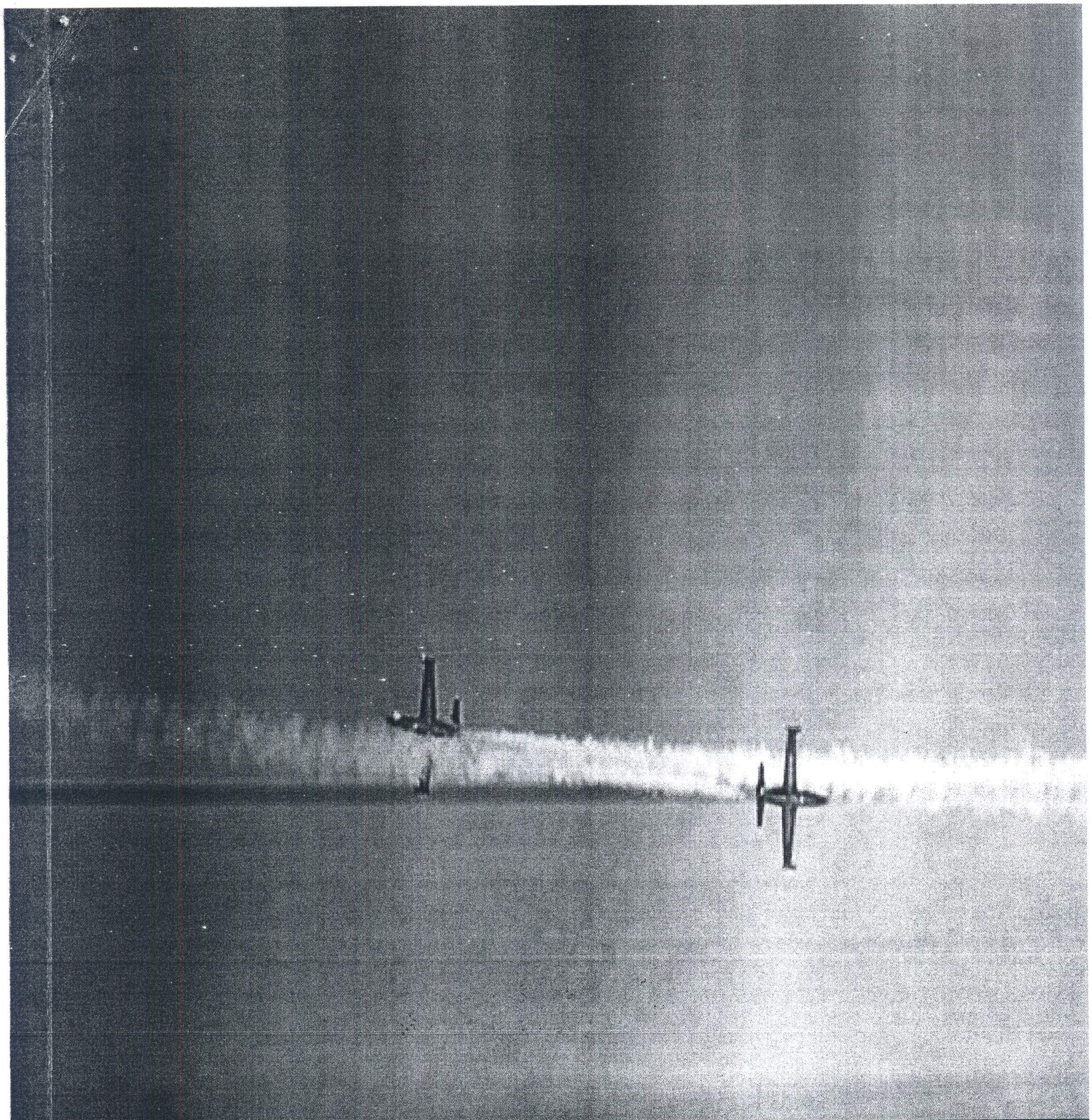

CBSEC 2011

1st Brazilian Conference on Critical Embedded Systems

1ª Conferência Brasileira de Sistemas Embarcados Críticos

São Carlos, Brazil
May, 2011

Evaluation of an embedded unscented Kalman filter for soil tomography

Marcos A. M. Laia[1], Paulo E. Cruvinel[2, 3]

[1]Physics Institute of São Carlos, University of São Paulo (USP), São Carlos, Brazil
marcoslaia@gmail.com
[2]Member of the Brazilian National Institute of Science and Technology on Embedded Critical Systems (INCT-SEC)
[3]Embrapa Instrumentation Center (CNPDIA), Rua XV de Novembro 1452, 13560-970 São Carlos SP, Brazil
cruvinel@cnpdia.embrapa.br

With the flexibility of using the Field Programmable Gate Arrays (FPGAs) is the possibility of using algorithms that were previously validated in general-purpose computers, as well as the use of most modern processors using proper management of energy expenditure and use of resources like memory and instruction cycles, and allows the use of parallel processing to eliminate bottlenecks and computational analysis of dynamic problems of the agricultural environment. The problem of estimating a noise-free signal involves foreknowledge of system variables, which is not always perfect due to the behavior of systems are influenced by the environment in which it is and the quality of the components of sampling equipment, and may be a optimal solution for control and measurement systems. Agricultural soil tomography aims at investigating soil proprieties as water and solute transport, soil porosity, soil contents, root growing and humidity. For a better analysis about these proprieties, an image quality is required. Previous works focused on image filtering or in the use of filters specialized in Gaussian process estimation and in an implementation in general purpose computers that have high processing power and memory but with a processing time depends on the resolution of the data to be acquired and the number of neurons in the neural network. This paper presents formulations for the use of unscented Kalman filter with neural networks in a joint estimation filtering (a filter for state and weight estimation) implemented in an embedded system with the objective of obtaining better quality in the signal / noise relation of the tomography projections. The aim of this work is to use the filter on a dedicated system as an invisible module to receive and filter the projections.

Index Terms — **Agriculture, Artificial Neural Network, Critical embedded systems, Kalman filter, Tomography.**

I. Introduction

Widely used in medical areas, the use of Computerized Tomography (CT) in soil science has been introduced by Petrovic, Siebert and Riek [1], Hainswoth and Aylmores [2] and by Crestana [3]. Petrovic has shown the possibility of using an X-ray computed tomography to measure the density of soil volumes, while Crestana has demonstrated that CT can solve problems related to studies of the physics of water in the soil. From these studies, it led to a project involving the development of a tomography to soil science [4] [5]. The use of the computer tomography is essential for the image reconstruction from projections.

The application of CT for the investigation of soil physics properties in grain and pore levels is important to the water and solute transport study in this environment, particularly in non saturate regions, as well as for the interaction investigations of soil and roots. Combined with other conventional techniques as neutrons probes, gravimetry, gamma and X-ray direct transmission, tracers, optical microscopy, electron microscopy scanning, mercury intrusion and other similar ones, it contributes greatly to resolve diverse problems of soil area. The results were obtained in a millimeter order scale, while various answers are expected in particle, macropore and micropore levels [6]. In the visualization of a tomographic image there is the presence of granularity, which is significant in the viewing of objects in low contrast. This granularity may be considered as a fake detail in image.

Besides, the use of X-ray computer tomography requires, as its application, the use of digital filters. They are necessary since the studied signal is represented discretely and due to the ability to treat an adaptive approach to promote a best filtering. One of them is the Kalman filter. This mathematical tool developed based on concepts such as (hidden) Markov chains [7], Bayesian estimation among others. It has the ability to obtain future and hidden states given the observation and to improve with the other techniques of estimation. In this paper is used as artificial neural networks, but can be applied either in genetic algorithms. These filters are seen as extensions of nonlinear filters and changes are made directly in the equations for filter measurement and correction.

The linear filtering main characteristic is the ability to make a prediction using a known linear function. For the discrete filter the translation matrix was used, where the difference from the future state and the current state is estimated. The observed value shall be the sum of these states after being corrected by the filter. The non-linear filtering can be made through the use of a nonlinear function for this estimation. This is done with the use of neural networks that promote a non-linear mapping and the use of the filter to estimate the neural network weights.

The techniques used in artificial intelligence and in estimation with Kalman filter are used to increase its filtering power to solve problems of higher orders. To determine the behavior of a function, it can use its own filter to perform a linear prediction or make a non-linear prediction using neural networks. Previous works focused on the use of Kalman filter in tomographic projections and presented the algorithm efficiency and quantifies the results using linear estimation or artificial neural networks [8] [9] [10].

The main source of noise in CT images is quantum mottle, defined as the spatial and temporal statistical variation in the number of X-ray photons absorbed in the detector. Other types of noise present in CT images are the rounding errors in the program of reconstruction (noise of the algorithm) and the electronic noise attributed by the system displays. Electronic noise can originate in not ideal electronic devices, such as not pure resistors and capacitors, not ideal terminal contacts, current leakage transistors, Joule and can also be independent of the signal, such as external interference (electrical or even mechanical) [11] [12] [13] [14].

The use of Kalman filter using the unscented transform for tomography projections is usually called Unscented Kalman filter (UKF) and it is presented with a solution with artificial neural networks (ANN) in dual estimation. The dual estimation consists in a two filters where one estimates the states and other estimates the weights for ANN [15]. For the application in embedded system, two filters may use a lot of system resources. Other problem is the filter stability. The dual estimation with UKF filter presents itself in a better stability than joint estimation, but with the use of square-root modification in UKF, the joint estimation become more reliable.

The unscented Kalman filter is similar to the extended version [16]. The distribution of states is represented by a Gaussian random variable, but is now specified using a minimum of sampling point sets chosen carefully. The sampled points capture the true mean and covariance of random variable and when it propagates through a truly non-linear system, it captures the mean and covariance accurately to promote a third order estimation for any nonlinearity. Thus, this is done through the use of unscented processing.

This method differs from the general methods of sampling (Monte-Carlo methods such as particle filters), which require orders of magnitude with more sample points in an attempt to define and propagate the state (possibly non-Gaussian) distributions. The unscented approaches result in more hits for the third order for Gaussian inputs for all nonlinearities.

The UKF don't need calculate Jacobian or Hessians matrices; in addition, the calculation total numbers are the same of extended filters related to nonlinear controls that require feedback from the states. In these applications the dynamic model is a physically based parametric model, which assuming is known.

Due to numerical instability related to the filter noise, and the use of the Cholesky factorization to determine the square root of probability matrix, Rudolph van der Merwe and Eric A. Wan have developed the square-root unscented Kalman filter (SRUKF) [17], which allows better control variance matrix values, bypassing the problem of becoming a negative or indefinite matrix.

With the knowledge of non-linear function of the process and a Kalman filter that supports non linear functions is possible to get a significant improvement in the signal. One solution is to use a neural network to promote a better function of the mapping process, reducing the noise present in the projections. For an estimation of the weights of the neural network together with the estimates of the states, we can use two methods of filtering: the joint estimation and dual estimation. These arrangements for determining the filtering initial weights are known, the next state is obtained in a linear mapping with the previous one.

Neural networks are composed of a network of parallel and distributed processing units interconnected in simple non-linear layer arrangements. Parallelism, modularity and dynamic adaptation are the three main characteristics typically associated with RNAs. An architecture based on reconfigurable computing can be well exploited to rapidly reconfigure and adapt weights and topologies. The use of a greater number of neurons is still a challenging task because the algorithms are rich in RNA and multiplications are really expensive to implement in an FPGA multiplier with fine granularity. Using the reconfigurability of FPGAs, there are strategies to implement ANNs in FPGA inexpensive and efficient.

Applications for the use of FPGAs is present in various areas include digital signal processing, medical imaging, computer vision, speech recognition and others areas. With the increase of their sizes, capacities and speeds, they began working with wider functions and are in a market for complete systems on processors. Applications can be found in any area and algorithms that can be used for use of parallelism are easily found to apply in its architecture. They are used in conventional high-performance computing where computational kernels such as FFT and convolution are performed on the FPGA instead of a microprocessor. The use of FPGAs for computing tasks is known as reconfigurable computing [18].

The use of Kalman filters in FPGAs is approached only in its extended version, not yet applied to filters in conjunction with neural network implementations or even newer ones as the Unscented Kalman filter that has a better efficiency than the filter functions in extended fully nonlinear not need linearization. The implementation costs are lower by not needing to be calculations of Jacobian and hessians who always demand high processing cost and not always a viable solution is found, and the use of memory is much reduced because it is an online learning where the number of neurons and layers can be very small compared with the filter off-line learning and training based on backpropagation.

FPGA is a semiconductor device containing logical components (logic blocks) and programmable internal connections. These blocks can be programmed to act as logic gates (AND, XOR) or complex functions that can be combined as decoders / encoders or mathematical functions. In most FPGAs, the logic blocks also include memory elements, which can range from simple flip-flops to complete blocks of memory [19].

The use of parallelism in the FPGA allows a considerable throughput even on a 500MHz clock below. The current generation can implement about 100 floating point units in a single clock cycle. The flexibility allows even higher performance by trading precision and range in the number format to a larger number of parallel arithmetic units [20]. The adoption of FPGAs in a high-performance computing is currently limited by the complexity of the project compared to conventional software and has a fairly long time to wait 4-8 hours after minor changes to source code.

Details of unscented Kalman filter implementations and modeling are shown in section 2. Section 3 presents a comparison of the results obtained by the filters. Finally, section 4 presents the conclusion.

II. METHODOLOGY

The equipment utilized is a CT mini-scanner tomography scanner developed at Embrapa Agricultural Instrumentation. The data acquisition process of the CT mini-scanner tomography provides a matrix with the sample values of projections. For the modeling process, it considers a matrix row that, by convention, is named sum ray. This signal is composed of various incidences with variable and non deterministic values, whose amplitude is given by

$$I_m[n] = I_0 e^{-\mu d}, \tag{1}$$

where d is the distance traveled by the photon ray within the evidence body, μ is the attenuation coefficient, Io is the free beam counting and I_m is the projection n attenuated beam.

In previous works, it were used two estimation states for linear filtering based in the discrete Kalman filter limitations. The UKF allow working with three estimation states leading a new filtering level. The new system equations are

$$\begin{bmatrix} P_{\theta_i}[n] \\ I_{m-1}\delta(n-m) \\ I_{m-2}\delta(n-m-2) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{m-1}\delta(n-m) \\ I_{m-1}\delta(n-m-1) \\ I_{m-1}\delta(n-m-2) \end{bmatrix} \tag{2}$$

$$P_{\theta_i}[n] = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{\theta_i}[n] \\ I_{m-1}\delta(n-m) \\ I_{m-2}\delta(n-m-2) \end{bmatrix} \tag{3}$$

The Kalman filter can be a nonlinear function and train parameters (weights). There is then the possibility of using a mapping function with neural networks where the filter trains the neurons and moves to a stable system where the weights are estimated and the mapping function has the lowest error rate possible. This filter allows working with higher orders (with the accuracy equivalent to the expansion of third order Taylor series), while the filter, in its extended form, works only with second order functions.

For a weight estimation of the neural network together with the states estimation, it is possible to use two methods of filtering: the state estimation and the dual estimation. The arrangements for determining the filtering initial weights are known, the next state is obtained in a linear mapping with the previous state. Thus, it has:

$$x_{k+1} = f(x_k, W_k, v_k) \tag{4}$$

Then, a Kalman filter to estimate the states and a Kalman filter to estimate the weights are used. This filtering allows the application in a system where the dynamics of status is unknown or chaotic (non-deterministic). Then it has a filtering system with joint estimation that can be written as

$$\begin{bmatrix} P_{\theta_i}[n] \\ W \end{bmatrix} = \begin{bmatrix} f(P_{\theta_i}[n-1], W, v_k) \\ W \end{bmatrix} \tag{5}$$

$$P_{\theta_i}[n] = h(P_{\theta_i}[n], n_k) \tag{6}$$

where W is the weight for the ANN.

The Kalman filter was developed in the MATLAB environment that allows you to use functions developed and optimized for mathematical calculations, avoiding bottlenecks and better memory management. MATLAB (Matrix Laboratory) is high performance interactive software aimed at the numerical calculation. MATLAB integrates numerical analysis, calculation with arrays, signal processing and building graphics easy to use environment where problems and solutions are expressed just as they are written mathematically, unlike traditional programming [21]. MATLAB is an interactive system whose basic element is an array of information that does not require scaling. This system allows the numerical solution of many problems in only a fraction of the time it takes to write a similar program in FORTRAN, Basic or C. Moreover, the solutions of problems are expressed almost exactly as they are written mathematically.

To simulate an input used to read a system variable (a vector) with the projection of data stored. The algorithm is an adapted function embedded in Simulink as a block. This block has only inlet and outlet. Despite being more practical, such a function could appear closed several bottlenecks, but also causes some functions and operations in MATLAB code run more efficiently in the translation to HDL due to the logical blocks are already optimized for this translation, as for example, the blocks designated by the manufacturer of the FPGAs. From this reading of the data, the algorithm does the filtering of data and stores it in another variable vector of filtered projections.

To design an FPGA, is necessary to configure it (or schedule it), choosing as the chip will work with a logic circuit diagram or a source code using HDL (hardware description language). The HDL enables easy handling large structures work because you can specify them numerically rather than having to draw them by hand all the parties. Moreover, the schematic entry allows a closer specification required.

The purpose of translating the code in MATLAB for a similar code in HDL using the tool itself in this MATLAB's Simulink. Simulink is a tool for modeling, simulation and analysis of dynamic systems. Its primary interface is a graphical diagramming tool for blocks and blocks of customized libraries. The software offers tight integration with the rest of the MATLAB environment. Simulink is widely used in control theory and digital signal processing design and simulation for multi-domains. Along with Real-Time Workshop, the Simulink can also automatically generate C code for implementing real-time systems. With increased efficiency and flexibility of this tool, it is increasingly adopted in production systems and embedded systems design for its flexibility and rapid iteration. The code created by Real-Time Workshop is efficient enough to be used in embedded systems.

The source files can be saved in the FPGA through software developed by the manufacturer through the various stages that produce a file. This file is downloaded to the FPGA through a serial interface or an external memory device such as an EEPROM.

III. RESULTS AND DISCUSSION

As the Kalman filter works on a white noise process it takes up the value of variance Q, which determines the degree of confidence in the process while the value of variance R determines the degree of noise system. To measure the quality of filtering, different types of phantoms and soil samples were used.

First, it was used calibration phantoms to test the filters performance and compare the results of joint estimation with the linear estimation of three states. The phantoms used were homogeneous, heterogeneous with several different substances (four different substances in addition to the Plexiglas) and another type with a different material from Plexiglas. A fourth sample was prepared with sand grains placed in the Plexiglas. The last three samples were taken from different soil types.

The comparison between the filters is required to validate the power of estimating the noise-free signal with the use of ANNs against the use of linear estimation.

The results were generated using data already present in the form of files. The filter was used in MATLAB, but was validated using the MATLAB's Simulink (Figure 1). Data were normalized to 0 and 1, because the activation function of hidden layer ANN is a sigmoid function, so values between 0 and 1 are the recommended range for this type of network.

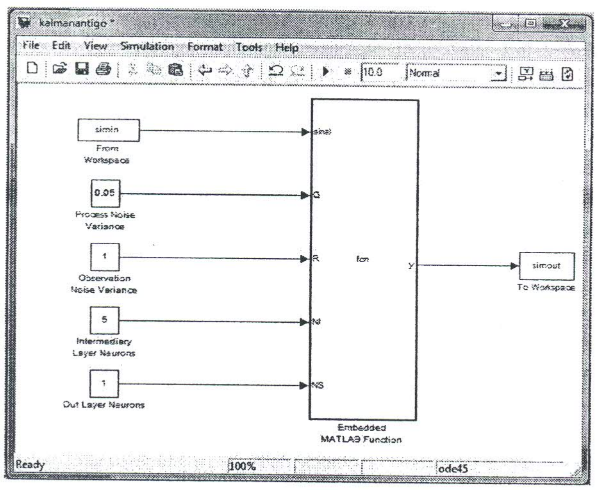

Figure 1 – Implementation using MATLAB's Simulink. The filter is being implemented, primarily, using embedded MATLAB function to analyze the type of code to be generated in HDL. After this stage, it will use the blocks provided by MATLAB and the FPGA manufacturer. These blocks are important because they will control the flow of data from the tomography scanner and organize them, generating a file with the data collected and filtered.

For filtration of phantoms was always used for the value 1 for the variance R. Since the Q value varies with the degree of filtration required. The variance Q determines the proximity between the projections. A high value that will be recognized as the noise signal, while a low will cause there is an excessive smoothing. For the calibration phantoms, we used the values of Q equal to 0.1 and 0.5 for the linear filtering and filtering with ANN respectively. For phantoms with soil samples, 0.001 and 0.05 were used because they present a different and due the larger amounts of data between intervals because theses phantoms having been acquired in a micro tomography.

For each entry in the filter, the signal was doubled because the sample size is small for the total convergence of filter and to remove estimation errors during the learning phase of the filter. In the end, we applied the algorithm to leave the RTS filtering more homogeneous final value of the original. In the end, all the filtered projections are meeting again in matrix form and is performed to image reconstruction. Due to the reconstruction filter applied, distortion may occur.

The results obtained by applying the filter to the calibration phantoms can be seen in Figure 2.

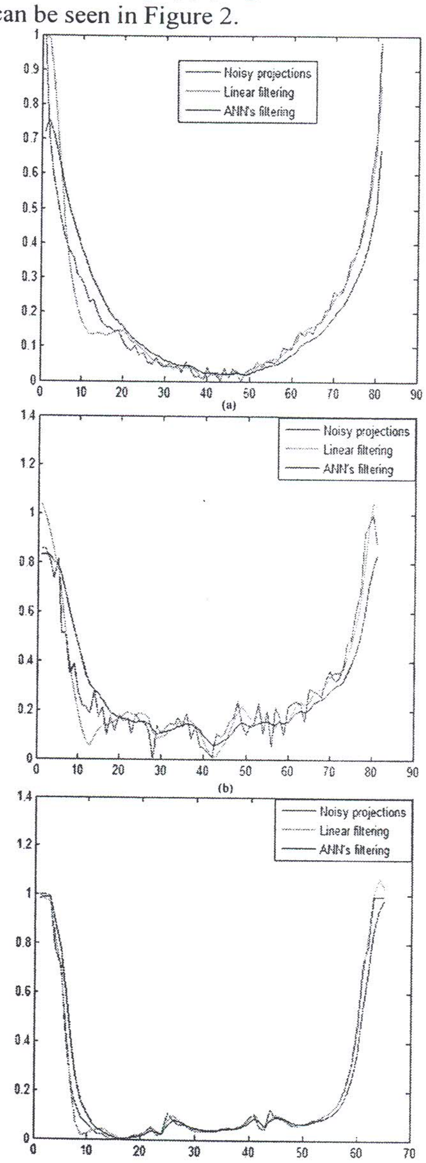

Figure 2 – Results using the SRUKF in calibration phantoms: (a) homogenous phantom, (b) heterogenous with a substance in Plexiglas (c) heterogenous with five substances.

The linear filtering experienced an estimation error that can be observed between the projections 10 and 20. The estimation is incorrect due inability to learn from past behavior data. Another important fact is noted by the filter always get the maximum values and understand them as filtered signal. Filtering with ANN is possible to verify that the estimation error was minimal and the signal is filtered between the peaks of maximum and minimum.

To get a better view of the noise affecting the system, it was used a sample prepared in the laboratory with sand grains. In it can clearly see the noise as spikes upward or downward. Although discrete, these noises are quite confused with the signal. The linear estimation filter ends up accepting these peaks as noise-free signals while the joint estimation with ANN can keep a more accurate estimation. The results from this sample can be seen in Figure 3.

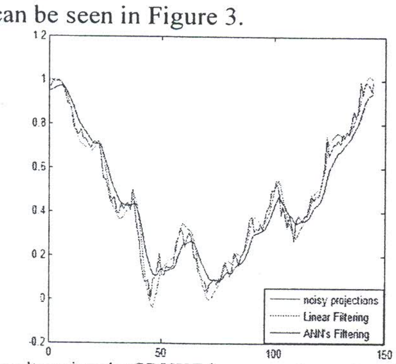

Figure 3 – Results using the SRUKF in prepared sample with sand grains.

The phantom was used that contained sand grains to determine the best values of Q and R for the other sample of micro tomography. As it has well defined grains, was chosen values that ensured a better image, clean and detail present. Applying these values to the soil samples were obtained the following results found in Figure 4.

After filtering the projections, the images were reconstructed with backprojection algorithm. The generated images are displayed in figure 5.

In the images it's possible to see unique detail richness, but a large noise presence in granular form too. In a soil sample, a set of grain in dark colors can be confused with micropores. A presence of grain in the lighter shades can be confused with material presence, hiding the micropores. In the images reconstructed using the linear estimation filter with one senses a strain at the bottom of the samples. This defect is provided by the estimation error that ends up leaving the images with different tones. In the reconstructed images where the filter was applied in the joint estimation, the tones are presented in a uniform manner. As in other figures, one can see the errors of estimation of the linear filter and the choice of values where the peaks of the samples. The UKF in joint estimation presented with a best estimate, always following the behavior of the noise-free signal and not the noisy signal.

The results were as expected and have, to some extent, the same visual results for the implementation of unscented and basic filters, proving that the characteristics of Poisson noise can be mapped by a neural network, where the complexity to understand the process of equations deeply can be replaced by an iterative intelligent system, able to find new sensor features over time (difference in temperature, equipment aging and new mechanical structures).

The results with soil sample show the joint estimation can reach a better image due to the presence of image details and no blurring effects in the sand grains.

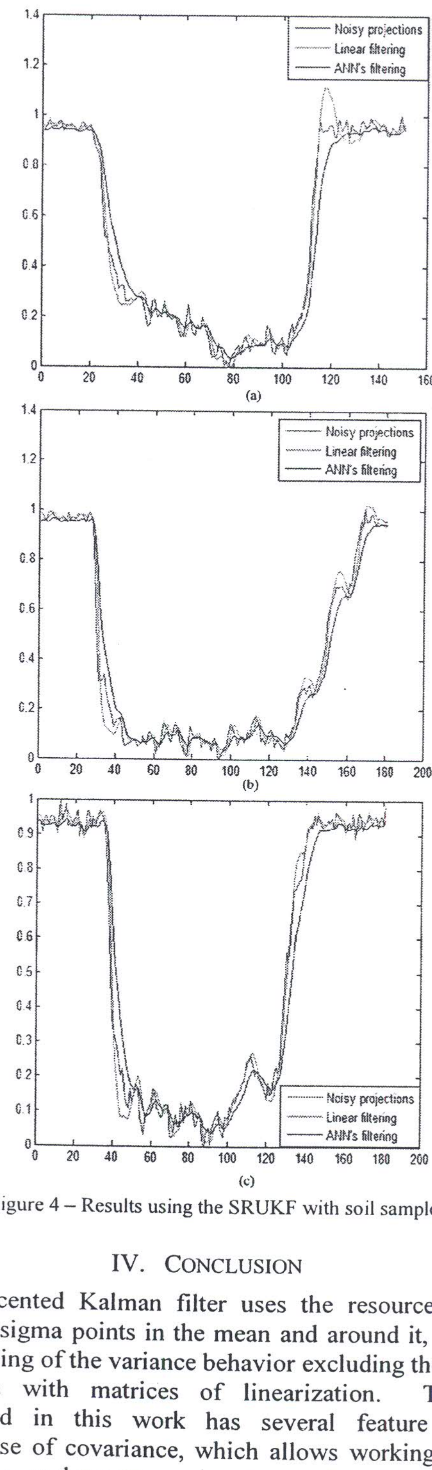

Figure 4 – Results using the SRUKF with soil samples.

IV. Conclusion

The unscented Kalman filter uses the resources for the creation of sigma points in the mean and around it, making a better mapping of the variance behavior excluding the need for calculations with matrices of linearization. The filter implemented in this work has several feature clusters: Increased use of covariance, which allows working with the signal, noises and process and system noise variances at the same time, allowing noise estimation, something that does not happen with the other Kalman filters; Use the type of filter to square root, which using the Cholesky factoration allows greater stability of the filter concerning the noise and a gain in the filter order; Freedom to use the algorithm without the need for a priori knowledge of the response functions; Accuracy equivalent to third-order functions without the need for neural networks.

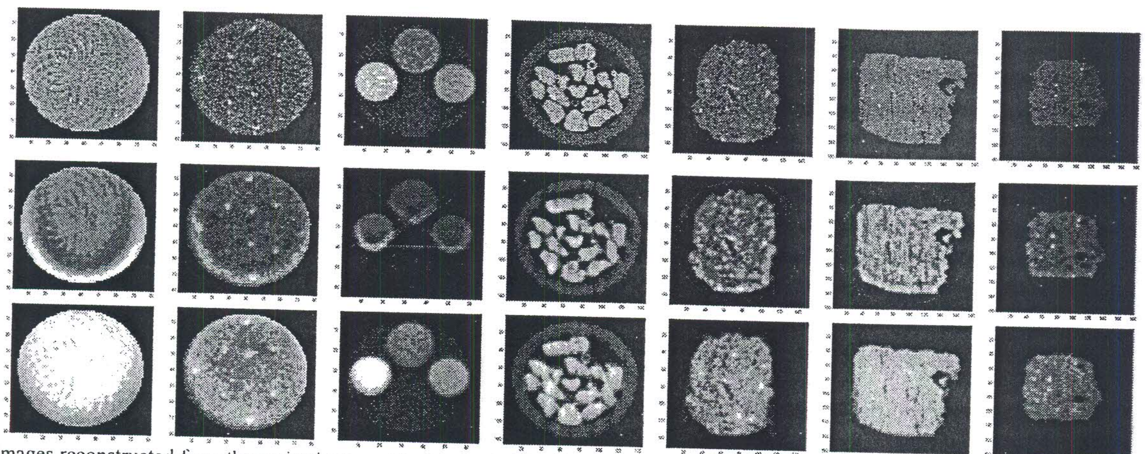

Figure 5 – Images reconstructed from the projections used in this work: The first line is displayed the original projections. In the second row, it has been used the SRUKF in linear estimation. In the third row, it has been used SRUKF with ANN in a joint estimation.

The use of neural networks with this filter type allowed mapping any function, with precise estimation of results. Additionally, by checking the results of the unscented Kalman filter with neural networks in phantoms, it was possible to observe the efficiency of the filter to adapt to the chaotic features, such as heterogeneities normally present in real samples. As a pre-filtering, maintaining details in an image should be the most important objective. Nevertheless, the code for some functions such as complex Cholesky factorization and matrix multiplication are already optimized by the MATLAB. Some functions are transformed into C later in HDL, which can cause new bottlenecks,. but due to its complexity, is perhaps not as efficiently as possible rewrites them in HDL. Although the system is a quote that reads data directly from memory, the system being implemented in the FPGA is expected to accumulate the projections in an array, filter them and generate an array with all the projections ready to be rebuilt. In a simulation environment, the performance of the filter has shown equivalent to the code generated in MATLAB. The algorithm still has some bottlenecks that will be optimized with the introduction of some code in HDL generated for the specialized Simulink blocks.

ACKNOWLEDGMENT

This work was supported in part by the National Council for Research and Development (CNPq) under Grant 306988/2007-0 and Brazilian Enterprise for Agricultural Research (Embrapa) under Grant 03.10.05.011.00.01.

REFERENCES

[1] A. M. Petrovic, J. E. Siebert, P. E. Rieke, "Soil bulk analysis in three dimensions by computed tomographic scanning", *Soil Sci. Soc. Am. J.*, n.46, p.445-450, 1982.

[2] J. M. Hainsworth, L. A. G. Aylmore, "The use of the computed-assisted tomography to determine spatial distribution of soil water content", *Aust. Journal Soil Res.*, N. 21, p.1435-1443,1983.

[3] S. Crestana, "A Tomografia Computadorizada com um novo método para estudos da física da água no solo", *São Carlos, USP, 140 p.*, 1985.

[4] P. E. Cruvinel, "Minitomógrafo de raios X e raios gama computadorizado para aplicações multidisciplinares". *Campinas, UNICAMP, 329 p.*, 1987.

[5] P. E. Cruvinel; R. Cesareo; S. Crestana; S. Mascarenhas, "X and γ-ray computerized minitomograph scanner for soil science", *IEEE Transactions on Instrumentation and Measurement*, V.39, N.5, p.745-750, October, 1990.

[6] S. Crestana, "Técnicas recentes de determinação de características do solo.", *Reunião brasileira de manejo e conservação do solo e da água*, vol. 10, 1994, Abstracts, Florianópolis: Sociedade Brasileira de Ciência do Solo, p86-97, 1994.

[7] R. E. Kalman,"A new approach to Linear Filtering and Prediction Problems." *Transaction of the ASME - Journal of basic Engineering*, 1960.

[8] M. A. M. Laia, P. E. Cruvinel, "Filtragem de projeções tomográficas utilizando Kalman Discreto e Rede Neurais", *IEEE América Latina,vol. 6, ed. 1, march*, 2008.

[9] M. A. M. Laia, P. E. Cruvinel, A. L. M. Levada, "Filtragem de projeções tomográficas da ciência do solo utilizando transformada de Anscombe e Kalman", *DINCON'07, São José do Rio Preto*, 2007.

[10] M. A. M. Laia, P. E. Cruvinel. "Filtragem de projeções tomográficas do solo utilizando Kalman e Redes Neurais numa estimação conjunta", *DINCON'08, Presidente Prudente*, 2008.

[11] A. J. Duerinckx, A. Macocski, "Polychromatic Streak Artifacts in Computed Tomography Images", *J.Comput. Assist. Tomogr., 2.481*, 1978.

[12] P. M. Joseph, R. D. Spital, "A method for correction bone-induced artifacts of CT scanners", 1978.

[13] G. S. Ibbott, "Radiation therapy treatment planning and the distortion of CT images", *Med. Phys, 7:261*,1980.

[14] L. F. Granato, "Algoritmo adaptativo para a melhoria em imagens tomográficas obtidas em múltiplas energias", *São Carlos, UFSCar, 135 p.*, 1998.

[15] M. A. M. Laia, P. E.Cruvinel, "Applying an Improved Square Root Unscented Kalman Filtering in Tomographic Projections of Agricultural Soil Samples", *Vetor-Revista de Ciências Exatas e Engenharias*, p.17-31, vol. 18, n. 1, 2008.

[16] S. J. Julier, J. K. Uhlmann, "A new extension of Kalman filter to nonlinear systems." *Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997.

[17] R. van der Merve, E. A. Wan. "The square-root unscented Kalman Filter for state and parameter-estimation." *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, 2001: 3461-3464.

[18] Field-programmable gate array. Accessed in April 12, 2011. http://en.wikipedia.org/wiki/Field-programmable_gate_array.

[19] FPGA - Field Programmable Gate Array and other programmable devices. Accessed in April 12, 2011. http://knol.google.com/k/fpga-field-programmable-gate-array

[20] R. Ayoubi, J. Dubois, R. Minkara. "FPGA Implementation of Generalized Maximal Ratio Combining Receiver Diversity". *World Academy of Science*, Engineering and Technology vol. 68 2010.

[21] J. J. Cai. "Evolutionary Bioinformatics with a Scientific Computing Environment" Texas A&M University, USA