Fast and Accurate Computation of the Myriad Filter via Branch-and-Bound Search

Rafael C. Núñez, *Student Member, IEEE*,
Juan G. Gonzalez, *Senior Member, IEEE*,
Gonzalo R. Arce, *Fellow, IEEE*, and John P. Nolan

***Abstract*—The myriad filter has demonstrated to be a robust countermeasure against the negative effect that impulsive noise has over electronic systems. However, its use is still limited in systems where processing speed is critical, as is the case of radar, sonar, and real-time audio and video processing. This limitation has its roots in the challenges imposed by the numerical approximation of the myriad filter. In particular, minimization operations at the interior of nonlinear operations are sensitive components that have a direct impact on the performance of the filtering algorithms. In the case of the myriad filter, the minimization of functions with multiple local minima is a common operation, and poorly chosen algorithms compromise the good behavior of the filter. In this correspondence, we present an alternative for the minimization of the objective function in the computation of the myriad filter. This solution exploits general concepts in global optimization and adapts them to the particular case of myriad filtering. This technique improves accuracy and speed in the computation of the myriad filter, making the method feasible in many problems.**

***Index Terms*—alpha-stable distributions, branch and bound, heavy tails, impulsive noise, myriad filters.**

I. Introduction

The myriad filter [1], [2] has demonstrated to be a robust countermeasure against the negative effect that impulsive noise has over electronic systems. During the last decade, myriad filters have been successfully integrated in numerous applications, including signal smoothing [3], applied astronomy [4], digital video [5], digital communications [6], industrial process control [7], image denoising [8], and electrocardiography [9].

Even though the myriad filter is now used in industrial applications, its use is still limited in systems where processing speed is critical, as is the case of radar, sonar, and real-time audio and video processing. This limitation has its roots in the challenges imposed by the numerical approximation of the myriad filter. One of the most critical operations in the computation of the myriad is the minimization of a (possibly) nonconvex function. The use of inappropriate minimization functions can compromise the accuracy of the filter [10], a situation that cannot be tolerated in most applications. To guarantee accuracy of the minimization operations, global search methods are preferred, but this kind of solutions typically demand extremely large computing times, making the use of the myriad unfeasible for applications where speed is of the essence.

Manuscript received April 6, 2007; revised December 20, 2007. This work was supported in part by Robust Analysis, Inc. under a research contract with the U.S. Navy. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sergiy A. Vorobyov.

R. C. Núñez is with Hasten Consulting, LLC, Aventura, FL 33180 USA (e-mail: rafael.nunez@hastenconsulting.com).

J. G. Gonzalez is with Robust Analysis, Inc., Weston, FL 33326 USA (e-mail: juan@robustanalysis.com).

G. R. Arce is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: arce@ece.udel.edu).

J. P. Nolan is with the Department of Mathematics and Statistics, American University, Washington, DC 20016 USA (e-mail: jpnolan@american.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2008.917889

The problem of the efficient computation of the myriad has been addressed before with the introduction of fixed point search methods [11], where the weighted myriad is formulated as one of the fixed points of a certain mapping. In order to compute these fixed points, [11] presents an iterative algorithm that is initialized with a single point (FPS-I). FPS-I computes the weighed myriad with a high degree of accuracy, at a relatively low computational cost. However, FPS-I does not guarantee the delivery of the global minimum in a given search space [10]. This limitation is addressed by a variant of the fixed point search algorithm, where the minimization process is initialized with multiple starting points (FPS-II) [11]. FPS-II provides a more accurate output of the myriad filter (though there is no guarantee of delivery of the global minimum), but it hazardously increases the computational complexity of the filtering operations. If the order of the myriad filter is small, this complexity is not a concern, but when the filter order increases, the complexity may explode.

In this correspondence, we present an alternative for the minimization of the objective function in the computation of the myriad filter. This solution exploits general concepts in global optimization and adapts them to the particular case of myriad filtering. This technique improves speed in the computation of the myriad filter, providing it with the required strength for the solution of speed-hungry problems.

II. Challenges in the Numerical Computation of the Myriad Filter

The myriad filter is a type of weighted digital filter inherently more powerful than both linear and median filters, and with high statistical efficiency in natural impulsive environments [1], [3]. Myriad filters are statistically robust and, when properly designed, they are able to emulate the deterministic behavior of traditional (linear) FIR and IIR filters while incorporating strong resistance against impulse noise and non-Gaussian background statistics [2], [12], [13].

The myriad filter is defined as a sliding-window filter that slides through the input signal, computing the *sample weighted myriad*:

Definition 1 [Sample Weighted Myriad]: Given a set of samples $x_1, x_2, \ldots, x_N$, and a set of corresponding weights $w_1, w_2, \ldots, w_N$, the sample weighted myriad is defined as

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} \rho(|w_i| \cdot (\mathrm{sgn}(w_i)x_i - \beta)), \tag{1}$$

where $\mathrm{sgn}(x)$ is the positive-zero sign function defined as $\mathrm{sgn}(x) = +1$ for $x \geq 0$, -1 otherwise, and $\rho(x) = \log(K^2 + x^2)$ is the so-called myriad cost function [2], [14]. Note that the weight signs are uncoupled from the weight magnitude values and are merged with the observation samples, thus allowing the use of negative weights (see [15] for more details).

The parameter K is called the *linearity parameter* of the myriad. When K is large, the myriad behaves like the sample weighted mean

$$\hat{\beta} \sim \frac{\sum_{i=1}^{N} w_i^2 \cdot \mathrm{sgn}(w_i) \cdot x_i}{\sum_{i=1}^{N} w_i^2}, \tag{2}$$

whereas when K is small, the myriad becomes a mode-like estimator with significant resistance to the presence of outliers. The ability to tune the linearity parameter according to the demands of the problem at hand gives significant versatility to the filter.

Formalizing the notation in a moving-window framework, let $\{x(\cdot)\}$ be a discrete-time continuous-valued signal. The myriad filter slides a window over the signal $\{x(\cdot)\}$ that selects, at each instant n, a set of samples that comprise the observation vector $\mathbf{x}(n) =$

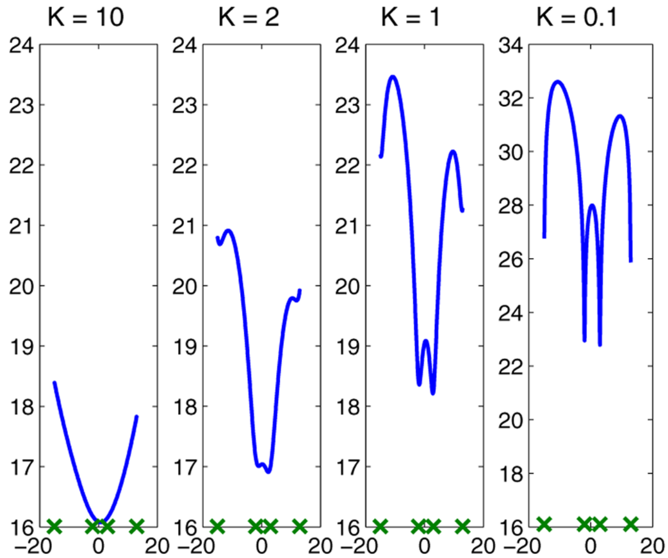

Fig. 1. Degree of nonconvexity in the optimization space of the myriad filter as the tuning parameter of the filter (i.e., K) is varied. The sample points are marked on the horizontal axis ($N = 4$).

$[x(n-N+1), x(n-N+2), \ldots, x(n)] = [x_1(n), x_2(n), \ldots, x_N(n)]$ where $x_i(n) = x(n-N+i)$, and N is the observation window size. For each window position, the filter output is given by

$$y(n) = \arg\min_{\beta} \sum_{i=1}^{N} \rho(|w_i| \cdot (\mathrm{sgn}(w_i)x_i(n) - \beta)). \quad (3)$$

Note that, at each window position, computing the myriad filter may carry a high computational cost. Although the use of fast commercial routines such as STABLE's[1] `stableqkloglik` can drastically speed up the computation of the myriad cost function, (3) requires the minimization of a possibly nonconvex function. This can be not only computationally expensive, but also hazardous, in the sense that using an algorithm that does not guarantee convergence to the global minimum could result in poor or inadequate solutions [10]. Fig. 1 illustrates how the number of local minima in the objective function of (3) grows progressively from zero to N as the linearity parameter of the filter is reduced. This indicates that when operating in highly nonlinear regions (low values of K), the numerical computation of the myriad filter is more challenging, and more susceptible to the quality of the minimization algorithm used.

III. The Branch-and-Bound Search Algorithm

Branch-and-Bound is a family of global optimization methods widely recognized as able to provide fast and accurate results by doing an efficient sampling of the search space [17]. Branch-and-bound methods are based on partition, sampling, and subsequent lower and upper bounding procedures applied iteratively to a collection of active ("candidate") subsets within the search space. Branching algorithms split the search space in intervals, and perform evaluations of those intervals. If an interval can potentially contain the global minimum, it is branched and the resulting subintervals are analyzed as new problems (they feed again the process of branching and bounding), otherwise, the interval is discarded. The discarding of nonpromising intervals avoids superfluous computations that delay the minimization process.

[1]The STABLE software package is developed and distributed by Robust Analysis, Inc. [16].

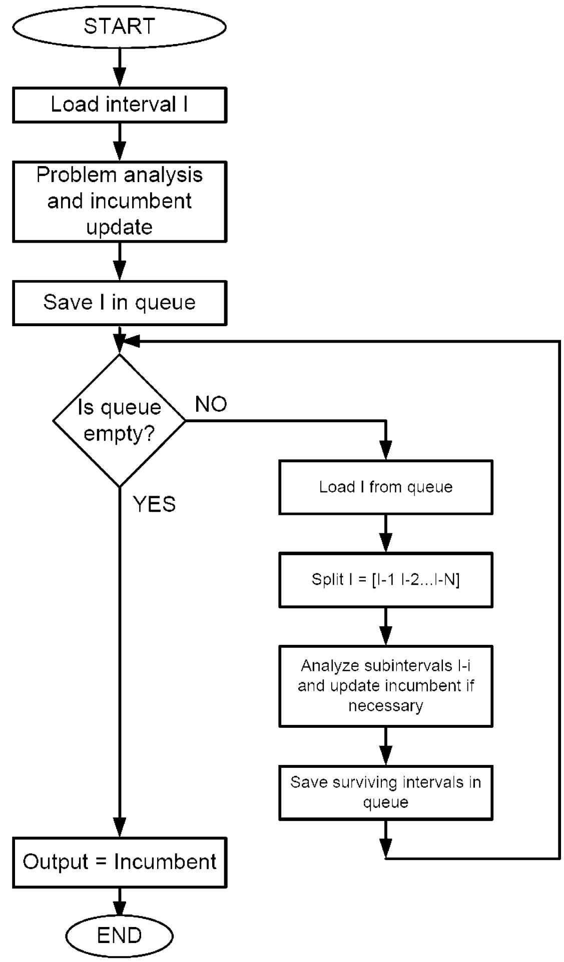

Fig. 2. Flow chart of typical Branch-and-Bound search algorithm.

A flow chart of a generic Branch-and-Bound algorithm is shown in Fig. 2. As Branch-and-Bound techniques are already mature in the global optimization arena, there are a large number of variants. Some of them use distinct branching schemes while others base the differences on the bounding principles [18].

Fig. 3 contains the pseudocode of the Branch-and-Bound algorithm, as it is proposed for the computation of the myriad filter. Based on the basic implementation shown in Fig. 3, different alternatives can arise, able to provide increased performances of the search algorithm.

IV. Lower Bounds of the Myriad Objective Function

The performance of Branch-and-Bound methods depends heavily on the bound functions that they implement. According to the degree of rigourousness applied to the evaluation of intervals, bound functions are usually classified as strong or weak. Techniques that apply a strong criterion to analyze the intervals, are able to identify and discard nonpromising intervals in early stages of the iterative process, accelerating the convergence of the global optimization process, and reducing the memory requirements of the device computing the algorithm. This reduction in the number of iterations of the algorithm is obtained in exchange of an increase in the computation time of each iteration. Weak bounding techniques provide a reduced computation time per iteration, at the cost of an increased number of iterations for the convergence of the algorithm.

In order to compare the two bounding alternatives in the context of the myriad filter, we have designed a weak bound function based on

Branch-and-Bound Algorithm

1 **Initialization**

- Allocate memory for queue elements. The queue is composed of 4 vectors: `Qxlow`, `Qzlow`, `Qxup`, `Qzup`, where `Qxlow` contains the lower limits of the intervals, `Qzlow` contains f(`Qzlow`), `Qxup` contains the upper limits of the intervals, and `Qzup` contains f(`Qup`).
- Initialize in zero both '`rp`' and '`wp`,' the reading and writing pointers of the queues.
- Initialize in zero '`occupiedqueue`,' a counter of the number of queued intervals.
- Compute:
  - `currentzlow` = f(`xlow`)
  - `currentzup` = f(`xup`)
  - `incumbent` = min(`currentzlow`, `currentzup`)
  - `xatincumbent` = $x|_{incumbent}$
- Load the first interval in the queues: [`xlow`, `zlow`, `xup`, `zup`], and update both '`wp`' and '`occupiedqueue`.'

2 **Load new interval**

- Remove an interval `I` = [`currentxlow`, `currentzlow`, `currentxup`, `currentzup`] from the queues, and update '`wp`' and '`occupiedqueue`.'
- Calculate the middle point of the interval `xc` = (`currentxlow` + `currentxup`) / `2`.
- Calculate `zc` = f(`xc`).
- Update `incumbent` (if necessary).
- If (`xc` − `currentxlow`) < `xtol`, go to **Step 5**.

3 **Branch**: Split the interval `I` in two subintervals I_i:

- Create the interval I_1 = [`currentxlow`, `currentzlow`, `xc`, `zc`].
- Create the interval I_2 = [`xc`, `zc`, `currentxup`, `currentzup`].

4 **Bound**: For each subinterval I_i:

- Calculate the lower bound `z_lb` of the subinterval I_i.
- If `z_lb` < `incumbent`, then insert the subinterval into the queues, and update both '`wp`' and '`occupiedqueue`' accordingly.

5 **Next iteration**: If the queue is not empty, go to **Step 2**.

6 **End**: Return '`solution` = `xatincumbent`' and '`globalmin` = `incumbent`.'

Fig. 3. Basic Branch-and-Bound algorithm for the computation of the myriad filter. The inputs to the algorithm are the interval [`xlow, xup`] and the stop criterion `xtol` (minimum size of an interval). The algorithm outputs the global minimum of the objective function $f(\mathrm{x})$ in the interval [`xlow, xup`] through the variable `globalmin`, and the value of x where the global minimum is located, using the variable `xatincumbent`.

Lipschitzian characteristics of the filter, and a strong bound function that relies on local properties of the myriad objective function.

A. Bounding Based on Lipschitzian Properties of the Myriad

Bounding based on Lipschitz constant is one of the most common bounding techniques for Branch-and-Bound [17]. Recall that the sample weighted myriad objective function is defined as $f(\beta) = \sum \rho(|w_i| \cdot (\mathrm{sgn}(w_i)x_i - \beta))$. With Lipschitz bounding, a lower bound for $f(\beta)$ on the interval $[a, b]$ is computed as:

$$f(\beta) \geq g(a,b) = \frac{f(a)+f(b)}{2} - L \cdot \frac{b-a}{2} \tag{4}$$

where L is a Lipschitz constant of f [17]. The challenge here is to determine a good Lipschitz constant for f, i.e., an L as small as possible, so that it allows the discarding of a large number of intervals early during the iteration process.

It can be proven that the following is a Lipschitz constant of the myriad objective function (see Appendix):

$$L = \frac{1}{K}\sum_{i=1}^{N}|w_i|. \tag{5}$$

The use of this type of bounding has the advantage that it makes the implementation of the algorithm easy.

B. Bounding Based on Local Properties of the Objective Function

The weakness of the Lipschitz constant in (5) is that it defines a suboptimal bound, generating a small amount of unnecessary calculations during the minimization operation, and thus increasing the

Modified Branch-and-Bound Algorithm

1 **Initialization**

- Allocate memory for queue elements. The queue is composed of 4 vectors: `Qxlow`, `Qzlowminus`, `Qxup`, `Qzupplus`, where `Qxlow` contains the lower limits of the intervals, `Qzlowminus` contains f^{-}(`Qzlow`) (the fraction of the objective function obtained from the samples $s_i <$ `xlow`), `Qxup` contains the upper limits of the intervals, and `Qzupplus` contains f^{+}(`Qup`) (the fraction of the objective function obtained from the samples $s_i <$ `xup`).
- Initialize in zero both '`rp`' and '`wp`,' the reading and writing pointers of the queues.
- Initialize in zero '`occupiedqueue`,' a counter of the number of queued intervals.
- Compute:
  - `currentzlowminus` = f^{-}(`xlow`)
  - `currentzlowplus` = f^{+}(`xlow`)
  - `currentzlow` = `currentzlowminus` + `currentzlowplus`
  - `currentzupminus` = f^{-}(`xup`)
  - `currentzupplus` = f^{+}(`xup`)
  - `currentzup` = `currentzzupminus` +
  - `currentzupplus`
  - `incumbent` = min(`currentzlow`, `currentzup`)
  - `xatincumbent` = $x|_{incumbent}$
- Load the first interval in the queues: [`xlow`, `zlowminus`, `xup`, `zupplus`], and update both '`wp`' and '`occupiedqueue`.'

2 **Load new interval**

- Remove an interval `I` = [`currentxlow`, `currentzlowminus`, `currentxup`, `currentzupplus`] from the queues, and update '`wp`' and '`occupiedqueue`.'
- Calculate the middle point of the interval `xc` = (`currentxlow` + `currentxup`) / `2`.
- Calculate `zcminus` and `zcplus`.
- Calculate `zc` = `zcminus` + `zcplus`.
- Update `incumbent` (if necessary).
- If (`xc` $-$ `currentxlow`) $<$ `xtol`, go to **Step 5**.

3 **Branch**: Split the interval `I` in two subintervals I_i:

- Create the interval I_1 = [`currentxlow`, `currentzlowminus`, `xc`, `zcplus`].
- Create the interval I_2 = [`xc`, `zcminus`, `currentxup`, `currentzupplus`].

4 **Bound**: For each subinterval I_i:

- Calculate the lower bound `z_lb` of the subinterval I_i, as $\texttt{z\_lb} = \texttt{zlowminus}_{Ii} + \texttt{zupplus}_{Ii}$.
- If `z_lb` $<$ `incumbent`, then insert the subinterval into the queues, and update both '`wp`' and '`occupiedqueue`' accordingly.

5 **Next iteration**: If the queue is not empty, go to **Step 2**.

6 **End**: Return '`solution` = `xatincumbent`' and '`globalmin` = `incumbent`.'

Fig. 4. Branch-and-Bound algorithm optimized for the myriad filter, using localized bounds. The inputs to the algorithm are the interval [`xlow, xup`] and the stop criterion `xtol` (minimum size of an interval). The algorithm outputs the global minimum of the objective function f(x) in the interval [`xlow,xup`] in the variable `globalmin`, and the value of x where the global minimum is located in the variable `xatincumbent`.

number of iterations (in comparison to the use of an optimal Lipschitz constant) in the Branch-and-Bound search. The lack of optimality in the bound defined by (5) has its roots in the fact that it is based on an approximation where the Lipschitz constant is obtained as the sum of contributions from single sample-weight subsets. A tighter bound can be obtained by analyzing all the sample-weight space as a single set.

Based on particular properties of the myriad objective function, it is possible to obtain a tighter lower bound that varies depending on the location of the interval with respect to the sample points.

For an interval $[a, b]$ and real x, the distance between x and $[a, b]$ is defined as

$$\mathrm{dist}(x, [a, b]) = \begin{cases} 0 & x \in [a, b] \\ \min(|x-a|, |x-b|) & x \notin [a, b]. \end{cases}$$

A localized bound for $f(\beta)$ on the interval $[a,b]$, can be derived as:

$$f(\beta) \geq \sum_{i=1}^{N} \inf_{\beta\in[a,b]} \rho(|w_i| \cdot (\operatorname{sgn}(w_i)x_i - \beta)) \quad (6)$$

$$\geq \sum_{i=1}^{N} \rho(r_i) \quad (7)$$

where $r_i = \operatorname{dist}(x_i, [a,b])$.

The advantage of this bound is that the single components $\rho(r_i)$ are very easy to be computed. The use of this bounding method (called "localized bounds") enables a redefined Branch-and-Bound algorithm, particularly optimized for the myriad filter, that increases even more the speed of the filter. This redefinition of the Branch-and-Bound search is shown in Fig. 4.

Additionally, the localized bounds approach generalizes to other cost functions $\rho(x)$ that are symmetric and increasing for $x > 0$. It also generalizes to multivariate problems.

V. Benchmark Results

The two algorithms described in the previous section were integrated in the computation of the myriad filter, and their performance was analyzed in a typical—yet simple—signal processing scenario. In order to asses the power of Branch-and-Bound based myriad filtering, the new implementations were compared against fixed point search algorithms. A total of four variants of the myriad filtering were analyzed:

- **FPS-I**: Corresponds to Fixed Point Search, the iterative algorithm proposed in [11] for the fast computation of myriad filters. The initial point for this algorithm is given by the output of the selection filter [19] applied to all the input samples.
- **FPS-II**: Corresponds to the Fixed Point Search algorithm (as described for FPS-I), initialized with each of the input samples. This technique increases the complexity of the algorithm significantly, but guarantees a more precise output.
- **B&B-I**: The Lipschitz-bounded Branch-and-Bound algorithm. The Lipschitz constant L is designed according to (5). The tolerance of the output is set to 10^{-3}.
- **B&B-II**: The Branch-and-Bound algorithm, bounded with the localized bound, as indicated by (7). The tolerance of the output is set to 10^{-3}.

The testing scenario consists on the filtering of a constant signal contaminated with additive α-stable noise with parameters $\alpha = [1, 1.5]$ and $\gamma = 1$. The contaminated signal is filtered using the myriad filter with tuning parameter $K = 1$ and weights $w_i = 1, i = 1,\ldots,N$. The benchmarks were coded using the STABLE Toolbox for Matlab [16], and all the experiments were executed in the same CPU.

A. Accuracy

As expected, FPS-I, FPS-II, B&B-I, and B&B-II evidence robustness against impulsive noise. In general, all these filters deliver accurate results. The mean absolute error of the filtered signal is almost the same for all of the fixed point search and Branch-and-Bound algorithms, as can be seen on Table I. More detailed information about the individual performance of each estimator can be appreciated through a pairwise performance comparison indicator that we call the *win rate*. For two statistics T and T', we define the *win rate* (WR) of T over T' when estimating β, as $\mathrm{WR}_{T/T'} = \Pr(|T-\beta| < |T'-\beta|)$ [2]. Table II shows the estimated win rates of FPS-I, FPS-II, and B&B-I when the best estimator is B&B-II for a typical subset of this simulation study. As expected, B&B-I renders exactly the same accuracy as B&B-II as their numerical value is always the same (i.e., they only differ on computational speed.) Note that B&B-II wins over both FPS-I and FPS-II more times than it loses to them. When compared against FPS-II, the win and lose rates of B&B-II are of the same order of magnitude. The small difference between them can be explained by precision errors. As described in [10], precision errors are controlled by the stop criteria of the algorithms, and do not represent a threat to the filters' performance. When compared against FPS-I, the win rate of B&B-II is an order of magnitude larger than the lose rate. As opposed to FPS-II, this big difference is explained by the presence of convergence errors in the FPS-I algorithm. Convergence errors are the consequence of minimization algorithms getting stuck in a local minimum of the objective function [10].

TABLE I
Mean Absolute Error of the Filtered Signal, Using a Myriad Filter of Order N, Tuning Parameter $K = 1$, and Weights $w_i = 1, i = 1,\ldots,N$, When the α-Stable Noise is Modeled With Parameters $\alpha = \gamma = 1$. Values not Shown in the Table Were not Calculated (the Time to Solve Those Particular Problems is Extremely Large)

Algorithm	N=4	N=8	N=16	N=32	N=64	N=128	N=512
FPS-I	0.805	0.460	0.302	0.209	0.152	0.104	0.047
FPS-II	0.807	0.460	0.302	0.209	0.152	0.104	0.047
B&B-I	0.807	0.460	0.302	0.210	—	—	—
B&B-II	0.807	0.460	0.302	0.210	0.153	0.106	0.050

TABLE II
Estimated Win Rates of FPS-I, FPS-II, and B&B-I When the Base Estimator is B&B-II

Minimization Algorithm	B&B-II loses	B&B-II wins
FPS-I	0.1101%	3.6233%
FPS-II	1.6715%	2.6124%
B&B-I	0.0000%	0.0000%

Fig. 5 illustrates a typical occurrence of such an error during our simulation study. In this particular example, the sample window size was $\mathrm{N} = 8$. The sample values are marked with crosses horizontally at the bottom of the plot, and the curve represents the objective function to be minimized by the myriad filter computation in Expression (1). The vertical lines show the solutions rendered by each one of the algorithms. Note that FPS-I got stuck in a local minimum that is far away from the correct answer. This points to a limitation of FPS-I that the Branch-and-Bound algorithms do not have: For FPS-I it is not possible to guarantee arbitrarily small filter errors, even if the number of iterations is made arbitrarily large.

B. Computational Speed

The significance of the new Branch-and-Bound algorithms introduced here is the fact that they guarantee accuracy comparable to that of the state of the art fixed point search algorithms, while their implementation can be up to two orders of magnitude faster than FPS for practical real-life scenarios.

Fig. 6 shows the time in seconds for the filtering of the complete contaminated signal ($\alpha = 1$) composed of 10 000 samples. Note how the execution time increases when the filter order grows. This phenomenon applies for all the minimization algorithms, but the rate at which the execution time increases is higher in the case of FPS-I, FPS-II, and B&B-I. The filter that uses B&B-II presents the best performance.

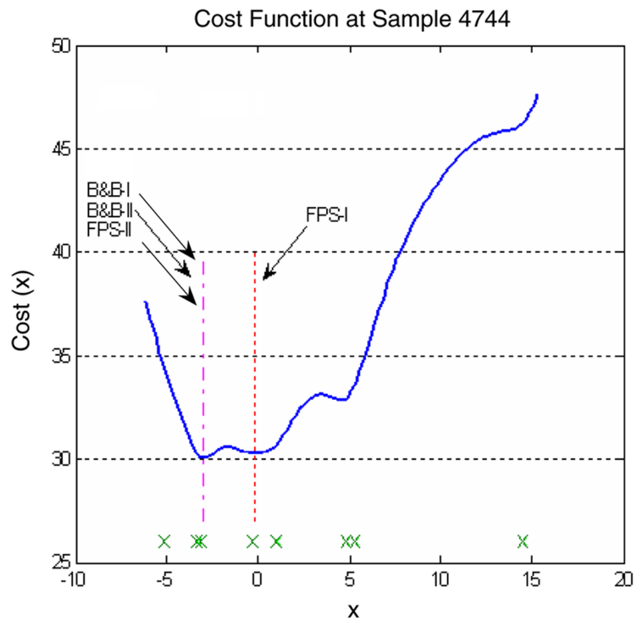

Fig. 5. Myriad filter objective function at sample 4744 $(\mathrm{N} = 8, \alpha = 1)$.

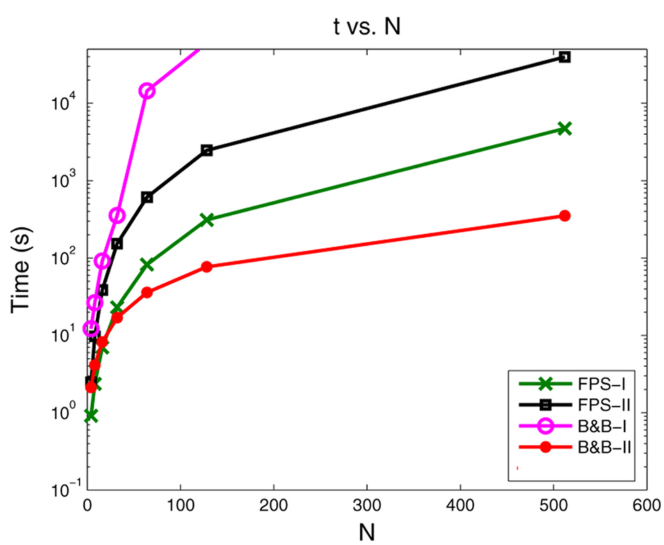

Fig. 6. Execution time of the myriad filter with different minimization algorithms, for a test signal with $\alpha = \gamma = 1$.

When the filter order is 512, B&B-II is approximately 10 times faster than FPS-I.

Fig. 7 shows the time in seconds for the filtering of the complete contaminated signal composed of 10 000 samples, but this time the noise is modeled with parameter $\alpha = 1.5$. In this case, where the filtering problem is simpler, suboptimal bounds like the one in B&B-I become useful: the filtering time in B&B-I is less than that of FPS-I. In this scenario, B&B-II is still the fastest algorithm.

VI. Conclusion

Branch-and-Bound search methods applied to myriad filtering offer a set of powerful tools in the quest for both accurate and fast filters, where the objective functions are characterized by frequent nonconvexities. Of particular importance is the method B&B-II introduced here, which uses a tight bound function. The use of this type of bound (called "localized bound") in the Branch-and-Bound optimization of the myriad objective function, guarantees the delivery of the global minimum, thus assuring the robustness of the filter. At the same time this method offers a reduction in the filter's computational complexity. The localized bound technique can also be extended to be used in other nonlinear regression problems, such as non-Gaussian least squares, and generalized stable filters [16].

It is also important the fact that the algorithms introduced here are susceptible to additional optimizations that can keep reducing the time spent in the filtering operations. Examples of further improvements are the use of global search with a relaxed stop criteria to initialize local (and very fast) minimizers, and the initialization of Branch-and-Bound algorithms with the result of the previous filtering operation.

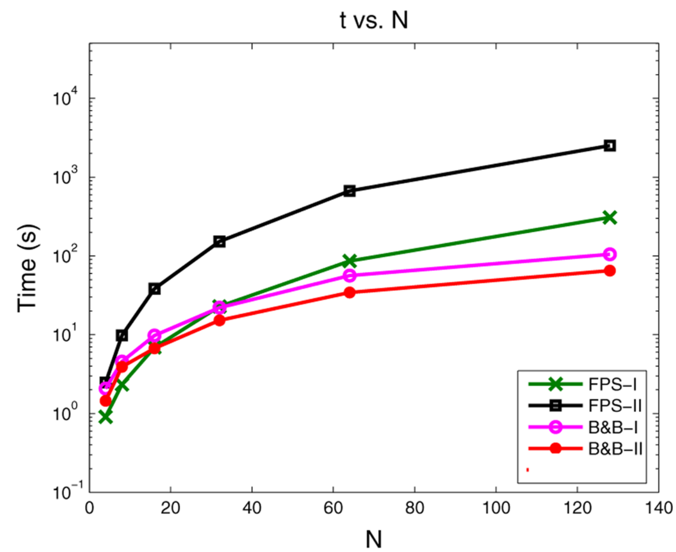

Fig. 7. Execution time of the myriad filter with the different minimization algorithms, for a test signal with $\alpha = 1.5$ and $\gamma = 1$.

Appendix I
Lipschitz Constant for B&B-Based Myriad Filtering

Let $f_i(\beta) = \rho(|w_i|(\operatorname{sgn}(w_i)x_i - \beta)) = \log(K^2 + w_i^2(\operatorname{sgn}(w_i)x_i - \beta)^2)$. Then $f_i'(\beta) = -(2w_i^2(\operatorname{sgn}(w_i)x_i - \beta))/(K^2 + w_i^2(\operatorname{sgn}(w_i)x_i - \beta)^2)$, $f_i''(\beta) = (2w_i^2)/(K^2 + w_i^2(\operatorname{sgn}(w_i)x_i - \beta)^2) - (4w_i^4(\operatorname{sgn}(w_i)x_i - \beta)^2)/((K^2 + w_i^2(\operatorname{sgn}(w_i)x_i - \beta)^2)^2)$, and $f_i^{(3)}(\beta) = (12w_i^4(\operatorname{sgn}(w_i)x_i - \beta))/((K^2 + w_i^2(\operatorname{sgn}(w_i)x_i - \beta)^2)^2) - (16w_i^6(\operatorname{sgn}(w_i)x_i - \beta)^3)/((K^2 + w_i^2(\operatorname{sgn}(w_i)x_i - \beta)^2)^3)$.

Solving $f_i''(\beta) = 0$, we get two solution $\beta_\pm = \operatorname{sgn}(w_i)x_i \pm (K)/(|w_i|)$.

Finally, since $f_i^{(3)}(\beta_+) = f_i^{(3)}(\operatorname{sgn}(w_i)x_i + (K)/(|w_i|)) = -(|w_i|^3)/(K^3) \leq 0$ and $f_i^{(3)}(\beta_-) = f_i^{(3)}(\operatorname{sgn}(w_i)x_i - (K)/(|w_i|)) = (|w_i|^3)/(K^3) \geq 0, \beta_+$ is a (local) maximum and β_- is a (local) minimum.

Now, let L_i be the Lipschitz constant of $\rho(\beta; x_i), i = 1, \ldots, N$ be

$$L_i = \frac{\partial}{\partial \beta}\rho(\beta)|_{\beta_+} \tag{8}$$

$$L_i = \frac{|w_i|}{K}. \tag{9}$$

A Lipschitz constant for the myriad filter is then given by

$$L = |\sum_{i=1}^{N} L_i| \tag{10}$$

which, using the triangle inequality, is bounded by

$$L = |\sum_{i=1}^{N} L_i| \leq \sum_{i=1}^{N} |L_i| \tag{11}$$

$$L \leq \frac{1}{K} \sum_{i=1}^{N} |w_i|. \tag{12}$$

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for helping improve the quality of the presentation of this paper. They also thank Robust Analysis, Inc. both for funding this research and for providing a stimulating environment for questioning the *status quo*. Fast, numerically efficient implementations of the Branch-and-Bound search algorithms introduced in this paper have been successfully integrated as part of the commercial release of Robust Analysis's STABLE Signal Processing Toolbox [16].

REFERENCES

[1] J. G. Gonzalez and G. R. Arce, "Statistically-efficient filtering in impulsive environments: Weighted myriad filters," *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 1, pp. 4–20, 2002.

[2] J. G. Gonzalez, "Robust techniques for wireless communnications in non-Gaussian environments," Ph.D. dissertation, Univ. Delaware, , 1997.

[3] G. R. Arce*, Nonlinear Signal Processing: A Statistical Approach*. New York: Wiley , 2005.

[4] E. Grün, M. S. Hanner, S. B. Peschke, T. Müller, H. Boehnhardt, T. Y. Brooke, H. Campins, J. Crovisier, C. Delahodde, I. Heinrichsen, H. U. Keller, R. F. Knacke, H. Krüger, P. Lamy, C. Leinert, D. Lemke, C. M. Lisse, M. Müller, D. J. Osip, M. Solc, M. Stickel, M. Sykes, V. Vanysek, and J. Zarnecki, "Broadband infrared photometry of comet Hale-Bopp with ISOPHOT," *Astron. Astrophys.*, vol. 377, pp. 1098–1118, Oct. 2001.

[5] M. E. Hassouni and H. Cherifi, "Spatio-temporal weighted myriad filter for alpha-stable noise removal in video sequences," in *ICSES04 Int. Conf. Signals and Electronic Syst.*, Poznan, Poland, Sep. 2004.

[6] J. G. Gonzalez, S. Manji, J. Paredes, and W. Ahmed, "Systems and methods for adaptively decoding transmitted frames," U. S. Patent Appl. 20030223512, Serial No. 161812, U.S. Class No. 375/316; 375/341, Int. Class No. H03K 009/00; H04L 027/06, Jun. 2002.

[7] R. Ganguli and V. P. Surender, "Adaptive myriad filter for improved gas turbine condition monitoring using transient data," in *ASME Turbo Expo 2004*, Vienna, Austria, Jun. 2004.

[8] A. Hamza and H. Krim, "Image denoising: A nonlinear robust statistical approach," *IEEE Trans. Signal Process.*, vol. 49, pp. 3045–3054, Dec. 2001.

[9] T. Pander, "Robust filtering of ECG signal with a myriad filter," *Polish J. Med. Phys. Eng.*, vol. 9, no. 1, pp. 17–30, 2003.

[10] R. C Nunez, "Novel techniques for the acceleration of non-Gaussian signal processing algorithms," M.S. thesis, Elect. Comput. Eng. Dept., Univ. of Delaware, Newark, DE, 2007.

[11] S. Kalluri and G. R. Arce, "Fast algorithms for weighted myriad computation by fixed-point search," *IEEE Trans. Signal Process.*, vol. 48, no. 1, pp. 159–171, 2000.

[12] S. Kalluri and G. R. Arce, "Adaptive weighted myriad filter algorithms for robust signal processing in alpha-stable noise environments," *IEEE Trans. Signal Process.*, vol. 46, no. 2, pp. 322–334, 1998.

[13] S. Kalluri, "Nonlinear adaptive algorithms for robust signal processing and communications in impulsive environments," Ph.D. dissertation, Univ. Delaware, , 1998.

[14] S. Kalluri and G. R. Arce, "Robust frequency-selective filtering using weighted myriad filters admitting real-valued weights," *IEEE Trans. Signal Process.*, vol. 49, no. 11, pp. 2721–2733, Nov. 2001.

[15] G. R. Arce, "A general weighted median filter structure admitting negative weights," *IEEE Trans. Signal Process.*, vol. 46, no. 12, pp. 3195–3205, Dec. 1998.

[16] User Manual for the STABLE 5.0 Library. Robust Analysis, Inc., 2007 [Online]. Available: http://www.robustanalysis.com

[17] J. D. Pinter, Global Optimization in Action. Norwell, MA, Kluwer Academic, 1996.

[18] R. Horst, P. Pardalos, and N. Thoai*, Introduction to Global Optimization*. Norwell, MA: Kluwer Academic, 1995.

[19] J. G. Gonzalez, D. L. Lau, and G. R. Arce, "Towards a general theory of robust nonlinear filtering: Selection filters," in *IEEE Int. Conf. Acoust., Speech Signal Process., ICASSP'97*, Munich, Germany, Apr. 1997, vol. 5, pp. 3837–3840.

On Kalman Smoothing With Random Packet Loss

Alex S. Leong, Subhrakanti Dey, and Jamie S. Evans

***Abstract*—This correspondence studies the performance of Kalman fixed lag smoothers with random packet losses and its comparison with the Kalman filter with packet loss. In terms of estimator stability via boundedness of the expectation of the error covariance, we show that smoothing does not provide any benefit over filtering. On the other hand, it is demonstrated that using a probabilistic notion of performance, smoothing can provide significant gains when compared to Kalman filtering. An analysis of Kalman filtering using two simple retransmission schemes and its comparison with Kalman smoothing is also made.**

***Index Terms*—Kalman filtering, Kalman smoothing, missing observations, retransmissions, stability.**

I. Introduction

Problems involving estimation over lossy communication networks have received considerable attention in recent years, due to their relevance in areas such as wireless sensor networks and networked control systems. When measurements from sensors are located at separate locations and have to be transmitted for processing through unreliable (e.g., wireless) channels, losses can occur, and how these packet losses affect the performance of the estimator is of significant interest.

Early work on state estimation with measurements losses include [1], where the optimal linear estimator for linear systems with independent identically distributed (i.i.d.) Bernoulli losses was derived, where the parameters of the loss process is known, but which of the individual measurements are lost/received is not explicitly known. This was later extended to the optimal linear smoother in [2]. More recently, in the case where we know which measurements are lost/received, it was shown in [3] that for an unstable system with i.i.d. Bernoulli losses there exists a critical threshold such that the expected value of the error covariance (which is randomly time varying due to the random losses) will be bounded if the packet arrival rate exceeds this threshold, but will diverge otherwise. Further avenues of research suggested in [3] include studying multiple sensors [4], correlated loss processes such as Markov [5], consideration of delays [6], and smoothing, which is the subject of

Manuscript received July 23, 2007; revised January 29, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sergiy A. Vorobyov. This work was supported by the Australian Research Council.

The authors are with the ARC Special Research Centre for Ultra-Broadband Information Networks, Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, VIC 3010, Australia (e-mail: a.leong@ee.unimelb.edu.au; s.dey@ee.unimelb.edu.au; j.evans@ee.unimelb.edu.au).

Digital Object Identifier 10.1109/TSP.2008.920470