SENSOR FAULT DIAGNOSIS USING PRINCIPAL COMPONENT ANALYSIS

A Dissertation

by

MAHMOUDREZA SHARIFI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2009

Major Subject: Mechanical Engineering

SENSOR FAULT DIAGNOSIS USING PRINCIPAL COMPONENT ANALYSIS

A Dissertation

by

MAHMOUDREZA SHARIFI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Reza Langari
Committee Members,	Shankar Bhattacharyya
	Richardo Guttierrez-Osuna
	Alexander Parlos
Head of Department,	Dennis O'Neal

December 2009

Major Subject: Mechanical Engineering

ABSTRACT

Sensor Fault Diagnosis Using Principal Component Analysis. (December 2009)

Mahmoudreza Sharifi, B.S. Sharif University of Technology;

M.S., University of Tehran

Chair of Advisory Committee: Dr. Reza Langari

The purpose of this research is to address the problem of fault diagnosis of sensors which measure a set of direct redundant variables. This study proposes:

1. A method for linear senor fault diagnosis
2. An analysis of isolability and detectability of sensor faults
3. A stochastic method for the decision process
4. A nonlinear approach to sensor fault diagnosis.

In this study, first a geometrical approach to sensor fault detection is proposed. The sensor fault is isolated based on the direction of residuals found from a residual generator. This residual generator can be constructed from an input-output model in model based methods or from a Principal Component Analysis (PCA) based model in data driven methods. Using this residual generator and the assumption of white Gaussian noise, the effect of noise on the isolability is studied, and the minimum magnitude of

isolable fault in each sensor is found based on the distribution of noise in the measurement system.

Next, for the decision process a probabilistic approach to sensor fault diagnosis is presented. Unlike most existing probabilistic approaches to fault diagnosis, which are based on Bayesian Belief Networks, in this approach the probabilistic model is directly extracted from a parity equation. The relevant parity equation can be found using a model of the system or through PCA analysis of data measured from the system. In addition, a sensor detectability index is introduced that specifies the level of detectability of sensor faults in a set of redundant sensors. This index depends only on the internal relationships of the variables of the system and noise level.

Finally, the proposed linear sensor fault diagnosis approach has been extended to nonlinear method by separating the space of measurements into several local linear regions. This classification has been performed by application of Mixture of Probabilistic PCA (MPPCA).

The proposed linear and nonlinear methods are tested on three different systems. The linear method is applied to sensor fault diagnosis in a smart structure and to the Tennessee Eastman process model, and the nonlinear method is applied to a data set collected from a fully instrumented HVAC system.

DEDICATION

To My Dear Parents

and

To My Lovely Wife, Faezeh

ACKNOWLEDGEMENTS

My deep appreciation goes to many people whose advice, assistance and encouragement have enabled me to get to this stage in my life. I am really fortunate to have met so many great people in my life. To those who are missing in this brief list, and were supposed to be here, I sincerely apologize.

First, I wish to express my sincere thanks to my advisor, Prof. Reza Langari. His knowledge, valuable guidance, and unlimited patience inspired the completion of this dissertation. His encouragement, understanding and willingness to spend his precious time with me are beyond my appreciation.

I express my sincere gratitude to the members of my graduate study committee: Prof. Shankar Bhattacharyya, Prof. Alexander Parlos, and Prof. Ricardo Gutierrez-Osuna for serving on my committee and for their valuable advice and help through the years I spent at Texas A&M University. I would like to thank Prof. Radu Stoleru for graciously accepting to read my dissertation and attending my final exam, and Prof. Bryan Rasmussen for generously providing me with valuable experimental data from facilities in his research laboratory.

I also like to thank Dr. Yeesock Kim for his help and support and during the writing my dissertation. His helpfulness and ultimate generosity have been always big sources of

inspiration. I also thank my great friend, Dr. Ali Sadighi, for his sincere help especially during my stay in Qatar.

I also acknowledge the Mechanical Engineering Department of Texas A&M University for providing me with excellent academic experiences which were essential to the accomplishment of my graduate study.

I do not have the words to express my gratitude to my parents for their endless love, care, and support. Finally, I would like to especially thank to my lovely wife, whose patience and compassion during the final stages of my study, was priceless.

TABLE OF CONTENTS

Page

CHAPTER Page

Page

LIST OF FIGURES

Page

Page

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION TO SENSOR FAULT DIAGNOSIS SYSTEMS

With the recent advances in safety and reliability standards and demands for reduction of *maintenance* cost, the importance of online fault detection has significantly increased [Wang et al. 2009][Isserman 2005][Nandi et al. 2005]. A subcategory of online fault detection, which is the subject of this study, is online detection of sensor faults and measurement errors. This problem is very similar to the general kinds of problems which are called *estimation of missing data* [Muteki et al. 2005] but more specifically directed at sensor technology, and is usually referred to as *online sensor monitoring, gross error detection*, *sensor data validation,* or *sensor fault diagnosis* (SFD). The latter term will be used in this research.

Sensor fault diagnosis is mainly applicable in industrial facilities, such as nuclear power plants [Gross et al. 1997][Dorr et al. 1997][Hines et al. 1998] and [Hines and Davis 2005], HVAC systems [Rossi and Braun 1993][Namburu et al. 2007][Yang et al. 2008],and aerospace systems where a large number of sensors are used for control and diagnostic purposes. In these systems, usually after several operational cycles, sensors go out of calibration. This causes loss of performance of the overall system. In more critical

---

This dissertation follows the style of *AIChE Journal*.

systems, such as aircraft engines, loss of accuracy of these sensors may even cause catastrophic failure [Duane et al. 1998]. The traditional method for alleviating this problem is periodic monitoring and replacement of sensors, which usually requires shutting down the entire system. Therefore the main advantage of SFD is to shift away from periodic maintenance to more efficient condition based maintenance (CBM).

All methods of SFDs are based on some type of redundancy in the measurements. When several sensors are used to measure a single variable of the system, we have *physical redundancy* and the task of SFD is much easier. However, application of multiple sensors is not possible in many systems due to design limitations. Moreover, increasing the number of sensors results in a more costly system and further increases the cost of maintenance because added sensors need to be maintained as well.

Another form of redundancy is *Analytical redundancy* which is based on inherent relationship between the variables of the system. This relationship can be found either by the physical model of the system in so called *model based* methods, or it may be derived based on a database gathered from a history of measurements from the system in *data driven* methods.

Analytical redundancy itself has two categories: (1) static or direct redundancy (2) dynamic or temporal redundancy. The direct redundancy methods use algebraic relationship between the measured data, regardless of their dynamic behavior. It is the most popular method for SFD in large industries such as power plants and large HVAC systems that use multiple redundant sensors.

Dynamic redundancy methods, on the other hand, consider the dynamics of the system and use estimation theories to monitor the measurements of the system. Dynamic redundancy methods of SFD are in fact a part of general fault diagnosis which has a complete theoretical background in model based methods [Frank and Ding 1997].

Data driven SFD methods were mainly initiated after Autoassociative neural network (AANN) (as a nonlinear principal component analysis) method was proposed [Kramer 1991]. The vast majority of data driven applications of SFD have used AANN as a basic tool [Guo and Musgrave 1995] [Hines et al. 1996] [Moller 1998][Guo et al. 1996][Wang and Cox 2004] [Hoffman and Kimble 2005]. However as reported by a number of researchers, there are many problems with using AANN in practice [Malthouse 1998] [Najafi et al. 2004] [Sharifi et al. 2004]. Since AANN is basically considered a principal component analysis (PCA) method, in this research these problems will be analyzed and classified into two separate or distinct categories. The first problem is that even a perfect method of PCA, without appropriate complementary algorithms, does not always work for SFD [Dunia et al. 1996]. The second problem is due to the inefficiency of AANN as a NLPCA tool as investigated by Malthouse [1998] and Moghaddam [2002]. Therefore it is necessary to study both of these issues (1) finding how to use principal component analysis, either linear or nonlinear, to reconstruct the faulty data, and (2) developing an effective NLPCA algorithm which is suitable for SFD.

Study of Linear SFD Using PCA

PCA is the main tool for multivariate statistical process monitoring [Kresta et al. 1991][Wise and Ricker 1991] [MacGregor et al. 1991]. Typically, monitoring charts based on Hotelling T^2 and the Q-Statistic, derived from the PCA analysis measured values, are used to detect faults using the concept of a contribution plot. In a contribution plot, the contribution of each individual variable to the Q-statistic and Hotelling T^2 can be obtained and the variable having the largest contribution reflects the potential fault source [Miller et al. 1998]. Dunia and his colleagues investigated the application of PCA (linear) for SFD [Dunia et al. 1996]. They proposed a sensor validity index (SVI) to determine the status of each sensor. SVI is a number between zero and one. The closer this number is to 0, the more probable that the sensor is faulty. Accurate study of SVI shows that it is basically equivalent to the estimation of the value of a single sensor using its linear relationship with the rest of the sensors in the system. Therefore, if we have n sensors in the system, in order to evaluate the validity of the measurement in each sensor, we need to solve a set of $n-1$ linear equations. This problem is more troublesome when we want to extend this method to the nonlinear case. In fact in the linear case, as we will see in the next section, the relationship between the sensors is easily found with the PCA method, but this is not the case in the nonlinear case. Two alternative approaches will be presented in this research. The first approach is to use the parity equation and find the faulty sensor based on the direction of residuals to form an index called Sensor Failure Index (SFI). The second approach is to use the distribution of residuals to form the conditional probabilities of error in each sensor given the value of the residuals.

Study of Nonlinear PCA

PCA has diverse application in machine learning and data compression. Therefore much research in this area has been performed to find a nonlinear version of PCA (NLPCA). Apart from AANN which was mentioned earlier; Principal Curves [Hastie 1984], Kernel PCA [Scholkopf et al. 1998], Mixture of Probabilistic PCA [Tipping and Bishop 2002], Input Training Neural Networks [Zhao and Xu 2004] and Gaussian Process Latent Variable Model [Lawrence 2005] are some of the suggested methods for implementing (and/or using) NLPCA. Some of these methods such as Kernel PCA (KPCA) and Principal Curves and have been used for process monitoring [Harkat et al. 2003][Cho et al. 2005] but only AANN and Kernel PCA have been used in SFD problems [Cho et al. 2004][Kramer 1992]. In this research, first the method of KPCA will be analyzed and a new method which is a combination of KPCA and Neural Networks is presented. Also a complete new algorithm for NLPCA will be presented which also uses properties of KPCA as will be explained in the next section.

In summary, this research is to pursue three fundamental tasks. First, a framework for data driven SFD is presented. This part of the research is to answer a basic question apart from the methodologies and algorithms used, i.e. whether it is logically possible to validate the value of a sensor based on the readings from other sensors in the system. The second task is to analyze one of the most common methods for SFD, which is the application of AANN [Kramer 1991]. This analysis will specify the conditions under which we are able to have a successful SFD algorithm using AANN. The final task is to

present an effective method for the SFD problem which will incorporate the state-of-the-art machine learning techniques.

Technical Background

Most data driven methods for SFD are nonlinear, i.e. it is assumed that there is a nonlinear correlation between the variables measured by the sensors. This is expected because most real systems have nonlinear behavior. Usually, an NLPCA algorithm is used in these methods. However, since there is no perfect algorithm for NLPCA, effectiveness of SFD is hindered by the inefficiency of existing NLPCA algorithms. With this in mind, in order to have a clear understanding of the limitations which are *not* related to inefficiency of NLPCA method, the first part of this research focuses on the linear methods for SFD. That is because orthogonal decomposition, which is used in linear PCA, can be considered as a perfect PCA method. As an introduction to this subject, the nonlinear PCA method of AANN is first introduced in Section 2.1. It is also explained how this method is used for SFD. In Section 2.2 the problem of linear SFD is explained and it is graphically demonstrated why PCA requires complementary algorithms to be used for sensor fault diagnosis. Finally, in Section 2.3 the KPCA method is explained.

Introduction to Linear Approaches

Before we study the 3 conditions of AANN introduced in 2.1, we need to investigate whether these conditions are satisfied under linear conditions. The linear form of AANN

is the general principal component analysis (PCA). The method of orthogonal decomposition or discrete Karhunene-Loeve transform, which is used in PCA, finds and orthonormal transformation of a set of correlated variables such that these variables have no correlation to each other in the new space.

Induction to the Method

Assume $\mathbf{Y} \in \boldsymbol{\mathcal{R}}^{m\times n}$ is a matrix of measured values from a set of n healthy sensors which are defined in the following form

$$\mathbf{Y} = [\mathbf{y_1} \quad \mathbf{y_2} \quad \cdots \quad \mathbf{y_n}], \tag{1}$$

where $\mathbf{y_i} \in \boldsymbol{\mathcal{R}}^{m}$ is the history of m measurements from the i^{th} sensor. We also define the k^{th} row of matrix $\mathbf{Y}$, $\boldsymbol{y}^{<k>} = [\mathrm{y}_1^{<k>} \quad \mathrm{y}_2^{<k>} \quad \ldots \quad \mathrm{y}_\mathrm{n}^{<k>}]$ which is the k^{th} observation from the set of n sensors.

If we use $\mathbf{T}$ to show the transformed form of the given data, we can write the following equation

$$\mathbf{Y} = \mathbf{T}\mathbf{U}^{\mathrm{T}} \tag{2}$$

where the matrix **U** is the PCA transformation matrix. The data in **T** are sorted according to their variance such that the first column has the maximum variance. Now if we decompose this matrix into a high variance part **X**, and a low variance part **N**, we have:

$$\mathbf{Y} = [\mathbf{X} \,\vdots\, \mathbf{N}] \begin{bmatrix} \mathbf{P}^{\mathrm{T}} \\ \mathbf{Q}^{\mathrm{T}} \end{bmatrix} = \mathbf{X}\mathbf{P}^{\mathrm{T}} + \mathbf{N}\mathbf{Q}^{\mathrm{T}} \tag{3}$$

where $\mathbf{U} = [\mathbf{P} \quad \mathbf{Q}]$. The first part, $\mathbf{X}\mathbf{P}^{\mathrm{T}}$, can be expressed as a latent variable model and the second part, $\mathbf{N}\mathbf{Q}^{\mathrm{T}}$, is the modeling error. It is important to notice that the dimension of **X** is m × r, where $r < n$. In other words, the space of data has been reduced from $\boldsymbol{\mathcal{R}}^n$ to $\boldsymbol{\mathcal{R}}^r$

Once we find the PCA transformation, **P**, we can transform any set of new measurements into the space of high variance variables or principal spaces,

$$\boldsymbol{x} = \boldsymbol{P}\boldsymbol{y}, \tag{4}$$

where $\boldsymbol{y} = [y_1 \;\; y_2 \;\; \ldots y_n]$ is a variable representing a set of measurement from this system. We can also simply regenerate the data from the latent variable space,

$$\hat{\boldsymbol{y}} = \boldsymbol{P}^T \boldsymbol{x}\,, \tag{5}$$

where $\hat{\boldsymbol{y}} = [\hat{y}_1 \quad \hat{y}_2 \quad \ldots \quad \hat{y}_n\,]$ is the estimated values of measurements after converting them into the latent variables space. Now if we combine these two transformations together, we find the relationship between the measured values and estimated values based on the existing linear relationship of the sensors. We will name this linear transformation, from measured values to estimated values, as PCA filtering,

$$\hat{\boldsymbol{y}} = \mathbf{P}^{\mathrm{T}}\mathbf{P}\boldsymbol{y}\,. \tag{6}$$

The value of the residue between the measurements and the estimated values is simply

$$\boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \boldsymbol{y} - \mathbf{P}^{\mathrm{T}}\mathbf{P}\boldsymbol{y} = (\mathbf{I}_{n\times n} - \mathbf{P}^{\mathrm{T}}\mathbf{P})\boldsymbol{y}\,. \tag{7}$$

We will show that, in general, PCA satisfies the identity and convergence condition but not the isolability condition. However, in some specific conditions PCA might also satisfy all three conditions. Only under those conditions PCA is usable for SFD.

This is graphically explained in Figure 1. This figure shows a 3-dimensional graphical representation of a set of 3 sensors which are linearly correlated to each other.

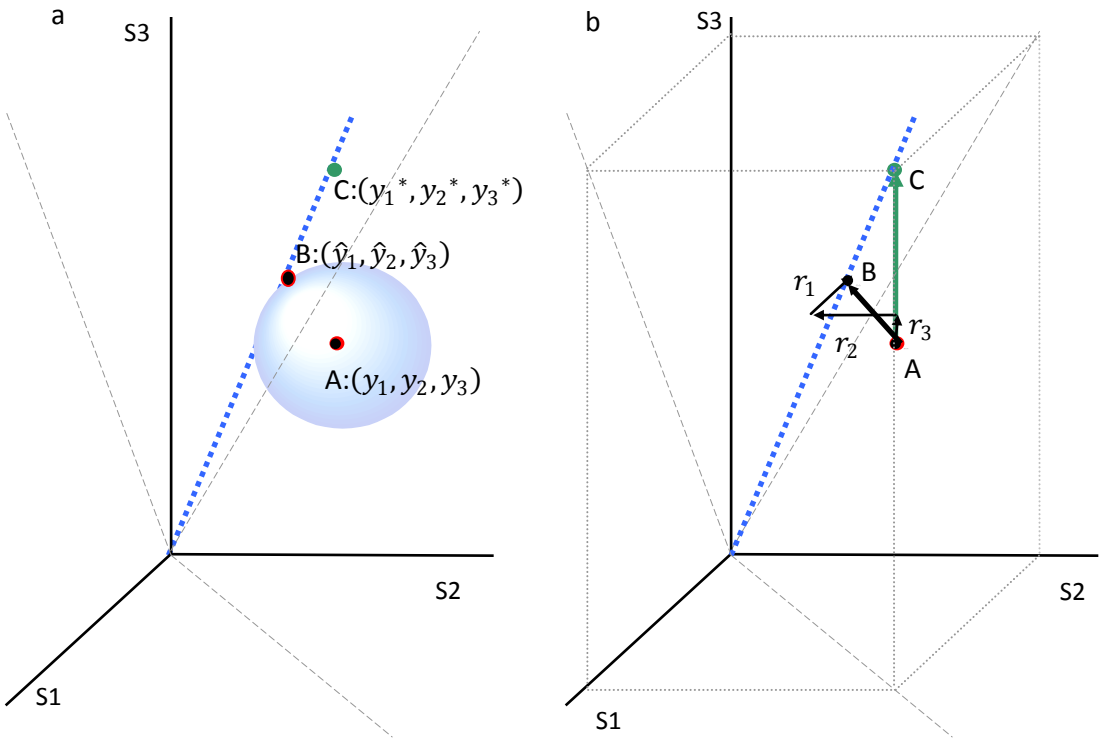


Figure 1 Graphical demonstration of PCA filtering

The blue dotted line is showing the correlation between Sensor S1, S3 and S3 which is found with PCA algorithm (Eq. (4)). Point A is the measured value point B is the reconstructed value and point C is the true value. i.e. every correct measurement should be on this line. Now assume we want to measure the point C: $(y_1{}^*, y_2{}^*, y_3{}^*)$ which is the

true value of the variables, and assume that S3 is damaged therefore we measure the point A:(y_1, y_2, y_3) instead. If we use the regular method of PCA filtering, the point A is transformed to point B:$(\hat{y}_1, \hat{y}_2, \hat{y}_3)$ which is the closest point on line L to this point in the sense of Euclidian norm shown. Therefore the residual vector, which is between the measured data and the estimated data is the vector $\overrightarrow{AB} = (r_1, r_2, r_3)$. It is clear from the right figure that the biggest element of this vector is not r_3. Therefore, it is not necessarily true that the residual corresponding to the failed sensor is the biggest residual. This is the basic assumption in AANN based SFD. Therefore the indentifiability condition that was explained in Section 2.1 is not always met.

It is obvious from this graph that even a perfect method of PCA does not always work for sensor fault diagnosis because of lack of isolability condition. Therefore, we need to modify the PCA method in order to make it capable of detecting the faulty sensor in the more general case. In other words the main problem for sensor fault diagnosis based on principal component analysis is how to map the data into the principal subspace such that it has minimum sensitivity to a single channel when we reconstruct them. This problem can be defined in the following form:

We have a vector of n variables $\boldsymbol{y} = [y_1 \quad y_2 \quad \cdots \quad y_n]$ which are linearly related to a vector of p variables (principal components) $\boldsymbol{x} = [x_1 \quad x_2 \quad \cdots \quad x_p]$ in a fewer dimension space $(p < n)$ by the matrix $\mathbf{A}_{[\mathrm{n}\times\mathrm{p}]} = [\mathrm{a}_{ij}]$.

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x} + \boldsymbol{e} \tag{8}$$

Here, $\boldsymbol{e}$ is the modeling error and is related to the precision of sensors. If we consider the equations corresponding to each row of $\mathbf{A}$ i.e. $[\mathrm{a}_{i1} \quad \mathrm{a}_{i2} \quad \cdots \quad \mathrm{a}_{ip}]$ as a separate Equation,

$$y_i = [\mathrm{a}_{i1} \quad \mathrm{a}_{i2} \quad \cdots \quad \mathrm{a}_{ip}]\boldsymbol{x} + e_i\ , \quad i = 1,2, \dots , n \tag{9}$$

The objective is to find $\boldsymbol{x}$ such that it satisfies the maximum number of theses equations.

This problem can be expressed as the famous problem of *robust regression* in which we want to find a linear regression which is not affected by departures from the model. Here is a typical robust regression problem:

$$\text{Least median of Squares } (LMS) = \mathbf{min_b}|w_i - \boldsymbol{vb}|^2, \tag{10}$$

where w_i is the i^{th} output and $\boldsymbol{v}$ is the vector of inputs and $\boldsymbol{b}$ is the linear relationship between the input and output variables that needs to be determined by regression analysis.

In Equation (10), if we replace $w_i, \boldsymbol{v}$ and $\boldsymbol{b}$ with y_i, the i^{th} row of $\boldsymbol{A}$, and $\boldsymbol{x}$, it will convert robust regression problem to the problem of linear SFD. Therefore, different solutions that exist for robust regression problem can also be used in SFD.

Previous Linear Approaches

The problem of sensor fault diagnosis using PCA is studied by [Dunia et. al. 1996]. They have proved that the method of PCA filtering satisfies the convergence condition but since it does not satisfy the indentifiability condition, they used an optimization approach to reconstruct the faulty data. In this method, the notion of Sensor Validity Index (SVI) is introduced. SVI is a number between zero and one. When a given sensor is healthy SVI is close to one and vice versa. So the faulty sensor is detected and the estimated value of the faulty data is substituted for the wrong measurement and this value will converge to the correct value after several iterations.

Before we introduce the sensor validity index, several auxiliary variables have to be defined. Consider the following PCA filtering matrix which is a combination of mapping

and de-mapping transformation matrices [Dunia et al. 1996]. Equations (11)-(16) are directly quoted from [Dunia et al. 1996].

$$\boldsymbol{C} = \mathbf{P}\mathbf{P}^{\mathrm{T}} = [\boldsymbol{c_1} \quad \boldsymbol{c_2} \quad \cdots \quad \boldsymbol{c_n}]. \tag{11}$$

As noted, the columns of this matrix are denoted by $\boldsymbol{c_i}$

$$\boldsymbol{c_i} = [c_{1i} \quad c_{2i} \quad \cdots \quad c_{ni}]. \tag{12}$$

Next, $\boldsymbol{G}_i$ were defined as

$$\boldsymbol{G}_i{}^T = [\boldsymbol{\xi_1} \quad \boldsymbol{\xi_2} \quad \cdots \quad \boldsymbol{g}_i \quad \cdots \quad \boldsymbol{\xi_n}], \tag{13}$$

where $\boldsymbol{\xi_i}$ is a unit norm vector with zeros in all entries except the i^{th} and

$$\boldsymbol{g_i^{\mathrm{T}}} = \frac{1}{1-c_{ii}}[\boldsymbol{c_{-i}}^{T} \quad 0 \quad \boldsymbol{c_{+i}}^{T}]. \tag{14}$$

where $\boldsymbol{c_{-i}} = [c_{i1}\ c_{i2}\ \dots c_{i(i-1)}]$ and $\boldsymbol{c_{+i}} = [c_{i(i+1)}\ c_{i(i+2)}\ \dots c_{in}]$.

Sensor validity index, η_i, is calculated by the following formula [Dunia et al. 1996]:

$$\eta_i = 1 - (1 - c_{ii})\frac{D_i}{\sum_{h=1}^{n} D_h(1 - c_{hh})} \tag{15}$$

where the D_i is defined by the following equation

$$D_i = \|\boldsymbol{y} - \boldsymbol{G_i y}\| \tag{16}$$

with $\boldsymbol{y}$ as the vector of measurements.

This method is computationally equivalent to solving $(n-1)$ system of equations for evaluating the validity of each sensor. For example consider the system of equations given in (8). In this system, we have n equations which corresponds to n sensor measurements and p unknowns ($p < n$) which are the latent variables e.g. if we want to test the validity of the measurement in Sensor S1, we find the value of the latent variables by excluding the first equation from the system. Then, using the first equation and we

estimate the value of the first sensor and compare it with the measured value. If the first sensor is the faulty one, the measure value will be different from the estimated value; otherwise they are very close to each other.

KPCA Based Methods

Kernel Principal Component Analysis is a method of Nonlinear PCA in which we apply a linear PCA in a nonlinear kernel space [Scholkopf et al. 1998]. In other words, KPCA finds the functions defined in the following form:

$$f_j(\boldsymbol{x}) = \sum_{i=1}^{m} \propto_{i,j} k(\boldsymbol{x}_i, \boldsymbol{x}), \tag{17}$$

such that maximizes the variance of $\boldsymbol{z} \equiv f_j(\boldsymbol{x})$. In this equation, $k(\boldsymbol{x}, \boldsymbol{y})$ is a kernel function in the dot product space $\mathcal{R}^m \times \mathcal{R}^m$ with fixed parameters, $f_j(x)$ is the j^{th} principal function and $\propto_{i,j}$ is i^{th} coefficient of the j^{th} principal component.

Because of the linearity of the algorithm, this method is very popular among various approaches to NLPCA. However, the dependency of the solution on the type of kernel function and the parameters of the kernel function is a major drawback of this method, because we need to have some qualitative information about the data to understand which kernel function has better performance. Another disadvantage is that since the order of

calculations is dependent on the number of observations, there is a significant computational cost for large data sets.

Method of KPCA has also been proposed for process monitoring [Cho et al. 2005] and sensor fault diagnosis [Cho et al. 2004]. The algorithm proposed for process monitoring does not need the reconstruction algorithm and defines two nonlinear monitoring indexes analogous to T^2 and SPE, and process monitoring is performed based on statistical analysis of these indexes. SPE is the root mean squared of the residual between the original and the PCA-filtered data and T^2 is the root mean squared of the principal components [Kresta et al. 1991][Wise and Ricker 1991] [MacGregor et al. 1991]. But, the sensor fault diagnosis method presented requires reconstruction from the principal space. This reconstruction requires nonlinear optimization, which makes the problem very difficult because a typical reconstruction error function for this problem has several local minima.

Another KPCA based approach for fault diagnosis is "Evolving KPCA" [Sun et al. 2007]. This approach addresses the problem of dependability of KPCA on the kernel function chosen a priori, but it does not provide the solution for SFD.

In most solutions to SFD, which is present in the literature, the hidden assumption is that there is only one faulty sensor at one time. In some of the solutions presented in this study this assumption is also made. Before presenting every solution, it is clarified whether this solution works under single or multiple faulty sensor condition.

CHAPTER II

PRINCIPAL COMPONENT ANALYSIS

Assume we have a set of data in n dimensions. Basically we can map these data on any set of n orthonormal vectors in n dimensional space which we call new coordinate system. Principal Component Analysis finds the a new coordinate system for a set of data in which we have the maximum of variance of data in the first dimension, and the second largest variance in the second dimension and so on.

If we have a set of data such that there is correlation between some of the dimensions, principal component analysis can decompose them into a new set of data such that in the new set there is no cross-correlation between the dimensions of data in new coordinate system. The resulting transformed data usually have very high variance in the first few dimensions and very low variance in the last few dimensions which in machine learning terminology are usually referred to as "noise dimensions". They are called noise dimension because their existence is the result of having error in the measurements or noise or any other factor in the system which cause "imperfect" linear correlation between different variables. Otherwise in a system in which we have perfect linear correlation between *k* variables, the last *k* variables in the PCA space would be zero. The PCA decomposition can be written in the following form:

$$\boldsymbol{Y}_{m\times n} = \boldsymbol{T}_{m\times n}\boldsymbol{U}^{T}{}_{n\times n}\,, \tag{18}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1\ \boldsymbol{u}_2\ \boldsymbol{u}_3 \ldots \boldsymbol{u}_n]$ is a set of orthonormal vectors spanning R^n.

In this equation, the original set of data, Y is decomposed into two elements; principal component scores T and principal component loadings U. Figure 2 illustrates how this transformation affects the variance of principal scores.

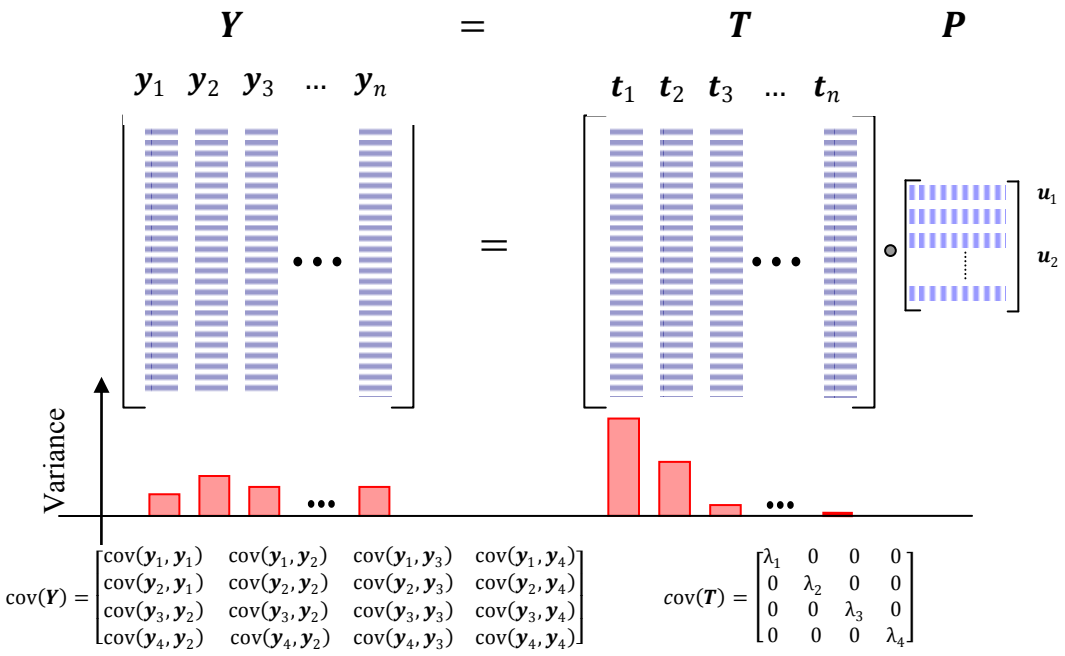


Figure 2 Graphical explanation of PCA transformation

If we decompose the matrix of scores into two parts of high variation scores and low variation scores as shown in Figure 3, we have:

$$\boldsymbol{T} = [\boldsymbol{X} \ \vdots \ \boldsymbol{N}] \tag{19}$$

where $\boldsymbol{X} = [\boldsymbol{t}_1 \ \ \boldsymbol{t}_2 \ \ \ldots \ \ \boldsymbol{t}_k]$, $\boldsymbol{N} = [\boldsymbol{t}_{k+1} \ \ \boldsymbol{t}_{k+2} \ \ \ldots \ \ \boldsymbol{t}_n]$ and $\boldsymbol{t}_{\mathrm{k}}$ is the vector of k^{th} scores.

therefore, equation 1 can be written into the following form:

$$\boldsymbol{Y} = \boldsymbol{T}\,\boldsymbol{U}^T = [\boldsymbol{X} \ \vdots \ \boldsymbol{N}] \begin{bmatrix} \boldsymbol{P}^T \\ \boldsymbol{Q}^T \end{bmatrix} = \boldsymbol{X}\boldsymbol{P}^T + \boldsymbol{N}\boldsymbol{Q}^T \tag{20}$$

where $\boldsymbol{P} = [\boldsymbol{u}_1 \ \ \boldsymbol{u}_2 \ \ \ldots \ \ \boldsymbol{u}_k]$ is the subspace of R^n which spans the space of high variance of data and $\boldsymbol{Q} = [\boldsymbol{u}_1 \ \ \boldsymbol{u}_2 \ \ \ldots \ \ \boldsymbol{u}_k]$ is the complement of $\boldsymbol{P}$ in R^n which spans the noise space.

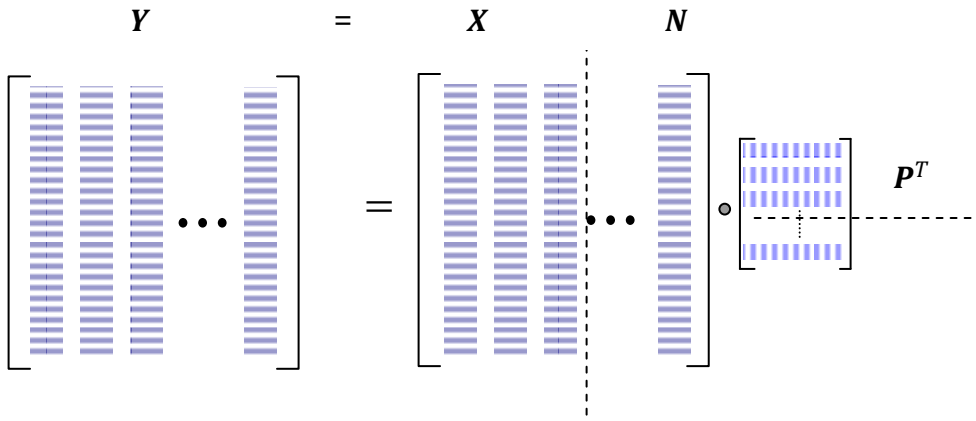


Figure 3 Graphical illustration of PCA decomposition

In fact, data matrix $\boldsymbol{Y}$ is decomposed into two parts as following:

$$\boldsymbol{Y} = \widehat{\boldsymbol{Y}} + \boldsymbol{E} \tag{21}$$

where

$$\widehat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{P}^T \tag{22}$$

is the estimated (or modeled) value of $\boldsymbol{Y}$, and

$$\boldsymbol{E} = \boldsymbol{X}\boldsymbol{Q}^T \tag{23}$$

is the non-modeled variations of $\mathbf{Y}$ or modeling error. These two components are orthogonal to each other because they are in different complementary subspaces of R^n.

$$\widehat{\boldsymbol{Y}}\boldsymbol{E}^T = \mathbf{0} \tag{24}$$

and this property is valid for each row of these matrices (each observation) which are represented as following:

$$\widehat{\boldsymbol{y}}\boldsymbol{e}^T = \mathbf{0} \tag{25}$$

Figure 4 shows the decomposition of data into these two components of principal component subspace and noise subspace.

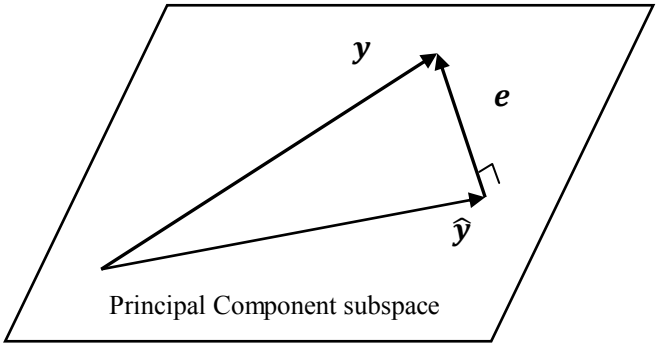


Figure 4 Decomposition of a data into principal component subspace and noise subspace

Whether this correlation comes from a logic base (as in the redundant variables of a system) or is completely stochastic, (as in the information from a picture or a recorded wave sound) as long as we have information about this correlation (principal loadings) we don't need to carry all the data; with the aid of PCA we can select just the part of data that contain most of the variance of data and eliminate the rest of dimensions. In this view, PCA is a very good technique for data compression.

Another Major Application of PCA method is in pattern analysis methods. In pattern analysis algorithms we first calculate a variety of different quantitative characteristics of an object which we call it features. Then we have to categorize the objects based on their location in the feature space. Usually, when the number of features is very large, the classification algorithms have low performance due to the large volume of calculations. Moreover, most of these features may be linearly correlated. Therefore we do not need to

use all of these features for classification algorithms. With the aid of PCA method we find a few of new feature which are not only uncorrelated, but also contains very large variance. PCA has also been used in control for model reduction.

The capability of PCA that we are using in this report is its application in multivariable statistical process monitoring. This technique is have these tree majors step 1-converting the data into PCA space 2- eliminating the dimensions with low variance 3- reverting from the PCA space to the original space. In this procedure those data which does not comply with the PCA model are more deviated than correct data and categorized as damaged measurement or flawed process. This algorithm is explained extensively in section 2.

Derivation of PCA

Several different approaches exist for the derivation of PCA. The simplest approach is based on this definition of PCA: a projection of data that have the most variance of the data.

Assume $\boldsymbol{a} = [a_1, a_2, \dots, a_n]$ is an arbitrary unit vector in R^n

Projection of a data vector $\boldsymbol{y} = [y_1, y_2, \dots, y_n]$ onto $\boldsymbol{a} = [a_1, a_2, \dots, a_n]$ is the linear combination:

$$\boldsymbol{a^T y} = \sum_{j=1}^{n} a_j y_j \tag{26}$$

Projected values of all data vectors in $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_n\ ]^{\boldsymbol{t}}$ onto $\boldsymbol{a}$ is $\boldsymbol{Ya}$ therefore the variance along $\boldsymbol{a}$ is

$$\sigma_a = (\boldsymbol{Ya})^T \boldsymbol{Ya} = \boldsymbol{aY}^T \boldsymbol{Ya} = \boldsymbol{a}^T \boldsymbol{Ca}, \tag{27}$$

where $\boldsymbol{C} = \boldsymbol{Y}^T \boldsymbol{Y}$ is the covariance matrix of the data.

Therefore, based on the aforementioned definition, the value of $\boldsymbol{a}$ is obtained from solution of this optimization problem

$$\boldsymbol{a} = \mathbf{argmax}(\boldsymbol{a}^T \boldsymbol{Ca}), \tag{28}$$

Constraint: $\boldsymbol{a^T a} = 1$

The solution of this problem is easily found with the LaGrange multiplier method. The generalized objective function is defined as follow:

$$\boldsymbol{u} = \boldsymbol{a}^T \boldsymbol{C}\boldsymbol{a} + \lambda(1 - \boldsymbol{a}^T \boldsymbol{a}) \tag{29}$$

This is a quadratic optimization problem and the solution is found by finding the roots of the first derivative of the function:

$$\frac{\boldsymbol{\partial u}}{\boldsymbol{\partial a}} = \mathbf{2}\boldsymbol{C}\boldsymbol{a} - \mathbf{2}\lambda\boldsymbol{a} = \mathbf{0} \tag{30}$$

$$(\boldsymbol{C} - \lambda \boldsymbol{I})\boldsymbol{a} = \mathbf{0} \tag{31}$$

As we see this is the characteristic equation of covariance matrix $\boldsymbol{C}$. therefore the solution of the problem are the eigenvectors of $\boldsymbol{C}$.

Application of PCA in Statistical Process Monitoring

Another major application of PCA method is in Statistical Process Monitoring (SPM). The basic idea of SPM based on PCA is to find a PCA model for a set of correct data from a healthy system and healthy measurements then apply PCA transformation to a

testing data. We consider the system to be healthy if this set of testing data complies with our original PCA model. In order to check the compatibility of testing data with the PCA model, we first map the original data into fewer dimension of principal components, and then reconstruct them out of these components the difference between the original data and estimated data out of its principal components suggest us the evaluate of compatibility of the data with the model.

$$\boldsymbol{x} = \boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{y} \tag{32}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{P}^{T}\boldsymbol{P}\boldsymbol{y} \tag{33}$$

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}} = (\boldsymbol{I} - \boldsymbol{P}^{T}\boldsymbol{P})\boldsymbol{y} \tag{34}$$

Figure 5 schematically describe the method of PCA based system monitoring for a simple 2-D example which is based on mapping and reconstructing data from PCA model. Figure 5(a) shows a group of measured data from two different variables of a system. It is clear from this graph that these two variables are linearly correlation to each other. If we transform these data to a PCA space, we have the same data in a new coordinate system such that there is no correlation between the data in the new coordinate system. Figure 5(b) shows the data in PCA space. In this figure, the variation of data in the second dimension v is negligible compared to the first dimension x. Therefore, we eliminate this dimension of data and keep the data in just x space. Graphical interpretation of this act is

like shifting all the data in Figure 5(b) to their closest point into x axis because we assume that v is zero for all the data in this space. Therefore, after reconstructing the data to the original space, as show in Figure 5(c), all the data would relocate to a straight line with no noise properties.

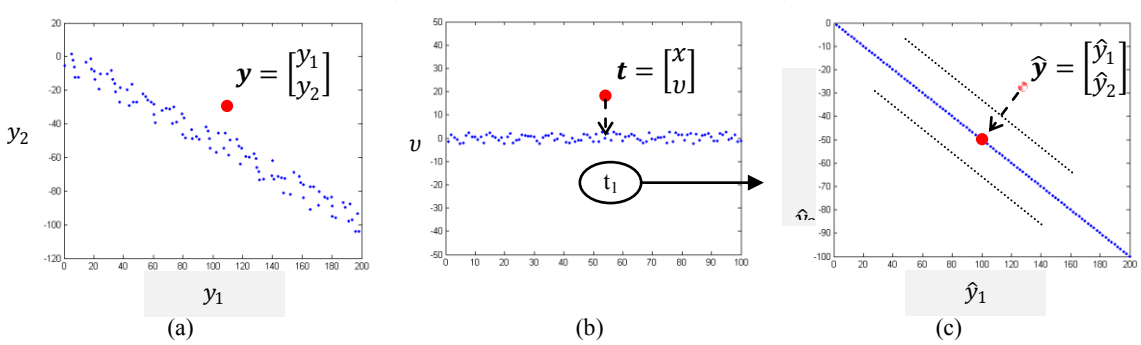


Figure 5 Sensor fault detection using PCA

The degree of being healthy is judged based of the amount of relocation from original to reconstructed data or $\|\hat{\boldsymbol{y}} - \boldsymbol{y}\|$ through this process. Usually we define a detection limit based on which, we can interpret the measurement/system as healthy or faulty.

A Standard method of testing is monitoring of Squared Prediction Error (SPE) in residual space defined as

$$Q_k = \|\hat{\boldsymbol{y}}_k - \boldsymbol{y}_k\| \tag{35}$$

Which the square of relocation distance is as discussed earlier. We define detection limits based on this parameter as following

$$95\%\ control\ limit = Q_{mean} \pm 2\sigma, \tag{36}$$

$$99\%\ control\ limit = Q_{mean} \pm 2\sigma, \tag{37}$$

where σ is the variance of Q.

Another popular method of testing is Hoteling T^2 test which does not need the reconstruction of data and is done in PCA space. T^2 test parameter is defined as:

$$T_i^2 = \sum_{j=1}^{k} \frac{t_{ij}^2}{\lambda_j} \tag{38}$$

where t_{ij} are the elements of principal component scores, λ_j is the j^{th} eigenvalue of covariance matrix of original data and k is the number of selected scores. Detection limits can also be defined in the same method as shown for SPE residual error. A typical graph of these two indexes with their 95% and 99% control limits are shown in Figure 6.

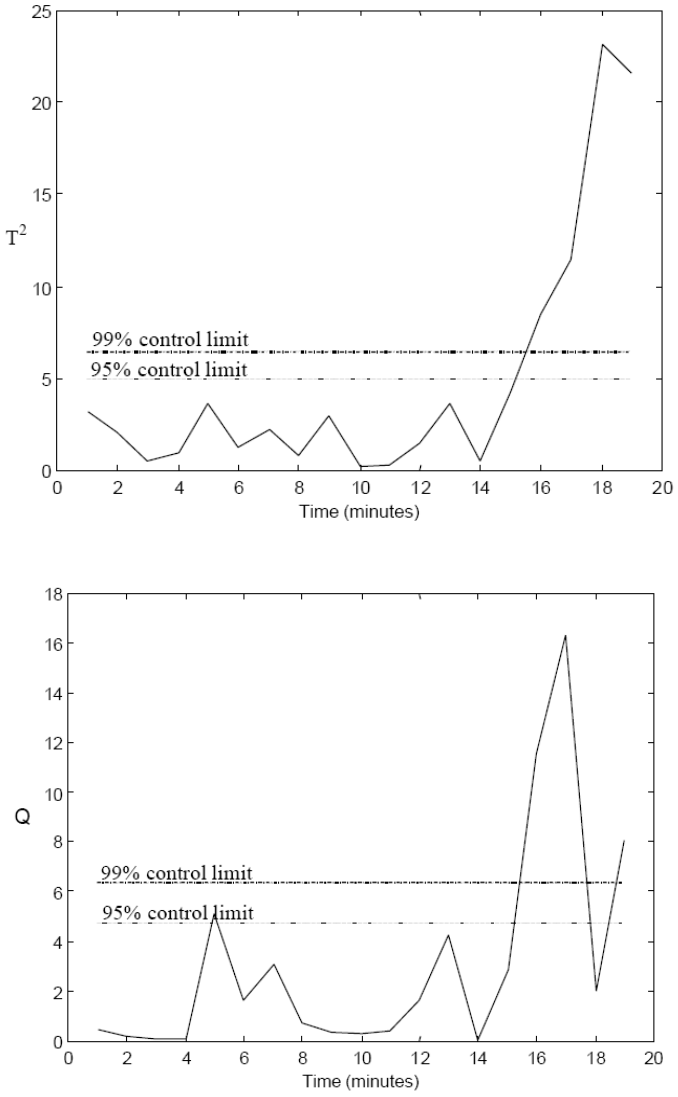


Figure 6 $\boldsymbol{T^2}$ and Squared Prediction Error (SPE)

Application of PCA in Sensor Fault Diagnosis

Although PCA is a popular method in process monitoring, it has not been used widely for sensor fault diagnosis. The reason is very simple; in the process monitoring, we just need to check of the new set of data is complying the existing correlation between the healthy system data or not. But in sensor fault diagnosis, we need to find faulty sensor which causes this inconsistency and find the correct value of that sensor at that time based on other sensors in the array and this make the problem much more challenging.

The first solution to this problem that comes to mind is to use the PCA filtering (PCA transformation and pseudo-inverse) and compare the value of measured data with the estimated data and the sensor with the most change in the value is considered faulty sensor.

The formulation is the same as formulation of process monitoring. We call the difference between the measured value and estimated value as $\boldsymbol{e}$

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}} = (\boldsymbol{I} - \boldsymbol{P}^T\boldsymbol{P})\boldsymbol{y} \tag{39}$$

where $\boldsymbol{e} = [e_1, e_2, \dots, e_n]$

Then the j^{th} sensor is considered faulty where

$$j = ind(max\ |e_j|) \tag{40}$$

Once we found the faulty sensor, we replace it with its corresponding estimated value.

$$\hat{\boldsymbol{y}}^{<new>} = [y_1\ y_2 \dots \hat{y}_j^{<old>} \dots\ y_n\ ] \tag{41}$$

Dunia et al. [1996] have proved that if there is just one faulty sensor and if we found the faulty sensor correctly, this procedure will converge to the correct value of sensor.

But the problem is that this method cannot always find the faulty sensor. The reason is that some sensors are much more sensitive than others in a sensor array and therefore have more influenced to this kind of filtering. This is explained graphically in Figure 7

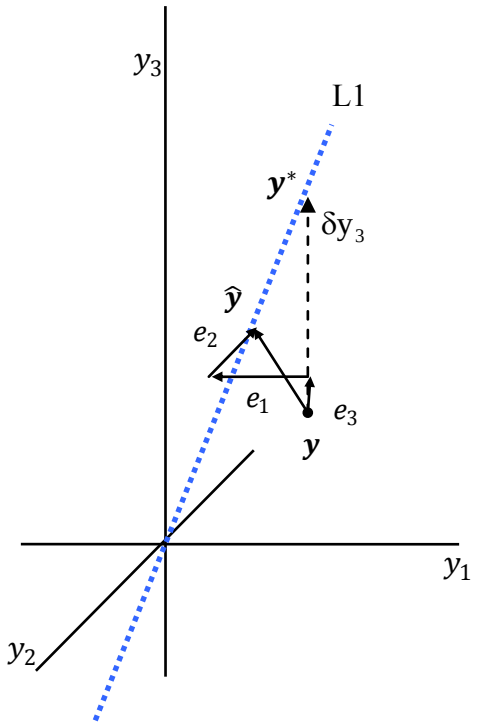


Figure 7 The value of a faulty sensor compared with the reconstructed value

In this figure, the value of data from 3 sensors is represented in 3D space. We assume that parameters measured by these 3 sensors are linearly correlated such that the correct measurement should be on the dotted line, L1.

Now assume that the sensor S_3 which measure the property of y_3 is faulty and instead of measuring $\boldsymbol{y}^* = [y_1^*, y_2^*, y_3^*]$, we have measured $\boldsymbol{y} = [y_1, y_2, y_3]$ where $|y_1 - y_1^*| < \epsilon_1$ and $|y_2 - y_2^*| < \epsilon_2$, but $|y_3 - y_3^*| < \epsilon_3$ and ϵ_k is the acceptable range of error in the k^{th} sensor.

For the sake of simplicity we assumed that ϵ_1, ϵ_2 and ϵ_3 are zero in the example shown in the graph.

Now, if we apply PCA filtering, the measured point $\boldsymbol{y}$ transfers to the closest point that meet the correlation line L1. This point is shown as $\hat{\boldsymbol{y}}$ in Figure 7.

As we see in this figure, the difference between the measured data $\boldsymbol{y}$, and estimated data $\hat{\boldsymbol{y}}$ is $\boldsymbol{e} = [e_1 \;\; e_2 \;\; e_3]$ so the maximum were the value of e_1 and e_2 is obviously larger than e_3 however the faulty sensor is S_3 therefore, this algorithm does not find the faulty sensor. Figure 8 shows the difference between perfect correction vector which transfers the measured data point to the correct data point and PCA correction vector which transfers the measured data to the estimated data point using PCA.

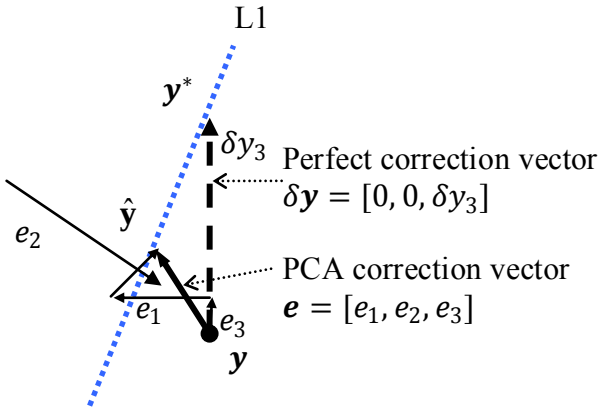


Figure 8 Comparison of PCA correction vector and perfect PCA vector

CHAPTER III

AUTO-ASSOCIATIVE NEURAL NETWORKS

As stated in the introduction, AANN has been commonly used for sensor fault diagnosis. In this section we analyze the efficiency of SFD using AANN and specify under what condition this method works.

Introduction to Auto-Associative Neural Networks

Auto-Associative Neural Networks are feed-forward neural networks with five layers where the middle layer has minimum number of neurons and the input layers has the same number of neuron as the output layer. Usually the second and forth layer have neurons with nonlinear activation functions and the middle (bottleneck) layer and last layer have linear neurons with linear activation function. Figure 9 shows a diagram of AANN.

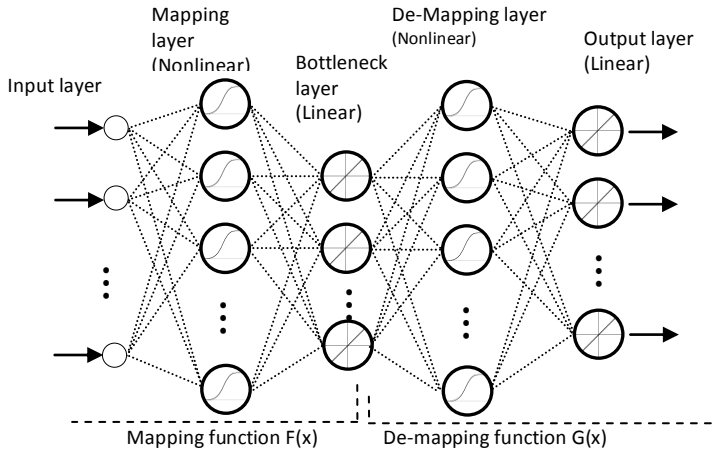


Figure 9 Architecture of AANN

This network is trained for an identity function i.e. the input data is equal to the target data. The premise is that since these data are forced to pass through a layer with fewer dimensions, the nonlinear relationship between the different channels of data is captured in the weights of the AANN during the training process. As a result, we have a network that in the ideal case replicates its input exactly. Here the term ideal refers to the situation where the input channels are consistent with each other (or correlated in the sense that is further elaborated below.) However, when any single input channel differs from its "correct" value, the output corresponding to that channel will be different from the input value and will be closer to the correct value.

Application of AANN to Sensor Fault Diagnosis

When we have a system with several redundant sensors, we can collect a complete history of the system measurements and train an AANN with these data. Then, during the operation of the system we can feed the AANN with these sensor readings and compare the output of the AANN with the input. As long as sensors have enough accuracy the relationship between their measured values are valid. Therefore, the output would be the same as its input. As soon as one of these sensors goes out of calibration or fails, the output of AANN will ideally show the corrected value of the sensor.

However, in real cases as Kramer has explained in his original paper [Kramer 1991], when we have a faulty measurement in one of these sensors, all of the output values would be distorted. But, we are able to single out the faulty sensor by analyzing the

residue vector between the input and output, because it is expected that the maximum absolute value of residue is in the faulty sensor [Kramer 1991].

After finding the faulty sensor, the next step is to feed the estimated value of the faulty sensor as AANN input. After several iterations, all of the residues will approach to zero and we have reconstructed the value of the faulty sensor by doing this. The diagram of Figure 10 shows this procedure [Kramer 1991].

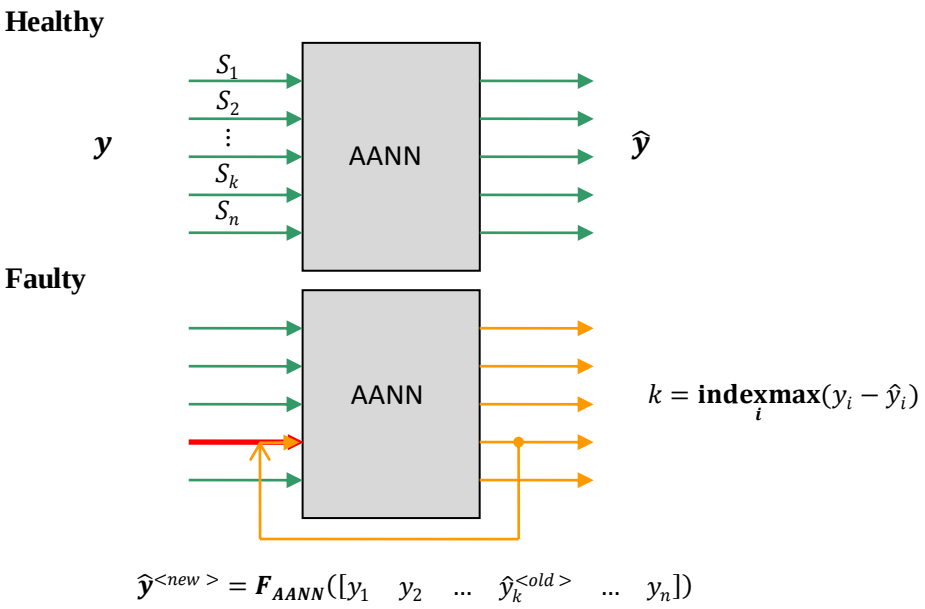


Figure 10 Detection and reconstruction of faulty signal using an AANN algorithm

Requirements of AANN as an SFD Method

According to this method, it is necessary that an AANN satisfies all of the following properties to be able to be used for SFD. In this research, these conditions have been observed and classified as:

Identity condition: When the input values are not faulty, the difference between the input and output of the AANN should be within a limited tolerance. This tolerance can be estimated by the precision of the sensor.

Isolability condition: When there is a wrong measurement in one of the sensors, the difference between input and output of the Neural Network should be highest in the faulty channel.

Convergence condition: when there is a wrong measurement in one of the sensors, the output of the network should converge to the true value if we feedback the estimated value for the faulty channel

One of the objectives of this project is to investigate each of these requirements and specify the conditions by which an AANN satisfies them. We are also seeking for alternative methods that meet these conditions.

Next section is to study these three conditions on a special form of the problem when there is a linear correlation between the monitored sensors.

CHAPTER IV

LINEAR METHOD OF SENSOR FAULT DIAGNOSIS

Fault detection and isolation (FDI) is the subject of interest in many industries and engineering fields. There is a wide range of methodologies and approaches to FDI. These methodologies are depend the type of knowledge that is available about the system, the type of the systems and even the type of the errors that need to be isolated. There are some outstanding review papers about these monitoring methods either model based methods [Gertler and Singer 1990] ,[Frank 1990],[Frank and Ding 1997], [Isermann 2005] or statistical methods [Bermis et al. 2007] or a comprehensive one containing both methods [Venkatasubramanian et al. 2003].

Sensor fault diagnosis can be viewed either as a subcategory of model based approaches to FDI or a problem in statistical process control. There are also some individual algorithms based on machine learning techniques. In the following sections a brief survey of these three approaches is discussed.

Model Based Fault Diagnosis

A category of model based methods which will be discussed in this article is the application of consistency (parity) equations. The parity equation-based approach is a well-developed category of general fault diagnosis in the analytical redundancy framework [Frank and Ding 1997]. In aerospace terminology, the input-output models,

generating residuals, are referred to as parity equations [Satin and Gates 1978]. These residuals are zero in the absence of noise, modeling error, and any fault in the system. Therefore, statistical analysis of residuals can lead to detection and isolation of potential faults in the system. Gertler [Gertler 1988] [Gertler and Singer 1990] offers a comprehensive survey of consistency equations-based fault diagnosis techniques. In this category of fault diagnosis, the ability to detect a failure in the system depends not only on the size of the failure and the level of noise, but also on the "direction of residuals relative to the failure" which depends on the *structure* of parity equations. With this in mind, Gertler proposed the method of *structured residuals* [Gertler and Singer 1990]. This method *re-structures* the parity equations by a linear transformation, and finds new residuals which are *orthogonal* to the system faults. By definition, a residual is orthogonal to a failure if it is unaffected by that failure. In the absence of failure, all of the residuals are close to zero but when a certain fault happens in a well-structured parity equation, one or more residuals remain close to zero but the rest become nonzero. Thus we can trace the specific fault through its corresponding orthogonal residuals. Gertler also defines the concepts of *Zero-threshold isolability* and *High-threshold isolability* based on the structure of parity equations. Roughly speaking, Zero-threshold and High-threshold isolability are the ability of tracing back a fault in a system based on a given set of residuals, respectively, in the absence and presence of noise [Gertler and Singer1990].

Generally these approaches are used for diagnosis of faults in different parts of a control system resulting from a failure inside the plant or in the controller, actuators or sensors.

When the problem is focused on the detection of faults in the sensors, it is called sensor fault diagnosis or Instrument Detection and Isolation (IDI).

Fault Diagnosis in Process Control

Parallel to the above development, and almost independent of that work, multivariable statistical approaches have been used for monitoring complex chemical processes [MacGregor et al. 1989] [Miller et al. 1998]. Multivariate Statistical Process Control (MSPC) uses Principal Component Analysis (PCA) to model normal process behavior. System faults are then detected by referencing the observed behavior against this model. MacGregor [MacGregor et al. 1989] presents a brief survey of these methods. MSPC can be considered a data driven approach to fault diagnosis. However, in its general form MSPC can only "detect" rather than "isolate" a given fault. To address this problem, Dunia et. al. proposed a method for isolation of faulty sensors through reconstruction of each sensor by an optimization approach [Dunia et al. 1996]. Dunia's work can also be categorized as a data-driven approach to *sensor fault diagnosis*.

PCA has also been further studied as a parity relation model for sensor fault diagnosis. Gertler defined the primary residuals as the projection of observed data into the residual space, and suggested the formation of new residuals which are selectively sensitive to different sensor faults by a linear transformation [Gertler and Monajemi 1995][Gertler et al. 1999]. In addition, Qin the used PCA method in combination with the structural residual approach to design an optimum residual generator structure. This structure makes one residual insensitive to one subset of faults while being most sensitive to other

faults [Qin and Li 1999]. In a more recent study, another optimal structure has been proposed which considers the ratio of the fault to the noise standard deviation as the objective function. The structure found is then obtained as a linear combination of eigenvectors spanning the residual space [Gertler and Cao 2005].

Individual Approaches for Sensor Fault Diagnosis

Sensor fault diagnosis basically targets the problem of detection and isolation of faulty sensors in large plants where several analytically redundant sensors are used. Industry has recently shown more interest in the data-driven approaches since these methods do not require an analytical input-output model, which may not be available in industrial settings [Hines et al. 1998]. One of the first intelligent based approaches to sensor fault diagnosis is based on the method of Autoassociative Neural Networks (AANN) [Kramer 1992]. AANN is a feedforward neural network with a specific architecture that is used to train an identity function (target data of the neural network is the same as its input data). It can be compared to the parity equation methods by considering the difference between the input and output of the network as the generated residual. The expectation from a perfect AANN is that once trained with sufficient amount of correct data, it can reconstruct the faulty observations as closely as the network allows. There is a wide range of industrial applications of AANN in sensor fault diagnosis, ranging from nuclear power plants to turbofan engines [Hines et al. 1998][Dune et al. 1998][Hoffman and Kimble 2005]. However as reported by a number of researchers [Malthouse 1998], [Najafi et al. 2004], [Sharifi et al. 2004], there are several difficulties with the application of AANN in

practice. Apart from its deficiency as a heuristic approach to nonlinear PCA, the main problem with AANN is lack of a theoretical foundation for this method. Therefore many critical questions about this method cannot be answered e.g., what is the optimum architecture of the network, how does one train the optimum AANN, or whether we can isolate a fault in any sensor.

Other techniques of pattern classification have also been used in data driven fault diagnosis. Fisher Discriminant Analysis (FDA) and Partial Lease Squares (PLS) are among them. FDA determines the portion of the observation space that is most effective in classification amongst several data classes. Thus, all fault classes information is utilized when the discriminant function is evaluated for each class. PLS are data decomposition methods for maximizing the covariance between the predictor (independent) block and the predicted (dependent) block for each component. PLS attempts to find loading and score vectors that are correlated with the predicted block while describing a large amount of the variation in the predictor block [Chiang and Braatz 2001]

In this study, we will define the concept of *minimal parity equation*. Then a sensor isolation method using this equation is presented. This methodology shows that once the minimal parity equation is obtained through PCA in data driven methods, the detection and isolation of faulty sensor does not need any additional restructuring. In this new perspective, the concept of isolability, which is defined by Gertler [Gertler and Singer 1990] as a property of a specific structure, will be considered as a property of the system.

Minimal Parity Equations

There are basically two types of analytical redundancy: Direct or static redundancy and temporal or dynamic redundancy. In static redundancy, there is a relationship between instantaneous redundant sensor outputs whereas dynamic redundancy is the result of dynamic relationship between sensor outputs and actuator inputs. Because of measurement availability in many industrial plants and chemical processes, there is static redundancy in such plants and sensor data validation can be performed without any knowledge about the dynamics of the system [Schweppe and Wildes 1970][Vaclavec 1969]. Therefore in this article we derive the parity equations for the static case.

Analytical Derivation

Consider a general input-output system in the following form:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{42}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{e} \tag{43}$$

where $\boldsymbol{x}$ is the $k \times 1$ state vector, $\boldsymbol{u}$ the $l \times 1$ input vector, $\boldsymbol{y}$ the $n \times 1$ vector of measured outputs and $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ are known matrices of appropriate dimensions. In this equation, $\boldsymbol{e}$ is the measurement error which is the difference between the true value of $\boldsymbol{y}$ and its measured value. It is a combination of noise, and a potential sensor fault.

When the number of measurements is more than the number of states of the system ($n > k$), we have static redundancy in the system. We shall show that, a necessary condition for detectability based on static redundancy is $n > k$, and a necessary condition for isolability based on static redundancy is $n > k + 1$.

The simplest form of residual is the difference between the measured data and its estimated value from the model. Rewriting Eq. (43), we have

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{C}\boldsymbol{x}. \tag{44}$$

The residual vector generated in this way is called the *primary residual*. Evidently we need to estimate the state of the system, $\boldsymbol{x}$, to be able to use this equation. Further, note that the dimension of the residual vector is the same as the dimension of the system output. Therefore, when there is a linear correlation among the output variables, the residuals found by this equation would also become correlated. Now If we find the matrix $\boldsymbol{Q}$ containing $n - k$ orthonormal row vectors such that $\boldsymbol{Q^T}$ spans the null space of $\boldsymbol{C^T}$ ($\boldsymbol{C^T Q^T} = \boldsymbol{0}$ or $\boldsymbol{QC} = \boldsymbol{0}$), and multiply it by both sides of Eq. (44), we have:

$$\boldsymbol{Qe} = \boldsymbol{Qy} - \boldsymbol{QCx} = \boldsymbol{Qy}. \tag{45}$$

Therefore, if we define a new residual vector $\boldsymbol{r} = \boldsymbol{Qe}$, based on Eq. (45) we have

$$\boldsymbol{r} = \boldsymbol{Q}\boldsymbol{y}. \tag{46}$$

Equation 5 is a parity equation which generates the maximum number of independent residuals. Since there is no redundancy between the residuals found in this equation, we call this equation the *minimal parity equation* and the residual vector generated by this equation is called the *minimal residual vector*. As long as the column rank of matrix $\boldsymbol{C}$ in Eq. (43) is less than its row rank, the matrix $\boldsymbol{Q}$ exists. Notice that $\boldsymbol{Q}$ in the minimal parity equation is not unique but the other variants of $\boldsymbol{Q}$ simply reflect different coordinate systems of the residual space. As we will see shortly, this equation is the main equation used in the sensor fault diagnosis method presented in this chapter, and the detectability and isolability of each sensor can be determined based on $\boldsymbol{Q}$.

PCA Based Derivation

Principal Component Analysis (PCA) is a common technique in applications such as dimensionality reduction, data compression and feature extraction. It is also referred to as Karhunen-Loeve transform or Hotelling transform. The PCA transformation provides a new lower dimensional coordinate system in which there are no correlations among the variables. If we show the variables in the original coordinate system by $\boldsymbol{y}$ and those in the new coordinate system by $\boldsymbol{t}$, we have:

$$\boldsymbol{t} = \boldsymbol{U}\boldsymbol{y}, \tag{47}$$

such that $\mathbf{cov}(t_i, t_j) = 0 \; \forall \; i \neq j$.

The orthonormal matrix $\boldsymbol{U} = [\boldsymbol{u}_1 \; \boldsymbol{u}_2 \; ... \boldsymbol{u}_n \;]^{\boldsymbol{T}}$ comprises the unit vectors $\boldsymbol{u}_i$ of the new coordinate system. Also, the variables in the new coordinate system are decreasingly ordered based on their standard deviations $\sigma(t_1) > \sigma(t_2) > \cdots > \sigma(t_n)$**.** As a result, when there is a linear correlation between a set of variables, depending on the order of the correlation, the variances of the last few variables are considerably lower than those of the first few variables in the new coordinate system. Separating the high variance variables, $\boldsymbol{x}$, from the low variance variables, $\boldsymbol{v}$, in the form:

$$\boldsymbol{t} = \begin{bmatrix} \boldsymbol{x} \\ - \\ \boldsymbol{v} \end{bmatrix}, \tag{48}$$

and their corresponding basis vectors, $\boldsymbol{P}$ and $\boldsymbol{Q}$, in the form:

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{P} \\ - \\ \boldsymbol{Q} \end{bmatrix} \tag{49}$$

we have:

$$\boldsymbol{x} = \boldsymbol{P}\boldsymbol{y}, \tag{50}$$

and,

$$\boldsymbol{v} = \boldsymbol{Q}\boldsymbol{y}, \tag{51}$$

where matrices $\boldsymbol{P} = [\boldsymbol{u}_1\ \boldsymbol{u}_2 \ldots \boldsymbol{u}_k\ ]^{\boldsymbol{T}}$ and $\boldsymbol{Q} = [\boldsymbol{u}_{k+1}\ \boldsymbol{u}_{k+2} \ldots \boldsymbol{u}_n\ ]^{\boldsymbol{T}}$ are the upper and lower parts of $\boldsymbol{U}$ respectively. Therefore, we have divided the entire space into a "model space" represented by $\boldsymbol{P}$ and a "noise space" represented by $\boldsymbol{Q}$. This matrix is the equivalent of matrix $\boldsymbol{Q}$ derived using the model based parity equation (Eq. (**45)**).

Now, if $\boldsymbol{y}$ is a vector of statically redundant measurements, Eq. (51) would be a minimal parity equation because $\boldsymbol{v}$ is zero in the absence of noise, and components of $\boldsymbol{v}$ are independent. Notice that (50) and (51) are equivalent to (43) and (46). In fact the matrix $\boldsymbol{Q}$ which is found here is equivalent to that found analytically in (45). The earlier is found based on a physical model of the system, whereas the later is found numerically from a database gathered from measuring of system variables.

Sensor Isolability Theory

The minimal parity equation is the basic equation used for sensor fault diagnosis. This section shows how to use this equation to find out whether it is possible to detect and isolate sensor faults, i.e. the analysis of isolability and detectability of sensors, and introduces a method for isolation of faulty sensors.

Consider the vector of measurements $\boldsymbol{y}$. Usually, this measurement contains the true value, measurement noise and a potential error in the system. So if we show the true value of $\boldsymbol{y}$ by $\boldsymbol{y}^*$ we have:

$$\boldsymbol{y} = \boldsymbol{y}^* + \boldsymbol{\eta} + \boldsymbol{\Delta}. \tag{52}$$

In this equation, $\boldsymbol{\eta}$ is the measurement noise, $\boldsymbol{\Delta} = [\delta_1 \quad \delta_2 \quad \dots \quad \delta_n]^T$ where δ_i is the error in i^{th} the sensor. Apparently, in the absence of any noise or measurement error, $\boldsymbol{y} = \boldsymbol{y}^*$ and the residual error is equal to zero:

$$\boldsymbol{r} = \boldsymbol{Q}\boldsymbol{y}^* = \boldsymbol{0}. \tag{53}$$

Now, assuming the amount of measurement noise is negligible; $\boldsymbol{\eta} \cong 0$, (We shall deal with the noisy case in section IV) owe have:

$$\boldsymbol{y} = \boldsymbol{y}^* + \boldsymbol{\Delta} \tag{54}$$

Calculating the minimal residual vector, we have:

$$\boldsymbol{r} = \boldsymbol{Q}\boldsymbol{y} = \boldsymbol{Q}(\boldsymbol{y}^* + \boldsymbol{\Delta}) = \boldsymbol{Q}\boldsymbol{\Delta}. \tag{55}$$

Next, consider the $\boldsymbol{Q}$ matrix in the form of a combination of column vectors $\boldsymbol{q}_i$,

$$\boldsymbol{Q} = [\boldsymbol{q_1}\ \boldsymbol{q_2}\ \cdots\ \boldsymbol{q_n}]. \tag{56}$$

Therefore, the residual vector is:

$$\boldsymbol{r} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \delta_j \boldsymbol{q_j}. \tag{57}$$

Assuming there is error only in one sensor and it is the i^{th} sensor, $\delta_i \neq 0$ and $\delta_j = 0$, $\forall j \neq i$,we have:

$$\boldsymbol{r} = \delta_i \boldsymbol{q_i}. \tag{58}$$

This means that the residual vector is in the same direction as the i^{th} column of $\boldsymbol{Q}$. This simple fact can be used to detect the single faulty sensor in the system. In other words, each column of $\boldsymbol{Q}$ corresponds to one sensor in the system. So the faulty sensor is the sensor whose corresponding column in $\boldsymbol{Q}$ has the same direction as the residual vector. Since the i^{th} column vector, $\boldsymbol{q_i}$, is the key to finding the fault in the i^{th} sensor we call it the *fault image vector* for the i^{th} sensor.

In order to compare the direction of the residual vector with the direction of columns of $\boldsymbol{Q}$, each column vector is normalized:

$$\boldsymbol{n_i} = \frac{\boldsymbol{q_i}}{\|\boldsymbol{q_i}\|} \quad \boldsymbol{i} = \mathbf{1}, \mathbf{2} \cdots \boldsymbol{n}. \tag{59}$$

we call $\boldsymbol{n_i}$ the *normalized fault image vector* of the i^{th} sensor. And similarly we define the normalized residual as:

$$\boldsymbol{n_r} = \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|}. \tag{60}$$

The inner product of the normalized residual and the normalized fault image vector is defined as the *sensor failure index* (SFI).

$$\boldsymbol{f_i} = \boldsymbol{n_r} \cdot \boldsymbol{n_i} \tag{61}$$

where the dot sign is inner product of two vectors.

When the i^{th} sensor is faulty, these two vectors are in the same or opposite directions; therefore, $|f_k| = 1$. In practice however, due to measurement noise the direction of residual vector deviate from that of fault image vector and this index becomes less than one. But assuming that there is a single faulty sensor in the system, the sensor which has the closest value of SFI to one is considered to be the faulty sensor.

Now we return to the basic question of sensor fault diagnosis. i.e. whether we are able to detect/isolate the fault in a specific sensor in a system using static analytical redundancy. In other words, whether there is significant static redundancy to be able to isolate the faulty sensor.

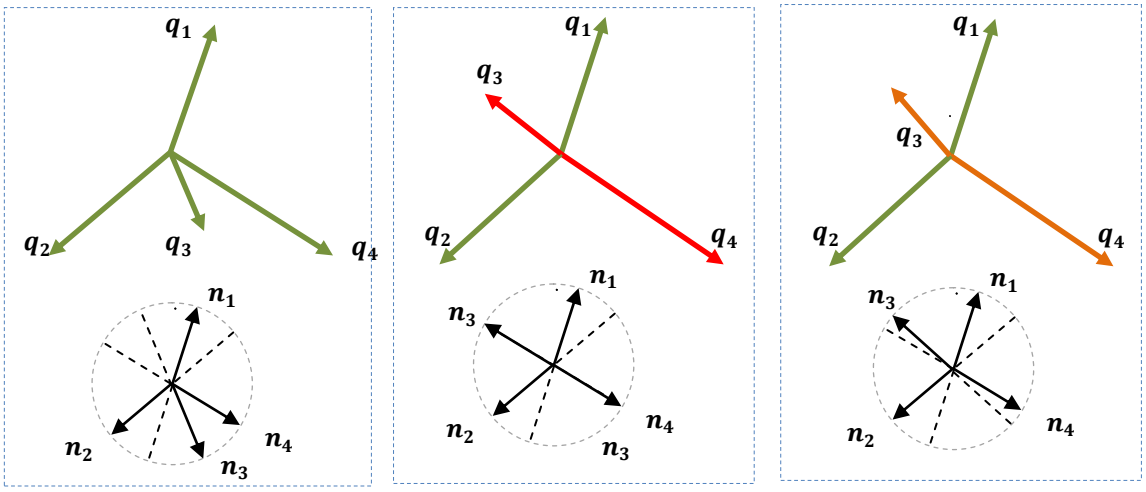
The aforementioned method of sensor fault diagnosis gives us a good suggestion on how to find the answer. In fact we can simply answer this question based on the structure of the minimal parity equation i.e. the $\boldsymbol{Q}$ matrix. As long as all column vectors of $\boldsymbol{Q}$ are nonzero and linearly independent, it is possible to detect and isolate a single sensor fault in any sensor. If the corresponding column/vector of a sensor in $\boldsymbol{Q}$ is zero, that sensor is not detectable. If it is nonzero but it is collinear with any other column vector in $\boldsymbol{Q}$, both sensors whose corresponding column vectors in $\boldsymbol{Q}$ are collinear are non-isolable. This isolability test method is explained in Example 2.

Definition: A sensor in a system is defined as *single fault isolable* if it is possible to detect and isolate a single fault in that sensor.

Theorem 1: A sensor in a system is statically single fault isolable iff its corresponding fault image vector is nonzero and linearly independent. (The proof is presented in Appendix A)

Notice that if the lines of fault image vector in two sensors are very close to each other, they are theoretically isolable, but in practice, we need a large number of sample measurements to be able to isolate them from each other. In this case, we call the system

as poorly isolable. Figure 11 illustrates the concept of isolability in a system of four measurements. In this figure fault image vectors are shown in the 2D space of residuals. This implies that the degree of redundancy is 2. The system shown in part (a) has four different fault image vectors which are nonzero and linearly independent. Therefore, all sensors are isolable. The system shown in part (b) has two collinear fault image vectors (q_3 and q_4). Therefore, the two sensors corresponding to these fault image vectors are not isolable. Part (c) show a poorly isolable system. In this system, there are two fault image vectors which are not collinear, but they are very close to the same line (q_3 and q_4). This situation makes the isolation task for these two sensor very difficult when they are contaminated with noise.



(a) Isolable system (b) Non-isolable system (c) Poorly isolable system

Figure 11 Graphical demonstration of isolability in different systems

Effect of Noise on Fault Isolation

In the last section, we assumed that the measurements are noise free. However, in real systems we always have some level of noise/inaccuracy in the measurements. Presence of noise reduces the ability to detect the fault. Clearly the higher the level of noise in the system, the less likely it is to detect the faults. But the adverse effect of noise is not only dependent on the noise level, but it is also dependent on the structure of the problem or the parity equation. In this section, the effect of noise on the linear sensor fault detection is studied.

As discussed earlier, the vector of measurements is composed of noise, $\boldsymbol{\eta}$ and a potential error in one channel.

$$\boldsymbol{y} = \boldsymbol{y}^* + \boldsymbol{\eta} + \boldsymbol{\Delta} \tag{62}$$

When we form the residual vector, for the single sensor error we have:

$$\boldsymbol{r} = \boldsymbol{Q}\boldsymbol{y} = \boldsymbol{Q}(\boldsymbol{y}^* + \boldsymbol{\Delta} + \boldsymbol{\eta}) = \boldsymbol{Q}\boldsymbol{\Delta} + \boldsymbol{Q}\boldsymbol{\eta} = \delta_i \boldsymbol{q_i} + \boldsymbol{v}, \tag{63}$$

where $\boldsymbol{q_i}$ is the k^{th} column of $\boldsymbol{Q}$, and $\boldsymbol{v} = \boldsymbol{Q}\boldsymbol{n}$ is the image of noise in the residual space. If we divide the residual vector by δ_i, we have:

$$\frac{1}{\delta_i}\boldsymbol{r} = \boldsymbol{q_i} + \frac{1}{\delta_i}\boldsymbol{v} \tag{64}$$

Based on Eq. , it is clear that the difference between the residual and its corresponding column in $\boldsymbol{Q}$ is proportional to $\frac{\|\boldsymbol{v}\|}{\delta_i}$ as graphically shown in Figure 12. Therefore, as long as the magnitude of fault is of the same order as the noise level, it is not possible to detect the error. But as the magnitude of fault exceeds the level of noise, the ratio of $\frac{1}{\delta_i}\boldsymbol{v}$ becomes smaller and detection of error becomes easier. Now the question is that what should be the maximum ratio of noise to error to be able to detect an error in the system.

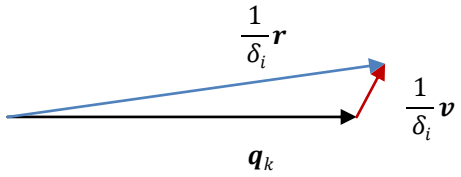


Figure 12 The vector of residual in the presence of noise and error in the k^{th} sensor

Minimum detectable level of error is different in each sensor. For finding the minimum detectable magnitude of fault in sensor i, we first need to find the sensor that has the closest fault image vector to that of the i^{th} sensor. Consider the vectors $\boldsymbol{q_i}$ shown in Figure 13 Accordingly, $\boldsymbol{q_j}$ is selected such that:

$$\boldsymbol{j = argmax}\left(\frac{|\boldsymbol{q_i} \cdot \boldsymbol{q_j}|}{\|\boldsymbol{q_i}\| \cdot \|\boldsymbol{q_j}\|}\right) \tag{65}$$

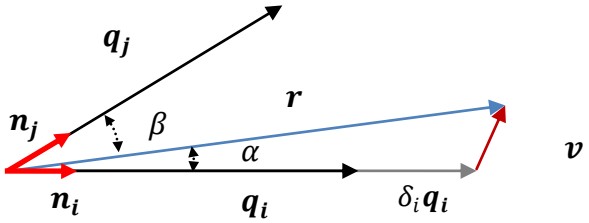


Figure 13 Effect of noise on the residual vector

In order to isolate the i^{th} sensor as faulty, $\boldsymbol{q_i}$ should be closer to the direction of the residual than $\boldsymbol{q_j}$, (in Figure 2, $\alpha < \beta$):

$$\frac{|\boldsymbol{r} \cdot \boldsymbol{q_i}|}{\|\boldsymbol{r}\| \cdot \|\boldsymbol{q_i}\|} > \frac{|\boldsymbol{r} \cdot \boldsymbol{q_j}|}{\|\boldsymbol{r}\| \cdot \|\boldsymbol{q_j}\|}. \tag{66}$$

Substituting the value of $\boldsymbol{r}$ from Equation 27, we have:

$$\frac{|(\delta_i \boldsymbol{q_i} + \boldsymbol{v}) \cdot \boldsymbol{q_i}|}{\|\delta_i \boldsymbol{q_i} + \boldsymbol{v}\| \cdot \|\boldsymbol{q_i}\|} > \frac{\left|(\delta_i \boldsymbol{q_i} + \boldsymbol{v}) \cdot \boldsymbol{q_j}\right|}{\|\delta_i \boldsymbol{q_i} + \boldsymbol{v}\| \cdot \left\|\boldsymbol{q_j}\right\|}. \tag{67}$$

Extracting δ_i from both sides, we have (Appendix B):

$$\boldsymbol{\delta_i} > \frac{\boldsymbol{\rho_{ij}}}{\left(\mathbf{1} - \left|\boldsymbol{n_i} \cdot \boldsymbol{n_j}\right|\right)\|\boldsymbol{q_i}\|} \tag{68}$$

where $\boldsymbol{n}_i = \frac{\boldsymbol{q_i}}{\|\boldsymbol{q_i}\|}$ $(i = 1,2, \dots n)$, $\rho_{ij} = \left|\boldsymbol{v} \cdot \boldsymbol{n}_{ij}\right|$ and $\boldsymbol{n}_{ij}$ is defined as following:

$$\boldsymbol{n_{ij}} = \begin{cases} \boldsymbol{n_i} + \boldsymbol{n_j}, & \boldsymbol{n_i} \cdot \boldsymbol{n_j} < 0 \\ \boldsymbol{n_i} - \boldsymbol{n_j}, & \boldsymbol{n_i} \cdot \boldsymbol{n_j} \geq \mathbf{0} \end{cases}. \tag{69}$$

In Equation (68), all parameters, except ρ_{ij} , are exclusively related to the structure of the parity equation. The scalar parameter ρ_{ij} is related to the noise vector. Assuming the measurement noise is a white Gaussian noise with a known parameters, $\boldsymbol{\eta} \sim \boldsymbol{N}(0, \boldsymbol{\Sigma})$, the probability distribution of ρ_{ij} is also Gaussian; $\rho_{ij} \sim N(0, \sigma)$ where

$$\sigma_{ij}{}^2 = \left(\boldsymbol{\Sigma_r n_{ij}}\right) \cdot \boldsymbol{n_{ij}} \tag{70}$$

with $\boldsymbol{\Sigma}_r$ as the covariance of $\boldsymbol{v}$. Note that $\boldsymbol{\Sigma}_r$ is related to the covariance of noise:

$$\boldsymbol{\Sigma}_r = \boldsymbol{Q\Sigma Q}^T. \tag{71}$$

The confidence limit for δ_i i.e. δ_{i_α} is found based on the confidence limit of ρ_{ij} :

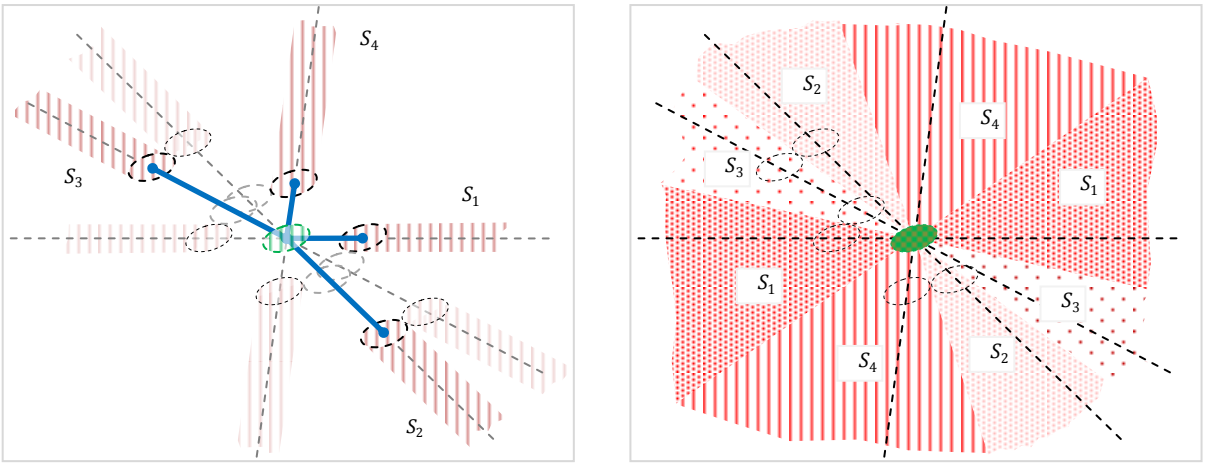
$$\delta_{i_\alpha} = \frac{\rho_{ij_\alpha}}{\mathbf{1} - \left| \boldsymbol{n_i} \cdot \boldsymbol{n_j} \right| \|\boldsymbol{q_i}\|} \tag{72}$$

where α is the level of significance.

Notice that this margin is found for a single set of measurements. In sensor fault diagnosis problems, we usually have continuous flow of measurements which helps to reduce the minimum isolation limit.

Figure 14 shows the two different isolation criteria. The left panel (a) shows the faulty regions attributed to each sensor based on a certain confidence limit. e.g. if the residual

falls in the region of senor S_1, with 95% certainty the sensor S_1 is faulty. The minimum isolable fault magnitude is proportional to the length of the widened solid line shown in this figure for each sensor. This figure shows how the minimum detectable fault magnitude is affected by the structure of minimal parity equation (i.e. the angle between different sensor fault image vectors) and the noise model (i.e. the size, shape and angle of the dashed ellipses). Panel (b) show different regions of sensor faults in the residual space based on maximum probability. e.g. if the residual falls in this region of senor S_1, the probability that this sensor is faulty is more than the probability of other sensors being faulty.



(b) Isolation based on confidence limit for each sensor (a) Isolation based on competitive probability

Figure 14 Graphical demonstration of isolability limit for each sensor

Algorithm of Single Fault Isolation in Real-Time Systems

As explained earlier, once the minimal parity equation is found, the detection and isolation of a single fault in the system can be easily done. Detection of fault in the system is performed by the statistical monitoring of the residual vector. Assuming there is a normally distributed noise in the system, the fault detection index is defined as [Qin and Li 1999]:

$$\boldsymbol{f = r^T \Sigma_r^{-1} r}. \tag{73}$$

When no fault is present, the fault detection index follows a Chi-squared distribution with l degrees of freedom where l is the dimension of the residual vector ($l = n - k$).

$$\boldsymbol{f \sim \chi(l)} \tag{74}$$

Therefore, the α significant level for f is

$$\boldsymbol{f_\alpha \sim \chi_\alpha(l)} \tag{75}$$

For the isolation of the faulty sensor, two different methods are presented. The first method is the calculation of the sensor failure index introduced in Eq. (61). Once the system is flagged as faulty, SFI for each sensor is calculated and the sensor with the maximum SFI is isolated as faulty.

As explained before, as long as the magnitude of the fault is large compared to the noise level, the angle between the residual vector and the image vector of the single faulty sensor is the lowest. As a result, the SFI of the faulty sensor is highest among SFIs of all sensors.

Another method of isolation is by forming *sensor residuals*. We define sensor residual as:

$$\boldsymbol{r_i} = \boldsymbol{N_i r}, \tag{76}$$

where $\boldsymbol{N}_i$ is the matrix containing orthonormal vectors that spans the orthogonal space of $\boldsymbol{q}_i$ ($\boldsymbol{N}_i \boldsymbol{q}_i = 0$).

Therefore, when the i^{th} sensor is faulty, the sensor residual for i^{th} sensor is:

$$\boldsymbol{r_i} = \boldsymbol{N_i}(\delta_i \boldsymbol{q_i} + \boldsymbol{v}) = \boldsymbol{N_i v}. \tag{77}$$

Sine $\boldsymbol{v}$ has a Gaussian distribution with known parameters, the distribution of $\boldsymbol{r}_i = \boldsymbol{N_i v}$ is found as $\boldsymbol{r}_i \sim N(0, \boldsymbol{\Theta}_i)$, where:

$$\boldsymbol{\Theta_i} = \boldsymbol{N_i^T \Sigma_i N_i}. \tag{78}$$

On the other hand, for other healthy sensors, the distribution of the sensor residual has a nonzero mean value which is related to the magnitude of error.

$$\boldsymbol{r_j} = \boldsymbol{N_j}(\delta_i \boldsymbol{q_i} + \boldsymbol{v}) = \delta_i \boldsymbol{N_j q_i} + \boldsymbol{N_i v} \tag{79}$$

Therefore, the distributions of the healthy residuals are $\boldsymbol{r_k} \sim N(\delta_i \boldsymbol{N}_j \boldsymbol{q}_i, \boldsymbol{\Theta}_i)$ where $\boldsymbol{\Theta_i}$ is defined in (78).

In summary, the procedure for online fault diagnosis is explained in the following algorithm.

Detection Algorithm

1. Take new measurements $\boldsymbol{y}$
2. Find the minimal residual based on measurements $\boldsymbol{r} = \boldsymbol{Q}\boldsymbol{y}$
3. Find the fault detection index $f = \boldsymbol{r}^T {\boldsymbol{\Sigma}_r}^{-1} \boldsymbol{r}$
4. If $f < f_\alpha$ flag the system as a healthy and go to Step 1. Otherwise go to the Isolation algorithm

Isolation Algorithm by Sensor Failure Index

1. Normalize the residual vector $\boldsymbol{n_r} = \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|}$
2. Find the sensor failure index for all sensors $\boldsymbol{f_i} = \boldsymbol{n_r} \cdot \boldsymbol{n_i}$ $(i = 1 \ldots n)$
3. Find the maximum sensor failure index and flag its corresponding sensor as faulty
4. Return to the detection

Isolation Algorithm by Sensor Residual

1. Form the sensor residual for all sensors $\boldsymbol{r_i} = \boldsymbol{N_i}\boldsymbol{r}$
2. Find the minimum sensor residual and flag its corresponding sensor as faulty
3. Return to the detection

Extension of Results to the Dynamic Redundancy Methods

In the algorithm of sensor fault diagnosis and the derivation of Detectability and Isolability, only the information from the "direct redundancy" of the sensors has been used. This approach can cover three different types of problems. The first one is a case where the degree of direct redundancy is high enough that we do not need to use the information from the dynamic of the system for the purpose of sensor fault detection. The second observation about this approach is that the dominant dynamic of the system is so slow that we can ignore the dynamic of the system and the third type is general systems but the algorithm is used in the static condition of the system ($\dot{\boldsymbol{x}} = 0$).

Either the degree of direct redundancy is enough to be able to isolate the faulty sensor or not, the information from the dynamic of the system can always enhance the performance of fault detection and Isolation. Also, when there is not enough direct redundancy for sensor fault detection, the application of the information from the dynamic of the system becomes necessary. In this part, the application of dynamic of the system to the PCA is explained and two simple examples are preformed to show the advantage of using the dynamic behavior of the system.

In the dynamic model, instead of using correlation matrix, we find *auto-correlation* by using *Augmented PCA*. Augmented PCA is used just by adding a time shifted values of the data to the same matrix of data. Figure 15 shows how to generate augmented data.

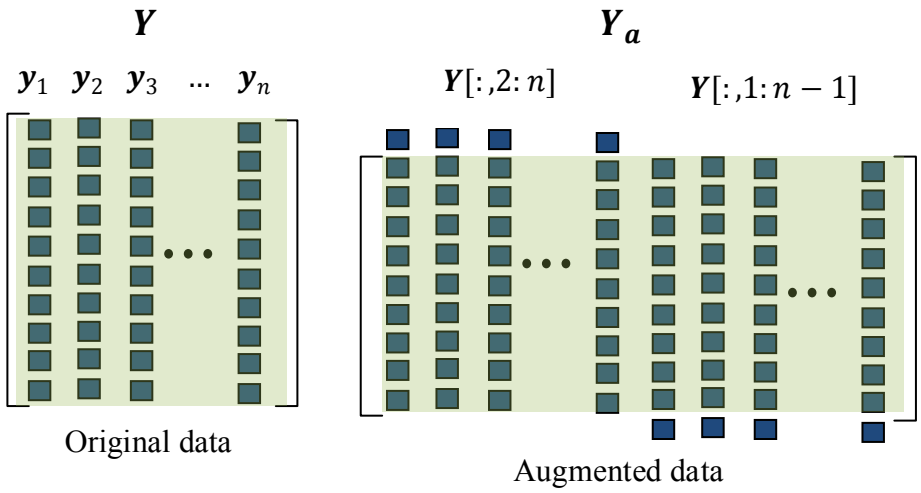


Figure 15 Augmented PCA

The rest of algorithm is exactly the same and direct redundancy method. This method is in fact equivalent to the combination of system identification and Kalman filter. For example assume we have a system as following:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) \end{aligned} \tag{80}$$

If we measure, $\boldsymbol{y}(k) = \boldsymbol{x}(k)$ or $\{\boldsymbol{C} = \boldsymbol{I}, \boldsymbol{D} = \boldsymbol{0}\}$ and then form the augmented data $\boldsymbol{Y_a}$ as explained, we can decompose $\boldsymbol{y_a}$ (= a representative variable from this system) into the principal components and the residual components, we have:

$$\boldsymbol{Q y_a = \nu} \tag{81}$$

where $\boldsymbol{y_a} = [\boldsymbol{x_{n+1}}\ \boldsymbol{x_n}\ \boldsymbol{u}]^T$

Accordingly we define $\boldsymbol{Q} = [\boldsymbol{Q_1}\ \boldsymbol{Q_2}\ \boldsymbol{Q_3}]$

Therefore we have

$$[\boldsymbol{Q_1}\ \boldsymbol{Q_2}\ \boldsymbol{Q_3}]\begin{bmatrix}\boldsymbol{x}(k+1)\\ \boldsymbol{x}(k)\\ \boldsymbol{u}\end{bmatrix} = \boldsymbol{\nu} \tag{82}$$

Or

$$\boldsymbol{Q_1 x_{n+1} + Q_2 x_n + Q_3 u = \nu} \tag{83}$$

This is equivalent to the general dynamic systems model (Eq. (85)), where,

$$\boldsymbol{A = -Q_1^{-1} Q_2}$$

$$\boldsymbol{B = -Q_1^{-1} Q_3}$$

$$\boldsymbol{C = I}$$

$$\boldsymbol{D = 0}$$

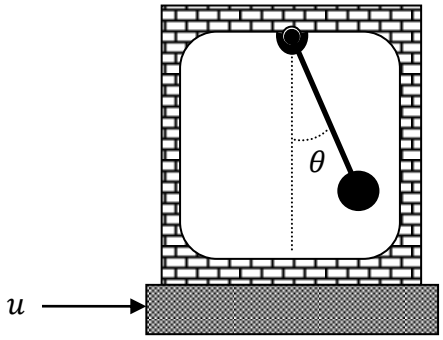
Notice that this formulation is valid only if we measure all stated of the system. In general systems we may not be able to measure all states of the system in these cases, in

order to extract the full dynamic of the system, we need to have further data augmentation. So in general case the dynamic system model would be in this form:

$$\begin{aligned} \boldsymbol{x}(k+\mathrm{s}) &= \boldsymbol{A_1 x}(k) + \boldsymbol{A_2 x}(k+1) + \cdots + \boldsymbol{A_{s-1} x}(k+s-1) + \boldsymbol{Bu}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{Cx}(k) + \boldsymbol{Du}(k), \end{aligned} \tag{84}$$

which can be easily reformulated into the general form of Eq. (85). This method has been explained in the following example.

Example 1: The pendulum box shown in Figure 16 is subject to random vibration of its base. Assume we have a history of measurements of the base vibrations u and the angle of the pendulum θ; we need to develop and algorithm to detect the fault in the angle sensor.



$u(mm)$	$\theta(rad)$
0.023	0.2411
-1.217	0.1931
-0.621	0.1323
1.901	0.0381
⋮	⋮
-0.726	0.2321

Figure 16 The model of pendulum

In this problem we have only one measured state where as the order of the system is two. Clearly there is no direct redundancy. Therefore, we need to use augmented PCA in order to find dynamic or temporal redundancy. We define the augmented database matrix as:

$$\boldsymbol{Y}_a = [\theta(k+2) \;\; \theta(k+1) \;\; \theta(k) \;\; u(k)].$$

The covariance of these data is equal to:

$$C = \begin{bmatrix} 0.0499 & 0.0459 & 0.0459 & 0.0001 \\ 0.0459 & 0.0499 & 0.0459 & 0 \\ 0.0459 & 0.0459 & 0.0499 & 0 \\ 0.0001 & 0 & 0 & 0.0099 \end{bmatrix},$$

which can be decomposed into:

$$\boldsymbol{C} = \begin{bmatrix} 0.407 & -0.708 & -0.002 & 0.577 \\ -0.817 & 0.001 & 0.000 & 0.577 \\ 0.409 & 0.706 & 0.003 & 0.577 \\ 0 & 0.004 & -1.0 & 0.001 \end{bmatrix} \begin{bmatrix} 0.004 & 0 & 0 & 0 \\ 0 & 0.004 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.142 \end{bmatrix} \begin{bmatrix} 0.407 & -0.817 & 0.409 & 0 \\ 0.708 & 0.001 & 0.706 & 0.004 \\ -0.002 & 0 & 0.003 & -1 \\ 0.577 & 0.577 & 0.577 & 0.001 \end{bmatrix}$$

The eigenvalues are shown in Figure 17

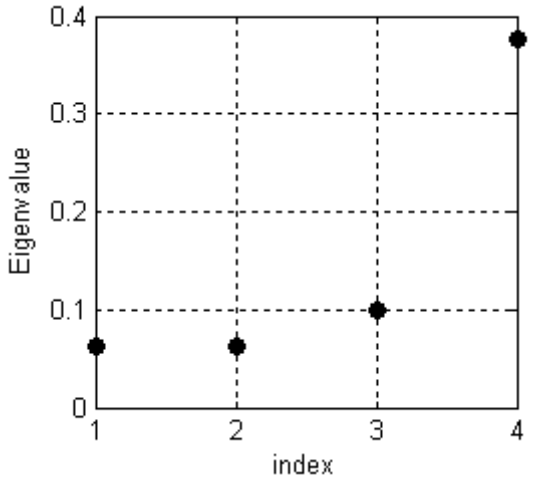


Figure 17 PCA analysis of example 1

If we assume the first third eigenvalues are due to the noise, we can find the residual generator as:

$$\boldsymbol{Q} = \begin{bmatrix} 0.407 & -0.817 & 0.409 & 0 \\ 0.708 & 0.001 & 0.706 & 0.004 \\ -0.002 & 0 & 0.003 & -1 \end{bmatrix}$$

$$\boldsymbol{\Sigma_r = QCQ^T} = \begin{bmatrix} 0.004 & 0 & 0 \\ 0 & 0.004 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$$

Then, the residual vector is found as $\boldsymbol{r = Q}[\theta(k+2)\ \ \theta(k+1)\ \ \theta(k)\ \ u(k)]^{\boldsymbol{T}}$, and the fault detection index is found using $f = \boldsymbol{r^T \Sigma_r}^{-1}\boldsymbol{r}$. The result of fault detection index for a set of artificially erroneous data are shown in Figure 18

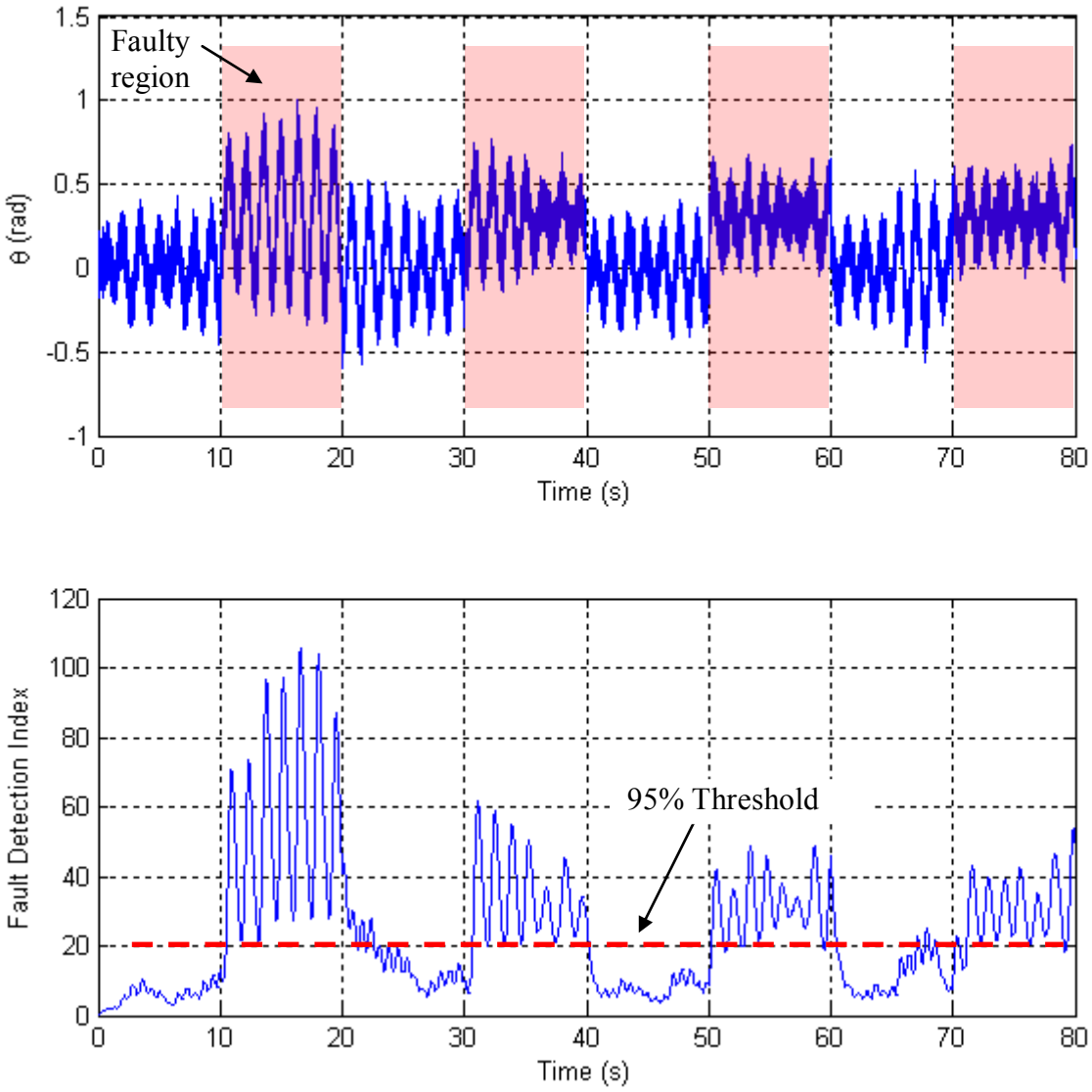


Figure 18 Dynamic fault diagnosis result of example 1

CHAPTER V

PROBABILISTIC METHOD OF SENSOR FAULT DIAGNOSIS

Introduction to the Redundancy Models

Depending on the type of the knowledge that we have about the system, different categories of FDI has been proposed in the literature. When the physical model of the system is available in the form of input-output model, we have a *systematic knowledge* about the system and model based methods are used. When the knowledge is in the form of a database of numerical records of the system during the healthy and faulty conditions, the data driven method is used which is usually in the form of statistical analysis methods. These two methods are among the quantitative methods however there is another category of methods called *qualitative methods* which is used when the knowledge of the system is in the form of cause-effect associations learned from history of the system. This kind of knowledge is called *expertise knowledge* and the methods the cased-base method are used for the diagnosis purposes [Lampis and Andrews 2009][Venkatasubramanian et al. 2003].

Generally, model based FDI methods use some form of redundancy to detect the inconsistency. This redundancy can be in the form of hardware redundancy or analytical redundancy. The hardware redundancy requires application of two or more sensors to measure a single variable of the system [Willsky 1976]. In such cases voting techniques are often used for fault detection. Voting schemes are easy to implement and are quick to

identify mechanical faults in instruments. However, the application of hardware redundancy is limited due to cost and design limitations.

Analytical redundancy, on the other hand is based on the inherent relationships between different system variables. These relationships are usually represented the mathematical model of the system. When this model is a simple algebraic relationship between the measured variables, we have *direct redundancy* or *algebraic redundancy* but when it is a dynamic output-model or state space model, we have *temporal redundancy* or *dynamic redundancy* [Frank 1990]. Since the models provide redundant measurements in addition to the real measurements from the plant, we can also call them redundancy models.

The first step toward the fault detection is to form some residuals based on the redundancy model. The residuals are the differences between the measured variables and their expected value from the redundancy model. Therefore, any discrepancy in the system will results in a nonzero residual. Usually, due to the noise in the system and inaccuracy of the model, the residuals are always nonzero, but in a healthy system the residual should have a distribution with the expected value of zero.

A wide range of different methods of residual generation is introduced in the literature, the most straightforward method is to estimate the states of the system based on the measurements using Kalman filter [Frank 1990], or Least square method [Isermann 1997] and then use these estimated states to find the expected value of the measurements. These methods, known as observer-based FDI [Venkatasubramanian et al. 2003], develop

a set of observers, each one of which is sensitive to a subset of faults while insensitive to the remaining faults and unknown inputs.

The more convenient method of residual generation is through *parity equations*. Parity equations are a rearranged mathematical representation of the input-output or state space model of the system that directly generates the residuals from the measurements [Chow and Willsky 1984], [Gertler and Singer 1990]. One of the main requirements of the parity equations is their robustness i.e. the residuals generated by these parity equations should have minimum sensitivity to the uncertainties and noise. Also, they should be able to isolate different fault in the system. Fault isolation requires the ability to generate residual vectors which are orthogonal to each other [Ben-Haim 1980, 1983].

In order to facilitate the isolation, two major methods of enhancement of residuals have been subjected: *Structural Residuals* and *Directional Residual*. In structural residual approach, the residuals are designed such that a unique subset of residuals is orthogonal to a specific fault. Therefore when that fault happens, the expected value of all residuals become nonzero except those corresponding to the fault [Gertler and Singer 1990]. In directional residual approach, the residual are designed such that each fault makes the expected residual vector to be a nonzero vector with a specific direction corresponding to that fault.

Decision Process

The next step after residual generation is the decision process [Chow and Willsky 1984]. In the decision process, the residuals are examined to first detect the fault, and then isolate the fault in the system. A decision process may consist of a simple threshold test on the residuals, moving averages of the residuals, or it may be directly based on advanced methods of statistical decision theories.

Although the papers in the area of FDI usually present a method for both of these stages, the main focus of most of them is on the first stage i.e. to find the most effective residuals such that they have minimum sensitivity to the disturbances and can efficiently detect and isolate the faults. Typically, in these papers, the decision process itself has two phases. The first one is to detect the existence of the fault and the second one is to isolate the sources of the fault [Venkatasubramanian et al. 2003]. Moreover, the final decision is in the binary form of either faulty or healthy.

In contrast, in this chapter the main focus is on the decision process. In the presented method, the residual generation is suggested through a simple equation which is called *Minimal Parity Equation*. Based on this equation, a probabilistic model is presented that can find the probability of error in each sensor separately. Therefore, the detection and isolation is performed together.

There are several advantages of this probabilistic model to the existing methods

1. The probabilistic model provides a better sense about the certainty of the decision and it provides more meaningful information to the end user.
2. The result of this analysis can be easily combined with other sources of information about the health condition of the system
3. In some cases that the fault is not isolable between two or more sensors, the algorithm provides equal probability of error in each one in the group such that the user can recognize the group of faulty sensors that has higher probability of error.
4. The detection and Isolation are performed concurrently.

Several researcher has used the probabilistic models for the sensor fault diagnosis these method are based on Bayesian Belief Networks(BBN) [Ibarguengoytia et al. 2006] [Maheshvari and Hakimi 1976] Fault trees [Huldle 2009] or combination of both [Andrews 2009]. These methods are mainly advantageous when there is no quantitative knowledge about the system. The difference between the probabilistic method presented in this chapter and the BBN based methods is that the former uses a residual generator which is based on an input-output model and then uses the probabilistic analysis on the residuals, whereas the later uses the "Expertise knowledge" to find a probabilistic cause-symptom model.

The main assumption made for derivation of the probabilities of error is that there is maximum one faulty sensor in the system. However, this method can be extended to multiple fault detection depending on the degree of redundancy in the model.

Probabilistic Decision Process

Assume sensors $S_1, S_2 \ldots S_n$ are used to measure the variable in the output vector Next assume the measurement in each Sensor S_i has an error with the fixed magnitude of δ_i, we have:

$$\boldsymbol{y} = \boldsymbol{y}^* + \boldsymbol{n} + \boldsymbol{\Delta}, \tag{85}$$

where $\boldsymbol{y}^*$ is the expected value of the measured variable, $\boldsymbol{n}$ is the noise and $\boldsymbol{\Delta} = [\delta_1\ \delta_2 \ldots \delta_n\ ]^{\boldsymbol{T}}$ is the vector of measurement errors.

Assuming the noise has a Gaussian distribution with known parameters,

$$\boldsymbol{n} \propto N(0, \boldsymbol{\Sigma}), \tag{86}$$

the distribution of $\boldsymbol{y}$ would become:

$$\boldsymbol{y} \propto N(\boldsymbol{y}^* + \boldsymbol{\Delta}, \boldsymbol{\Sigma}). \tag{87}$$

Since the residual of the expected value is zero, $\boldsymbol{Qy}^* = \mathbf{0}$, the distribution of the residual is just dependent on the noise and the fault.

$$\boldsymbol{r} \propto N\left(\sum_{j=1}^{n} \delta_j \boldsymbol{q}_j \, , \boldsymbol{\Sigma}_{\mathbf{r}}\right) \tag{88}$$

where $\boldsymbol{\Sigma}_{\mathbf{r}} = \boldsymbol{Q\Sigma Q^T}$ and $\boldsymbol{q}_j$ is the j^{th} column of $\boldsymbol{Q}$ called fault image vector.

Eq. (88) shows the distribution residual in the general case, however if we assume only one sensor can be faulty at a time, it will simplify to (Assuming the faulty Sensor is S_i)

$$\boldsymbol{r} \propto N(\delta_i \boldsymbol{q}_i, \boldsymbol{\Sigma}_{\mathbf{r}}). \tag{89}$$

Based on this distribution the following conditional probabilities can be found:

$$p(\boldsymbol{r}|\mathcal{H}) = N(0, \boldsymbol{\Sigma}_{\mathbf{r}}). \tag{90}$$

$$p(\boldsymbol{r}|\mathcal{S}_i, \delta_i) = N(\delta_i \boldsymbol{q}_i, \boldsymbol{\Sigma}_{\mathbf{r}}). \tag{91}$$

where $p(\boldsymbol{r}|\mathcal{H})$ is the probability distribution of $\boldsymbol{r}$ when the system is healthy and $p(\boldsymbol{r}|\mathcal{S}_i, \delta_i)$ is probability distribution of $\boldsymbol{r}$ when there is a error in measurement of Sensor S_i with the magnitude of δ_i.

Note that this distribution is dependent on the unknown variable δ_i but we can marginalize this variable by integrating it over δ_i,

$$p(\boldsymbol{r}|\mathcal{S}_i) = \int p(\boldsymbol{r}|\mathcal{S}_i, \delta_i)\, p(\delta_i) \mathrm{d}\delta_i. \tag{92}$$

Assuming δ_i has a uniform distribution, (there is no a priori about the magnitude of fault) the result of this integration is:

$$p(\boldsymbol{r}|\mathcal{S}_i) = p(\boldsymbol{r}_i|\mathcal{S}_i) = N(0, \boldsymbol{\Sigma}^k), \tag{93}$$

where $\boldsymbol{r}_i$ is defined as directional residual for the i^{th} sensor and $\boldsymbol{r}_i = \boldsymbol{Q}_i \boldsymbol{r}$, $\boldsymbol{\Sigma}^i = \boldsymbol{Q}_i \boldsymbol{\Sigma}_{\mathbf{r}} \boldsymbol{Q}_i{}^T$, and $\boldsymbol{Q}_i$ is the matrix containing orthonormal vectors in the null space of $\boldsymbol{q}_i$.

Now, in order to find the posterior probabilities or the probability of each sensor faulty given an observed value of $\boldsymbol{r}$, we can use the Bayes' formula:

$$p(\mathcal{S}_i|\boldsymbol{r}) = \frac{p(\boldsymbol{r}|\mathcal{S}_i)p(\mathcal{S}_i)}{p(\boldsymbol{r})}, \tag{94}$$

$$p(\mathcal{H}|\boldsymbol{r}) = \frac{p(\boldsymbol{r}|\mathcal{H})p(\mathcal{H})}{p(\boldsymbol{r})}, \tag{95}$$

where the marginal probability can be found as:

$$p(\boldsymbol{r}) = p(\boldsymbol{r}|\mathcal{H})p(\mathcal{H}) + \sum_{i=1}^{n} p(\boldsymbol{r}|\mathcal{S}_i)p(\mathcal{S}_i). \tag{96}$$

The a priori's $p(\mathcal{H})$ and $p(\mathcal{S}_i)$ can also be estimated by their probability given the previous set of observations:

$$p(\mathcal{H}) = p(\mathcal{H}|\boldsymbol{r}^{<old>}) \tag{97}$$

$$p(\mathcal{S}_i) = p(\mathcal{S}_i|\boldsymbol{r}^{<old>}) \tag{98}$$

where $\boldsymbol{r}^{<old>}$ is the value of residual from the previous measurement.

Study of Sensor Fault Detectability

In a set of sensors measuring variables of a system, based on the degree of redundancy and the association of measured variables *i*n the redundancy relationship, some sensors

might not be detectable. It means that their contribution to the primary residual is zero or so small that if a fault happens in those sensors, the change in the residual cannot be detected. One fast method to find non-detectable sensors is to check their fault image vectors**.** The sensors, whose fault image vectors have very low magnitude, would probably be non-detectable. However, this method is however just a rough estimation. The more accurate answer should also consider the noise level and especially the direction of the noise. For example, in the system shown in figure 19, although the fault image vector of Sensor S_3 is smaller than that of Sensor S_1, it is easier to detect a fault with the same relative magnitude in Sensor S_3 rather than S_1. In this figure, a two dimensional residual space is shown for a system of 3 redundant sensors. The 3 solid lines are the sensor fault images corresponding to these three sensors and the dashed ellipse is the contour of constant probability density of having a healthy system in the residual space. Now assume two cases where a single fault with relative magnitude of α happens in Sensor S_1 and Sensor S_3 base on this figure, the probability density of first case is less than that of the second $(\boldsymbol{r} = \alpha \boldsymbol{q}_1 \,|\mathcal{H}) > p(\boldsymbol{r} = \alpha \boldsymbol{q}_3 \,|\mathcal{H})$. Consequently, their posterior probabilities will also have the same relationship $p(\mathcal{H} \,|\boldsymbol{r} = \alpha \boldsymbol{q}_1) > p(\mathcal{H} \,|\boldsymbol{r} = \alpha \boldsymbol{q}_3)$ which means that the estimated probability of error for the first case is less than that of the second case.

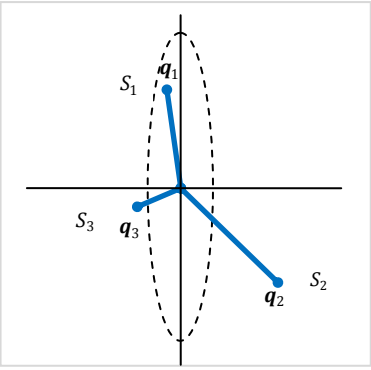


Figure 19 Different level of detectability in sensors

A quantitative measure for the level of detectability of sensors can be the value of probability density of for the case that a single fault with a unit magnitude $\alpha = 1$ happens given that the system is healthy:

$$DM_i = p(\boldsymbol{r} = \boldsymbol{q}_i \,|\mathcal{H}) \tag{99}$$

where DM_i is the detectability index for Sensor i. The more this number is, the less is the probability to detect a fault in that Sensor.

CHAPTER VI

NONLINEAR SENSOR FAULT DETECTION

In this chapter, a methodology for sensor fault diagnosis in nonlinear systems using Mixture of probabilistic principal component analysis (MPPCA) is introduced. MPPCA algorithm separates the input space or measurement space into several local linear regions. Each region is associated with a Probabilistic PCA model which is characterized with a Gaussian distribution model with reduced dimension and with the unit vectors of its principal subspace. Therefore the detectability and isolability properties of sensor are analyzed separately in each region. During the real-time monitoring of the sensors, first the local linear region and its corresponding PCA model is recognized using the probability distribution of the Gaussian component model. Then, the residuals are generated based on the recognized PCA model and using the probabilistic analysis of the residuals, probability of having error in each sensor is estimated.

Introduction

In the linear sensor fault diagnosis, a methodology was presented to detect and reconstruct the faulty sensor among a set of analytically redundant sensors [Sharifi and Langari 2009]. In that approach, the principal component analysis technique provides a static linear model and a parity equation of the system. This parity equation was then used to validate the future measurements of the system and to detect and reconstruct the abnormality in the measurements.

However, linear methods are applicable as long as the relationship between the sensors is linear or it can be approximated as a linear correlation with adequate accuracy. Therefore, it is needed to expand the method to cover the nonlinear systems as well. Unfortunately, in the literature of data driven sensor fault diagnosis, the nonlinear methods are developed almost independent of the linear methods. Consequently, there is not a sound theoretical background for nonlinear approaches. For example, one of the most frequently used methods of nonlinear sensor fault diagnosis is through the application of Autoassociative Artificial Neural Network (AANN) introduced by Kramer [Kramer 1992]. In this method, the expectation is that with a perfectly trained AANN, the absolute error between the measurement values (input of AANN) and the estimated values (output of AANN) is highest in the faulty sensor. However, this expectation is contradictory to the proven results found in the linear approaches [Dunia et. al. 1995].

In this approach, MPPCA method is used which is based on separation of the input space into several local linear areas. As a result, we are able to apply the linear sensor fault diagnosis results. Also, the method of KPCA helps to achieve model selection in MPPCA.

Probabilistic PCA

Before explaining MPPCA, it is needed to have a close look at probabilistic PCA to clarify the advantages of probabilistic approach especially when it is used in sensor fault diagnosis. PPCA is an expression of PCA as the maximum likelihood solution of a probabilistic latent variable model. This probabilistic model has two advantages.

Therefore, when we have some information about uncertainty of measurements, we can apply it to the model to prepare a more precise model. Moreover with a probabilistic model, Mixture of probabilistic PCA models can be formulated in a principled way and be trained using Expectation-Maximization (EM) algorithm [Dinov 2008].

Classical PCA can be written as a probability density model by using a latent variable approach, in which the data $\boldsymbol{y} \in \boldsymbol{\mathcal{R}}^n$ is expressed by a linear combination of hidden variable $\boldsymbol{x} \in \boldsymbol{\mathcal{R}}^r$ with fewer dimensions ($r < n$) [Tipping and Bishop 2002].

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x} + \boldsymbol{\mu} + \boldsymbol{v} \tag{100}$$

where the hidden variable $\boldsymbol{x}$ has a Gaussian distribution with zero mean and unit isotropic variance, $\boldsymbol{x} \sim N(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{\mu}$ is constant and $\boldsymbol{v}$ is the process noise which is dependent on $\boldsymbol{y}$ which is equivalent of measurement uncertainty in the context of sensor fault diagnosis and P is the projection matrix. Usually the noise model is a Gaussian process with zero mean and a diagonal covariance matrix, $\boldsymbol{v} \sim N(\boldsymbol{0}, \boldsymbol{\Phi})$. Therefore the model for $\boldsymbol{y}$ has a normal distribution $\boldsymbol{x} \sim N(\boldsymbol{\mu}, \boldsymbol{P}\boldsymbol{P}^{\boldsymbol{T}} + \boldsymbol{\Phi})$. Since $\boldsymbol{\Phi}$ is diagonal, the observed variables $\boldsymbol{y}$ are conditionally independent given the values of latent variables $\boldsymbol{x}$.

If we assume the noise model has isotropic variance, $\boldsymbol{\Phi} = \sigma^2\boldsymbol{I}$, the probability model for PPCA can be written as a combination of the conditional distribution:

$$\boldsymbol{p(y|x) = \frac{1}{(2\pi\sigma^2)^{n/2}} exp\left(-\frac{\|y - Px - \mu\|^2}{2\sigma^2}\right)} \tag{101}$$

and the latent variable distribution:

$$\boldsymbol{p(x) = \frac{1}{(2\pi)^{r/2}} exp\left(-\frac{x^T x}{2}\right)} \tag{102}$$

The marginal distribution of $\boldsymbol{y}$ is then obtained by integrating out the latent variables $\boldsymbol{x}$**:**

$$\boldsymbol{p(y) = N(\mu, \Sigma) = \frac{1}{(2\pi\sigma^2)^{n/2}\Sigma^{1/2}} exp\left(-\frac{1}{2}(y - \mu)\Sigma^{-1}(y - \mu)\right)} \tag{103}$$

where

$$\boldsymbol{\Sigma = PP^T + \sigma^2 I} \tag{104}$$

To fit this model to the m set of measurements, we use log-likelihood as an error measure:

$$\boldsymbol{\mathcal{L} = \sum_{i=1}^{m} -\frac{m}{2}\left[n\, log(2\pi) + log|\Sigma| + tr(\Sigma^{-1}S)\right]}, \tag{105}$$

where

$$\boldsymbol{S} = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{y}_i - \boldsymbol{\mu})(\boldsymbol{y}_i - \boldsymbol{\mu})^T \tag{106}$$

is the sample covariance of the observed data, provided that $\boldsymbol{\mu}$ is set to its maximum likelihood estimate, which is the sample mean. Estimates of $\boldsymbol{P}$ and σ^2 can be obtained by an iterative maximization of $\mathcal{L}$ using an EM. But, in this case it is possible to find an analytic solution for the maximum likelihood estimate:

$$\boldsymbol{P}_{ML} = \boldsymbol{U}_r(\boldsymbol{\Lambda}_{\mathrm{r}} - \sigma^2\boldsymbol{I})^{1/2}\boldsymbol{R} \tag{107}$$

where $\boldsymbol{U}_r$ is a matrix with r column vector which are corresponding to the first r eigenvectors of $\boldsymbol{S}$ and $\boldsymbol{\Lambda}_r$ is a diagonal matrix which has the first r eigenvalues of $\boldsymbol{S}$ as its diagonal elements. Matrix $\boldsymbol{R}$ is an arbitrary $r \times r$ orthonormal (rotation) matrix.

If we substitute maximum likelihood solution for $\boldsymbol{P}$, then the maximum likelihood estimator for σ^2 is given by:

$$\sigma_{ML}^2 = \frac{1}{n-r}\sum_{j=r+1}^{n} \lambda_j \tag{108}$$

which is in fact the equivalent noise of the system or the average of variances eliminated in the mapping to the latent space.

If we assume that that we have the parameter of the noise model ($\boldsymbol{\Phi}$ is known) likelihood estimator for σ^2 is related to the real noise by:

$$\sigma_{ML}^2 = \frac{1}{n-r}\sum_{j=r+1}^{n} \boldsymbol{u}_j^T \boldsymbol{\Phi} \boldsymbol{u}_j \tag{109}$$

where $\boldsymbol{u}_j$ is the j^{th} eigenvectors of $\boldsymbol{S}$

Figure 20 graphically illustrates how the noise in the measurement in two dimensions is related to the estimated noise by the assumption of isotropic noise. In this simple example, the covariance matrix of noise in the system $\boldsymbol{\Phi}$ is known and it is diagonal. The dashed ellipse is the one standard deviation contour of the noise covariance. And the dashed circle is the modeled noise with PPCA. Assuming $\boldsymbol{\sigma_1}$ and $\boldsymbol{\sigma_2}$ are the standard deviation errors of measurements $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$,

$$\Phi = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}, \tag{110}$$

the estimated standard deviation error σ is found as:

$$\sigma^2 = \boldsymbol{u}_2^{\mathrm{T}} \Phi \boldsymbol{u}_2 = \sigma_1^2 u_{2,y_1} + \sigma_2^2 u_{2,y_2}, \tag{111}$$

where u_{2,y_1} and u_{2,y_2} are the components of u_2 in direction of $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$,

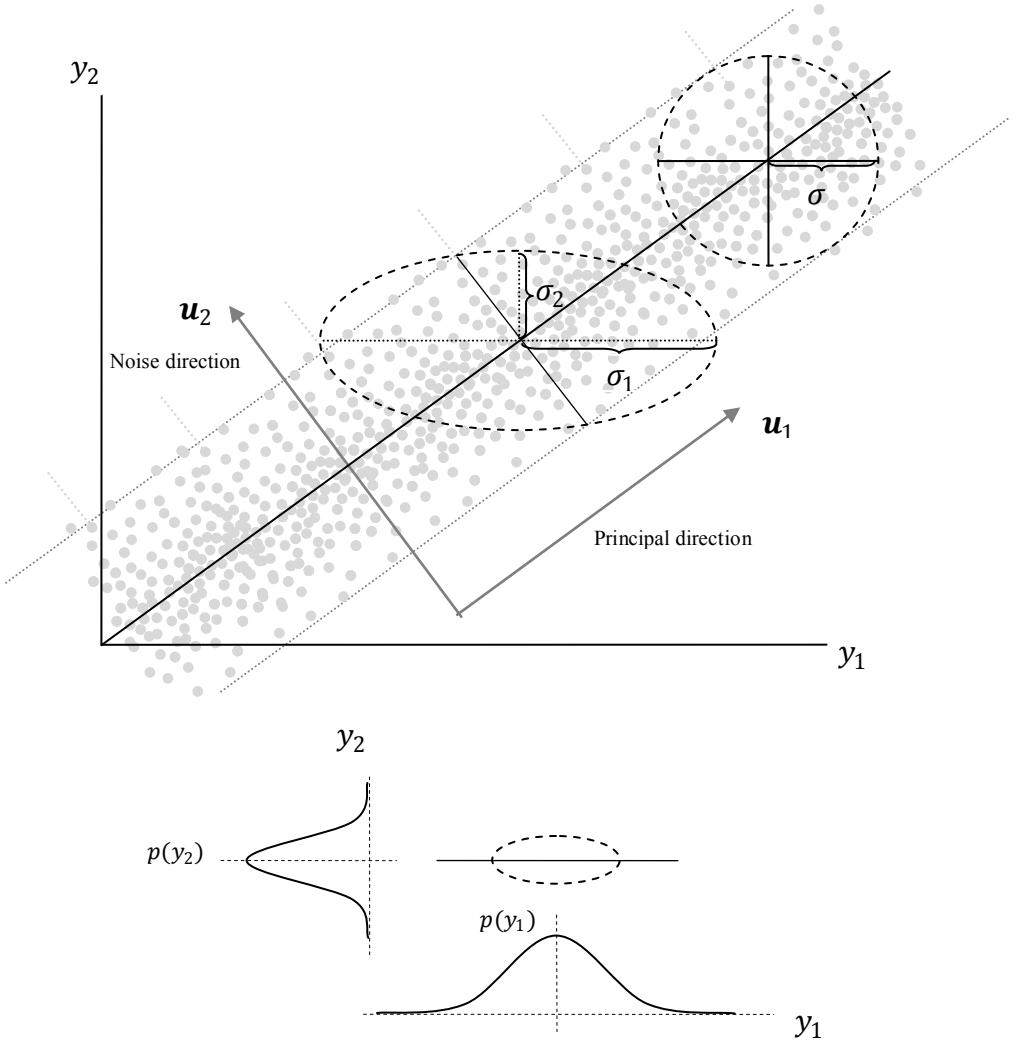


Figure 20 Comparison of estimated noise and real noise models in a two dimensional system.

Usually, in the problem of sensor fault diagnosis we have a good estimation of the noise in the system. This information may come from precision analysis of sensors provided by the sensor manufacturer, or though analysis of performance of sensors in the system. This

information about the noise is particularly valuable when we are dealing with nonlinear PCA methods or specifically Mixture of Probabilistic PCA.

Mixture of Probabilistic PCA

Mixture of PPCA is a nonlinear method of PCA which is based on separation of the input space into several local linear regions. Each region is defined by its own Gaussian model which is obtained by PPCA algorithm. Therefore, the whole space is defined by a Gaussian Mixture Model (GMM) which simulates the nonlinear behavior of the data. Assuming we have mixture of k Gaussian models with mean value and covariance of $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, and mixing coefficient of π_k for each model, Gaussian mixture distribution can be written as a linear combination of Gaussians in the form:

$$p(\boldsymbol{y}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{y}|k) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{y}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{112}$$

where probability of each component is a PPCA model which is associated with mean vector $\boldsymbol{\mu}_k$, projection matrix $\boldsymbol{P}_k$, and noise model σ_k^2.

The log-likelihood of this model over the observed data is:

$$\mathcal{L} = \sum_{i=1}^{m} ln \sum_{k=1}^{K} \pi_k p(\boldsymbol{y^i}|k). \tag{113}$$

Finding the maximum log-likelihood of this model is a typical problem of GMM which is solved by EM algorithm. In the E-step, we find the *responsibility* R_{ik} of component k for generating data point $\boldsymbol{y^i}$ by Bayesian Inference:

$$R_{ik} = P(k|\boldsymbol{y^i}) = \frac{p(\boldsymbol{y^i}|k)\pi_k}{p(\boldsymbol{y^i})} \tag{114}$$

In the M-step the parameters of each component is updated.

$$\pi_k^{<new>} = \frac{1}{N} \sum_{i=1}^{m} P^{<old>}(k|\boldsymbol{y^i}) \tag{115}$$

$$\boldsymbol{\mu}_k^{<new>} = \frac{\sum_{i=1}^{m} P^{<old>}(k|\boldsymbol{y^i})\boldsymbol{y^i}}{\sum_{i=1}^{m} P^{<old>}(k|\boldsymbol{y^i})} \tag{116}$$

Then the covariance structure is updated using the responsibility of the k^{th} component.

$$\boldsymbol{S}_k = \frac{1}{\pi_k^{<new>} m} \sum_{i=1}^{m} P^{<old>}(k|\boldsymbol{y}^i) \left(\boldsymbol{y}^i - \boldsymbol{\mu}^{<new>}\right)\left(\boldsymbol{y}^i - \boldsymbol{\mu}^{<new>}\right)^T \tag{117}$$

In practice, this algorithm is much faster when it is mixed with a separate clustering algorithm such as K-means clustering. Usually the early estimation of components of the mixture model is done using some iteration of K-means clustering algorithm then the EM algorithm is applied on the mixture model. A good advantage of this algorithm is its versatility to add more source of information to find a more accurate model of the data. This is very important when we want to apply in for sensor fault diagnosis. In the next part we explore what kind of information we can potentially have in the SFD problem and how to apply it on the algorithm to find an accurate model of the system.

Sensor Fault Diagnosis Using Mixture of Probabilistic PCA

The general approach in sensor fault diagnosis using MPPCA is to find an accurate mixture model from the input space. Then, during the online monitoring, once a new set of data is measured, we find the responsibility of each component with this new set. The component with the maximum responsibility is the winner and the parameters of the

winner component are used to test the validity of measurement using linear sensor fault diagnosis method.

The most important problem in this approach is how to model the input space that effectively represent the nonlinear correlation between different dimensions e.g. finding the optimum number of Gaussian models or clusters, K, the best location of center of each cluster $\boldsymbol{\mu}_k$, and its projection matrix $\boldsymbol{P}_k$. These parameters are dependent on training method and number of iteration used for the training.

As explained before, when we want to apply MPCCA for a measurement set, we might have additional information that can help to find a more accurate model of the system. Here we discuss these information and find out how they can improve on the algorithm of training. There are four type of information that we have about the data: (1) The covariance of the noise in each measurement is known (2) The covariance matrix of the noise is the same for all Gaussian models (3) There is no discontinuity in the space of measurement data (4) There is a continuous change in the direction of principal components. However, depending on the specific type of the fault diagnosis problem, some of these assumptions may not be valid.

One of the most important parameters in developing a MPPCA model is the number of Gaussian models. If the number of components is too low, there might not be enough linear behavior in each component and if the number of components is too high there is a chance to model the noise in the system. So it is very important to select the optimum number of Gaussian components. Figure 21 shows a simple data of two dimensions

which has been modeled by MPPCA with four different numbers of components. In this figure, the dashed ellipse is the one standard deviation contour of the noise covariance and solid lines shows the direction of principal components. Mixture models in parts a,b,c and d is composed of 3,5,7 and 15 components. graph (a) has too few components therfore the data inside each component has nonlinear behaviour. Overfitting proplem is also clearly shown in graph (d) when the number of components is too much. It clearly shows that without any information about the noise in the system, the model tries to model the noise in the data when we increase the number of components. This problem is known as over-fitting problem in the machine learning terminology. In this figure, the dashed ellipse is the one standard deviation contour of the noise covariance and solid lines shows the direction of principal components. Mixture models in parts a,b,c and d is composed of 3,5,7 and 15 components. As you can see model (a) has too few components therfore the data inside each component has nonlinear behaviour. Overfitting proplem is also clearly shown in model (d) when the number of components is too much.

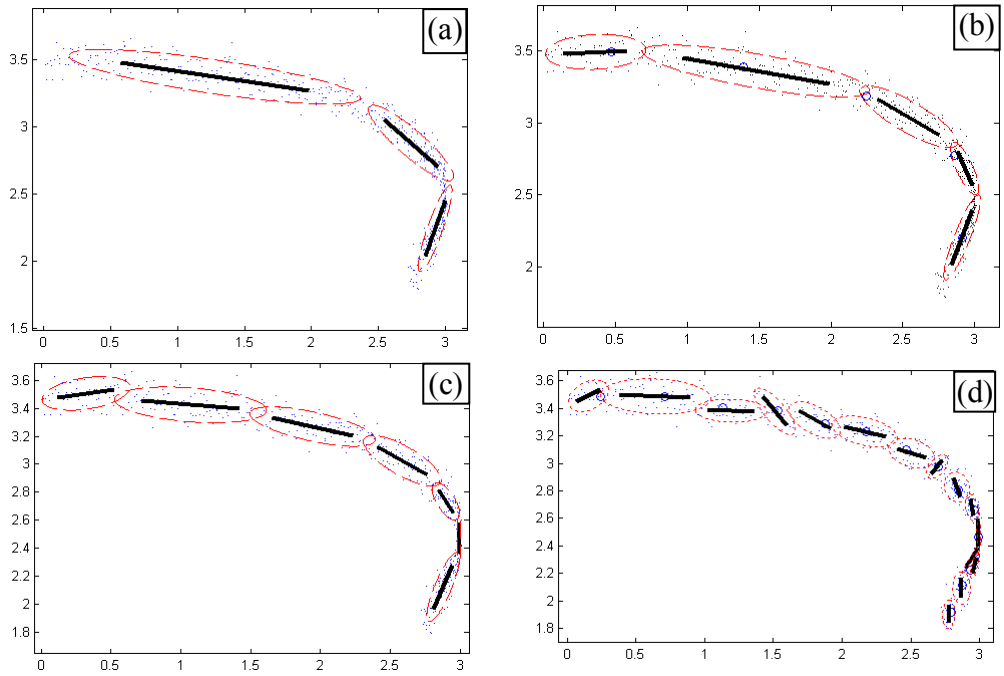


Figure 21 Traning a MPPCA model for a set of data with different number of Gaussian models

Now assume we have Gaussian distribution of noise, we want to see how we can use this information to model the noise. As explained in the PPCA section, we assume there is isotropic noise in the model, or the covariance of the noise is $\boldsymbol{\Phi} = \sigma^2 \boldsymbol{I}$. This means that the different sensors that is used in the measurement set has the same precision. This assumption is valid when the sensors of the system are identical e.g. there are several temperature sensors measuring the temperature of different parts of a plant and they are under the same working conditions. Therefore, the covariance of the noise is isotropic for all models and they are identical. The value of σ^2 is also found from characteristics of the sensor. On the other hand, the value of noise in the MPPCA is found through EM

algorithm. Therefore, we can compare the theoretical value of noise with the noise found from EM algorithm. These two values should be equal as long as we have an accurate model. However we will show that this is only the necessary condition for having a correct model.

In most cases the assumption of isotropic noise is not valid. Therefore, the noise found in MPPCA mode is an equivalent value of the noise which is related to the real noise trough Eq. (109). It can be seen that since the term $\boldsymbol{u}_j$ exist in this equation, the relationship between the experimental noise and general noise covariance matrix is also a function of principal components. This dependency is shown in Figure 22. In this figure the level of noise in y_2 is twice as big and the level of noise in y_1 therefore the modeled noise is dependent on the local slope or principal component of the model, however the covariance matrix of the noise is fixed at all points. The dashed ellipse is 3 standard deviation contour of the noise covariance and the dashed lines are 3 standard deviation contour of data variation. The dashed circle shows 3 standard deviation contour of the equivalent isotropic noise modeled by PPCA algorithm.

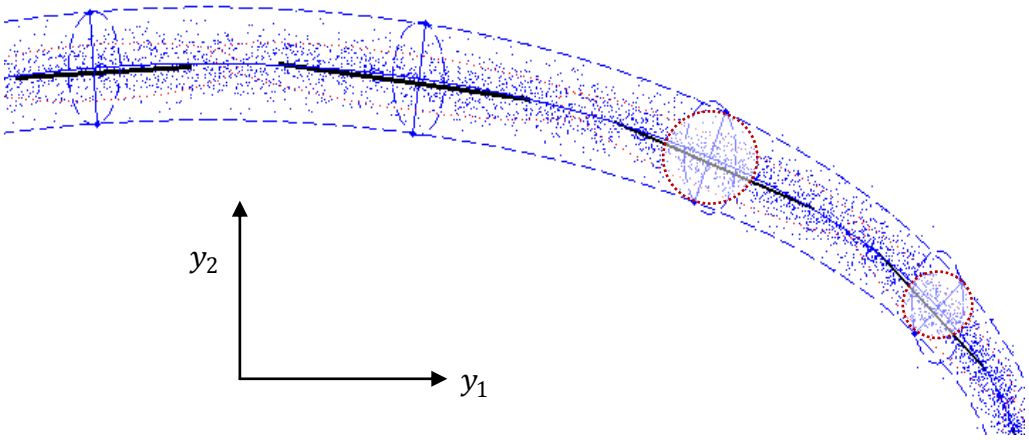


Figure 22 The relationship between the modeled noise and real noise

CHAPTER VII

CASE STUDIES AND CONLUTIONS

In this chapter, three sample project of sensor fault diagnosis has been explained. The first one is a linear analysis of measurements from model of a smart structure. The second system is the model of Tennessee Eastman Reactor. The probabilistic algorithm of sensor fault diagnosis has been applied on that. Finally, the nonlinear sensor fault diagnosis has been applied to the measurement from a HVAC system.

System 1: Linear SFD in a Model of a Smart Structure

In this section, a three-story building structure employing a MR damper is presented. The goal is to evaluate the performance of the proposed PCA methodology for the sensor fault detection of the semiactive nonlinear fuzzy control system. A typical example of a building structure employing a MR damper is depicted in Figure 23. Note, the MR damper can be applied to arbitrary locations within this building structure. The reason to install the MR damper into the 1st floor is that previous researchers have demonstrated the effectiveness of this approach. In addition, Figure 24 shows how a MR damper is implemented into a building structure.

The associated equation of motion is given by

$$\mathbf{M}^*\ddot{\mathbf{x}} + \mathbf{C}^*\dot{\mathbf{x}} + \mathbf{K}^*\mathbf{x} = \Gamma\mathbf{f}_{\mathrm{MR}}(t, x_1, \dot{x}_1, v_1) - \mathbf{M}^*\boldsymbol{\Lambda}\ddot{\boldsymbol{w}}_{\mathrm{g}} \tag{118}$$

where the system matrices are given by:

$$\mathbf{M}^* = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \tag{119}$$

is the mass matrix,

$$\mathbf{C}^* = \begin{bmatrix} c_1 + c_2 & -c_2 & 0 \\ c_2 & c_2 + c_3 & c_3 \\ 0 & -c_3 & c_3 \end{bmatrix} \tag{120}$$

is the damping matrix,

$$\mathbf{K}^* = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ k_2 & k_2 + k_3 & k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \tag{121}$$

is the stiffness matrix,

$$\mathbf{f}_{\mathrm{MR}}(t, x_1, \dot{x}_1, v_1) = \begin{bmatrix} f_{\mathrm{MR}}(t, x_1, \dot{x}_1, v_1) \\ 0 \\ 0 \end{bmatrix} \tag{122}$$

is MR damper force input matrix; $\ddot{w}_g$ denotes the ground acceleration; m_i are the mass of the i^{th} floor; k_i are the stiffness of the i^{th} floor columns; c_i are the damping of the i^{th} floor columns; the vector $\mathbf{x}$ and $\dot{\mathbf{x}}$ are the displacement and velocity relative to the ground; $\ddot{\mathbf{x}}$ is the absolute acceleration; $x_1$and $\dot{x}_1$ are the relative displacement and the relative velocity at the 1st floor level to the ground of the three story building structure, respectively; v is the voltage level to be applied; and Γ and $\mathbf{\Lambda}$ are location vectors of control forces and disturbance signal, respectively.

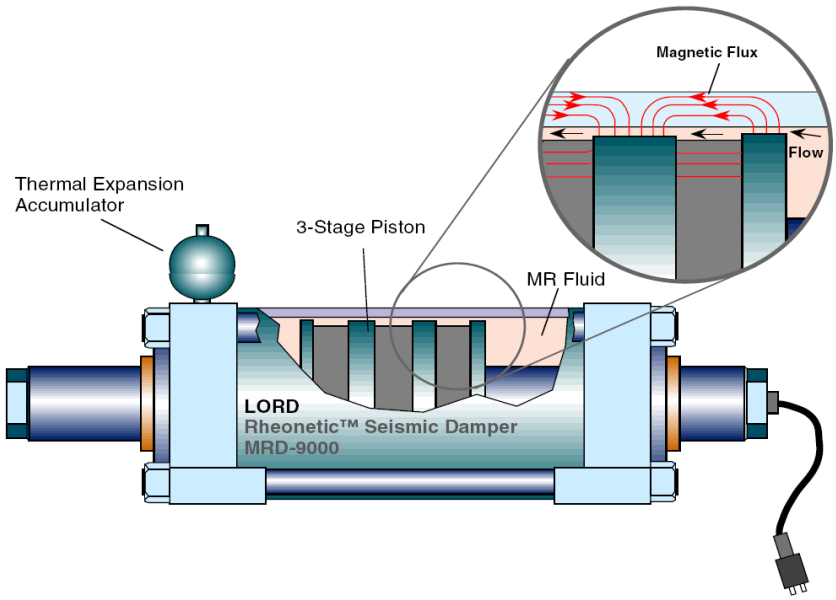


Figure 23 Schematic of the prototype 20-ton large-scale MR damper

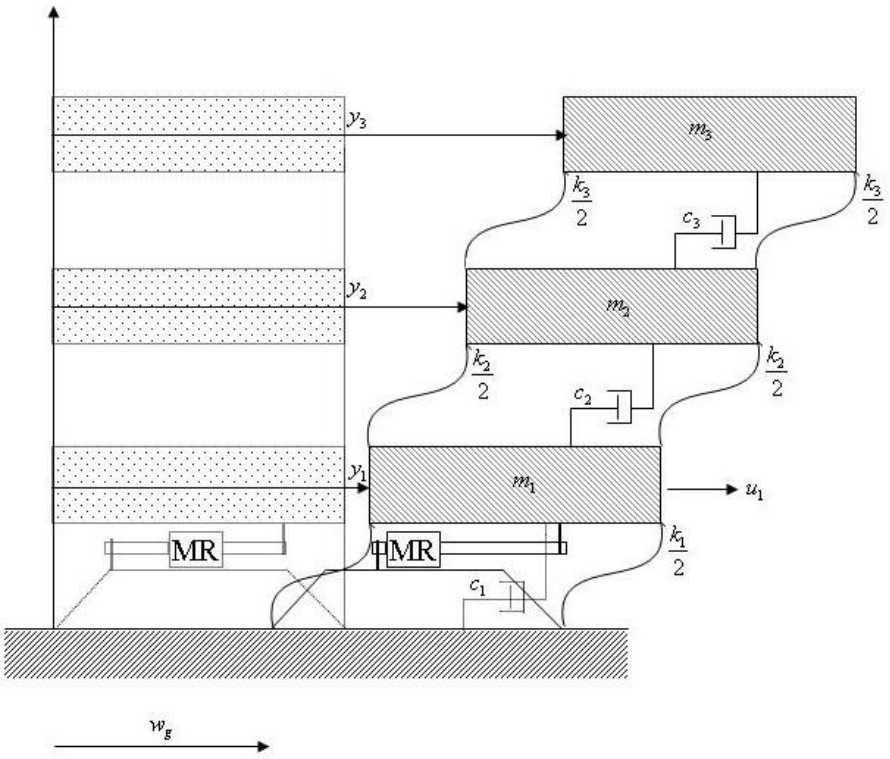


Figure 24 Integrated building structure-MR damper system

This second order differential equations can be converted into standard state space form

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{f}_{\mathrm{MR}}(t, \mathrm{z}_1, z_4, v) - \mathbf{E}\ddot{w}_{\mathrm{g}} \\ \boldsymbol{y} &= \mathbf{C}\mathbf{z} + \mathbf{D}\mathbf{f}_{\mathrm{MR}}(t, \mathrm{z}_1, z_4, v) + \mathbf{n} \end{aligned} \tag{123}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{*-1}\mathbf{K}^{*} & -\mathbf{M}^{*-1}\mathbf{C}^{*} \end{bmatrix} \tag{124}$$

is the state matrix,

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{*-1}\mathbf{F} \end{bmatrix} \tag{125}$$

is the input matrix,

$$\mathbf{C} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{*-1}\mathbf{K}^{*} & -\mathbf{M}^{*-1}\mathbf{C}^{*} \end{bmatrix} \tag{126}$$

is the output matrix,

$$\mathbf{D} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{M}^{*-1}\mathbf{F} \end{bmatrix} \tag{127}$$

is the feedthrough matrix,

$$\mathbf{E} = \begin{bmatrix} \mathbf{0} \\ \mathbf{F} \end{bmatrix} \tag{128}$$

is the disturbance location matrix,

$$\mathbf{F} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{bmatrix} \tag{129}$$

is the location matrix that a Chevron brace is located within a building structure, and z_1 and z_4 are the displacement and the velocity at the 1st floor level of the three story building structure, respectively. In this building structure, a SD-1000 MR damper has been applied.

Properties of the three-story building structure are adopted from a scaled model (Dyke et al. 1996) of a prototype building structure. The mass of each floor $m_1 = m_2 = m_3 = 98.3$ kg; the stiffness of each story $k_1 = 516{,}000$ N/m, $k_2 = 684{,}000$ N/m, and $k_3 = 684{,}000$ N/m; and damping coefficients of each floor $c_1 = 125$ Ns/m, $c_2 = 50$ Ns/m and $c_3 = 50$ Ns/m. In addition, a SD-1000 MR damper whose approximate capacity is 1,500 N is assumed to be installed into the 1st floor using a Chevron brace, which leads to a nonlinear dynamic model, i.e., a building-MR damper system. In what follows, the PCA technique is applied to the building-MR damper system and then, sensor faults within the smart structure are isolated and detected.

PCA Modeling

The first step for the sensor fault detection of the smart structures is to analyze the data to find out which sensor is isolable among the set of sensors: (1) the data used for analysis is taken from the simulation of the smart structure model. (2) the data obtained from each sensor is normalized so as to have a mean value of zero and standard deviation of one. The reason for this normalization is to give equal quantitative value to each sensor. (3) Then, the covariance matrix of the data is calculated and the eigenvalues of the covariance matrix is found. Figure 25 shows these eigenvalues in descending order.

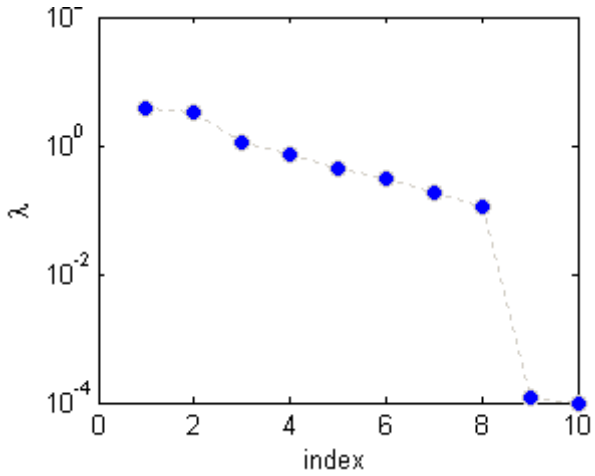


Figure 25 The Eigenvalues of the covariance matrix of measured data

As shown in Figure 25, the last two values of eigenvalues are considerably lower than the rest. It means that the dimension of noise space is 2 and the dimension of principal space is 8. In Other words, the measured data can be projected on 8 dimensions and the

projected data can be reconstructed back to the original space without losing any information. This is true as long as all of the measurements are uncorrupted or they follow the same correlation relationship. If the projection of the measured data into the noise space, i.e., residuals would be greater than a threshold, the faulty sensors of the system can be detected. The noise space is spanned by the eigenvectors corresponding to the minimum eigenvalues which is shown as $\boldsymbol{Q}$. In this problem, $\boldsymbol{Q}$ is found as:

$$\boldsymbol{Q} = \begin{bmatrix} 0.1449 & -0.7417 & 0.6322 & 0.0004 & 0.0025 & 0.0019 & -0.0020 & -0.1235 & 0.1181 & -0.0045 \\ 0.6985 & -0.3025 & -0.4801 & 0.0019 & -0.0017 & -0.0002 & 0.1535 & -0.0510 & -0.2244 & 0.3369 \end{bmatrix}$$

Each column of this matrix is the fault image vector for a sensor in the system. Since the dimension of residual space is 2, the fault image vectors in the residual space can be easily represented as shown in

From Figure 26, it can be judged if the sensor faults of the smart structure are detectable and isolable. (1) The line of fault image vector for sensors S_1 , S_3 and S_9 are different from the other sensors, which means these 3 sensors are both detectable and isolable. (2) Fault image vectors in the sensor S_2 is in the same line as the sensor S_8 and fault image vectors of sensor S_7 is in the same line as the sensor S_{10}, which means these sensors are detectable but not isolable. For instance, if a fault happens in sensor S_2, the abnormality can be detected but it is not possible to identify that the fault is due to error from either sensor S_2 or sensor S_8. (3) The fault image vectors for S_4, S_5 and S_6 are very close to zero. If a fault happens in any of these sensors, it does not show up in the residual space

since the vector of residual is equal to multiplication of fault image vector by the magnitude of the fault. Therefore, these three sensors are not even detectable. These results are summarized in Table 1.

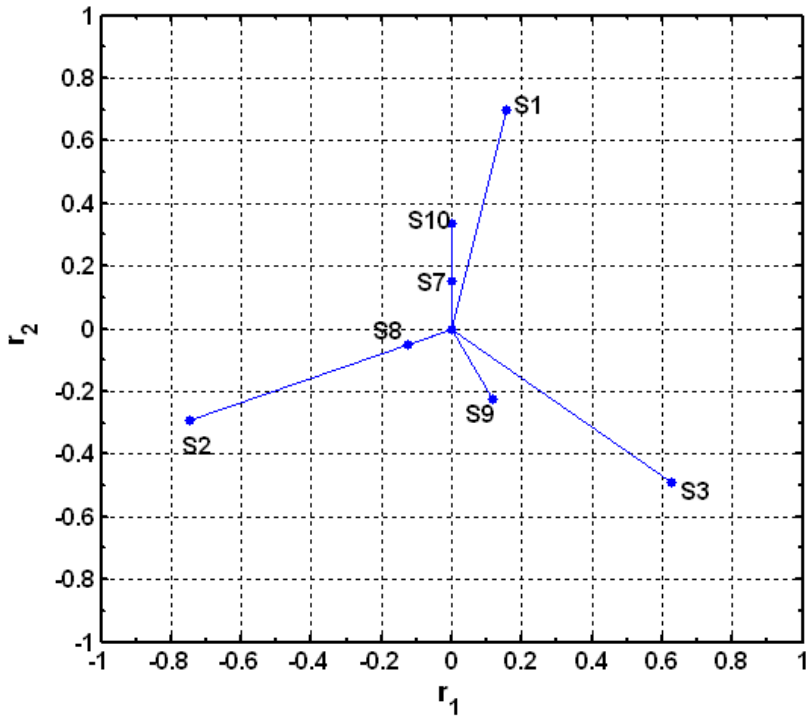


Figure 26 Sensor fault image vectors

Table 1 Detectability and isolability of sensors

Sensor #	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Detectable	✓	✓	✓	×	×	×	✓	✓	✓	✓
Isolable	✓	×	✓	×	×	×	×	×	✓	×

Linear Sensor Fault Diagnosis

As explained previously, the detection and isolation of sensor faults are identified using the parity equation. In the detection step, fault detection index expressed in terms of the residual vector and covariance matrix of the residual is calculated. The level of noise in the measurement is closely related to the performance of fault detection and the minimum level of detectable faults. In this analysis 3 different cases with different signal to noise ratios are considered. Table 2 shows these three cases and their corresponding $\boldsymbol{\Sigma_r}$ which is calculated by projecting of the noise covariance on residual space.

Table 2 Noise to signal ratio in different sensors (percent)

	Acceleration sensors	Displacement sensors	Other sensors	$\boldsymbol{\Sigma_r}$
Case 1	2	2	2	$\begin{bmatrix} 4.0e-4 & 0 \\ 0 & 4.0e-4 \end{bmatrix}$
Case 2	5	2	2	$\begin{bmatrix} 4.6e-4 & -0.416e-4 \\ -0.416e-4 & 5.6e-4 \end{bmatrix}$
Case 2	5	5	5	$\begin{bmatrix} 2.5e-3 & 0 \\ 0 & 2.3e-3 \end{bmatrix}$

In order to test the effectiveness of the algorithm, a shift sensor fault is generated in sensors in different times Figure 27 shows the pattern of the generated faults. The faulty region of data is shown with bolded lines. Note that in order to better compare the isolability and detectability of the sensors, the measured values have been normalized such that they all have a mean value of zero and a standard deviation of one. Then, a fixed error has added to the measurements. Since the capability of the sensors to detect and isolate the fault is also a direct function of magnitude of error, two different values of 0.3 and 1 is considered as the magnitude of error. The magnitude of error in Figure 27 is 1. The projection of these data onto the residual space is also shown in Figure 28.

The fault detection is performed via fault detection index introduced in Eq.(121) If there is no fault in the system, the index must follow a chi-square distribution. The index values are shown in Figure 29 for two different fault magnitudes. The top graph and bottom graph shows the index values when the magnitude of the fault is 1 and 0.3, respectively. In the left graph, these values are filtered to have a better distinction between the normal and faulty modes. It is clear from these graphs that: (1) the faults in sensors S_4, S_5 and S_6 are not detectable (2) even for the detectable sensor, there are different levels of sensitivity of the fault detection index. In other words, although some sensors are logically detectable, it is practically impossible to detect the fault because the magnitudes are too low, i.e., the sensitivity to the fault is too low.

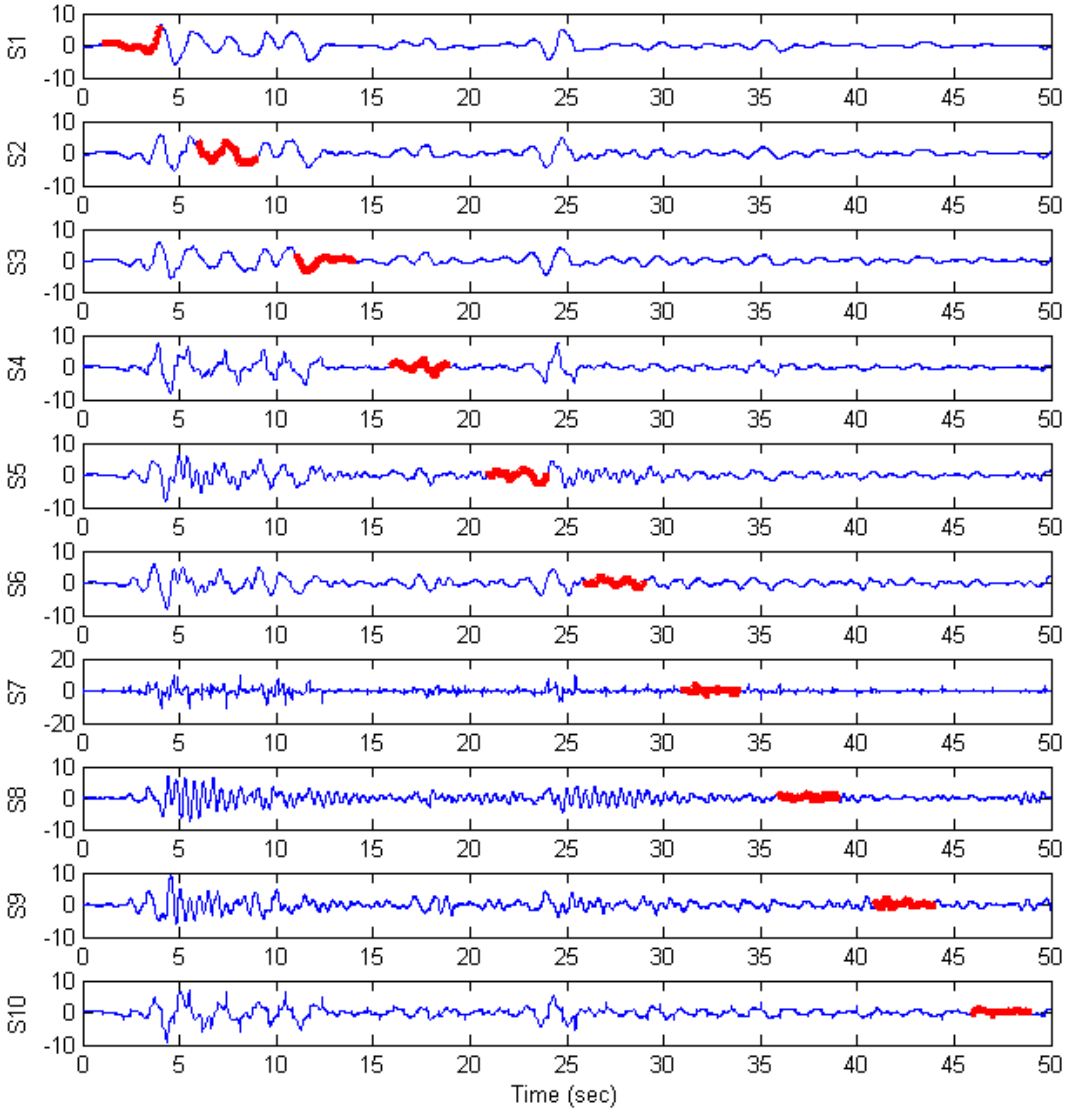


Figure 27 The scaled sensor data contaminate with noise and faults

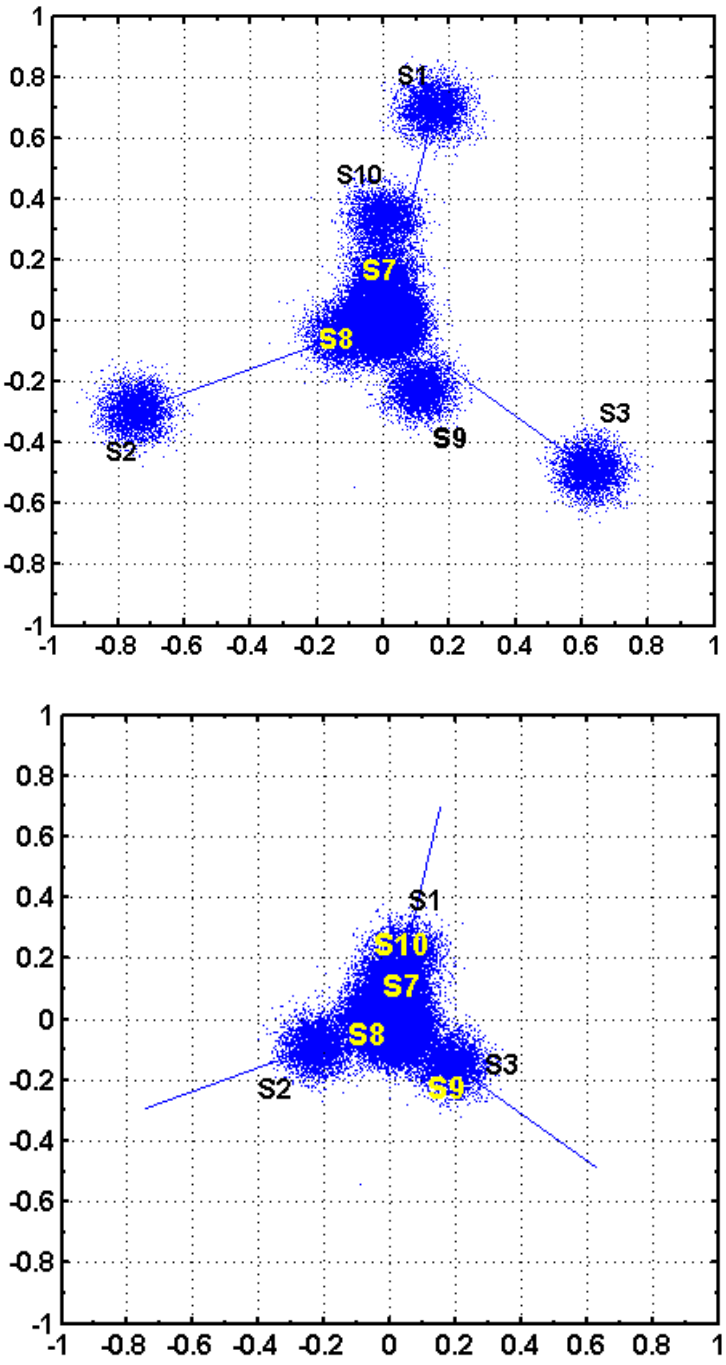


Figure 28 Projection of measurement into the residual space

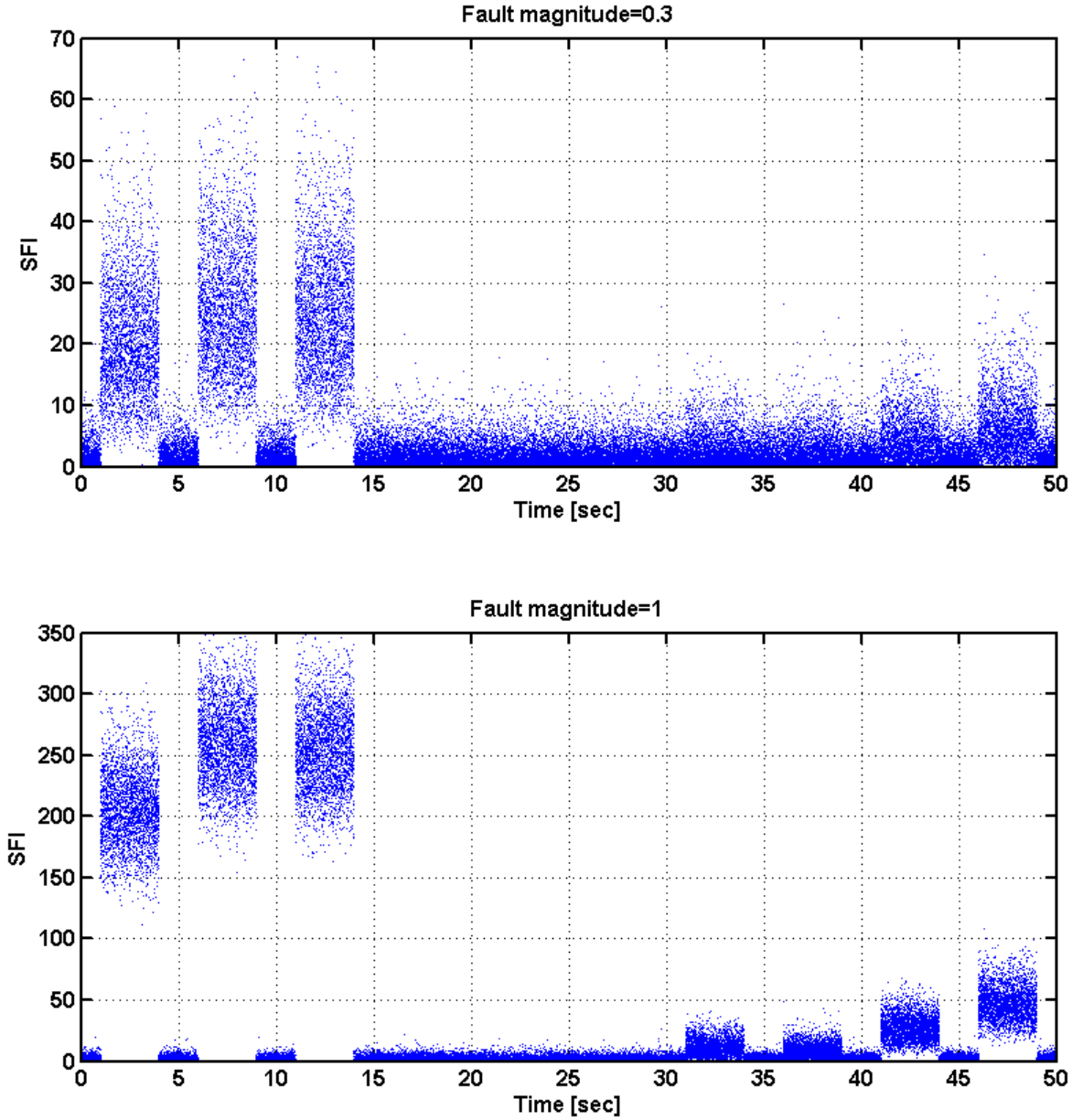


Figure 29 The value of fault detection index for different fault magnitudes

System 2: Probabilistic SFD for the Model of Tennessee Eastman Reactor

The proposed method of sensor fault diagnosis has been applied to the benchmark model presented by Downs and Vogel [Downs and Vogel 1993]. They have presented a model of an industrial chemical process for the purpose of developing, studding and evaluating process control technology. It consists of a reactor, a separator and recycle unit. This system was originally modeled in FORTRAN platform. Since then, the model has been evolved and now it is available as a SIMULINK model on web with several multivariable controllers designed for that. The whole plant has 12 valves for manipulation and 41 measurements available for monitoring or control. In order to compare the results of this study with a reference [Gertler et al. 1999], in this study we only consider the reactor unit which has 12 variables (Figure 30). These 12 variables are listed in Table 3.

These variables are reactor cooling water outlet temperature T_c , reactor cooling water flow V_c , and the fraction of each component of the input flow $X_A \ldots X_F$.

Of these variables, the cooling water flow is the manipulated input and the rest of them are measured values. The complete system is simulated with nine of the control loops operating with constant set points. The tree variables T, P and L are controlled with their setpoints varied. The model was run with 1330 different setpoints of T, P and L and the steady state values of them along with the values of F, T_c, V_c and X_A where recorded. Then, these variables were centered and normalized such that they have mean value of

zero and standard deviation of one. The eigenvalues of covariance matrix of these data are shown in Figure 31.

Table 3 Variables of TE reactor

Sensor #	*Variable*	*notation*	*Status*	*Equivalent in Downs &Vogel model*
$\boldsymbol{S_1}$	Rate of feed flow	F	Measured input	$XMEAS(6)$
$\boldsymbol{S_2}$	Reactor pressure	P	Measured output	$XMEAS(7)$
$\boldsymbol{S_3}$	Level of reactor	L	Measured output	$XMEAS(8)$
$\boldsymbol{S_4}$	Reactor temperature	T	Measured output	$XMEAS(9)$
$\boldsymbol{S_5}$	Reactor cooling water temperature	T_c	Measured output	$XMEAS(21)$
$\boldsymbol{S_6}$	Fraction of component A	X_A	Measured input	$XMEAS(23)$
$\boldsymbol{S_7}$	Reactor cooling water flow	V_c	Manipulated input	$XMV(10)$
Not used	Fraction of component $B \ldots F$	$X_B \ldots X_F$	Measured output	$XMEAS(24) \ldots XMEAS(24)$

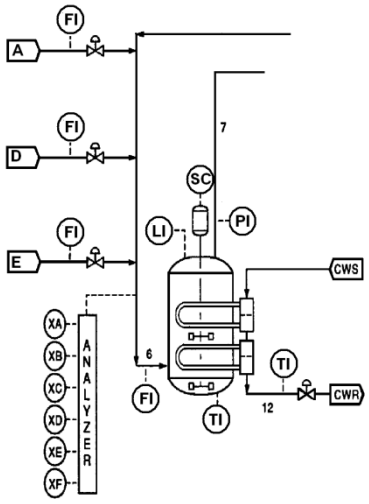


Figure 30 Tennessee Eastman reactor [Down and Vogel 1993]

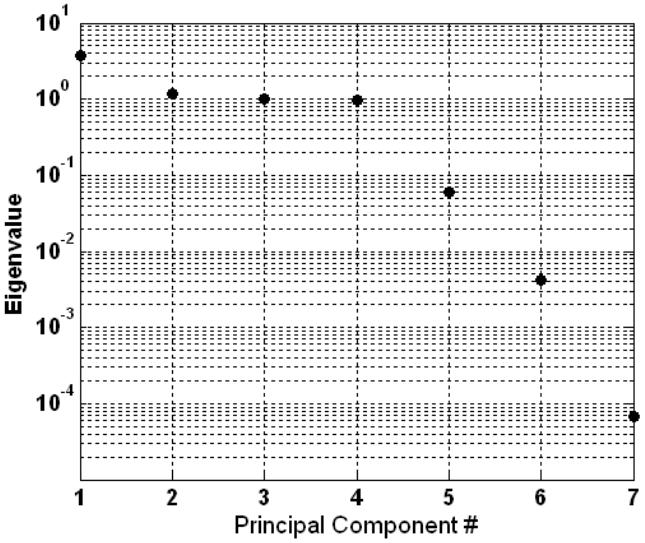


Figure 31 PCA results

Table 4 The directional analysis of covariance

	Residual Direction	Orthonormal direction	Covariance of the directional residual
	$\boldsymbol{q}_i = \boldsymbol{Q}(:, i)$	$\boldsymbol{Q}_i = \boldsymbol{Null}(\boldsymbol{q}_i^T)$	$\boldsymbol{\Sigma}^i = \boldsymbol{Q}_i \boldsymbol{\Sigma}_{\mathbf{r}} \boldsymbol{Q}_i{}^T$
$\boldsymbol{S_1}$	$\begin{bmatrix} -0.0007 \\ -0.0152 \\ 0.8157 \end{bmatrix}$	$\begin{bmatrix} -0.0187 & -0.9998 \\ 0.9997 & -0.0186 \\ -0.0186 & 0.0012 \end{bmatrix}$	$\begin{bmatrix} 0.0043 & -0.0001 \\ -0.0001 & 0.0001 \end{bmatrix}$
$\boldsymbol{S_2}$	$\begin{bmatrix} -0.0196 \\ -0.0808 \\ 0.4053 \end{bmatrix}$	$\begin{bmatrix} -0.1952 & 0.9796 \\ 0.9636 & 0.1825 \\ 0.1825 & 0.0839 \end{bmatrix}$	$\begin{bmatrix} 0.0060 & 0.0017 \\ 0.0017 & 0.0006 \end{bmatrix}$
$\boldsymbol{S_3}$	$\begin{bmatrix} -0.0004 \\ -0.0046 \\ 0.0034 \end{bmatrix}$	$\begin{bmatrix} -0.8015 & 0.5940 \\ 0.3990 & 0.4454 \\ 0.4454 & 0.6698 \end{bmatrix}$	$\begin{bmatrix} 0.0127 & 0.0187 \\ 0.0187 & 0.0280 \end{bmatrix}$
$\boldsymbol{S_4}$	$\begin{bmatrix} -0.7121 \\ -0.3990 \\ -0.2508 \end{bmatrix}$	$\begin{bmatrix} -0.4672 & -0.2937 \\ 0.8810 & -0.0748 \\ -0.0748 & 0.9530 \end{bmatrix}$	$\begin{bmatrix} 0.0037 & -0.0046 \\ -0.0046 & 0.0548 \end{bmatrix}$
$\boldsymbol{S_5}$	$\begin{bmatrix} 0.7017 \\ -0.4189 \\ -0.2408 \end{bmatrix}$	$\begin{bmatrix} 0.4917 & 0.2826 \\ 0.8674 & -0.0762 \\ -0.0762 & 0.9562 \end{bmatrix}$	$\begin{bmatrix} 0.0036 & -0.0047 \\ -0.0047 & 0.0552 \end{bmatrix}$
$\boldsymbol{S_6}$	$\begin{bmatrix} 0.0000 \\ -0.0010 \\ -0.0021 \end{bmatrix}$	$\begin{bmatrix} 0.4477 & 0.8942 \\ 0.8002 & -0.3991 \\ -0.3991 & 0.2029 \end{bmatrix}$	$\begin{bmatrix} 0.0124 & -0.0062 \\ -0.0062 & 0.0032 \end{bmatrix}$
$\boldsymbol{S_7}$	$\begin{bmatrix} -0.0101 \\ -0.8115 \\ 0.2225 \end{bmatrix}$	$\begin{bmatrix} -0.9643 & 0.2644 \\ 0.0811 & 0.2519 \\ 0.2519 & 0.9309 \end{bmatrix}$	$\begin{bmatrix} 0.0039 & 0.0142 \\ 0.0142 & 0.0526 \end{bmatrix}$

The first four principal components are dominant. Therefore, the last three components would be considered the noise component and their eigenvectors are the transformation into the noise space.

$$\boldsymbol{Q} = \begin{bmatrix} -0.0007 & -0.0196 & -0.0004 & -0.7121 & 0.7017 & 0.0000 & -0.0101 \\ -0.0152 & -0.0808 & -0.0046 & -0.3990 & -0.4189 & -0.0010 & -0.8115 \\ 0.8157 & 0.4053 & 0.0034 & -0.2508 & -0.2408 & -0.0021 & 0.2225 \end{bmatrix}$$

For the decision process, the covariance of the noise in the noise space is calculated as:

$$\boldsymbol{\Sigma_r} = \boldsymbol{Q\Sigma Q^T} = \begin{bmatrix} 6.83e-005 & 0 & 0 \\ 0 & 4.30e-3 & 0 \\ 0 & 0 & 6.04e-2 \end{bmatrix}$$

Then, the transformation matrix into the directional residuals and their corresponding covariance of noise are calculated as shown in Table 4.

Next, these using these distributions the diagnostic algorithm were modeled in a SIMULINK model. The block diagram of the SIMULINK model is shown in Figure 32.

In the SIMULINK model, the measured variables are first normalized, then the primary residual is found by pre-multiplication of residual by $\boldsymbol{Q}$ and the directional residuals are found by pre-multiplication of primary residual by $\boldsymbol{Q_i}$. Next, using the probability distribution of the primary and the directional residuals, the conditional probabilities of fault in each sensor $p(\boldsymbol{r}|\mathcal{S}_i)$ conditional and probability of having healthy measurement $p(\boldsymbol{r}|\mathcal{H})$ is found. In the last step, using the Bayes' formula, the probability of error in each sensor is calculated.

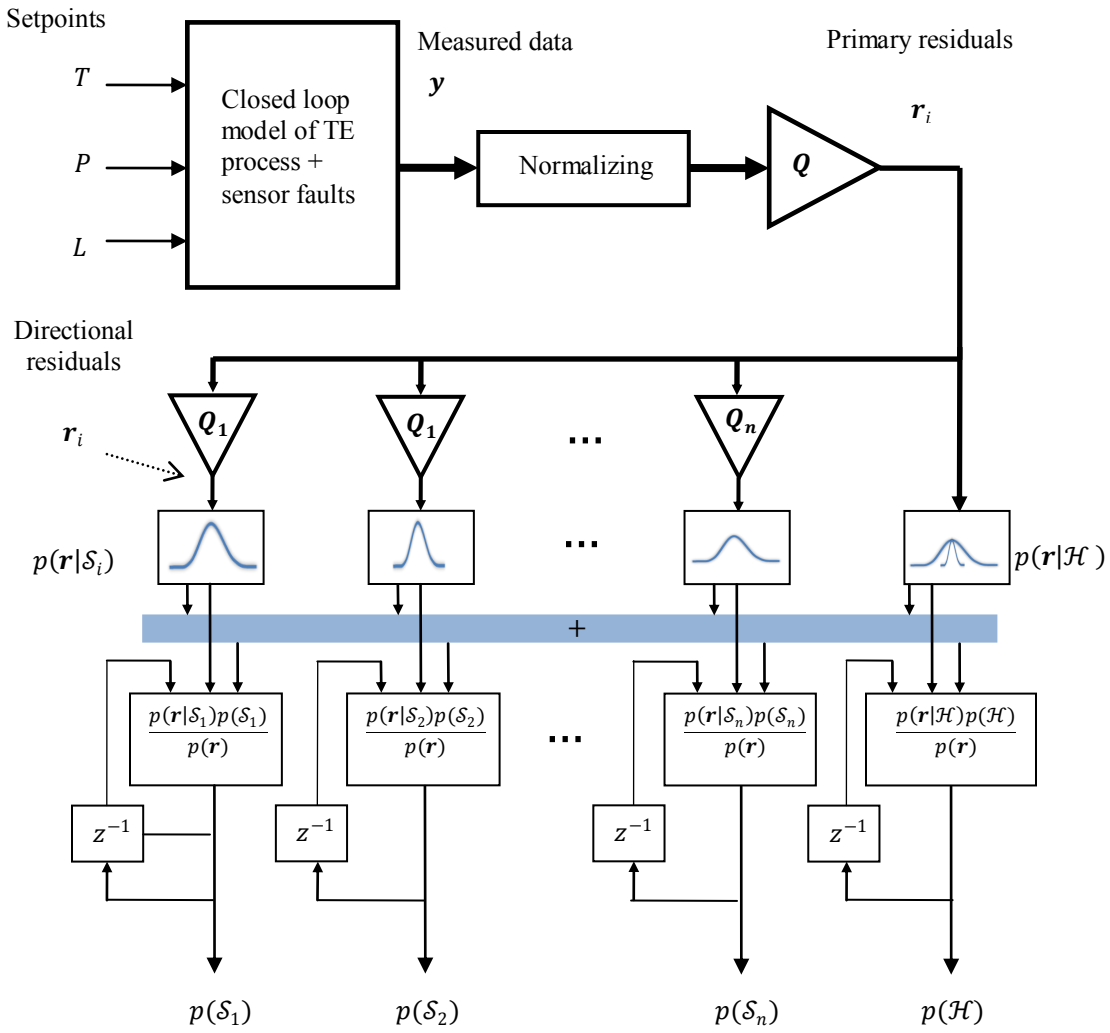


Figure 32 Diagnostic procedure modeled in SIMULNK

Before starting the simulation, the detectability index for each sensor was formed to find the level of detectability of each sensor in the system. These values are found and listed in Table 5.

The sensors in Table 5 are sorted based on their detectability. Based on this table, we can expect that the last two sensors $\boldsymbol{S}_3$ and $\boldsymbol{S}_6$ are not detectable these two sensors are

measuring the level of the reactor L and the fraction of component A in the entering flow X_A.

Table 5 Sensors sorted based on their detectability index

	S_4	S_5	S_7	S_1	S_2	S_3	S_6
Detectability index	0	0	0	1.47	3.42	483	486

The process was simulated for 70 hours of working and during these 70 hours the setpoint values of temperature pressure and level of reactor changed slowly. In order to test the capability of the algorithm, several artificial faults were added to the measurements of the process. In each 10 hours of operation a period of 8 hours of progressive amount of error, was added to one of measurements. Figure 33 shows the values of sensors contaminated with error.

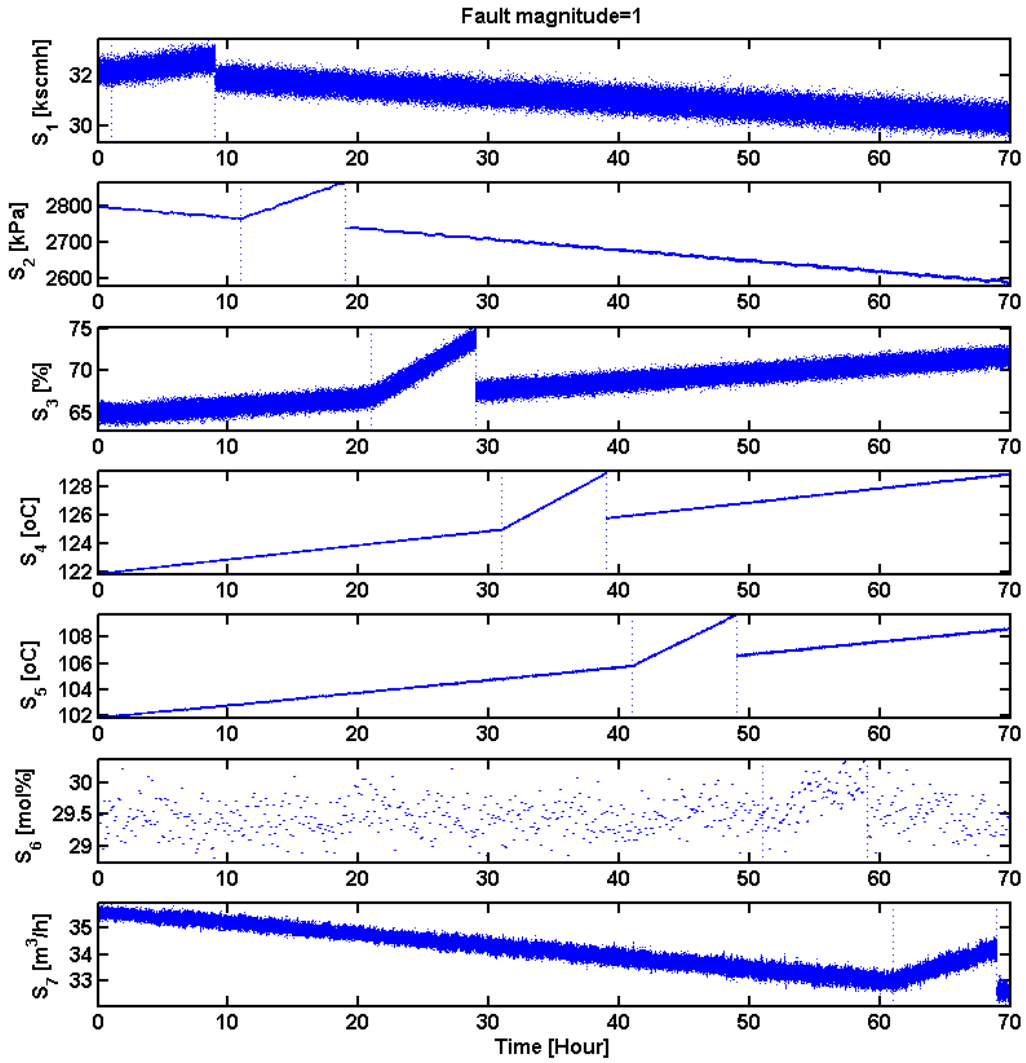


Figure 33 The measured values contaminated with error

These data then were centered and normalized and then using the primary residual and the directional residuals, the probability of error in each sensor were estimated. These probabilities are shown in Figure 34.

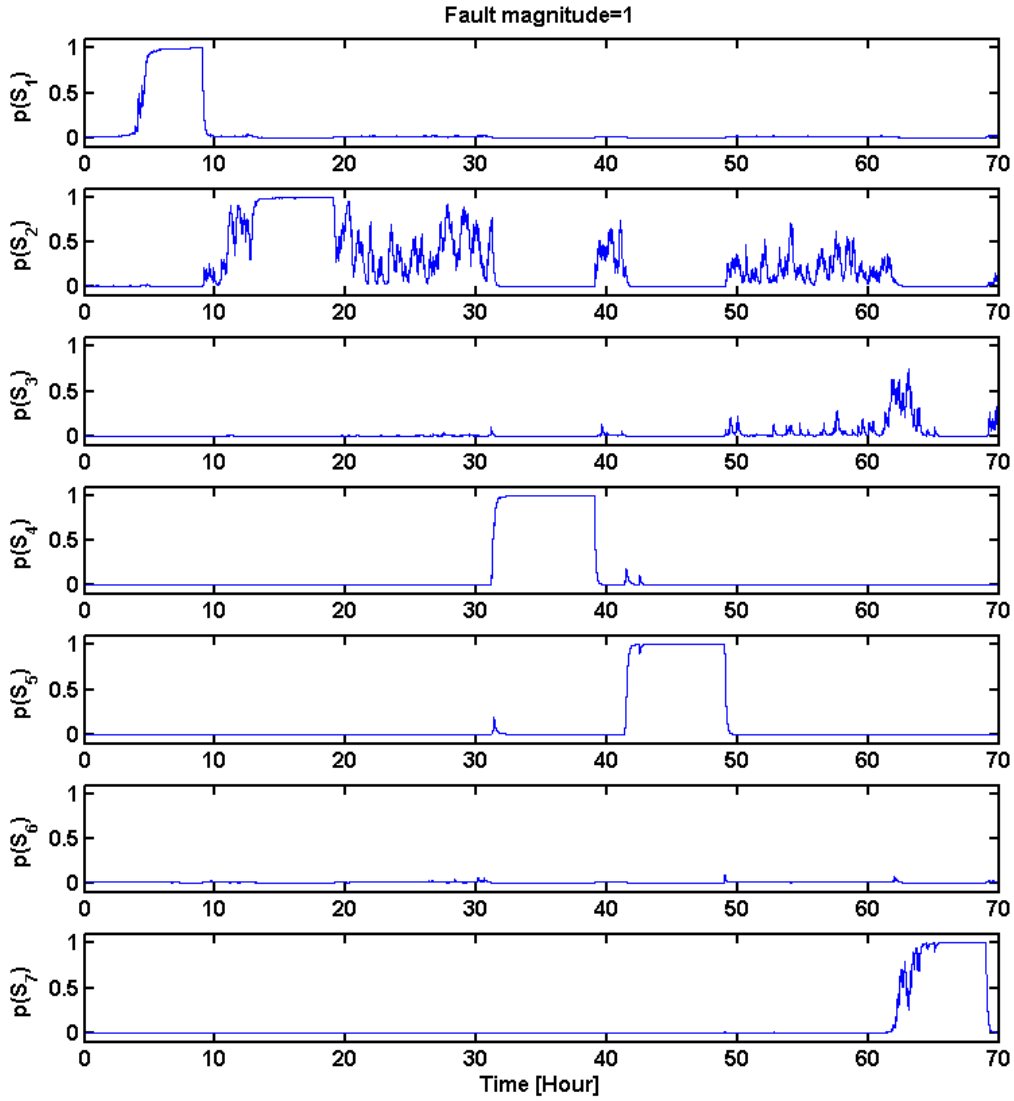


Figure 34 he estimated probability of error in each sensor

Figure 34 shows that error in sensor $\boldsymbol{S}_3$ and $\boldsymbol{S}_6$ has not been detected and this is in accordance with our detectability analysis. Also, it is clear that the faults in sensors $\boldsymbol{S}_4$ and $\boldsymbol{S}_5$ has been easily detected even with the small magnitude of error.

System 3: Nonlinear SFD on a Data From a Real HVAC System

A fully instrumented HVAC system has been studied in this part. The data was gathered from this system in four different working conditions describe in Table 7 and during each working condition the parameters of the system was recorded. The diagram of cooling system for the HVAC is shown in Figure 35. The measured parameters of the system are listed in Table 6.

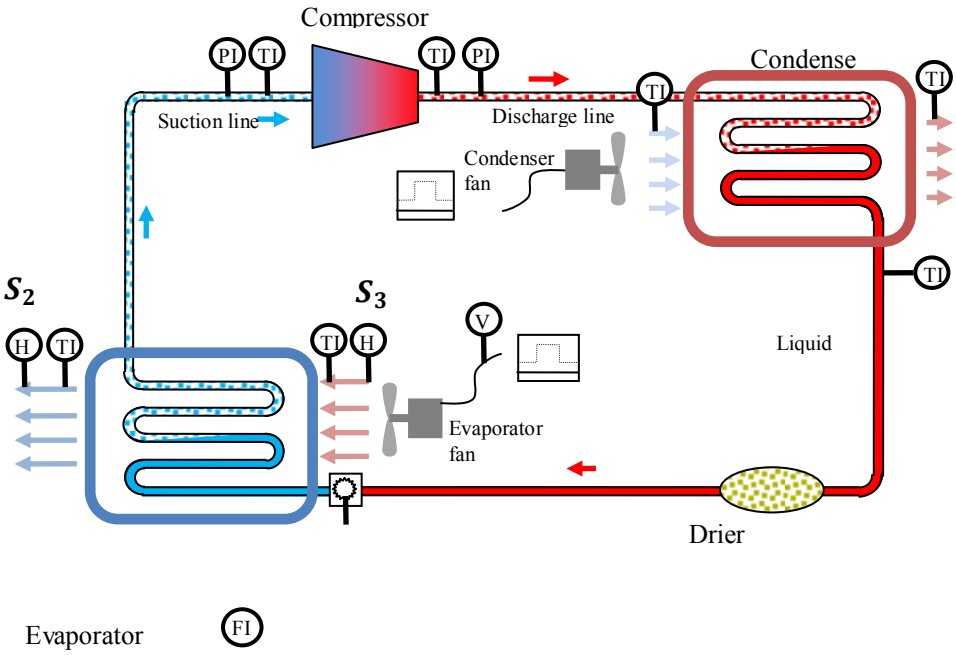


Figure 35 Diagram of the cooling system in HCAC

TI: Temperature Indicator HI: Humidity Indicator
PI: Pressure Indicator FI: Flow Indicator
VI: Voltage Indicator

Each test was run for 60 minutes. Each of the four conditions described above were tested for both a step change in the evaporator fan and a step change in the condenser fan. Aside from #2, all tests were run with the compressor at first stage. The default settings for startup were:

Evaporator Fan – 3V input signal (~13.3 m/s outlet air velocity)

Condenser Fan – 5V signal

Compressor at first stage (Except in #2)

Scenario #3 was simulated by partially (approximately half way) closing the shut off maintenance valve right after the condenser. Scenario #4 was simulated by blocking off half of the entry air path to the evaporator with cardboard boxes. The step changes are described below:

Evapfanstep – Step change in the evaporator fan introduced 20 minutes after startup, and brought back down 20 minutes later.

20 min – Default Settings

20 min – Evaporator Fan sent 10V Signal (~16.4 m/s outlet air velocity)

20 min – Default settings

Condfanstep – Step change in the condenser fan introduced 20 minutes after startup, and brought back down 20 minutes later.

20 min – Default Settings

20 min – Condenser Fan sent 10V Signal (~16.4 m/s outlet air velocity)

20 min – Default settings

Table 6 List of sensors used in the HVAC system

Sensor #	Description	Unit
S_1	Superheat	℃
S_2	Relative Humidity Air Handler Outlet	%
S_3	Relative Humidity Air Handler Inlet	%
S_4	Refrigerant Mass Flow	L/min
S_5	Refrigerant Pressure Compressor Inlet	psig
S_6	Refrigerant Pressure Compressor Outlet	psig
S_7	Power Consumption Outdoor Unit	Watts
S_8	Subcooling	℃
S_9	Condenser Air Outlet temp.	℃
S_{10}	Condenser Refrigerant Outlet	℃
S_{11}	Air Handler Air Outlet	℃
S_{12}	Compressor Refrigerant Inlet	℃
S_{13}	Compressor Refrigerant Outlet	℃
S_{14}	Condenser Air Inlet	℃
S_{15}	Air Handler Air Inlet	℃

Table 7 Explanation of working conditions

Index #	Working conditions
1	Compressor Stage 1- Evaporator Fan -medium, Condenser Fan – medium
2	Compressor at Stage 2. Evaporator Fan -medium, Condenser Fan – medium
3	Same as #1 expansion valve is partially blocked
4	Same as #1 evaporator is partially blocked

Figure 36-Figure **43** show the value of measured data from the sensors in different working conditions. For the training of the Mixture of Probabilistic PCA model, the data from different working conditions were mixed, then the data for each variable were separately normalized such that all variables has the mean value of zero and standard deviation of one. Different numbers of Gaussian components were used to model the entire data and the optimum number of components was estimated to be 10.

Algorithm of nonlinear sensor fault diagnosis was applied to these sensor data and the results in detail can be found in appendix C. Figure 44 shows the result of this algorithm when the magnitude of fault is different values of 0.3, 0.5, 0.7 and 1. The top graph in this figure shows the value of normalized data. The widened red regions are the faulty regions and the magnitude of errors is written behind them. The second graph shows the

estimated probability of error in that sensor. The third figure is the moving average value of the probability of error. The fourth graph specify whether the error is detected or not and the last graph shows whether the error is isolated or not. The values of last two graphs are binary of 0 and 1 where 0 is not detected or not isolated and 1 mean detected or isolated.

Notice that as the magnitude of error increases, the ability to detect and isolate that error is also increasing. However, when the magnitude of error is too big, the algorithm cannot detect or isolate the error. Higher detectability and isolability with bigger errors are predictable and is discussed in the linear method however when the magnitude of error is too big, a problem that happens in the nonlinear sensor fault diagnosis is that it will be difficult to detect that the measured data set belongs to which category. Therefore the misclassification error happens and although the error might be detected but it may not be able to isolate the source of error. In order to understand the effect of misclassification, Figure 45 has been produced. The top graph is this figure show the rate of classification error. The rest of graphs in this figure are the same as Figure 44 but under the assumption that the classification has been done correctly. As you seen for the lower magnitudes of error there is almost no classification error. However, as the magnitude of error increases, the rate of classification error also increases and for the magnitude of error of 1 it is almost always misclassified. Also notice that under the assumption of correct classification, the isolation is performed perfectly with high magnitudes of error.

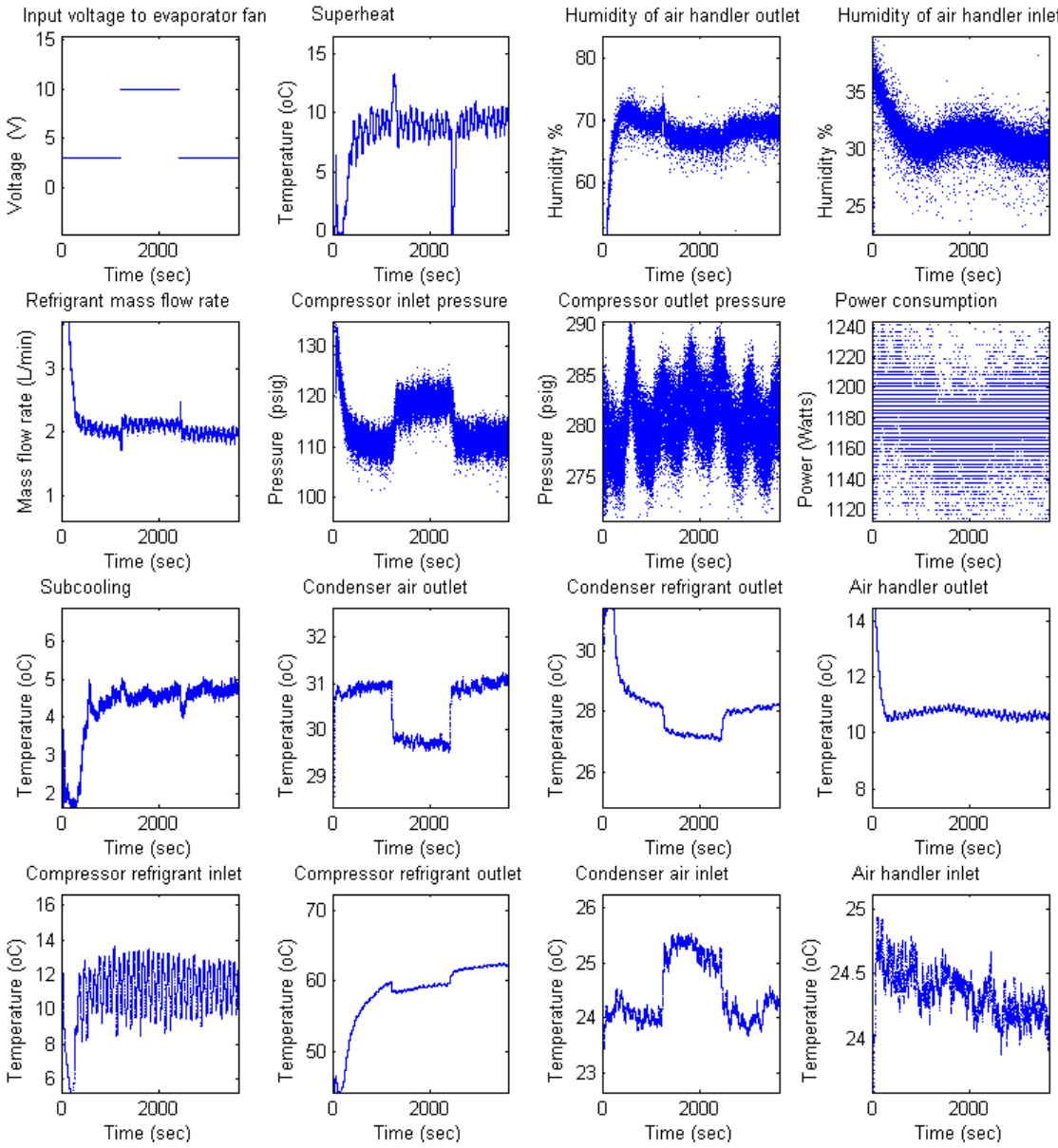


Figure 36 Sensor data for the Case # 1 and evaporator fan step

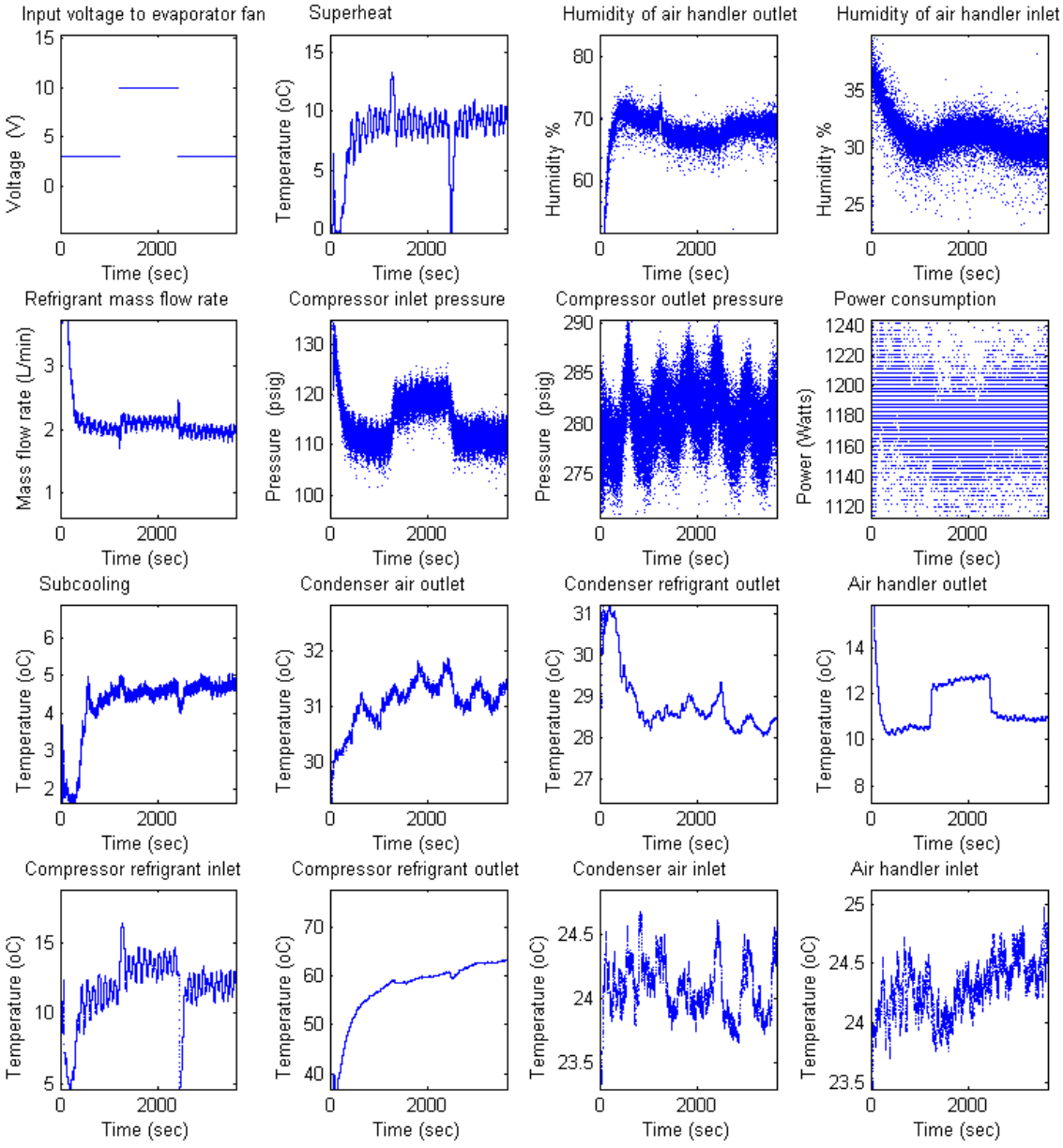


Figure 37 Sensor data for the Case # 1 and condenser fan step

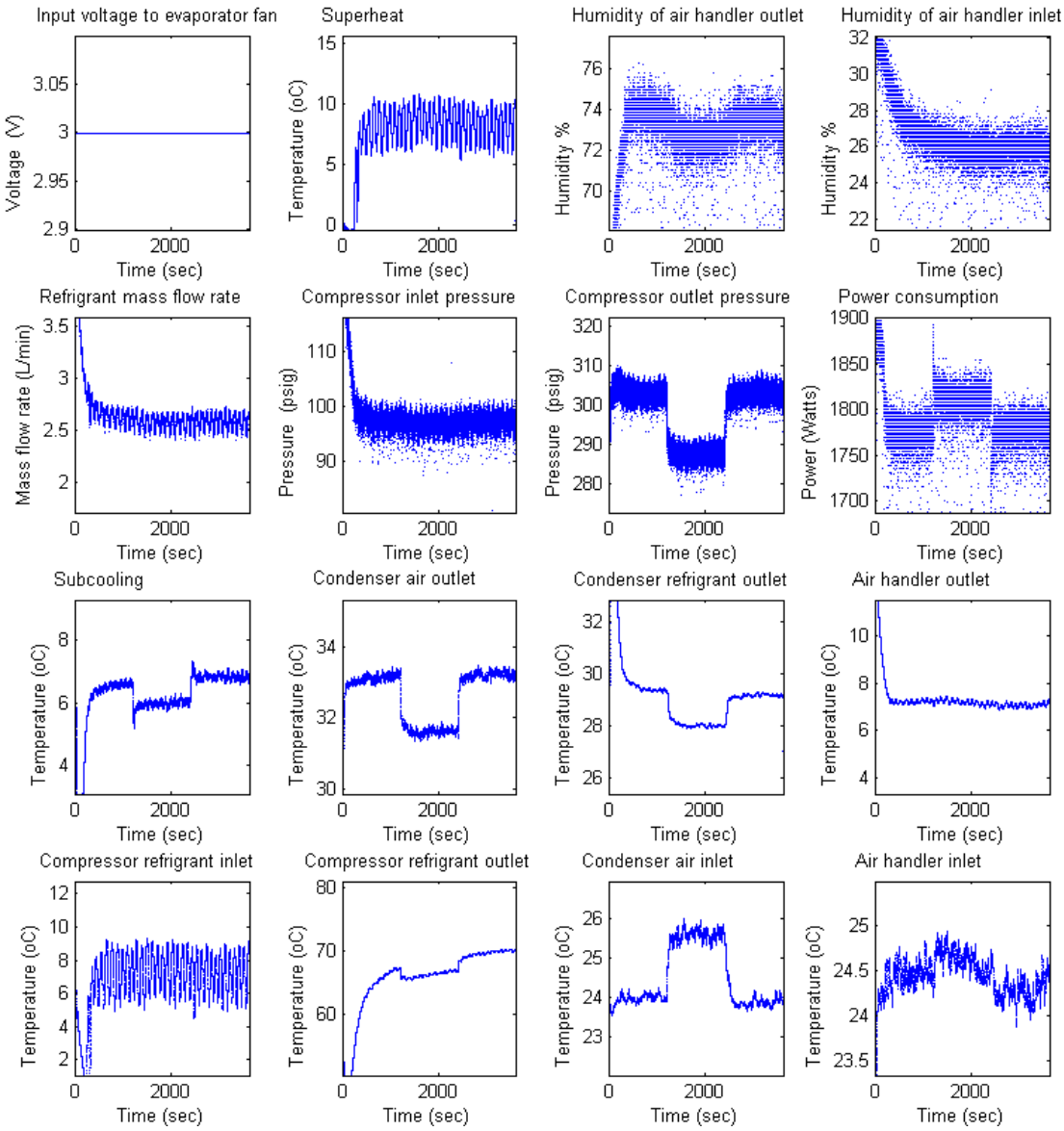


Figure 38 Sensor data for the Case # 2 and evaporator fan step

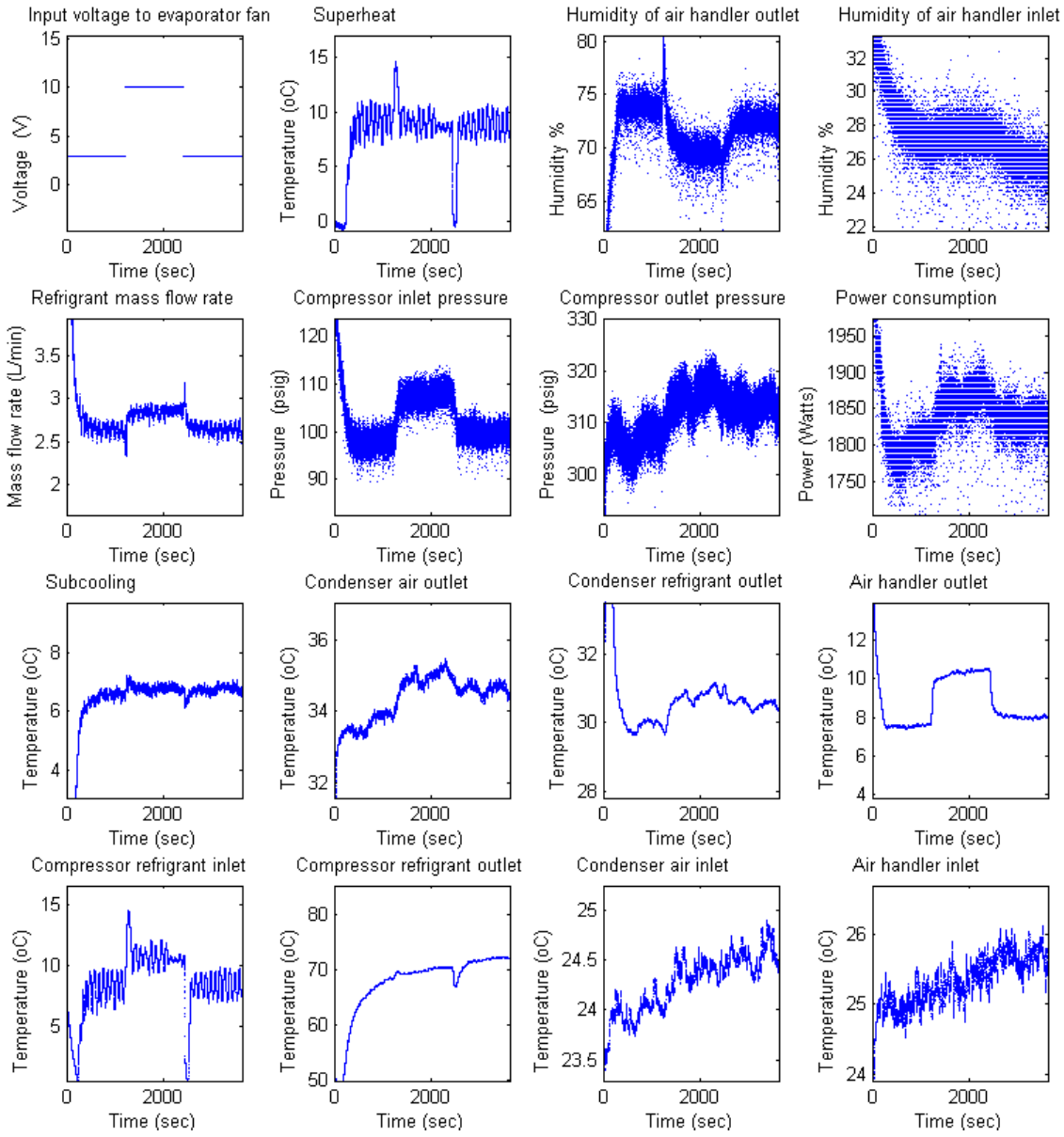


Figure 39 Sensor data for the Case # 2 and condenser fan step

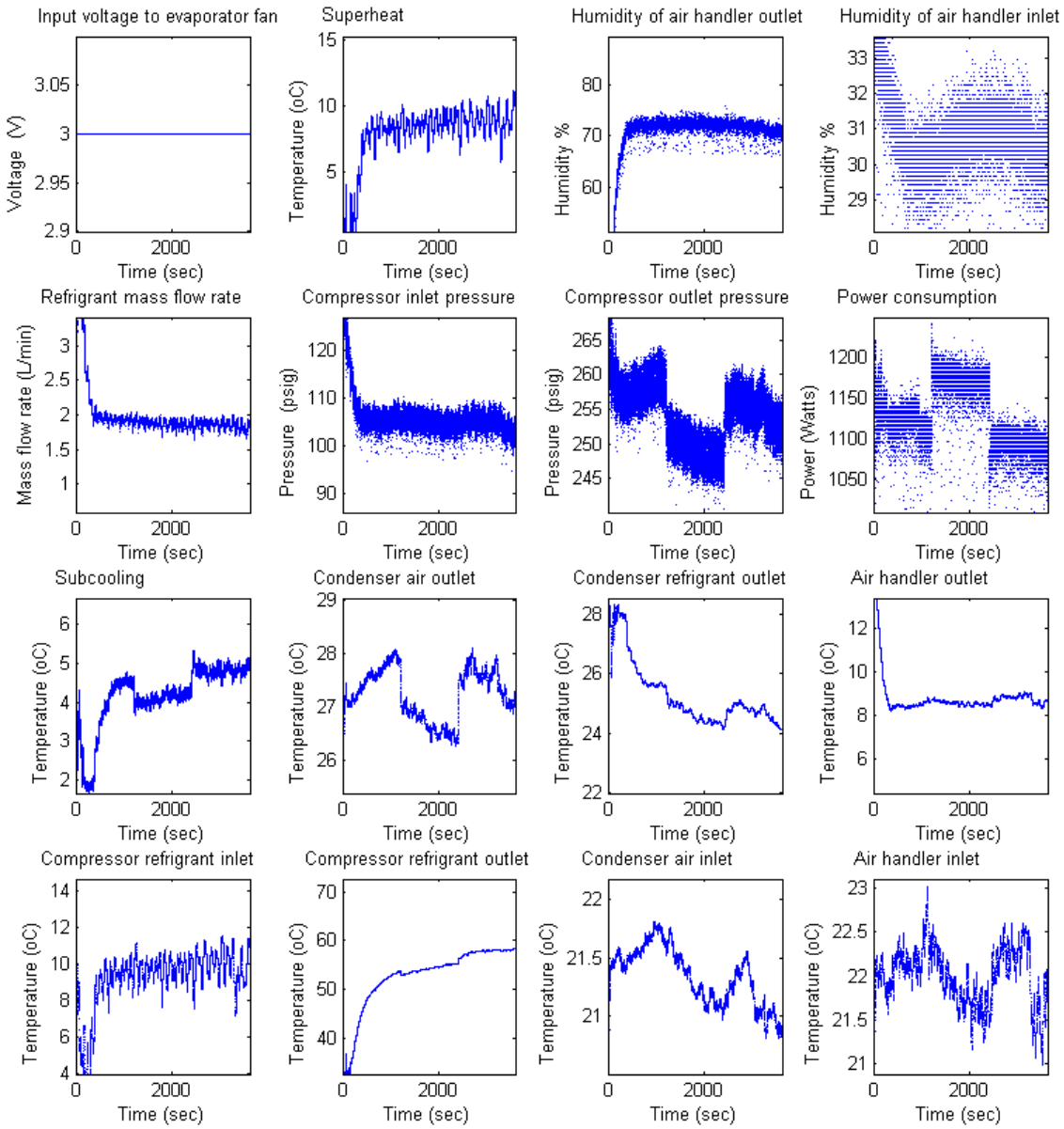


Figure 40 Sensor data for the Case # 3 and evaporator fan step

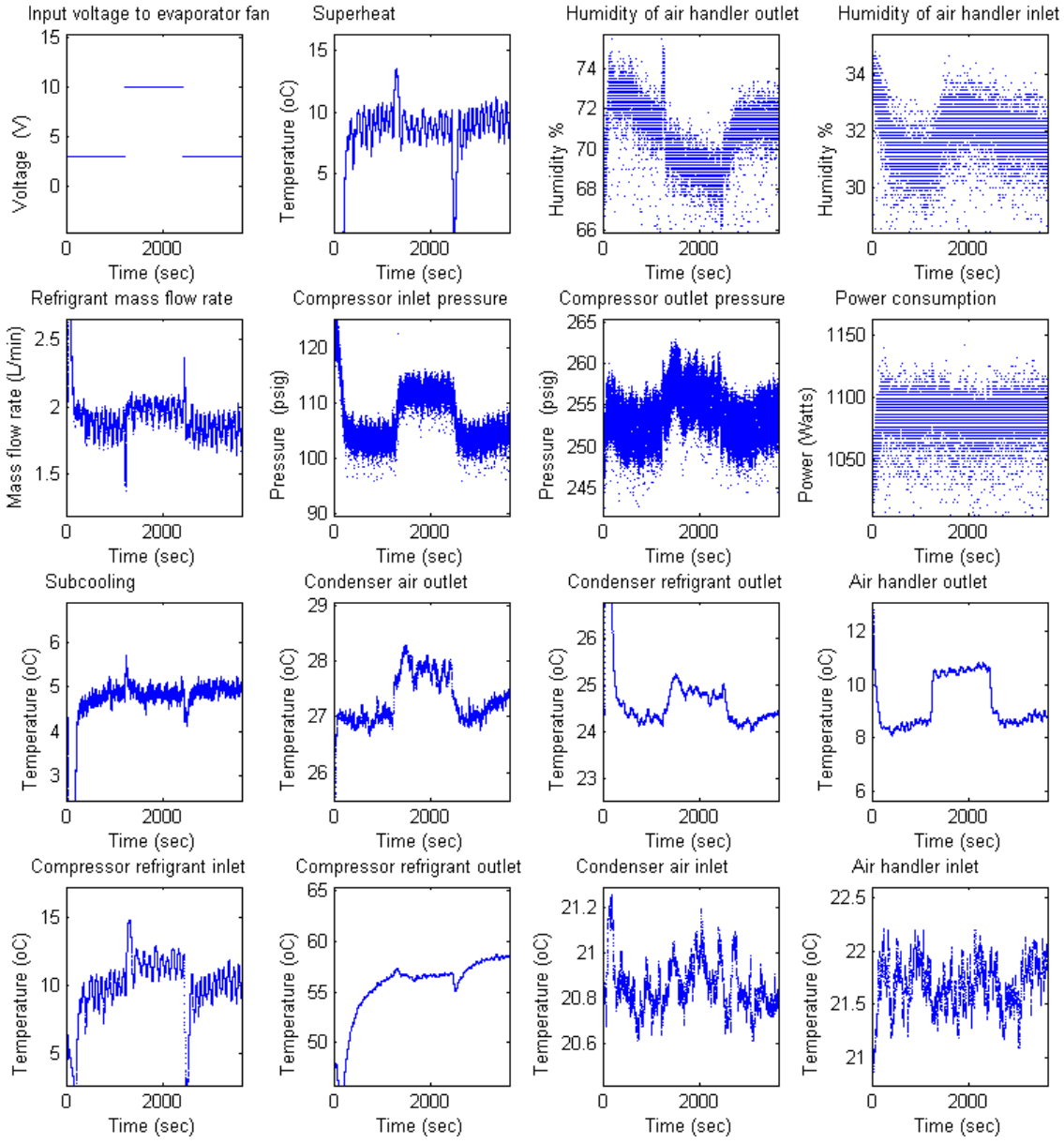


Figure 41 Sensor data for the Case # 3 and condenser fan step

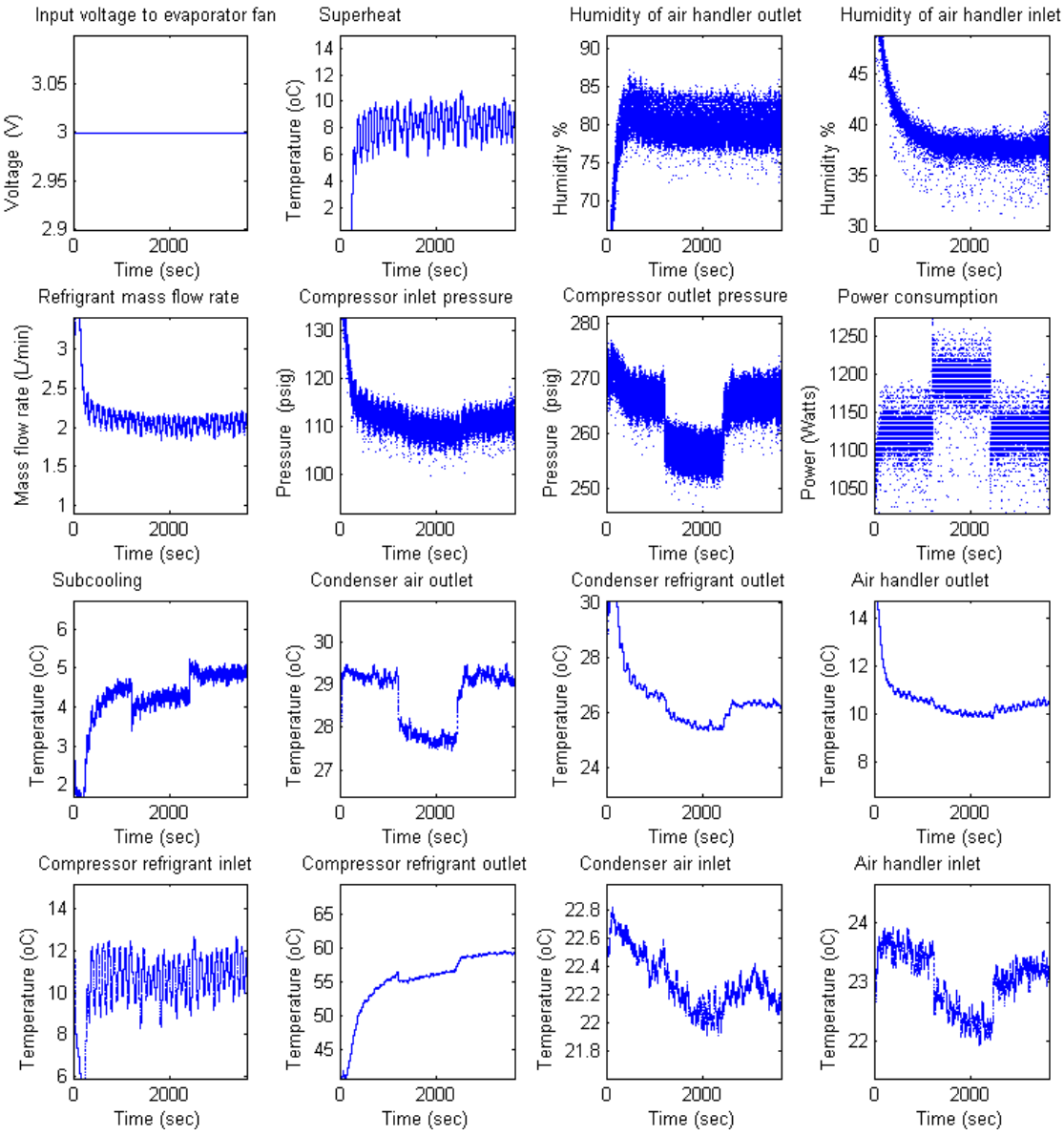


Figure 42 Sensor data for the Case # 4 and evaporator fan step

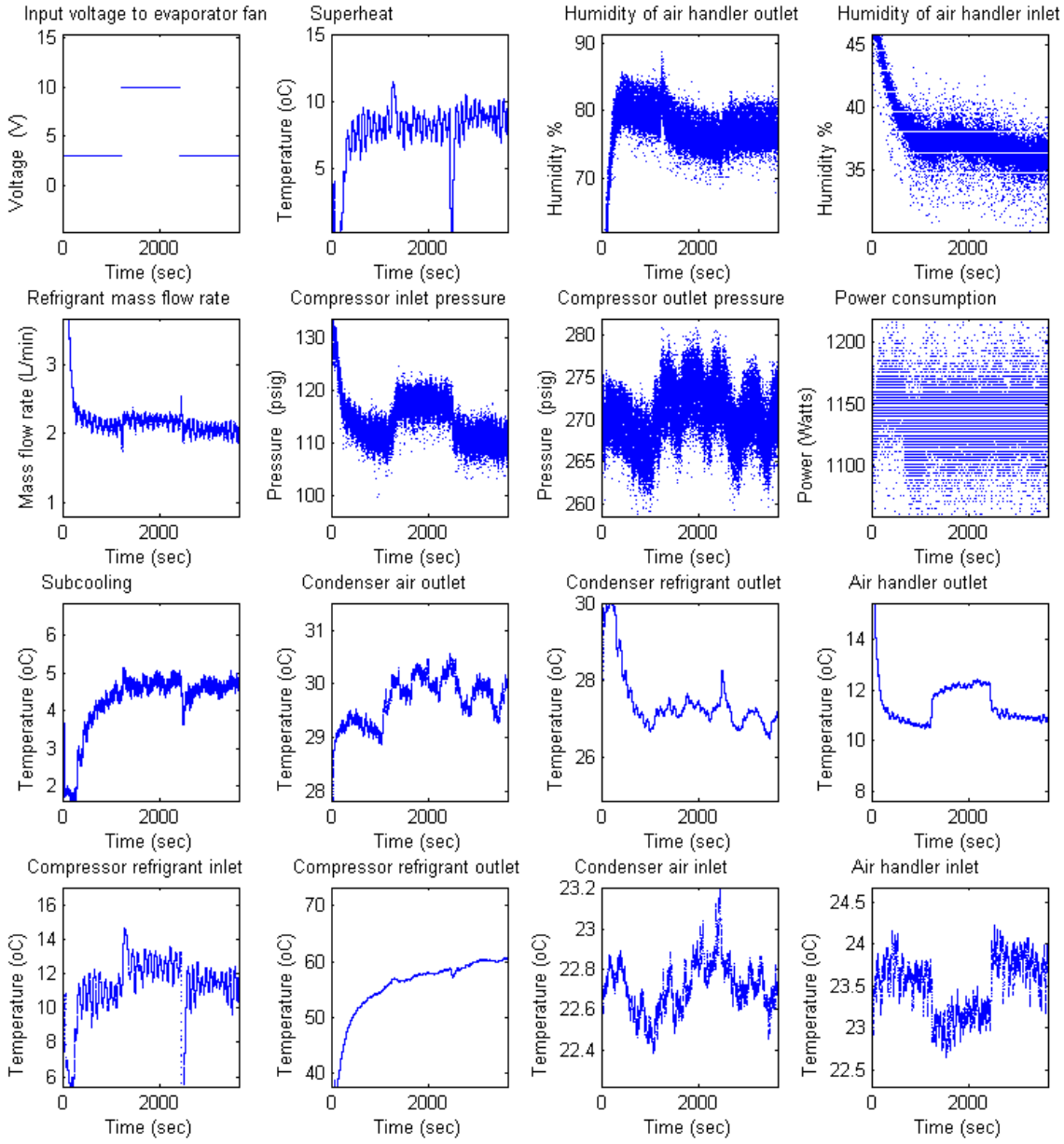


Figure 43 Sensor data for the Case # 4 and condenser fan step

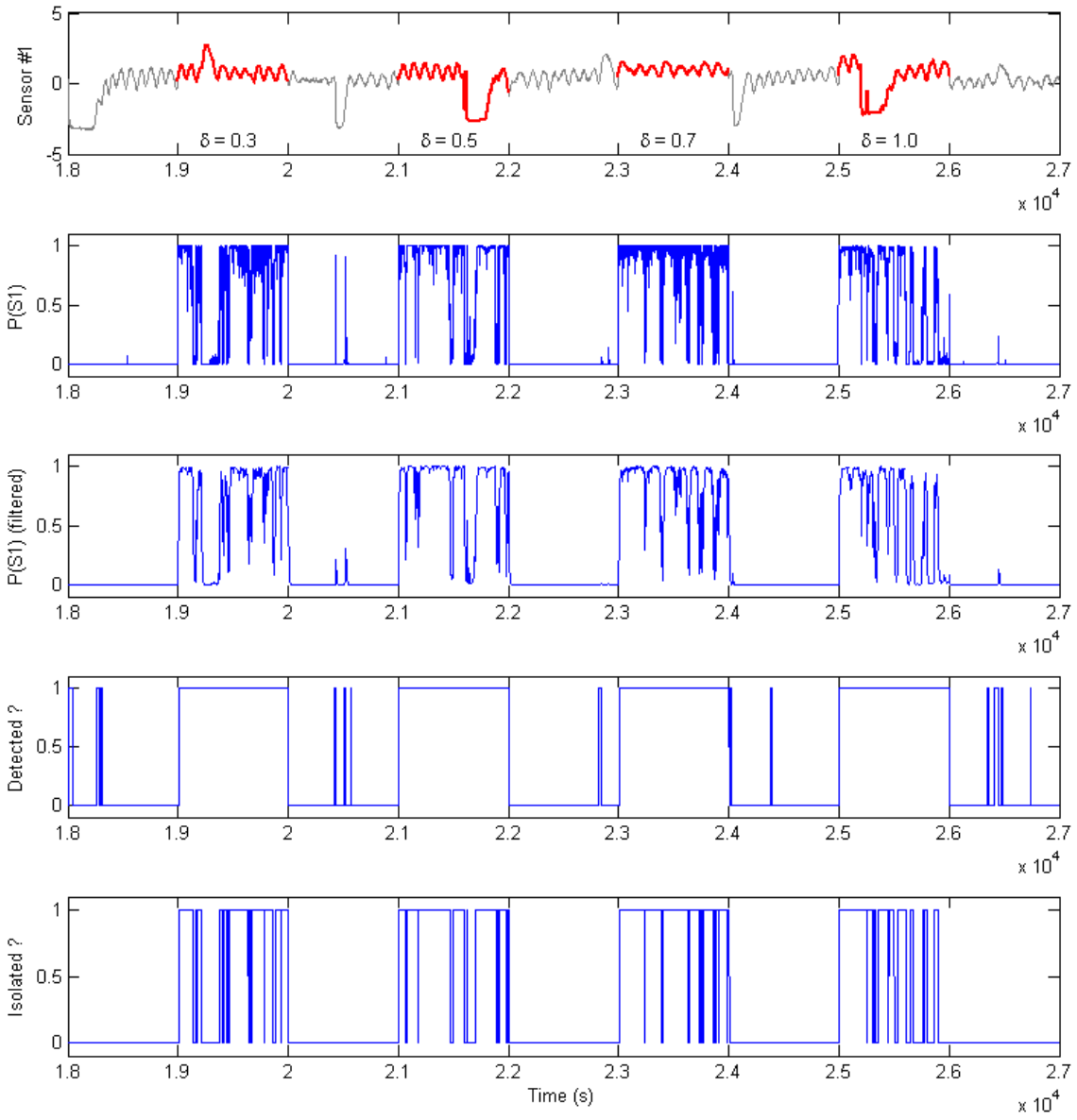


Figure 44 Results of nonlinear sensor fault detection

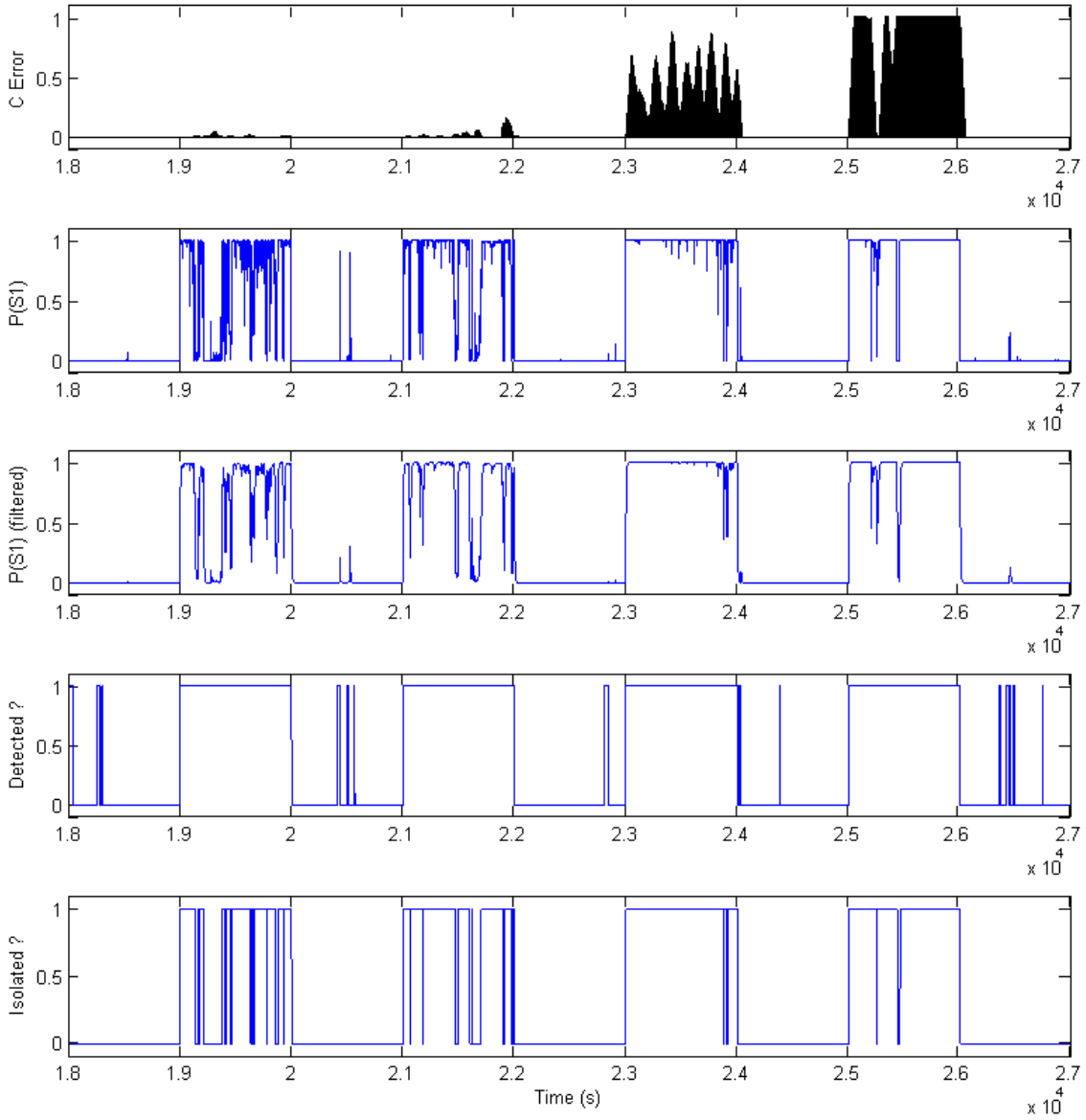


Figure 45 Analysis of detectability and isolability vs. classification error

LITERATURE CITED

Ben-Haim Y., "An Algorithm for Failure Location in a Complex Network," *Nuclear Sci. Engng*, **75**, 191 (1980).

Ben-Haim Y., "Malfunction Location in Linear Stochastic Systems-Application to Nuclear Power Plants," *Nuclear Sci. Engng*, **85**, 156 (1983).

Bermis S., S. Psarakis and J. Panaretos, "Multivariate Statistical Process Control Charts: An Overview," *Quality Reliability Engineering International*, **23**, 517 (2007).

Chiang L. H. and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*, Springer Verlag, London (2001).

Cho J.H, J.M. Lee, S. W. Choi and D. L. Lee, "Sensor Fault Identification Based on Kernel Principal Component Analysis," *Proceedings of the 2004 IEEE International Conference on Controls Applications*, **2**, 1223, 2-4 Sept. (2004).

Cho J.H., J.M. Lee, W. C. Sang, D. Lee and B. Lee, "Fault Identification for Process Monitoring Using Kernel Principal Component Analysis," *Chemical Engineering Science*, **60**(1), 279 (2005).

Chow E.Y. and A.S. Willsky, "Analytical Redundancy and the Design of Robust Failure Detection Systems," *IEEE Transactions on Auto. Ctrl.*, **AC-29**, 7 (1984).

Dorr R., F. Kratz, J. Ragot, F. Loisy, and J. Germain, "Detection, Isolation, and Identification of Sensor Faults in Nuclear Power Plants," *IEEE Transaction on Control Systems Technology*, **5**, 1 (1997).

Dunia R., S. J. Qin, T. F. Edgar and T. J. McAvoy, "Identification of Faulty Sensors Using Principal Component Analysis," *AIChE Journal*, **42**(10), 2797 (1996).

Duane L. M., C. J. Link, T.H. Guo, R. Graham and R. William, "Using Neural Networks for Sensor Validation," *AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Cleveland, OH, **98**, 3547, July 13-15 (1998).

Dinov I.D., *Expectation Maximization and Mixture Modeling Tutorial*, California Digital Library, Statistics Online Computational Resource, Paper EM_MM. http://repositories.cdlib.org/socr/EM_MM , December 9 (2008).

Downs J. J. and E.F. Vogel, "A Plant-wide Industrial Process Control Problem," *Computers Chem. Engng*., **17**(3), 245 (1993).

Frank P.M., "Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-Based Redundancy: A Survey and Some New Results," *Automatica*, **26**(3), 459 (1990).

Frank P.M. and X. Ding, "Survey of Robust Residual Generation and Evaluation Methods in Observer-Based Fault Detection Systems," *Journal of Process Control*, **7**(6), 403 (1997).

Gertler J., "Survey of Model-Based Failure Detection and Isolation in Complex Plants," *IEEE Control System Magazine*, **12**, 3 (1988).

Gertler J. and D. Singer, ''A New Structural Framework for Parity Equation Based Fault Detection and Isolation," *Automatica*, **26**, 381 (1990).

Gertler J., W. Li, Y. Huang and T. McAvoy, "Isolation Enhanced Principal Component Analysis," *AIChE J*, **45**, 2 (1999).

Gertler J. and J. Cao, "Design of Optimal Structured Residual from Partial Principal Component Models for Fault Diagnosis in Linear Systems," *Journal of Process Control*, **15**, 585 (2005).

Gertler J. and R. Monajemi, "Generating Directional Residuals with Dynamic Parity Relations," Automatica, **31**(4), 627,(1995).

Gross K.C., R.M. Singer, J.P. Herzog, R. VanAlstine and S.W. Waterish, "Application of a Model-Based Fault Detection System to Nuclear Plant Signals," *Proceedings of the 9th International Conference on Intelligent Systems Applications to Power Systems*, Seoul, Korea, 66 (1997).

Guo T.H and J. Musgrave, "Neural Network Based Sensor Validation for Reusable Rocket Engines," *Proceedings of the American Control Conference*, Seattle, Washington, 1367 (1995).

Guo T.H., J. Saus, C.F. Lin and J.H. Ge, "Sensor Validation for Turbofan Engine Using an Autoassociative Neural Network," *AIAA Guidance and Navigation and Control Conference*, San Diego, CA,331, July 29-31 (1996).

Harkat M.F., G. Mourot and J. Ragot, "Nonlinear PCA Combining Principal Curves and RBF Networks for Process Monitoring," *Proceedings of the 42nd IEEE Conference on Decision and Control*,221, Maui, Hawaii, December (2003).

Hastie T. J., *Principal Curves and Surfaces*, Ph.D. dissertation, Stanford Univ. (1984).

Hines J.W., R.E. Uhrig and D. J. Wrest, "Use of Autoassociative Neural Networks for Signal Validation," *Journal of Intelligent and Robotic Systems*, **21**(2), 143, February (1998).

Hines J.W. and E. Davis, "Lessons Learned from the U.S Nuclear Power Plant Online Monitoring Programs," *Progress in Nuclear Energy*, **46**(3),176 (2005).

Hines J.W., D.J. Wrest and R.E. Uhrig, "Plant-Wide Sensor Calibration Monitoring," *Proceedings of the 1996 International Symposium on Intelligent Control*, Dearborn, MI September 15-18, 3212 (1996).

Hoffman J. and K. Kimble, "Development of Real Time Engine Diagnostic Tools at the Arnold Engineering Development Center Engine Test Facility," *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, 473, 10-13 July (2005).

Ibarguengoytia P. H., S. Vadera and L. E. Sucar, "A Probabilistic Model for Information and Sensor Validation," *The Computer Journal*, **49**, 1, (2006).

Isermann R. "Supervision, Fault Detection and Fault Diagnosis Methods- An Introduction," *Control Engineering Practice*, **5**(5), 639, (1997).

Isserman R., "Model Based Fault Detection and Diagnosis Methods," *Proceedings of the American Control Conference*, Seattle Washington, 78, June (2005).

Kramer M. A., "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks," *AIChE J.*, **37**, 223 (1991).

Kramer M.A., "Autoassociative Neural Networks," *Computers & Chemical Engineering*, **16**(4), 313, (1992).

Kresta J. V., J. F. MacGregor and T. E. Marlin, "Multivariate Statistical Monitoring of Process Operating Performance," *The Canadian Journal of Chemical Engineering* , **69**, 35 (1991).

Lampis M. and J.D. Andrews, 2009 "Bayesian Belief Networks for System Fault Diagnostics," *Qual. Reliab. Engng. Int.*, **25**, 209 (2009).

Lawrence N.D., "Probabilistic Non-Linear Principal Component Analysis with Gaussian Process Latent Variable Models," *Journal of Machine Learning Research*, **16**, 44, (2005.)

MacGregor J. F., T. E. Marlin, J. Kresta and Skagerberg B., "Multivariate Statistical Methods in Process Analysis and Control," *AIChE Symposium Proceedings of the Fourth International Conference on Chemical Process Control*, *AIChE* , **P-67**,79, (1991).

Maheshvari S. N. and S. L. Hakimi, "On Models for Diagnosable Systems and Probabilistic Fault Diagnosis," *IEEE Trans. Comp*, **c-25**(3), 66 , March (1976).

Malthouse E.C., "Limitations of Nonlinear PCA as Performed with Generic Neural Networks," *IEEE Transactions on Neural Networks,***9**(1),165 (1998).

Miller, P., R.E. Swanson and C.F. Heckler, "Contribution Plots: A Missing Link in Multivariate Quality Control," *Appl. Math. Comput. Sci.*, **8(4)**, 775 (1998).

Moghaddam B., "Probabilistic Visual Learning for Object Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(7), 696 (1997).

Moller J.C., "Neural Network-Based Sensor Validation for Turboshaft Engines," *AIAA/ASME/SAE/ASEE 34th Joint Propulsion Conference and Exhibit*, Cleveland, OH, 455, July (1998).

Muteki K., J.F. MacGregor and T. Ueda, "Estimation of Missing Data Using Latent Variable Methods with Auxiliary Information," *Chemometrics and Intelligent Laboratory Systems*, **78**, 41 (2005).

Najafi M., C. Culp, R. Langari, "Enhanced Auto-Associative Neural Networks for Sensor Diagnostics (E-AANN)," *IEEE International Conference on Fuzzy Systems*, **1**,453, July (2004).

Namburu S. M., M. S. Azam, K. C. Luo and K. R. Pattipati, "Data-Driven Modeling, Fault Diagnosis and Optimal Sensor Selection for HVAC Chillers," *IEEE Transactions on Automation Science and Engineering*, **(4)**, 3, July (2007).

Nandi S., H.A. Toliyat and X. Li, "Condition Monitoring and Fault Diagnosis of Electrical Motors—A Review," *IEEE Transactions on Energy Conversion* , **20**, 4, December (2005).

Qin S.J. and W. Li, "Detection, Identification, and Reconstruction of Faulty Sensors with Maximized Sensitivity," *AIChE J.*, **45**, 9, September (1999).

Rossi, T. and J. Braun, *Classification of Fault Detection and Diagnostic Methods*. Technical Research Center of Finland, Helsinki, 1993.

Satin, A. L. and R. L. Gates, "Evaluation of Parity Equations for Gyro Failure Detection and Isolation," *Journal of Guidance and Control*, **1**, 14 (1978).

Schölkopf B., A.J. Smola and K.R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, **10**(5), 1299, (1998).

Sharifi R., D. Shahmirzadi, R. Langari and R. Gutierrez-Osuna, "Combinatorial Utilization of Artificial Neural Network and PCA Dimension Reduction Technique to Data Fault Detection and Correction," , *2nd Annual TAMUS Pathways Research Symposium*, Corpus Christi, TX, 421, October, (2004).

Sharifi R. and R. Langari, "A Hybrid AANN-KPCA Approach to Sensor Data Validation," *Proceedings of the 7th Conference on 7th WSEAS International Conference on Applied Informatics and Communications*, 361, Vouliagmeni , Greece (2007).

Sun R., F. Tsung and L. Qu, "Evolving Kernel Principal Component Analysis for Fault Diagnosis," *Computers & Industrial Engineering*, **53**, 361 (2007).

Venkatasubramanian V., R. Rengaswamy, K. Yin and S.N. Kavuri, "A Review of Process Fault Detection and Diagnosis: Part I: Quantitative Model-Based Methods," *Computers and Chemical Engineering*, **27**, 293 (2003a).

Venkatasubramanian V., R. Rengaswamy, K. Yin and S.N. Kavuri, "A Review of Process Fault Detection and Diagnosis: Part II: Qualitative Models and Search Strategies," *Computers and Chemical Engineering*, **27**, 313 (2003b).

Venkatasubramanian V., R. Rengaswamy, K. Yin and S.N. Kavuri, "A Review of Process Fault Detection and Diagnosis: Part III: Process History Based Methods," *Computers and Chemical Engineering*, **27**, 327 (2003c).

Wang P.J. and C. Cox, "Study on the Application of Auto-Associative Neural Network," *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, Shanghai, 26, August (2004).

Wang H., T. Chai, J. Ding and M. Brown, "Data Driven Fault Diagnosis and Fault Tolerant Control: Some Advances and Possible New Directions," *Acta Automatica Sinica*, **35**, 6, June (2009).

Wise B. M. and N. L. Ricker, "Recent Advances in Multivariate Statistical Process Control: Improving Robustness and Sensitivity," *IFAC Symposium on Advanced Control of Chemical Processes*, 125, Toulouse, France, October (1991).

Willsky A.S., "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica*, **12**, 601 (1976).

Tipping M. E. and C. M. Bishop, "Mixtures of Probabilistic Principal Component Analyzers," *Neural Computation*, **11**(2), 443 (2002).

Yang H., S. Cho, C. Tae and M. Zaheeruddin, "Sequential Rule Based Algorithms for Temperature Sensor Fault Detection in Air Handling Units," *Energy Conversion and Management*, **49**, 2291 (2008).

Zhao S.J. and Y. M. Xu, "Levenberg-Marquardt Algorithm for Nonlinear Principal Component Analysis Neural Network Through Inputs Training," *World Congress on Intelligent Control and Automation*, **4**, 3278, (2004).

APPENDIX A

PROOF OF THEOREM 1

Proof: First we prove that if a sensor is isolable, its corresponding fault image vector is unique. Assume the sensors $\boldsymbol{S_i}$ and $\boldsymbol{S_i}$ in a system have the same fault image vector. This means that their corresponding column vectors in the parity equation are mutually collinear.

Consider $\boldsymbol{q_i}$ and $\boldsymbol{q_j}$, the column vectors in the minimal parity structure, corresponding to sensors $\boldsymbol{S_i}$ and $\boldsymbol{S_i}$

$$\boldsymbol{Q} = [\boldsymbol{q_1}\ \boldsymbol{q_2}\ \dots \boldsymbol{q_i} \dots \boldsymbol{q_j} \dots]$$

If sensors $\boldsymbol{S_i}$ and $\boldsymbol{S_i}$ in a system have the same fault image vector, it means:

$$\frac{\boldsymbol{q_i}}{\|\boldsymbol{q_i}\|} = \frac{\boldsymbol{q_j}}{\|\boldsymbol{q_j}\|}$$

Therefore, $\boldsymbol{q_i}$ and $\boldsymbol{q_j}$ are mutually collinear, which can be written as

$$\boldsymbol{q_i} = \boldsymbol{\alpha q_j}$$

where α is a parameter which is dependent on the relative magnitudes of $\boldsymbol{q_i}$ and $\boldsymbol{q_j}$.

Now, if we have a fault with magnitude of d in sensor i, we have

$$\mathbf{y} = \boldsymbol{y}^* + \boldsymbol{n} + \delta_i \boldsymbol{\xi_i}$$

And the minimal residual is:

$$\boldsymbol{r} = \boldsymbol{Qy} = \boldsymbol{Qn} + \delta_i \boldsymbol{q_i} = \boldsymbol{Qn} + \delta_i \alpha \boldsymbol{q_j},$$

which is the same as the residual when we have fault in the j^{th} sensor with magnitude of $\delta_i \alpha$. They also have the same residual in any other parity equation, because all other residuals are just a linear transformation of minimal residual. Therefore these two sensors are not isolable from each other.

Now we prove that if a sensor has a unique sensor fault image, it is isolable. Assume the sensor i has a unique fault image vector, $\boldsymbol{n_i}$; therefore, its corresponding column vector, $\boldsymbol{q_i}$ is linearly independent from all other column vectors of $\boldsymbol{Q}$. Therefore, when a fault with the magnitude of d happens in sensor S_i , the minimal residual is

$$\boldsymbol{r} = \boldsymbol{Qy} = \boldsymbol{Qn} + \delta_i \boldsymbol{q_i}.$$

Assuming the noise has a Gaussian distribution $\boldsymbol{n} \sim \boldsymbol{N}(0, \boldsymbol{\Sigma})$, the distribution of the residual is $\boldsymbol{r} \sim \boldsymbol{N}(\delta_i \boldsymbol{q_i}, \boldsymbol{Q\Sigma Q})$

Since $\boldsymbol{q_i}$ is linearly independent from all other column vectors of $\boldsymbol{Q}$, the mean value of residual for error in sensor i, $\delta_i \boldsymbol{q_i}$, is also independent from any other single sensor fault. Therefore, it has a unique direction and it is isolable.

APPENDIX B

DERIVATION OF EQUATIOB 68

In order to derive Eq. (68) we consider the following four different cases and show in each case the equation is valid

$\boldsymbol{r}\cdot\boldsymbol{q_i} > 0$ and $\boldsymbol{r}\cdot\boldsymbol{q_j} > 0$

$\boldsymbol{r}\cdot\boldsymbol{q_i} < 0$ and $\boldsymbol{r}\cdot\boldsymbol{q_j} > 0$

$\boldsymbol{r}\cdot\boldsymbol{q_i} > 0$ and $\boldsymbol{r}\cdot\boldsymbol{q_j} < 0$

$\boldsymbol{r}\cdot\boldsymbol{q_i} < 0$ and $\boldsymbol{r}\cdot\boldsymbol{q_j} < 0$

Case 1: $\boldsymbol{r}\cdot\boldsymbol{q_i} > 0$ and $\boldsymbol{r}\cdot\boldsymbol{q_j} > 0$

$$\frac{(\delta_i\boldsymbol{q_i}+\boldsymbol{v})\cdot\boldsymbol{q_i}}{\|\delta_i\boldsymbol{q_i}+\boldsymbol{v}\|\cdot\|\boldsymbol{q_i}\|} > \frac{(\delta_i\boldsymbol{q_i}+\boldsymbol{v})\cdot\boldsymbol{q_j}}{\|\delta_i\boldsymbol{q_i}+\boldsymbol{v}\|\cdot\|\boldsymbol{q_j}\|}$$

Expanding the first part and elimination the common denominator factor $\|\boldsymbol{\delta q_i}+\boldsymbol{v}\|$, we have

$$\frac{\delta_i\boldsymbol{q_i}\cdot\boldsymbol{q_i}+\boldsymbol{v}\cdot\boldsymbol{q_i}}{\|\boldsymbol{q_i}\|} > \frac{(\delta_i\boldsymbol{q_i}+\boldsymbol{v})\cdot\boldsymbol{q_j}}{\|\boldsymbol{q_j}\|}$$

$$\frac{\delta_i \|\boldsymbol{q_i}\|^2 + \boldsymbol{v} \cdot \boldsymbol{q_i}}{\|\boldsymbol{q_i}\|} > \frac{(\delta_i \boldsymbol{q_i} + \boldsymbol{v}) \cdot \boldsymbol{q_j}}{\|\boldsymbol{q_j}\|}$$

$$\delta_i \|\boldsymbol{q_i}\| + \boldsymbol{v} \cdot \boldsymbol{n_i} > (\delta_i \boldsymbol{q_i} + \boldsymbol{v}) \cdot \boldsymbol{n_j}$$

$$\delta_i \|\boldsymbol{q_i}\| - \delta_i \boldsymbol{q_i} \cdot \boldsymbol{n_j} > \boldsymbol{v} \cdot \boldsymbol{n_j} - \boldsymbol{v} \cdot \boldsymbol{n_i}$$

Factoring the term $\boldsymbol{\delta}\|\boldsymbol{q_i}\|$ from the left hand side and the term $\boldsymbol{v}$ from the right hand side,

$$\delta_i \|\boldsymbol{q_i}\|(1 - \frac{\boldsymbol{q_i} \cdot \boldsymbol{n_j}}{\|\boldsymbol{q_i}\|}) > \boldsymbol{v} \cdot (\boldsymbol{n_j} - \boldsymbol{n_i})$$

$$\delta_i \|\boldsymbol{q_i}\|(1 - \boldsymbol{n_i} \cdot \boldsymbol{n_j}) > \boldsymbol{v} \cdot (\boldsymbol{n_j} - \boldsymbol{n_i})$$

On the other hand in equation 28 we defined $\boldsymbol{n}_{ij}$ as:

$$\boldsymbol{n}_{ij} = \begin{cases} \boldsymbol{n_i} + \boldsymbol{n_j}, & \boldsymbol{n_i} \cdot \boldsymbol{n_j} < 0 \\ \boldsymbol{n_i} - \boldsymbol{n_j}, & \boldsymbol{n_i} \cdot \boldsymbol{n_j} \geq \boldsymbol{0} \end{cases}.$$

since $(\delta_i \boldsymbol{q_i} + \boldsymbol{v}) \cdot \boldsymbol{q_i} > 0$ and $(\delta_i \boldsymbol{q_i} + \boldsymbol{v}) \cdot \boldsymbol{q_j} > 0$, as a result, $\boldsymbol{q_i} \cdot \boldsymbol{q_j} > 0$ which means $\boldsymbol{n_i} \cdot \boldsymbol{n_j} > 0$ and according to Eq 28. $\boldsymbol{n}_{ij} = \boldsymbol{n_i} - \boldsymbol{n_j}$ therefore, we have

$$\delta_i > \frac{-\boldsymbol{v} \cdot (\boldsymbol{n_{ij}})}{\|\boldsymbol{q_i}\|(1 - \boldsymbol{n_i} \cdot \boldsymbol{n_j})}$$

On the other hand $\boldsymbol{n_i} \cdot \boldsymbol{n_j} > 0$ as a result $\boldsymbol{n_i} \cdot \boldsymbol{n_j} = |\boldsymbol{n_i} \cdot \boldsymbol{n_j}|$ also defining $\rho_{ij} = |\boldsymbol{v} \cdot \boldsymbol{n}_{ij}|$ we have

$$\delta_i > \frac{\boldsymbol{\rho_{ij}}}{\|\boldsymbol{q_i}\|\left(\mathbf{1} - \left|\vec{\boldsymbol{n}}_{\boldsymbol{i}} \cdot \vec{\boldsymbol{n}}_{\boldsymbol{j}}\right|\right)}$$

Case 2: $\boldsymbol{r} \cdot \boldsymbol{q_i} < 0$ and $\boldsymbol{r} \cdot \boldsymbol{q_j} > 0$

Simalrly in this case we can show:

$$-\delta_i \|\boldsymbol{q_i}\|(1 + \boldsymbol{n_i} \cdot \boldsymbol{n_j}) > \boldsymbol{v} \cdot \left(\boldsymbol{n_j} + \boldsymbol{n_i}\right)$$

On the other hand in equation 28 we defined $\boldsymbol{n}_{ij}$ as:

$$\boldsymbol{n}_{ij} = \begin{cases} \boldsymbol{n_i} + \boldsymbol{n_j}, & \boldsymbol{n_i} \cdot \boldsymbol{n_j} < 0 \\ \boldsymbol{n_i} - \boldsymbol{n_j}, & \boldsymbol{n_i} \cdot \boldsymbol{n_j} \geq \mathbf{0} \end{cases}.$$

since $(\delta_i \boldsymbol{q_i} + \boldsymbol{v}) \cdot \boldsymbol{q_i} < 0$ and $(\delta_i \boldsymbol{q_i} + \boldsymbol{v}) \cdot \boldsymbol{q_j} > 0$, as a result, $\boldsymbol{q_i} \cdot \boldsymbol{q_j} < 0$ which means $\boldsymbol{n_i} \cdot \boldsymbol{n_j} < 0$ and according to Eq 28 $\boldsymbol{n}_{ij} = \boldsymbol{n_i} + \boldsymbol{n_j}$ therefore, we have $\delta_i > \frac{-\boldsymbol{v} \cdot (\boldsymbol{n_{ij}})}{\|\boldsymbol{q_i}\|(1 + \boldsymbol{n_i} \cdot \boldsymbol{n_j})}$

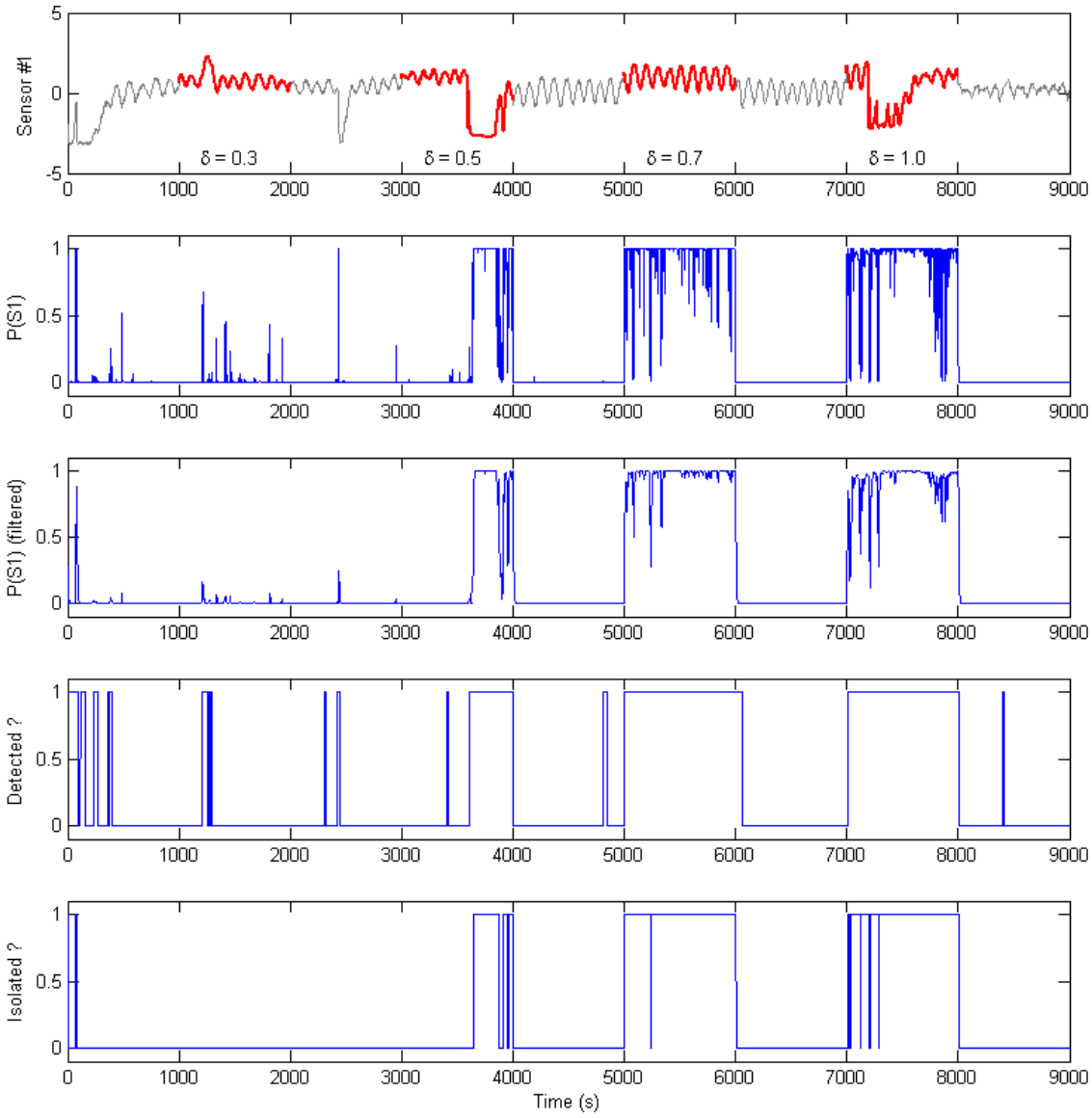
On the other hand $\boldsymbol{n_i} \cdot \boldsymbol{n_j} < 0$ as a result $\boldsymbol{n_i} \cdot \boldsymbol{n_j} = -\left|\boldsymbol{n_i} \cdot \boldsymbol{n_j}\right|$ also defining $\rho_{ij} = \left|\boldsymbol{v} \cdot \boldsymbol{n}_{ij}\right|$ we have

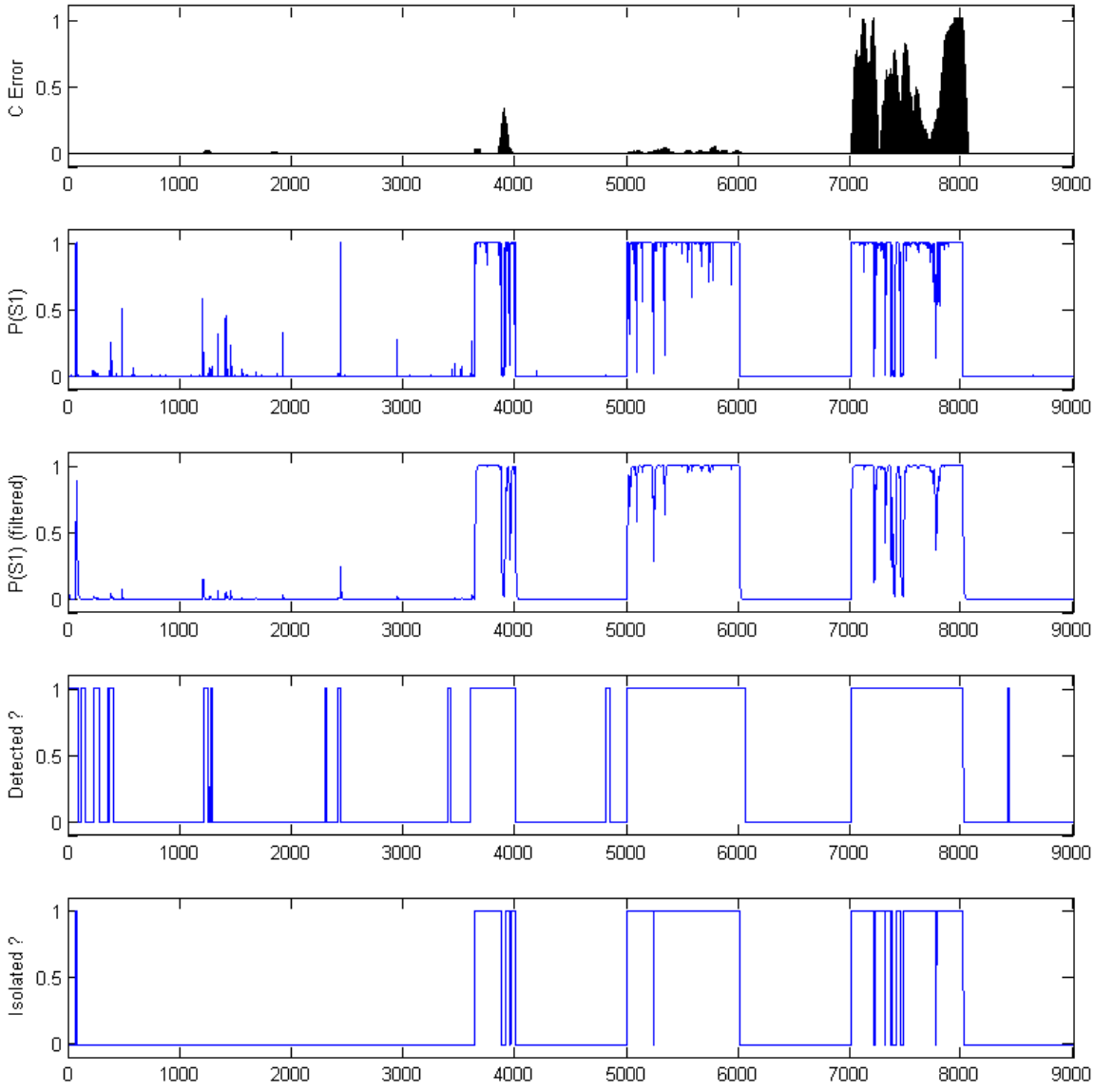
$$\delta_i > \frac{\boldsymbol{\rho_{ij}}}{\|\boldsymbol{q_i}\|\left(\mathbf{1} - \left|\vec{\boldsymbol{n}}_{\boldsymbol{i}} \cdot \vec{\boldsymbol{n}}_{\boldsymbol{j}}\right|\right)}$$

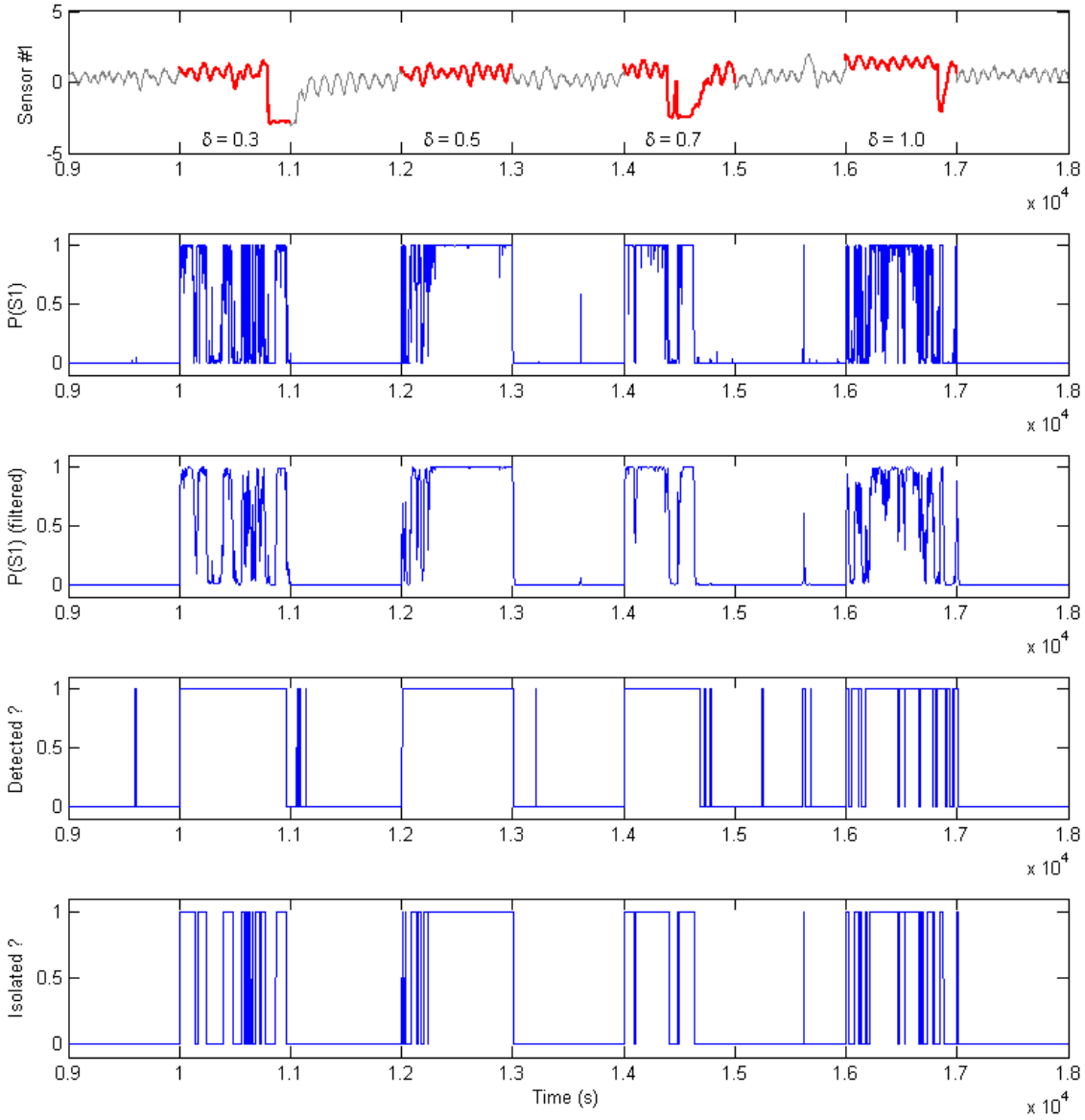
For the two other cases similarly we can show this expression is valid.

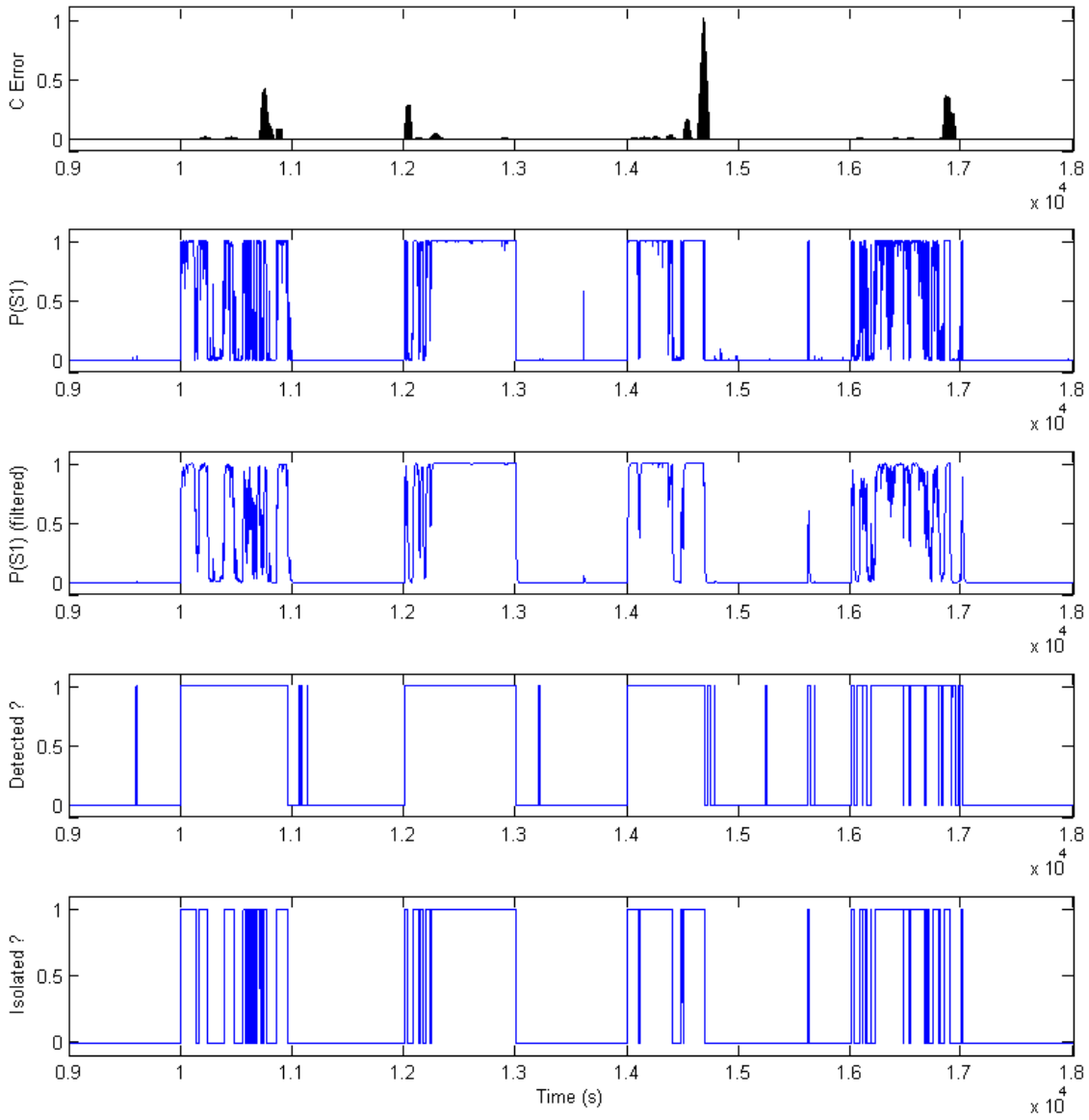
APPENDIX C

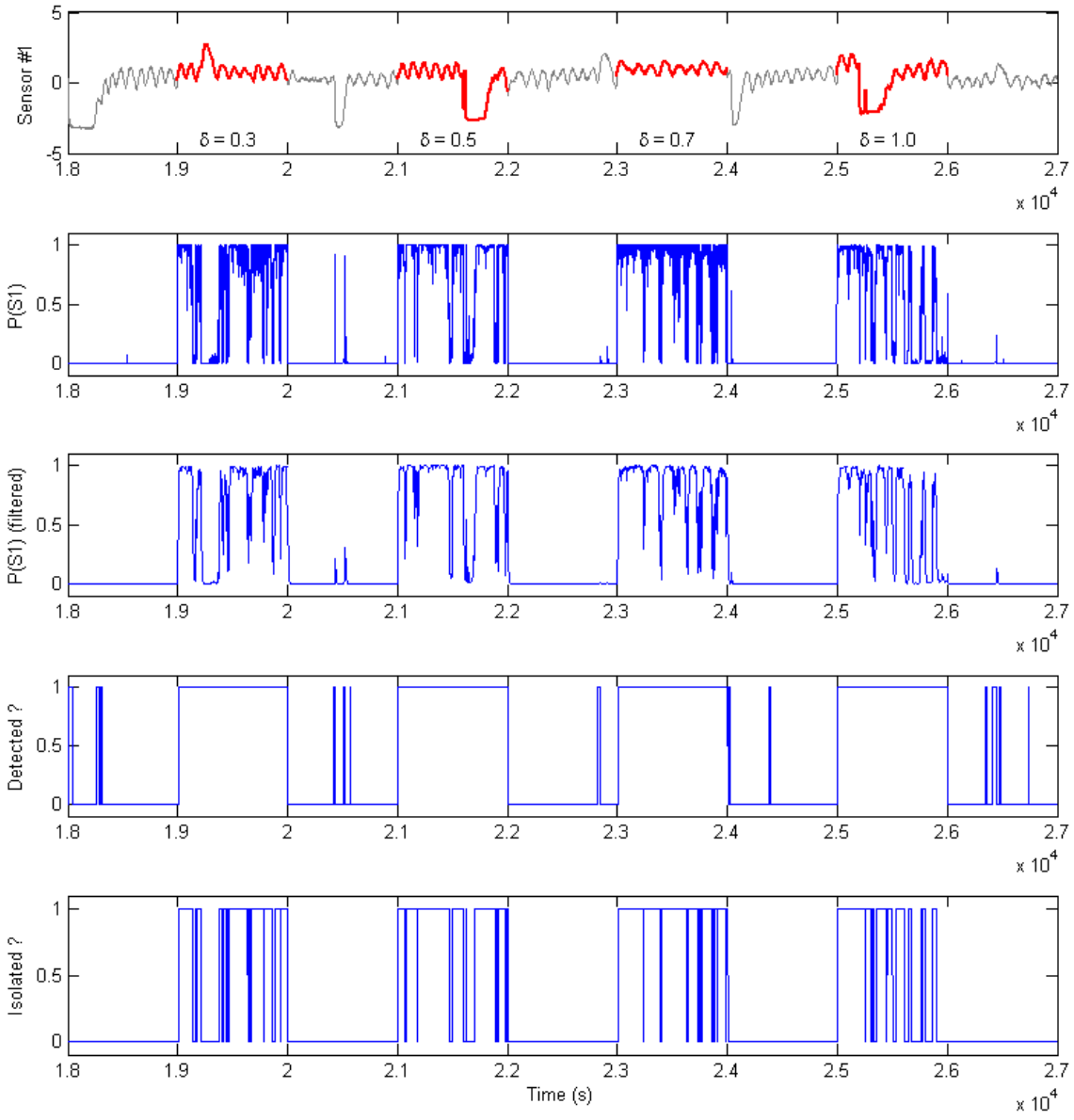
COMPLETE ANALYSIS OF SENSOR DIAGNOSIS IN HVAC

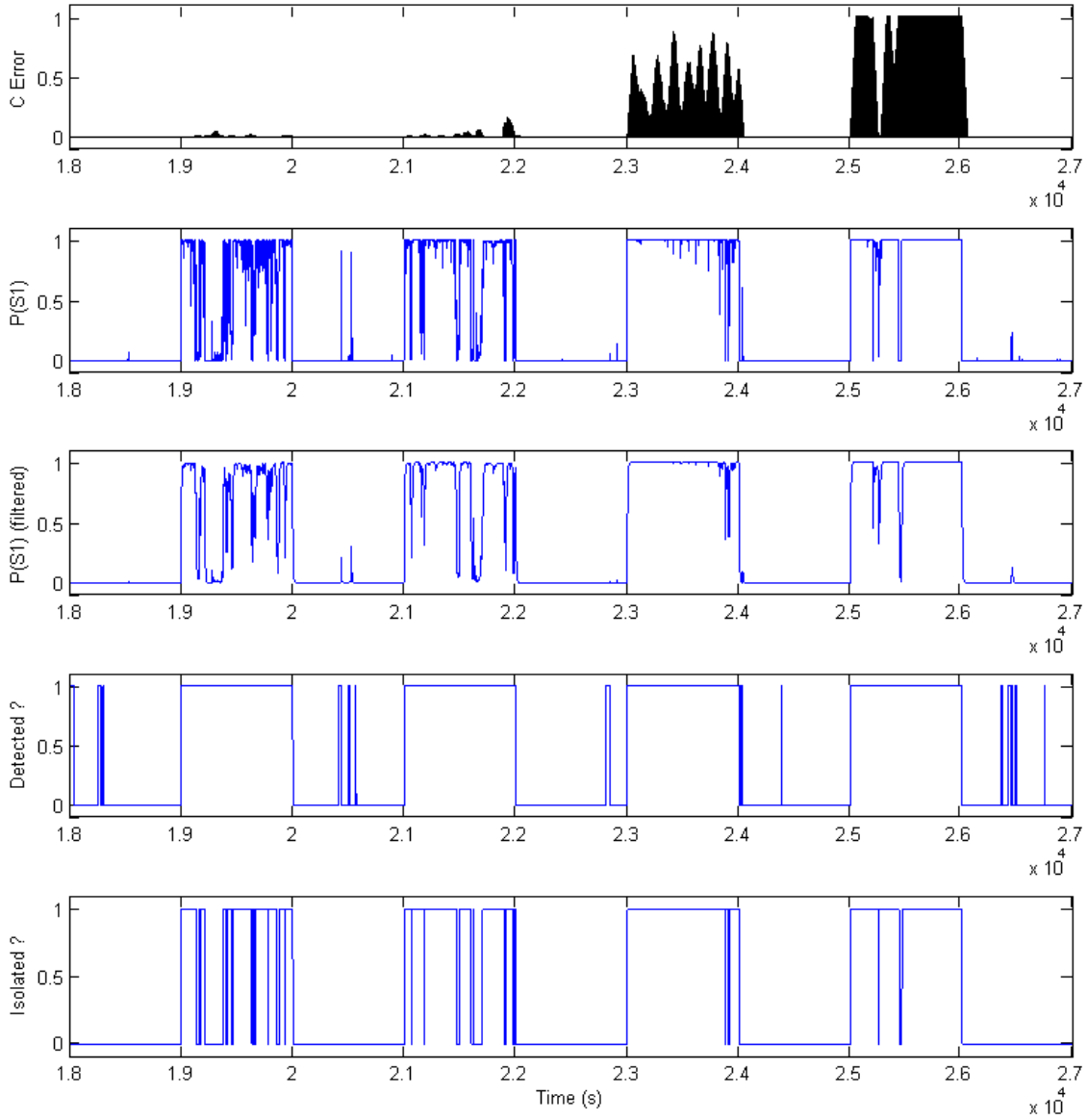


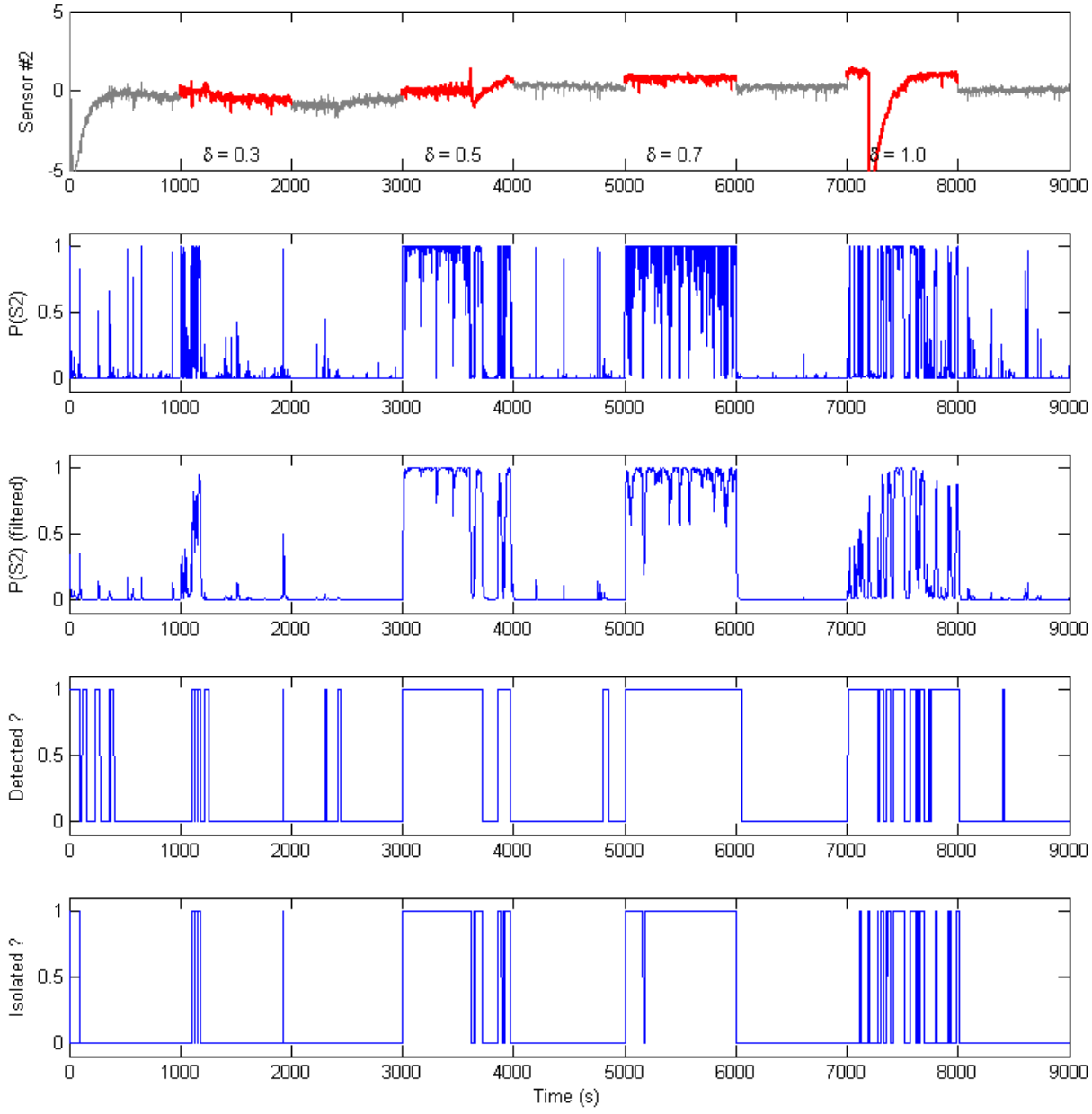

C Error
P(S1)
P(S1) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000

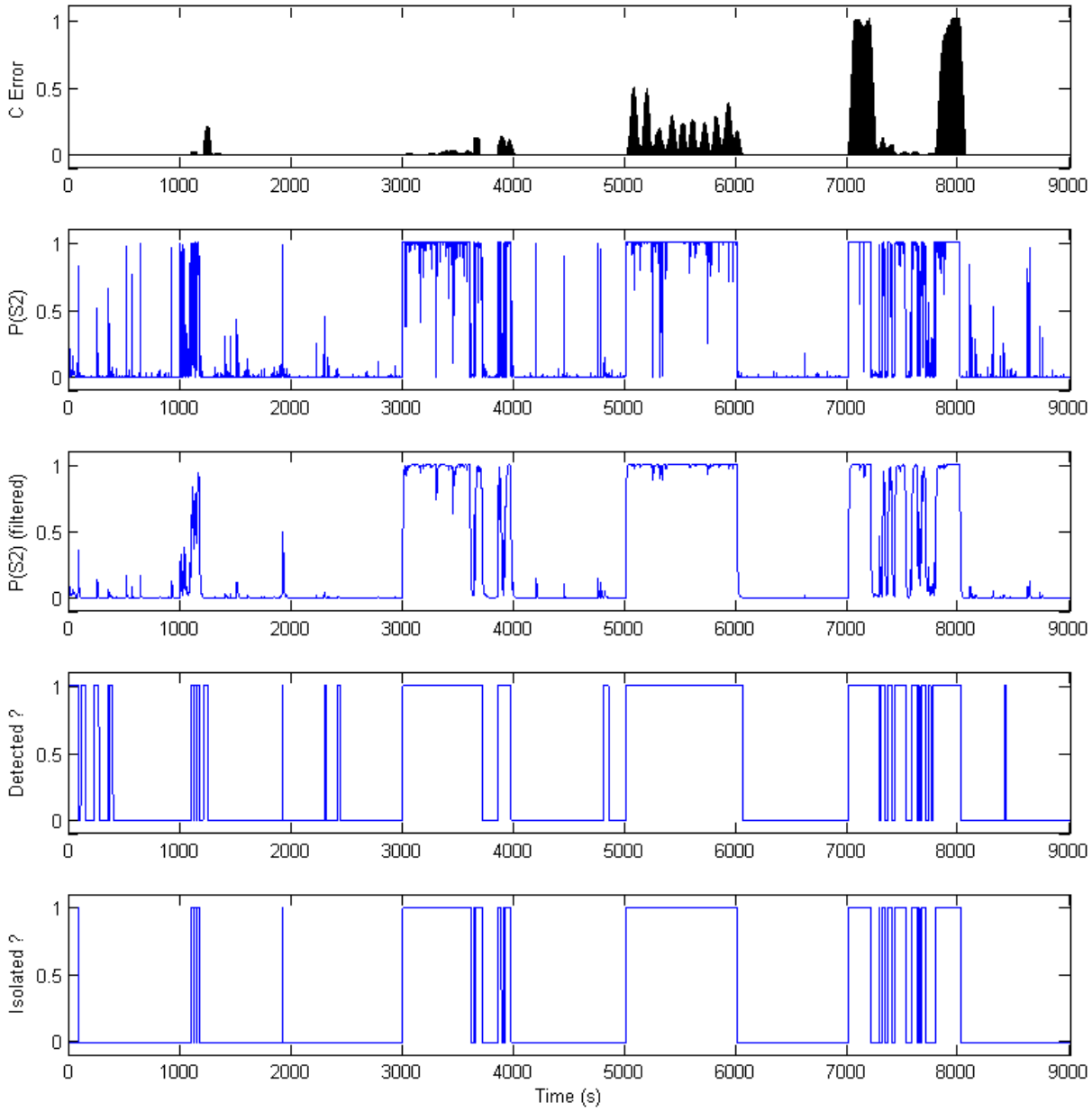

Sensor #1
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S1)
P(S1) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

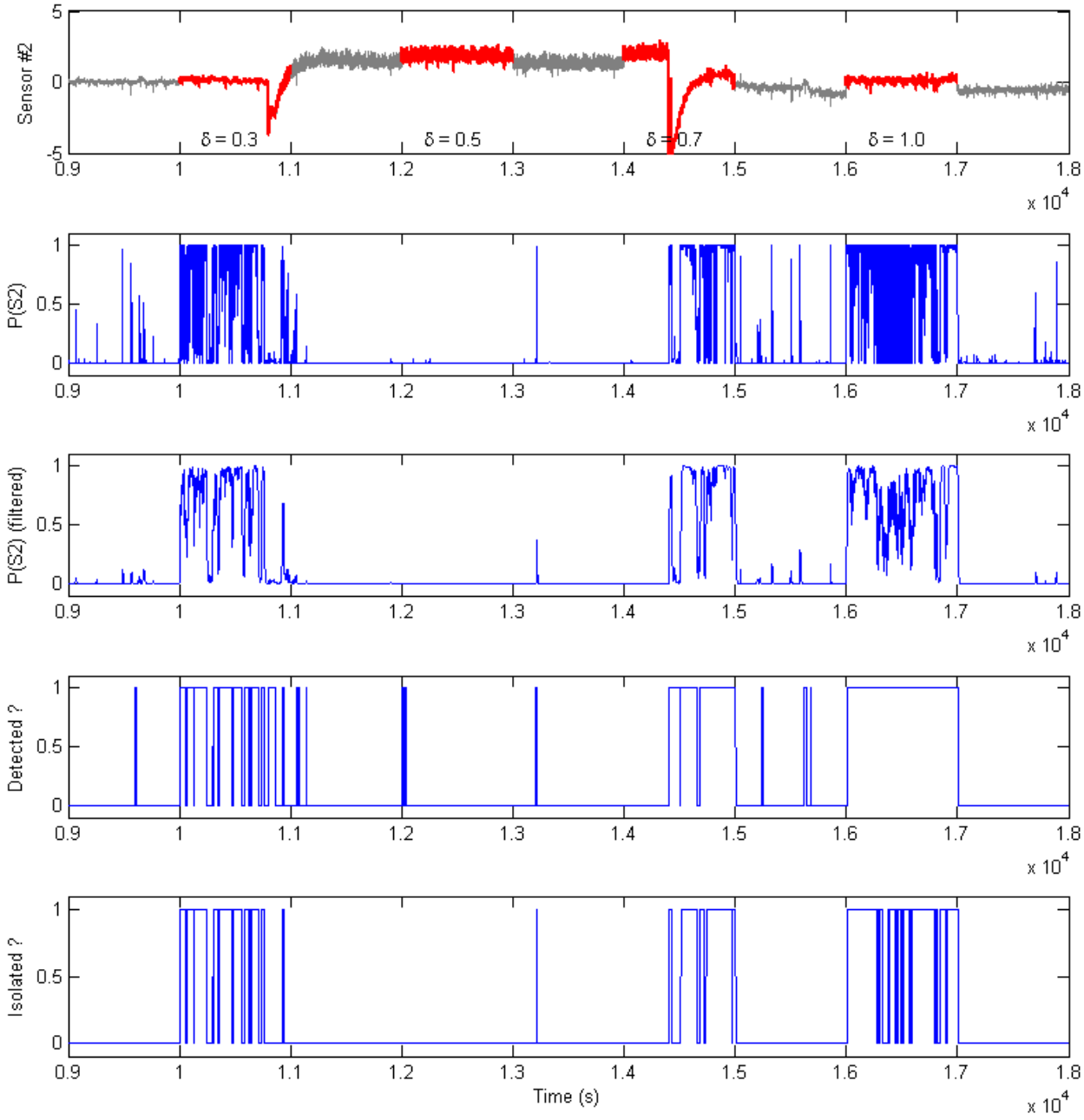

C Error
P(S1)
P(S1) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

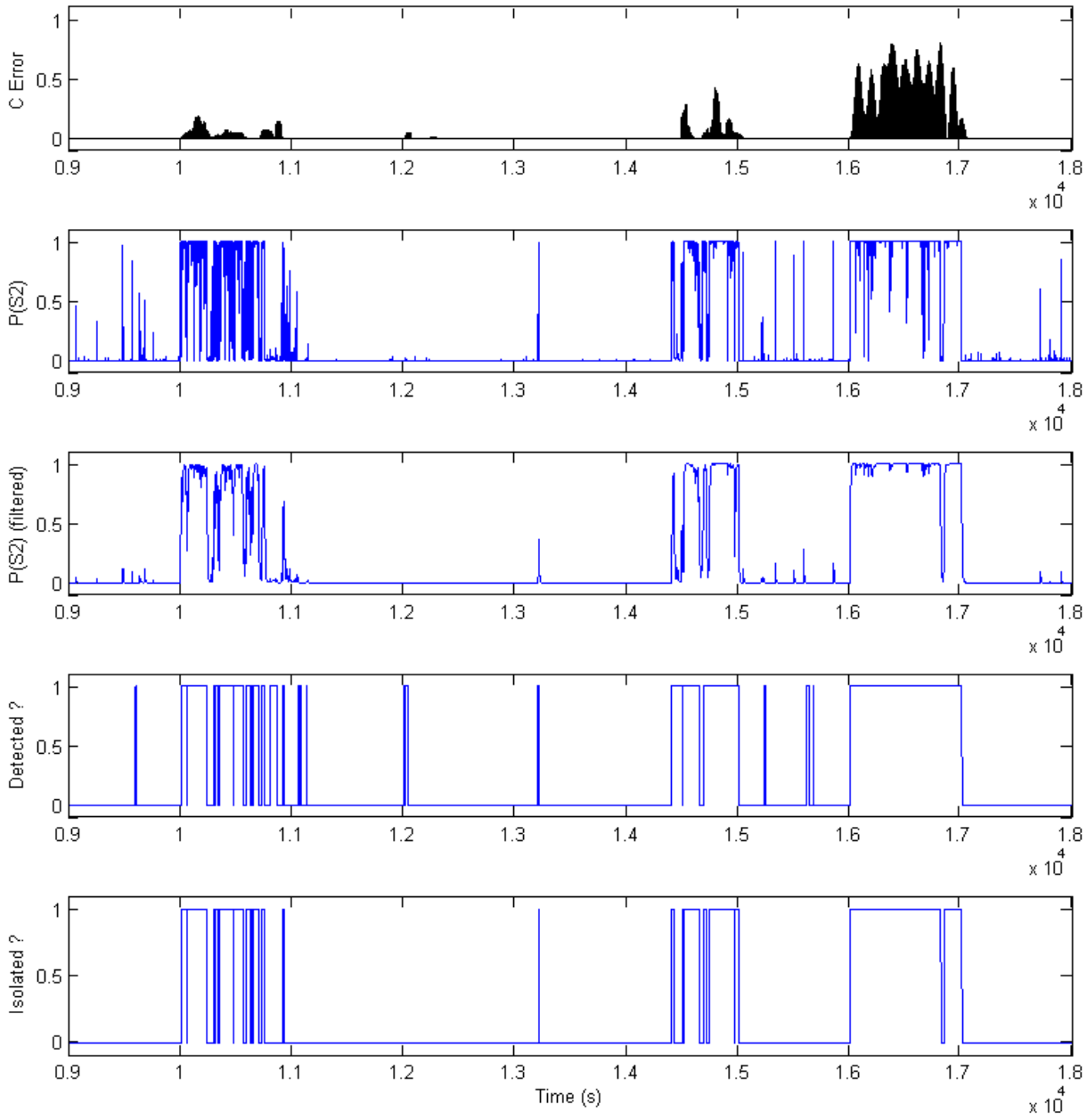

Sensor #1
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S1)
P(S1) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

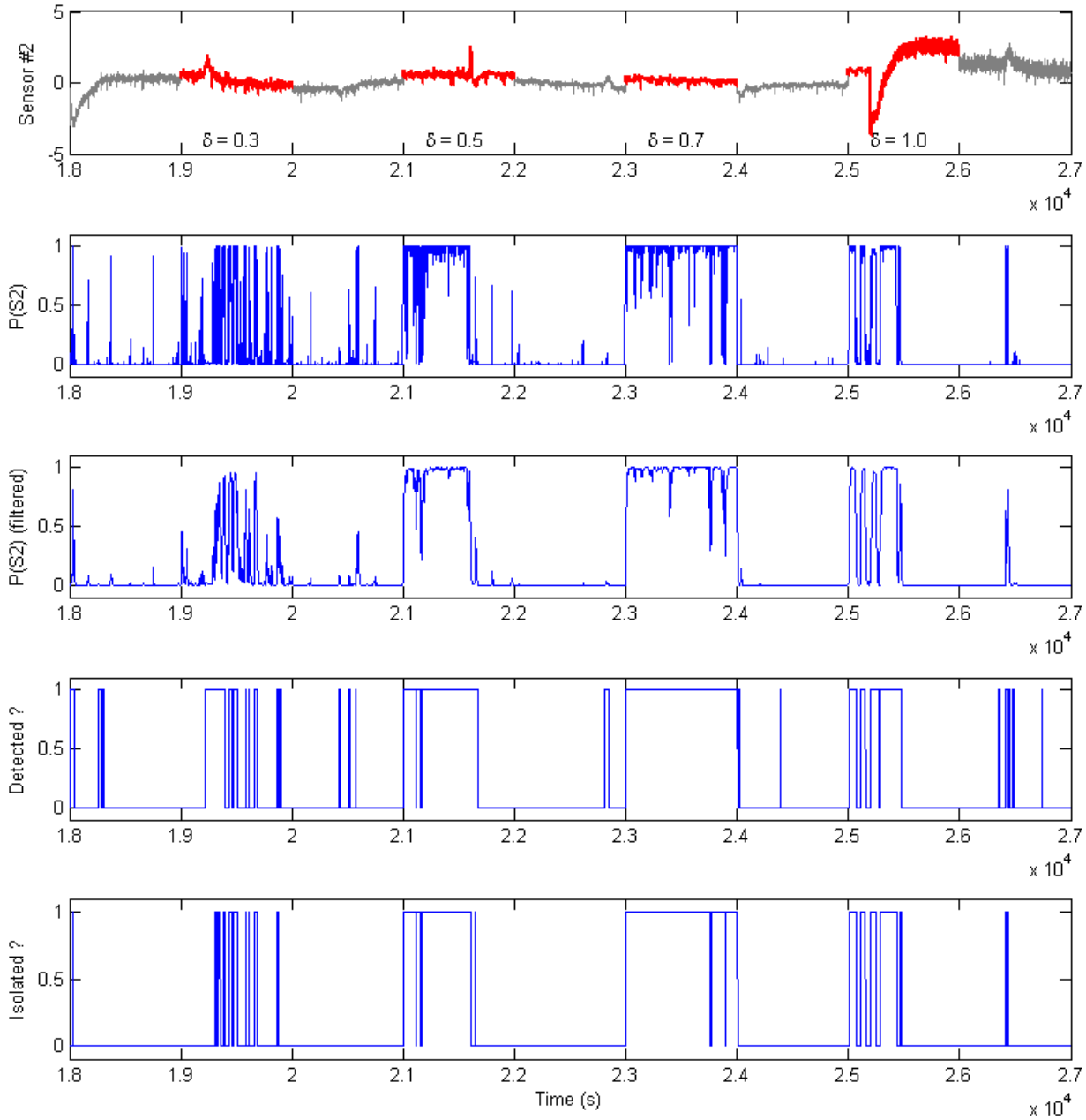

C Error
P(S1)
P(S1) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

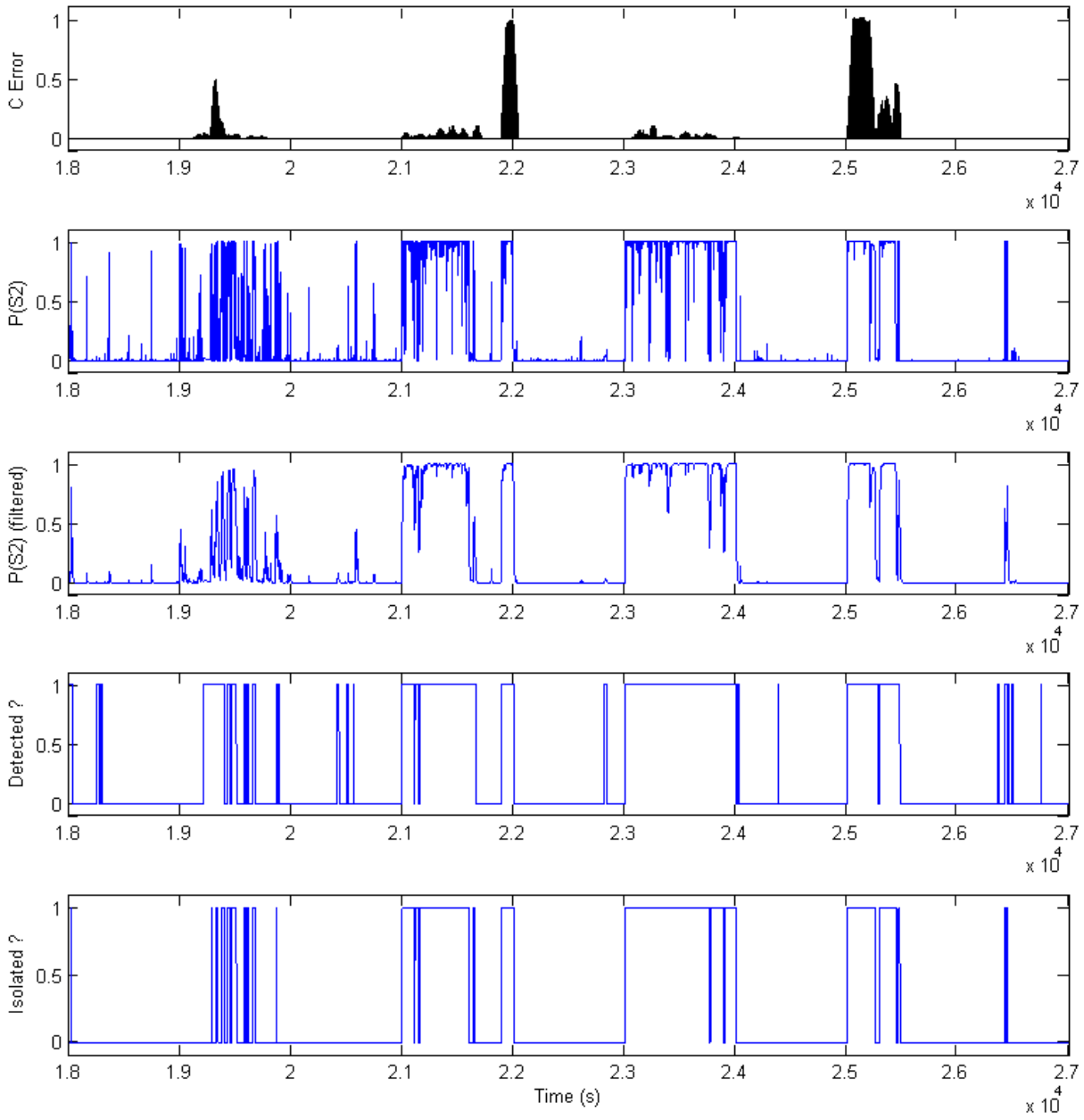

Sensor #2
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S2)
P(S2) (filtered)
Detected ?
Isolated ?
Time (s)

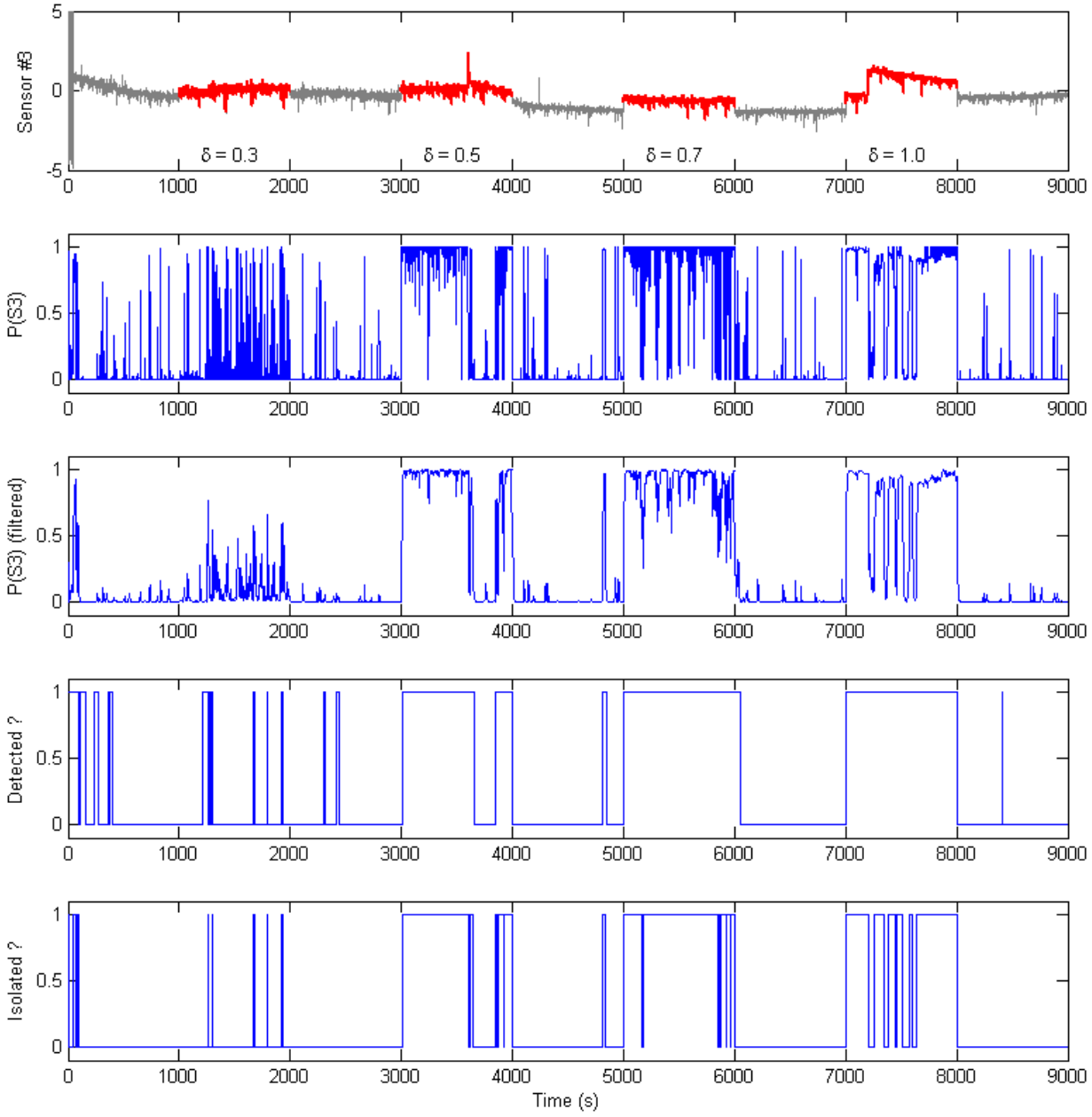

C Error
P(S2)
P(S2) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

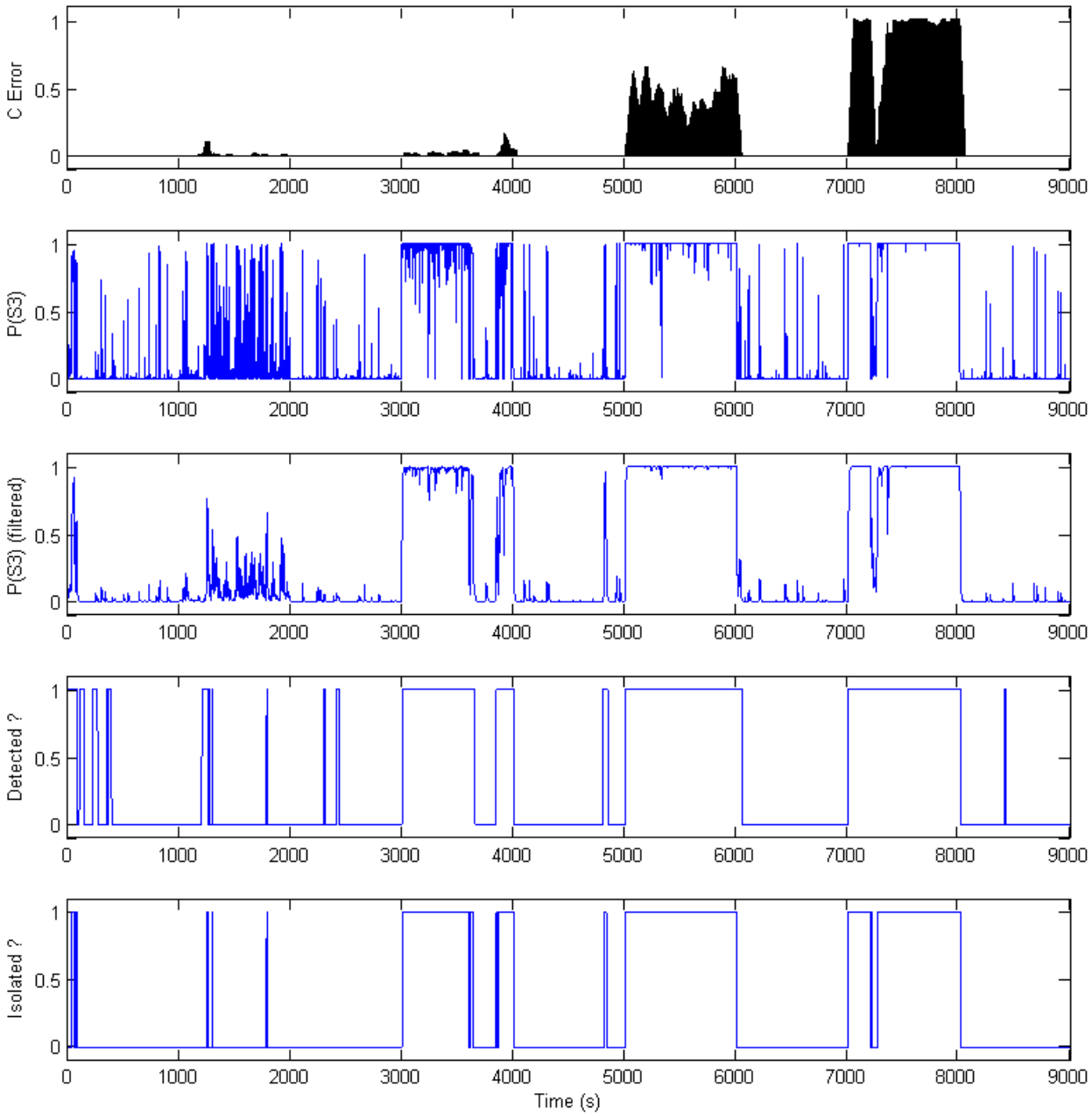

Sensor #2
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S2)
P(S2) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

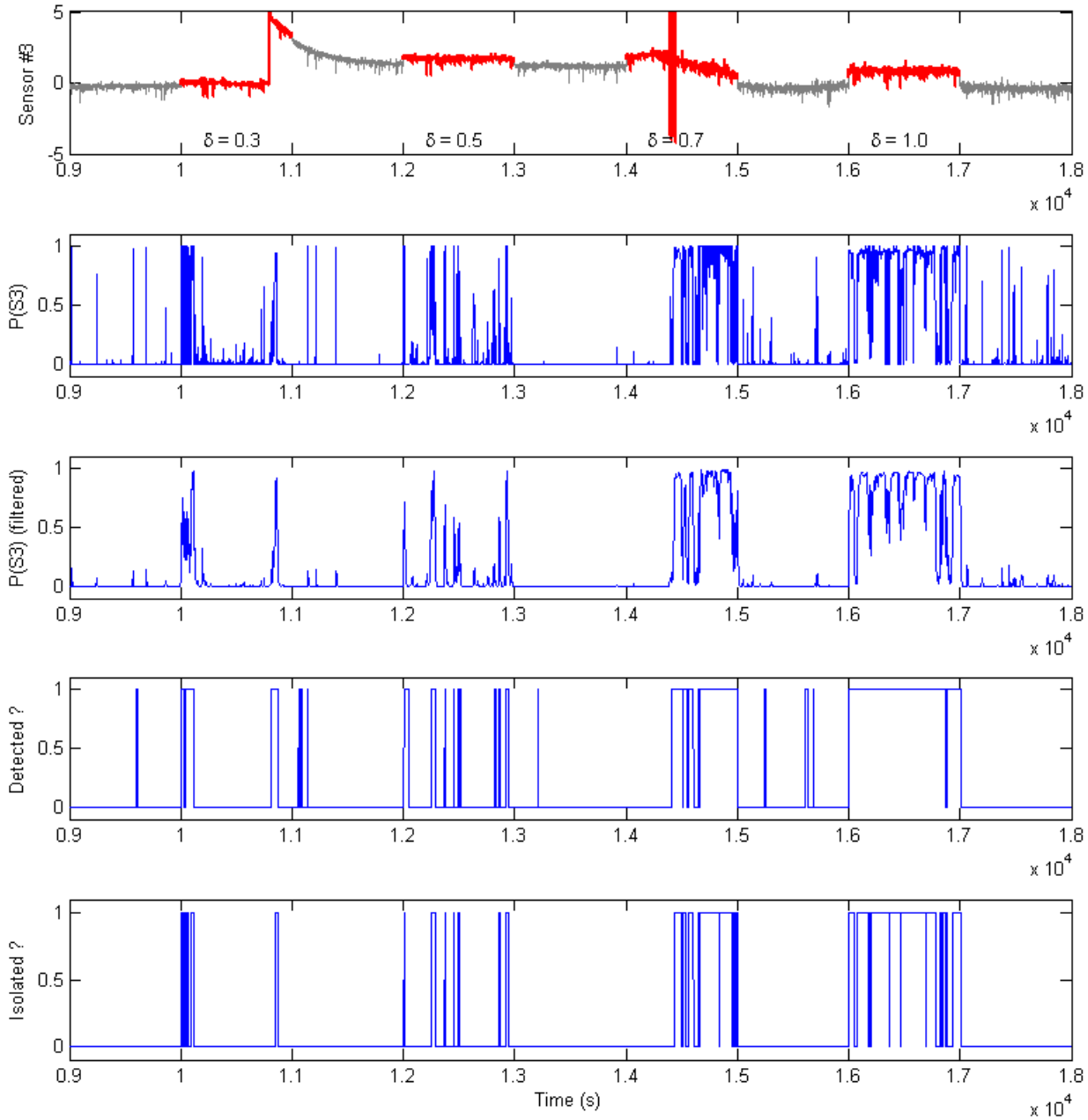

C Error
P(S2)
P(S2) (filtered)
Detected ?
Isolated ?
0
0.5
1
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
x 10^4
Time (s)

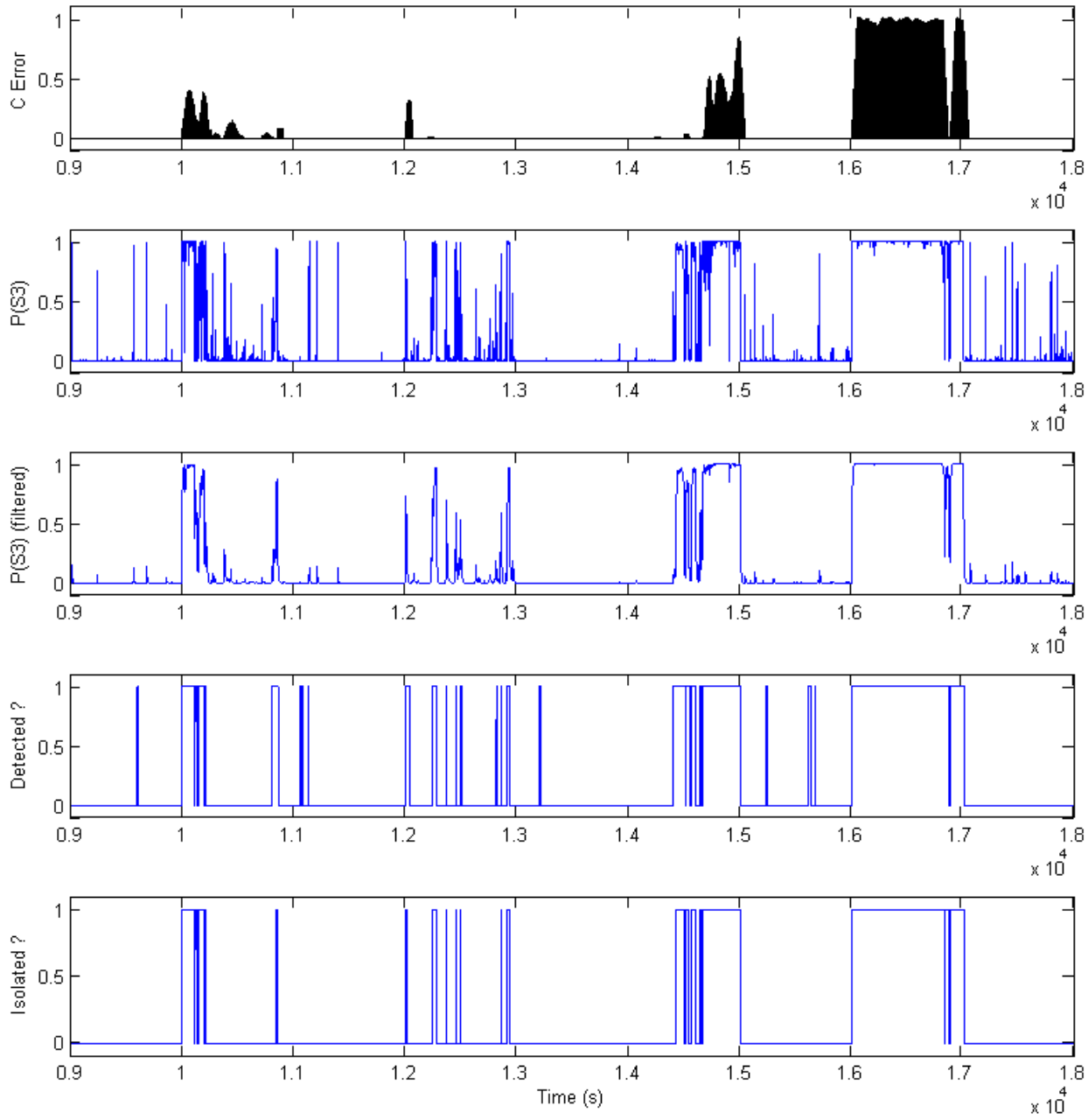

Sensor #2
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S2)
P(S2) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

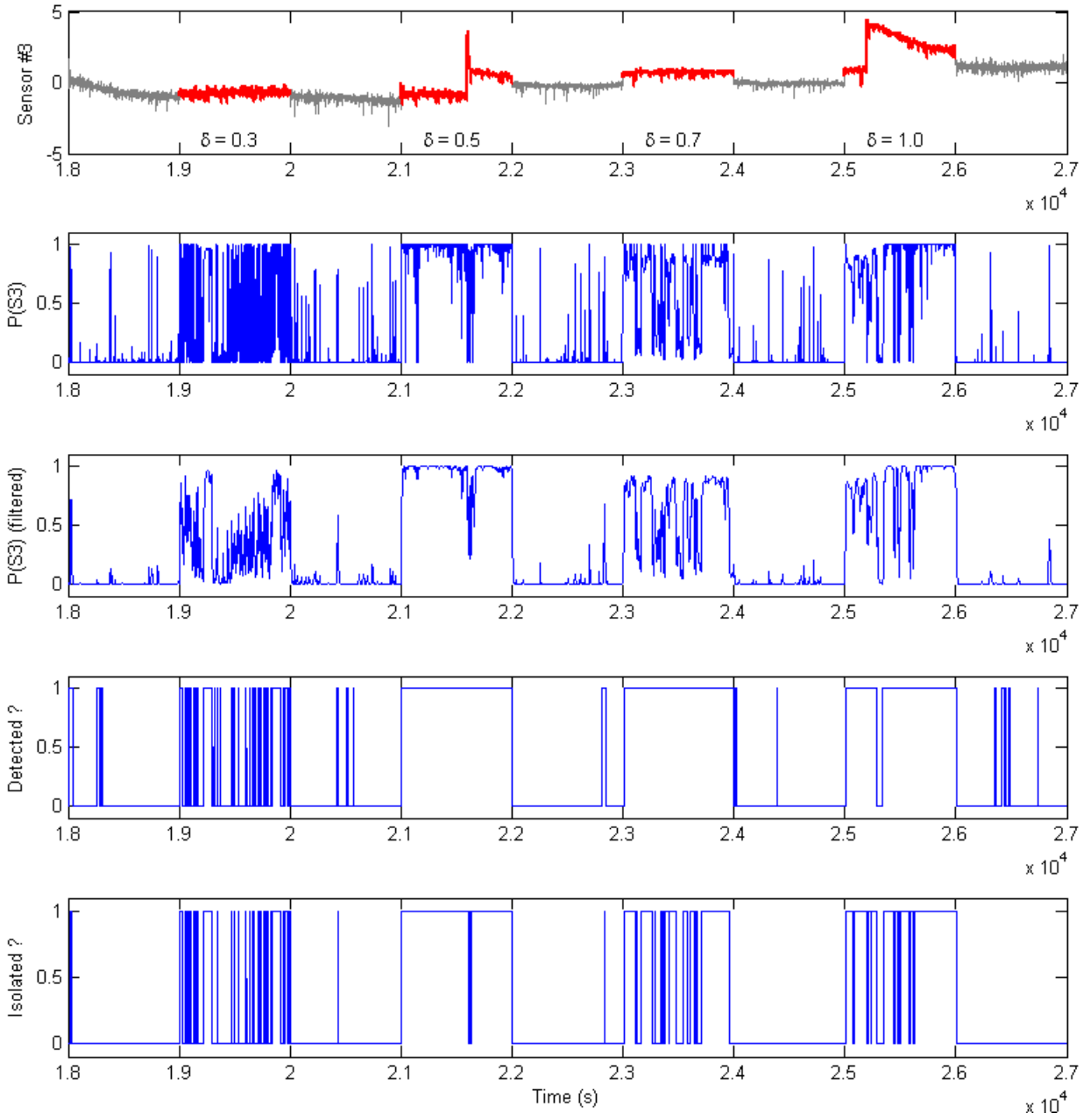

C Error
P(S2)
P(S2) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

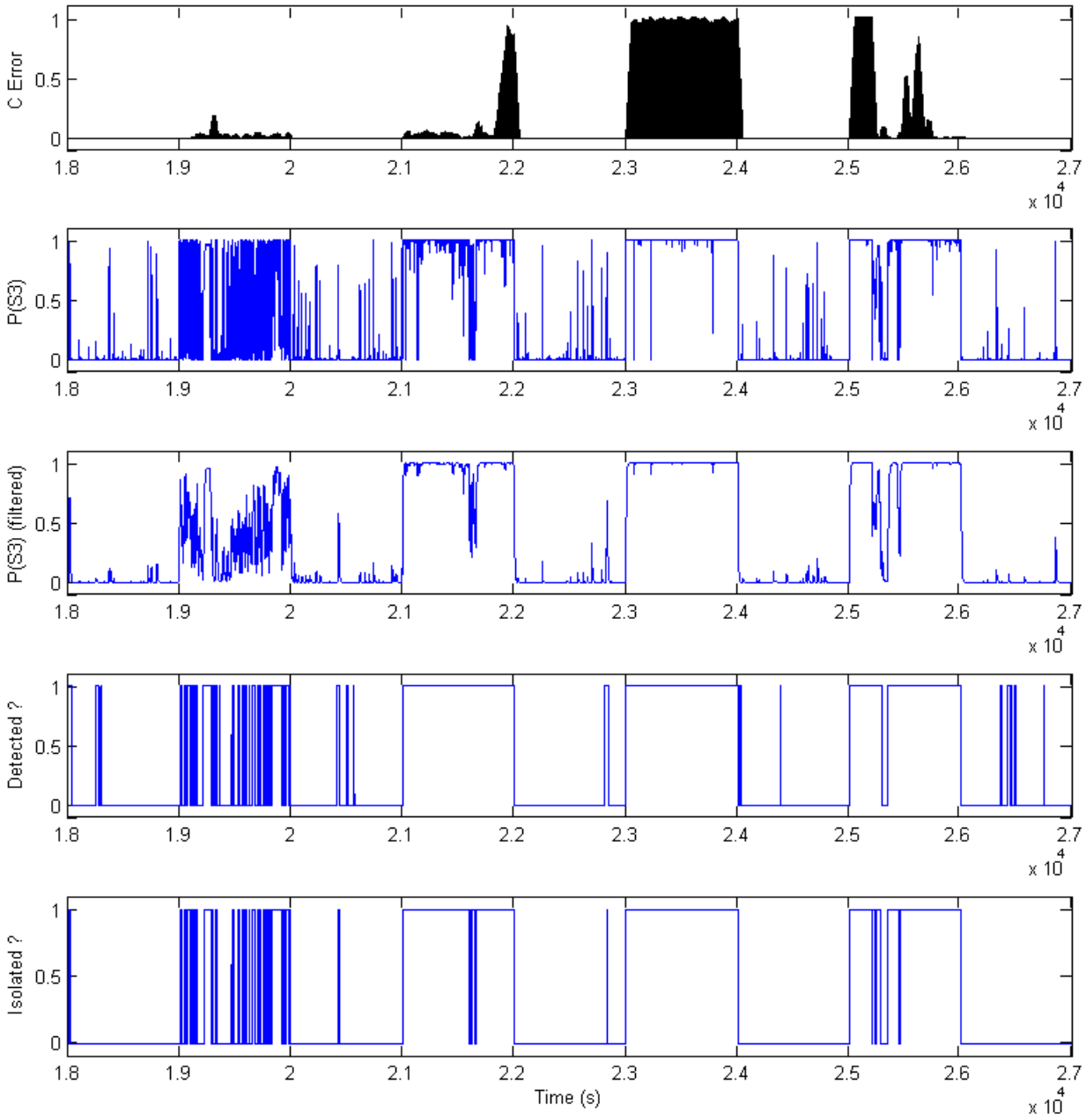

Sensor #3
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S3)
P(S3) (filtered)
Detected ?
Isolated ?
Time (s)

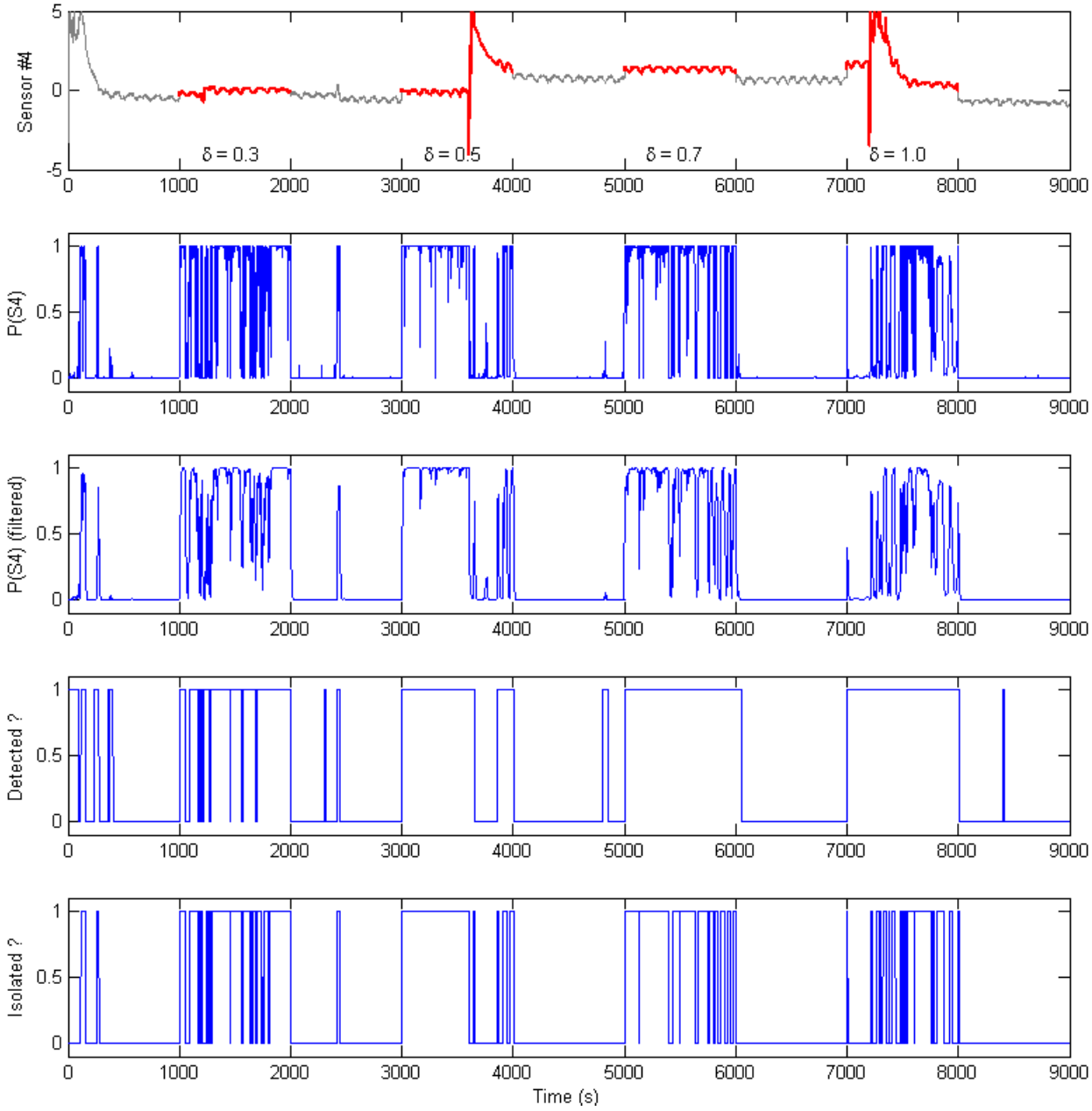

C Error
P(S3)
P(S3) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

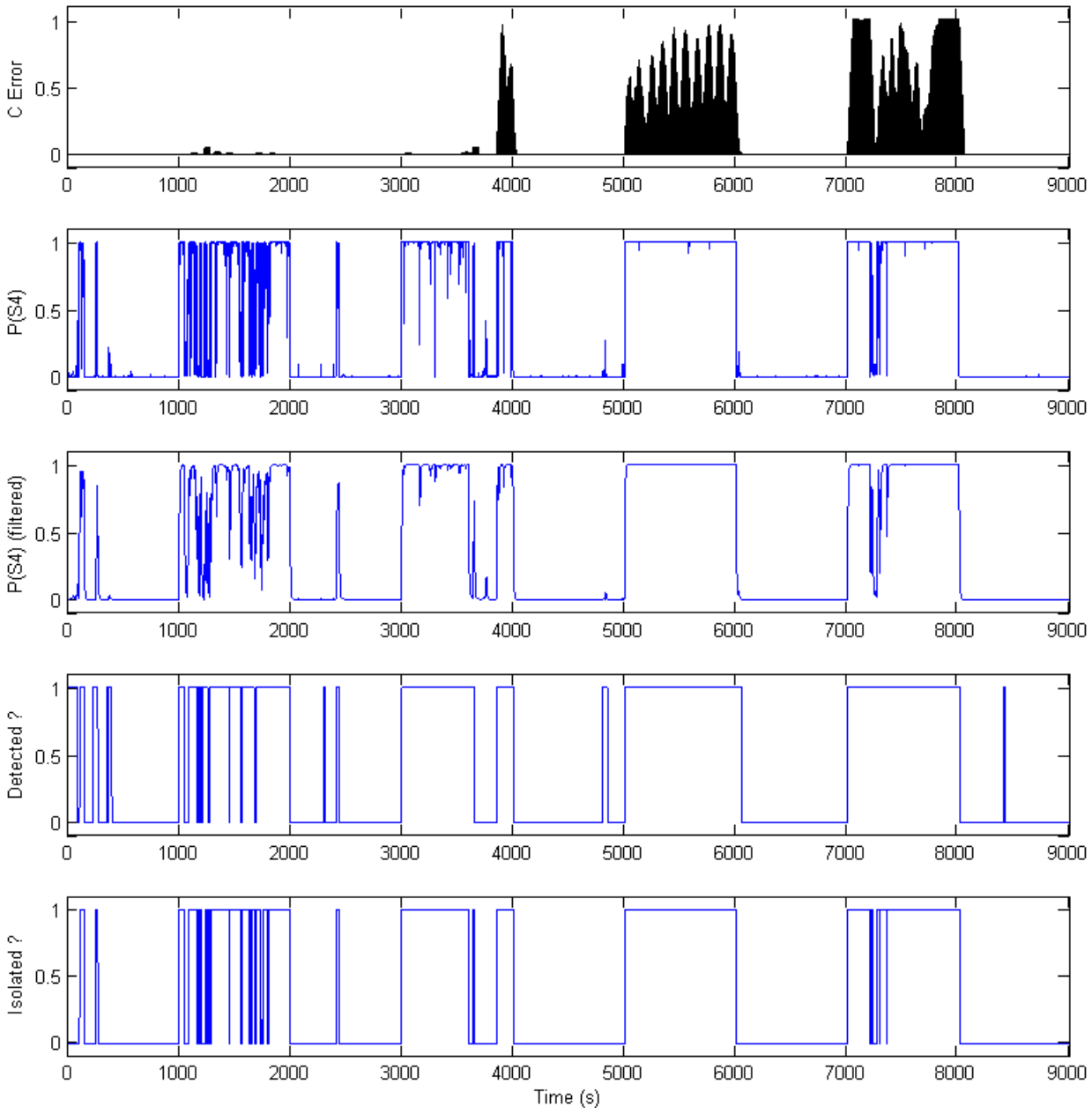

Sensor #3
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S3)
P(S3) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

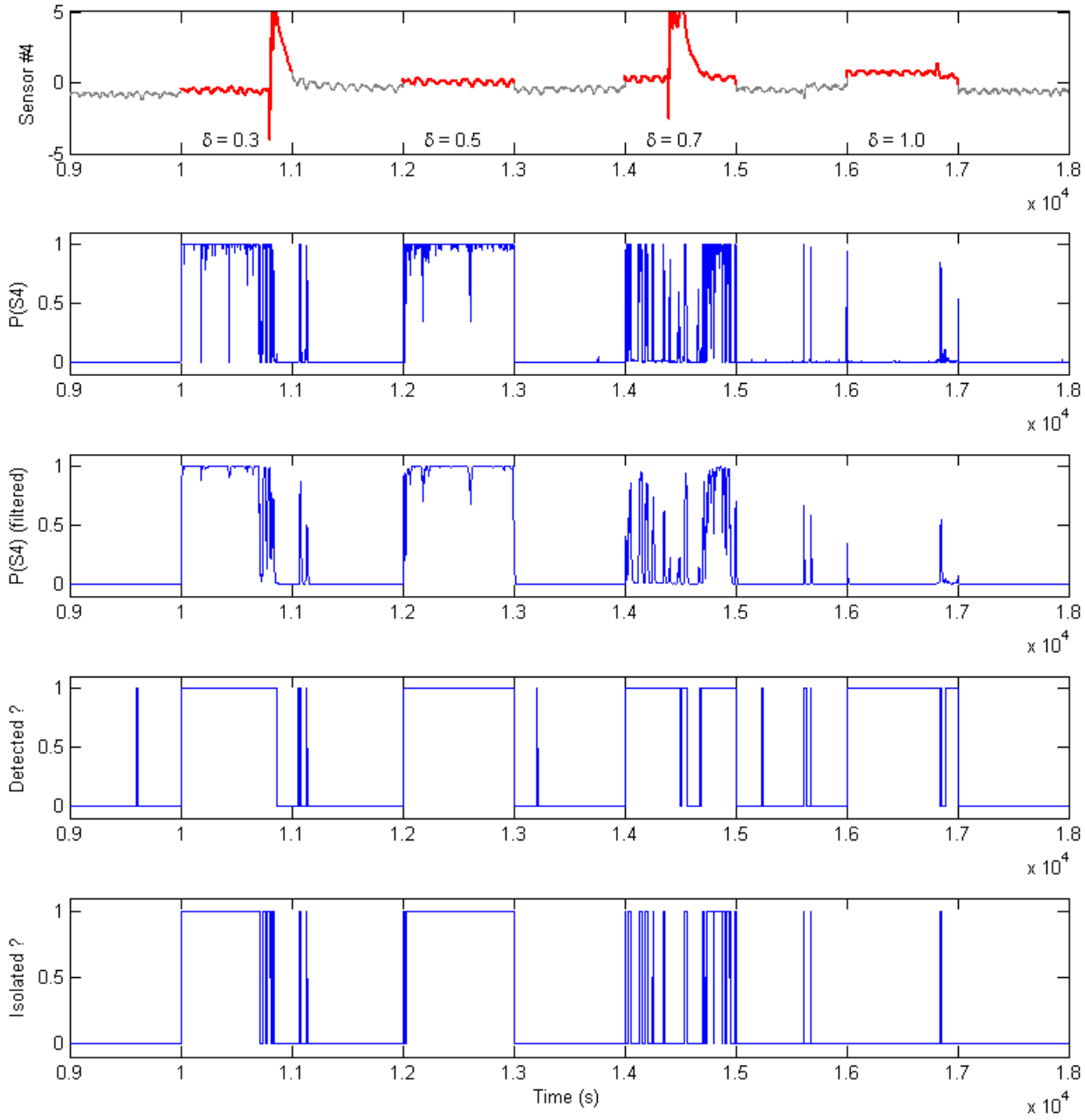

C Error
P(S3)
P(S3) (filtered)
Detected ?
Isolated ?
0
0.5
1
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
x 10^4
Time (s)

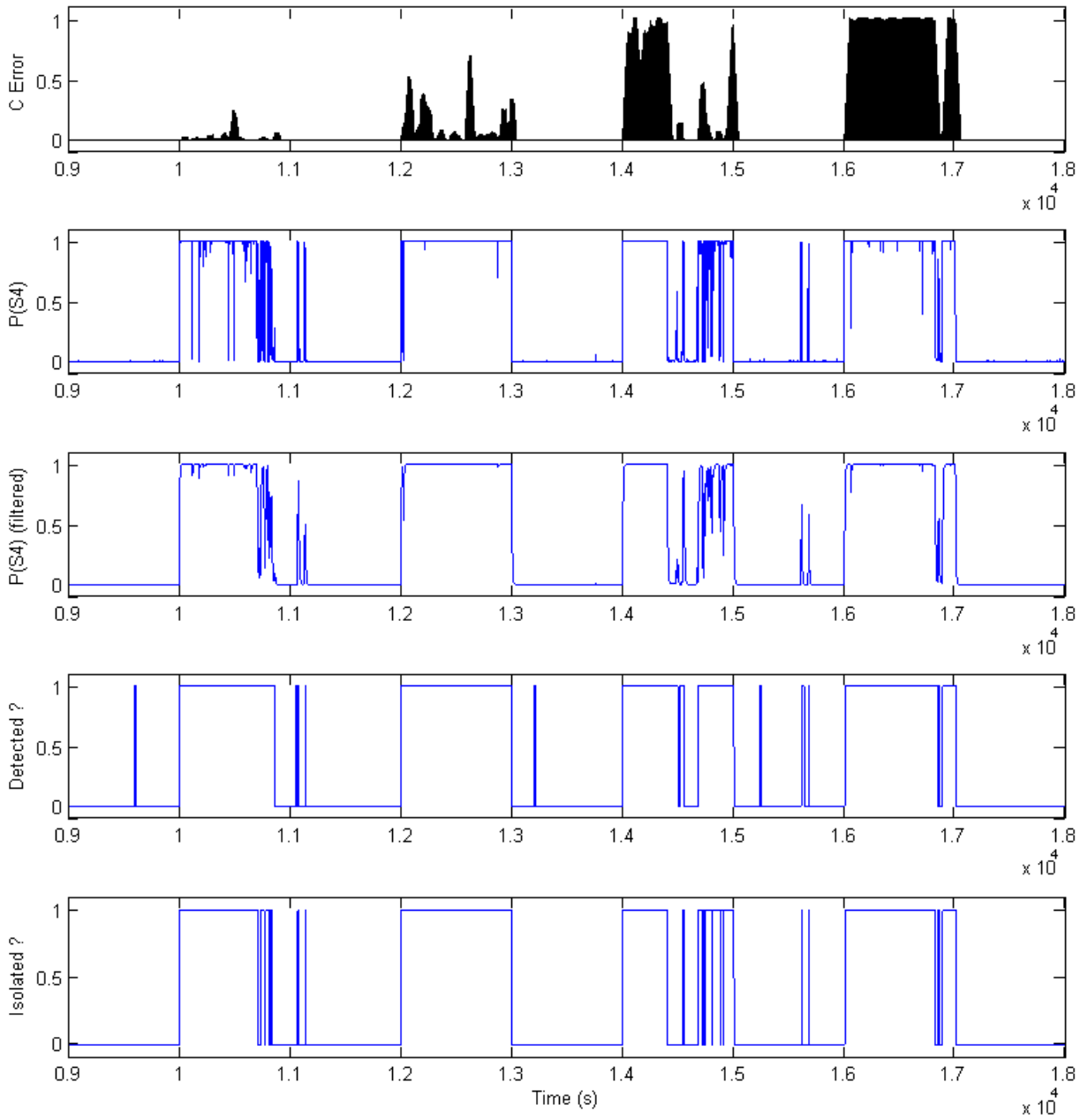

Sensor #3
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S3)
P(S3) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

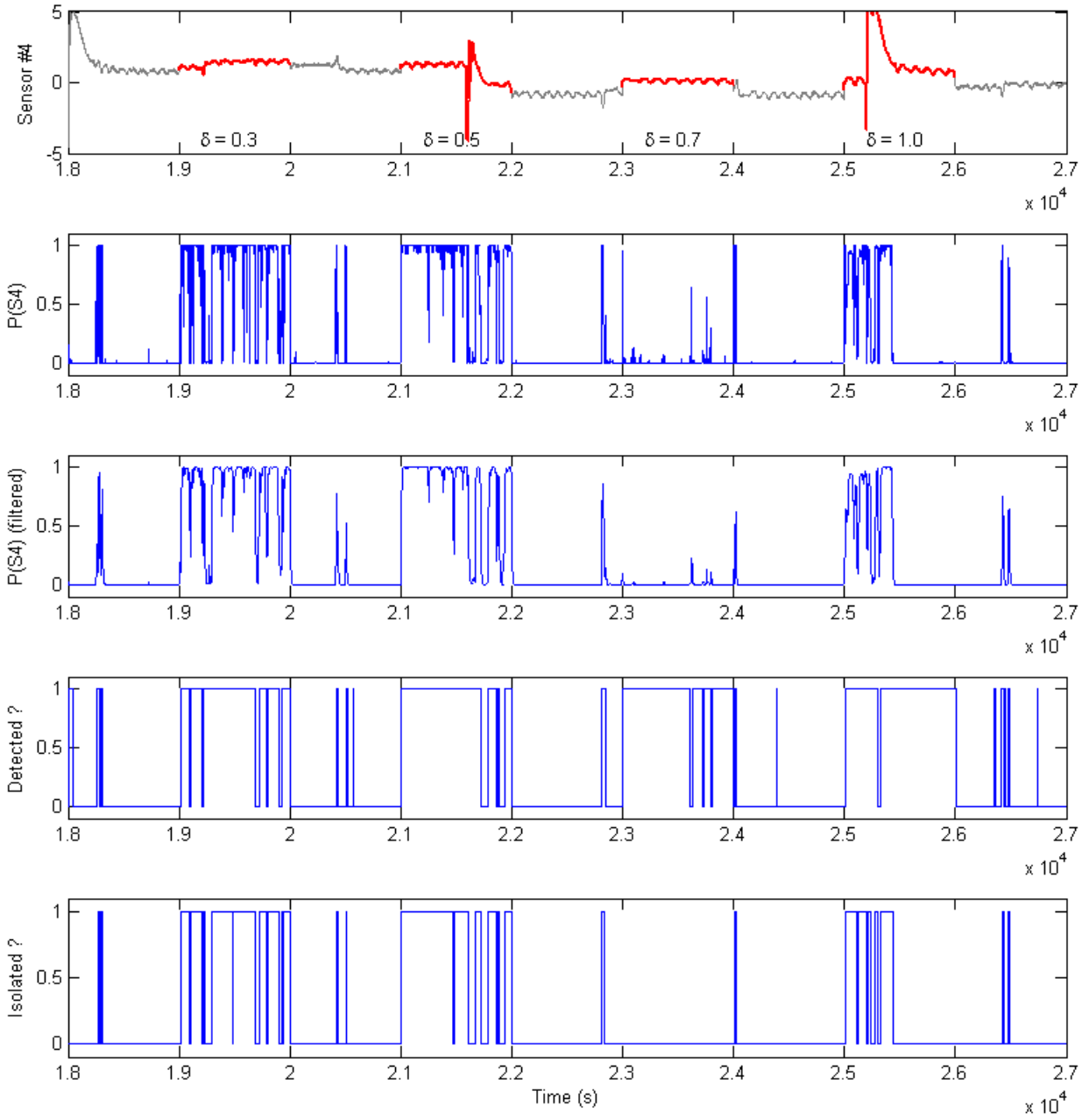

C Error
P(S3)
P(S3) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

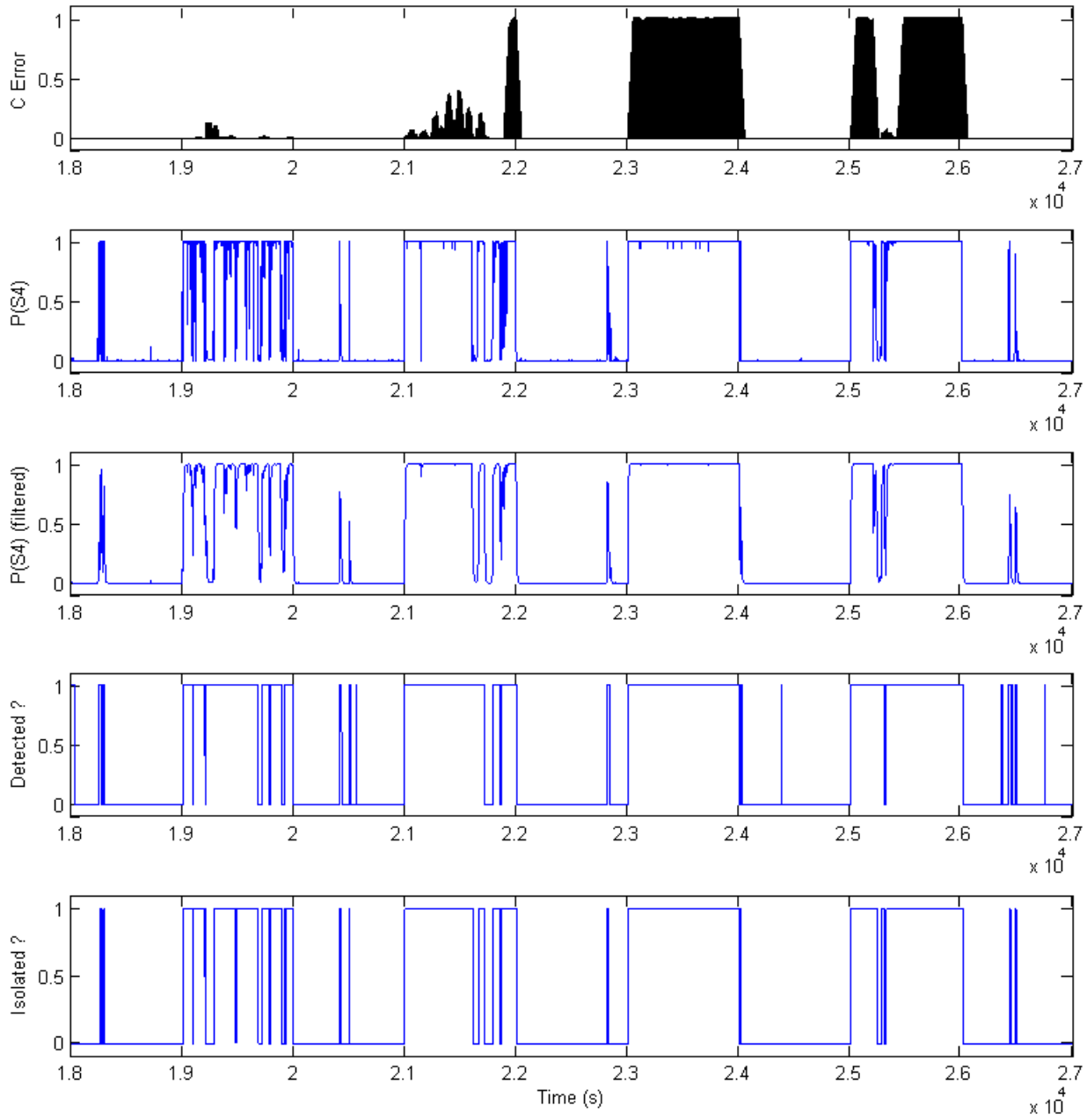

Sensor #4
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S4)
P(S4) (filtered)
Detected ?
Isolated ?
Time (s)

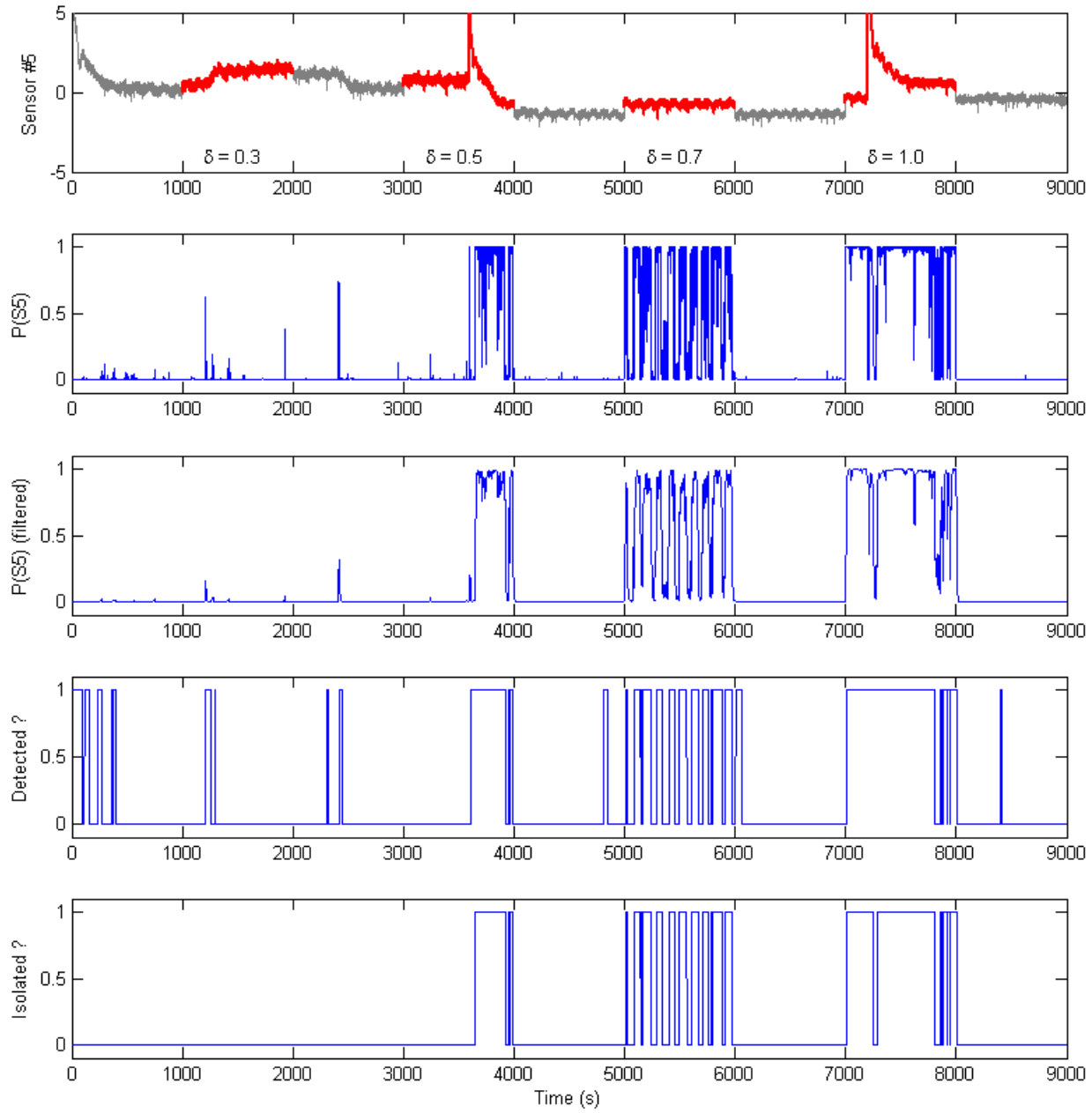

C Error
P(S4)
P(S4) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

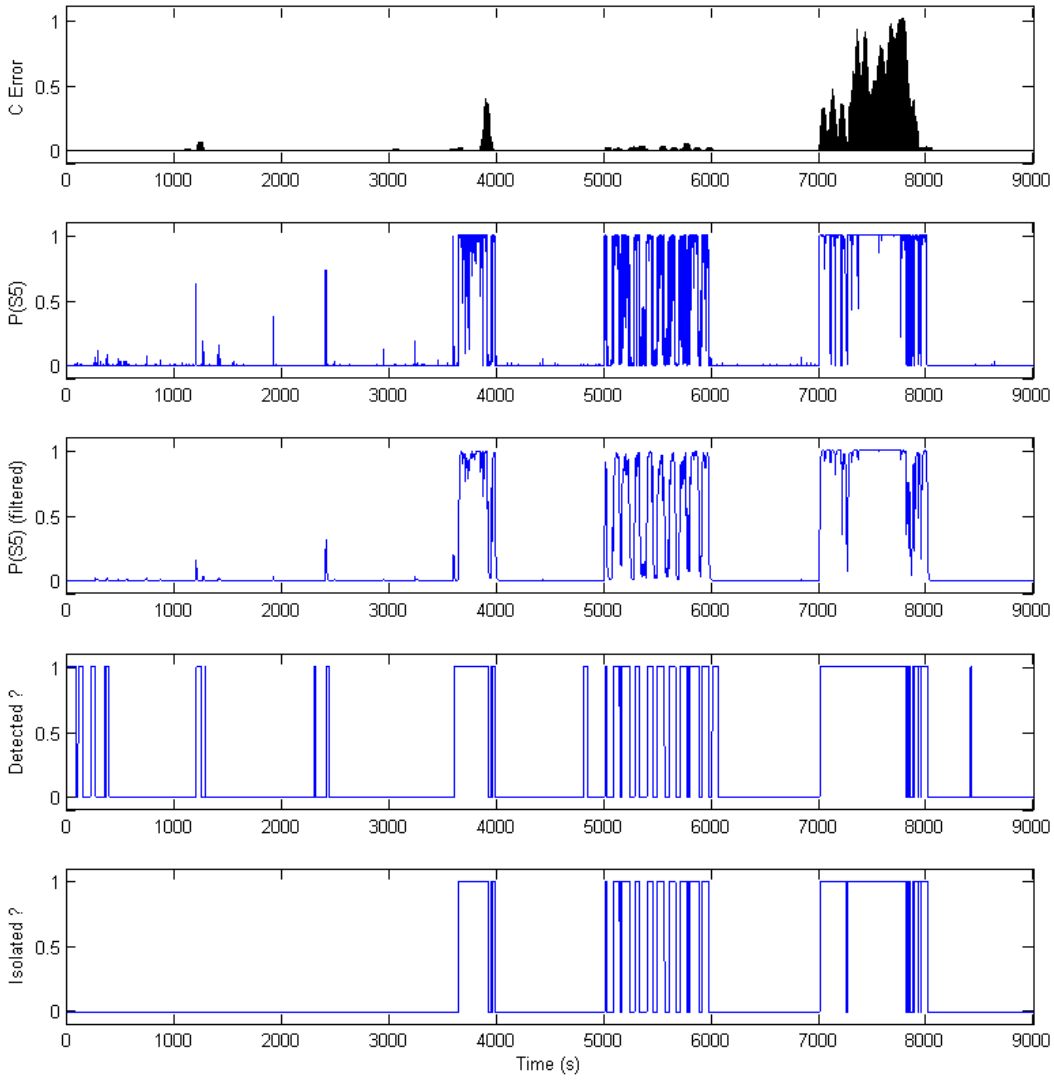

Sensor #4
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S4)
P(S4) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

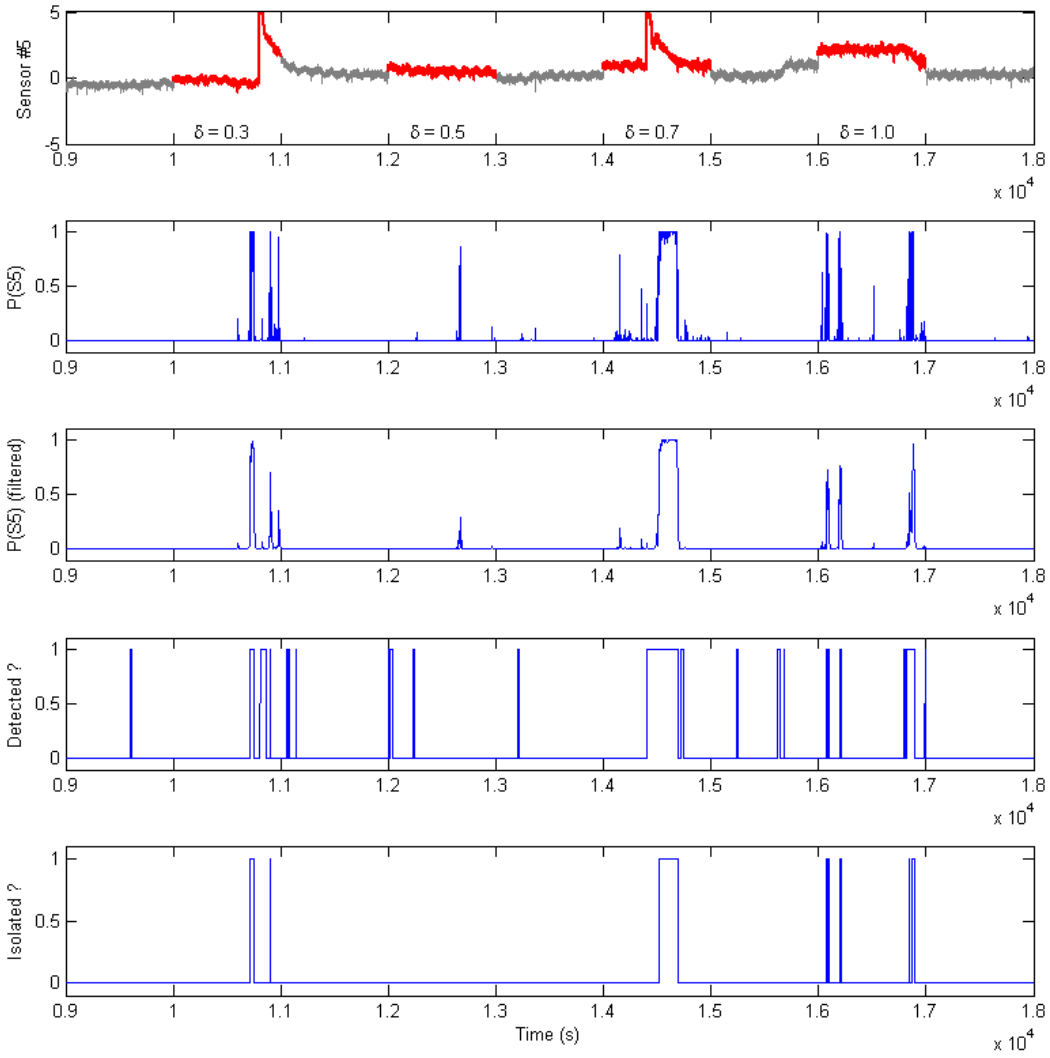

C Error
P(S4)
P(S4) (filtered)
Detected ?
Isolated ?
0
0.5
1
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
x 10^4
Time (s)

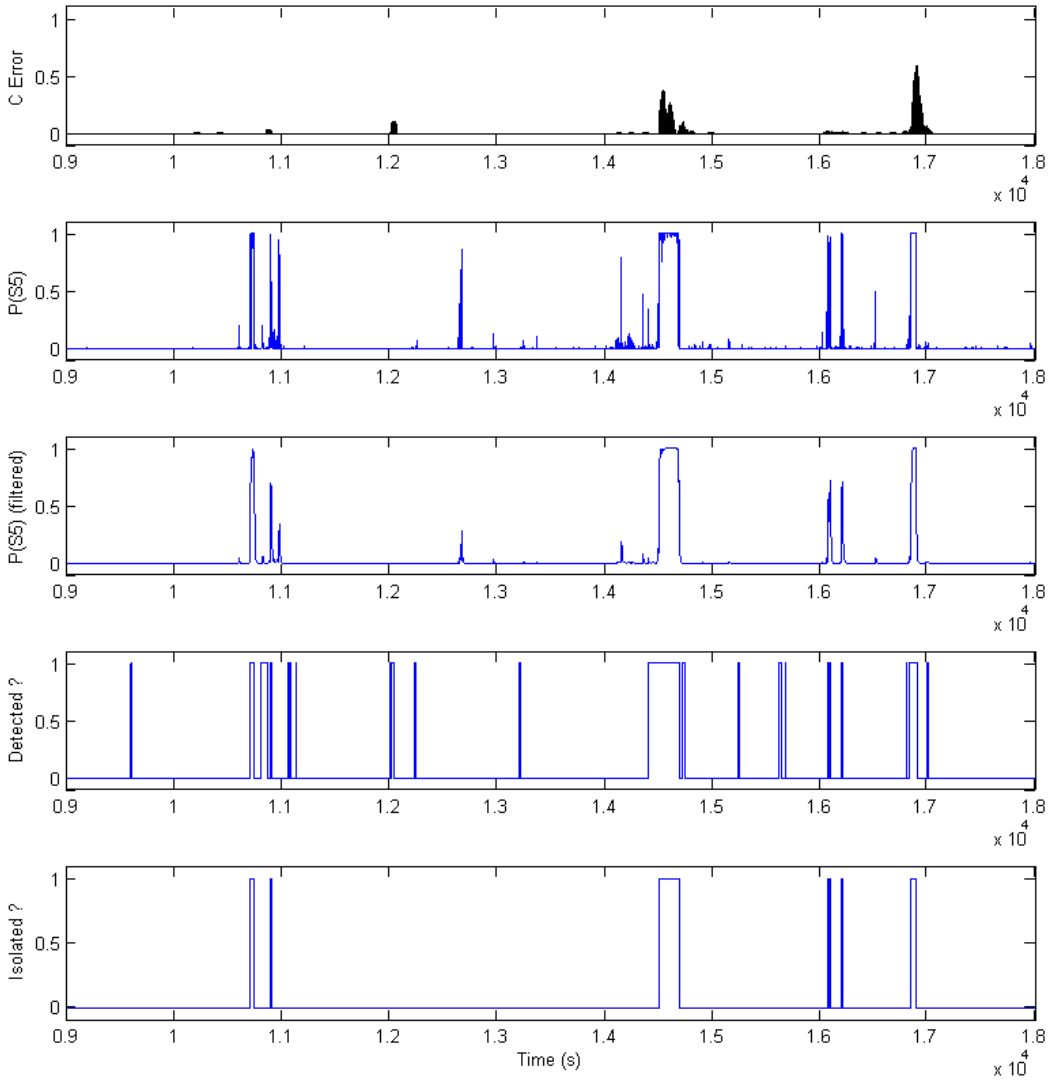

Sensor #4
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S4)
P(S4) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

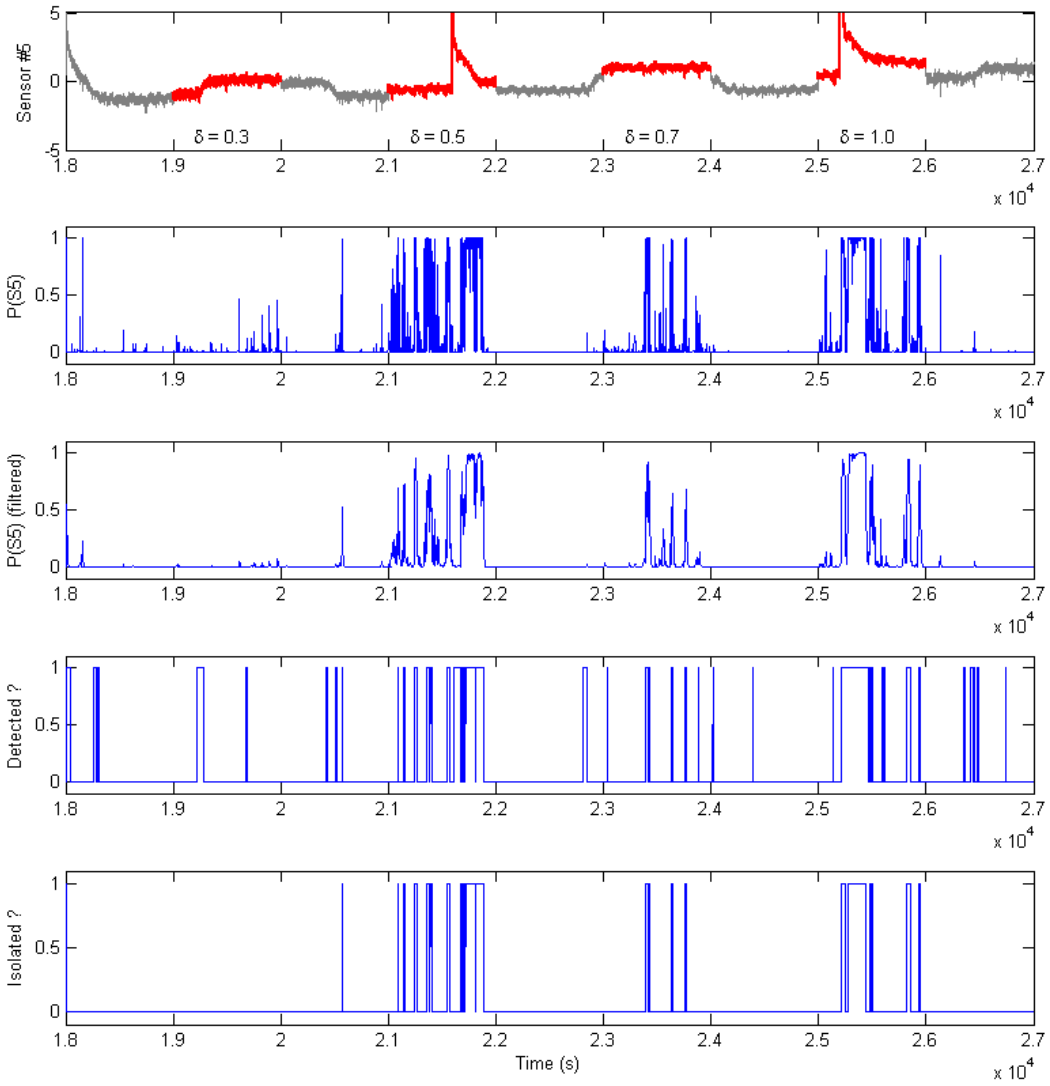

C Error
P(S4)
P(S4) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

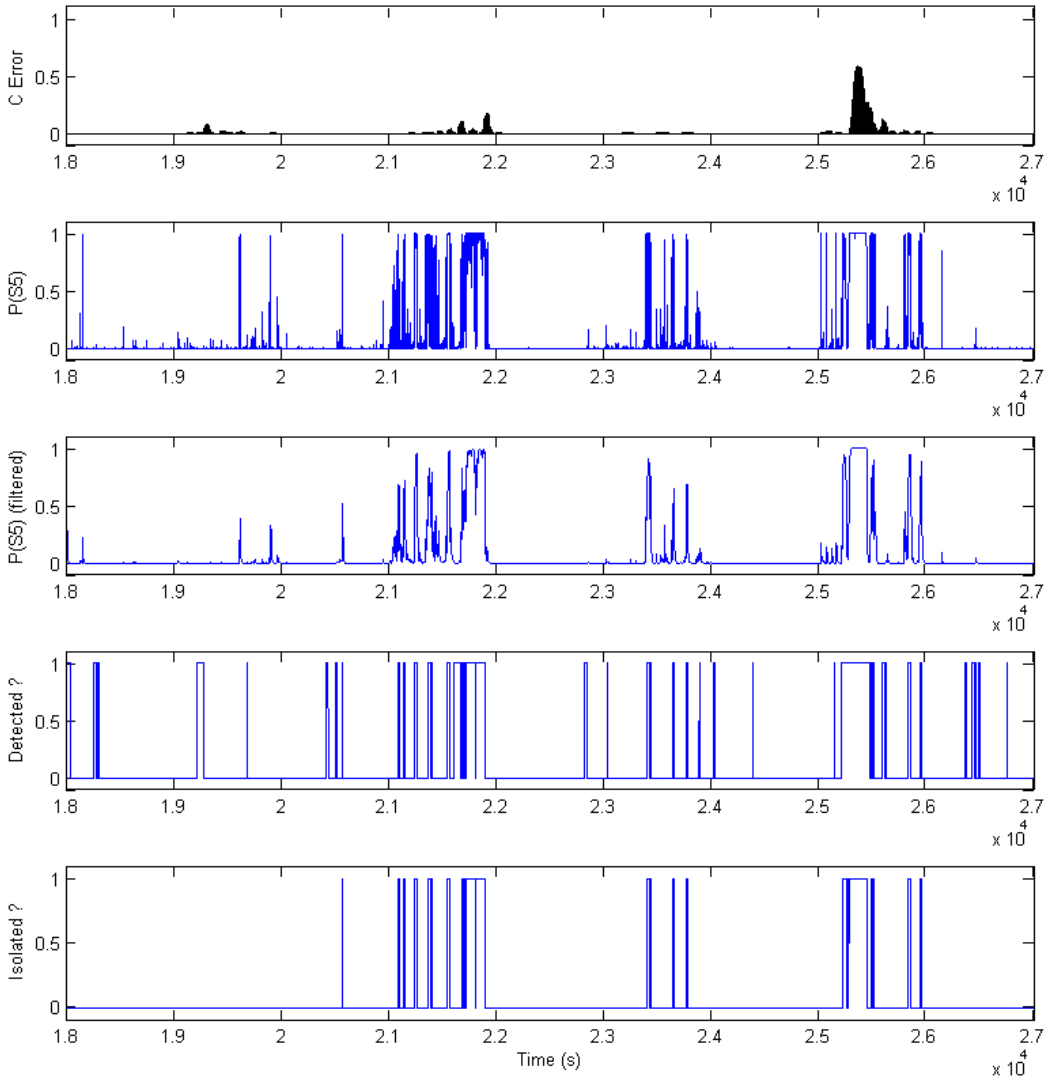

Sensor #5
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S5)
P(S5) (filtered)
Detected ?
Isolated ?
Time (s)

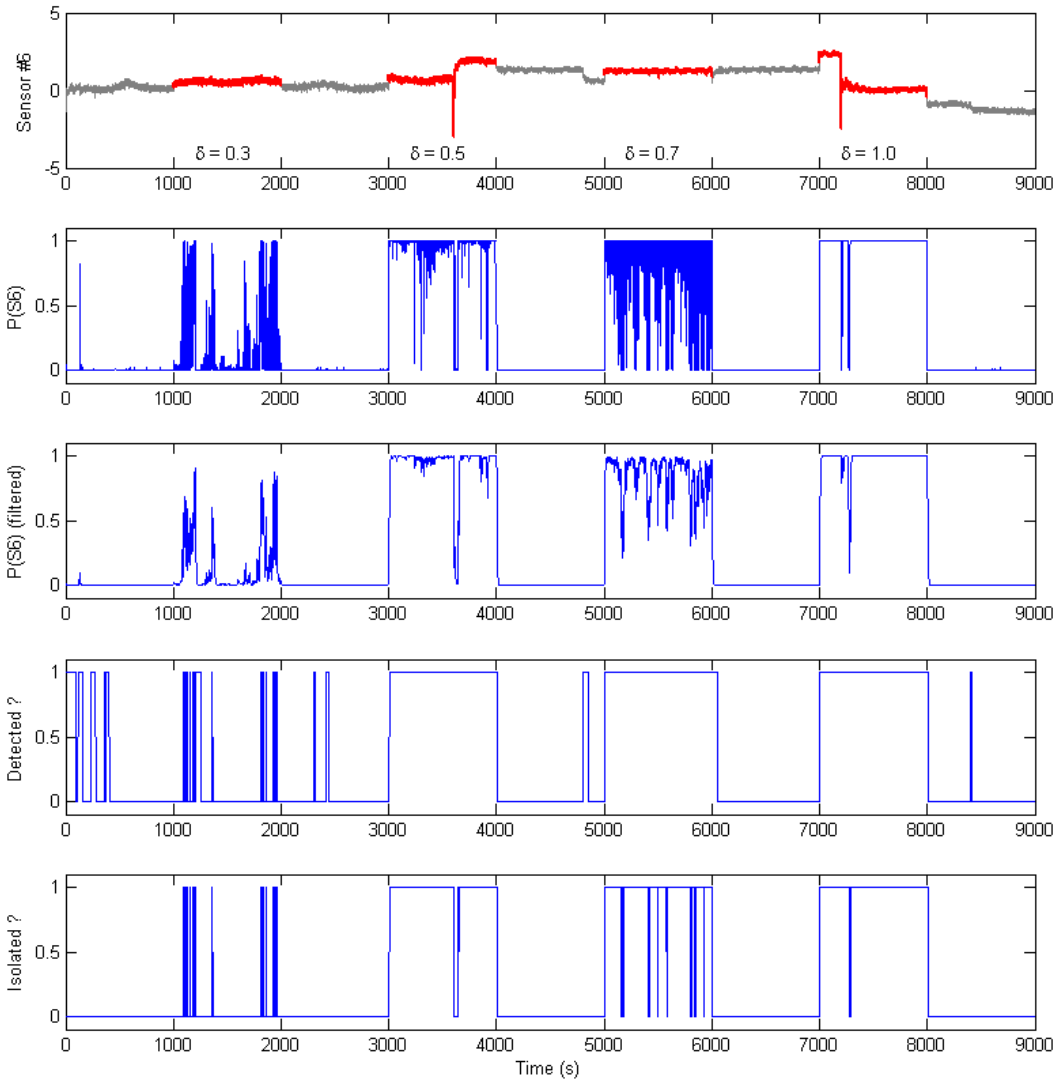

C Error
P(S5)
P(S5) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

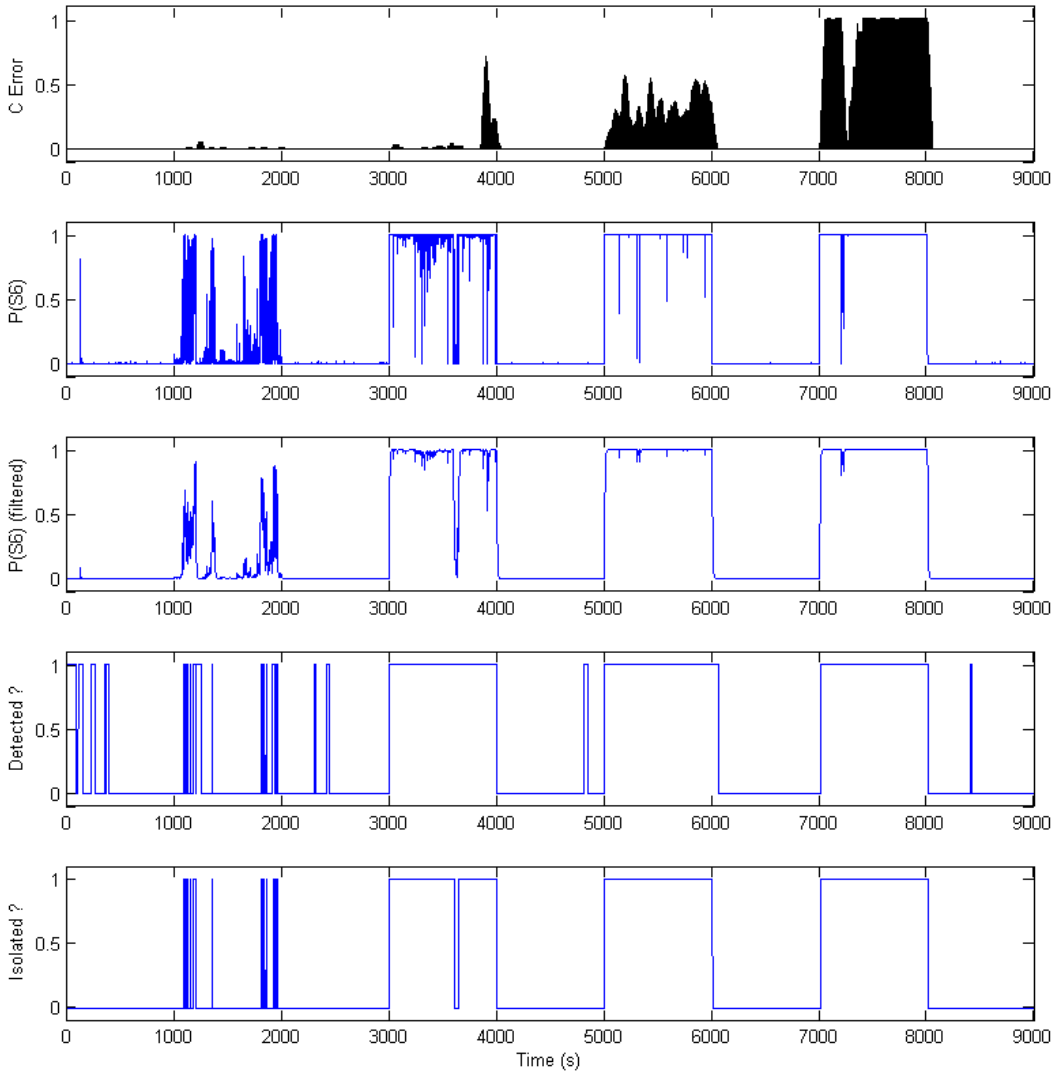

Sensor #5
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S5)
P(S5) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

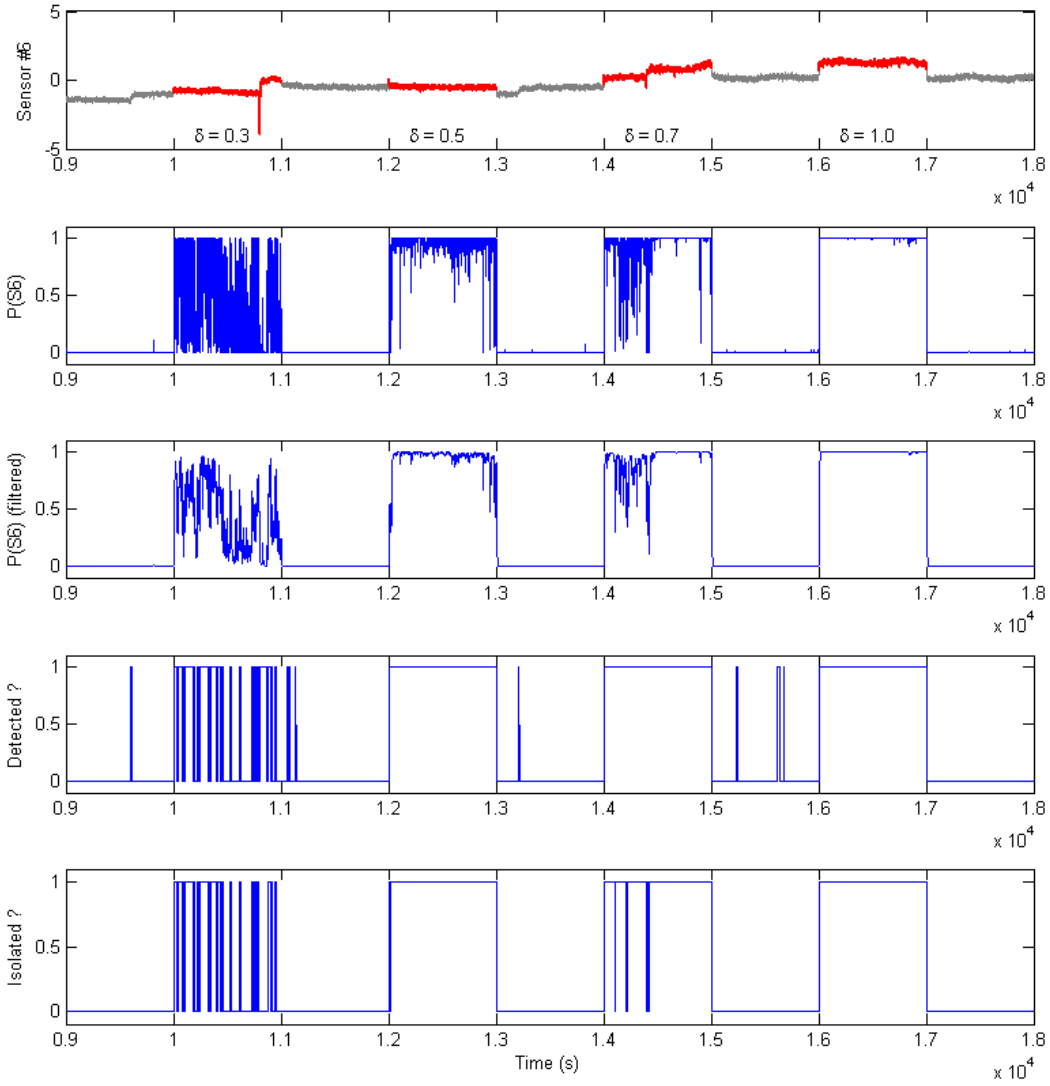

C Error
P(S5)
P(S5) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

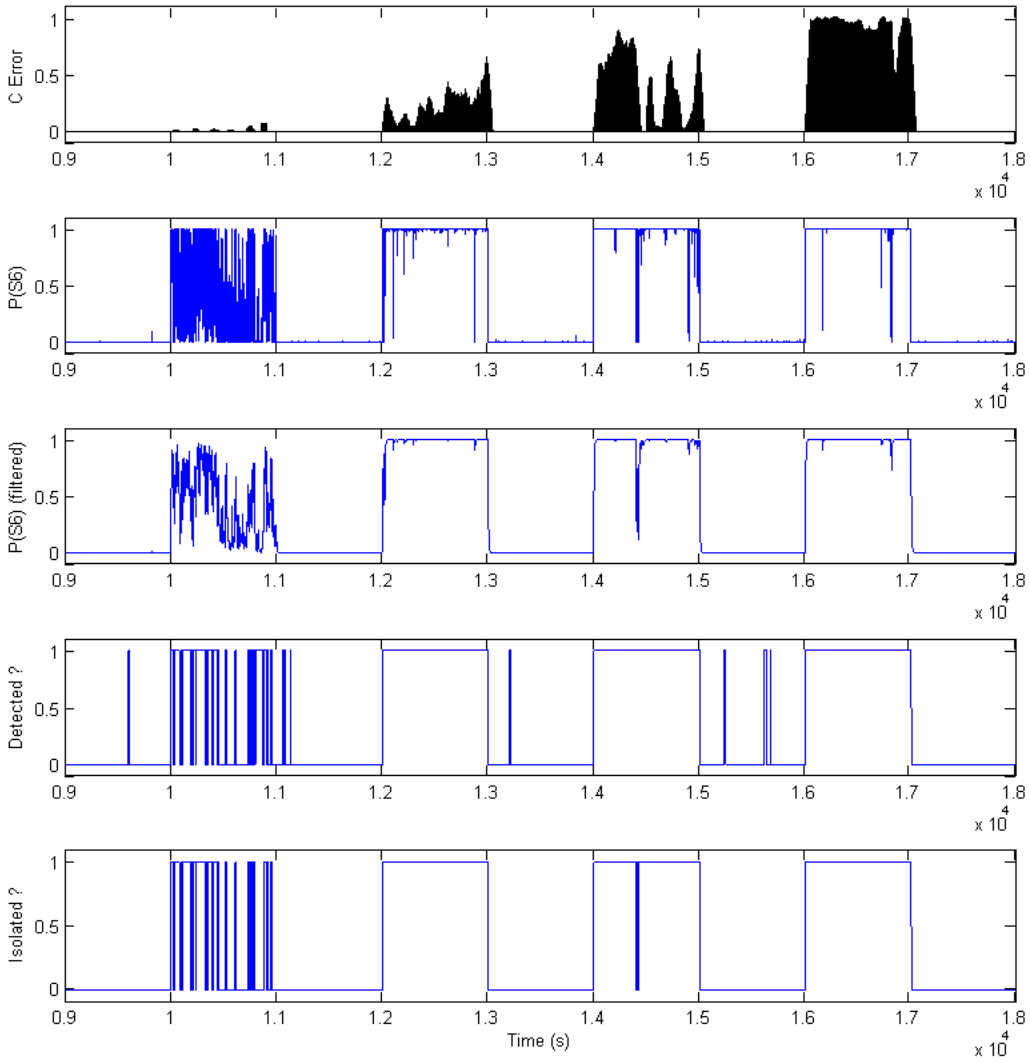

Sensor #5
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S5)
P(S5) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

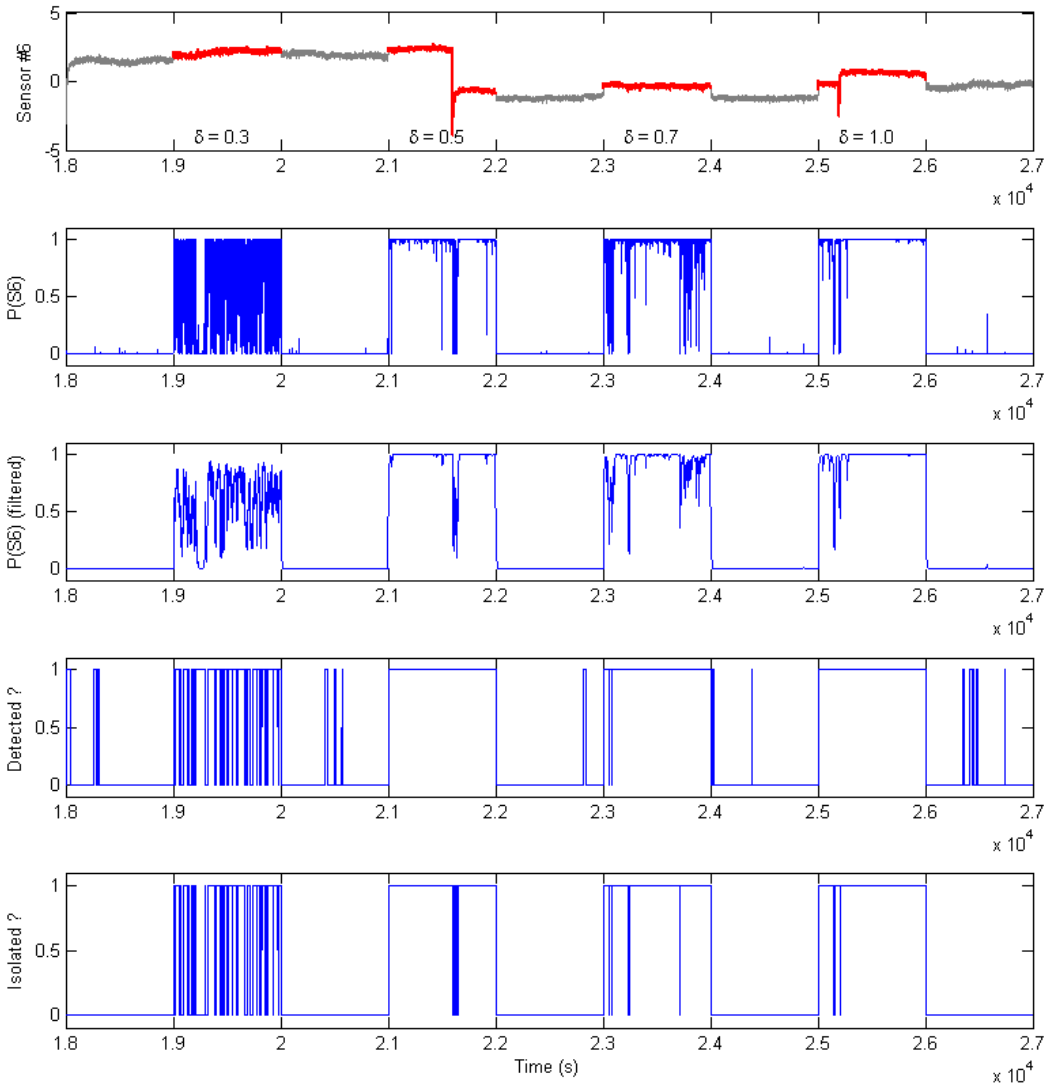

C Error
P(S5)
P(S5) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

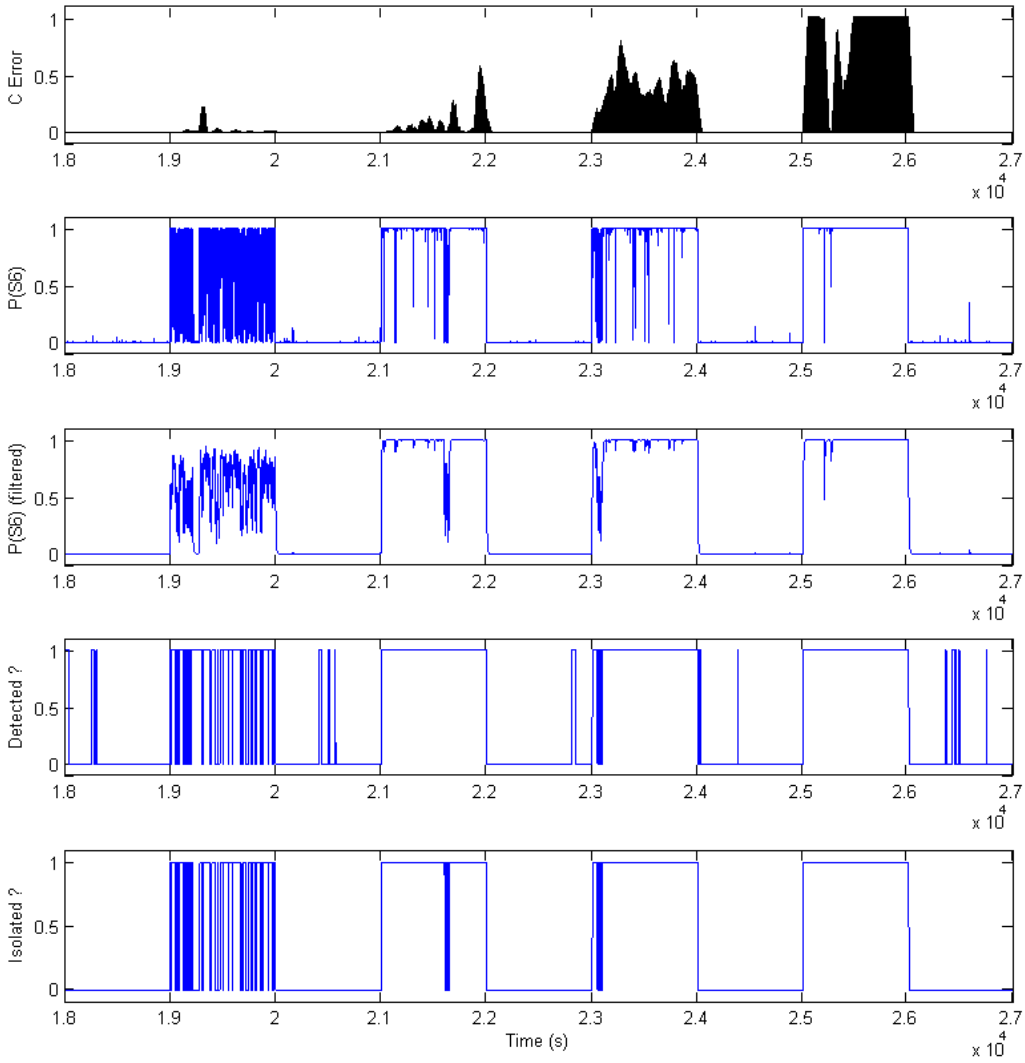

Sensor #6
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S6)
P(S6) (filtered)
Detected ?
Isolated ?
Time (s)

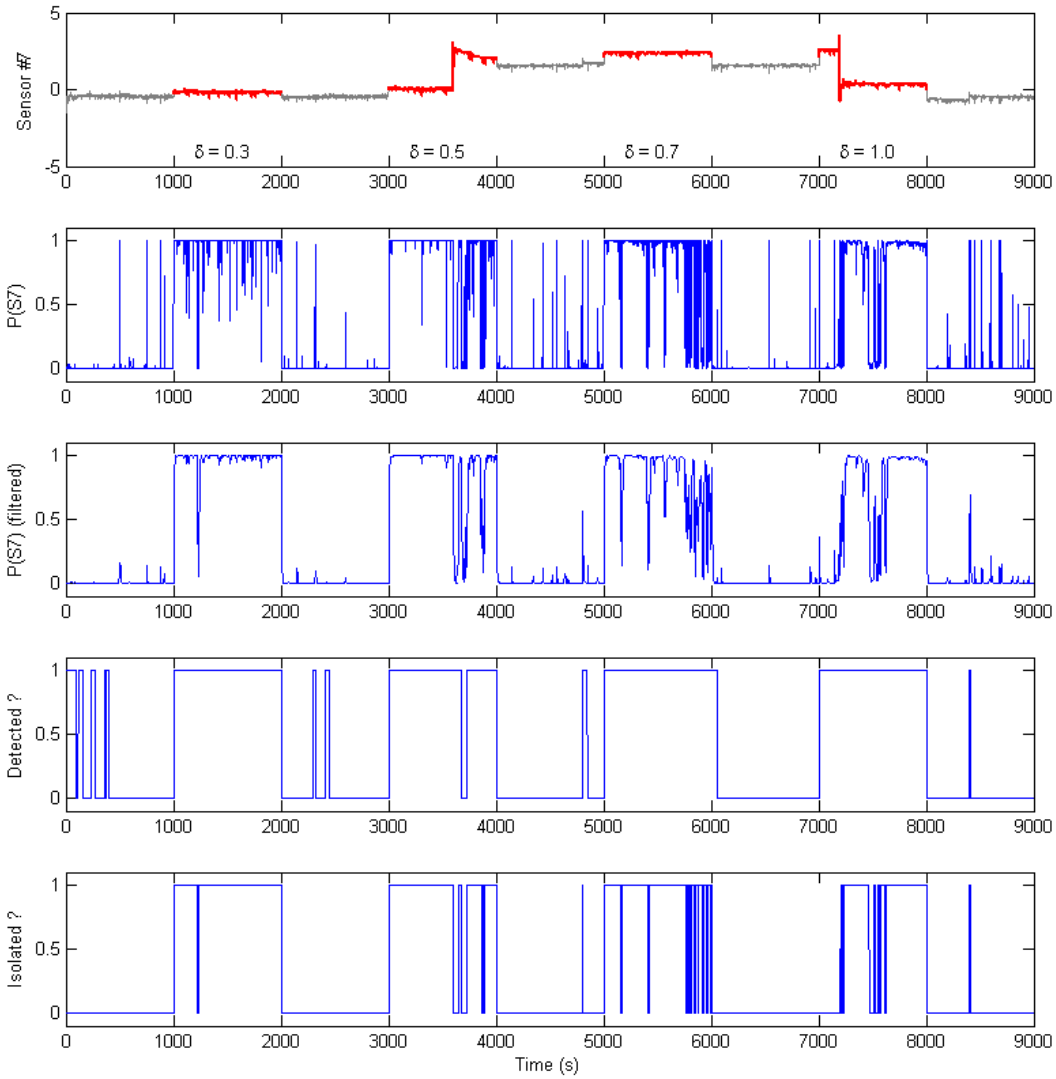

C Error
P(S6)
P(S6) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

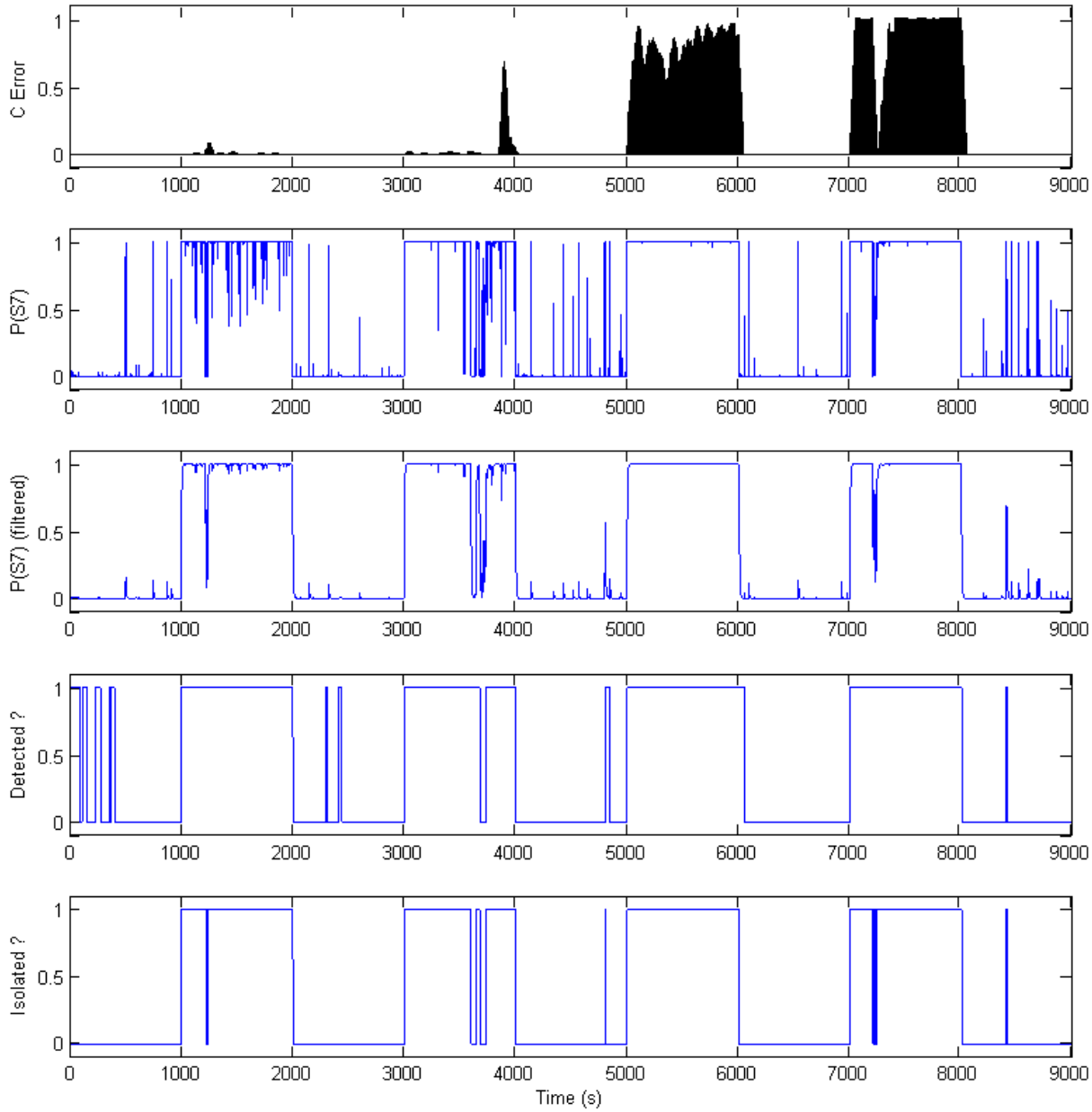

Sensor #6
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S6)
P(S6) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

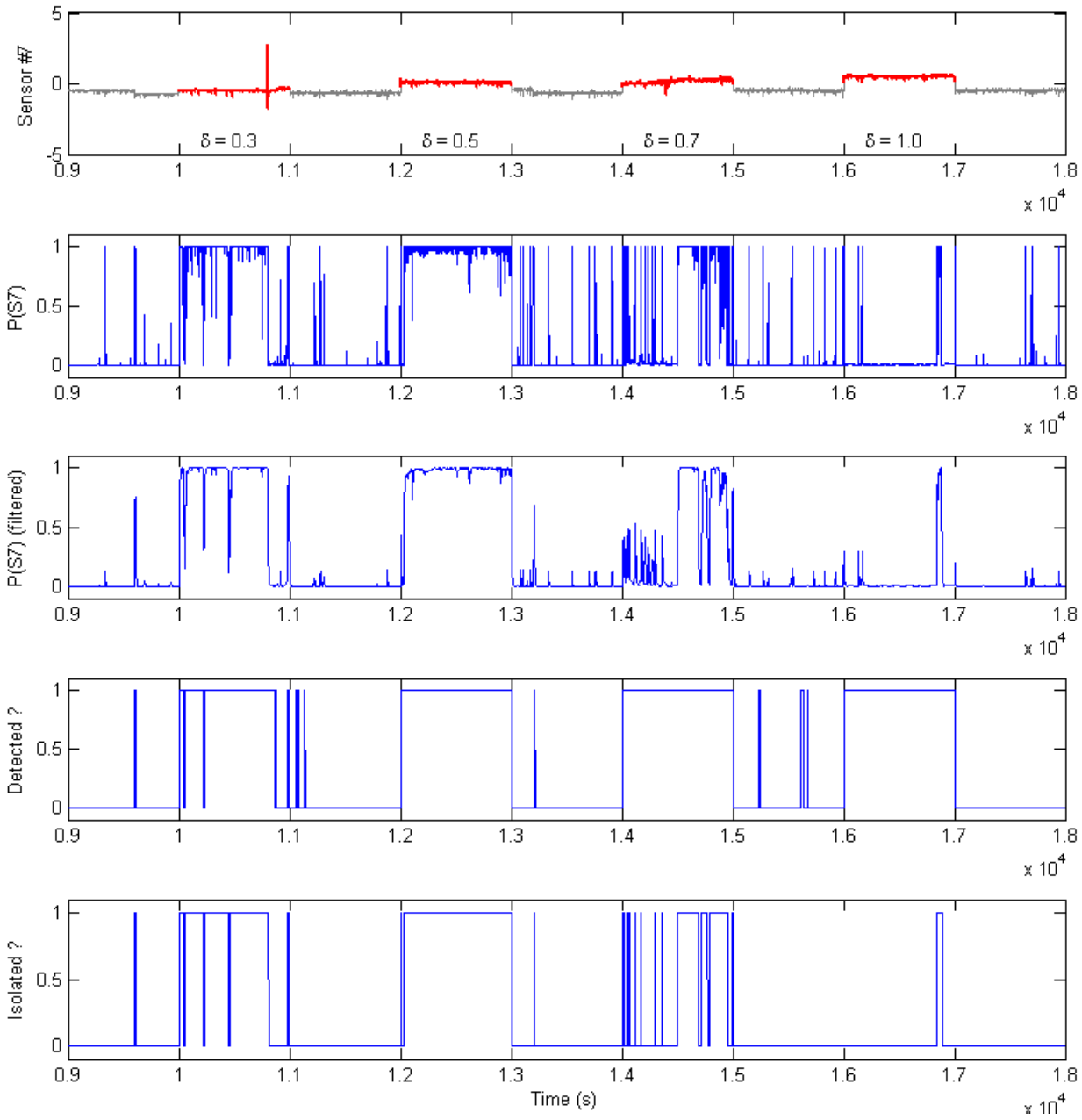

C Error
P(S6)
P(S6) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

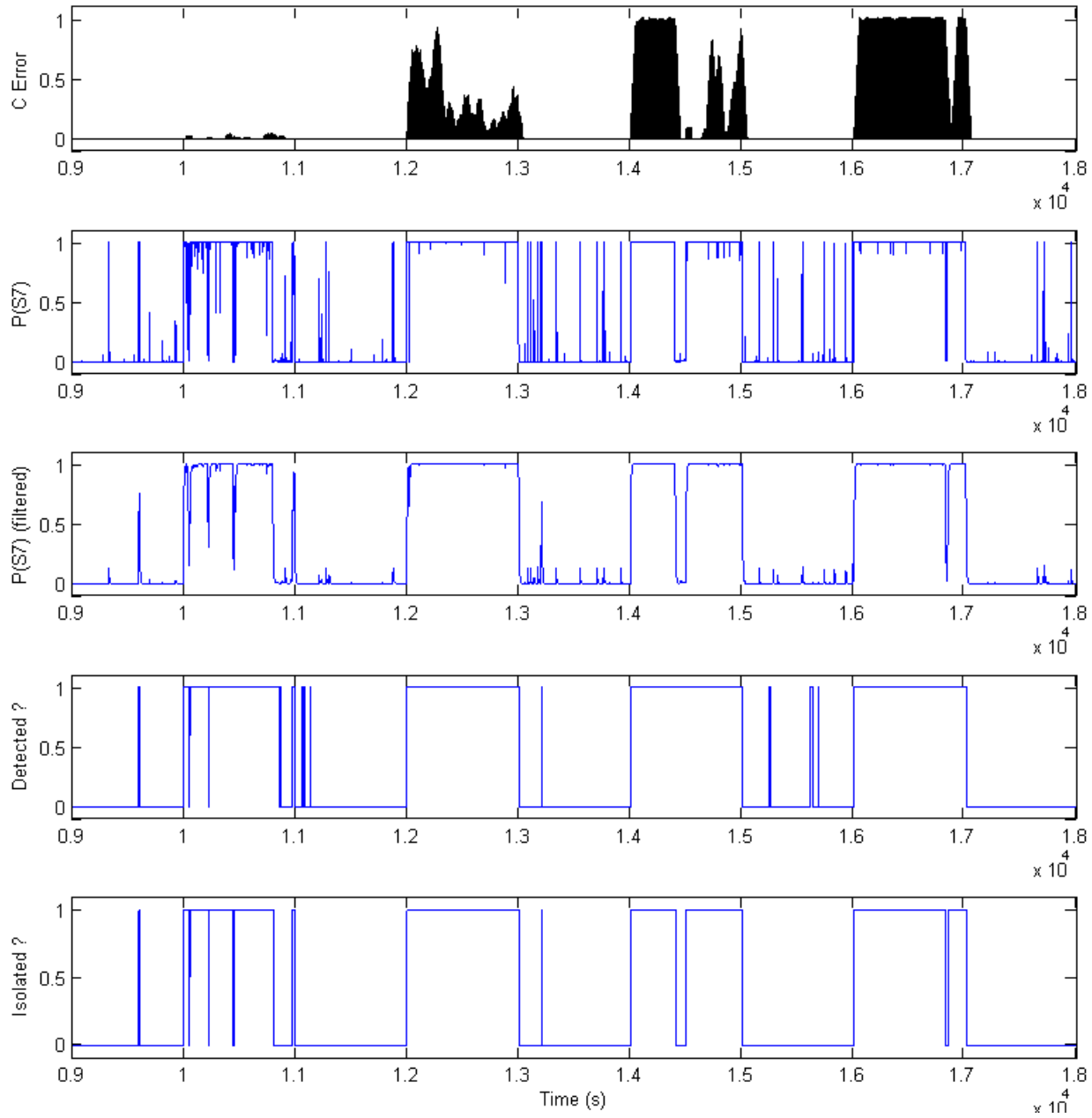

Sensor #6
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S6)
P(S6) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

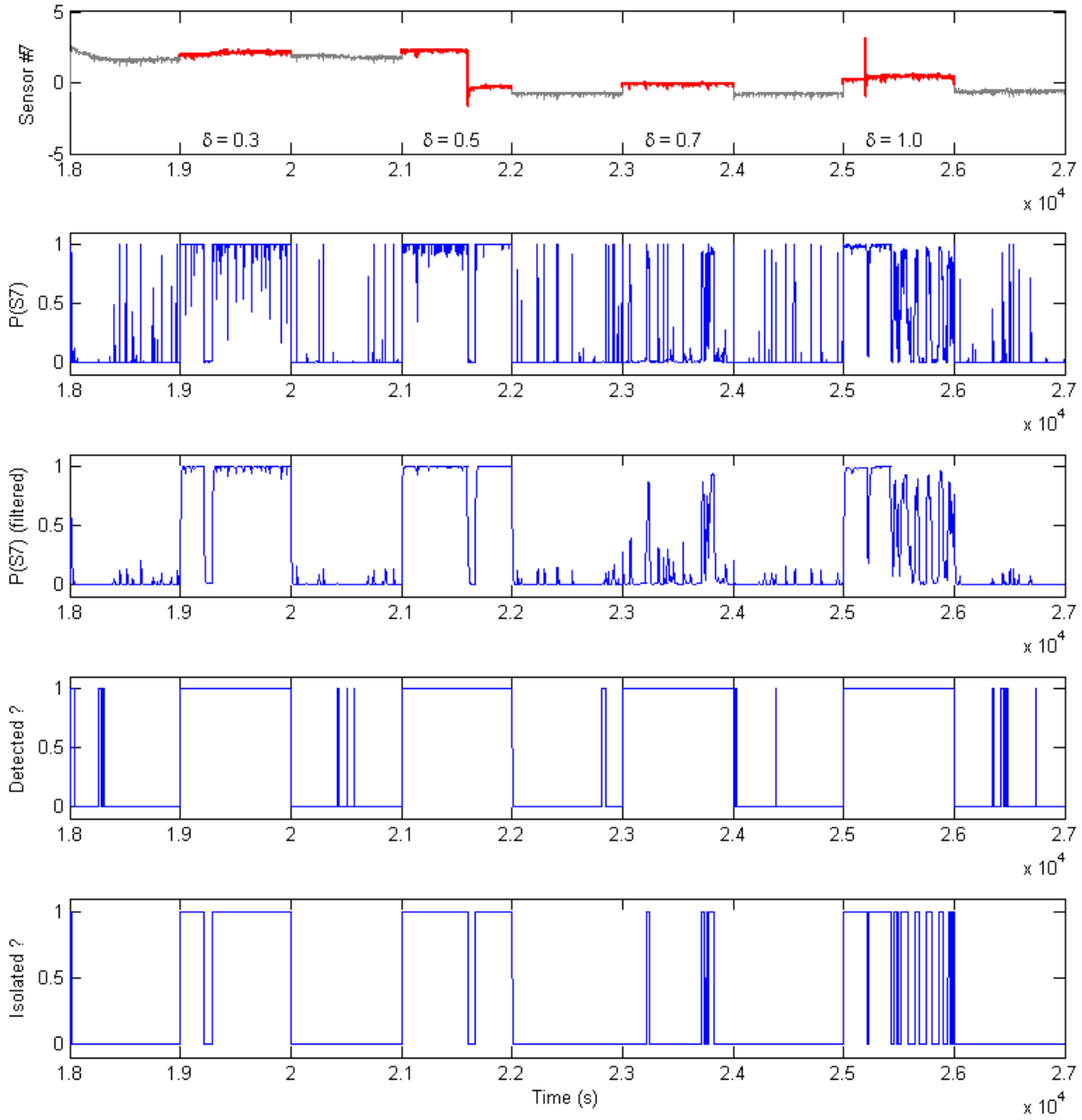

C Error
P(S6)
P(S6) (filtered)
Detected ?
Isolated ?
1
0.5
0
1.8
1.9
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7
x 10^4
Time (s)

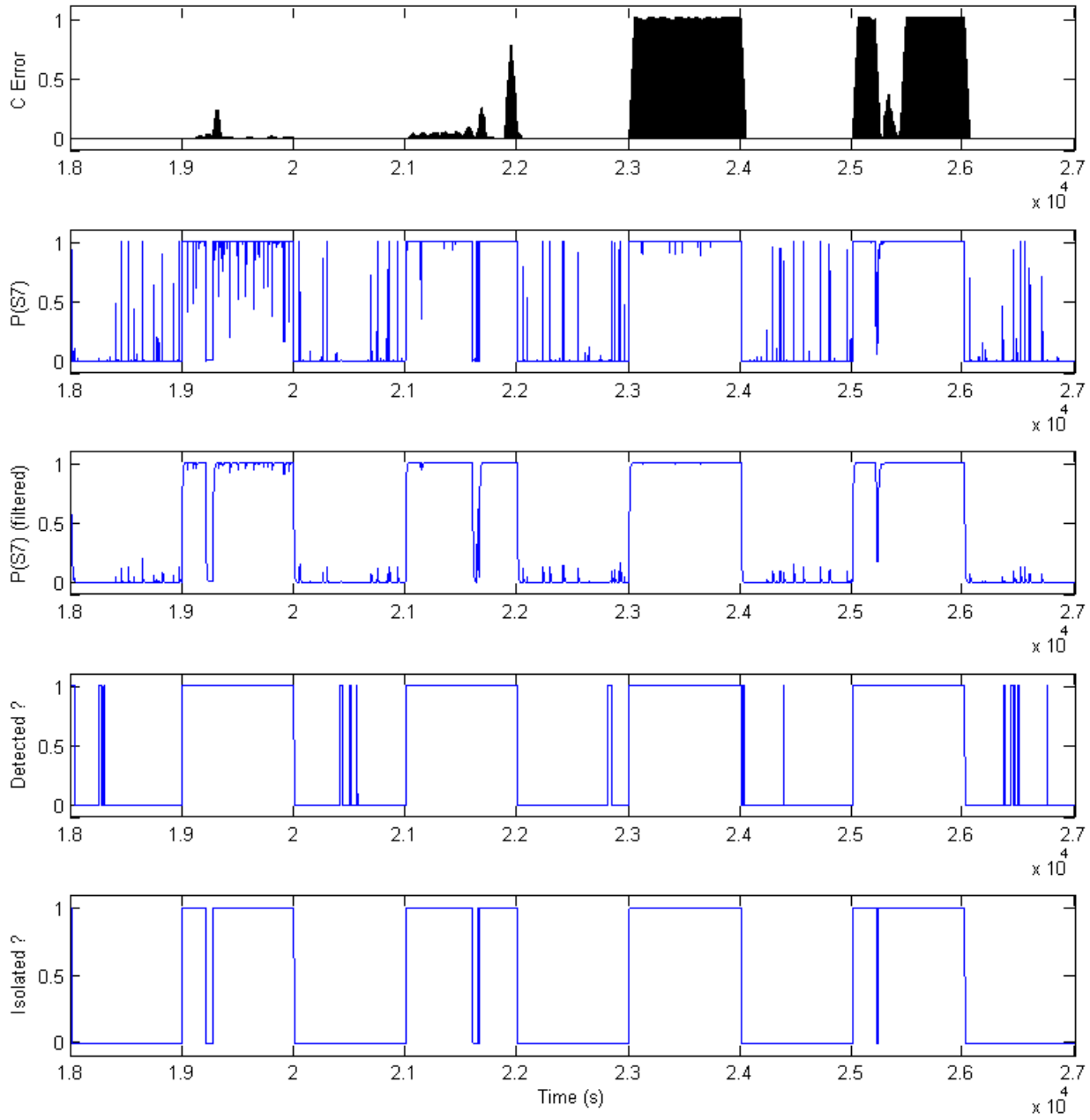

Sensor #7
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S7)
P(S7) (filtered)
Detected ?
Isolated ?
Time (s)

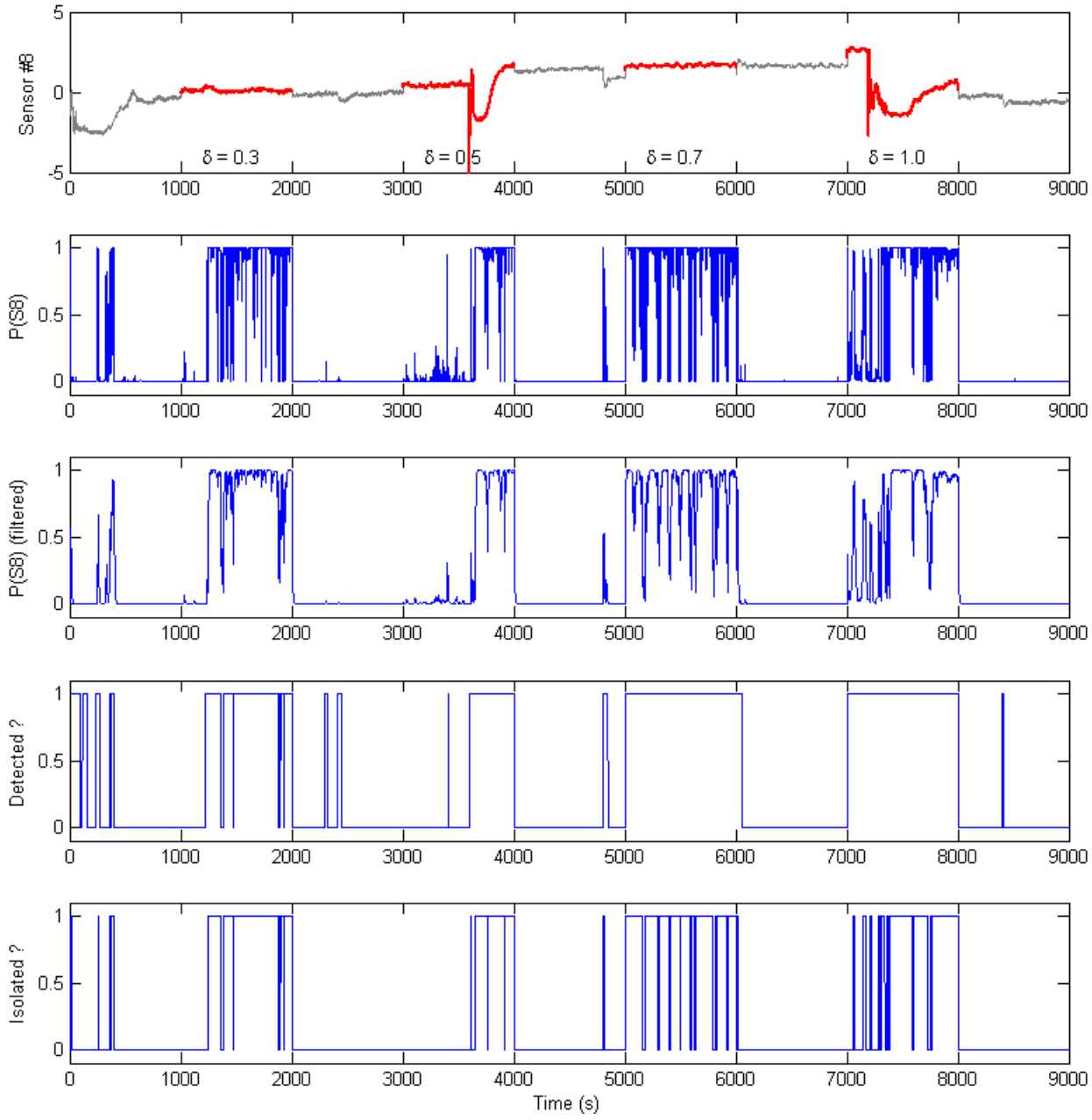

C Error
P(S7)
P(S7) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

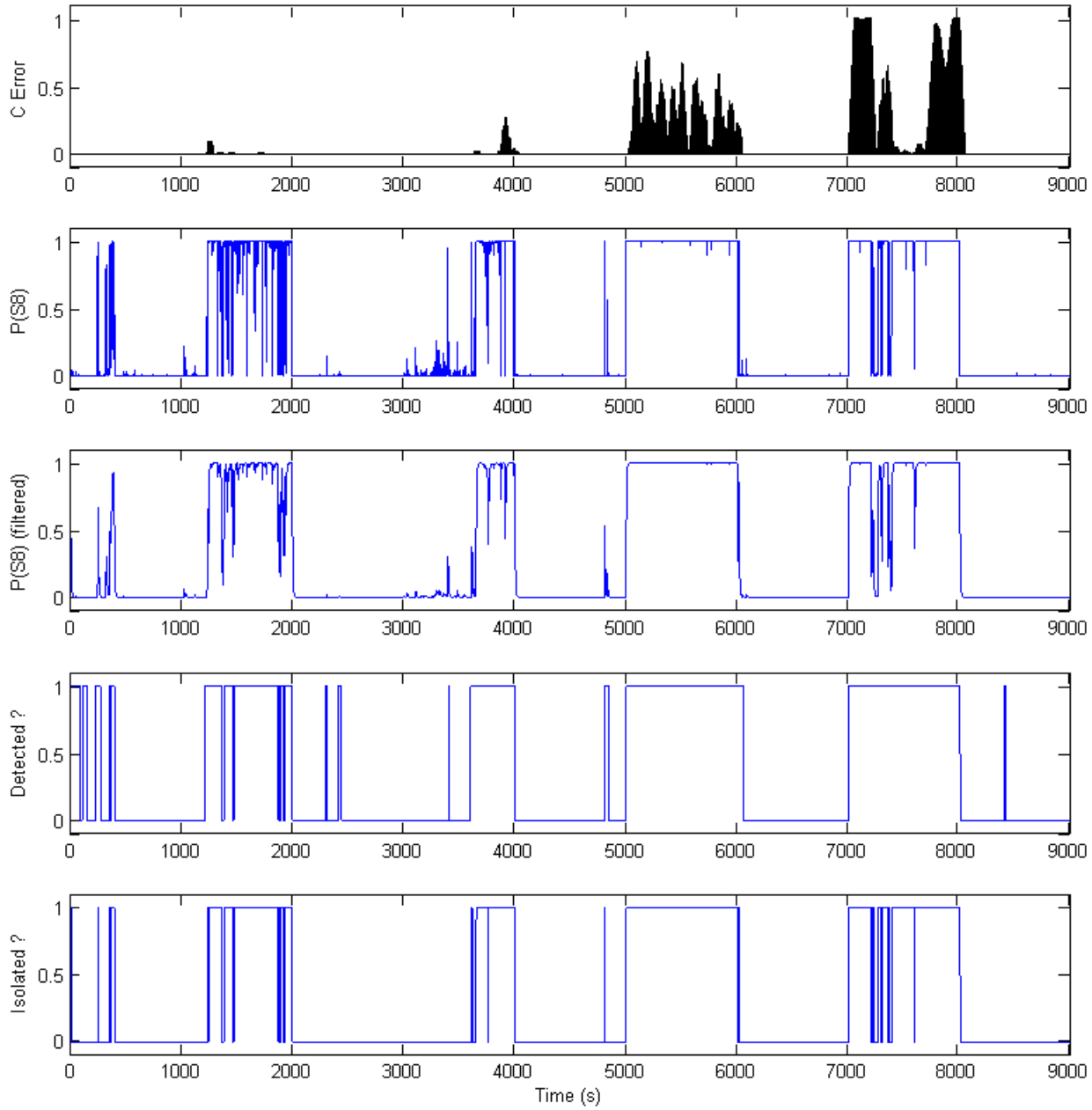

Sensor #7
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S7)
P(S7) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

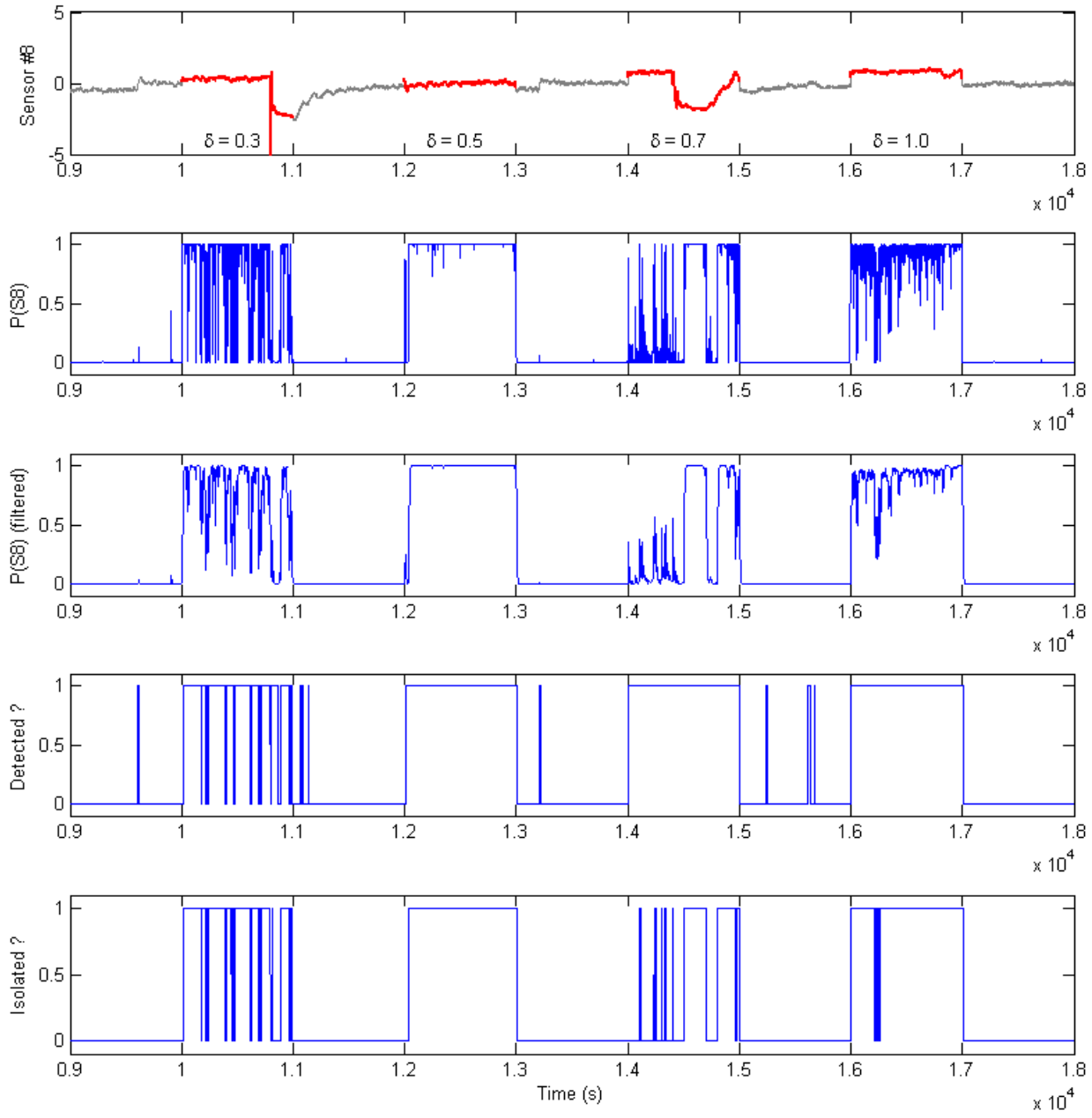

C Error
P(S7)
P(S7) (filtered)
Detected ?
Isolated ?
0
0.5
1
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
x 10^4
Time (s)

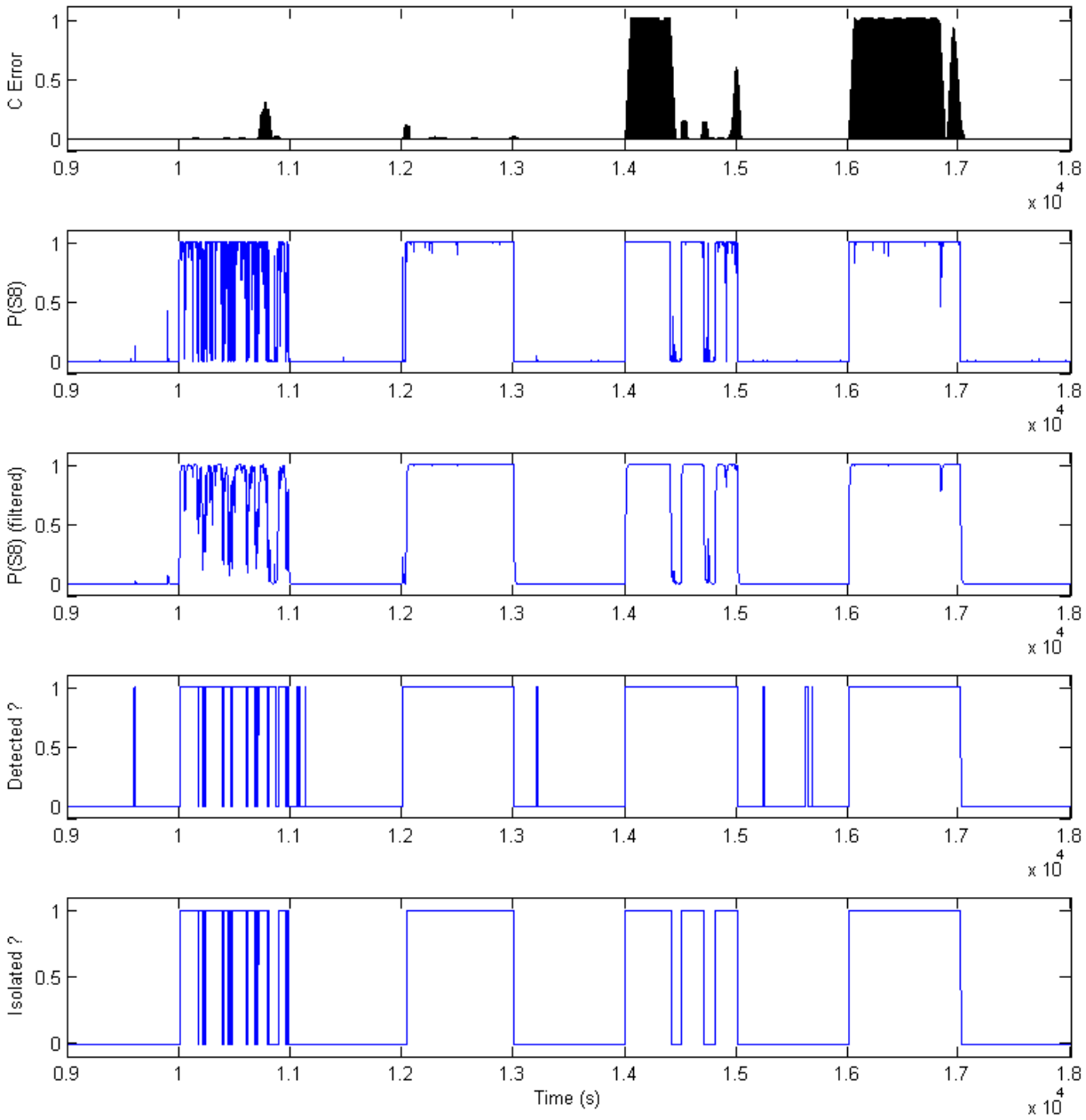

Sensor #7
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S7)
P(S7) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

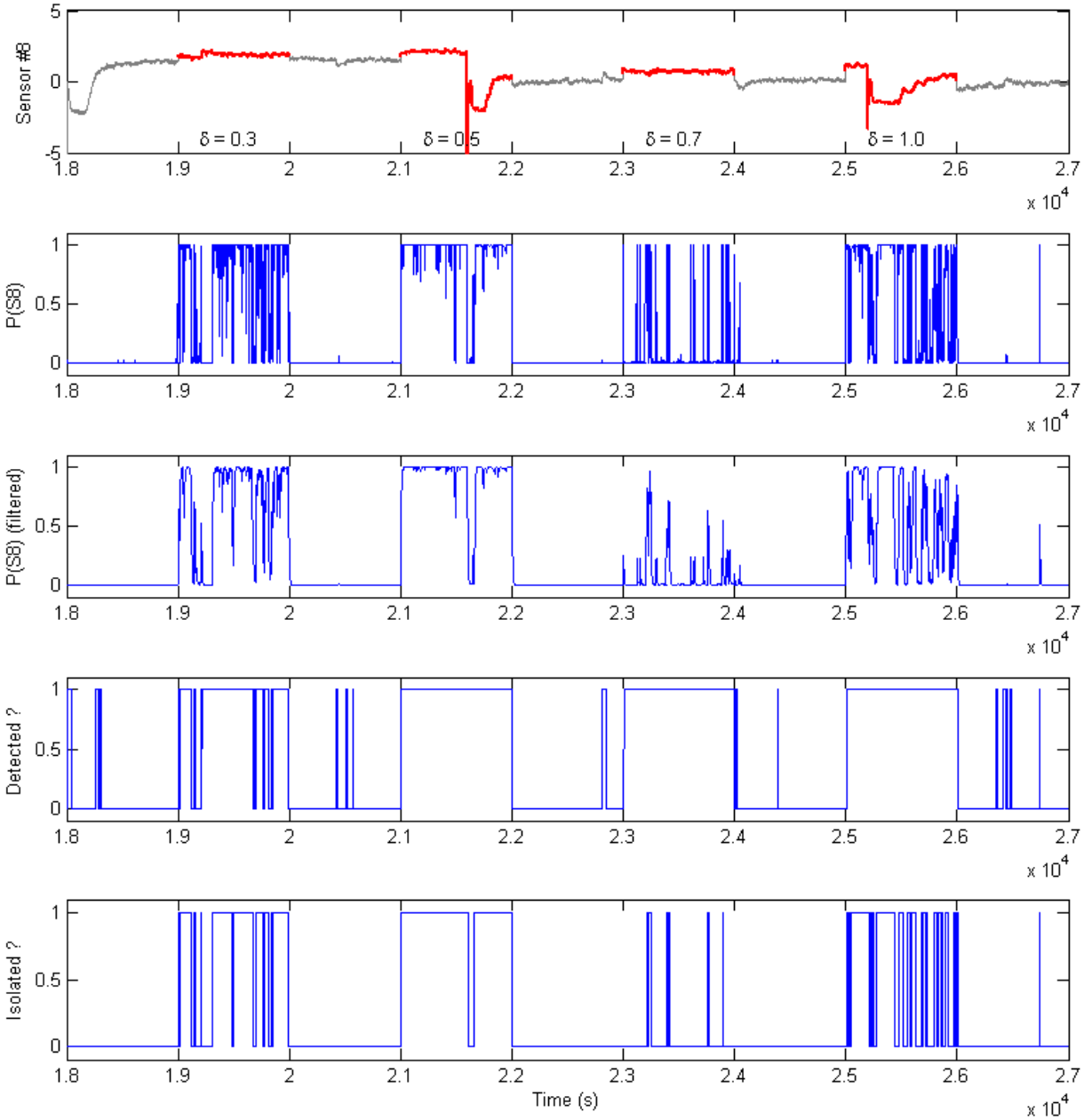

C Error
P(S7)
P(S7) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

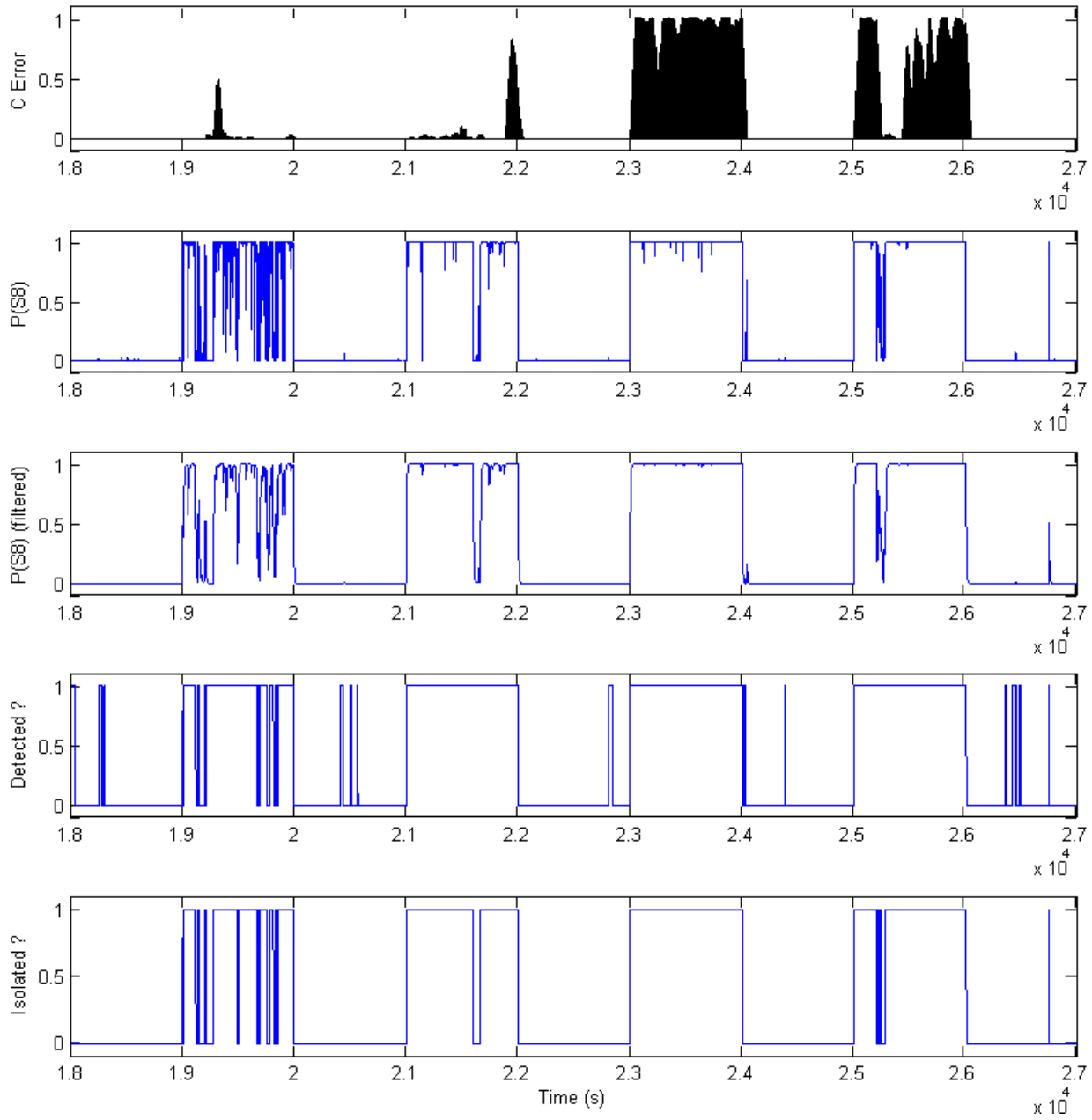

Sensor #8
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S8)
P(S8) (filtered)
Detected ?
Isolated ?
Time (s)

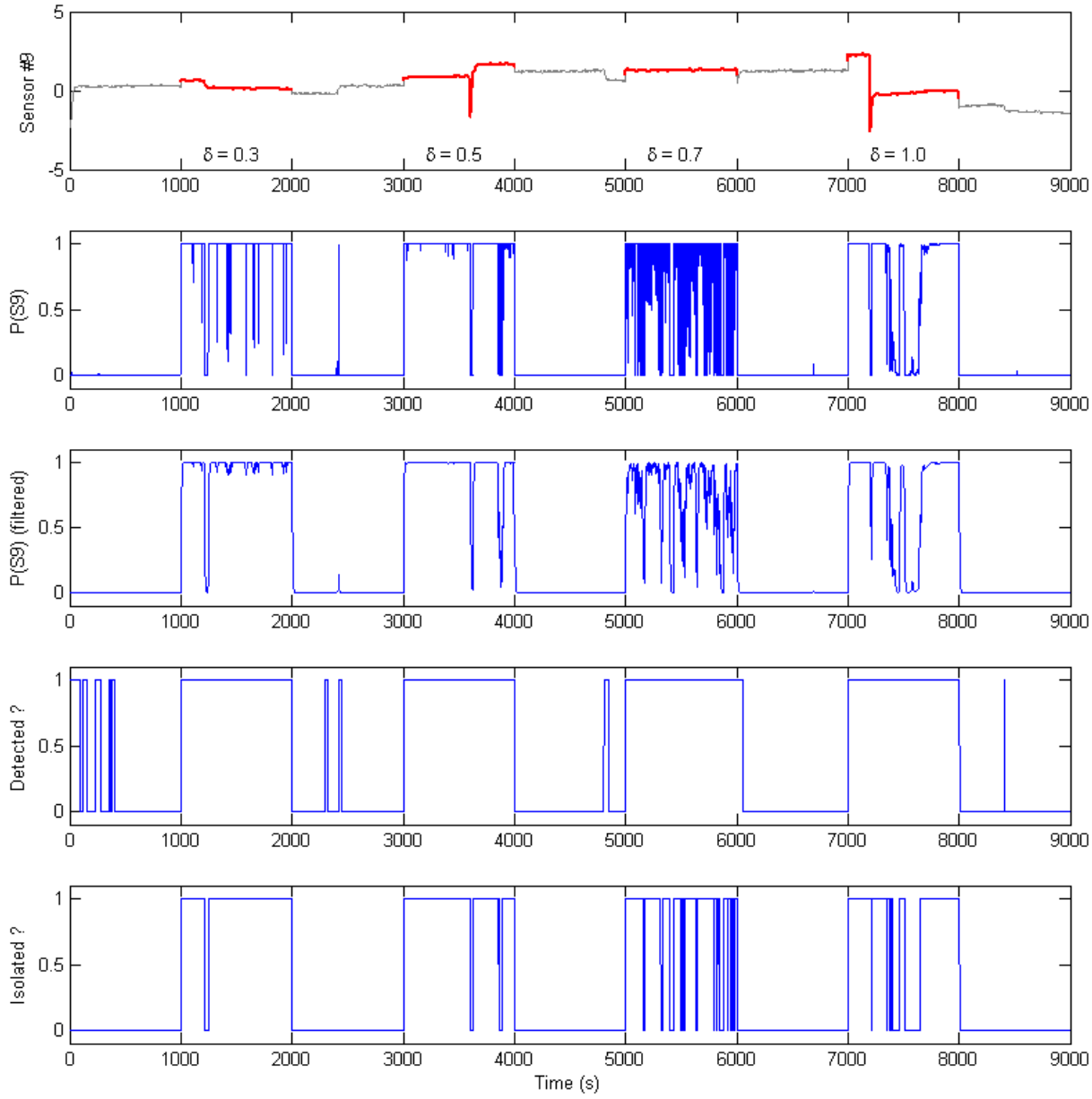

C Error
P(S8)
P(S8) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

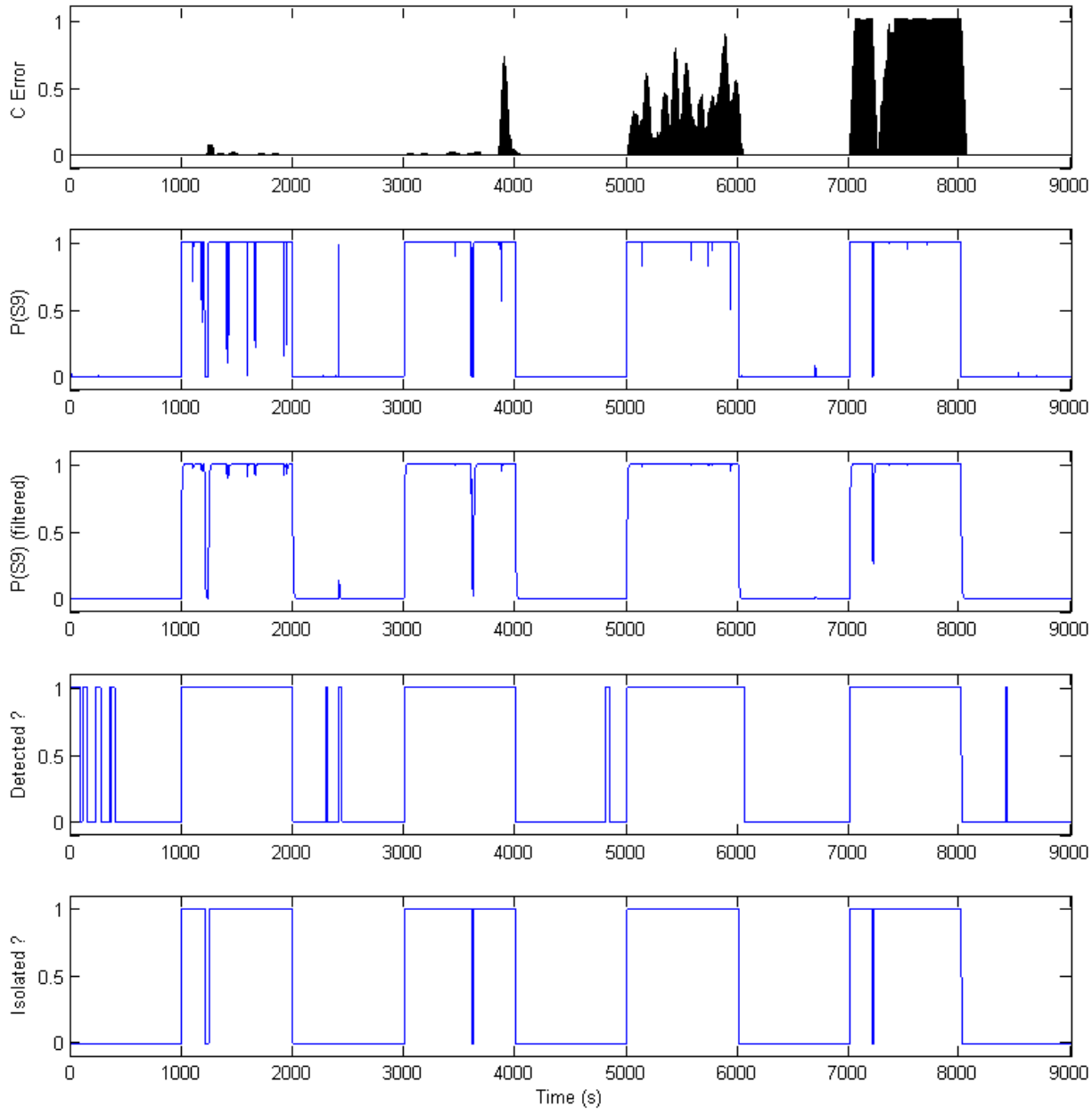

Sensor #8
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S8)
P(S8) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

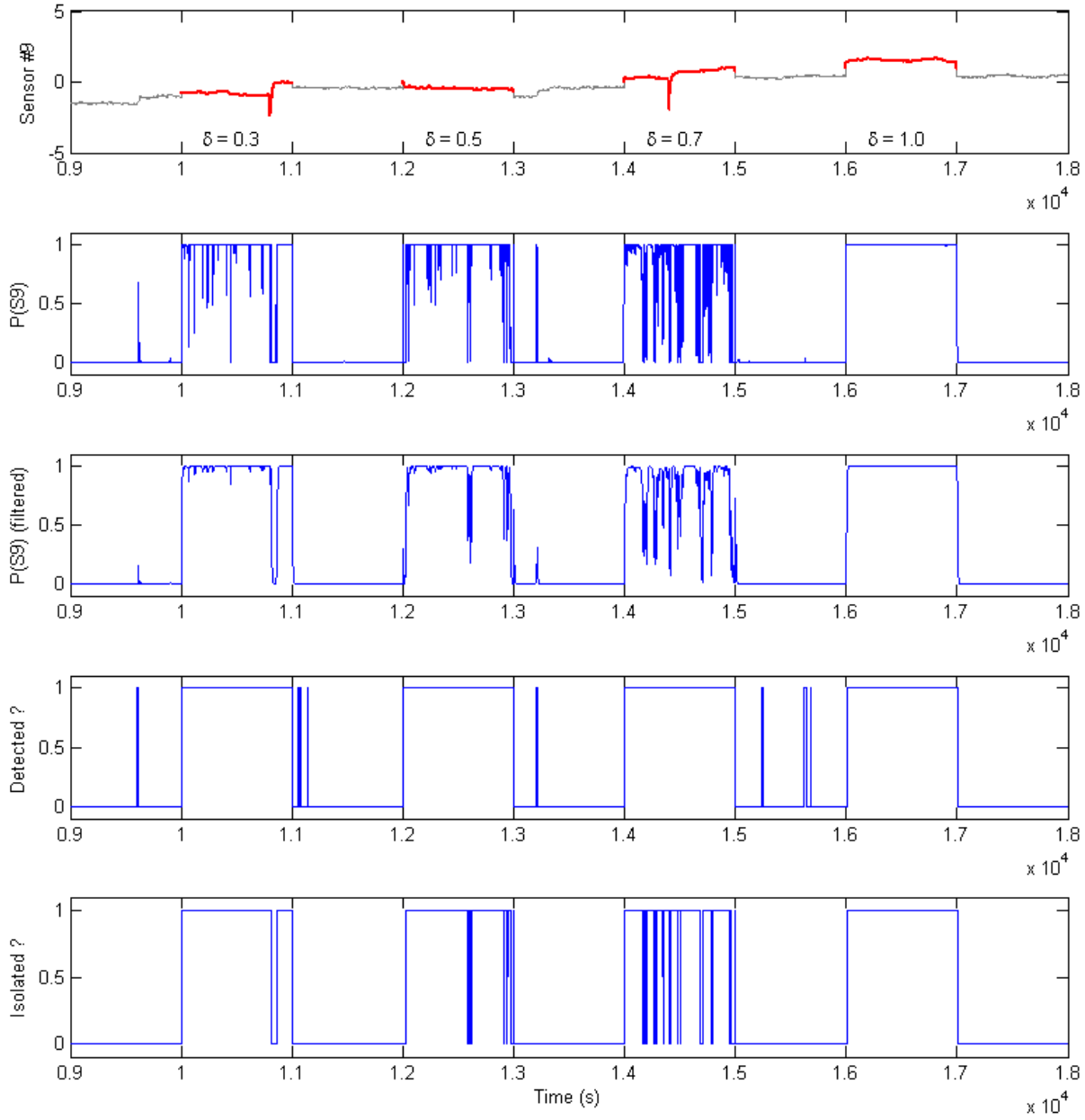

C Error
P(S8)
P(S8) (filtered)
Detected ?
Isolated ?
0
0.5
1
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
x 10^4
Time (s)

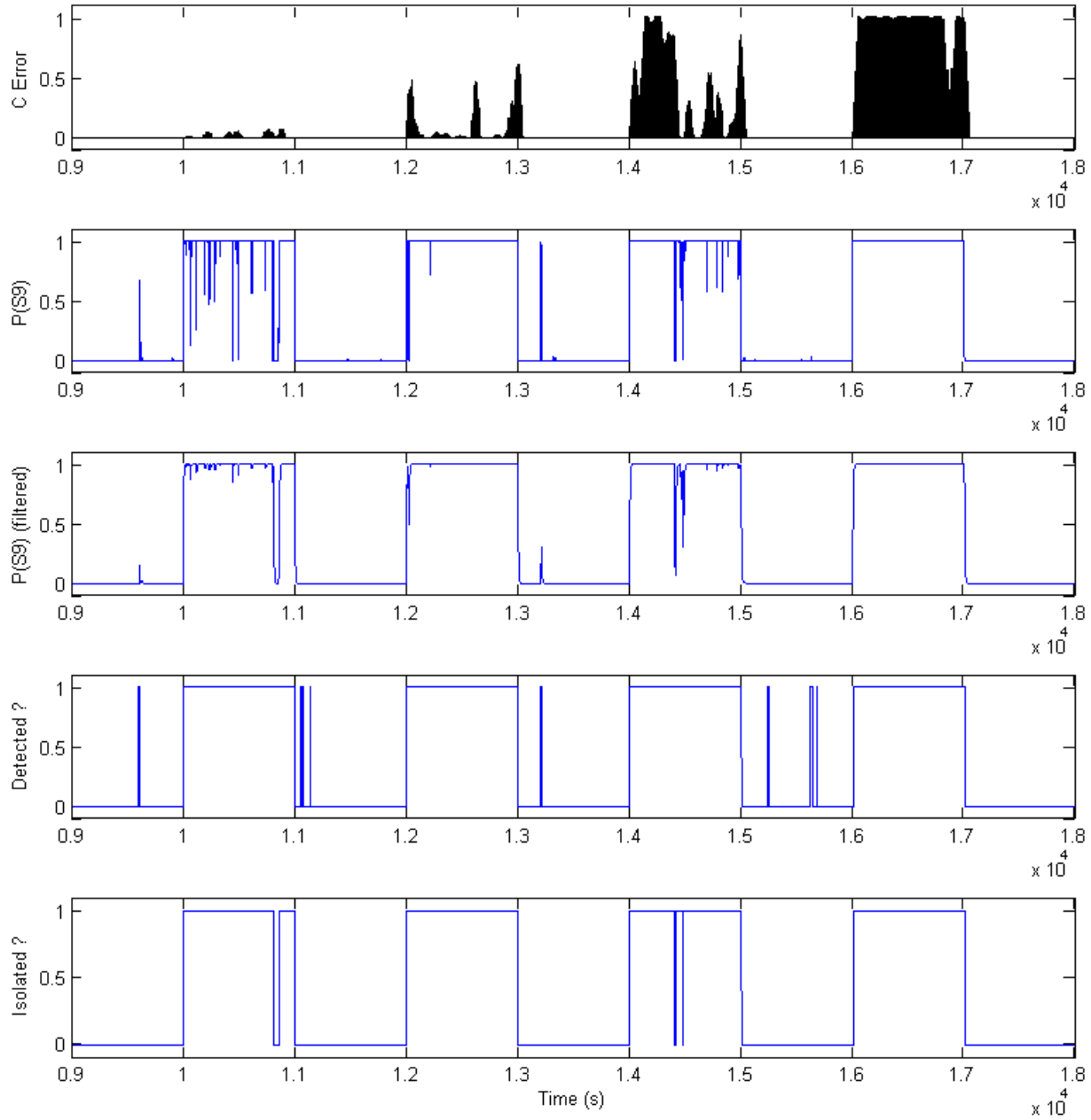

Sensor #8
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S8)
P(S8) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

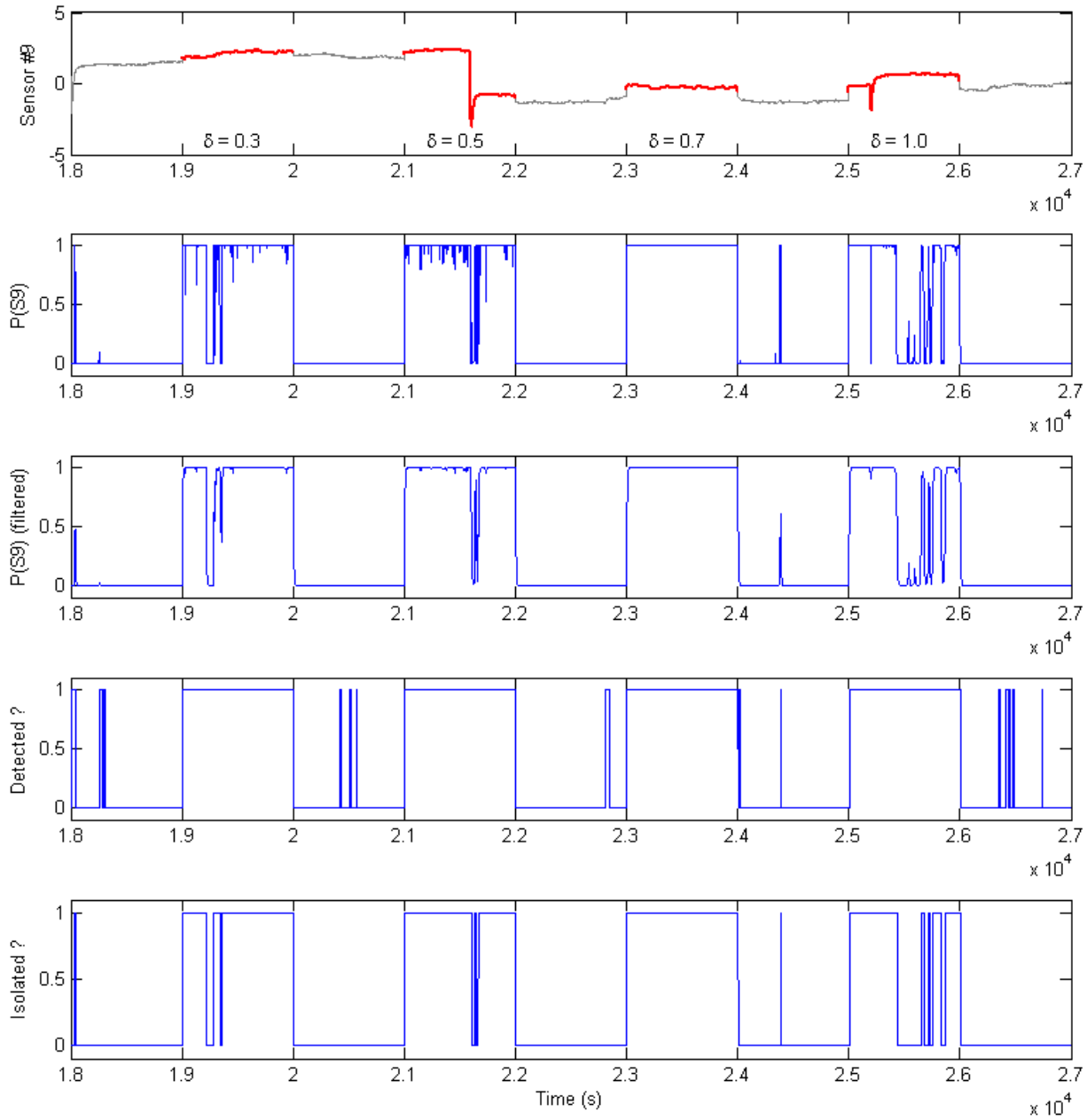

C Error
P(S8)
P(S8) (filtered)
Detected ?
Isolated ?
1
0.5
0
1.8
1.9
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7
x 10^4
Time (s)

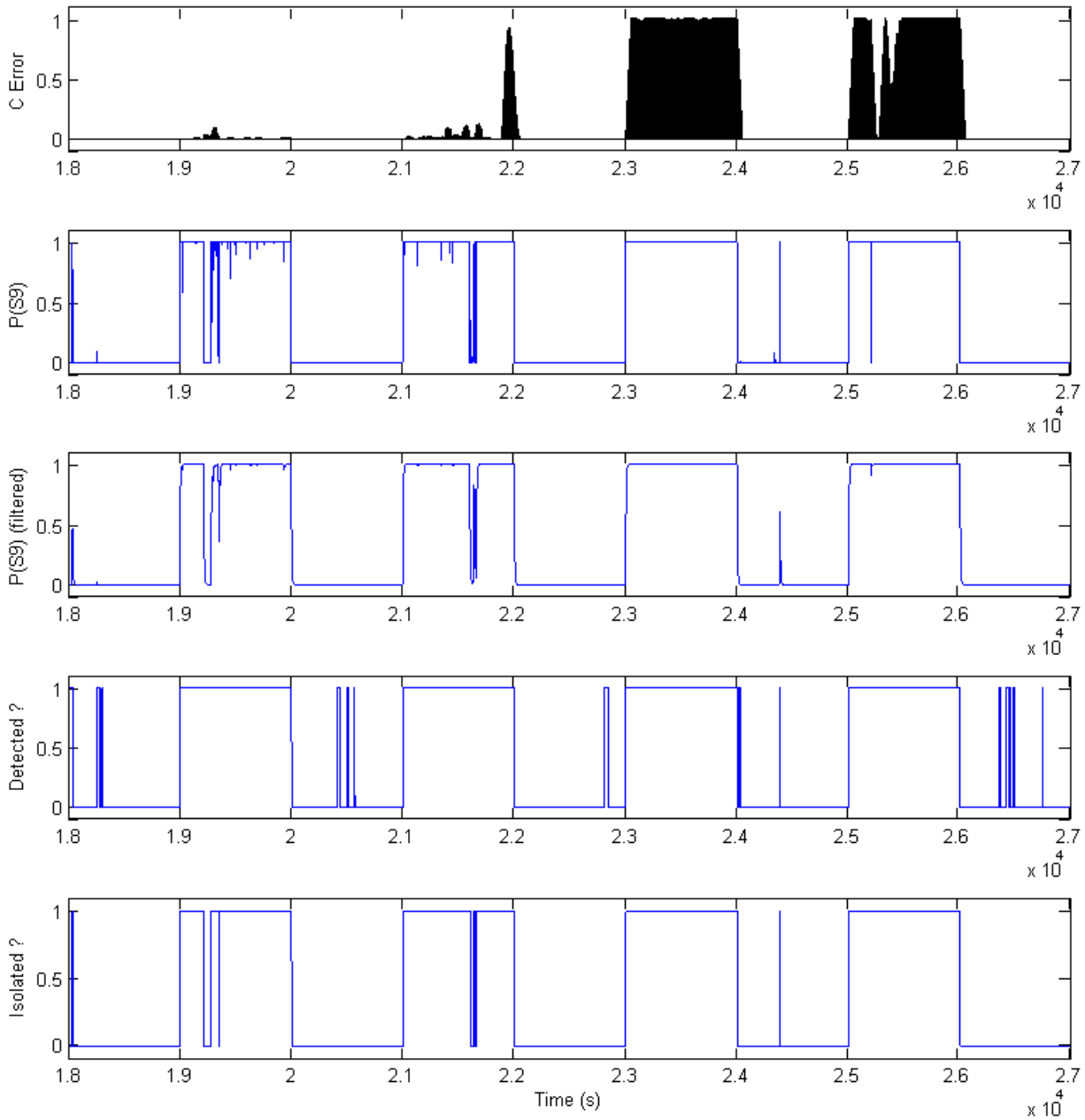

Sensor #9
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S9)
P(S9) (filtered)
Detected ?
Isolated ?
Time (s)

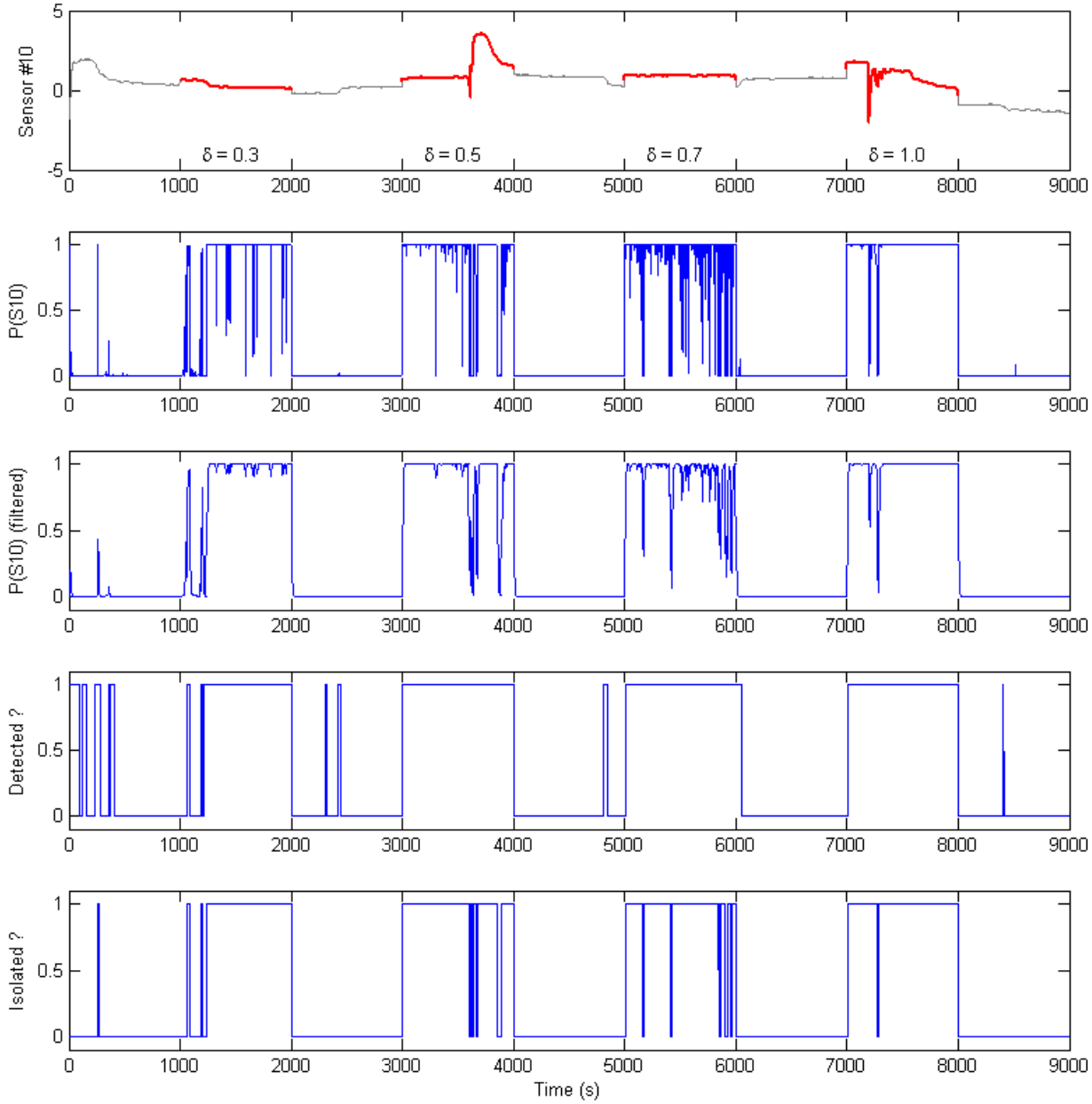

C Error
P(S9)
P(S9) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

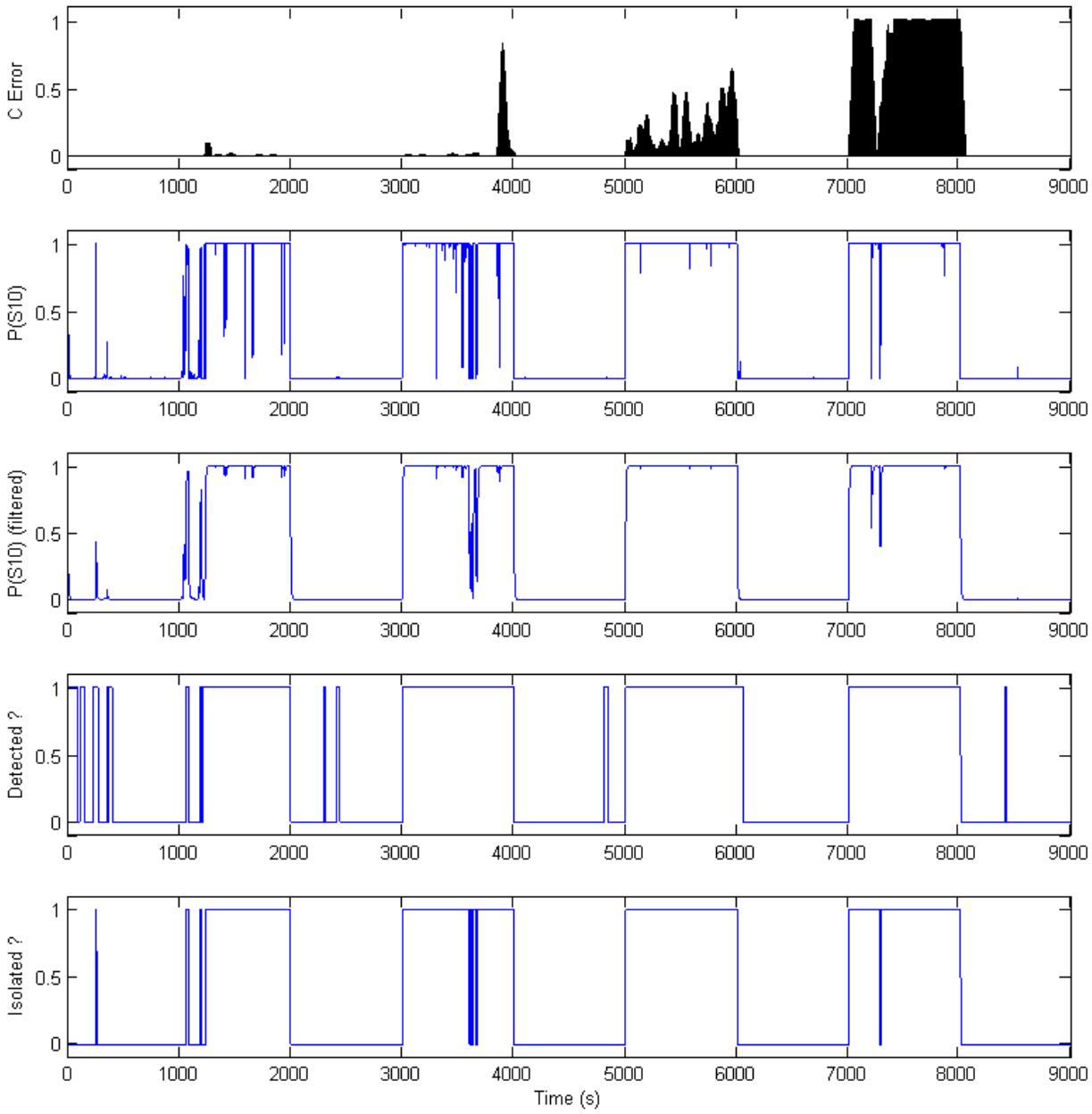

Sensor #9
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S9)
P(S9) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

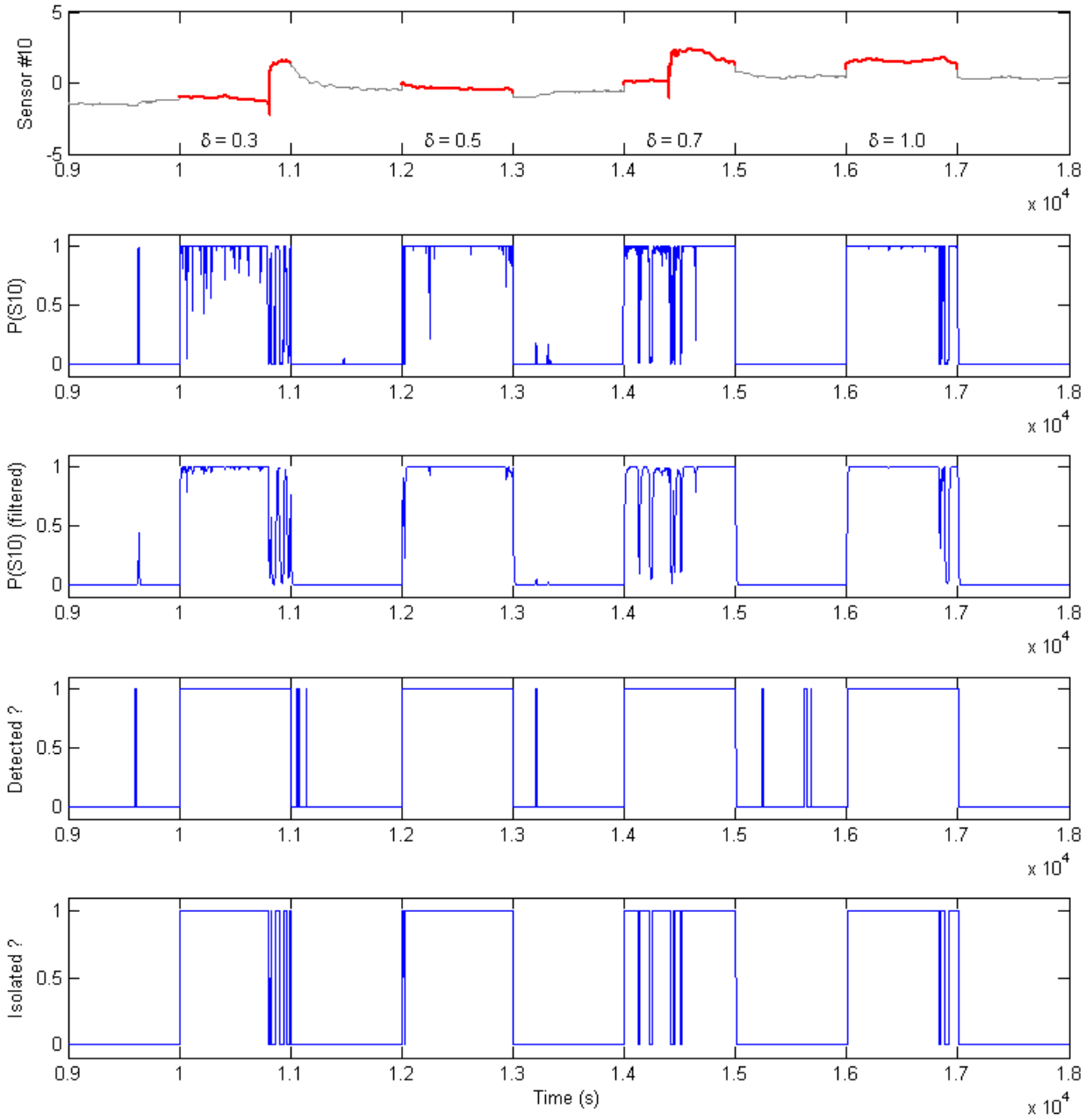

C Error
P(S9)
P(S9) (filtered)
Detected ?
Isolated ?
0
0.5
1
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
x 10^4
Time (s)

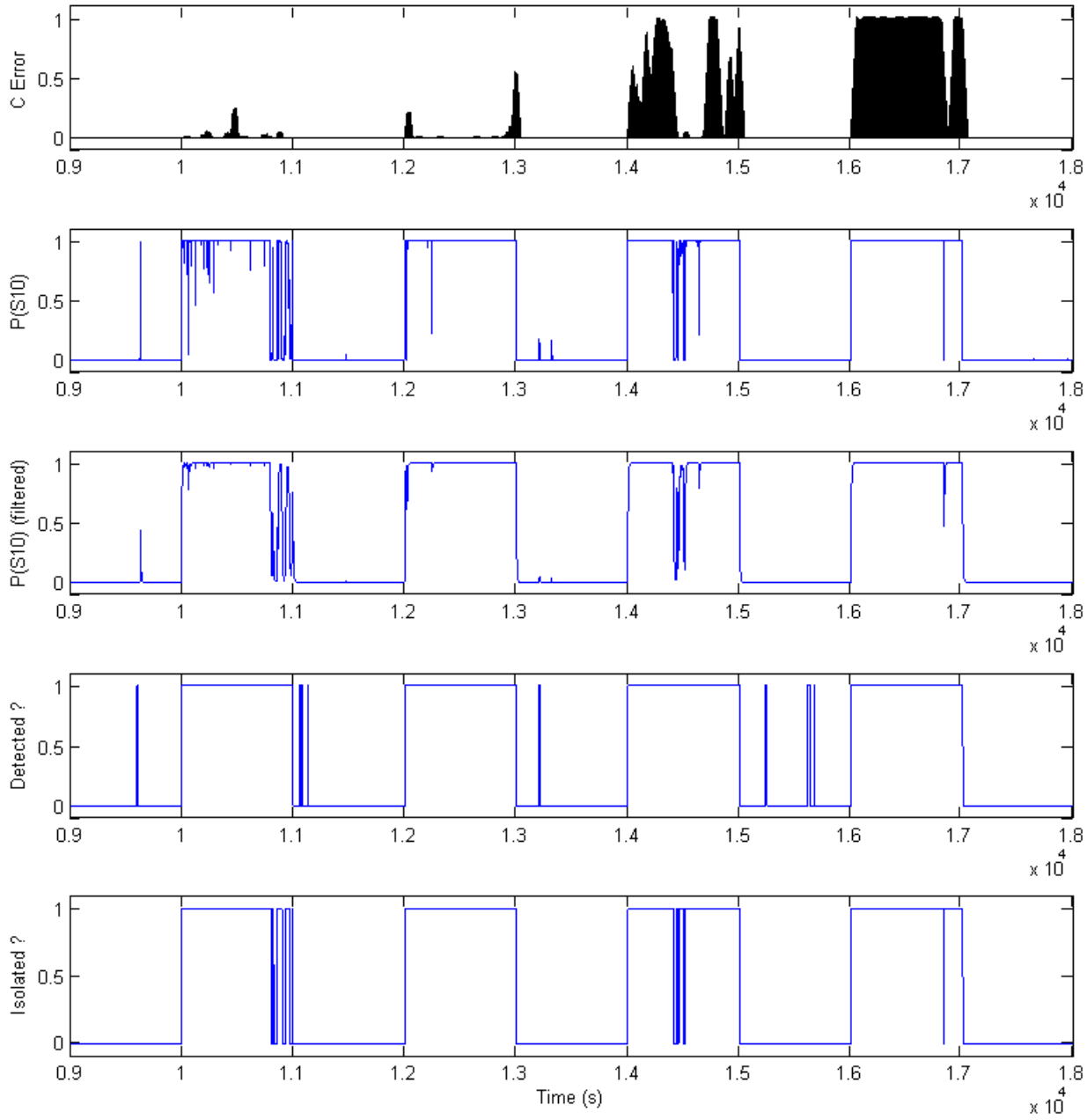

Sensor #9
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S9)
P(S9) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

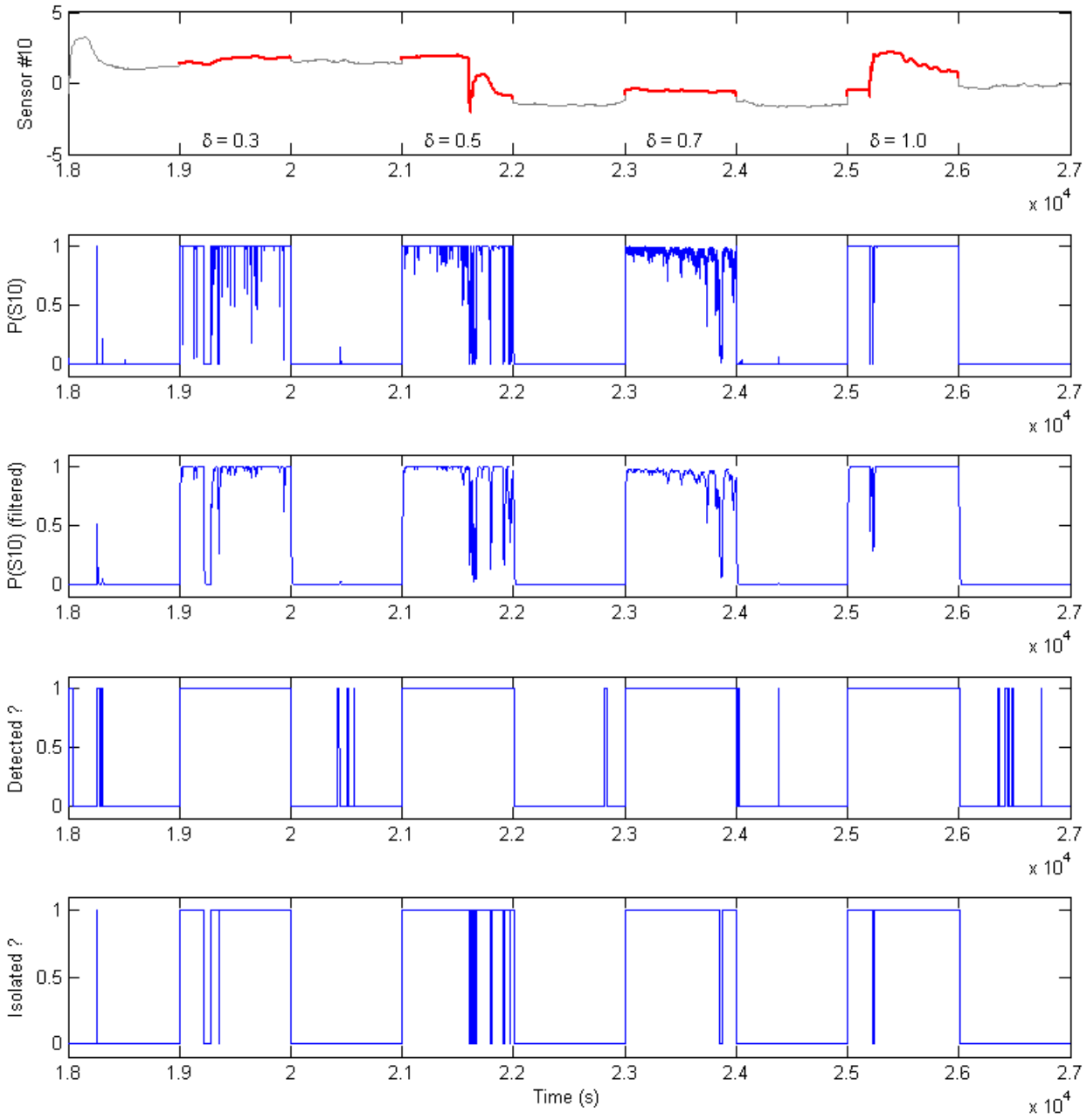

C Error
P(S9)
P(S9) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

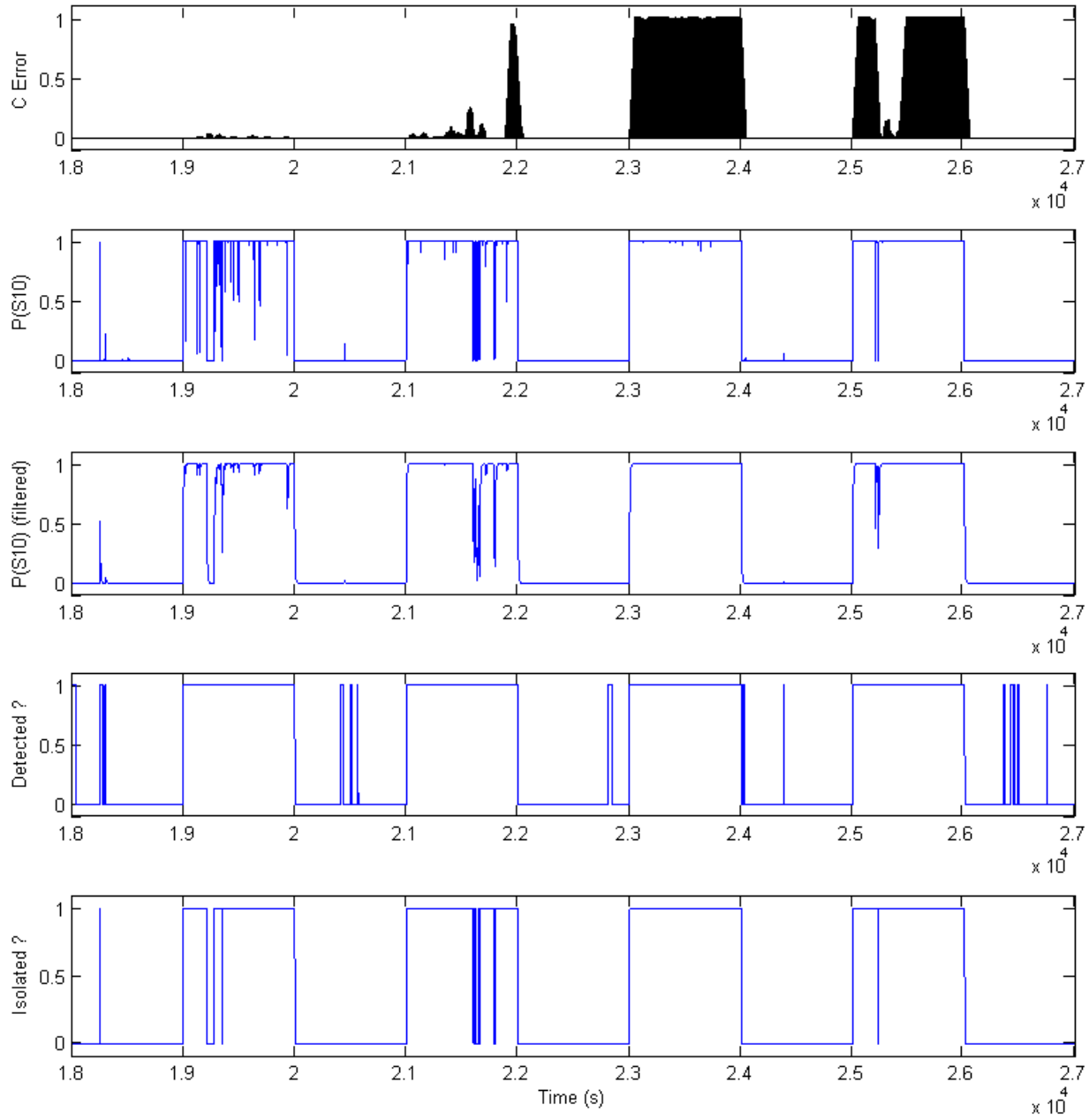

Sensor #10
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S10)
P(S10) (filtered)
Detected ?
Isolated ?
Time (s)

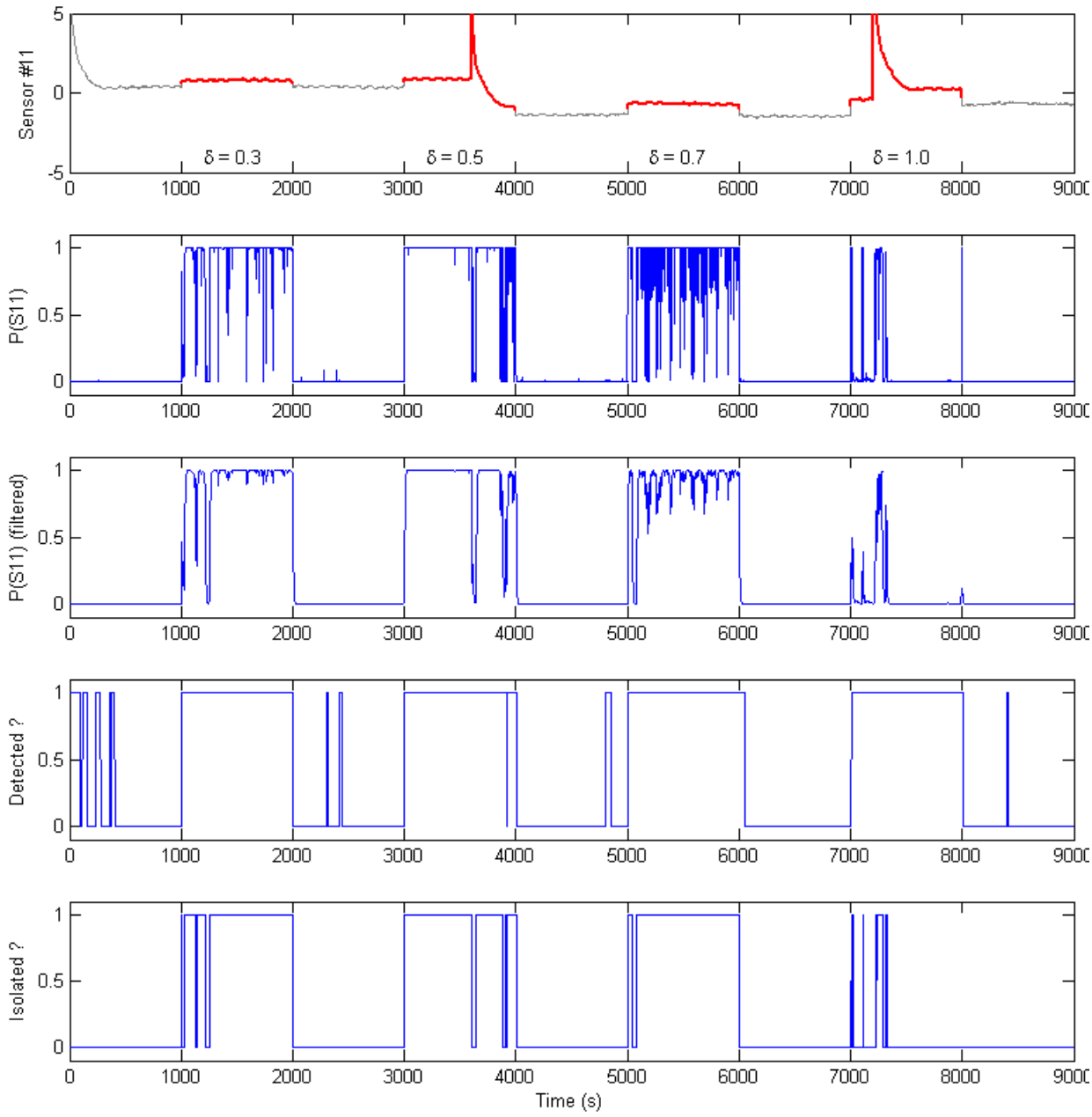

C Error
P(S10)
P(S10) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

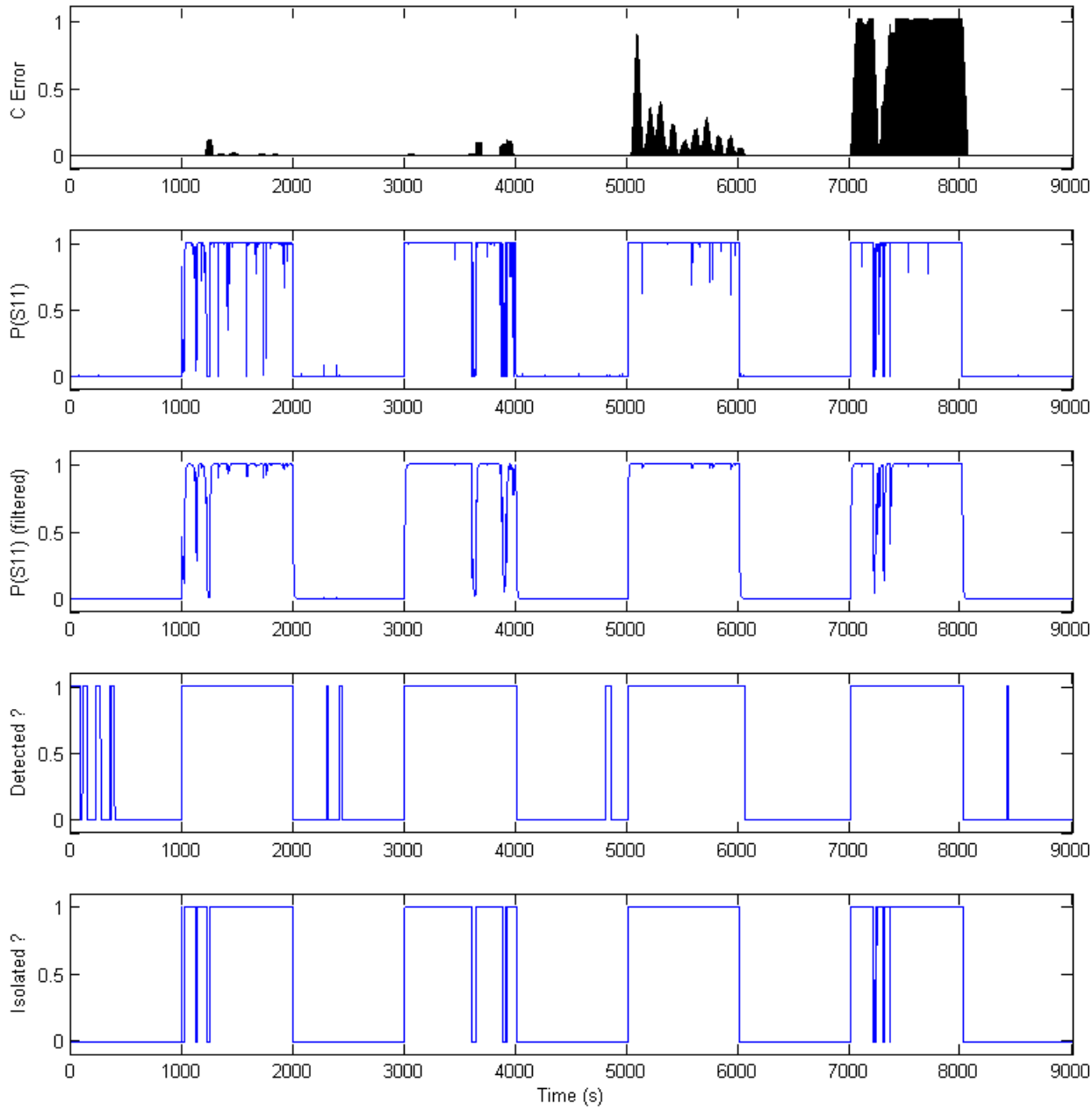

Sensor #10
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S10)
P(S10) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

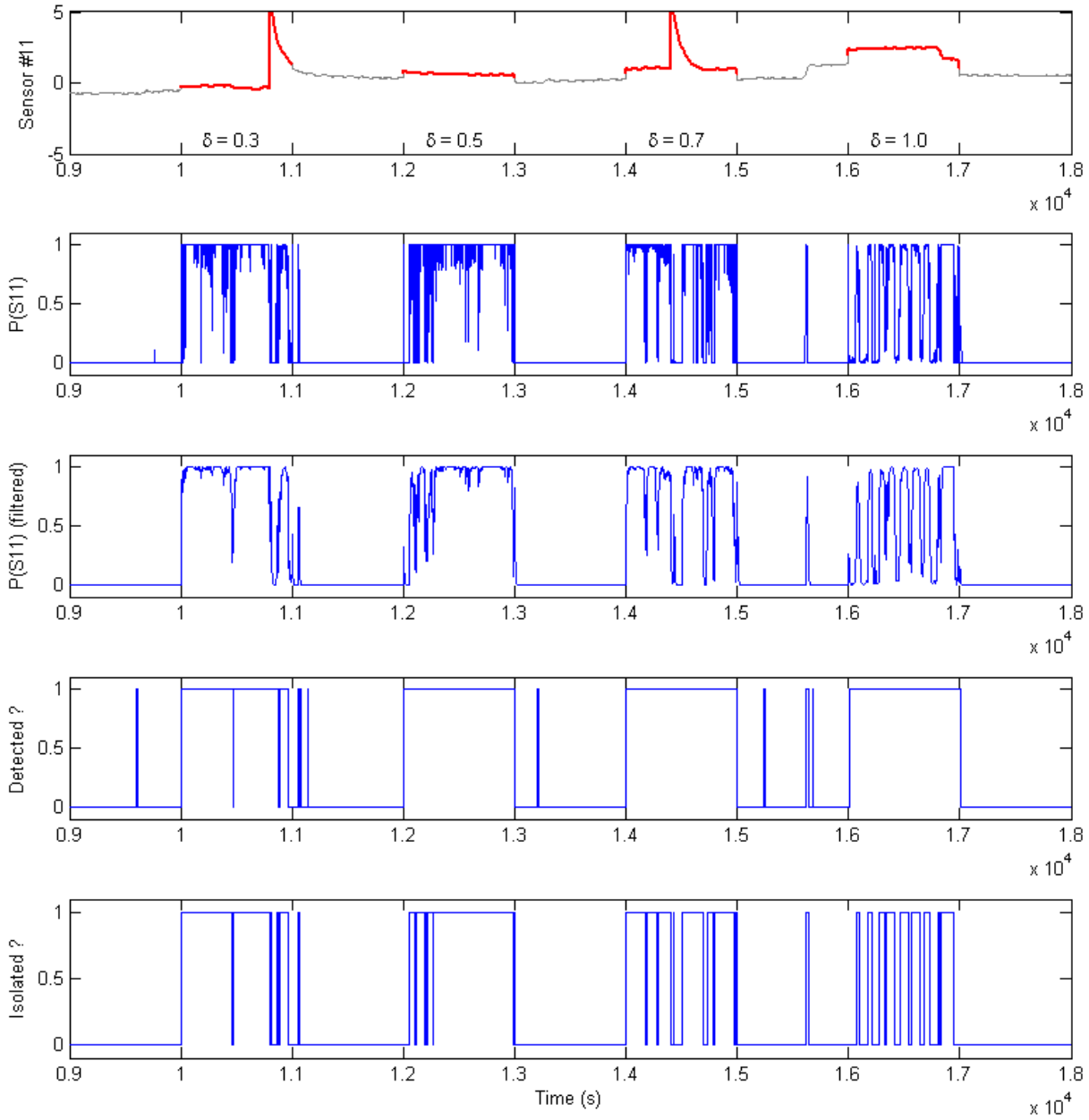

C Error
P(S10)
P(S10) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

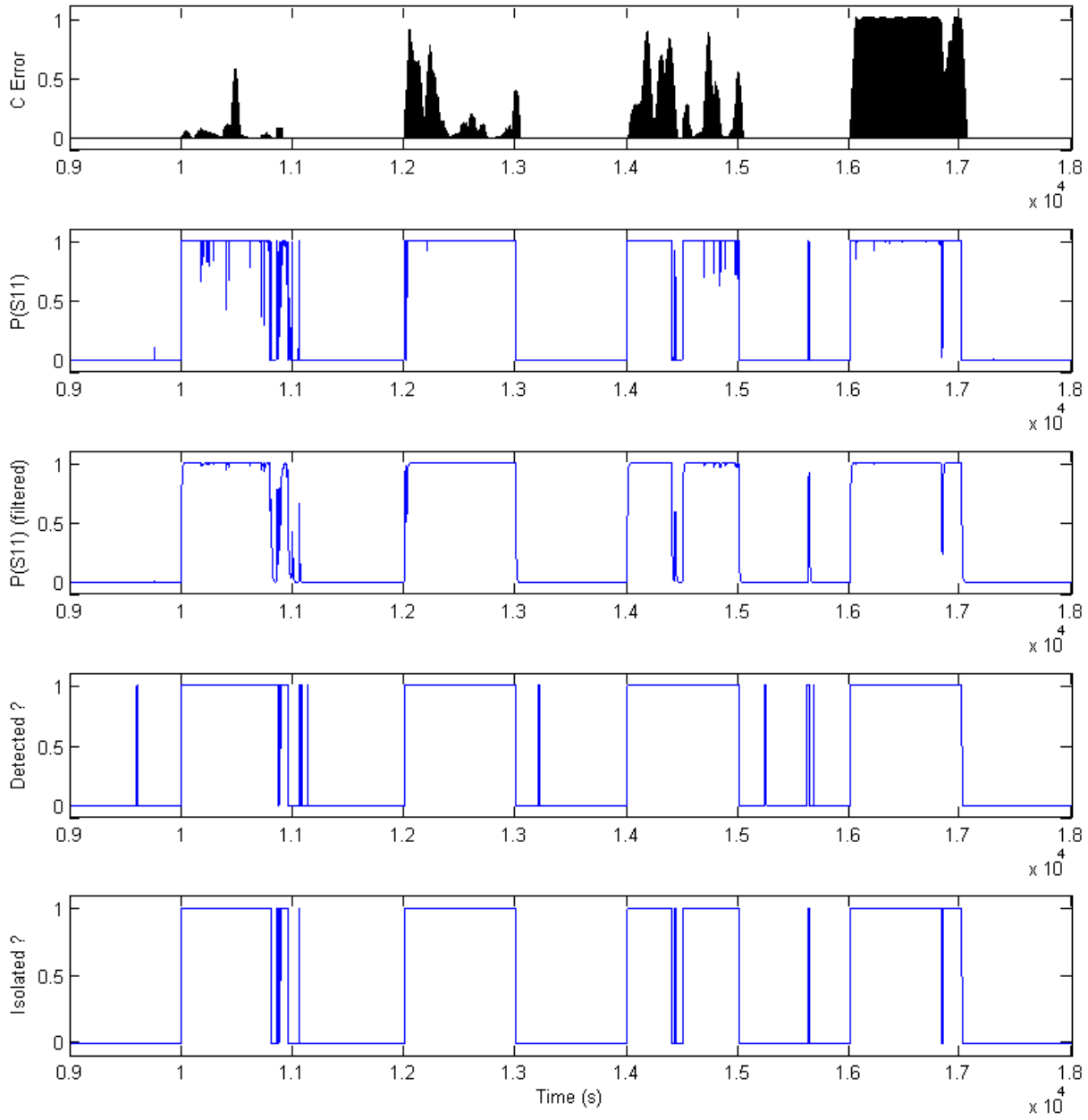

Sensor #10
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S1D)
P(S1D) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

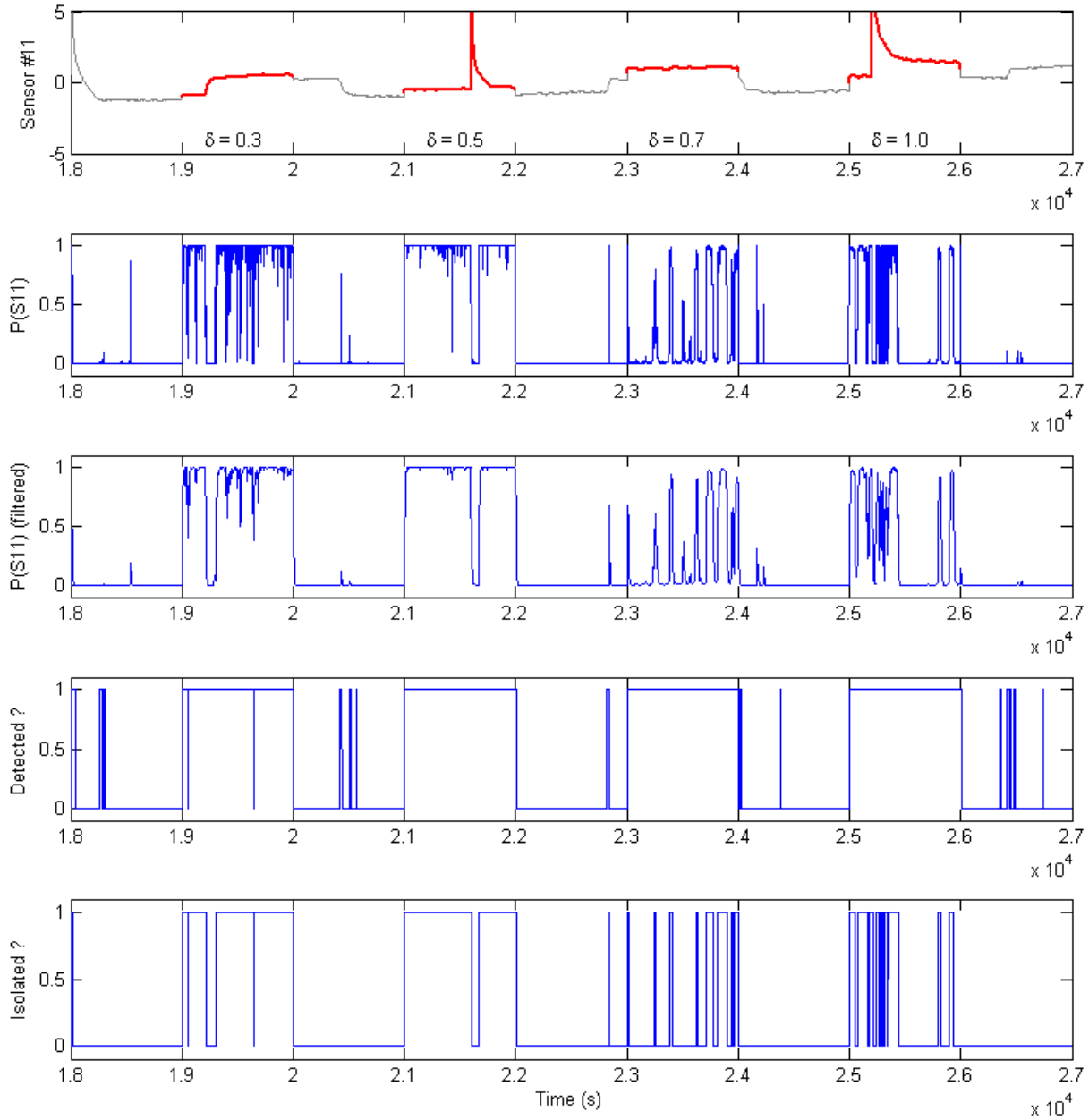

C Error
P(S10)
P(S10) (filtered)
Detected ?
Isolated ?
1
0.5
0
1.8
1.9
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7
x 10^4
Time (s)

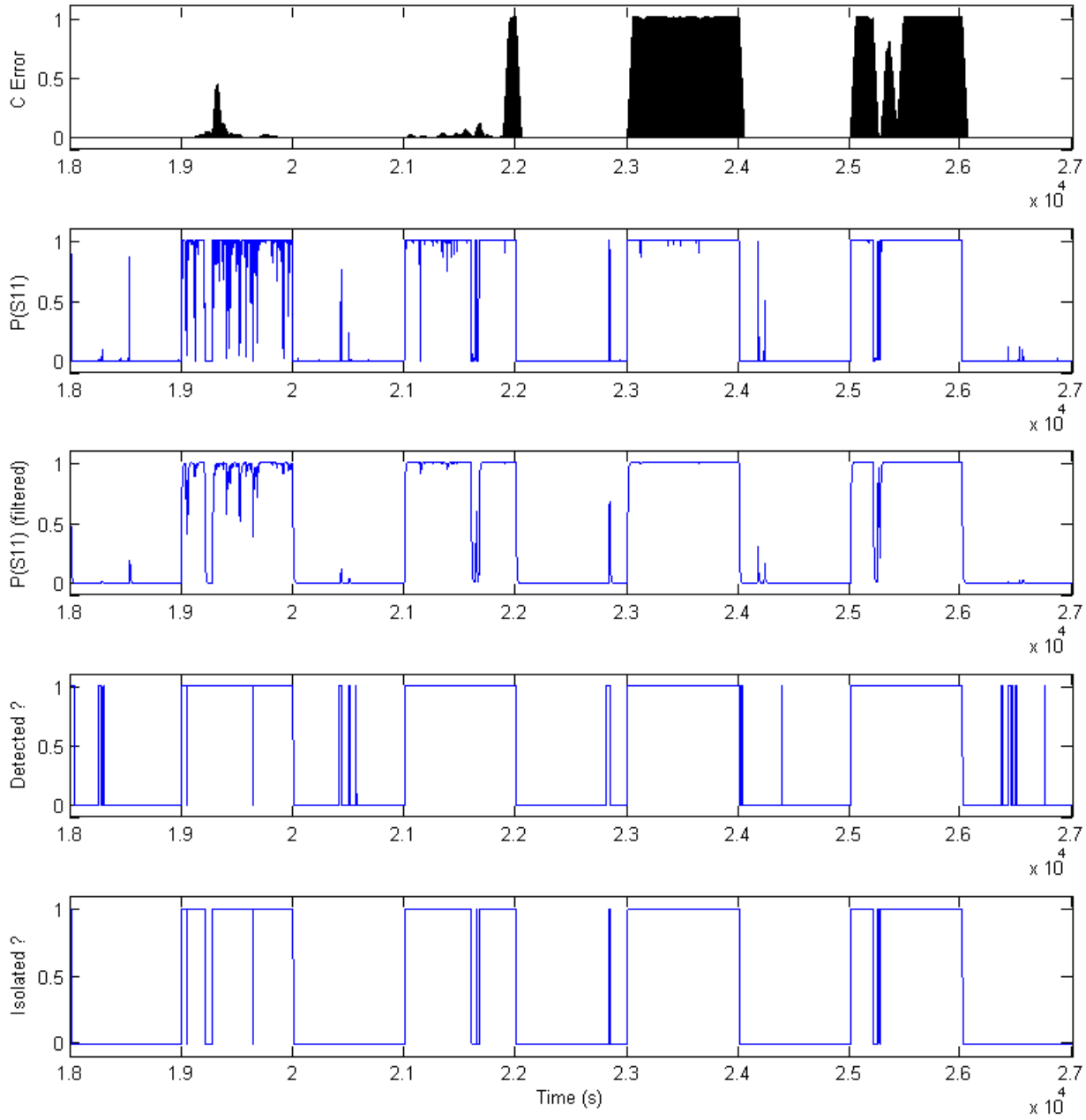

Sensor #11
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S11)
P(S11) (filtered)
Detected ?
Isolated ?
Time (s)

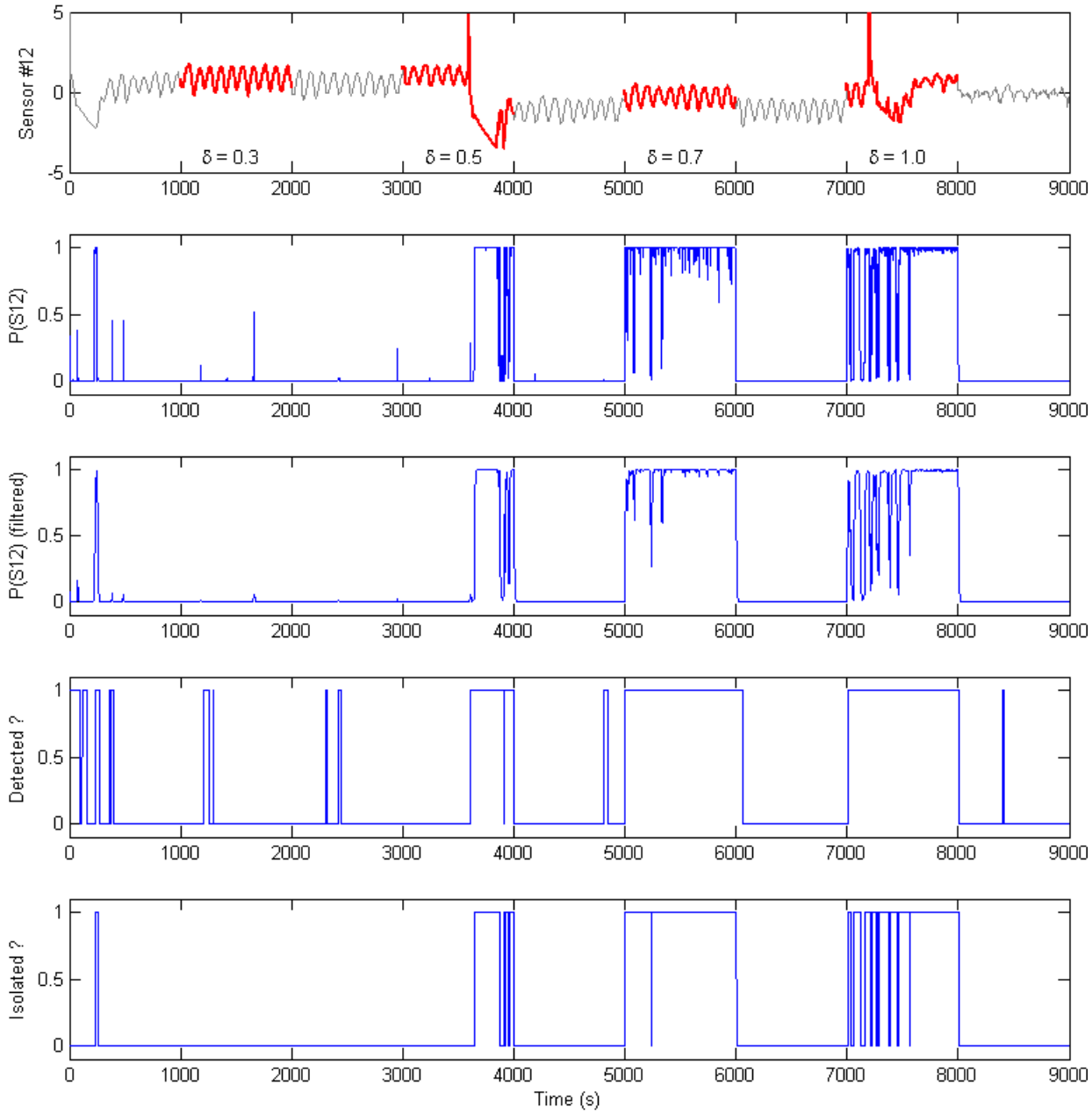

C Error
P(S11)
P(S11) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

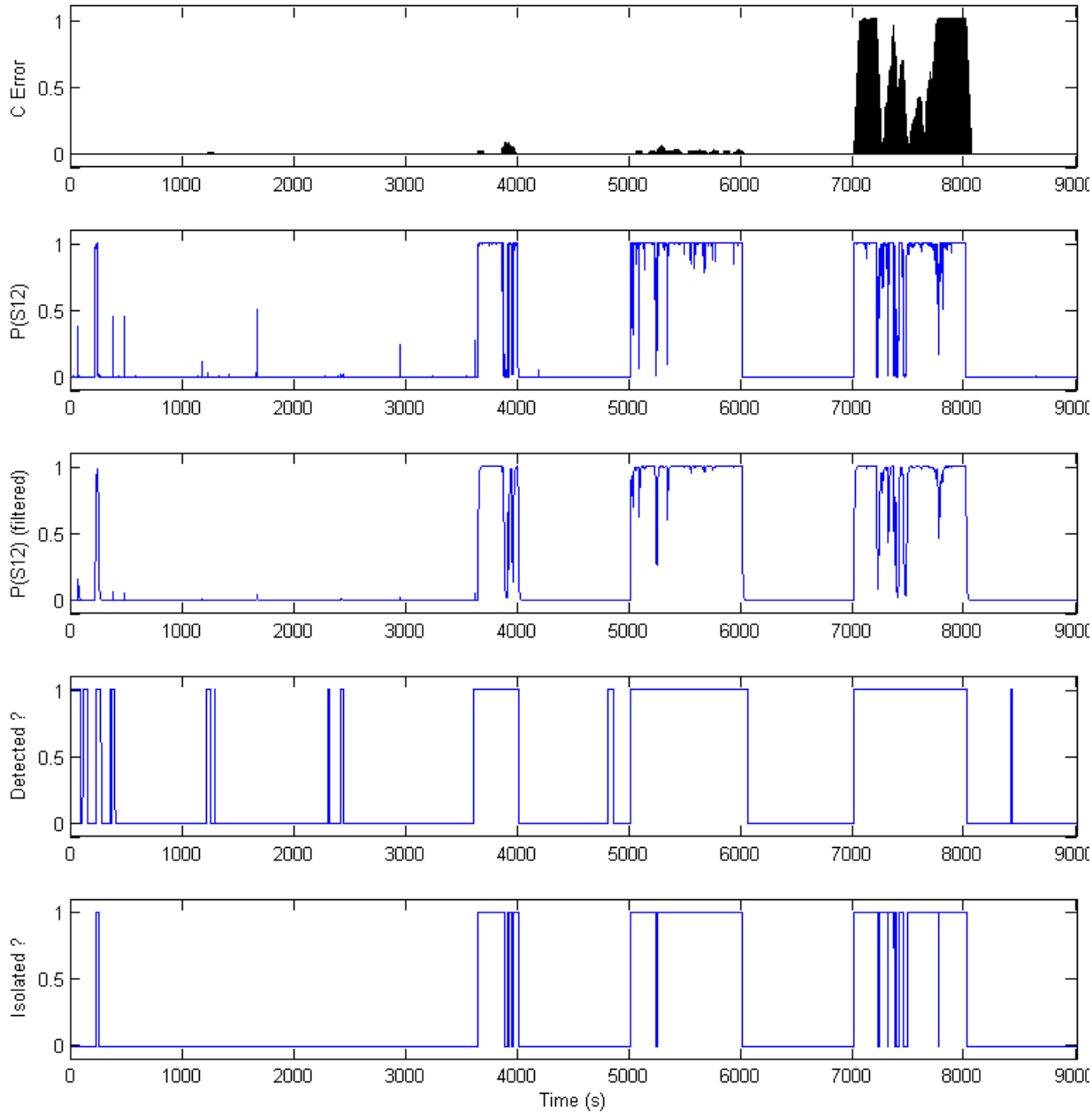

Sensor #11
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S11)
P(S11) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

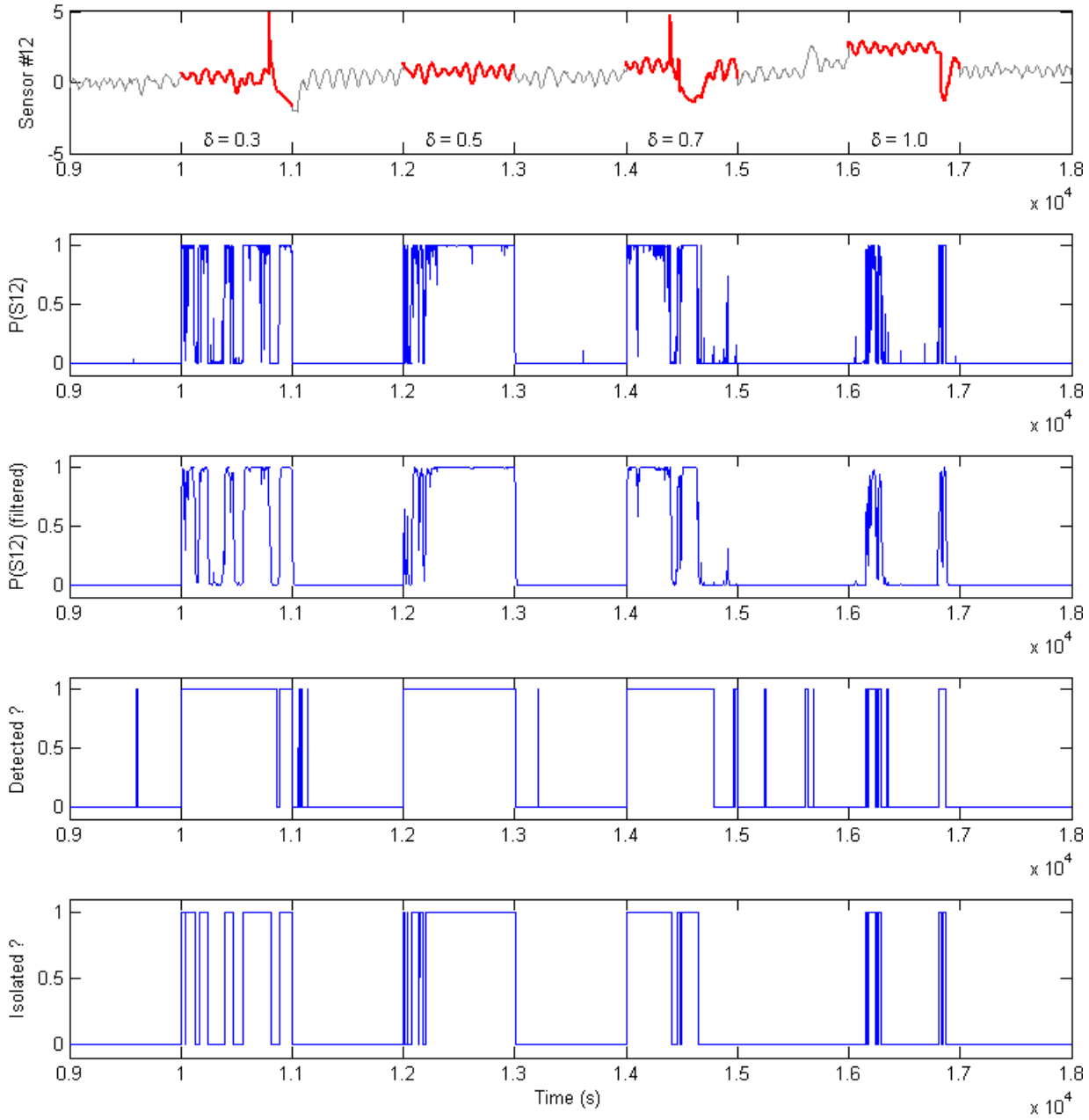

C Error
P(S11)
P(S11) (filtered)
Detected ?
Isolated ?
0
0.5
1
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
x 10^4
Time (s)

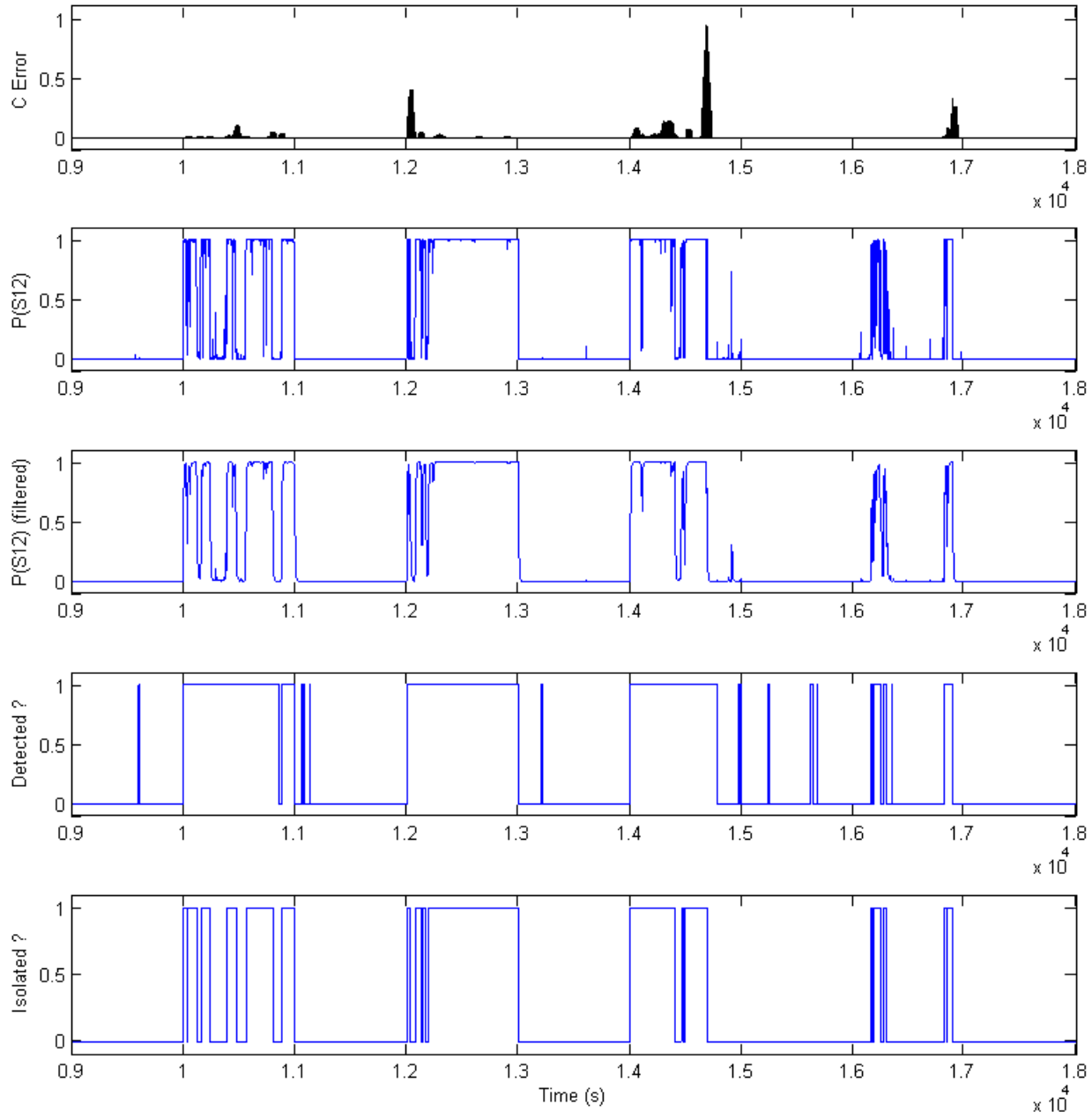

Sensor #11
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S11)
P(S11) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

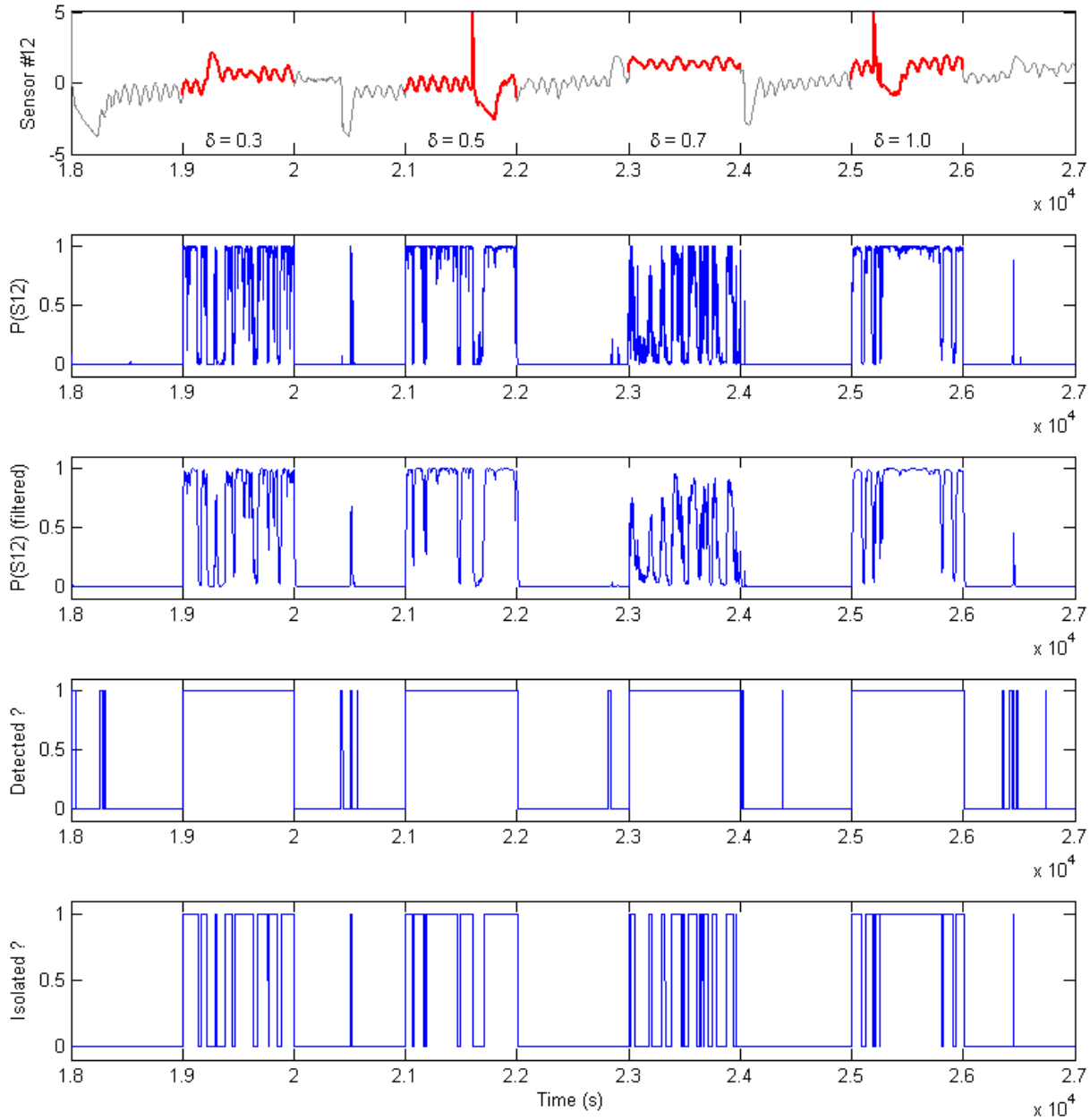

C Error
P(S11)
P(S11) (filtered)
Detected ?
Isolated ?
1
0.5
0
1.8
1.9
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7
x 10^4
Time (s)

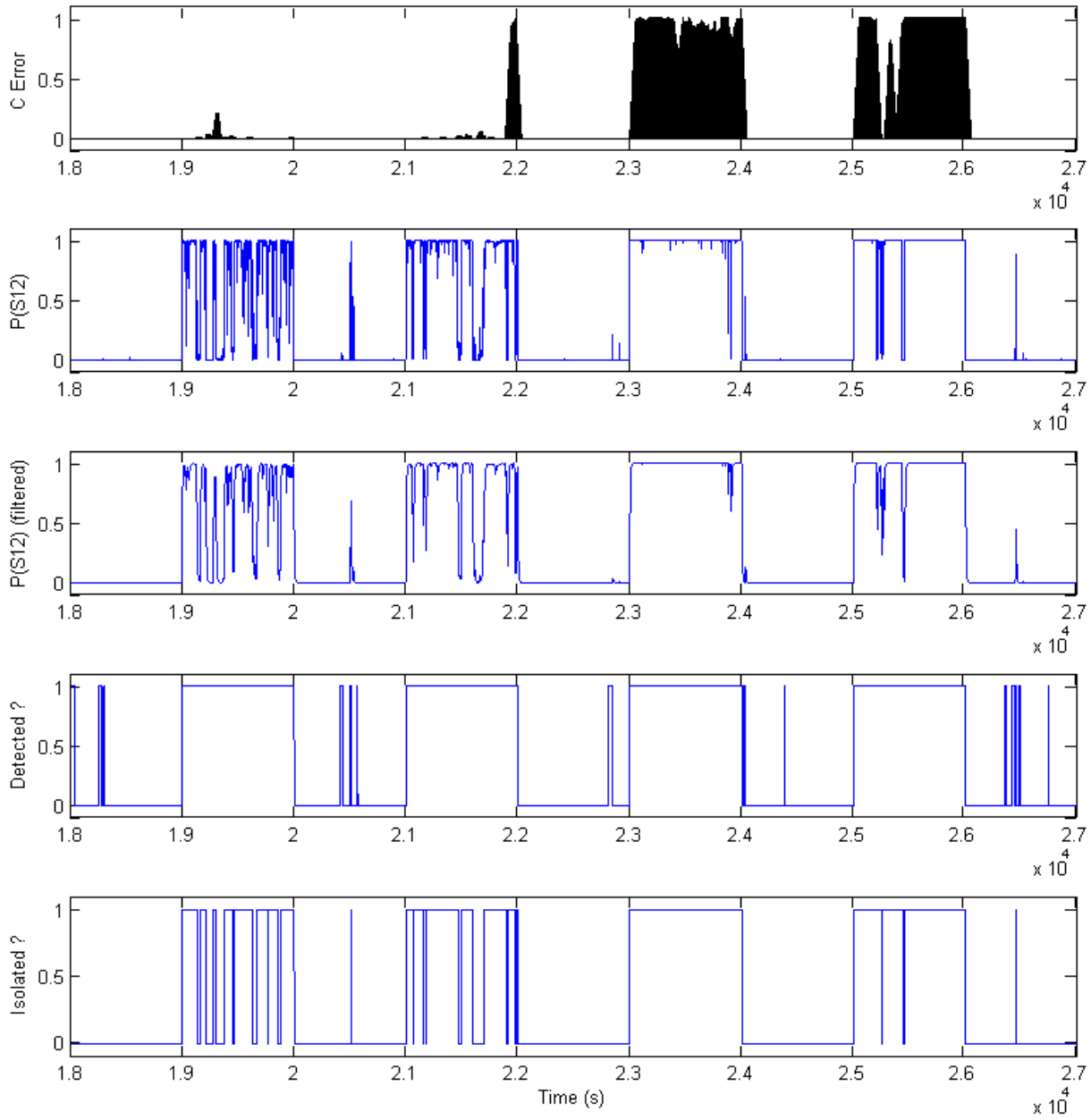

Sensor #12
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S12)
P(S12) (filtered)
Detected ?
Isolated ?
Time (s)

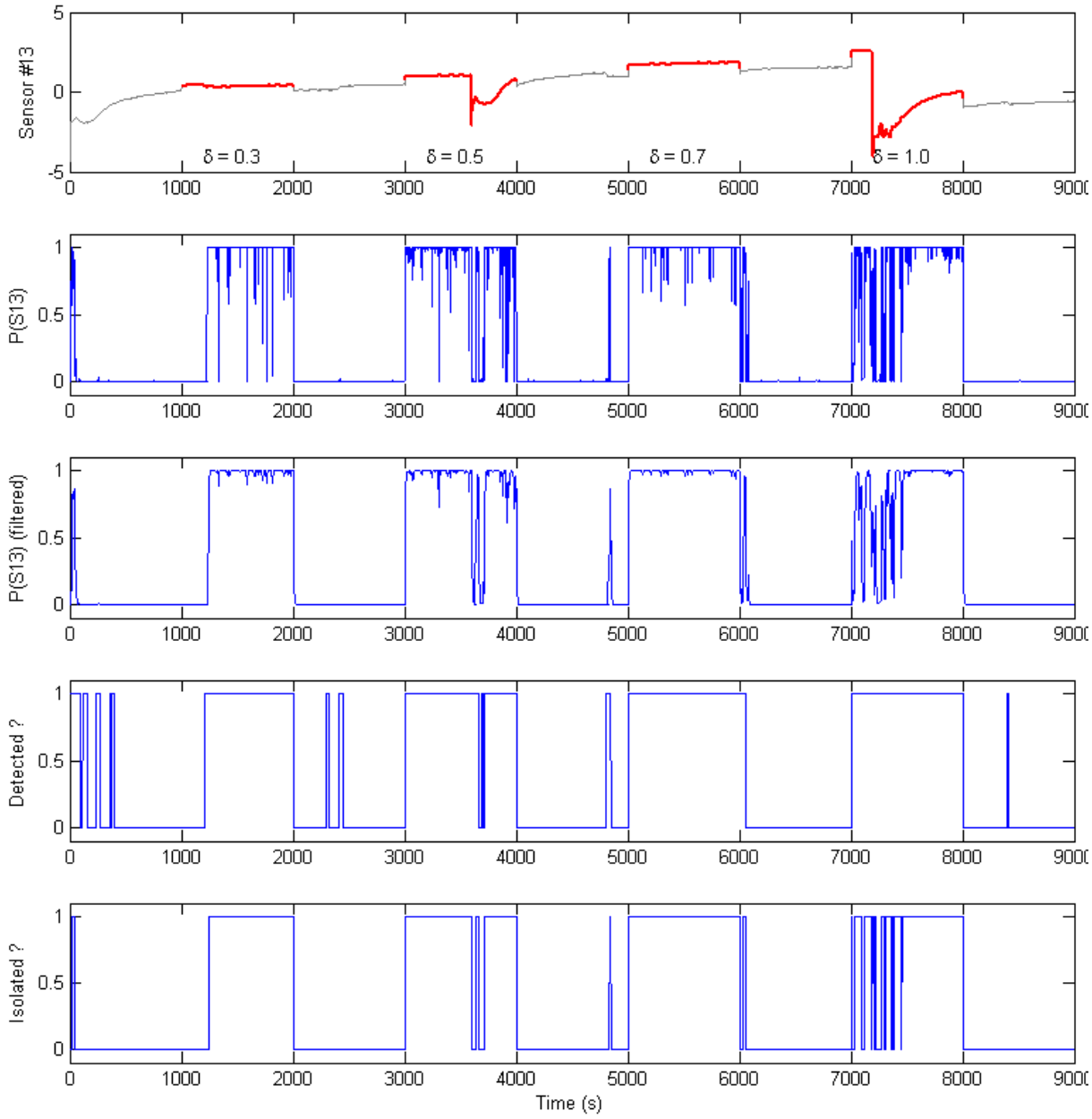

C Error
P(S12)
P(S12) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

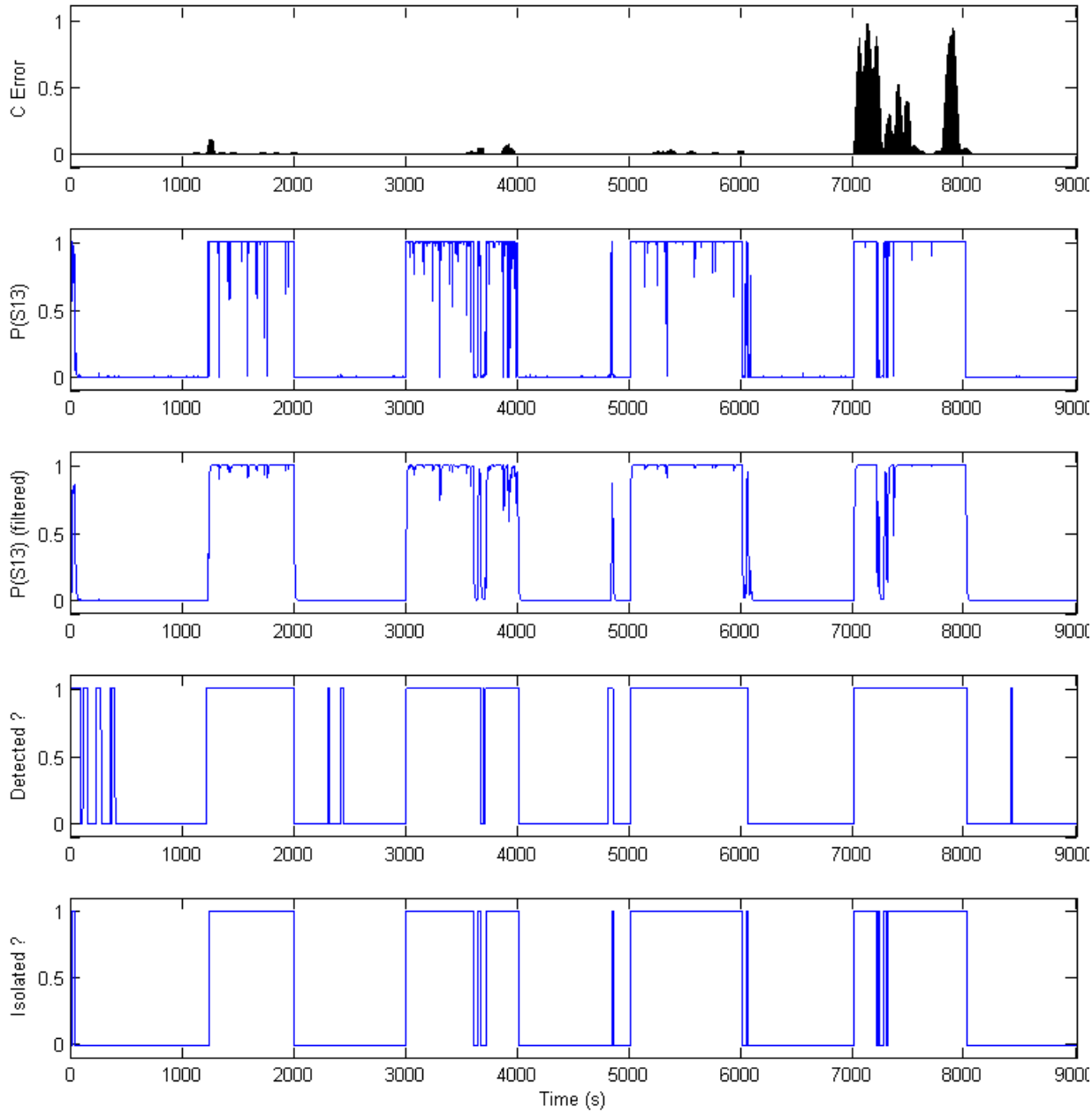

Sensor #12
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S12)
P(S12) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

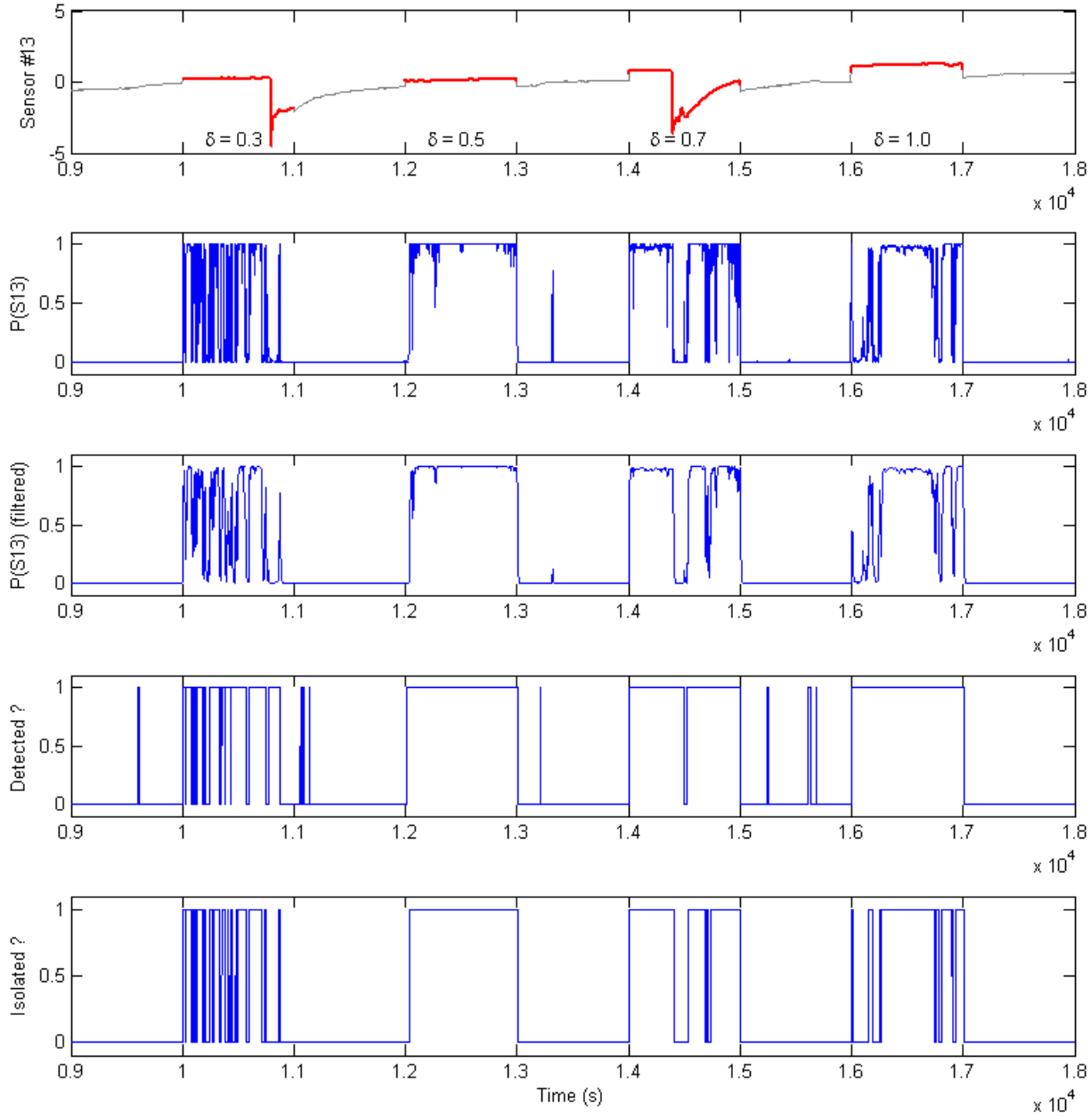

C Error
P(S12)
P(S12) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

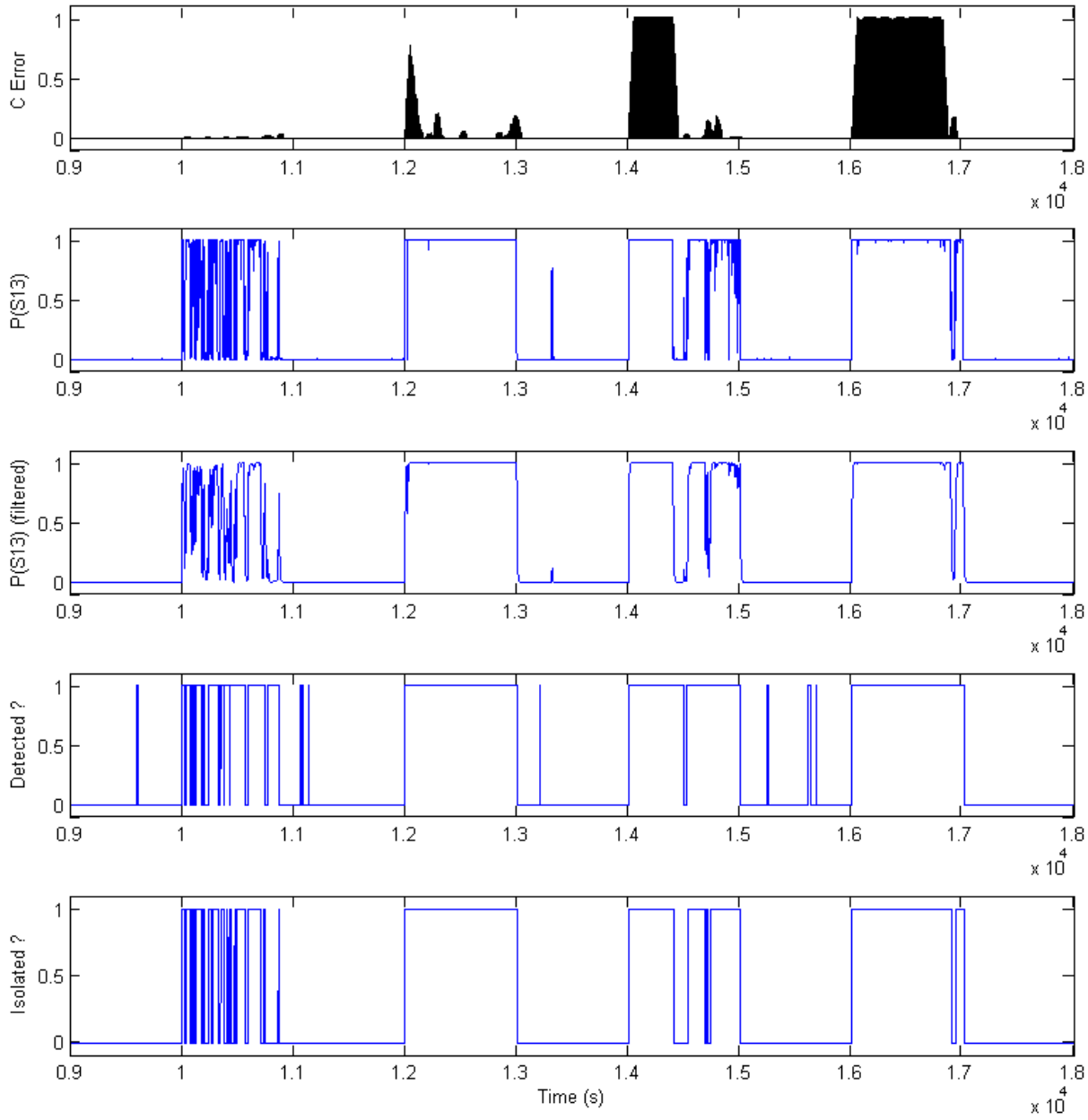

Sensor #12
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S12)
P(S12) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

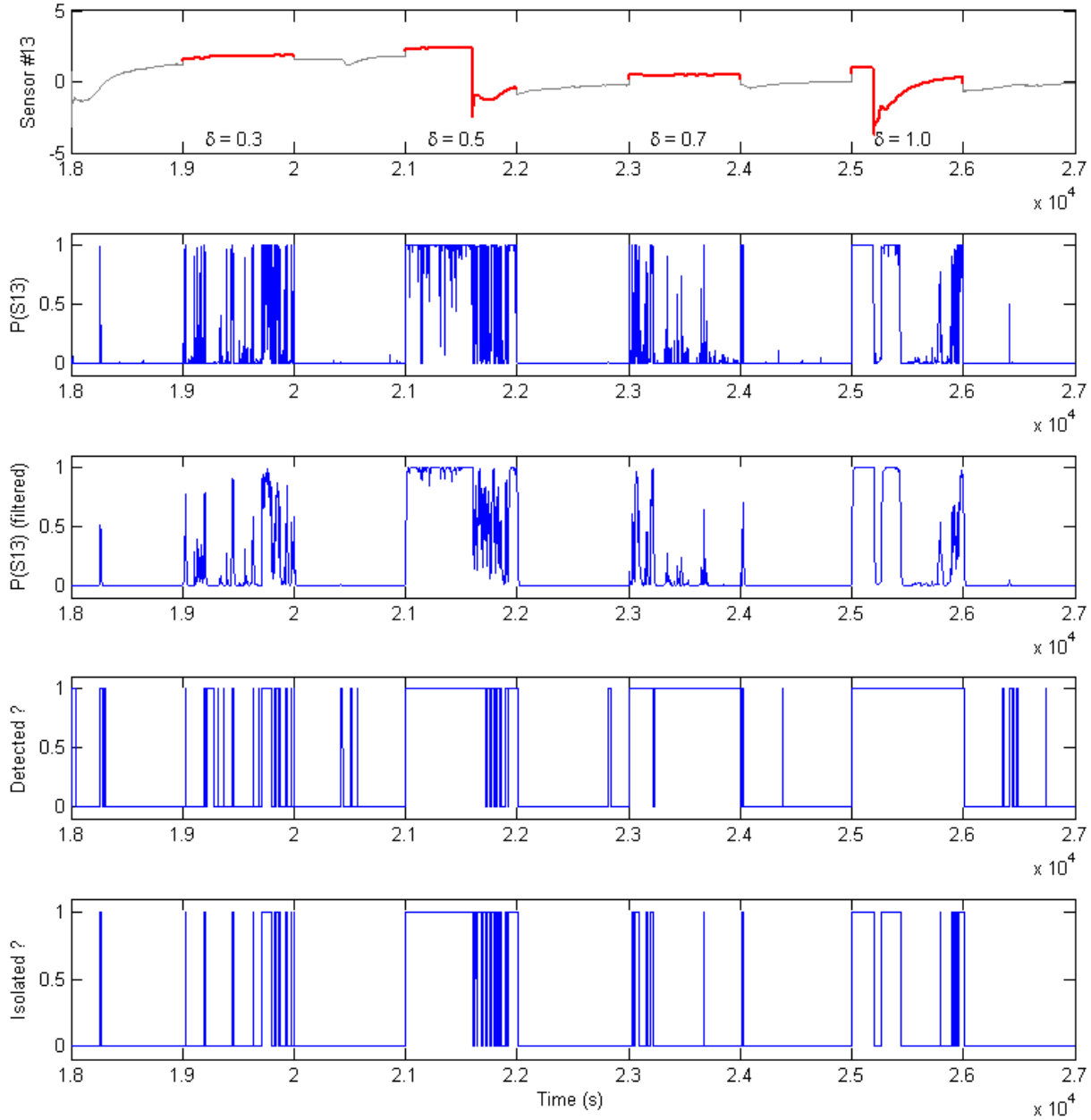

C Error
P(S12)
P(S12) (filtered)
Detected ?
Isolated ?
1
0.5
0
1.8
1.9
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7
x 10^4
Time (s)

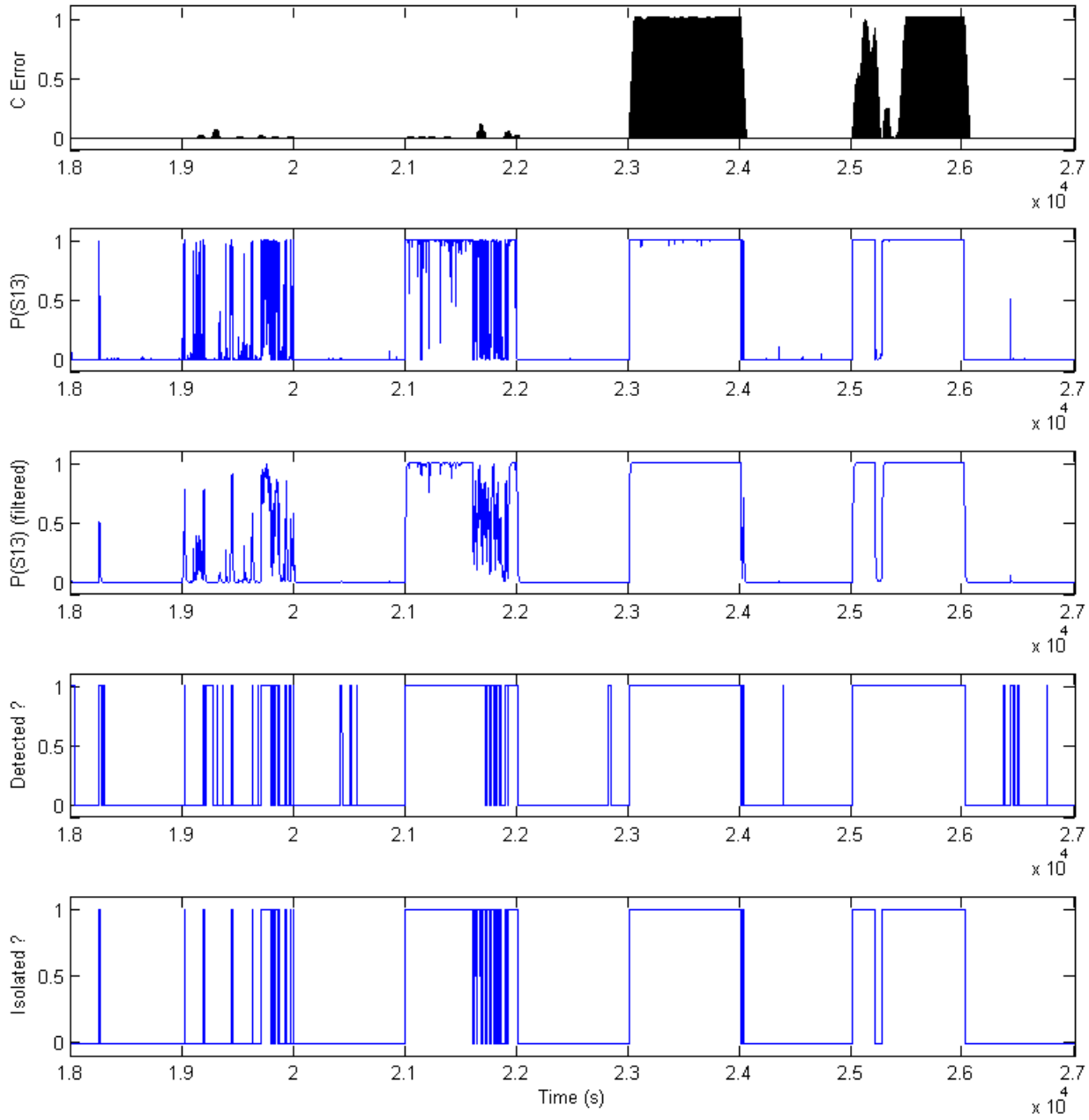

Sensor #13
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S13)
P(S13) (filtered)
Detected ?
Isolated ?
Time (s)

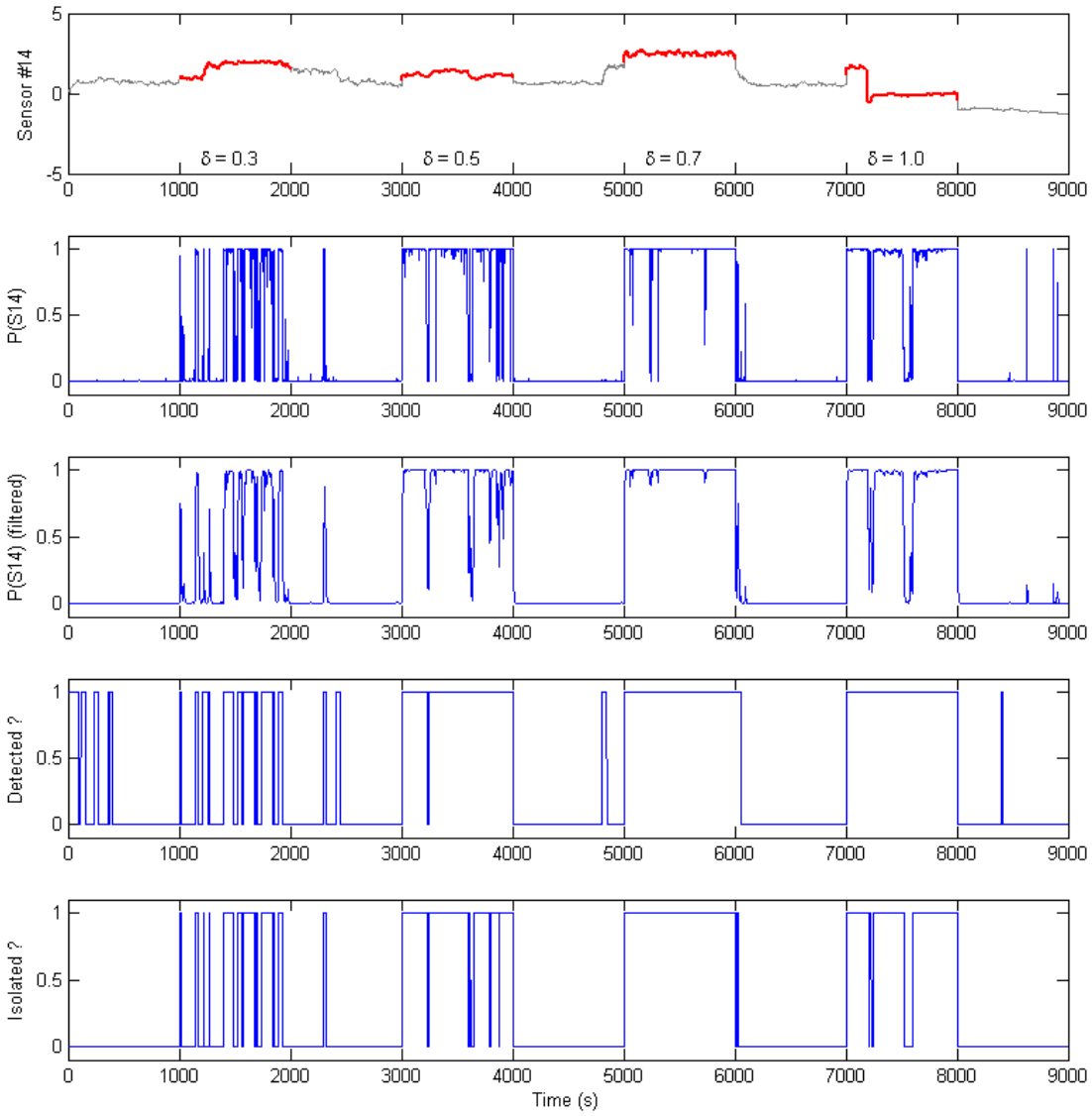

C Error
P(S13)
P(S13) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

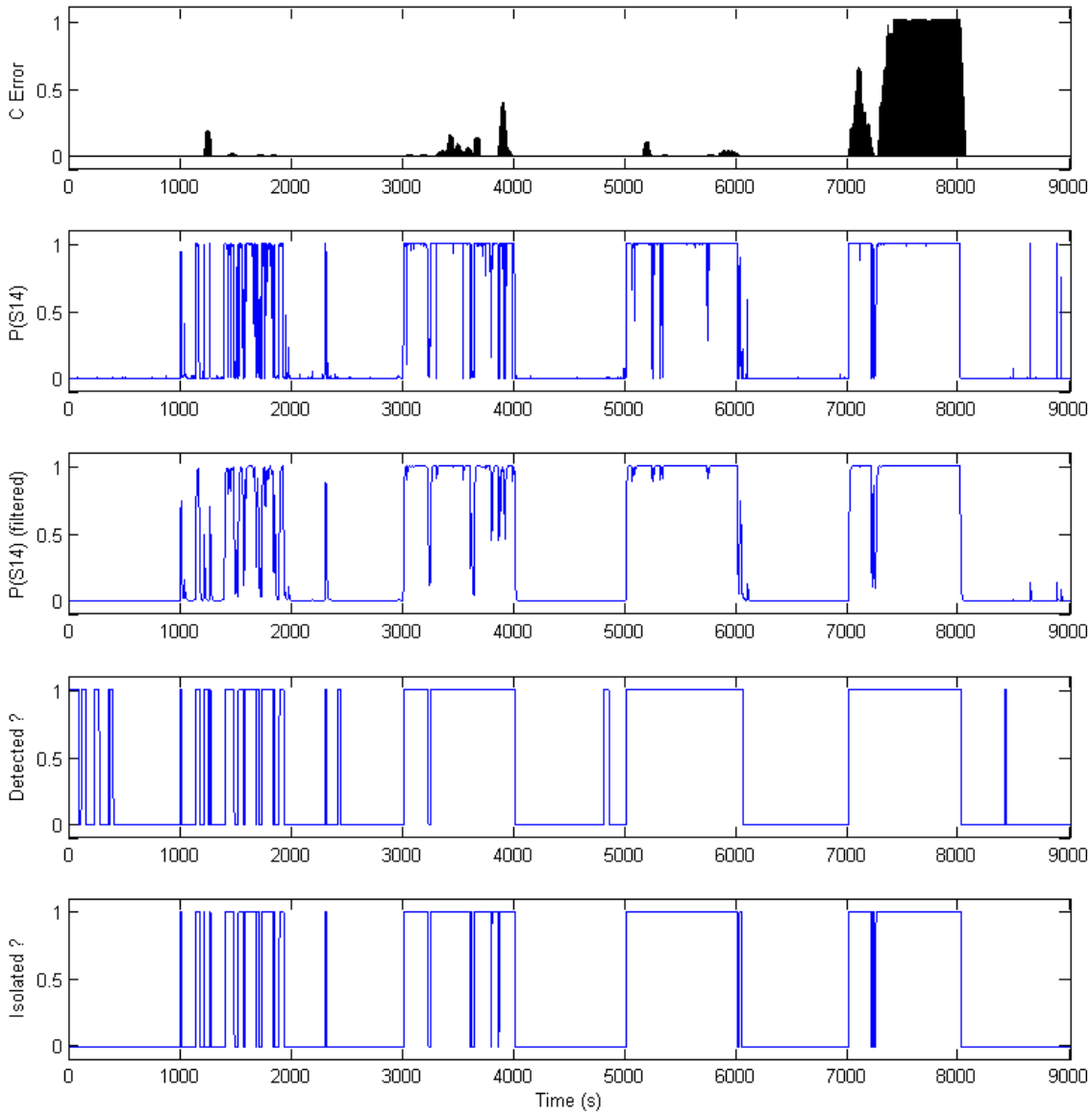

Sensor #13
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S13)
P(S13) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

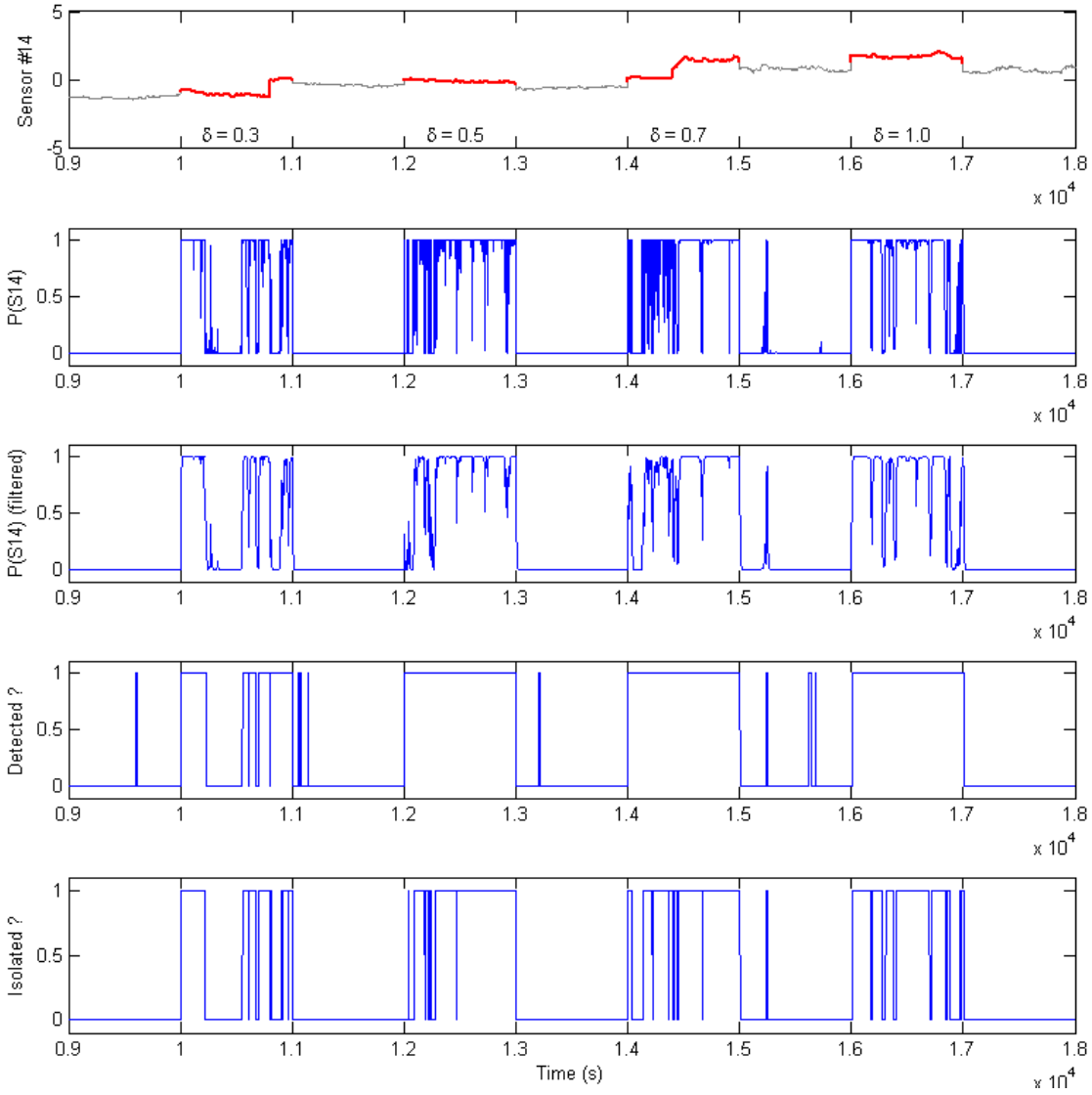

C Error
P(S13)
P(S13) (filtered)
Detected ?
Isolated ?
0
0.5
1
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
x 10^4
Time (s)

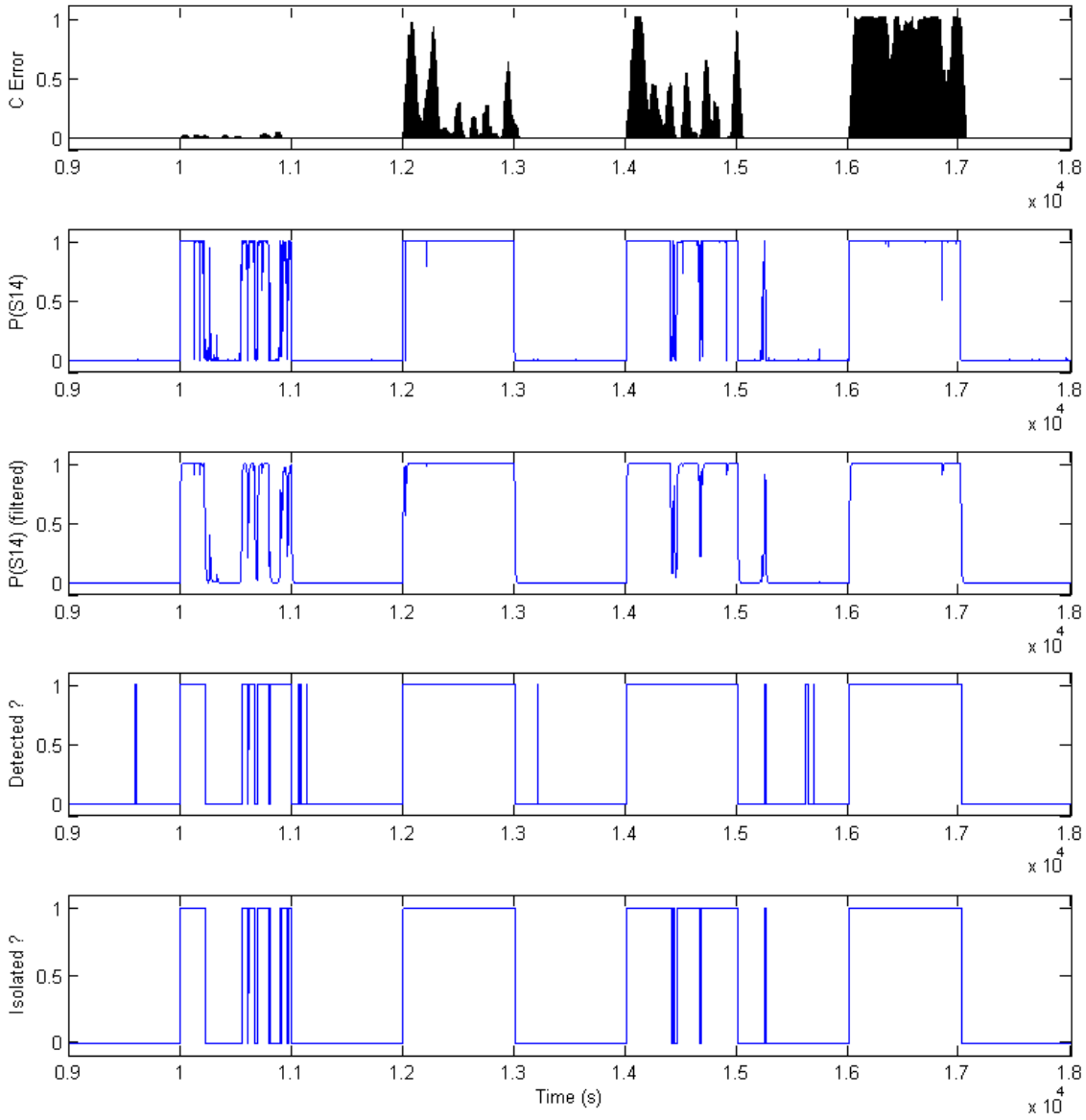

Sensor #13
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S13)
P(S13) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

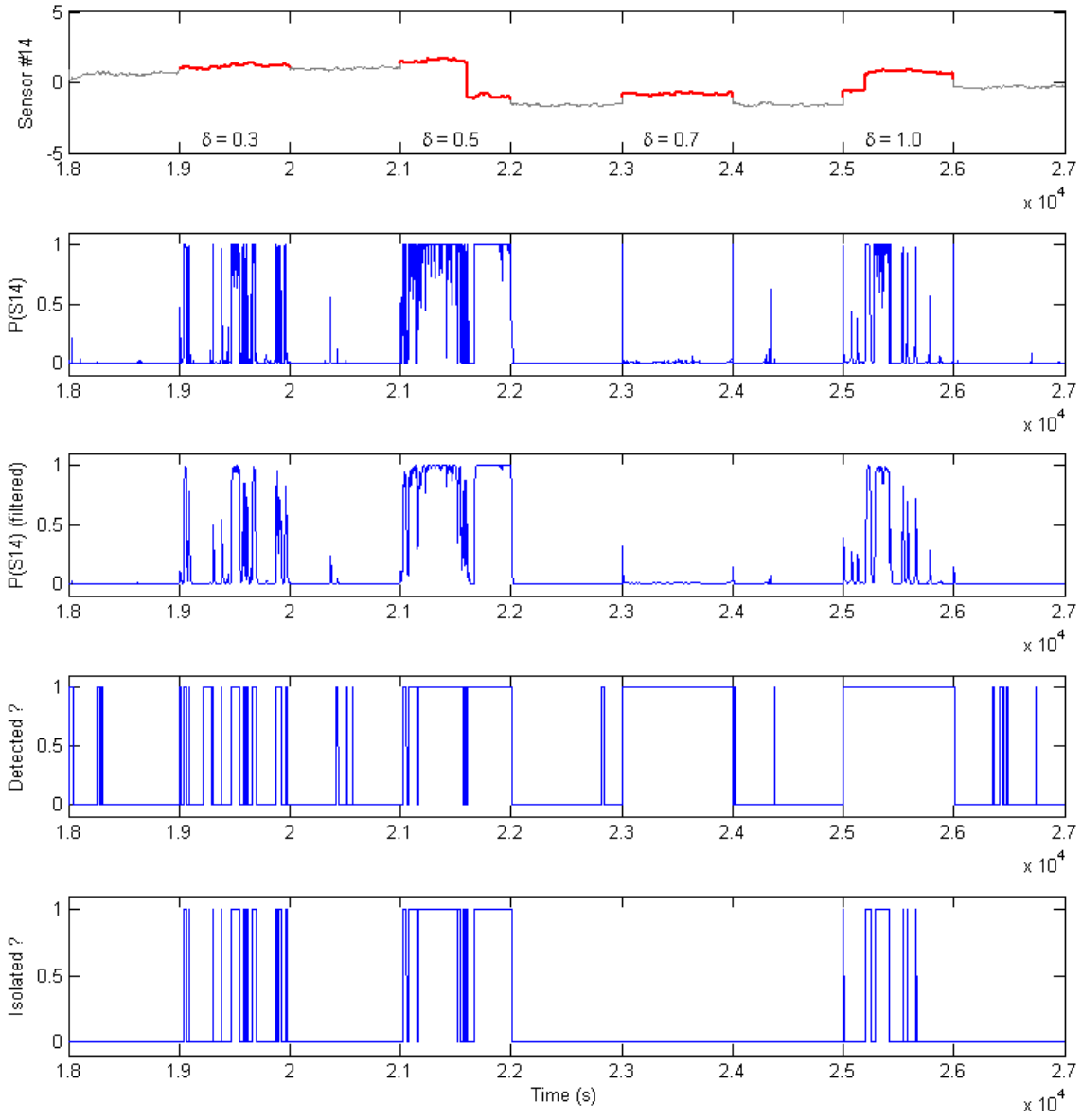

C Error
P(S13)
P(S13) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

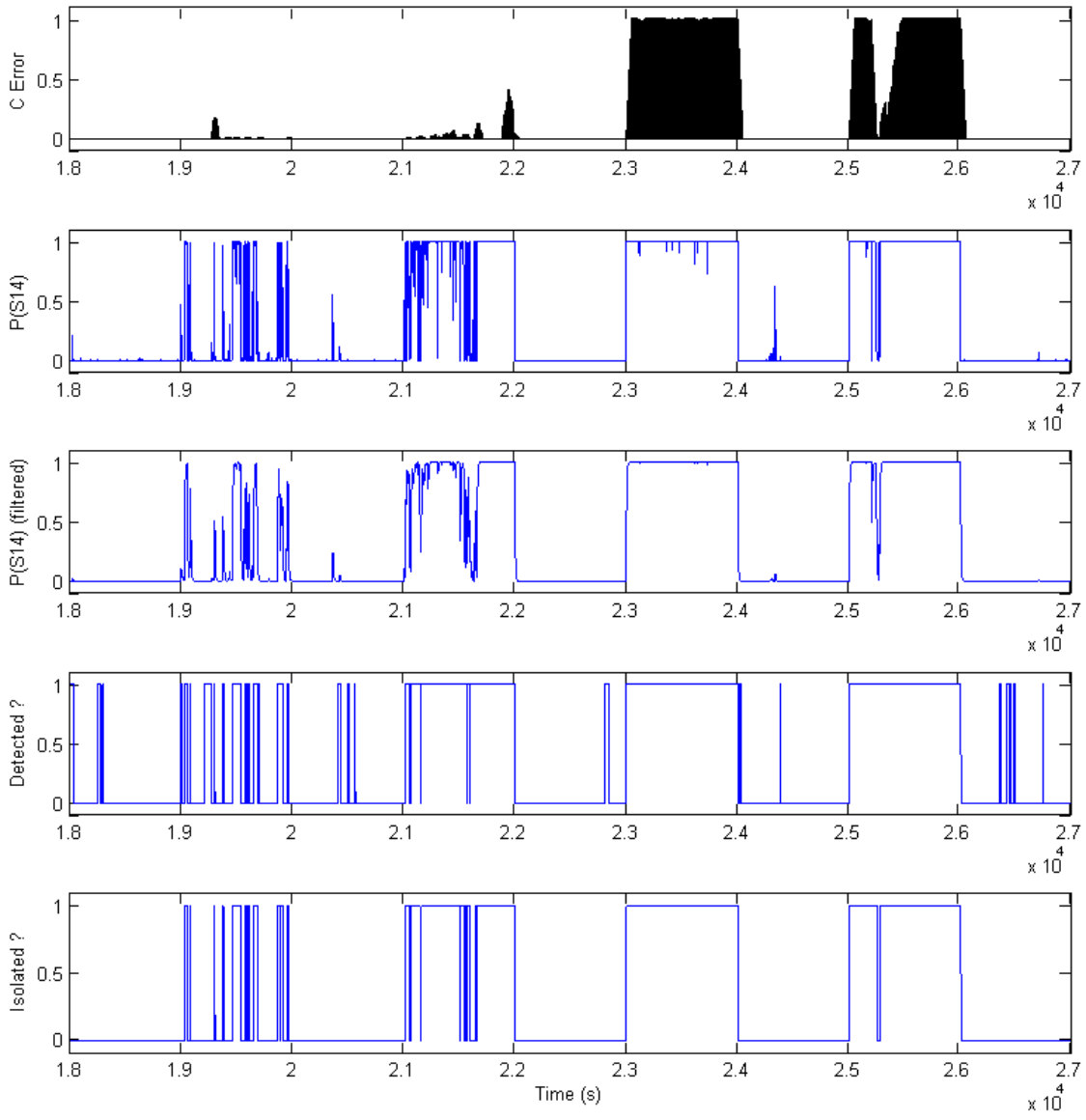

Sensor #14
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S14)
P(S14) (filtered)
Detected ?
Isolated ?
Time (s)

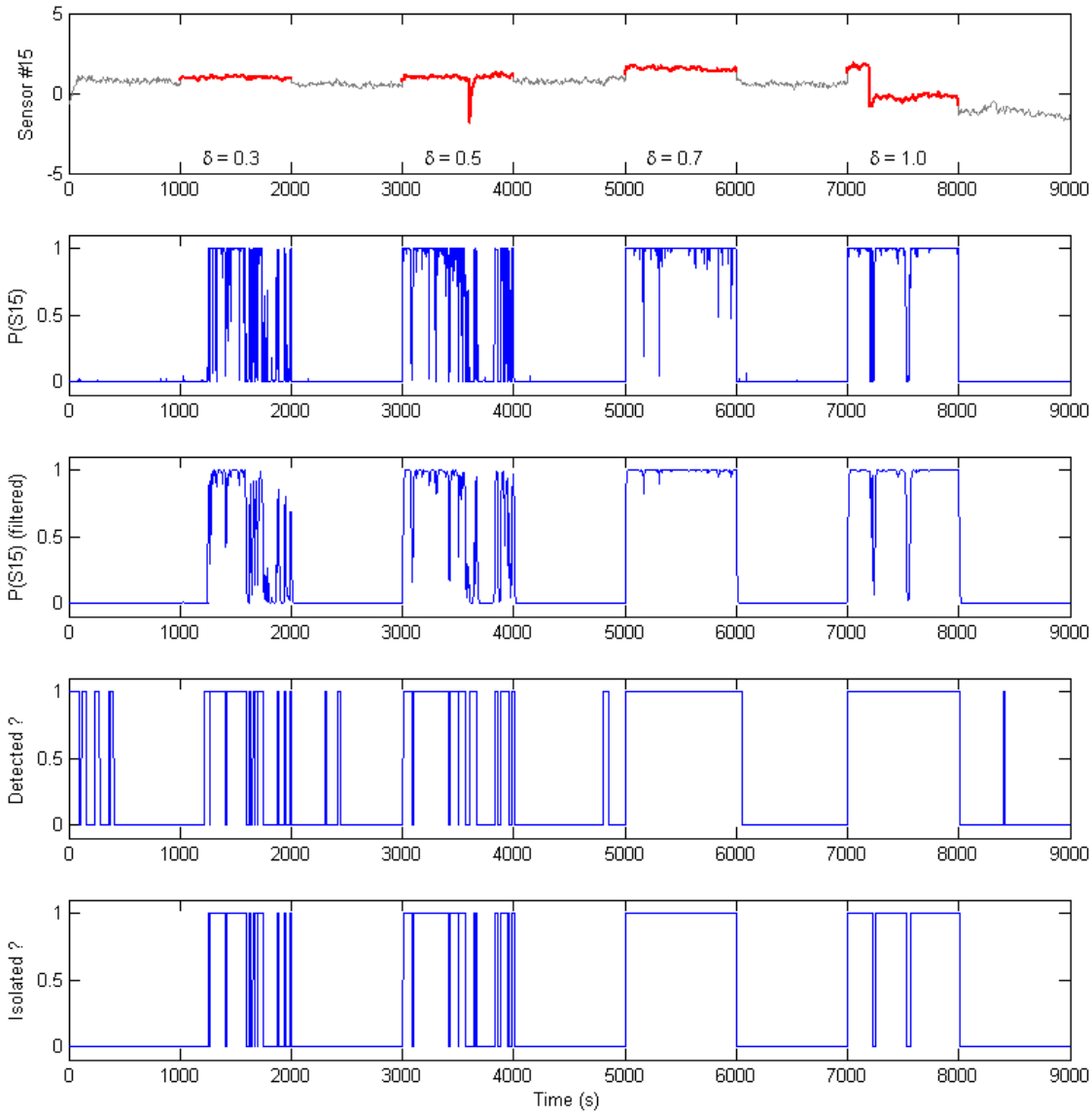

C Error
P(S14)
P(S14) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)

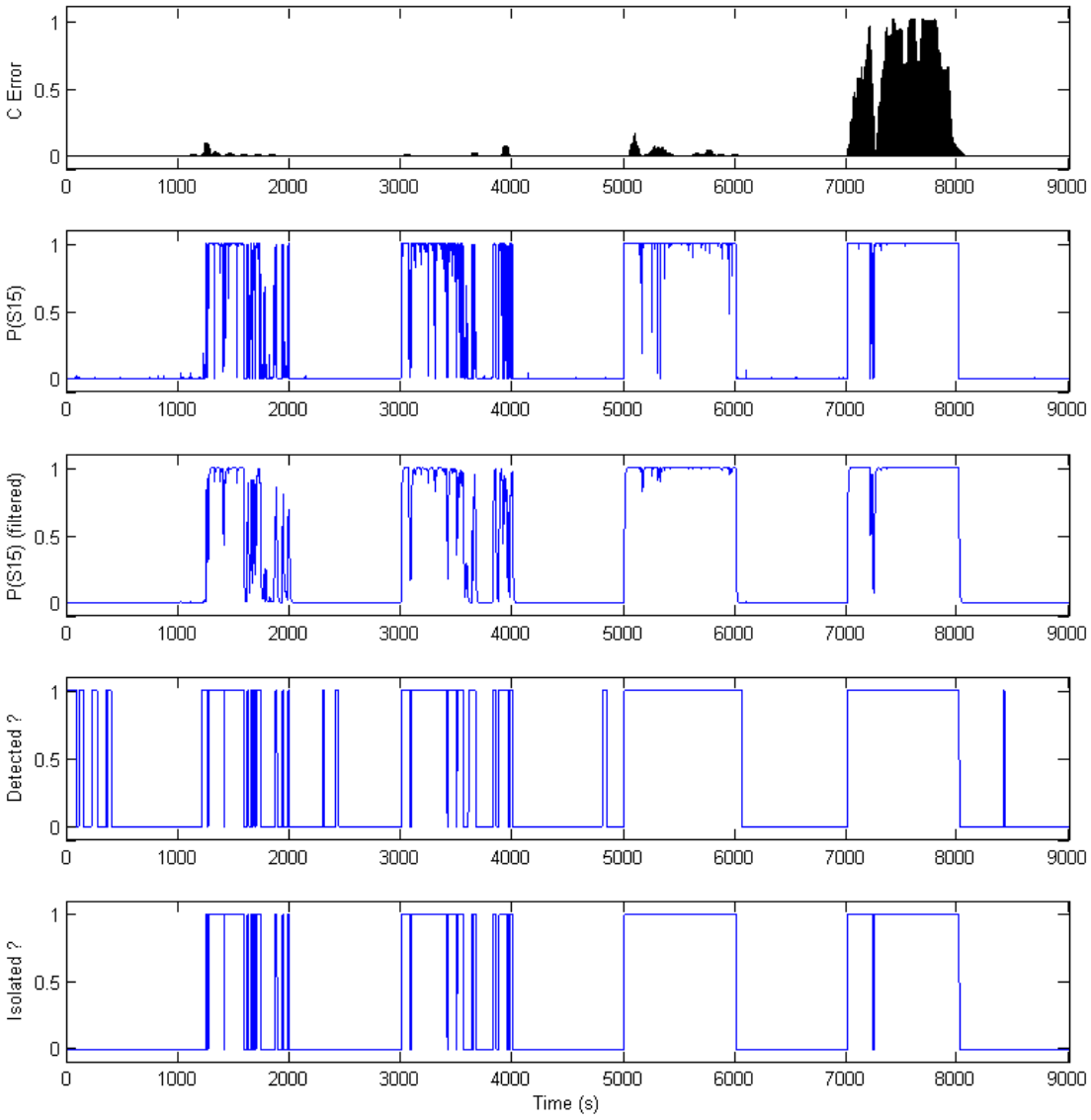

Sensor #14
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S14)
P(S14) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

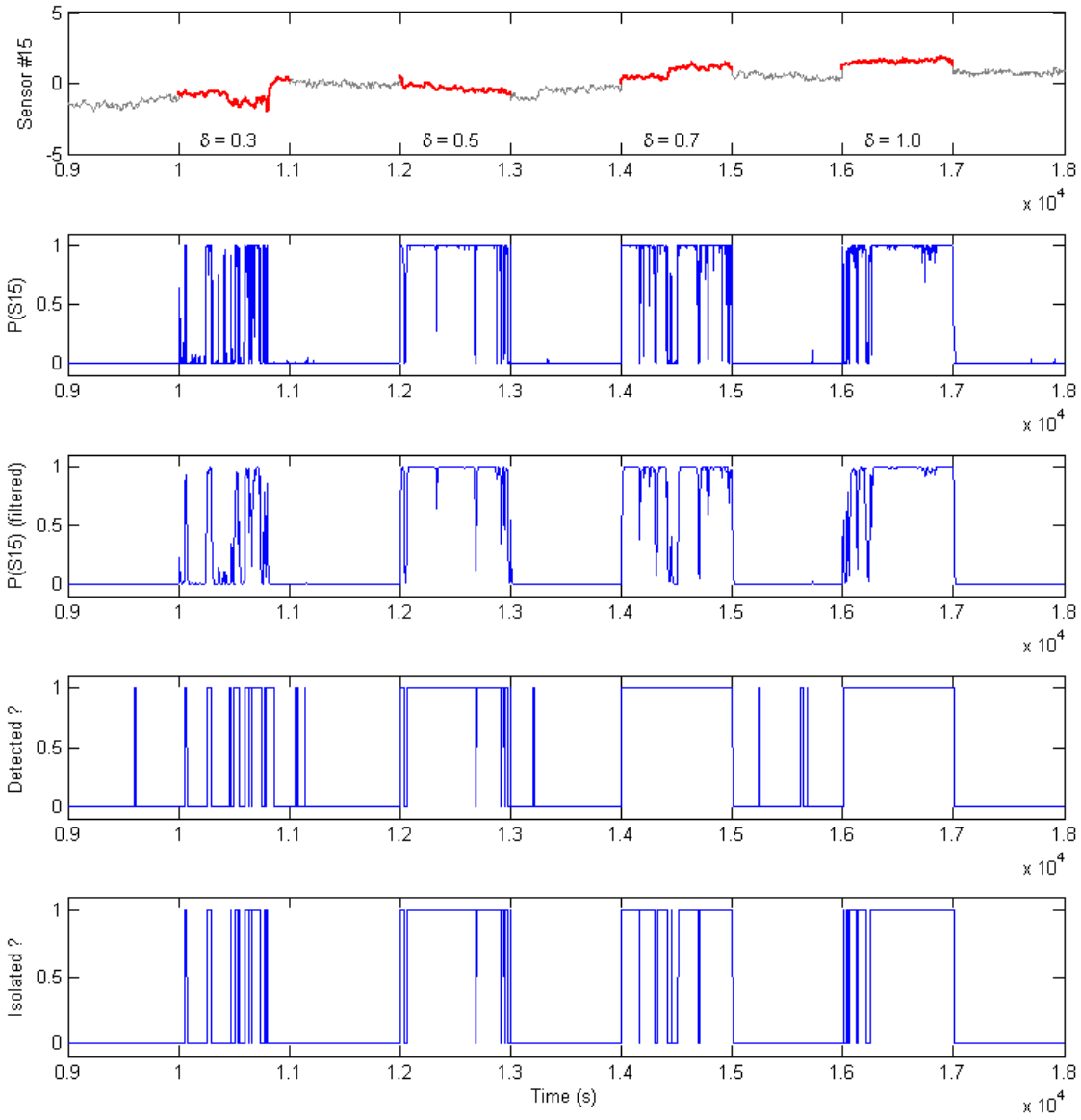

C Error
P(S14)
P(S14) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

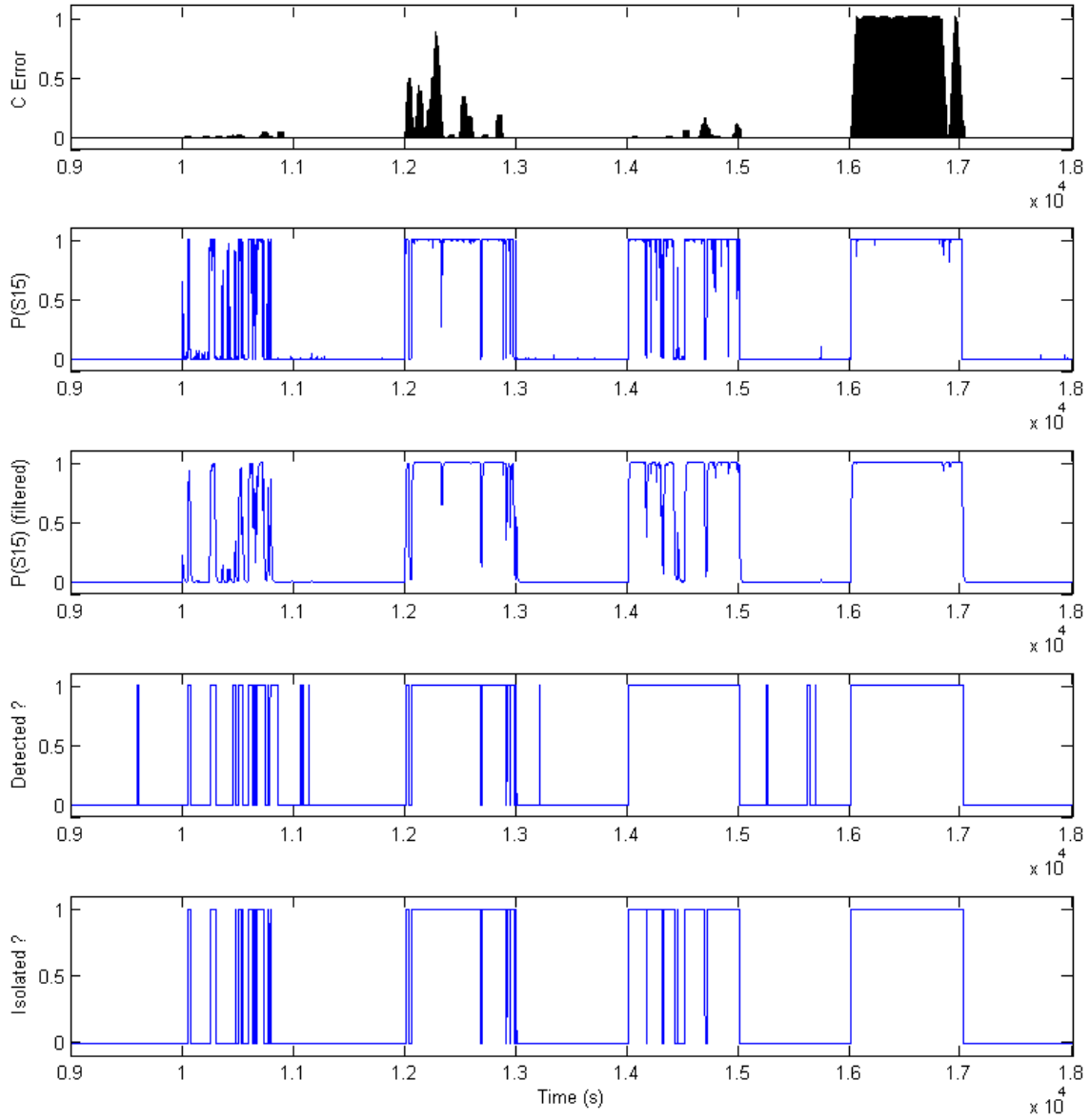

Sensor #14
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S14)
P(S14) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4

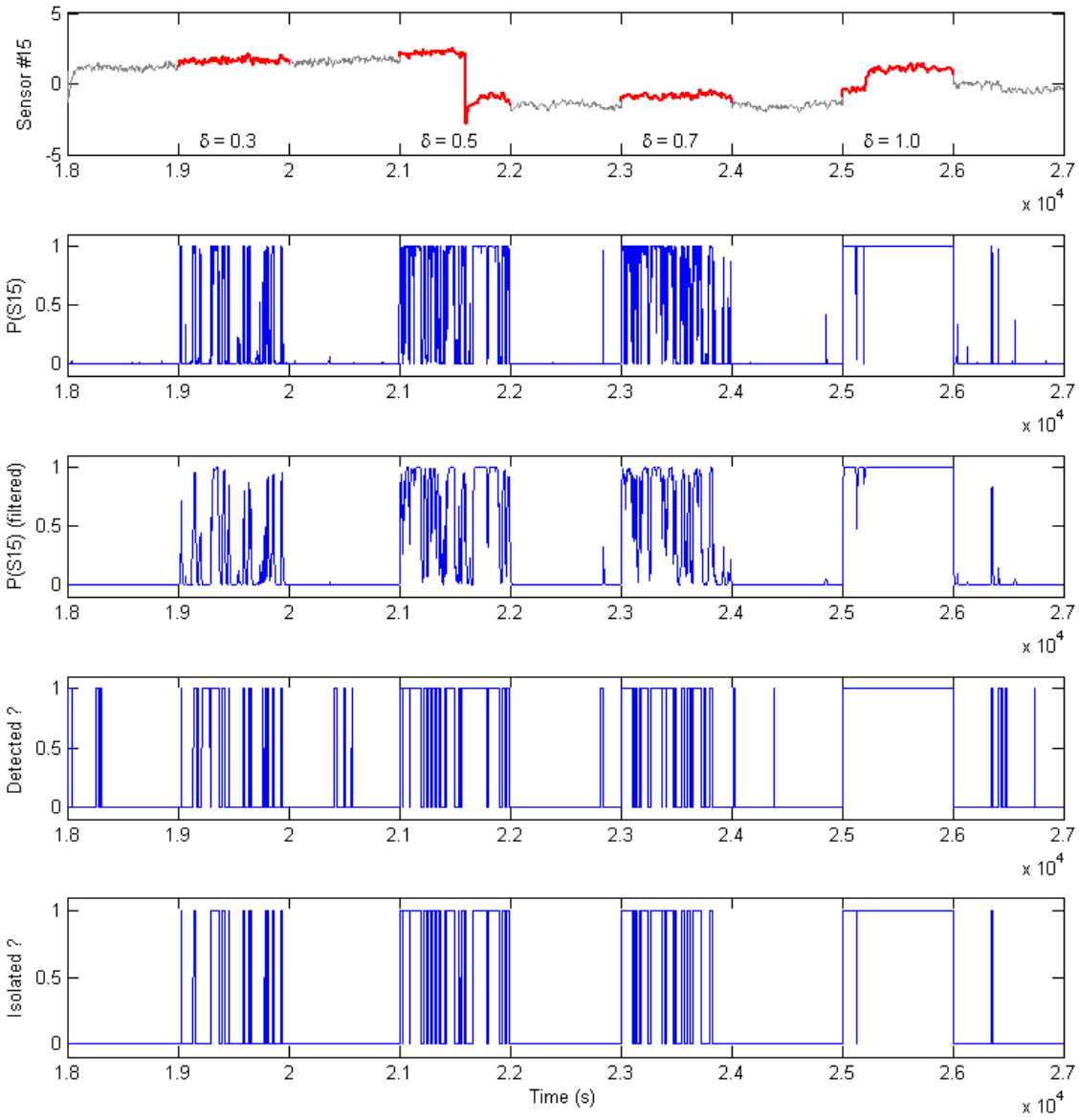

C Error
P(S14)
P(S14) (filtered)
Detected ?
Isolated ?
1
0.5
0
1.8
1.9
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7
x 10^4
Time (s)

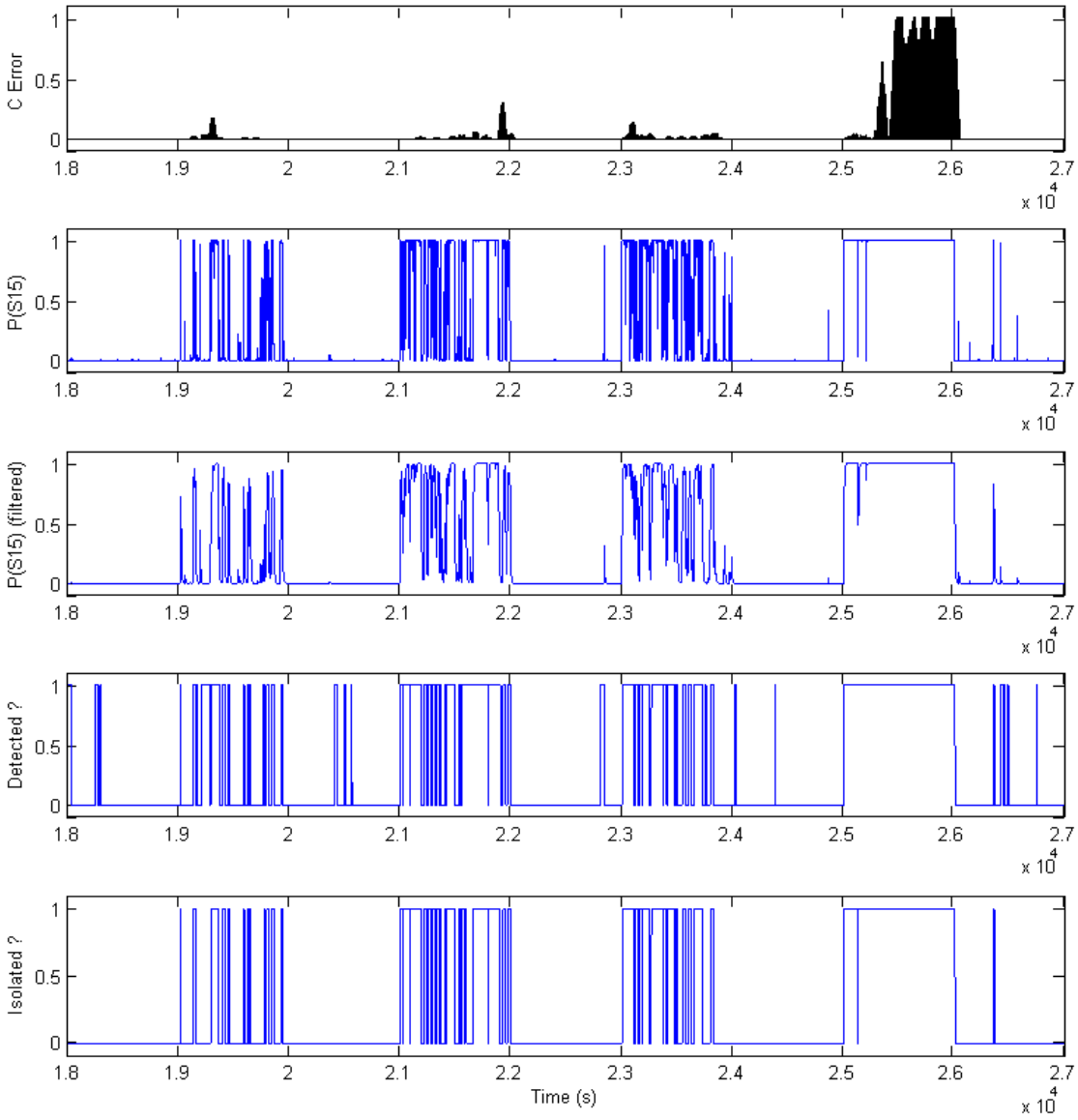

Sensor #15
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S15)
P(S15) (filtered)
Detected ?
Isolated ?
Time (s)


C Error
P(S15)
P(S15) (filtered)
Detected ?
Isolated ?
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time (s)


Sensor #15
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S15)
P(S15) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4


C Error
P(S15)
P(S15) (filtered)
Detected ?
Isolated ?
0
0.5
1
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
x 10^4
Time (s)


Sensor #15
δ = 0.3
δ = 0.5
δ = 0.7
δ = 1.0
P(S15)
P(S15) (filtered)
Detected ?
Isolated ?
Time (s)
x 10^4


C Error
P(S15)
P(S15) (filtered)
Detected ?
Isolated ?
1
0.5
0
1.8
1.9
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7
x 10^4
Time (s)

VITA

Name: Mahmoudreza Sharifi

Address: Department of Mechanical Engineering, University of Connecticut, Storrs CT 06269

Email Address: reza.sharifi@gmail.com

Education: B.S., Aerospace Engineering, Sharif University of Technology, 1997

M.S., Mechanical Engineering, University of Tehran, 2000

Ph.D., Mechanical Engineering, Texas A&M University 2009