind deconvolution algorithm can be considered to be equivalent to the minimization of an associated cost function J. In this context, global convergence refers to the ability of the algorithm to reach a global minimum of the cost function from any initial condition. Uniqueness of the solution refers to the existence of a single global minimum. Both of these properties are related to the convexity of the function J. If a function is convex, then we can ensure convergence to a global minimum using numerical descent routines. If a function is strictly convex, then we can additionally guarantee that this desired global minimum is unique [4].

B. Algorithm Stability

The convergence of a signal processing algorithm is also related to its stability. We focus on discrete-time algorithms described by a recursive time domain relationship of the form

$$\mathbf{u}^{k+1} = \mathbf{F}_k(\mathbf{u}^k) \tag{1}$$

where $\mathbf{u}^k \in \mathbb{R}^n$, and $\mathbf{F}_k : \mathbb{R}^n \to \mathbb{R}^n$ for all $k \geq 0$. In most iterative signal processing algorithms, we want to find a $\bar{\mathbf{u}}$ that is invariant under the mapping $\mathbf{F}_k$ [i.e., $\bar{\mathbf{u}} = \mathbf{F}_k(\bar{\mathbf{u}})$]. Such a parameter set is called an *equilibrium solution* of (1) [5]. Given an initial condition $\mathbf{u}_0$, we are often concerned with whether or not the recursion will lead to an equilibrium solution $\bar{\mathbf{u}}$. Furthermore, we would like the recursion to converge to $\bar{\mathbf{u}}$ given any $\mathbf{u}_0$. An algorithm that exhibits this attribute is called *globally asymptotically stable* [5]; global convergence of the algorithm to a unique solution is ensured. Many iterative algorithms, such as the set of zero-memory nonlinear deconvolution methods discussed in Section II-D, are in the form of (1).

C. Lyapunov's Direct Method

Lyapunov's direct method can be used to provide sufficient conditions for the asymptotic stability of a given nonlinear recursion of the form of (1) [5]. Lyapunov analysis entails the selection of an "energy" function commonly referred to as a Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$, which maps the parameters of a given nonlinear recursion to a scalar quantity. If a function V can be found that exhibits certain properties, which we state in Theorem 1, then the recursive algorithm is globally asymptotically stable. We make use of the following theorem [5].

Theorem 1 (Global Asymptotic Stability): The equilibrium $\bar{\mathbf{u}}$ of (1) is globally asymptotically stable if there is a function $V : \mathbb{R}^n \to \mathbb{R}$ such that

1) $V(\bar{\mathbf{u}}) = 0$;

2) there are continuous, strictly increasing functions $\alpha : \mathbb{R} \to \mathbb{R}$, and $\beta : \mathbb{R} \to \mathbb{R}$, where $\alpha(0) = \beta(0) = 0$ and $\alpha(||\mathbf{u} - \bar{\mathbf{u}}||) \leq V(\mathbf{u}) \leq \beta(||\mathbf{u} - \bar{\mathbf{u}}||)$ for all $\mathbf{u} \in \mathbb{R}^n$;

3) V is radially unbounded, i.e., $V(\mathbf{u}) \to \infty$ as $||u|| \to \infty$;

4) $\Delta V_k \triangleq V(\mathbf{u}^{k+1}) - V(\mathbf{u}^k) < 0$ for all $k \geq 0$;

where $||\cdot||$ is the standard Euclidean norm.

By selecting an appropriate Lyapunov function candidate V, we can determine the conditions that ensure that properties (1)–(4) of Theorem 1 hold. These conditions will be sufficient for global asymptotic stability of the algorithm and will therefore ensure that there is convergence to a unique solution $\bar{\mathbf{u}}$ for any initialization.

D. Zero-Memory Nonlinearity Deconvolution Methods

Several blind deconvolution algorithms fall under the category of zero-memory nonlinearity deconvolution methods [1]. We mentioned two such methods—the Bussgang class of techniques and the NAS-RIF algorithm—in the introduction. These methods follow the architecture shown in Fig. 1 to restore the signal f. In such methods, deconvolution is performed by inverse filtering the degraded signal g with a FIR filter u. The output of this filter $\hat{f}$ is an estimate of the true signal f. This estimate is passed through a zero-memory nonlinearity to produce a signal $\hat{f}_{\rm NL}$, which is considered to be an even better estimate of f. The FIR deconvolution filter u is updated by trying to minimize the difference between $\hat{f}$ and $\hat{f}_{\rm NL}$. The associated cost function is given by

$$J(\mathbf{u}) = \sum_{\forall \mathbf{r}} [\hat{f}(\mathbf{r}) - \hat{f}_{\rm NL}(\mathbf{r})]^2 \tag{2}$$

where $\mathbf{u}$ is a column vector representing an ordered set of parameters of the FIR filter u,[2] $\mathbf{r}$ is the discrete signal index (it is scalar in the 1-D situation and a vector in the multiple dimension situation, e.g. for images), $\hat{f}(\mathbf{r}) = g(\mathbf{r}) * u(\mathbf{r})$, $\hat{f}_{\rm NL}(\mathbf{r}) = {\rm NL}\{\hat{f}(\mathbf{r})\}$, ${\rm NL}\{\cdot\}$ is the zero-memory nonlinearity, and $*$ represents the linear convolution operator. The corresponding update law for the deconvolution filter is given by

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \mu \nabla J(\mathbf{u}^k) \tag{3}$$

where

$\mathbf{u}^k$	vector of filter coefficients $u(\mathbf{r})$ at the kth iteration of the algorithm;
$\mu > 0$	update step size;
$\nabla J(\mathbf{u})$	gradient of J with respect to $\mathbf{u}$.

It is easy to see that (3) is in the same form as (1).

The distinct algorithms of this class differ in the zero-memory nonlinearity ${\rm NL}\{\cdot\}$ used for restoration. This function often depends on the prior knowledge available about the signal f; it could be in the form of statistical information for applications such as seismic analysis and data communications, or it may be purely deterministic as for image processing applications.

In this paper, we focus on the analysis of the non-negativity and support constraints recursive inverse filtering algorithm

[2] For the 2-D case, $\mathbf{u}$ is the lexicographically ordered (i.e., row-ordered) column vector of the filter u.

(NAS-RIF) algorithm for blind image restoration [3]. The method is applicable to situations in which one or more objects of known finite support are imaged against a uniformly grey background. The zero-memory nonlinearity is given by

$$\hat{f}_{\rm NL}(\mathbf{r}) = {\rm NL}\{\hat{f}(\mathbf{r})\} = \begin{cases} \hat{f}(\mathbf{r}), & \text{if } \hat{f}(\mathbf{r}) \geq 0 \text{ and } \mathbf{r} \in D_{\rm sup} \\ 0, & \text{if } \hat{f}(\mathbf{r}) < 0 \text{ and } \mathbf{r} \in D_{\rm sup} \\ L_B, & \text{if } \mathbf{r} \in \bar{D}_{\rm sup} \end{cases} \tag{4}$$

where

- $\mathbf{r}$ 2-D vector representing discrete image pixel coordinates;
- $D_{\rm sup}$ region of support, which is assumed to be known (and which we discuss below);
- $\bar{D}_{\rm sup}$ its complement;
- L_B pixel value of the background grey level.

The associated cost function reduces to

$$J(\mathbf{u}) = \sum_{\mathbf{r}\in D_{\rm sup}} \hat{f}^2(\mathbf{r}) u_s(-\hat{f}(\mathbf{r})) + \sum_{\mathbf{r}\in \bar{D}_{\rm sup}} [\hat{f}(\mathbf{r}) - L_B]^2 + \gamma\left[\sum_{\forall \mathbf{r}} u(\mathbf{r}) - 1\right]^2 \tag{5}$$

where $u_s(\cdot)$ is the unit step function, $\hat{f}(\mathbf{r}) = g(\mathbf{r}) * u(\mathbf{r})$, and the third term with parameter $\gamma > 0$ is used to avoid the trivial all-zero solution when $L_B = 0$, as explained in [3]. The corresponding update law is given by

$$u^{k+1}(\mathbf{i}) = u^k(\mathbf{i}) - 2\mu\left[\sum_{\mathbf{r}\in D_{\rm sup}} g(\mathbf{r}-\mathbf{i})\hat{f}^k(\mathbf{r})u_s(-\hat{f}^k(\mathbf{r})) + \sum_{\mathbf{r}\in \bar{D}_{\rm sup}} g(\mathbf{r}-\mathbf{i})(\hat{f}^k(\mathbf{r}) - L_B) + \gamma\left(\sum_{\forall \mathbf{r}} u^k(\mathbf{r}) - 1\right)\right] \tag{6}$$

where $u^k(\mathbf{i})$ is the ith element of the vector $\mathbf{u}^k$, and $f^k(\mathbf{r}) = g(\mathbf{r}) * u^k(\mathbf{r})$.

We consider situations in which the imaged scene is comprised of object(s) and a uniform background. We define the region of support $D_{\rm sup}$ as a set including all pixels encompassing the object(s) within the image frame. This region can be of any shape. For example, in [3], $D_{\rm sup}$ is defined as the smallest rectangle encompassing the object. $\bar{D}_{\rm sup}$ is the complement of $D_{\rm sup}$ (i.e., the set of all pixels not within the predefined region of support); the regions $D_{\rm sup}$ and $\bar{D}_{\rm sup}$ are disjoint. We give examples of possible kinds of supports in Fig. 2. Since the restored image $\hat{f}$ is computed by linearly convolving the blurred image g with u, $\hat{f}$ grows in size with the size of u. The edges of the image are extended during filtering process. The dimensions of $\hat{f}$ are $(N_{xg} + N_{xu} - 1) \times (N_{yg} + N_{yu} - 1)$, where $N_{xg} \times N_{yg}$ and $N_{xu} \times N_{yu}$ are the dimensions of g and u, respectively.

The NAS-RIF algorithm attempts to extract information concerning the frequencies attenuated due to blurring by using information concerning the support and non-negativity of the original data. A similar problem has been considered by Gerchberg [6] (and later by Papoulis [7]) for the frequency extrapolation of bandlimited functions with known finite support. The difference in the problem they consider is that the frequencies within the band limit of the degradation are assumed to be unaltered, whereas in the NAS-RIF algorithm, it is assumed that the attenuation of all frequencies is unknown, and we exploit the fact that the degradation is in the form of a spatial-domain linear shift invariant convolution that can be undone with inverse filtering. Wiley, in [8], shows that the projection-based algorithm used in [6] and [7] is a special case of generalized contraction mapping developed by Sandberg [9], which guarantees the convergence of the algorithm to a unique solution. Our algorithm is somewhat similar to the class of iterative algorithms considered by Sandberg, except that the projection operator is not necessarily one-to-one; thus, convergence and uniqueness issues cannot be easily addressed using the work in [9].

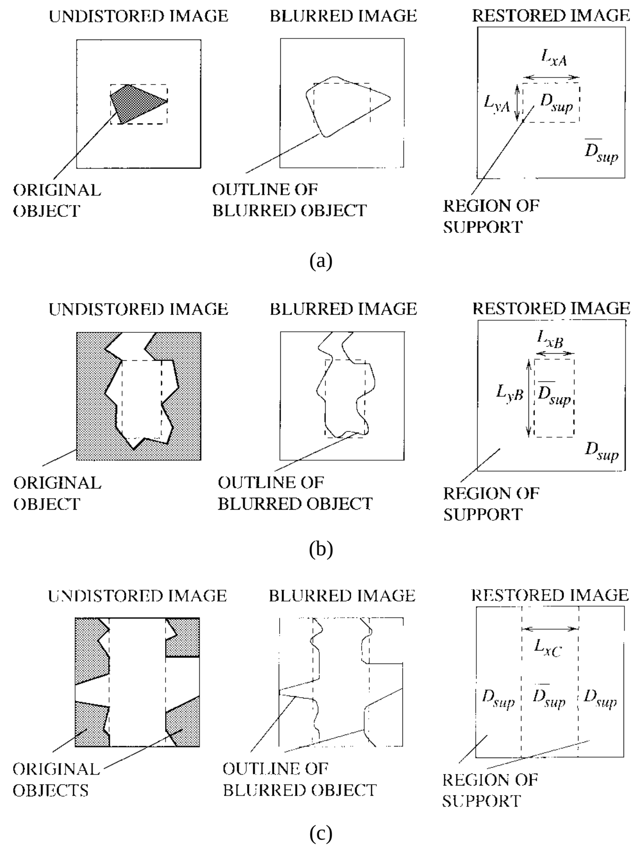

Fig. 2. Different regions of support encountered in image processing applications. Three different cases are shown. For each case, the undistorted image, blurred image, and restored image frames are drawn. Blurring causes the object(s) in the image frame to spread spatially; the outlines of the blurred objects are shown and subsequently overlap the $\bar{D}_{\rm sup}$ region due to the spreading of the object. The restored image is of a larger size than the blurred image as it is computed by linearly convolving the blurred image with the deconvolution filter u.

III. Analysis

A. General Methodology

In this section, we provide a systematic technique to find sufficient conditions to ensure the unique global convergence of an iterative blind deconvolution method using an FIR filter. We base our analysis on Theorem 1.

1) Select a Lyapunov function candidate.

   For example, we may select $V(\mathbf{u}) = J(\mathbf{u}) - J(\bar{\mathbf{u}})$, where J is the associated cost function of the algorithm, and $\bar{\mathbf{u}}$ is the desired equilibrium solution to the update law of (3) and is a global minimum of J.

2) Restrict $V(\mathbf{u})$ to the four conditions outlined in Theorem 1, and translate these to constraints on the algorithm parameters (e.g., nonlinearities, update gains, etc.).

   Continuing with the previous example, the conditions translate to the following constraints on $J(\mathbf{u})$:

   J1) $\alpha(\|\mathbf{u}-\bar{\mathbf{u}}\|) + J(\bar{\mathbf{u}}) \leq J(\mathbf{u}) \leq \beta(\|\mathbf{u}-\bar{\mathbf{u}}\|) + J(\bar{\mathbf{u}})$ for all $\mathbf{u} \in \mathbb{R}^n$, where $\alpha(\cdot)$ and $\beta(\cdot)$ are as stated in (1) of Theorem 1.

   J2) $J(\mathbf{u}) \to \infty$ as $\|\mathbf{u}\| \to \infty$.

   J3) $\Delta V_k = J(\mathbf{u}^{k+1}) - J(\mathbf{u}^k) < 0$ for all $k \geq 0$, where $\mathbf{u}^k \neq \bar{\mathbf{u}}$.

3) Form the function $\Delta V_k = V(\mathbf{u}^{k+1}) - V(\mathbf{u}^k)$ and substitute $\mathbf{u}^{k+1}$ with the right side of the update law of (3) to get ΔV_k solely in terms of $\mathbf{u}^k$.

   In our example, ΔV_k becomes

$$\begin{aligned}\Delta V_k &= V(\mathbf{u}^{k+1}) - V(\mathbf{u}^k)\\ &= J(\mathbf{u}^{k+1}) - J(\mathbf{u}^k)\\ &= J(\mathbf{u}^k - \mu\nabla J(\mathbf{u}^k)) - J(\mathbf{u}^k)\end{aligned} \tag{7}$$

4) Determine the conditions for which all the constraints obtained from steps 2 and 3 are fulfilled.

   From Theorem 1, we establish that these conditions are sufficient to ensure global convergence and a unique solution to the algorithm. If such conditions cannot be obtained or are not satisfactory, go back to Step 1, this time selecting a different Lyapunov function candidate.

   We continue this stage of the analysis on the example in the next section.

Our analysis approach is general in the sense that it can theoretically provide sufficient conditions for the global convergence to a unique solution of an algorithm with an update law of the form of (1). The drawback is that the approach works as long as an appropriate Lyapunov function can be found that satisfies the conditions stated in Theorem 1. It should be emphasized that the difficulty of the method lies in determination of the candidate Lyapunov function V. Selection of a good function can be a nontrivial task; experience makes the selection process easier. Vidyasagar [5] and Khalil [10] give some good examples; however, there exists no general procedure to determine a good Lyapunov function for a given problem.

B. Conditions to Ensure Global Convergence to a Unique Solution

From (7), we see that for our specific choice of candidate Lyapunov function, finding the conditions that ensure that the constraints J1) to J3) hold implies that the algorithm will converge globally to a unique solution under these conditions. To obtain these specific requirements, we make use of the following definition and theorem [4].

Definition 1 (Hessian of a Function): The Hessian of a function $\mathcal{J} : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\nabla^2\mathcal{J}(\mathbf{z}) = \begin{bmatrix} \frac{\partial^2\mathcal{J}(\mathbf{z})}{\partial z_1^2} & \frac{\partial^2\mathcal{J}(\mathbf{z})}{\partial z_1\partial z_2} & \cdots & \frac{\partial^2\mathcal{J}(\mathbf{z})}{\partial z_1\partial z_n}\\ \frac{\partial^2\mathcal{J}(\mathbf{z})}{\partial z_2\partial z_1} & \frac{\partial^2\mathcal{J}(\mathbf{z})}{\partial z_2^2} & \cdots & \frac{\partial^2\mathcal{J}(\mathbf{z})}{\partial z_2\partial z_n}\\ \vdots & \vdots & & \vdots\\ \frac{\partial^2\mathcal{J}(\mathbf{z})}{\partial z_n\partial z_1} & \frac{\partial^2\mathcal{J}(\mathbf{z})}{\partial z_n\partial z_2} & \cdots & \frac{\partial^2\mathcal{J}(\mathbf{z})}{\partial z_n^2}\end{bmatrix}$$

where $\mathbf{z}$ is an n-dimensional vector comprised of components $z_1, z_2, \ldots z_n$.

Theorem 2: Let $\mathcal{J}(\mathbf{z})$ be twice differentiable in $\mathbb{R}^n$. Then, given $\mathbf{z}_A \in \mathbb{R}^n$ and a $\mathbf{z}_B \in \mathbb{R}^n$ in a neighborhood about $\mathbf{z}_A$

$$\mathcal{J}(a\mathbf{z}_A + (1-a)\mathbf{z}_B) < a\mathcal{J}(\mathbf{z}_A) + (1-a)\mathcal{J}(\mathbf{z}_B) \tag{8}$$

for all $a \in (0,1)$ and $\mathbf{z}_A \neq \mathbf{z}_B$ if $\nabla^2\mathcal{J}(\mathbf{z})$ is positive definite for all $\mathbf{z}$ [which is denoted by $\nabla^2\mathcal{J}(\mathbf{z}) > 0$ for all $\mathbf{z} \in \mathbb{R}^n$].

Applying Theorem 2, we see that if we constrain $\nabla^2 J(\mathbf{u})$ to be positive definite (i.e., $\nabla^2 J(\mathbf{u}) > 0$), then J fulfills

$$J(a\mathbf{u}_A + (1-a)\mathbf{u}_B) < aJ(\mathbf{u}_A) + (1-a)J(\mathbf{u}_B) \tag{9}$$

for $a \in (0,1)$ and for all $\mathbf{u}_A, \mathbf{u}_B \in \mathbb{R}^n$ such that $\mathbf{u}_A \neq \mathbf{u}_B$. Hence, (9) implies that $J(\mathbf{u})$ increases along any ray originating from the global minimum $\bar{\mathbf{u}}$ [4]. We can then bound $J(\mathbf{u}) - J(\bar{\mathbf{u}})$ such that $\kappa\|\mathbf{u}-\bar{\mathbf{u}}\|^p \leq J(\mathbf{u}) - J(\bar{\mathbf{u}}) \leq \lambda\|\mathbf{u}-\bar{\mathbf{u}}\|^p$ for some integer $p \geq 1$ and $0 < \kappa < \lambda < \infty$, which fulfills J1). Furthermore, the increasing nature of $J(\mathbf{u})$ along any ray originating from $\bar{\mathbf{u}}$ ensures that $J(\mathbf{u}) \to \infty$ as $\|\mathbf{u}\| \to \infty$ [see J2)]. Inequality (9) also suggests that J must be strictly convex by definition [4]. The strict-sense convexity of J implies that J has no points (other than at $\bar{\mathbf{u}}$), which have a gradient value of zero. Therefore, with an appropriate gain γ, it is always possible to decrease the cost at the next iteration using the gradient update law of (3) for any $\mathbf{u} \neq \bar{\mathbf{u}}$. As a result, Condition J3) is fulfilled because $J(\mathbf{u}^{k+1}) - J(\mathbf{u}^k) < 0$ for all k, where $\mathbf{u}^k \neq \bar{\mathbf{u}}$.

Thus, using our analysis methodology, we have determined that $\nabla^2 J(\mathbf{u}^k) > 0$ for all k ensures the global convergence of the algorithm to a unique solution. Using a different candidate Lyapunov function can provide a different set of conditions.

We show in Appendix A that the Hessian of the cost function for the NAS-RIF algorithm is given in matrix-vector notation by

$$\begin{aligned}\nabla^2 J(\mathbf{u}) = 2\mathbf{G}^T\mathbf{I}_{\text{sup}}(\mathbf{u})\mathbf{G} + 2\mathbf{G}^T\mathbf{I}_{\overline{\text{sup}}}\mathbf{G}\\ + \frac{2\gamma}{N_{xu}N_{yu}}\mathbf{i}_{N_{xu}N_{yu}}\mathbf{i}_{N_{xu}N_{yu}}^T.\end{aligned} \tag{10}$$

We define and discuss the associated variables below. We assume for simplicity that the blurred image $g(x,y)$ of dimension $N_{xg} \times N_{yg}$ is indexed from $(0,0)$ to $(N_{xg}-1, N_{yg}-1)$ and the that FIR deconvolution filter $u(x,y)$ of size $N_{xu} \times N_{yu}$ is indexed from $(0,0)$ to $(N_{xu}-1, N_{yu}-1)$.

Since the vector $\mathbf{u}$ is the row-ordered vector of filter coefficients $u(x,y)$, it is given by

$$\begin{aligned}\mathbf{u} = [u(0,0)\,u(0,1)\,\cdots\,u(0,N_{yu}-1)\,u(1,0)\cdots\\ u(1,1)\,\cdots\,u(N_{xu}-1,N_{yu}-1)]^T.\end{aligned} \tag{11}$$

Similarly, $\hat{\mathbf{f}}$ is the row-ordered vector of the pixels of $\hat{f}(x,y) = g(x,y) * u(x,y)$, i.e.,

$$\hat{\mathbf{f}} = [\hat{f}(0,0)\hat{f}(0,1)\cdots\hat{f}(0,N_{yf}-1)\cdots \hat{f}(1,0)\hat{f}(1,1)\cdots\hat{f}(N_{xf}-1,N_{yf}-1)]^T. \tag{12}$$

We also use (x_i, y_i) to denote the pixel location in the image $\hat{f}(x,y)$ of the element corresponding to the ith row of $\hat{\mathbf{f}}$. For example, $(x_1, y_1) = (0,0)$, $(x_2, y_2) = (0,1)$, and $(x_n, y_n) = (\bar{x}, \bar{y})$ such that $n = \bar{x}N_{yu} + \bar{y} + 1$.

The remaining matrices and vectors in (10) are defined as follows. $\mathbf{G}$ is an $(N_{xg}+N_{xu}-1)(N_{yg}+N_{yu}-1)\times(N_{xu}N_{yu})$ matrix such that $\hat{\mathbf{f}} = \mathbf{G}\mathbf{u}$. Specifically

$$\mathbf{G} = \begin{bmatrix} \tilde{\mathbf{g}}_{0,0} \\ \tilde{\mathbf{g}}_{0,1} \\ \vdots \\ \tilde{\mathbf{g}}_{0,N_{yg}+N_{yu}-1} \\ \tilde{\mathbf{g}}_{1,0} \\ \vdots \\ \tilde{\mathbf{g}}_{N_{xg}+N_{xu}-2,N_{yg}+N_{yu}-2} \end{bmatrix} \tag{13}$$

where $\tilde{\mathbf{g}}_{x,y}$ is a $1 \times N_{xu}N_{yu}$ vector given by

$$\tilde{\mathbf{g}}_{x,y} = [\tilde{g}(x,y)\tilde{g}(x,y-1)\cdots\tilde{g}(x,y-N_{xu}+1)\tilde{g}(x-1,y) \cdots\tilde{g}(x-1,y-1)\cdots\tilde{g}(x-N_{xu}+1,y-N_{yu}+1)] \tag{14}$$

and $\tilde{g}(x,y)$ is defined as (15), shown at the bottom of the page. The matrix $\mathbf{I}_{\text{sup}}(\mathbf{u})$ is a diagonal matrix dependent on $\mathbf{u}$ with diagonal entries given by

$$[\mathbf{I}_{\text{sup}}(\mathbf{u})]_{ii} = \begin{cases} u_s(-\hat{f}(x_i,y_i)), & \text{if } (x_i,y_i)\in D_{\text{sup}} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where $[\cdot]_{ij}$ denotes the (i,j)th element of a given matrix, and (x_i, y_i) is the pixel location in the restored image $\hat{f}$ corresponding to the ith element of $\hat{\mathbf{f}}$. Similarly, $\mathbf{I}_{\overline{\text{sup}}}$ is a diagonal matrix (independent of $\mathbf{u}$) with the diagonal entries given by

$$[\mathbf{I}_{\overline{\text{sup}}}]_{ii} = \begin{cases} 0 & \text{if } (x_i,y_i)\in D_{\text{sup}} \\ 1 & \text{otherwise} \end{cases} \tag{17}$$

where D_{sup} is the set of all pixels within the region of support as described in Section II-D. The vector $\mathbf{i}_{N_{xu}N_{yu}}$ is an all-ones column vector of dimension $N_{xu}N_{yu}\times 1$, i.e.,

$$\mathbf{i}_{N_{xu}N_{yu}} = \Big[\underbrace{1\ \ 1\ \ \cdots\ \ 1}_{N_{xu}N_{yu}\ \text{terms}}\Big]^T. \tag{18}$$

We denote the matrix transpose operator by $(\cdot)^T$.

The right side of (10) is comprised of three components. It is straightforward to see that the third component is positive-semidefinite and cannot be positive definite as it has a rank of one (i.e., it is not full rank). The first component is also positive semi-definite in general. Because the $\mathbf{I}_{\text{sup}}(\mathbf{u})$ diagonal matrix is a function of $\mathbf{u}$, its rank depends on $\mathbf{u}$, and we can always find a $\mathbf{u}$ such that $\mathbf{G}^T\mathbf{I}_{\text{sup}}(\mathbf{u})\mathbf{G}$ becomes positive semidefinite. For example, substituting $\mathbf{u} = \mathbf{0}$ results in it reducing to the all-zero matrix. Thus, it is not possible in general to constrain the component to be positive definite for all $\mathbf{u}$.

The second component, however, has no dependence on $\mathbf{u}$ and can be constrained to be positive definite. Thus, we force

$$\mathbf{M} \triangleq \mathbf{G}^T\mathbf{I}_{\overline{\text{sup}}}\mathbf{G} > 0 \tag{19}$$

which implies that the $N_{xu}N_{yu}\times N_{xu}N_{yu}$ matrix $\mathbf{M}$ must be full rank. We can use this information to determine whether a given deconvolution filter size can guarantee a unique solution. The matrix $\mathbf{M}$ is a function of the blurred image pixels (due to the presence of $\mathbf{G}$) and the support size of the true image (due to the presence of $\mathbf{I}_{\overline{\text{sup}}}$). In the next section, we show that in some cases, we can determine an optimal (in terms of accuracy) filter size, which also guarantees a unique solution.

C. The Accuracy and Uniqueness Tradeoff

Using (19), we can rewrite $\mathbf{M}$ as

$$\mathbf{M} = \sum_{(x,y)\in\bar{D}_{\text{sup}}} \tilde{\mathbf{g}}_{x,y}^T\tilde{\mathbf{g}}_{x,y} \tag{20}$$

which is the sum of $||\bar{D}_{\text{sup}}||$ rank one $N_{xu}N_{yu}\times N_{xu}N_{yu}$ matrices, where $||\bar{D}_{\text{sup}}||$ denotes the number of elements in $\bar{D}_{\text{sup}}$. Therefore, a necessary (but not sufficient) condition to guarantee that $\mathbf{M}$ is full rank[3] is

$$||\bar{D}_{\text{sup}}|| \geq N_{xu}N_{yu} \tag{21}$$

which can provide an upper bound on the size of u. We evaluate this condition for the three cases shown in Fig. 2.

Case A) $||\bar{D}_{\text{sup}}||$ depends on the values of both N_{xu} and N_{yu}. Using Fig. 2 and the definitions of L_{xA} and L_{yA} as labeled in the figure, (21) reduces to

$$\begin{aligned} ||\bar{D}_{\text{sup}}|| &= (N_{xg}+N_{xu}-1)(N_{yg}+N_{yu}-1) - L_{xA}L_{yA} \geq N_{xu}N_{yu} \\ &\Leftrightarrow (N_{xg}-1)(N_{yg}-1) - L_{xA}L_{yA} + N_{xu}(N_{yg}-1) + N_{yu}(N_{xg}-1) \geq 0 \end{aligned} \tag{22}$$

which is always fulfilled if the blurred image has dimensions greater than $L_{xA}\times L_{yA}$ (dimensions of the support in Case A of Fig. 2) and greater than 1×1. This suggests that there is no upper bound on the size of u. An arbitrarily large size of practical relevance can be chosen.

[3] Recall that the condition that $\mathbf{M}$ is full rank is a sufficient condition for uniqueness. Thus, (21) is necessary for a sufficient condition for uniqueness [that is, for (19) to hold].

$$\tilde{g}(x,y) = \begin{cases} g(x,y), & \text{for } x = 0,1,\ldots,N_{xg}-1 \quad \text{and} \quad y = 0,1,\ldots,N_{yg}-1 \\ 0, & \text{otherwise} \end{cases}. \tag{15}$$

Case B) $||\bar{D}_{\text{sup}}||$ does not depend on N_{xu} or N_{yu}. Condition (21) gives us

$$||\bar{D}_{\text{sup}}|| = L_{xB}L_{yB} \geq N_{xu}N_{yu} \tag{23}$$

where L_{xB} and L_{yB} are defined as in Case B of Fig. 2. An upper bound of $N_{xu} = L_{xB}$, $N_{yu} = L_{yB}$ can be selected.

Case C) $||\bar{D}_{\text{sup}}||$ depends on N_{yu} only. The condition

$$||\bar{D}_{\text{sup}}|| = L_{xC}(N_{yg} + N_{yu} - 1) \geq N_{xu}N_{yu} \tag{24}$$

where L_{xC} is shown in Case C of Fig. 2 is necessary to ensure that $\mathbf{M}$ is full rank. We may select any N_{xu} and N_{yu} that fulfills (24) as an upper bound on the size of u.

Similar analysis can be carried out for any arbitrary support shape.

It has been shown in [11] that increasing the size of u improves the accuracy of the restored image. However, our previous analysis shows that in some cases, there is an upper bound on the size of u to ensure a unique solution. Thus, there is an inherent tradeoff between accuracy and the possibility of a unique solution. The matrix inequality of (19) provides a testable condition to evaluate the convergence properties of the algorithm prior to restoration. An optimal FIR deconvolution filter of selected dimensions can be chosen to guarantee a unique solution using the following general procedure.

1) Select the maximum initial size for the filter $\mathbf{u}$ that fulfills (21) (i.e., a necessary condition for $\mathbf{M}$ to be full rank). We provided three examples in this section (above) on the use of (21) to find maximum values for N_{xu} and N_{yu}. If bounds can be found as in Cases B and C, the filter size is set to these values. If the filter size has no bounds (as in Case A), an arbitrarily large filter size is used.
2) Build the matrix $\mathbf{M}$ given in (19) using the following procedure:
   2a) Form the matrix $\mathbf{G}$ using (13) given the blurred image $g(x, y)$.
   2b) Form $\mathbf{I}_{\overline{\text{sup}}}$ from the given region of support D_{sup}. D_{sup} is assumed to be known prior to restoration; otherwise, a support-finding algorithm such as in [3] can be used.
   2c) Compute $\mathbf{M} = \mathbf{G}^T\mathbf{I}_{\overline{\text{sup}}}\mathbf{G}$.
3) Test the rank of $\mathbf{M}$. If $\mathbf{M}$ is not full rank, decrease the size of $\mathbf{u}$, and go to 2. Otherwise, stop and use the preceding filter size tested in the NAS-RIF algorithm.

In practice, $\mathbf{M}$ is almost always full rank for the upper bound on the size of $\mathbf{u}$. However, in some cases, additive noise and quantization of the blurred image may result in a loss of rank. We provide one such instance next. Since the NAS-RIF algorithm can be used for 1-D applications such as gamma ray spectra processing, we give a simple 1-D numerical example.

Example 1—No Quantization: Consider the situation in which the true image has a support at locations $f(0)$, $f(1)$, $f(5)$, and $f(6)$ and is given by

$$\begin{aligned}\mathbf{f} &= [f(0) \quad f(1) \quad f(2) \quad f(3) \quad f(4) \quad f(5) \quad f(6)] \\ &= [\ 1 \quad 2 \quad 0 \quad 0 \quad 0 \quad 2 \quad 1\ ].\end{aligned}$$

The background pixel value is $L_B = 0$, and $\bar{D}_{\text{sup}}$ includes the three center locations. The blurring function is

$$\begin{aligned}\mathbf{h} &= [h(0) \quad h(1) \quad h(2)] \\ &= [\ 0.2 \quad 0.6 \quad 0.2\ ].\end{aligned}$$

The resulting convolution product is given by

$$\begin{aligned}\mathbf{g} &= [g(0) \quad g(1) \quad g(2) \quad g(3) \quad g(4) \quad g(5) \quad g(6)] \\ &= [\ 1.0 \quad 1.4 \quad 0.4 \quad 0 \quad 0.4 \quad 1.4 \quad 1.0\ ]\end{aligned} \tag{25}$$

which forms the blurred signal such that $N_{xg} = 7$ and $N_{yg} = 1$; we assume that no quantization of the blurred image pixels occurs and that the blurring at the ends of the signal are truncated from the signal support as in Case B. Following the procedure outlined in Section 3.1, we have the following.

1) We select a filter size of 3 (i.e., $N_{xu} = 3, N_{yu} = 1$) using the results of Case B (i.e., $N_{xu} = L_{xB}$ and $N_{yu} = L_{yB}$).

2a) We form the matrix $\mathbf{G}$ as follows:
We form the row vectors $\tilde{\mathbf{g}}_i$ for $i = 0, 1, \ldots, N_{xg} + N_{xu} - 2$.

$$\begin{aligned}
\tilde{\mathbf{g}}_0 &= [\tilde{g}(0) \quad \tilde{g}(-1) \quad \tilde{g}(-2)] = [1.0 \quad 0.0 \quad 0.0] \\
\tilde{\mathbf{g}}_1 &= [\tilde{g}(1) \quad \tilde{g}(0) \quad \tilde{g}(-1)] = [1.4 \quad 1.0 \quad 0.0] \\
\tilde{\mathbf{g}}_2 &= [\tilde{g}(2) \quad \tilde{g}(1) \quad \tilde{g}(0)] = [0.4 \quad 1.4 \quad 1.0] \\
\tilde{\mathbf{g}}_3 &= [\tilde{g}(3) \quad \tilde{g}(2) \quad \tilde{g}(1)] = [0.0 \quad 0.4 \quad 1.4] \\
\tilde{\mathbf{g}}_4 &= [\tilde{g}(4) \quad \tilde{g}(3) \quad \tilde{g}(2)] = [0.4 \quad 0.0 \quad 0.4] \\
\tilde{\mathbf{g}}_5 &= [\tilde{g}(5) \quad \tilde{g}(4) \quad \tilde{g}(3)] = [1.4 \quad 0.4 \quad 0.0] \\
\tilde{\mathbf{g}}_6 &= [\tilde{g}(6) \quad \tilde{g}(5) \quad \tilde{g}(4)] = [1.0 \quad 1.4 \quad 0.4] \\
\tilde{\mathbf{g}}_7 &= [\tilde{g}(7) \quad \tilde{g}(6) \quad \tilde{g}(5)] = [0.0 \quad 1.0 \quad 1.4] \\
\tilde{\mathbf{g}}_8 &= [\tilde{g}(8) \quad \tilde{g}(7) \quad \tilde{g}(6)] = [0.0 \quad 0.0 \quad 1.0].
\end{aligned} \tag{26}$$

We use (13) to form the matrix $\mathbf{G}$

$$\mathbf{G} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 1.4 & 1.0 & 0.0 \\ 0.4 & 1.4 & 1.0 \\ 0.0 & 0.4 & 1.4 \\ 0.4 & 0.0 & 0.4 \\ 1.4 & 0.4 & 0.0 \\ 1.0 & 1.4 & 0.4 \\ 0.0 & 1.0 & 1.4 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}. \tag{27}$$

2b) We form $\mathbf{I}_{\overline{\text{sup}}}$, using the fact that $\bar{D}_{\text{sup}}$ is of size 3 and is centered on the restored image $\hat{f}$.

$$\mathbf{I}_{\overline{\text{sup}}} = \begin{bmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{bmatrix}. \tag{28}$$

2c) We compute

$$\mathbf{M}_3 = \mathbf{G}^T \mathbf{I}_{\overline{\text{sup}}} \mathbf{G} = \frac{1}{25} \begin{bmatrix} 53 & 14 & 4 \\ 14 & 8 & 14 \\ 4 & 14 & 53 \end{bmatrix}. \tag{29}$$

3) Since the determinant of $\mathbf{M}_3$ is nonzero, it is full-rank and positive definite. Therefore, we may use a filter size of 3 to guarantee a unique solution.

In our next example, we show how quantization noise can cause the matrix to lose rank.

Example 2—Quantization of the Blurred Signal: We perform the same steps once more on the quantized blurred signal g. We round the convolution results to the nearest integer so that

$$\begin{aligned} \mathbf{g}_Q &= [g(0) \quad g(1) \quad g(2) \quad g(3) \quad g(4) \quad g(5) \quad g(6)] \\ &= [\ 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1\ ]. \end{aligned} \tag{30}$$

We select a filter size of 3 (i.e., $N_{xu} = 3, N_{yu} = 1$) using the results of Case B once more. Using the same steps as in Example 1, computation of $\mathbf{M}_3$ gives

$$\mathbf{M}_3 = \mathbf{G}^T \mathbf{I}_{\overline{\text{sup}}} \mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{31}$$

Since the determinant of $\mathbf{M}_3$ is zero, it is not full-rank. Thus, a unique solution cannot be guaranteed for a deconvolution filter size of 3.

Repeating the procedure for $N_{xu} = 2$ and $N_{yu} = 1$, we find that

$$\mathbf{M}_2 = \mathbf{G}^T \mathbf{I}_{\overline{\text{sup}}} \mathbf{G} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{32}$$

which is not full rank. Thus, a unique solution cannot be guaranteed for any filter size greater than 1 due to the coarse quantization imposed in our example.

IV. Conclusion

In this paper, we employ Lyapunov's direct method to analyze the convergence properties of iterative blind deconvolution algorithms. The analysis approach is shown to be successful in providing sufficient conditions for the unique global convergence of an iterative blind deconvolution algorithm. Depending on the choice of the Lyapunov function V, different sets of conditions may be obtained. The most difficult part of the proposed analysis approach is the selection of an appropriate Lyapunov function V. This limitation, however, allows the technique to be general and to be applied to a broad class of algorithms.

The method is shown to be feasible and straightforward in determining sufficient conditions for a unique solution to the NAS-RIF algorithm. We successfully convert the problem of establishing conditions for global convergence to determining constraints that guarantee the full rank of algorithm-related matrices. We also develop insight from our analysis framework to determine a method to find an FIR deconvolution filter that guarantees the most accurate unique solution. We demonstrate this with the use of examples.

Appendix
Derivation of the Hessian of J

We provide a brief derivation for the Hessian of $J(\mathbf{u})$ given (10). From Definition 1, we see that we must take the second derivative of J with respect to the elements of $\mathbf{u}$. The variables used in our analysis have been defined in Sections II-D and III-B.

From (5), we see that the derivative of J with respect to an element of $\mathbf{u}$, namely $u(\mathbf{i})$, is given by

$$\begin{aligned} \frac{\partial J(\mathbf{u})}{\partial u(\mathbf{i})} &= \sum_{\mathbf{r} \in D_{\text{sup}}} 2\hat{f}(\mathbf{r}) \frac{\partial \hat{f}(\mathbf{r})}{\partial u(\mathbf{i})} u_s(-\hat{f}(\mathbf{r})) \\ &\quad + \sum_{\mathbf{r} \in D_{\text{sup}}} \hat{f}^2(\mathbf{r}) \frac{\partial u_s(-\hat{f}(\mathbf{r}))}{\partial u(\mathbf{i})} \\ &\quad + \sum_{\mathbf{r} \in \bar{D}_{\text{sup}}} 2[\hat{f}(\mathbf{r}) - L_B] \frac{\partial \hat{f}(\mathbf{r})}{\partial u(\mathbf{i})} + 2\gamma \left[\sum_{\forall \mathbf{r}} u(\mathbf{r}) - 1 \right] \end{aligned} \tag{33}$$

where we use the product and chain rules for differentiation. Since

$$\frac{\partial \hat{f}(\mathbf{r})}{\partial u(\mathbf{i})} = g(\mathbf{r} - \mathbf{i}) \tag{34}$$

and

$$\frac{\partial u_s(-\hat{f}(\mathbf{r}))}{\partial u(\mathbf{i})} = \hat{f}^2(\mathbf{r}) \delta(-\hat{f}(\mathbf{r})) \frac{\partial \hat{f}(\mathbf{r})}{\partial u(\mathbf{i})} = 0 \tag{35}$$

we find that our expression reduces to

$$\begin{aligned} \frac{\partial J(\mathbf{u})}{\partial u(\mathbf{i})} &= 2 \sum_{\mathbf{r} \in D_{\text{sup}}} \hat{f}(\mathbf{r}) g(\mathbf{r} - \mathbf{i}) u_s(-\hat{f}(\mathbf{r})) \\ &\quad + 2 \sum_{\mathbf{r} \in D_{\text{sup}}} [\hat{f}(\mathbf{r}) - L_B] g(\mathbf{r} - \mathbf{i}) 2\gamma \left[\sum_{\forall \mathbf{r}} u(\mathbf{r}) - 1 \right]. \end{aligned} \tag{36}$$

Similarly, we can differentiate once more with respect to $u(\mathbf{j})$ to give

$$\begin{aligned} \frac{\partial^2 J(\mathbf{u})}{\partial u(\mathbf{i}) \partial u(\mathbf{j})} &= 2 \sum_{\mathbf{r} \in D_{\text{sup}}} g(\mathbf{r} - \mathbf{i}) u_s(-\hat{f}(\mathbf{r})) g(\mathbf{r} - \mathbf{j}) \\ &\quad + 2 \sum_{\mathbf{r} \in \bar{D}_{\text{sup}}} g(\mathbf{r} - \mathbf{i}) g(\mathbf{r} - \mathbf{j}) + 2\gamma. \end{aligned} \tag{37}$$

Using the definition of the Hessian (Definition 1), as well as our indexing notation described in Section III-B in which the indices of the 2-D FIR filter u range from $(0, 0)$ to $(N_{xu} - 1, N_{yu} - 1)$, it is straightforward to write $\nabla^2 J(\mathbf{u})$

$$\nabla^2 J(\mathbf{u}) = \begin{bmatrix} \frac{\partial^2 \mathcal{J}(\mathbf{u})}{\partial u^2(0,0)} & \frac{\partial^2 \mathcal{J}(\mathbf{u})}{\partial u(0,0)\partial u(0,1)} & \cdots & \frac{\partial^2 \mathcal{J}(\mathbf{u})}{\partial u(0,0)\partial u(N_{xu}-1,N_{yu}-1)} \\ \frac{\partial^2 \mathcal{J}(\mathbf{u})}{\partial u(0,1)\partial u(0,0)} & \frac{\partial^2 \mathcal{J}(\mathbf{u})}{\partial u^2(0,1)} & \cdots & \frac{\partial^2 \mathcal{J}(\mathbf{u})}{\partial u(0,1)\partial u(N_{xu}-1,N_{yu}-1)} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 \mathcal{J}(\mathbf{u})}{\partial u(N_{xu}-1,N_{yu}-1)\partial u(0,0)} & \frac{\partial^2 \mathcal{J}(\mathbf{u})}{\partial u(N_{xu}-1,N_{yu}-1)\partial u(0,1)} & \cdots & \frac{\partial^2 \mathcal{J}(\mathbf{z})}{\partial u^2(N_{xu}-1,N_{yu}-1)} \end{bmatrix} \tag{38}$$

as (38), shown at the top of the page, or equivalently in matrix-vector notation using (37) as

$$\nabla^2 J(\mathbf{u}) = \mathbf{G}^T \mathbf{I}_{\text{sup}}(\mathbf{u})\mathbf{G} + 2\mathbf{G}^T \mathbf{I}_{\overline{\text{sup}}}\mathbf{G} + \frac{2}{N_{xu}N_{yu}} \mathbf{i}_{N_{xu}N_{yu}} \mathbf{i}^T_{N_{xu}N_{yu}}. \tag{39}$$

A similar but more detailed derivation can be found in [11].

ACKNOWLEDGMENT

The authors wish to thank the associate editor P. C. Doerschuk and the reviewers of this paper for their helpful comments and suggestions.

REFERENCES

[1] S. Bellini, "Bussgang techniques for blind deconvolution and equalization," *Blind Deconvolution*, S. Haykin, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1991, pp. 8–59.

[2] D. Kundur and D. Hatzinakos, "Blind image restoration via recursive filtering using deterministic constraints," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, GA, May 1996.

[3] ——, "Recursive blind deconvolution of still images based on nonnegativity and support constraints," *IEEE Trans. Signal Processing*, vol. 46, pp. 375–390, Feb. 1998.

[4] J. B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms I.* New York: Springer-Verlag, 1993.

[5] M. Vidyasagar, *Nonlinear Systems Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1993, 2nd ed.

[6] R. W. Gerchberg, "Super-resolution through error energy reduction," *Optica-Acta*, vol. 21, no. 9, pp. 709–720, Sept. 1974.

[7] A. Papoulis, "A new algorithm in spectral analysis and band-limited extrapolation," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 735–742, Sept. 1975.

[8] R. G. Wiley, "On an iterative technique for recovery of bandlimited signals," *Proc. IEEE*, vol. 66, pp. 522–523, Apr. 1978.

[9] I. W. Sandberg, "On the properties of some systems that distort signals—I," *Bell Syst. Tech. J.*, vol. 42, no. 5, pp. 2003–2046, Sept. 1963.

[10] H. K. Khalil, *Nonlinear Systems*. Toronto, Ont., Canada: Maxwell-Macmillan, 1992.

[11] D. Kundur, "Blind deconvolution of still images using recursive inverse filtering," M.A.Sc. thesis, Dept. Elect. Comput. Eng., Univ. Toronto, Toronto, Ont., Canada, 1995. Available WWW: http://www.comm.toronto.edu/∼deepa/masc.html.

Deepa Kundur (S'93) was born in Toronto, Ont., Canada. She received the Bachelor of Applied Science degree in 1993 and the Master of Applied Science degree in communications in 1995, both from the Department of Electrical and Computer Engineering, University of Toronto. She is currently pursuing doctoral studies at the University of Toronto.

Her research interests include digital watermarking of multimedia, blind image restoration, and data fusion for the classification of remote sensing imagery.

Ms. Kundur is currently an engineer-in-training with the Professional Engineers of Ontario (PEO).

Dimitrios Hatzinakos (M'90) received the Diploma degree from the University of Thessaloniki, Thessaloniki, Greece, in 1983, the M.A.Sc. degree from the University of Ottawa, Ottawa, Ont., Canada, Canada, in 1986, and the Ph.D. degree from Northeastern University, Boston, MA, in 1990, all in electrical engineering.

From January 1990 until August 1990, he was a Visiting Assistant Professor at the Department of Electrical and Computer Engineering, Northeastern University. From September 1990 until June 1995, he was an Assistant Professor at the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ont., where he now holds the rank of Associate Professor with tenure. His research interests span the fields of digital signal and image processing with applications, higher order spectral analysis, blind and adaptive filter algorithms, and digital communications. He has organized and taught many short courses in signal processing devoted to continuing engineering education and given numerous seminars in the areas of higher order spectra and blind deconvolution. He is author/co-author of more than 60 papers in technical journals and conference proceedings and has contributed to four books in his area of interest. His industrial and consulting experience includes consulting through Electrical Engineering Consociates, Ltd. His consulting work has included contracts with United Signals and Systems, Inc., Burns and Fry, Ltd., Pipetronix, Ltd., and the Defense Research Establishment Ottawa (DREO).

Dr. Hatzinakos served as a member of the Conference Board of the IEEE Statistical Signal and Array Processing Committee (SSAP) from 1992 until 1995 and Co-Technical Program Chair of the Fifth Workshop on Higher Order Statistics, Banff, Alta., Canada, in July 1997. He is a member of EURASIP, the Professional Engineers of Ontario (PEO), and the Technical Chamber of Greece.