Introduction to the Kalman Filter and Tuning its Statistics for Near Optimal Estimates and Cramer Rao Bound

by

Shyam Mohan M, Naren Naik, R.M.O. Gemson, M.R. Ananthasayanam

Technical Report : TR/EE2015/401

arXiv:1503.04313v1 [stat.ME] 14 Mar 2015

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

FEBRUARY 2015

Introduction to the Kalman Filter and Tuning its Statistics for Near Optimal Estimates and Cramer Rao Bound

by

Shyam Mohan M[1]**, Naren Naik**[2]**, R.M.O. Gemson**[3]**, M.R. Ananthasayanam**[4]

[1]Formerly Post Graduate Student, IIT, Kanpur, India

[2]Professor, Department of Electrical Engineering, IIT, Kanpur, India

[3]Formerly Additional General Manager, HAL, Bangalore, India

[4]Formerly Professor, Department of Aerospace Engineering, IISc, Banglore

Technical Report : TR/EE2015/401

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

FEBRUARY 2015

ABSTRACT

This report provides a brief historical evolution of the concepts in the Kalman filtering theory since ancient times to the present. A brief description of the filter equations its aesthetics, beauty, truth, fascinating perspectives and competence are described. For a Kalman filter design to provide optimal estimates tuning of its statistics namely initial state and covariance, unknown parameters, and state and measurement noise covariances is important. The earlier tuning approaches are reviewed. The present approach is a reference recursive recipe based on multiple filter passes through the data without any optimization to reach a 'statistical equilibrium' solution. It utilizes the a priori, a posteriori, and smoothed states, their corresponding predicted measurements and the actual measurements help to balance the measurement equation and similarly the state equation to help form a generalized likelihood cost function. The filter covariance at the end of each pass is heuristically scaled up by the number of data points is further trimmed to statistically match the exact estimates and Cramer Rao Bounds (CRBs) available with no process noise provided the initial covariance for subsequent passes. During simulation studies with process noise the matching of the input and estimated noise sequence over time and in real data the generalized cost functions helped to obtain confidence in the results. Simulation studies of a constant signal, a ramp, a spring, mass, damper system with a weak non linear spring constant, longitudinal and lateral motion of an airplane was followed by similar but more involved real airplane data was carried out in MATLAB$^{\circledR}$. In all cases the present approach was shown to provide internally consistent and best possible estimates and their CRBs.

ACKNOWLEDGEMENTS

Shyam Mohan M (shyammoh.2014@iitkalumni.org)

Naren Naik (nnaik@iitk.ac.in)

R.M.O. Gemson (mogratnam@rediffmail.com)

M.R. Ananthasayanam (sayanam2005@yahoo.co.in)

Contents

List of Symbols^{*}

x	State vector of size $n \times 1$
Θ	Parameter vector of size $p \times 1$
$X = [x, \Theta]^T$	Augmented state vector of size $(n + p) \times 1$
Z_k	Measurement Vector of size $m \times 1$ at discrete time index 'k'
$\mathbf{X_0}, \mathbf{P_0}$	Initial state and its covariance
$\mathbf{R}, \mathbf{Q}$	Measurement and Process noise covariance matrix
$\mathbf{J0\text{-}J8}$	Cost functions
$X_{k\|k-1}$	Prior state estimate at time index k based on data upto k-1
$X_{k\|k}$	Posterior state estimate at time index k based on data upto k
$X_{k\|N}$	Smoothed state estimate at time index k based on data upto N
$Xd_{k\|N}$	Dynamical state estimate at time index k based on data upto N
$P_{k\|k-1}$	Prior state covariance matrix at time index k given data upto k-1
$P_{k\|k}$	Posterior state covariance matrix at time index k given data upto k
$P_{k\|N}$	Smoothed state covariance matrix at time index k given data upto N
K_k	Kalman Gain based on data upto time index k
$K_{k\|N}$	Smoothed Gain based on all data upto time index N
$F_{k-1} = \left[\frac{\partial f}{\partial X}\right]_{X=X_{k-1\|k-1}}$	State Jacobian matrix using posterior state estimate
$F_{k-1\|N} = \left[\frac{\partial f}{\partial X}\right]_{X=X_{k-1\|N}}$	State Jacobian matrix using smoothed state estimate
$Fd_{k-1\|N} = \left[\frac{\partial f}{\partial X}\right]_{X=Xd_{k-1\|N}}$	State Jacobian matrix using dynamical state estimate
$H_k = \left[\frac{\partial h}{\partial X}\right]_{X=X_{k\|k-1}}$	Measurement Jacobian matrix using prior state estimate
$H_{k\|k} = \left[\frac{\partial h}{\partial X}\right]_{X=X_{k\|k}}$	Measurement Jacobian matrix using posterior state estimate
$H_{k\|N} = \left[\frac{\partial h}{\partial X}\right]_{X=X_{k\|N}}$	Measurement Jacobian matrix using smoothed state estimate

Most other symbols are explained as and when they occur in the report.

Chapter 1

Historical Development of Concepts in Kalman Filter

1.1 Randomness from Ancient Times to the Present

It is useful to understand the concept of randomness before introducing the Kalman filter. Randomness occurs and is inevitable in all walks of life. The ontology (the true nature) is one thing and the epistemology (our understanding) is another thing. A computer generated sequence of random numbers is deterministic ontology but for the user who does not know how they are generated it is probabilistic epistemology. Randomness is patternless but not propertyless. Randomness could be our ignorance. Quantum Mechanics seems to possess true randomness. One feels that randomness is a nuisance, and should be avoided. However we have to live with it and compulsively need it in many situations. In a multiple choice question paper no examiner would dare to put all the correct answers at the same place! As another example the density, pressure, and temperature, or even many trace constituents in air can be measured with a confidence only due to the random mixing that invariably takes place over a suitable space and time scale. It is worthwhile to state the introduction of random process noise into the kinematic or dynamical equations of motion of aircraft, missiles, launch vehicles, and satellite system helps to inhibit the onset of Kalman filter instability and thus track these vehicles. Chance or randomness is not something to be worried about presently the most logical way to express our knowledge.

The well known statistician Rao[50] (1987) states that statistics as a method of learning from experience and decision making under uncertainty must have been practiced from the beginning of mankind. The inductive reasoning in these processes have never been codified due to the uncertain nature of the conclusions drawn from any data. The break-

1

through occurred only at the beginning of the twentieth century with the realization that inductive reasoning can be made precise by specifying additionally just the amount of uncertainty in the conclusions. This helped to work out an optimum course of action involving minimum risk in uncertain situations in the deductive process. This is what Rao[50] (1987) has stated as a new paradigm of thinking to handle uncertain situations,

Uncertain knowledge + Knowledge of the amount of uncertainty in it
= Usable knowledge

For our purpose randomness is common to probability, statistics, random process, and estimation theory as can be seen from Fig. 1.1. The Kalman filter can be stated as a sequential statistical analysis of time dependent data. The way it does is to account for the change, capture the new knowledge or measurement, and assimilate it for change as explained in the next section.
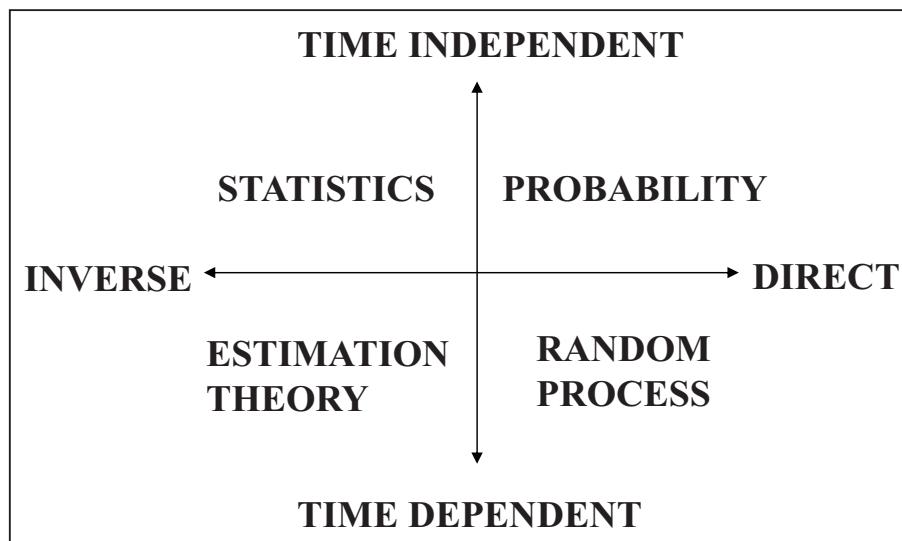
Figure 1.1: Relationship among Probability, Statistics, Random process, and Estimation Theory. Randomness is common to all.

1.2 Conceptual Development of the Kalman Filter

The triplets of change, capture and correct forms the conceptual basis for the Kalman filter and it can be stated as follows :

Change + Capture $\longrightarrow$ Correct

The '+' sign above has a deep significance in the way the present and the new information are combined to have progress in the correct direction based on an appropriate criterion. The above can be stated among other possible ways in science and technology such as

$$\text{Present} + \text{New} \longrightarrow \text{Updated information or knowledge}$$

$$\text{Theory} + \text{Experiment} \longrightarrow \text{Progress}$$

$$\text{Model} + \text{Data} \longrightarrow \text{Estimate the unknowns}$$

$$\text{Intuition} + \text{Experiments} \longrightarrow \text{Rigorous Mathematical Framework}$$

The origin of such an ubiquitous Kalman filter (KF) in Estimation Theory (ET) can be traced since ancient times. Its progress is similar to any physical theory moving randomly across intuition, experiments and mathematical framework.

1.3 Estimation Theory in Ancient Indian Astronomy

One of the simplest examples of estimation is to observe one quantity and infer another quantity connected to it by a certain relation which may be empirical or theoretical. Thus obtaining the 'unobservables' from the 'observables' is a very subtle concept that can be stated as 'observe one and infer another'. For example by making kinematic measurements such as the position, velocity and accelerations on a system one can infer the internal dynamical properties of the system such its mass, moment of inertia, aerodynamic characteristics by suitable modeling.

The ancient Indian astronomers at least since AD 500 used the above concept to update the parameters for predicting the position of celestial objects for timing their Vedic rituals based on measurements carried out at various time intervals which can be stated as

$$\text{Updated parameter} = \text{Earlier parameter} +$$

$$\text{(Some quantity)} \times (\text{ measured - predicted}) \text{ Position of the celestial object}$$

The 'some quantity' as we will see later on is the Kalman gain. Note the measured longitude of the celestial object is different from the state that is updated which is the number of revolutions in a yuga just as state and measurements are different in many Kalman filter applications!

The earliest astronomical manual Suryasiddhantha (Burgess[2] 1935) is dated before AD 500. When deviations were noted from its predictions the ancient Indian astronomers revised the parameters of the planetary positions based on observations at later times. Aryabhata made independent observations and gave corrections to it that provided accurate results around his time of AD 522. The constants or cannons of Aryabhata was revised by a group of astronomers in AD 683-684 known as Parahita system. After a long time lapse large deviations occurred from Parahita system and Parameswara of the Kerala School of astronomy revised it based on astronomical observations as promulgated in his work Drgganita in AD 1431.

The next revision was by Nilakantha as stated in his work Tantrasangraha (Ramasubramanian and Sriram[107] 2011) in AD 1443. Nilakantha had stated "the eclipses cited in Siddhanthas can be computed and the details verified. Similarly other known eclipses as well as those currently observable are to be studied. In the light of such experience future ones can be computed and predicted (**extrapolation!**). Or eclipses occurring at other places can be studied taking into account the latitude and longitude of the places and on this basis the methods providing the parameters for the Sun, Moon can be perfected (**data fusion!**). Based on these, the past and future eclipses of one's own place can be studied and verified with appropriate refinement of the technique." This is just the idea of **'smoothing'!** When Tantrasamgraha too was becoming inaccurate, observations were carried out by astronomers on the west coast for thirty years from AD 1577 to AD 1607 and a revision was promulgated in AD 1607 in Drkkarana which quotes about all the above earlier revisions (good literature survey!). It stated that henceforth also deviations would occur and these should be carefully observed.

Not all the above canons were based on the latest observations. The well known historian Billard[19] (1971) noted that many canons were evolved and one such canon around AD 898 had a very high accuracy valid over a larger number of centuries. He inferred that these must have been based on **the astronomical elements of an earlier time and the new observations of the later time.** A more accurate canon valid over a larger time period is not possible without a weighted addition of the earlier canons (which are the ones that are available since no back dated observations are possible!) and present observations. They must have chosen such a weight or in other words the gain 'K' subjectively to help in giving weights to earlier astronomical parameters and the present measurements. The present day Kalman filter is packed with many desirable and fashionable subjective

assumptions regarding the state, measurement and noise characteristics to facilitate mathematical tractability (and thus obtain the Kalman gain) and the capability to handle far more involved applications. The Table-1.1 provides some of the well known updates of the planetary parameters by ancient Indian astronomers.

It is highly worthwhile to translate the works of Kerala astronomers and Billard, use the best available programs to calculate the earlier positions of the celestial objects to understand how the updates were done.

1.4 Estimation Theory during the Medieval Period

The asteroid Ceres sighted by Piazzi on the first day of 1801 was tracked subsequently for 41 days after which it vanished behind the Sun's rays. Piazzi's data covering only 9 degrees of arc of the celestial sphere consisted of the right ascension, declination and time was published in September of that year. Its orbital elements could not be determined by the then available methods. Newton had stated it as the most difficult nonlinear problem then in astronomy. Gauss made calculations using the observational data, estimated its orbit and sent it to Piazzi who found it again on the last day of 1801 (Tennenbaum and Director[93] 2006) .

This was possible based on the Least Squares (LS) technique used by Gauss (Dover[11] 1963, reprint of Gauss 1809) to make calculations regarding the trajectory and predict its location later in time. Independently the method of LS had been discovered and published by Legendre of France and Robert Adrian of the United States. In fact, even before Gauss was born, the physicist Lambert had used it! (Grewal and Andrews[100] 2008).

Gauss had provided almost all the foundations of estimation theory. He postulated that a system model should be available, minimum number of measurements for observability, the redundant data helping to reduce the influence of measurement errors, a cost function based on the difference between the measurement and that predicted by the model should be minimized. There should also be some a priori knowledge concerning the unknowns to be estimated. Further since the errors could be unknown or unknowable Gauss had given hints about probabilistic approach, normal distribution, method of maximum likelihood estimation, linearization and the Gaussian elimination procedure.

Gauss did not balance the governing differential equation, but tried to fit the measure-

ment with the prediction. If he had tried the former, he would have been led to a biased solution! Fortune favours the brave! This is where a proper mathematical framework helps to understand if an algorithm converges to the correct value with more and more measurements. Begin with intuition, try out with some examples and finally cast in a mathematical framework. Mathematics itself grew like this, but present day students are taught the other way round!

1.5 Estimation Theory (ET) During the Twentieth Century

A very general characterization of ET when time independent data analysis is made it is statistics and when sequential processing of the time dependent data is carried out it is the Kalman filter as mentioned earlier. The post Gaussian contributions in estimation theory consists of the method of moments, method of maximum likelihood estimation, the Kalman filter and its variants, frequency domain approach and the capability to handle time varying state and parameters. As far as experiments, the testing times are short and accurate, with optimal inputs to excite the system to obtain the best possible parameter estimates have been developed. Further the use of matrix theory, and real time processing by computers exist. We are dealing with more difficult situations, but the conceptual framework to solve these problems had been fully laid out by Gauss!

The Kalman filter equations are recursive and it is helpful to estimate the time varying state variables and parameters. The sequential least squares was rediscovered by Plackett[3] (1950) and Kalman[7] (1960). Thus in fact Sprott[40] (1978) had questioned if the Kalman filter is really a significant contribution! The point is the frequency domain approach of the Wiener Filter[4] (1949) has been enhanced to the natural time domain approach by the Kalman filter. Further the shift from batch to the sequential approach is very convenient to handle continuous measurement data flow. It is to the credit of Kalman that apart from unifying earlier results that he introduced the concept of controllability and observability to make the estimation problem well posed. The only slight difference, but very momentous between the Recursive Least Squares (RLS) and the Kalman filter is the time propagation of the state and covariance estimates before their update by using the measurements.

During the mid-twentieth century the Kalman filter in discrete time (Kalman,[7] 1960), and in continuous time (Kalman and Bucy,[8] 1961) helped the Apollo project to the Moon. It

is interesting to note that during the periods of its development whether in mythological, medieval, or modern times its connection with celestial objects is remarkable.

1.6 Present day Applications of Estimation Theory

Presently the scale and magnitude of many difficult and interesting problems that ET is handling could not have been comprehended by ancient Indian astronomers, Gauss or Kalman. In many present day applications one does not even know quite well the structure of the state and measurement equations as well as the parameters in them and the statistical characteristics of the state and measurement noises. It is possible to add the unknown initial conditions of the state as well. One can summarize that almost nothing is known but everything has to be determined or estimated from the measurements alone! This means the connecting relationship between the state and the measurement has to be found out. This has to be from previous knowledge or intuition such that it is meaningful, reasonable, acceptable, or useful. Even this has to be achieved only by the internal consistency of the various quantities and/or the variables that occur at different times during the filter operation through the data. Further the systems are large and the best possible optimal estimation has to be worked out in real time. However due to the enormous computing power that is presently available it has been possible to handle the above situations.

Typical present day applications of the Kalman filter include target tracking (Bar-Shalom et al.[79] 2001), evolution of the space debris scenario (Ananthasayanam et al.[91] 2006), fusion of GPS and INS data (Grewal et al.[99] 2007), study of the tectonic plate movements (Kleusberg and Teunissen[62] 1996), high energy physics (Fruhwirth et al.[75] 2000), agriculture, biology and medicine (Federer and Murthy[66] 1998), dendroclimatology (Visser and Molenaar[52] 1988), finance (Wells[63] 1996), source separation problem in telecommunications, biomedicine, audio, speech, and in particular astrophysics (Costagli and Kuruoglu[97] 2007), and atmospheric data assimilation for weather prediction (Evensen[102] 2009).

1.7 Three Main Types of Problems in Estimation Theory

If we denote the system model as (S), measurement model as (M), state noise as ($\mathbf{Q}$) and measurement noise as ($\mathbf{R}$) three main types of problems emerge in estimation theory

(Klein[42] 1979). Generally the state equations are differential equations and the measurement equations related to the states are algebraic in nature. Both the above could have unknown or inaccurately known parameters and noise characteristics which will have to be estimated.

The first dealing with balancing the state differential equations without specifying the characteristics of the process noise is known as the Equation Error Method (EEM). The simplest EEM formulation is obtained by assuming that noise free state variables and noisy state derivatives measurements are available (Maine and Iliff[46] 1981) to balance the governing state differential equations. If the measurements of the states are noisy then EEM leads to biased parameter estimates. However if the noise in the state measurements are very small then good parameter estimates are still possible. The EEM is amenable for solution by simple least squares technique.

The second combination of state, measurement and measurement noise forms the Output Error method (OEM) thus with no process noise (Klein[42] 1979). The OEM formulation matches the predicted measurement based on the states obtained from the state differential equations with the actual measurements. The matching based on the squared difference between the actual measurement and the predicted measurement summed over all the time points leads to a cost function $\mathbf{J}$ that can be minimized by any suitable batch processing optimization algorithm. The minimization of $\mathbf{J}$ leads to the estimates of the unknown parameters in both the states and measurements. It may be mentioned that the results from the above approach is used as an anchor in the present work for tuning the Kalman filter statistics when there is no process noise.

The most general third formulation is when the state and the measurement equations with both the state and measurement noises are present. Before an experiment is carried out we can talk of probabilistic outcomes. Once the data is available the unknown parameters and quantities can be treated as deterministic unknowns. It is useful to read the discussion in Rao[50] (1987) wherein he describes a random trace of a signal as the sum of many deterministic functions. Thus once again a matching of the measurements with those derivable by propagating the state differential equations with estimated parameters has to be carried out. Since the measurements are random they could correspond to random states. The propagation of the state being also random there is a difference between the predicted and updated states and this difference can be used to derive a cost function from the state differential equations which also can be minimized. Since the contributions from the various

states and the measurements have different units it is desirable to normalize each contribution using their mean square values. Thus one can treat estimation operationally as a deterministic optimization problem for a given measurement data. One should note that even from deterministic approaches, based on the second derivatives of the cost function (based on the Hessian, the first derivative of $\mathbf{J}$ being zero at the minimum) a measure of the uncertainty for all the estimated unknown quantities can be obtained. In all the above formulations if a certain suitable probability distribution is not assumed for the $\mathbf{Q}$ and $\mathbf{R}$ then it is deterministic. Another simple example is a constant signal added with noise. It is trivial to get an estimate of the mean and standard deviation denoting the effect of dispersion. If only the noise is assumed to be Gaussian then quantitative values can be attached to the standard deviation of the estimate.

However if the state and measurement noises are characterized as random variables (in say the simplest form being zero mean, white and Gaussian) then all the above three types of estimation problems can be cast in a probabilistic framework and one of the most prolific is the Method of Maximum Likelihood Estimation (MMLE) helps to attach quantitative values for the uncertainties of the estimates. Due to the existence of the state noise the state propagation is not a deterministic trajectory but becomes a random process. This predicted state has to be statistically combined with the noisy measurements at every point based on a suitable criterion and updated at every measurement and then again propagated. This is the Kalman filter formalism as will be elaborated in the next Chapter.

When the parameters in the state and measurement equations are treated as augmented state it is called as the Extended Kalman Filter (EKF). Either the MMLE or EKF both of which are equivalent can be utilized for the estimation of unknown parameters and noise characteristics. For the sake of mathematical tractability and simplicity the various random variables in the Kalman filter are assumed to follow a Gaussian distribution. Thus the initial states are assumed to follow a multidimensional Gaussian distribution. If the governing state equations are linear then, only the parameters in the Gaussian distribution change. Then if the distribution of the measurement noise is Gaussian then after statistically combining the above once again leads to a Gaussian distribution for the updated state which is again propagated in time. The process is repeated till all the measurement data are utilised.

However when the state differential equations are nonlinear then between the measure-

ments the time propagated Gaussian distribution becomes non Gaussian. It is then a suitable linearization of the state and measurement (if nonlinear) are carried out to keep the formalism similar to the linear case. This is known as the Extended Kalman Filter (EKF) formulation to handle nonlinear systems and measurements. In this report we consider the problem of estimating the unknown parameters and also **Q** and **R** if these are also unknown. This is also called the problem of tuning the Kalman filter statistics.

Generally, in the MMLE for dynamical systems, one deals with mainly where the measurement noise alone is present. However in many present day applications, modeling errors and random state noise input conditions occur. Hence it becomes compelling to deal with MMLE including the process noise as well. In such a situation one needs to estimate or account for the process noise while estimating the unknown parameters. This makes the problem difficult by an order of magnitude due to the requirement of more computer memory and time as well as the convergence difficulties of the various algorithms. Thus it is generally far more difficult to handle the estimation problem in the presence of unknown or uncertain process and measurement noises.

The MMLE is widely used, since it yields realistic results for practical problems and its estimates have many desirable statistical properties such as consistency, efficiency and sufficiency. The numerical effort in the EEM, OEM and the most general MMLE or EKF options are of order 1, 10 and 100 respectively.

1.8 Summary of different aspects of Estimation Theory

The basic framework of Estimation Theory (Eykhoff[47] 1981) consists of

1. Modeling the system, measurement and all the noise characteristics.

2. A criterion to match or mix the model output with the measurements.

3. A numerical algorithm for the above task and consequently obtain the estimates and the uncertainty of the estimated quantities and

4. An internal consistency check to ensure that all the above steps are consistent and if not shows the need for modification.

The first aspect requires some 'a priori' knowledge about the system under investigation. The model can be true or best or adequate and generally one aims for the last one. The

model in the form of mathematical equations is expected to characterize the essential or desired aspects of the state process.

The second needs the matching of model output with the measurements and it can be based either on deterministic or probabilistic criterion. In a comprehensive way the model is made to follow the output measurements under the given input conditions. In order to match in a quantitative sense between the postulated model and the measurements, one has to choose a criterion function or a cost function $\mathbf{J}$.

The third step is the selection of a numerical algorithm to satisfy the above criterion. Generally the matching (or combining) of the model output with the measurements have to be carried out by a suitable optimization technique to minimize the above cost function.

The last one is the process of model validation. The task of model validation is crucial to understand the adequacy of the postulated state and measurement models and the convergence of the numerical algorithm. Statistical hypothesis testing procedure is the main tool to check the consistency of all the aspects of the estimation process.

The above four aspects help to evolve a suitable mathematical model of the system by properly estimating the unknown parameters and other noise characteristics based on the input and measurement data. Subsequently such a model can be utilized in further studies like prediction, control and system optimization. The present work concentrates in particular on the Step 3 of the ET framework.

Table 1.1: Corrections of Planetary Parameters in Ancient India.

Credit : Ananthasayanam and Bharadwaj[105] 2011

COMPARATIVE STUDY OF THE REVISIONS

The parameters according to the different revisions mentioned above has been set out in the following Table, with the textual verses reproduced as footnotes for verification. This should enable a two-pronged enquiry being made into the subject, first towards identifying the exact methods by which each revision was made and secondly towards determining the relative merit of the revisions.

Table

Sourcebook	Date A.D.	Sun	Moon	Moon's Apogee	Moon's Node	Mars
Āryabhaṭiyam[13]	499	43,20,000	5,77,53,336	4,88,219	2,32,226	22,96,824
Parahita of Haridatta[14]	563	,,	,, 314·2	,, 121·9	,, 307·3	,, 862·3
Dṛgganita of Parameśvara[15]	1431	,,	,, 321·009	,, 123·229	,, 297·832	,, 862·959
Tantrasaṃgraha of Nīlakaṇtha[16]	1443	,,	,, 320	,, 122	,, 300	864
Golasāra of Nīlakaṇtha[17]	c. 1486	,,	,, 320	,, 120	,, 300	864
Siddhāntadarpaṇa of Nīlakaṇtha[18]	c. 1500	,,	,, 332·321	,, 123·318	,, 226·745	,, 862·137
Dṛkkaraṇa[19]	1607	,,	,, 320	,, 122	,, 298	,, 863

Table *(Continued)*

Sourcebook	Mercury	Jupiter	Venus	Saturn	No. days per yuga
Āryabhatiyam	1,79,37,020	3,64,224	70,22,388	1,46,564	1,57,79,17,500
Parahita of Haridatta	,, 378.0	,, 184·0	,, 257·8	,, 581·0	,,
Dṛgganita of Parameśvara	,, 072·112	,, 172·296	,, 270·775	,, 626·695	,, 517·019
Tantrasaṃgraha of Nīlakaṇtha	,, 048	,, 180	,, 268	,, 612	,,
Golasāra of Nīlakaṇtha	,, 040	,, 180	,, 270	,, 600	,,
Siddhantadarpaṇa of Nīlakaṇtha	,, 120·175	,, 160·611	,, 270·552	,, 571·016	,, 839·500
Dṛkkaraṇa	,, 100	,, 166	,, 272	,, 612	,,

Chapter 2

Introduction to the Kalman Filter

2.1 Important Features of the Kalman Filter

Due to the seemingly unpretentious fact of splitting the state and measurement equations and switching between the state propagation and its update using the measurements, very interesting outcomes have been shown to be possible. Any amount of deep study and understanding of the state or the measurement equations separately may not be able to comprehend the exciting possibilities and abilities when both are combined together. This is similar to the components of a watch, or the cells in an organism leading respectively to the time keeping ability or life, which do not exist in the individual components. The GPS is another brilliant example of such a synergism. The competence of the Kalman filter is similar to the saying 'wholes are more than the sum of their parts' as stated by Minsky[51] (1988). It is the above feature that can be called as synergistic, parallel, operator splitting, or a combination of theory and experiment that is the remarkable and profound aspect of the Kalman filter rather than describing it as a sequential least square estimator, or capable of handling time varying states and measurements.

As mentioned in the beginning the triplets of change, capture and correct form the Kalman filter. The filter can be viewed or understood from different perspectives. The modeling of the state of a system is subjective (or in other words intuitive) and the system measurements are objective. Generally the knowledge being uncertain (or inaccurately known) and the measurements are inaccurate or corrupted by noise the Kalman filter combines the two to expand the knowledge front. Another way to look at the Kalman filter is that it combines or assimilates the information from two sources namely uncertain system and measurement models in a statistically consistent way. One other way of understanding

13

the Kalman filter is that it matches the model and the measurement and in the process improves both by suppressing the noise in the measurement improves the accuracy of the state and the parameters in it. There could be many ways or criteria of combining the model and the measurements. Each one could give different results but the criterion to accept any result is that the estimates should be meaningful, reasonable, acceptable and useable. Thus one should note that one cannot be at the truth but around the truth. The only way to reach the truth or in other words get to know the absolute source from which the data has come about is to have infinite data together with an algorithm being capable of reaching the truth.

2.2   Importance of Proper Data Processing

In this section we discuss the importance of proper processing of the data with a simple example. Consider the estimation of the parameter 'a' in the equation

$$y = ax \tag{2.1}$$

A set of N noisy measurements of x, y are given by

$$x_m = x + w$$

$$y_m = y + v$$

The mean of the random measurement noises are assumed to be zero. In order to estimate 'a' one can use different formulae all of which look reasonable such as

$$a = \frac{1}{N} \sum (\frac{y_m}{x_m}) \qquad\qquad a = \frac{1}{N} \sum (\frac{y_m}{\sum x_m})$$

$$\frac{1}{a} = \frac{1}{N} \sum (\frac{x_m}{y_m}) \qquad\qquad \frac{1}{a} = \frac{1}{N} \sum (\frac{x_m}{\sum y_m})$$

 Also the Least square (LS) estimate is $a = \sum(x_m y_m) / \sum(x_m x_m)$. Substituting for x_m and y_m into the above and simplifying with further assumption $w \ll x$ and $v \ll y$ and expanding in a Taylor's series and truncating one may note that even if N tends to infinity only some of the above provide unbiased estimates assuming that the mean value of 'w' and 'v' are zero. Even the LS estimate which balance the basic equation provides unbiased estimate only if the mean value of 'w' and 'v' equals zero. The above feature shows that any arbitrary way of estimating the parameter in a problem may not in general lead to proper

estimates and we need more mathematically sound approaches.

Even in the least squares fit there are many variants depending on whether the departures of the data from the fitted line is measured (i) vertically (the standard exercise in most text books) (ii) horizontally, (iii) bisector of the above two lines, (iv) perpendicularly, and (v) measured both perpendicularly and vertically are possible. However these are dictated by the underlying scientific mechanisms in the various fields of application. A good discussion of the above is available in Isobe et al.[54] (1990) and Feigelson and Babu[56] (1992). The above is mentioned to stress the importance of a good physical understanding of the problem which arises from earlier studies or by intuition in newer problems.

In fact such an estimate of obtaining unknown parameters is provided by the Method of Maximum Likelihood Estimation (MMLE) in the statistics pioneered by Gauss (Dover[11] 1963, reprint of Gauss 1809) and reinvented by Fisher[1] (1922) is extensively used in present day estimation theory. If you wish to contribute to estimation theory then read more statistics! Thus one can note that even if an algorithm converges it does not guarantee the result to be correct or even reasonable. This defect can be overcome by considering simulated studies where one knows the true values of the system parameters and the noise statistics. Since the Kalman filter is supposed to provide an estimate together with appropriate uncertainty it is good to generate the same by another procedure that is not filter based at all but does however provide the same. In this report when there is only measurement noise but no process noise we are able to generate the results by using the Newton Raphson minimization technique which helps us to understand the filter operation and its results by comparison with the former. When both the measurement and process noises are present then we are in an unknown territory and have to depend on other ways of having confidence in the filter results. In the present work this has been possible when one introduces the generalized cost function that depends on not only the 'innovation' but on the other quantities like the 'filtered residue' and the 'smoothed residue' as well as those depending on the balance of the governing state equations. One can add another cost function based on the initial state conditions as well.

2.3   Problem Formulation and Extended Kalman Filter Equations

One source of information is the state differential equations and the other source of information that captures the above change are the measurements made on the system. The

correction to the state is provided by the measurements based on a proper criterion leading to reduced uncertainty of the state variables. Such a criterion is provided by a probabilistic weighted linear addition of the predicted state and the actual measurement data. The state and measurement variables are related through appropriate functional relationship. Such an update corrective process is repeated at suitable intervals. In fact everything about the state and the measurement equations are to be learnt and estimated from the measurements alone. The above process is described below leads to an optimization problem to be solved which is equivalent to tuning the Kalman filter statistics which are often unknown.

Consider the following nonlinear filtering problem defined for discrete time instants given by k = 1, 2,... N

$$x_k = f(x_{k-1}, \Theta, u_{k-1}) + w_k$$
$$Z_k = h(x_k, \Theta) + v_k$$

where 'x' is the state vector of size $n \times 1$, 'u' is the control input and 'Z' is the measurement vector of size $m \times 1$. The 'f' and 'h' are non linear functions of state and measurement equations respectively. The injected process noise, $w_k \sim \mathcal{N}(0, \mathbf{Q})$ and the injected measurement noise, $v_k \sim \mathcal{N}(0, \mathbf{R})$ are assumed to be zero mean Additive White Gaussian Noise (AWGN), and are identically and independently distributed (iid). The '$\mathcal{N}$' represents Normal or Gaussian distribution and

$$E\left[w_k w_j^T\right] = \mathbf{Q}\delta(k-j) \ \& \ E\left[w_k\right] = 0$$
$$E\left[v_k v_j^T\right] = \mathbf{R}\delta(k-j) \ \& \ E\left[v_k\right] = 0$$
$$E\left[w_k v_j^T\right] = 0 \ \forall \ j, k = 1, 2,... N$$

where N is the total number of sampling instants. E[ ] is the expectation operator, δ is Kronecker delta function defined as

$$\delta(k-j) = \begin{cases} 0 & \text{if } k \neq j; \\ 1 & \text{if } k = j. \end{cases}$$

The parameter vector 'Θ' of size $p \times 1$ is augmented as additional states,

$$\begin{bmatrix} x_k \\ \Theta_k \end{bmatrix} = \begin{bmatrix} f(x_{k-1}, \Theta_{k-1}, u_{k-1}) \\ \Theta_{k-1} \end{bmatrix} + \begin{bmatrix} w_k \\ 0 \end{bmatrix}$$

The non linear filtering problem is now defined as

$$X_k = f(X_{k-1}) + w_k \tag{2.2}$$

$$Z_k = h(X_k) + v_k \tag{2.3}$$

where 'X' and 'w' are respectively the augmented state and process noise vector is of size $(n+p) \times 1$ and thus $w_k \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & 0 \end{bmatrix}\right)$. The control input 'u' and the 'hat' symbol for estimates are not shown for brevity. A formal solution to the above problem is the Extended Kalman Filter (EKF) summarised as Brown and Hwang[108] (2012)

Initialisation : $\quad \mathbf{X_0} = E[X_t]$

$$\mathbf{P_0} = E[(X_0 - X_t)(X_0 - X_t)^T]$$

Predict step : $\quad X_{k|k-1} = f(X_{k-1|k-1}, u_{k-1})$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + \mathbf{Q}$$

Update step : $\quad K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + \mathbf{R})^{-1}$

$$X_{k|k} = X_{k|k-1} + K_k(Z_k - h(X_{k|k-1}))$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

where all the symbols have their usual meaning and

True initial state	$:X_t$
Initial state estimate	$:X_{0|0} = \mathbf{X_0}$
Initial state covariance matrix	$:P_{0|0} = \mathbf{P_0}$
State Jacobian matrix	$:F_{k-1} = \left[\dfrac{\partial f}{\partial X}\right]_{X = X_{k-1|k-1}}$
Measurement Jacobian matrix	$:H_k = \left[\dfrac{\partial h}{\partial X}\right]_{X = X_{k|k-1}}$

The innovation $\left(v_{\mathbf{k}} = Z_k - h(X_{k|k-1})\right)$ following a Gaussian distribution whose probabil-

ity when maximized leads to the Method of Maximum Likelihood Estimation (MMLE), which is operationally equivalent to minimizing the cost function

$$
\begin{aligned}
\mathbf{J} &= \frac{1}{N} \sum v_{\mathbf{k}} (H_k P_{k|k-1} H_k^T + \mathbf{R})^{-1} v_{\mathbf{k}}{}^T \\
&= \mathbf{J}(\mathbf{X_0}, \mathbf{P_0}, \mathbf{Q}, \mathbf{R}, \Theta) \\
&= \mathbf{J}(\mathbf{X_0}, \Theta, \mathbf{K}(\text{traded for } \mathbf{P_0}, \mathbf{Q}, \mathbf{R}))
\end{aligned}
$$

based on the summation over all the N measurements and thus solving for either $\mathbf{X_0}, \mathbf{P_0}, \mathbf{Q}$, $\mathbf{R}, \Theta$ or solving for $\mathbf{X_0}, \Theta$, K as the case may be. When $\mathbf{Q} = 0$, the MMLE is called as the output error method with the Kalman gain matrix being zero. In the usual Kalman filter implementation generally one does not solve for the statistics $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$ but tweak manually to obtain acceptable values. The numerical effort of minimizing $\mathbf{J}$ cannot be swept under the rug and it has to appear in the estimation of the filter statistics.

The estimation of $\mathbf{X_0}$, $\mathbf{P_0}$, Θ, $\mathbf{R}$ and $\mathbf{Q}$ in the Kalman filter is known as adaptive filter tuning. The ghost of filter tuning chases every possible formulation or any variant of the Kalman filter be it EKF, Unscented Kalman Filter (UKF), Particle filter (PF), or the Ensemble Kalman Filter (EnKF) or their combinations. The best possible tuning is necessary if one desires to get near optimal solutions. If not properly tuned it is difficult to infer if the performance of the variants of Kalman filter are due to their formulation or filter tuning!

Though Kalman filter has been applied in so many fields of science and technology we are not in a position to estimate the parameters and their uncertainty as denoted by Cramer Rao Bound (CRB). Even a routine adaptive filtering technique to estimate a constant signal with measurement noise does not seem to exist!

Even if the tunable unknowns are not available or inaccurately known the filter should be able to estimate all of them only from the measurements. Generally the initial estimates are kept not too far from the expected estimates for any algorithm to converge. However for the present approach the initial choice can be over a large range.

There are five steps in the Kalman filter, namely state and covariance propagation with time, Kalman gain calculation and the state and covariance update by incorporating the measurement. In the filter statistics approach all the five steps have to be gone through. However when the constant Kalman gain approach is used only the three steps namely

the state propagation, Kalman gain calculation and the state update are necessary.

The Kalman filter is not a panacea to obtain better results when compared to simpler techniques of data analysis. The accuracy of the results using Kalman filter depends on its design based on the choice of $\mathbf{X_0}$, $\mathbf{P_0}$, Θ, $\mathbf{R}$ and $\mathbf{Q}$. If the above values are not chosen properly then the filter results can be inferior to that obtained by simpler techniques.

There could be other suitable cost functions that one can develop depending on the situation. One cost function called the (Integral of Time multiplied by Absolute Error) ITAE considers the time as a scaling factor. This is meaningful since it is important to ensure a zero error after the filter has converged. This performance index is given by

$$J_{ITAE} = \frac{1}{T} \sum a_i |\nu_\mathbf{k}| dt$$

where the a_i is suitable weight such as related to the innovation covariance. Another cost function useful to study the effects of inadequate modeling in state estimation problem that is very common in Kalman filter studies has been proposed and used in rendezvous and docking problem (Philip and Ananthasayanam[83] 2003),

$$J_{MODEL} = \frac{1}{N} \sum (x(i) - x_{ref}(i))^T \mathbf{P}^{-1} (x(i) - x_{ref}(i))$$

with the summation is over all the N time points and the suffix 'ref' refers to a desired reference trajectory to be followed and the argument in x (.) denotes the time step or point. The P is the covariance matrix obtained with nominal values for the unknown disturbances. If the variations or a deficiency in the modeling is beyond the statistical fluctuations as denoted by the covariance then the above cost function changes substantially and indicates a degradation of the filter performance. The surprising thing is that when $\mathbf{R}$ alone exists in a problem a suitable cost function is considered but when $\mathbf{Q}$ also exists most people seen to avoid the cost function. Is this because of the ad hoc approach of choosing the statistics?

In the present work we introduce a generalized cost function by an expansion of the usual 'innovation' to other quantities generated by the Kalman filter such as the prior and posterior state respectively before and after the measurement is assimilated, the smoothed state after all the measurements are processed. We demonstrate that it is such an approach that decisively indicates the best possible solution in the simulated data analysis and more

so in the analysis of real flight data analysis.

Some fundamental differences exist between the cost function handled by classical optimization problems and the Kalman filter. The former deals with a cost function that is static, with a fixed model and the number of data, with deterministic unknowns. In contrast the Kalman filter has to grapple with a dynamical cost function due to time varying state and measurement model structure and the parameters therein, with continuously increasing number of measurements and the unknowns are both deterministic quantities and the statistic of probabilistic variables.

2.4   Aesthetics, Beauty, and Truth of the Kalman Filter

The aesthetics of the Kalman filter is to consider only the uncertain estimate and the covariance representing the uncertainty. From a deterministic case to move to probabilistic scenario just only one additional quantity is used for describing the results which is economical.

Also this is in some ways similar to many other problems in science and engineering wherein only the first and second derivatives or moments alone are considered. This is just the reason and fortuitously as well for using velocity and pressure or temperature in equilibrium thermodynamics, which depend on the first and second moments respectively of the distribution function governing the random velocity of the gas molecules. This leads to the consideration of fewer moments or states to describe the dynamics of the gas flow. Imagine what would happen if many higher order moments had become relevant! Even in the equations of motion of classical dynamics only up to linear and angular accelerations, which are second derivatives exist!

Another subtle reasoning can be provided for the above feature. Take a rectangular distribution and with increase in sample size the lower order moments converge faster than the higher order moments. This is because away from the middle the tails control the higher order moments. For a very similar reason, the Boltzmann equation deals only with single particle distribution as against multi particle distribution function.

An analogy can be given from real life. Perhaps it is easy to judge a person early as soft or hard. It would take little more time to find out if how flexible he is in dealing with situations or advice. It would take far more time to understand his way of handling

difficult situations in life.

The beauty in the Kalman filter is whether it is true or otherwise the many random variables occurring in a problem are assumed to follow or represented by a multivariate Gaussian distribution, which is completely specified by the mean and covariance. The Gaussian distribution provides an enormous amount of mathematical tractability exactly for linear systems and approximately for nonlinear systems.

The truth in the Kalman filter equations is that once it is derived in one way, it is possible to derive it in a variety of ways with slightly different assumptions, but leading to similar set of basic equations as for linear problems. It is interesting to note that the simplest formulation of the Kalman filter is based on minimum amount of a priori knowledge in probability, statistics, and random process providing respectively the Gaussian distribution, linear relationship among the variables, and white noise. The author of each book has his own derivation! If necessary other suitable distributions, nonlinearity, and coloured noise can be introduced later into the filter framework.

2.5 Definition and Different ways of Looking at the Kalman Filter

We now propose a possible definition of the Kalman filter. The Kalman filter assimilates the measurement information with uncertain system and measurement models based on probabilistic weighted linear addition of the predicted state and the measurement data to adapt both the state and measurement models and their noise statistics in a statistically consistent way. Rao[50] (1987) had stated that all methods of acquiring knowledge are essentially statistics. Hence the analysis of time dependent process carried out sequentially is contained is the knowledge front by assimilating newer information in a meaningful way is the Kalman filter!

2.5.1 Inverse Problem

What is the simplest way to look at the ET? The problem is given the measurements made on the system how can one obtain the structure of both the system and the measurement models? In general due to the redundant amount of data there is no unique solution and this gives rise to a search for best, optimum or even adequate approaches together with subjective assumptions to get a unique solution. Based on previous experience or other-

wise one has some feel about a dynamical system. Many systems are deterministic being governed by well known Newton's laws of mechanics or gravity, or laws of electricity, and magnetism and so on. Though their structure is known the parameters in them may not be known with desirable accuracy. If a system is stochastic in nature such as the population, flow in rivers, stock market then suitable stochastic models have to be evolved and the unknown parameters in them estimated.

In direct problems given the rules we generate the outcomes and there is nothing more to it. Given a certain scenario today can we say how it had evolved? That is why predicting the future is relatively easy but inferring the past is far more difficult! Given a data how it came to be is far more difficult to answer than answer how it will evolve. You see a certain food stuff, can you say the sequence by which it has been prepared but you can prepare a stuff to reach a certain taste. This is just the reason people patent food items! Again the inverse problem is more difficult than direct problem. If you see a person today can you infer how he had reached the present condition? However knowing him now perhaps you can predict better what he could be later on.

2.5.2 Qualitative Modeling

How to qualitatively model the states and the measurements? The system identification according to Zadeh[10] (1962) is the determination on the basis of input, of a system within a specified class of systems, to which the system under test is equivalent. This equivalence implies a loss or error or cost function **J**. Thus when a qualitatively best, proper or adequate model structure for the state and measurement models are chosen it implies identification. Different model structures can be tested with different criteria and the most acceptable one can be chosen. At times one can model the unknown control input as random process noise. In general for present day activities the algorithms and computing power exist but the physical modeling for example at high angles of attack and sideslip of an aircraft due to their nonlinear, coupled nature with unsteady effects presents a formidable challenge for a flight mechanics specialist to pass on suitable inputs to his control counterpart.

2.5.3 Quantitative Estimation

When to accept the quantitative estimates? The quantitative parameter estimation is defined by Eykhoff[28] (1974) as the experimental determination of parameters for the above qualitative model. Modeling has to satisfy the conditions of controllability and observability. In simple physical terms these mean that a control has to excite the states and the excited states have to be measured to determine the parameters. For efficient determination of parameters optimal inputs have been proposed. These excite the various modes of the system controlled by the parameters.

A simple example of qualitative and quantitative estimation together with consistency checks is provided in Appendix-B. No matter what you do mathematically in statistics note that there is a manual override and thus it extends to Kalman filter as well! It also shows that some statistical tests due to the level of confidence at which the estimates pass can be deceptive and hence we look for decisive tests and/or criterion in the report.

2.5.4 Handling Deterministic State and Measurement Errors

How to handle deterministic state and measurement errors? If there is a random error in the state model or the measurement model it can be handled by including the noise. How to deal with deterministic errors such as an improper structure of the state model or a bias or scale error in the measurement? We can still handle such deterministic errors in the state and measurement equations respectively by adding the process and measurement noise to encompass them.

2.5.5 Estimation of Unmodelable Inputs by Noise Modeling

How to handle unmodellable inputs? It is argued subsequently that the effect of unmodellable or unmodelled errors in the state and measurement equations can be offset by adding noise in the model. A more interesting question is can we estimate them as well? The answer is yes and has been done in many cases. The random noise can be modelled as 'white' or a more general Gauss-Markoff process (Gelb[27] 1974) of a suitable order thus providing a structure for the random component. The qualitative structure of the random noise modelled with some free parameters whose estimation can provide a quantitative estimate of that noise.

2.5.6 Unobservables from Observables

How to estimate the unobservables from the observables? Use intuition or otherwise to connect the unobservables in the state with the measurements. The measurements made on the system are functions of the state and may not always correspond directly to the states and the parameters. The unknown state and parameters may be unobservable and have to be inferred from the measurements. The estimation of the aerodynamic lift, drag, and moment of aerospace vehicles by flight test data analysis is just one example of the above!

2.5.7 Expansion of the Scenario

How to get even more knowledge from the measurements? The answer is to improve or in other words put more information into the state model. The expanded state equations contain our understanding that is further improved by the measurement information. As mentioned in the previous section the measurement space can in general be different from the state space. The state equations can be modelled with increasing level of sophistication. In order to appreciate this better consider the radar measurements of an airplane which suffered an accident. The radars provide range, azimuth, and elevation information. If the airplane is treated as a point mass governed by kinematic equations then more accurate estimates of its position, velocity, and perhaps acceleration can be estimated. Next if the state equations are formulated as dynamical equations then the mass, as well as the aerodynamic parameters can be estimated. A further sophisticated model of the airplane's equations of motion helps us to expand the scenario to obtain the forward speed, angles of attack and sideslip, the pitch, yaw, and roll angles, and even the way in which the throttle and the controls have been operated thus facilitating the analysis of the accident and apportion the cause to airframe, pilot, weather or other factors. The expansion of the scenario can be taken as a more sophisticated version of the unobservables from the observables.

2.5.8 Deterministic or Probabilistic Approach?

What is the difference between deterministic and probabilistic approach in ET? Gauss as mentioned earlier had been the earliest to utilize the measurements of the position of the

asteroid Ceres in the celestial sphere to determine its orbital parameters and predict its position at a later time for sighting it. He used the method of least squares in a deterministic way to handle the more number of redundant measurements. He had provided many clues even for the probabilistic approach. The least squares estimate and the uncertainty are only qualitative. When the characteristics of process and measurement noise denoted respectively by 'w' and 'v' are specified quantitatively through appropriate probability distributions these get translated after the filter operations through the data into the distribution of the estimates in a quantitative manner. Gauss had understood the deterministic and probabilistic formulations and also that the assumptions in the latter need not always be true.

Before the data is available one can talk of probability. Once the data is available it is deterministic and the entire unknown both deterministic and the statistics of the random variables can all be treated as deterministic unknowns. This is evident by the translation of the problem to minimizing the cost function $\mathbf{J}$ which is deterministic. Of course one can use probabilistic approaches to solve a deterministic optimization problem.

For theoretical expediency and minimal information the $\mathbf{Q}$ and $\mathbf{R}$ are generally assumed to be white Gaussian. It is not the validity but the ability of the above that provide acceptable results. There is a charm in doing it that way as between a game of 'draught' and 'chess' where the latter a few more rules provides panoramic situations. Rules made by humans led them to problems. The basic laws of nature have always been very simple.

2.5.9   Handling Numerical Errors by Noise Addition

How to handle numerical errors? In many real time applications propagating the state and covariance equations of the Kalman filter could be highly time consuming. Here one could use less accurate but fast solvers and incur some numerical errors. The less accurate solvers can be interpreted as a modeling deficiency. Once this is accepted then an additional random process noise can offset numerical inaccuracy. However the addition of process noise cannot be a 'panacea' for large scale modeling or numerical deficiencies. Such noise inputs always come at the expense of increased uncertainty of the state or a parameter. One can offset modest amount of numerical errors at the cost of a small increase in the uncertainty of the estimates.

2.5.10 Stochastic Corrective Process

Are there other characterizations of the Kalman filter operation? There is a philosophical reason for the sequential approach to be preferable. The evolution of many systems for large times is unpredictable. Typical examples are the population growth, economics, climate, weather, satellite orbit, and space debris. In all the above the dynamics of the system could be changed consciously with time by the society or unmodellable forces or features. Thus additional information at various times regarding the system based on the newer data helps to predict the system better before the next data arrival.

As an example a satellite's orbit cannot be predicted accurately for all times to come no matter how accurately one accounts for the model of the earth's gravity field, an atmospheric model, and all other perturbations. Due to the very random nature of unmodellable forces on the satellite its trajectory would depart from the one based on any assumed model after quite sometime. Another classic example is that of weather prediction. No matter how accurate the atmospheric model is one is compelled to take recourse to measurements which is a newer information about the system that has run in parallel in nature without of course solving any governing equations! Thus at various times one is compelled to offset the limitations in the model and input using the measurements. In other words theory and experiment should go hand in hand! Such processes which get corrected at various times have been called as 'Stochastic Corrective Processes' by Narasimha[29] (1975).

2.5.11 Data Fusion and Statistical Estimation by Probabilistic Mixing

How are the states and measurements combined? A remarkable feature of present day ET is a variety of information from different sources, rates and accuracies are combined not just algebraically but statistically in an optimal way. This can also be called as data fusion and assimilation (Raol[103] 2010). Such a fused data is combined with states using suitable weightages to obtain the best possible estimate of $\mathbf{X_0}$, $\mathbf{P_0}$, Θ, $\mathbf{Q}$ and R all from just the measurement Z only!

The Kalman filter was called a mixer by Gelb[27] (1974) and the final effect is the filtering of the noise. It is well known that when two independently distributed random variables are added then the variance of their sum is the sum of the individual variances. This means

the uncertainty of the new random variable has increased. However if we add them in a weighted manner say half to each one of them then the resulting random variable will have a lower variance than either of the first two. Getting the proper or the so called optimal weight depending on the various individual noise covariances is an important problem in ET and manifests as filter tuning which is a big industry.

2.5.12   Optimization in the Time Domain

How to use the formal local filter update equations over the complete data? The earlier sections dealing with propagation from one time to the next and update at a given data point are formal in the sense given an initial $\mathbf{X_0}$, $\mathbf{P_0}$, Θ, $\mathbf{Q}$ and R the filter can operate on the full data. With improper choice for the above any result can be generated. In order to make sense from the full data, an overall cost function have to be minimized and in the process all the tuning parameters ($\mathbf{X_0}$, $\mathbf{P_0}$, Θ, $\mathbf{Q}$ and $\mathbf{R}$) are estimated. This provides a consistent solution valid over the full data. Since the minimization is in general a non linear problem and an iterative processing of the data by the filter equations are needed. The updates for all the above can be carried out at each and every data point or over a window or full data by repeated processing through the data. The optimisation updates based on the sensitivity of the difference between model and measurement if only measurement noise is present is fairly straight forward. The use of Newton Raphson method had been prolific but it turns out if the measurement channels are very few or the data length is very short other techniques have to be utilised. Also probabilistic methods like genetic algorithms have also been successfully applied. After the convergence of $\mathbf{J}$ further checks have to be carried out if the initial assumptions regarding the signal and noise are satisfied like if the fit is good, smooth, and the innovations are white and Gaussian. Care has to be taken to see that the above are not statistically deceptive but decisive.

2.5.13   Frequency Domain Analysis

Is a frequency domain approach possible? Though generally one deals in the time domain by combining or matching of state and measurement it is possible to transform both the above into frequency domain and carry out the analysis. The advantage being in particular for real flight test data, the contamination of the pure airframe excitation, with noise from

other sources such as engine, structural vibrations can be looked into in the frequency domain and cut off. A formulation of MMLE was provided in the frequency domain (Klein and Morelli[96] 2006). The analysis in the frequency domain due to the transformation of the data can lead to changes in the estimates as well as their uncertainties. However it is restricted to linear systems. The earliest filtering technique by Wiener[4] (1949) is in the frequency domain but it had many difficulties in practical implementation. The contribution of Kalman in dealing with estimation problems has two aspects. One is the change from a 'batch' processing mode to 'sequential' mode and the other is to switch from 'frequency' domain approach to 'time' domain approach. It is felt that based on the sequential least square estimation, such a data processing would have any way come about, but the greater contribution is to move away from frequency to time domain which is more natural and with extensions to general nonlinear systems.

2.5.14   Smoothing of the Filter Estimates

What are the best state, measurement and parameter estimates from the filter? During a forward pass of the filter over the complete data only the last point has an estimate based on the complete data. In order to obtain the best possible filter estimates at all data points a backward smoothing can be carried out. This is based on a backward filter pass over the data. Then at every data point by statistically combining the estimates from the forward and the backward filter passes the best estimates can be obtained. After such a smoothing process the state represents the best possible signal content. It is the smoothed states that greatly helped in the estimation of **Q** in the present work.

2.5.15   Improved States and Measurements

Are the noises in both the states and measurements reduced? The mixing of the complimentary information from the state and measurement helps to obtain better estimates of each as also the parameters in them. After the filter smoothing process, improved state estimate is available and this helps to smooth the measurement. The measurements help to provide improved state variable and their unknown parameters. The Kalman filter can be understood as an advanced model based moving average filter. The usual moving average filters use a simple polynomial fit over a suitable window size of the data. The Kalman filter using knowledge based state and measurement models is able carry out much better

and more accurate signal and noise separation and estimation.

2.5.16 Estimation of Process and Measurement Noise

How to estimate $\mathbf{Q}$ and $\mathbf{R}$? When the deterministic parts of the state and measurement equations are balanced all over the data what remains is the corresponding noise. However since the filter generates among others predicted, updated, and smoothed state estimates and the corresponding measurements one has to find out which combination of the above helps to provide a suitable 'statistic' to obtain an estimate of $\mathbf{Q}$ and $\mathbf{R}$ if they are also unknown.

2.5.17 Correlation of the Innovation Sequence

How to conclude if the estimates are consistent? It is possible to look at the behaviour of the innovation sequence for having tuned the filter statistics $\mathbf{X_0}$, $\mathbf{P_0}$, Θ, $\mathbf{Q}$ and $\mathbf{R}$ properly. If the filter is optimal then the correlation of the innovation sequence should be 'white'. This implies all the information from the data has been extracted and what remains is pure random 'white' noise with no information content. This can happen only when the model and the measurement structures are proper, the parameter in them have been obtained after the numerical optimization algorithm has converged properly. Even if any one of the above is not correct then the above sequence would not be 'white' within the statistical limits of acceptance. Further an ideal correlation coefficient matrix for the parameter estimates should be an identity matrix. This can happen only with infinite data and with limited data correlations would exist among the estimated parameters and this should be kept in mind in their subsequent use. The 'white' noise is the worst data that will fail all modelling, and algorithms in ET and can provide any random result with no meaning. One can refer to Rao[50] (1987) for some fallacious inferences drawn from random sequences as if they are laws of nature! If the atmosphere, and earthquakes are truly random then researchers in such fields can stop working!

2.5.18 Reversing an 'Irreversible' Process

What and how is it finally achieved? Once the signal and noise are mixed they become indistinguishable unless a proper modeling of signal and measurement is made and further

numerical effort is put in to separate them. An analogy may be given as mixing of salt and water when the identity of salt is lost. The situation is similar to a signal corrupted by noise. Our understanding is that with evaporation salt alone remains and water vapour goes up. Such a compulsive understanding of the behaviour of salt and water helps to separate them. This is precisely what is demanded in ET, namely model the system and the measurement before processing the data. In the Kalman filter numerical effort is required to propagate the state and the covariance equations (if there is process noise), form a cost function, and optimize it by utilizing various techniques (in general iterative) before the solution is obtained. Thus one can note the mixing of noise with a signal is relatively easy! Thus adding noise is an 'irreversible' process and to bring the system back to its original state a lot of effort is required.

2.5.19 Some Human Activities from ET Perspective

How to view some human activities from ET perspective? The advances in science and technology help to improve the cultural style and the living standard of a society provided the wealth is distributed equitably. An inequitable distribution of wealth leads to tensions and turmoil in a society. The cost function for the society need to have proper weightage over food, shelter, health, and leisure for all.

Inventers and innovators in the society must be rewarded. But it should not be very small or very large imposing burden on the society. In the language of the filter what is the optimum (=reward) Kalman gain? With the availability of new gadgets in sports the rules, regulations, and umpiring will have to change. In cricket these are the stump cameras, and the Hawkeye. The cost function here is not dependent on the gadgets but the immediate excitement, and enjoyment provided to the spectators. The reference to a third umpire in cricket or the challenge by the player in tennis provides a quick and correct decision. The umpiring mistakes are very much reduced. But if such sport gadgets are not inducted the 'status quo' attitude can lead to acrimonious and long drawn disputes. Thus there is a need to sense the change, capture and correct our life.

A fine example of data assimilation is our digestive system extracting the nutrients and rejecting the rest. Accumulating undesirable contents leads to digestive problems or disease. Another fantastic example of assimilation is the evolution of life where the important experiences of the earlier generations are assimilated and compactly coded in the DNA. The

spiritualistic state corresponds to system and the materialistic world corresponds to observations. Buddha leading a princely life observed the sufferings due to decay, disease, and death. He assimilated such observations and changed his way of living discarding wine, women, and wealth.

For estimating or controlling a variable it is not always necessary or feasible to handle the same variable. The removal of violence and terrorism could perhaps be achieved more effectively by following a sattvic way of life. The physical, personal, psychological, and professional suffering of people can be overcome not necessarily by direct actions but by remote measures subtly connected to the issues. Such perspectives of some activities in ET (Ananthasayanam and Bharadwaj[105] 2011) are provided in Table - 2.1.

2.6 Tuning of the Kalman Filter Statistics

How does one achieve all the above stated objectives in designing the Kalman filter for any application? It is not known to many that the enthusiasm which followed soon after the Kalman filter was introduced was damped since the noise statistics had to be provided to design the filter. Obviously the effort to be put in minimizing $\mathbf{J}$ cannot be escaped! In relation to its importance to obtain as close to an optimal solution as is possible the corresponding effort does not appear to have been put in the field. Generally one manually tweaks the statistics to reach acceptable results instead of tuning properly to get even better results. Thus the filter tuning has not matured to a level for routine use. Gauss had a relatively ideal situation with a good system model and only the measurements had noise. Kalman when he proposed the filter required the statistics of the process and the measurement noise to be known and dealt with only state estimation. The adaptive approach or filter tuning tries to obtain the filter statistics $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$ by using the filter operating over the measurement data.

The filter tuning varies from ad hoc, through heuristic to rigorous methods. Generally the tuning is manual but adaptive processes are needed to obtain better results. The ad hoc quick fix solutions are such as limiting P from going to zero, or add $\mathbf{Q}$ to increase P before calculating the gain, multiply P by a factor to limit K all have obviously limitations in handling involved problems or scenarios. In the fading memory filter the estimates based on the current set is averaged with the previous estimate, with a weighting parameter be it for $\mathbf{R}$ or $\mathbf{Q}$. All the above are arbitrary and can lead to inaccurate results. The third

rigorous approach could be time consuming to the extent of solving the whole problem. Exact solutions are very hard, approximate choice can lead to inappropriate results but the middle path of heuristic approaches are quite appealing and is the present work. We derive in this report a fairly general adaptive approach to tune the Kalman filter statistics for any system. The present recursive filter recipe helps to achieve statistical equilibrium for all the unknowns together with their internal consistency checks.

2.7 Important Qualitative Features of the Filter Statistics

Should the $\mathbf{P_0} = \mathbf{Q} = 0$ then the filter will not learn anything from the measurements all of which will be ignored. The $\mathbf{R}$ is fairly objective and can be determined from the measured data. The $\mathbf{P_0}$ is tricky and generally the off diagonal elements are set to zero and the diagonal elements are set to large values but however their relative values are crucial for an optimum filter operation. The $\mathbf{P_0}$ controls the handover from the initial transient to $\mathbf{Q}$ for steady state filter behaviour and has to be chosen carefully. The $\mathbf{P_0}$ is important like for a child with deficient or overdose of nutrition when young will have some shortfall even with appropriate nutritious food he/she may have in later life. The $\mathbf{Q}$ is notoriously difficult but it helps to inject uncertainty into the state equations to assist the filter to learn from the measurements and it also controls the steady state filter response. Even when there is no state noise but only the measurement noise in the data, starting with some initial estimate $\mathbf{X_0}$ and Θ which is somewhat far away from the true values in order to assist the filter to learn from the measurements a non zero $\mathbf{Q}$ has to be injected into the state equations since the effect of $\mathbf{P_0}$ will fade away quickly. A large value of $\mathbf{Q}$ will lead to a short transient with large steady state uncertainty of the estimates and vice versa for small $\mathbf{Q}$. In the present work since both $\mathbf{X_0}$ and $\mathbf{P_0}$ are simultaneously tuned for data without process noise the requirement for injecting $\mathbf{Q}$ does not arise.

Thus though $\mathbf{Q}$ is considered notorious it is the life line of the Kalman filter doing good work all the time. Grossly misunderstood just like good people thought to be bad for a long time! Some classic examples for such systems are the GPS receiver clocks, satellite, trajectory of aircraft, missiles and re-entry vehicles. These are handled by using the kinematic relations between the position, velocity and acceleration all assumed to be driven by white Gaussian noise $\mathbf{Q}$ of suitable magnitude to enable the filter to track these systems. Generally the filter parameters are tuned off line using simulated data and

subsequently used for on line and real time applications perhaps with efficient numerical procedures.

It is only by the **Q** that an analyst can handle the unmodelled or unmodellable errors and even account for numerical errors! The **Q** is also helpful to track systems whose dynamical equations are not known. The process noise can estimate, account or offset for some deficiency, inaccuracy, or error in the following namely the initial conditions, system and measurement model equations, control or external input, measurement noise statistics, the numerical state and covariance propagation or update operations.

Similar to the different derivation of the filter equations there could be different methods of implementing the filter for any practical situation. One could work with time varying full, diagonal, or constant matrices **Q** and R, or work with constant Kalman gain matrix, or with important Kalman gain matrix elements. The constant **Q** and R matrices approach is generally preferable and the constant gain approach is more easily implementable though not necessarily optimal.

2.8 Constant Gain Approach

In many problems after the initial transients the Kalman gain matrix tends to a non-zero constant value due to finite amount of data or modeling errors. It is useful to trade the filter statistics to the constant Kalman gains to minimize the above **J** by using an optimization algorithm assuming the innovation covariance to be a constant. The advantage of using constant gains is that the covariance equations need not be propagated and updated thus enormously saving computational time. The gains can be chosen using a suitable cost function based on the normalized innovation over a period of time just as for the filter statistics.

For the constant gain approach any suitable optimization algorithm can be utilized to minimize the cost function. It has been found that the simple genetic algorithm are quite effective in handling a variety of aerospace problems such as the docking and rendezvous (Philip and Ananthasayanam[83] 2003), the evolution of the space debris (Ananthasayanam et al.[91] 2006), re-entry of a space debris object (Anilkumar et al.[98] 2007), integrating the MEMS-INS and GPS (Helen Basil et al.[87] 2004), and in estimating the total electron content in the atmosphere (Anandraj et al.[90] 2005). Based on the above successful applications we suggest the Kalman gain approach in huge data assimilation problems.

2.9 Kalman Filter and some of its Variants

The simplest formulation of a Kalman filter is when the state and measurement equations are linear. For linear systems during the evolution and the update the normal distribution is maintained, but with changed mean and covariance. For nonlinear systems even if the initial distribution is normal, it gets distorted after propagation and so a suitable local approximation or quasi linearization has to be made.

In the Extended Kalman filter (EKF) the nonlinear systems and/or measurements equations (Smith et al.[9] 1962) are approximated by first order Taylor series expansion about the previous updated state. The probability density is approximated by a Gaussian, which may distort the true structure and at times could lead to the divergence between the filter prediction and the measurements.

In the Unscented Kalman filter or UKF (Julier et al.[65] 1997) approach instead of linearizing the functions, a set of chosen points are propagated through the nonlinear transformation. These points are so chosen such that the mean, covariance, and possibly also higher order moments, match with the propagated distribution.

The Interactive Multiple Model (IMM) was introduced to handle rapid system dynamics (Blom et al.[53] 1988), since the gain cannot change as rapidly as the system dynamics. Here additional model state equations describing velocity, acceleration, jerk and so on are introduced. Here each filter corresponding to the dynamic model computes the states in parallel using the given measurements and the responses from individual filters are combined based on their mode probabilities to form the response from the IMM. This interaction requires the specification of the transition probability matrix p_{ij} that the model 'i' is at the current observation time given that model 'j' was at the previous observation time as well as a sojourn time 'τ_i' to be specified by the analyst to calculate suitable weights for each filter's output.

The particle filtering (Gordon[58] 1993) is a Monte Carlo technique for state estimation that can handle nonlinear models together with non Gaussian noise. Here the state probability density is approximated by using point particles having positive weights. Based on the initial distribution the weights are chosen and then the particles are propagated following the system dynamics together with the state noise. Then using the measurement their weights are adjusted and normalized among all the particles. The particles that can track

the measurements gain weight and the ones far away lose their weights. However, after a while all, but one weight, will become zero leading to degeneracy. A resampling scheme is introduced to solve the degeneracy problem that discard the particles with small weights and focus on the particles with more significant weights.

For large size systems, such as those occurring in geophysical studies maintaining the covariance matrix computationally being difficult, in the ensemble KF (EnKF) for large problems (Evensen[84] 2003), the estimate and the covariance matrix are replaced by the sample covariance from a large number of ensemble members similar to a particle in the particle filter. Each member of the ensemble is propagated including the process noise and later updated using a so-called virtual observation.

One may note the evolution of the variants of the Kalman filter possesses some similarities as it progressed to handle simple, complex, to massive problems as in many other fields such as fluid dynamics or structural mechanics. In these cases commencing from simple geometries one obtains closed form analytical solutions (as in the linear KF wherein the gains can be pre computed to process the data as and when they arrive) followed by numerical calculations in involved cases (as in EKF). When the geometry is complex and the boundary conditions are involved it becomes necessary to discretise and form cells over appropriate space and time (as in PF) to obtain the solution. Further when massively complex geometries and boundary conditions occur other innovative approaches (as in EnKF) have been developed. An extensive bibliography of the non linear estimation is provided by Georgios[78] (2001) and an excellent review of non linear filters is given by Fred Daum[89] (2005).

Table 2.1: Perspectives of Some Activities in Terms of Estimation Theory

No	Description	Estimation Theory	Astronomy	Society and its evolution	Intellectual property rights	Cricket or Tennis
1	Change	System model	Predict the position of celestial objects	Living standards and culture	Inventions	Existing Rules and umpiring
2	Capture	Measurement model	Observations with Unavoidable errors	Scientific and technological advances	Impact on society	Camera and Hawkeye
3	Correct	The above estimates are combined	Combine the above to get improved position of objects	Changing standard and culture	Awards, patents, and copyrights	Third umpire
4	Nature of cost or its criteria	Highly subjective but cast subjectively reasonably	Mean square error between model and measurements	Many but should be reasonable	Proper recognition, rewards and returns	Refer to the third umpire or player challenge
5	Cost function	Reduce the difference between model and measurements	Minimize the above error	Improved standard of living and happiness	Encourage inventions without burdening society	More number of correct decisions
6	Effect of proper Correction	Better than either the model and Measurement	Improves estimated position of celestial objects	Improved and Equitable living for all	Encourages further inventions	Increased excitement and enjoyment
7	Effect of improper Correction	Can make it worse than model and the Measurement	Celestial objects can be lost for later tracking	Unequal change can lead to tension and turmoil	Exploitative economics leads to disparity in society	Incorrect decisions spoils excitement and enjoyment

Chapter 3

Kalman Filter Tuning : Earlier and the Present Approach

3.1 Introduction to Kalman filter Tuning

The estimation of $\mathbf{X_0}$, $\mathbf{P_0}$, Θ, $\mathbf{R}$ and $\mathbf{Q}$ in the Kalman filter is known as the problem of adaptive filter tuning. The ghost of filter tuning chases every variant of the Kalman filter which can at best be minimized but not completely ignored if one desires to get near optimal solutions. Further it becomes difficult for one to infer if the performance of the variants of Kalman filter are due to their formulation or filter tuning! It should be remarked that in the best spirit of the estimation theory in particular the recursive Kalman filter approach even if $\mathbf{X_0}$, $\mathbf{P_0}$, Θ, $\mathbf{R}$ and $\mathbf{Q}$ namely the initial states, their covariances, parameters in the state and measurement equations, the measurement and state process noise covariances are not available or inaccurately known the filter should still have the ability to estimate all the above from the 'observables' that are measured and commencing not too far from the proper estimates for the algorithm to converge. One would like to have the initial choice of all the unknowns should not be very critical. The filter should be self consistent in estimating all the unknowns. Generally one tune the filter statistics off line using simulated data and later use it to process real data on line or even in real time. The Table-3.1 discusses the triplets (Ananthasayanam and Bharadwaj[105] 2011) occurring in the Kalman filter.

The present work shows that there is a compulsive proper choice of $\mathbf{X_0}$ and $\mathbf{P_0}$ in order to derive the best possible state and parameter estimates and their uncertainties using the filter. For both the states and parameters the choice of $\mathbf{X_0}$ appears to be relatively weak for the filter estimates but the choice of $\mathbf{P_0}$ is very important in particular for the parameter uncertainty represented by CRB. Generally even with not quite a good tuning

of the filter statistics the estimates could be near the optimum but lead to large variation in the uncertainty represented by the CRB. However it may be cautioned that for data with only measurement noise if a Newton Raphson (NR) technique is used to minimize the cost function the effect of the initial unknown state can affect the estimates but can be handled if this is also treated as an unknown. In the case of the filter by the process of smoothing the unknown initial state can be also estimated.

As we will see later the elements of $\mathbf{P_0}$ have to be chosen not just low or high but in such a way they lead towards optimum filter estimates and the CRB. The choice of suitable $\mathbf{X_0}$, and $\mathbf{P_0}$ reminds one of the bitter struggles in statistics in specifying the 'prior' for the Bayesian approach which had been studied enormously.

3.2   Some Simple Choices for Initial $\mathbf{X_0}$ for States and Parameters

There are many 'ad hoc' methods to make the filter operate satisfactorily. Since some of the states are generally measured either the first or the average of the first few measurements can be taken as the initial value $\mathbf{X_0}$ for the state. The parameters are used as augmented states in the EKF route. Since either some computational or experimental results are available, these can be set as initial $\mathbf{X_0}$ for the parameter values. At times if possible a least square parameter estimates based on balancing the governing differential equations using the appropriate measurements can also be used as start up values.

3.3   Importance of Initial $\mathbf{P_0}$ for States and Parameters

One might wonder as to where is the necessity for tuning $\mathbf{P_0}$ lies, when in principle it can be arbitrary. The answer to this lies in the following. For example if $\mathbf{P_0}$ is set equal to zero (for a very confident choice for the initial estimates) then the filter ignores and learns nothing from the measurements. If $\mathbf{P_0}$ is extremely large (a pessimistic choice for the initial values), then the filter believes the measurements much more and provides very little weightage or ignores the state model values leading to large fluctuations in the state and parameter estimates along with large final uncertainty. Thus there appears to be a proper choice for $\mathbf{P_0}$ which is neither zero nor infinity to provide proper estimates and uncertainty.

This $\mathbf{P_0}$ is one of the important tuning parameters as stressed by very few like Maybeck[41]

(1979), Candy[49] (1986), and Gemson[55] (1991) but most people seem to treat the above casually. Usually one assumes a guess $\mathbf{P_0}$ which tends to become very low after some data points and in order to make the filter learn from the subsequent measurements introduce an additional $\mathbf{Q}$ by trial and error into the state equations. This finally leads to some estimates and uncertainties. The former usually may be close but the latter generally away from the correct values. The peculiar situation is one has introduced $\mathbf{Q}$ even when there is no model structure uncertainty. Only to learn from the measurements the $\mathbf{Q}$ was introduced. We looked for a recursive procedure that provides a proper $\mathbf{P_0}$ without any $\mathbf{Q}$ in such cases and it turns out to be so as will see subsequently.

The important point is that the initial state and parameter $\mathbf{P_0}$ can affect the final covariance $(\mathrm{P}_{N|N})$ from the filter operation. This can become crucial in certain state estimation problems such as impact point estimation and its uncertainty for target tracking. In such cases the results for the uncertainty from improper or inaccurate tuning can be highly deceptive. Even in parameter estimation problems the estimates and the uncertainties that are attached can be important in the design of control systems.

3.4 Choice of Initial R

Usually a good initial estimate for $\mathbf{R}$ can be obtained from the calibration of the measuring instrument and generally it is assumed to be constant over the data length. In the NR procedure of minimizing the cost function based on data without process noise even $\mathbf{R}$ can be treated as an unknown and estimated along with other unknown quantities. One usually assumes an initial value for $\mathbf{R}$ in the filter which could be close to zero and later adaptively update and estimate it.

3.5 Choice of Initial Q

In principle the $\mathbf{Q}$ should reflect the uncertainty in the assumed state model or any 'unmodellable' feature of the state or even unknown random state input. The $\mathbf{Q}$ along with the initial $\mathbf{P_0}$ plays a very important role in the filter operation without divergence. The value of $\mathbf{Q}$ should be small enough to retain the learning potential from the measurement but not large to increase the uncertainty so that the filter estimates become useless. One usually assumes an initial value for $\mathbf{Q}$ in the filter which could be close to zero and later

adaptively update and estimate it.

3.6 Simultaneous Estimation of R and Q

When the data contains the effect of both the measurement and process noise it becomes far more difficult for analysis. In general minimization of the cost function by treating **R** and **Q** as unknowns during which the simultaneous update and estimation of **R** and **Q** appears to be difficult as reported by many researchers in the field. For a given **R** changing the sequence of measurement noise makes the dynamics noisy means more blurred. In contrast the varying sequence of process noise alters differently every time the dynamical behaviour of the system making it stochastic meaning drift randomly. The interesting point is the filter by tracking the drifted dynamical behaviour even with large **Q** estimates the parameters controlling the original dynamics of the system without the effect of **R** and **Q**. Since **R** and **Q** occur respectively in the measurement and state equations their effects are negatively correlated. The **Q** represents the average rate of change of the state, tracking ability by indirectly controlling the rate at which the old data are forgotten, and affects the estimation accuracy (Bohilin 1976). The effect of **R** is opposite to that of **Q**. Thus during simultaneous recursive estimation if the statistics for estimating them are not properly chosen then **R** is over estimated and **Q** is under estimated and vice versa. This is just the reason due to which Gemson[55] (1991) in his approach had to update **R** and **Q** alternately.

3.6.1 Tuning Filter Statistics with only R

In the spirit of recursive filtering operations we have sought to tune all the filter statistics by repeated iterative processing of the data and nowhere used any optimization procedures. When only measurement noise is present in the data it is possible to get both the estimates and the CRB by minimizing the cost function **J** formed by the difference of the estimated model parameters and the actual measurement. The NR procedure carries out the above task very efficiently for simple to involved systems (Ananthasayanam et al.[80] 2001). It is the results from the above NR procedure that served as an anchor for tuning the filter parameters to get the closest possible estimates and the CRB from the present recursive filter passes through the data. It appears to be not feasible to exactly reproduce the NR results from the filter for each and every data but sufficiently close such that the

over an ensemble of data there is very little difference.

3.6.2 Tuning Filter Statistics with both R and Q

However when process noise is also present in the data we have to look for consistency based on a comparison between the injected and estimated values of **R** and **Q**. Further the additional cost functions proposed in our work based on balancing the state and measurement equations helped to obtain confidence in the results. When the filter operates through the data it generates prior, post, and smoothed state estimates and their covariances which help to generate candidate 'statistic' to estimate both **R** and **Q**. From these one can also generate additional cost functions to see how best the state equations are balanced. Then all the above state estimates and covariances can be transformed into measurement space to be compared once again with the actual measurements and its covariance to generate candidate 'statistic' to estimate **R**. These help to form more cost functions to see how best the measurement equations are balanced. It is necessary to choose from among the above 'statistics' the proper combination of 'statistics' for simultaneously estimating **R** and **Q**. Any improper combination does not lead to proper filter operation and even if it does leads to inappropriate results as is the case in the earlier approaches to solve the filter tuning problem.

The subsequent sections after a review of earlier approaches propose the present method. The Chapter 4 demonstrates the results using the simulated data followed by real data in Chapter 5 based on a comparison of the present results with earlier approaches.

3.7 Earlier Approaches for Tuning the Kalman Filter Statistics

3.7.1 Mathematical Approaches and Desirable Features

Most mathematical treatments implicitly assume unlimited data. However we have to deal with finite data and wish to get the solution within a few filter passes. Further most such treatments do not provide sensitivity among the input and output variables in a problem which could lead to insensitive results and conclusions. The Appendix-B using the OEM approach for a simple curve fitting provides an example for such inferences.

In any field of science or technology one needs at times stability and at other times sensi-

tivity. For example in the Newton Raphson optimization technique there is stability in the procedure commencing from different reasonable initial values converging to the same result for a given measurement data. Should the measurement data change then the converged parameters also change(!) thus showing sensitivity. We have established such a 'stability' and 'sensitivity' for the present reference recursive recipe.

3.7.2 Review of Earlier Adaptive Kalman Filtering Approaches

One brute force method for tuning filter statistics is to carry out an optimization exercise to solve for the statistics based on minimizing a suitable cost function over thousands of combinations of the unknowns. These would be too unwieldy requiring such a massive numerical exercise to be carried out for each and every new problem scenario. We should thus look for elegant approaches which help tune the filter with reasonable numerical effort. In particular in EKF if the unknown noise covariances are incorrectly specified biased estimates can arise (Ljung[43] 1979, Ljungquist and Balchen[60] 1994). Even when the system parameters are known, if an inaccurate description of the noise statistics are used the filter may give poor estimates, or even diverge.

There are broadly four approaches for adaptive filtering namely Bayesian, Maximum Likelihood, Covariance Matching and Correlation Techniques (Mehra[15] 1970).

Bayesian Method

Every update in a Kalman filter is obviously a Bayesian update. The various approaches have been called differently based on what and how they handle such as Bayesian, ML based on the innovation, filter statistics, and the autocorrelation of the innovation to adaptively estimate or account for the unknown covariances. The work of Alspach[23] (1974) deals with a bank of autonomous Kalman filter run with a range of Kalman gains. Each one stores a running sum of the square of the residuals. Subsequently it is possible to obtain the estimates of the unknowns based on a weighted sum over the grid points of the gain. Hilborn and Lainiotis[13] (1969) show that a Bayesian optimal adaptive estimation system converges to the average performance of an (unrealizable) optimal system operating with knowledge of the parameters which are unknown to a Bayesian adaptive/learning system. Thus an adaptive system learns the true parameters based on assimilating the information from the data.

Maximum Likelihood (ML) and Expectation Maximization (EM)

The ML and EM methods need the specification of the probability distribution (usually a Gaussian) followed by the difference between the measurement and the model prediction called innovation or prediction error. The maximum likelihood methods (Kashyap[16] 1970, Bohlin[32] 1976) maximize the likelihood function based on the innovations containing the unknown covariances $\mathbf{Q}$ and $\mathbf{R}$. Usually time consuming gradient based numerical optimisation or other schemes that reprocess the data several times are required to estimate the unknown covariances.

In order to avoid nonlinear optimization in the ML approach Shumway and Stoffer[74] (2000) proposed an iterative method using the expectation maximization (EM) technique. The EM consists of expectation and maximization steps. First the states are estimated using an initial guess of the unknown parameters based on a Kalman smoother. The unknown parameters are next estimated by ML method. This process of estimating the states using the Kalman smoother and optimizing the parameters using is repeated until parameter convergence.

Bavdekar et al.[104] (2011) used both a direct ML method and an EM method for nonlinear systems, based on the extended Kalman filter. Their first approach directly optimizes the likelihood function of the innovation sequence (a product of the likelihoods for each innovation since the innovations are white for an optimal estimator) generated by the EKF using a standard constrained nonlinear programming strategy such as sequential quadratic programming. The EM method maximizes the conditional density function of the complete data set of the states and measurements. When the parameters are known the extended EM algorithm provides the estimates for the noise covariances based on algebraic relations without the need for any derivative calculation.

Recently Zagrobelny and Rawlings[111] (2014) similarly proposed a ML method to identify $\mathbf{Q}$ and $\mathbf{R}$ assuming the parameters are known. By writing the outputs in terms of the process and measurement noises, they form a normal distribution for the sequence of measurements. The variance of this distribution is a function of the unknown noise covariances, and the likelihood is optimized with respect to these covariances. Simulations are used to compare the ML method to several existing methods like the ALS method, an alternate ML method based on the innovations, and an EM method.

Covariance Matching

During the filter pass over the data a number of random variables such as pre, post, and smoothed states arise and these transformed to the measurements along with actual measurements many more statistics are available. Combining these one can form innovation, residue, smoothed residues and so on. As mentioned earlier among the five filter equations two refer to the sample and the other three refer to the ensemble characteristics. If the filter is performing well then the sample statistics formed by the above should be internally consistent with their ensemble properties also provided by the filter. These can be matched by running many Monte Carlo simulations. However taking sample statistics at various time instants during a filter pass one can form equations connecting the many random variables, their combinations and their covariances to estimate the unknown covariances. The different approaches of Myers and Tapley[34] (1976), Gemson[55] (1991), Mohamed and Schwarz[70] (1999), Bavdekar et al.[104] (2011), and the present approach are examples for the covariance matching. Different statistics are chosen either over a window or the full data length, after each pass for estimating $\mathbf{R}$ and $\mathbf{Q}$. These estimates could be used in subsequent passes as is necessary. Myers and Tapley[34] (1976) approach using the innovation (available before update) can at times make the $\mathbf{R}$ lose its positive definiteness. This is because $\mathbf{R}$ depends on the difference of two matrices which could at times become negative. In order to overcome such an eventuality they proposed a remedy of simply turning the estimate as positive! Jazwinski[18] (1970) and Mohamed and Schwarz[70] (1999) suggested a more stable statistic based on the residue (available after update) for estimating $\mathbf{R}$ which is the sum of two matrices. The $\mathbf{R}$ estimates are generally more accurate than $\mathbf{Q}$. Simultaneous estimation of $\mathbf{R}$ and $\mathbf{Q}$ is not easy perhaps as they negatively affect the filter performance. These covariance matching techniques are preferred due to their simplicity and speed despite being suboptimal. Gemson[55] (1991) and Gemson and Ananthasayanam[69] (1998) showed the importance of $\mathbf{P_0}$ for parameter estimation. Similarly many statistics are available and are utilized for estimating $\mathbf{Q}$.

The Correlation Technique

The innovation theorem of Kailath[14] (1970) states that the innovation sequence is zero mean white Gaussian. Kailath further stated that if any gain other than the optimal is used then estimates will be suboptimal and the innovation mean will be non zero and the

autocorrelation will not follow the Kronecker delta function. Also if the covariance of the innovations are not as expected they are indicative of the choice of any or all of the system matrices as well as the covariances are incorrect.

The correlation technique pioneered by Mehra[15,20] (1970, 1972) and Carew and Belanger[21] (1973) and Belanger[26] (1974) is the earliest and highly cited method for determining the unknown covariances. Mehra's approach is based on the properties of the innovations process that must be white and Gaussian. Starting from an assumed value for the unknown $\mathbf{R}$ and $\mathbf{Q}$ an initial estimate for the steady state Kalman gain is obtained. This sequence is checked to see whether the particular Kalman gain implemented generates a statistically acceptable white noise sequence. However the Kalman gain can take correct value even when $\mathbf{R}$ and $\mathbf{Q}$ are far away from their true values. This is because different combinations of $\mathbf{R}$ and $\mathbf{Q}$ can lead to the same gain.

Later Neethling and Young[24] (1974) noted large covariances from Mehra's approach due to some other deficiencies. The approach of Oussalah and De Schutter[76] (2001) improves Mehra's[15] (1970) and Carrew's and Bellanger's (1973) approaches by incorporating information about the quality of the innovation estimates leading to a weighted least squares methodology instead of the earlier least squares methodology. The weights are determined using a Bhattacharyya distance criterion between the ideal probability and the distribution referring to the current first and second order statistics of the autocorrelation functions. The latter helps to generate a convergent sequence to the steady state filter, which after some manipulations allows one to determine the values of the noise statistics $\mathbf{Q}$ and $\mathbf{R}$.

The latter work of Odelson et al.[94] (2006) showed that based on some counter examples the mathematical conditions regarding the system and measurement matrices are not sufficient and not necessary in Mehra's[15] work (1970). Also they showed that the variance estimates of Mehra are larger. They proposed a method called constrained Autocovariance Least Squares (ALS) method corrected the above and obtained a much smaller and better estimates and none negative as shown in Fig. 3.1 for a typical case.

Some Approaches in an Airplane Flight Test Data Analysis

In an airplane flight test data analysis many approaches have been suggested to determine the parameters and the noise covariances. For the sake of brevity we will only briefly men-
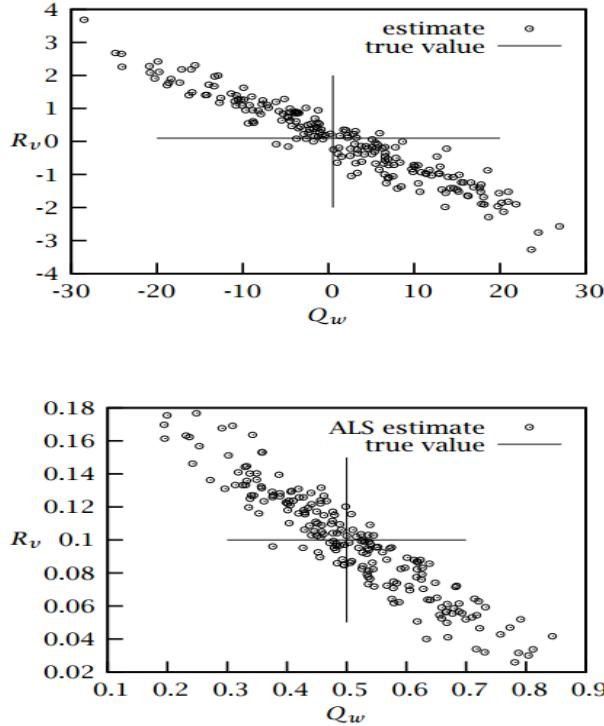
Figure 3.1: Estimates of Q_w and R_v (**Q** and **R** in our notation) using

Mehra's method (top) and ALS method (bottom)

Credit: Odelson et al.[94] (2006)

tion them as natural formulation of Schultz[33] (1976), innovation, a combined formulation which takes the advantage of natural and innovation called MMLE3 of Maine and Iliff (1980), the combined formulation of Ishimoto[64] (1997) to solve the practical problems in natural and innovation formulations. A more detailed discussion on aircraft flight test data analysis can be found in Klein and Morelli[96] (2006) and Jategaonkar[95] (2006).

One may ask the question as to why there are so many formulations for solving an optimization problem. The reason is the unknowns do not occur in a simple way in the cost function, and there are many transformed variables with which one tries to solve for the basic unknowns, the size and the required compatibility conditions among the transformed variables lead to the many difficulties not found in the classical optimization problems. An analogy may be given as when one tries simple difference schemes to solve a problem in Computational Fluid Dynamics one could invariably find himself in distress. A similar problem generally arises in using standard optimization techniques in tuning the Kalman filter for the purpose of system identification and estimation.

Some Other Approaches

Valappil and Georgakis[72] (2000) used the available information about the model parametric uncertainties and translate this information to the process noise covariance $\mathbf{Q}$. They propose two methods called linearized and Monte Carlo approaches. They also assume that $\mathbf{R}$ is readily available from the measuring instrument characteristics. The Monte Carlo simulations are run with parameter values sampled from the assumed normal distribution, with means and covariances. Then the difference between the nominal and the random dynamical state trajectory over many simulations is taken to provide the process noise at any time instant to be used in the filter equations.

Some attempts have been made like Powell[82] (2002) using the simplex method, Oshman and Shaviv[73] (2000) using the genetic algorithm, and controlled random search (Anilkumar[71] 2000). However when the dimension, nonlinearlty, and the range of search space become large these could become computationally prohibitive and could lead to local minima.

Manika Saha et al.[110] (2014) felt that $\mathbf{X_0}$ and $\mathbf{P_0}$ are not critical since they affect only the initial transient filter performance and $\mathbf{P_0}$ has to be decided by designers since $\mathbf{P}$ changes as the filter operation proceeds reaching a steady state if only the system dynamics does not substantially change and a suitable choice of $\mathbf{R}$ can be easily made. Thus for a chosen suitable values for $\mathbf{X_0}$, $\mathbf{P_0}$, and $\mathbf{R}$ they identified the critical $\mathbf{Q}$ by forming two metrics based on the innovation covariance. These vary from zero to the number of measurements and vice versa as $\mathbf{Q}$ changes from zero to infinity they proposed that $\mathbf{Q}$ can be chosen around the intersection point of these two cost functions.

Lau and Lin[106] (2011) also discuss the limitations of simulated annealing and particle optimization techniques for filter tuning. One can summarize that deterministic or probabilistic optimization approaches do not appear to be efficient for solving the filter tuning problem. Hence we tried to see if a recursive filtering approach would work and fortuitously it had worked and will be demonstrated subsequently.

3.7.3 Earlier versus the Present Approach for Filter Tuning

Much of the earlier research work have in fact and rightly so concentrated their effort in using simulated data to tune the filter off line for obtaining the statistics to be used later

for on line and real time applications. While this is commendable one important feature has been forgotten in such approaches. Many variations in the methods used could lead to convergence and one could feel happy about the same. The convergence of any technique even in simulation studies is no guarantee for a proper solution to the problem. Even the simple case of a linear fit to a set of data many variants tend to different results. It is like a necessary condition but not a sufficient one. It is just the reason that we still lean on simulation studies but wherein exact solution is available to the analyst. Hence presently the filter methods have been applied firstly to some very simple cases such as a constant signal, a ramp function, spring-mass-damper system, longitudinal, and lateral motion of an airplane for which when only the measurement noise exists ($\mathbf{R} > 0$) and the process noise does not exist ($\mathbf{Q} = 0$). For such a situation exact reference results are derivable by using the Newton Raphson technique and these can be compared with filter generated results. Subsequently when the process noise in included in the system under study one is in an uncharted territory and for filter tuning it is necessary to look for the consistency based on a comparison between the injected measurement and process noise values. Further many other cost functions based on balancing the states and measurement equations that are introduced help to move from deceptive to decisive solutions.

There are five basic filter operations namely the state and covariance propagation, Kalman gain evaluation, and the state and covariance updates. The state propagation and update refers to sample values and the covariance propagation, its update, and the Kalman gain refer to the population characteristics. The estimation of $\mathbf{R}$ and $\mathbf{Q}$ is possible provided one utilizes appropriate choice of the estimation 'statistics' based on many quantities that arise in the filter operation like the pre and post filter states as well as the ones derivable from the measurements such as the innovation, residue, smoothed and their covariances. The behaviour of the ensemble characteristics of the samples should be consistent with population characteristics.

The above helped to guide us in the choice of appropriate initial filter statistics namely $\mathbf{X_0}, \mathbf{P_0}, \Theta$, $\mathbf{Q}$ and $\mathbf{R}$. In particular after the first filter pass (using some initial statistics) through the data which is a must it has guided the way the filter statistics have to be set from the second iteration onwards. It helps to answer the questions like what should be the $\mathbf{P_0}$ for the states and the parameters be derived from the value at the end of the pass to be used in the next iteration, should they be full, diagonal, or even zero, whether the $\mathbf{Q}$ has to be injected into the state and/or the augmented parameter states. Since the reference

NR estimates of the parameters, the Cramer Rao bound (CRB), and the measurement noise are available it has been possible to settle such above questions and in particular the CRB played a crucial role to guide the choices for the above. Generally the Kalman filter provides fairly acceptable estimates for the parameters but unless the tuning is good it is very difficult to match the CRBs from the EKF with the NR values.

There are many quantities that occur at various times during the filter pass through the data. The present work has shown that a simultaneous update and convergence of **R** and **Q** towards proper values are possible provided one utilizes appropriate choice of the estimation 'statistics' among the many quantities that arise in the filter operation like the pre and post filter states as well as the ones derivable from the measurements such as the innovation, filtered residue, smoothed residue and their covariances.

3.8 Present Approach for Tuning Filter Statistics

Fundamentally the Estimation Theory (ET) is an optimization problem. Hence a suitable cost function **J** has to be chosen. In general there are many estimation algorithms which have desirable characteristics (Beck and Arnold[37] 1977, Goodwin and Payne[38] 1977, Sorenson[45] 1980) that can be utilised to solve an estimation problem. Essentially there are two elements in ET (i) Defining a cost function, (ii) Adopting a suitable algorithm to minimize the cost function. The Likelihood cost (L) for normally distributed error (e) (Bohn[77] 2000) is given by

$$L(\Theta) = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2} (e_k)^T A_k^{-1} (e_k) + log(det(A_k)) \qquad (3.1)$$

where A is the error covariance matrix and det(A) represent determinant of matrix A. It may be noted that the unknown parameters (Θ) occurs implicitly and not explicitly in the cost function L. Since the filter provides many quantities it is possible to have many more terms in the cost function which is a function of errors in the initial state and parameter estimates, process noise driving the system and the measurement noise introduced by the measurement system, each weighted appropriately through suitable weighting covariance matrices. Thus the new cost function **J** in a weighted least square sense accounts for (i) A priori knowledge about the initial estimates, (ii) Balancing the measurement equations.

(iii) Balancing the system equations. One can call this as generalized MLE ($\mathbf{J}$),

$$\mathbf{J} = \mathbf{J0} + \mathbf{J1} + \mathbf{J2} + \mathbf{J3} + \mathbf{J4} + \mathbf{J5} + \mathbf{J6} + \mathbf{J7} + \mathbf{J8} \qquad (3.2)$$

whose terms are defined as

$$\mathbf{J0} = \frac{1}{2}(\mathbf{X_0} - X_t)^T \mathbf{P_0}^{-1}(\mathbf{X_0} - X_t)$$

$$\mathbf{J1} = \frac{1}{N}\sum_{k=1}^{N}(Z_k - h(X_{k|k-1}))^T S1_k^{-1}(Z_k - h(X_{k|k-1}))$$

$$\mathbf{J2} = \frac{1}{N}\sum_{k=1}^{N}(Z_k - h(X_{k|k}))^T S2_k^{-1}(Z_k - h(X_{k|k}))$$

$$\mathbf{J3} = \frac{1}{N}\sum_{k=1}^{N}(Z_k - h(X_{k|N}))^T S3_k^{-1}(Z_k - h(X_{k|N}))$$

$$\mathbf{J4} = \frac{1}{N}\sum_{k=1}^{N}(Z_k - h(Xd_{k|N}))^T (Z_k - h(Xd_{k|N}))$$

$$\mathbf{J5} = \frac{1}{N}\sum_{k=1}^{N}(Z_k - h(X_{k|k-1}))^T S1_k^{-1}(Z_k - h(X_{k|k-1})) + log(det(S1_k))$$

$$\mathbf{J6} = \frac{1}{N}\sum_{k=1}^{N} w1_{k|N}^T W1_k^{-1} w1_{k|N}$$

$$\mathbf{J7} = \frac{1}{N}\sum_{k=1}^{N} w2_{k|N}^T W2_k^{-1} w2_{k|N}$$

$$\mathbf{J8} = \frac{1}{N}\sum_{k=1}^{N} w3_{k|k}^T W3_k^{-1} w3_{k|k}$$

The 'S' and 'W' are functions of the second order moments given by

$$S1_k = H_k P_{k|k-1} H_k^T + \mathbf{R}$$

$$S2_k = -H_{k|k} P_{k|k} H_{k|k}^T + \mathbf{R}$$

$$S3_k = -H_{k|N} P_{k|k-1} H_{k|N}^T + \mathbf{R}$$

$$W1_k = -P_{k|N} - F_{k-1|N} P_{k-1|k-1} F_{k-1|N}^T + P_{k,k-1|N} F_{k-1|N}^T + P_{k,k-1|N}^T F_{k-1|N} + \mathbf{Q}$$

$$W2_k = -P_{k|N} - Fd_{k-1|N} P_{k-1|k-1} Fd_{k-1|N}^T + P_{k,k-1|N} Fd_{k-1|N}^T + P_{k,k-1|N}^T Fd_{k-1|N} + \mathbf{Q}$$

$$W3_k = P_{k|k-1} - P_{k|k}$$

If the initial states are known then J0 is not necessary but if they are unknown, their estimate and covariance can be obtained respectively by the smoothed estimates $\mathbf{X_{0|N}}$ and $\mathbf{P_{0|N}}$. The cost $\mathbf{J1, J2}$ and $\mathbf{J3}$ are expected to tend towards the number of measurements

(*m*). The cost **J6, J7** and **J8** defined for states with process noise are expected to tend towards the number of states (n). The cost **J4** is expected to tend towards the trace of **R** for **Q** = 0 case and **J5** is the negative log likelihood function.

3.8.1 Choice of X_0

Commencing from an assumed reasonable initial choice for X_0, P_0, Θ, **R** and **Q** the first filter pass through the data is made. Then a backward smoothing procedure is carried out. The Rauch Tung and Striebel (RTS[12] 1965) and Modified Bryson Frasier (MBF[36] 1977) smoothing gave almost the same numerical estimates, the RTS was used as a standard smoothing procedure throughout the present work. The smoothing leads to the best possible state and parameter estimates as well their covariances. It may be noted after smoothing the state estimates and their covariances change but not that of the parameters. We next describe how the above choices are updated for further filter passes through the data to eventually converge which denotes statistical equilibrium. If the exact value of X_0 if not given it can be obtained by the smoothed estimates. In such probabilistic approaches any number of unknowns are only determined in a probabilistic sense and thus contain uncertainty due to the percolation of all the noise effects over all the unknowns. The RTS smoothing equations for discrete time instants k = N-1, N-2,... 0 are given by

$$K_{k|N} = P_{k|k}F_k P_{k+1|k}^{-1}$$
$$X_{k|N} = X_{k|k} + K_{k|N}(X_{k+1|N} - X_{k+1|k})$$
$$P_{k|N} = P_{k|k} + K_{k|N}(P_{k+1|N} - P_{k+1|k})K_{k|N}^T$$

where $K_{k|N}, X_{k|N}$ and $P_{k|N}$ is the smoothed gain, smoothed state estimate and smoothed state covariance matrix respectively. All the other variables have their usual meaning which are obtained from the EKF. After an estimate of the initial state ($X_{0|N}$) if one uses the smoothed initial covariance ($P_{0|N}$) for tuning the filter then the parameter CRBs are not properly estimated but has to be scaled up as seen in the simulated case studies in Chapter 4.

3.8.2 Choice of $\mathbf{P_0}$

The choice of $\mathbf{P_0}$ for the next filter pass is very tricky. If one were to take the smoothed initial state covariance ($P_{0|N}$) and use it as the $\mathbf{P_0}$ for the next pass then the final covariance keep on decreasing with further filter passes and eventually tend towards zero. We know that such a fact is unrealistic. In order to remedy the above behaviour the final covariance at the end of the pass was scaled up by the number of data points (N) and used at the beginning of the next pass. The only heuristic reasoning that can be provided from statistics is based on the fact that the mean from a sample has an uncertainty P that keeps decreasing with sample size as P/N where P is the population variance. Since in the filter steps the estimates and its update refer to the sample and the other covariance propagation, its update, and the calculation of the Kalman gain refer to the ensemble characteristics before every filter pass we carry out the above scale up method to obtain the $\mathbf{P_0}$ for the next filter pass. The importance of scaling up is discussed in Appendix-C. We also propagated backwards the final covariance using the estimated parameters and it turned out that there was not much of a difference with the $\mathbf{P_0}$ that was used in the forward pass. Of course the above process does not end and we have to further modify the above $\mathbf{P_0}$ to obtain the best possible CRB after some passes. The scaled up $\mathbf{P_0}$ is a full matrix. Many changes such as using only the diagonal elements and many more variations were tried out. Finally the reference $\mathbf{P_0}$ to obtain the proper CRB for the parameter estimates turns out to have all the elements are zero except the diagonal elements corresponding to the parameters. If all the elements of the parameter covariances were included and the state covariances set to zero, it did not make much of a difference in the final results. The scaled up $\mathbf{P_0}$ (Shyam[109] 2014) is given by

$$\mathbf{P_0} = N \times P_{N|N} \tag{3.3}$$

With the above changes, the surprising feature is to set the initial covariance as zero even for the states which are measured. The usual recommendation (Mehra[15] 1970) in such cases is to set $\mathbf{P_0} = \mathbf{R}$. Even by using the IIM (Inverse of Information Matrix) approach of Gemson[55] (1991) obtained the same estimate as $\mathbf{R}$ for $\mathbf{P_0}$. The IIM is given by

$$\mathbf{P_0} = \left[\frac{1}{N} \sum_{k=1}^{N} F_{k-1}^T H_k^T R^{-1} H_k F_{k-1} \right]^{-1} \tag{3.4}$$

3.8.3 Estimation of R and Q using EM Method

The expression for **R** and **Q** using the Expectation Maximisation (EM) method extended to a non-linear system by Bavdekar et al.[104] (2011) is given by,

$$\mathbf{R} = \frac{1}{N} \sum_{k=1}^{N} E\left[v_k v_k^T | Z_N\right] \tag{3.5}$$

$$\mathbf{Q} = \frac{1}{N} \sum_{k=1}^{N} E\left[w_k w_k^T | Z_N\right] \tag{3.6}$$

Estimation of R

Consider the measurement noise, $v_k = Z_k - h(X_k)$ which can be approximated using first order Taylor series expansion around the smoothed estimate, $X_{k|N}$ given by

$$v_k \approx Z_k - h(X_{k|N}) - H_{k|N}\tilde{X}_{k|N}$$

where $H_{k|N} = \frac{\partial h}{\partial X}|_{X=X_{k|N}}$ and $\tilde{X}_{k|N} = X_k - X_{k|N}$

$$v_k v_k^T = Z_k Z_k^T - Z_k h^T(X_{k|N}) - Z_k \tilde{X}_{k|N}^T H_{k|N}^T - h(X_{k|N})Z_k^T + h(X_{k|N})h^T(X_{k|N})$$
$$+ h(X_{k|N})\tilde{X}_{k|N}^T H_{k|N}^T - H_{k|N}\tilde{X}_{k|N}Z_k^T + H_{k|N}\tilde{X}_{k|N}h^T(X_{k|N}) + H_{k|N}\tilde{X}_{k|N}\tilde{X}_{k|N}^T H_{k|N}^T$$

We know that

$$E[\tilde{X}_{k|N}] = E[X_k - X_{k|N}] = 0$$
$$E[\tilde{X}_{k|N}\tilde{X}_{k|N}^T] = E[(X_k - X_{k|N})(X_k - X_{k|N})^T] = P_{k|N}$$

Thus the conditional expectation is given by

$$E\left[v_k v_k^T | Z_N\right] = Z_k Z_k^T - Z_k h^T(X_{k|N}) - h(X_{k|N})Z_k^T + h(X_{k|N})h^T(X_{k|N}) + H_{k|N}P_{k|N}H_{k|N}^T$$

Rearranging the above terms and using Eq-3.5, we get

$$\mathbf{R} = \frac{1}{N} \sum_{k=1}^{N} \left\{ (Z_k - h(X_{k|N}))(Z_k - h(X_{k|N}))^T + H_{k|N}P_{k|N}H_{k|N}^T \right\} \tag{3.7}$$

53

Estimation of Q

Consider the process noise, $w_k = X_k - f(X_{k-1})$ which can be approximated using first order Taylor series expansion around the smoothed estimate, $X_{k-1|N}$ given by

$$w_k \approx X_k - f(X_{k-1|N}) - F_{k-1|N}\tilde{X}_{k-1|N}$$

where $F_{k-1|N} = \frac{\partial f}{\partial X}|_{X=X_{k-1|N}}$ and $\tilde{X}_{k-1|N} = X_{k-1} - X_{k-1|N}$.

$$
\begin{aligned}
w_k w_k^T = {} & X_k X_k^T - X_k f^T(X_{k-1|N}) - f(X_{k-1|N})X_k^T + f(X_{k-1|N})f^T(X_{k-1|N}) \\
& - X_k \tilde{X}_{k-1|N}^T F_{k-1|N}^T + f(X_{k-1|N})\tilde{X}_{k-1|N}^T F_{k-1|N}^T - F_{k-1|N}\tilde{X}_{k-1|N}X_k^T \\
& + F_{k-1|N}\tilde{X}_{k-1|N}f(X_{k-1|N})^T + F_{k-1|N}\tilde{X}_{k-1|N}\tilde{X}_{k-1|N}^T F_{k-1|N}^T
\end{aligned}
\tag{3.8}
$$

The following results are used in calculating the conditional expectation in Eq-3.6

$$
\left.
\begin{aligned}
E[X_k X_k^T | Z_N] &= X_{k|N}X_{k|N}^T + P_{k|N} \\
E[X_k X_{k-1}^T | Z_N] &= X_{k|N}X_{k-1|N}^T + P_{k,k-1|N}
\end{aligned}
\right\}
\tag{3.9}
$$

The lag-one covariance, $P_{k,k-1|N}$ for k = N-1, N-2,... 1 is given by

$$P_{k,k-1|N} = E[(X_k - X_{k|N})(X_{k-1} - X_{k-1|N})^T]$$

$$P_{N,N-1|N} = (I - K_N H_N)F_{N-1}P_{N-1|N-1}$$

$$P_{k,k-1|N} = P_{k|k}K_{k-1|N}^T + K_{k|N}(P_{k+1,k|N} - F_k P_{k|k})K_{k-1|N}^T$$

where $K_{k|N}$ is the smoothed gain at discrete time 'k' obtained from the RTS smoothing algorithm-3.8.1. Using Eq-3.8 and Eq-3.9 we get the conditional expectation as

$$
\begin{aligned}
E\left[w_k w_k^T | Z_N\right] = {} & X_{k|N}X_{k|N}^T + P_{k|N} + X_{k|N}f^T(X_{k-1|N}) - f(X_{k-1|N})X_{k|N}^T \\
& + f(X_{k-1|N})f^T(X_{k-1|N}) - P_{k,k-1|N}F_{k-1|N}^T - P_{k,k-1|N}^T F_{k-1|N} + F_{k-1|N}P_{k-1|k-1}F_{k-1|N}^T
\end{aligned}
$$

Rearranging the above terms and using Eq-3.6, we get

$$\mathbf{Q} = \frac{1}{N}\sum_{k=1}^{N}\{w1_{k|N}w1_{k|N}^T + P_{k|N} + F_{k-1|N}P_{k-1|N}F_{k-1|N}^T - P_{k,k-1|N}F_{k-1|N}^T - P_{k,k-1|N}^T F_{k-1|N}\} \tag{3.10}$$

where $w1_{k|N} = X_{k|N} - f(X_{k-1|N})$

3.8.4 The Present DSDT Method for Estimating Q

We now estimate **Q** in a different way using the difference between the stochastic and dynamical trajectory (DSDT method) by following the extended EM method. The stochastic trajectory with the process noise can be approximated using the first order Taylor series expansion around a nominal point (X_n) as

$$X_k = f(X_{n_{k-1}}) + f'(X_{n_{k-1}})(X_{k-1} - X_{n_{k-1}}) + w_k \tag{3.11}$$

where f' represents partial differentiation operation on f and thus $f' = \frac{\partial f}{\partial X}$. Consider the dynamical trajectory (Xd) without the process noise defined as

$$Xd_k = f(Xd_{k-1})$$
$$Xd_k = f(X_{n_{k-1}}) + f'(X_{n_{k-1}})(Xd_{k-1} - X_{n_{k-1}}) \tag{3.12}$$

where and $Xd_0 = X_0$. It is assumed that the nominal point (X_n) of both the above trajectories are close to the estimated dynamical trajectory ($X_{n_k} \approx Xd_{k|N}$) where $Xd_{k|N} = f(Xd_{k-1|N})$ and $Xd_{0|N} = X_{0|N}$. Subtracting Eq-3.12 from Eq-3.11 we get

$$X_k - Xd_k = f'(Xd_{k-1|N})(X_{k-1} - Xd_{k-1|N} - Xd_{k-1} + Xd_{k-1|N}) + w_k$$
$$w_k = X_k - Xd_k - Fd_{k-1|N}(X_{k-1} - Xd_{k-1}) \tag{3.13}$$

where the dynamical state Jacobian, $Fd_{k-1|N} = \frac{\partial f}{\partial X}|_{X=Xd_{k-1|N}}$.

$$w_k w_k^T = X_k X_k^T - X_k Xd_k^T - X_k X_{k-1}^T Fd_{k-1}^T + X_k Xd_{k-1}^T Fd_{k-1}^T - Xd_k X_k^T + Xd_k Xd_k^T$$
$$+ Xd_k X_{k-1}^T Fd_{k-1}^T - Xd_k Xd_{k-1}^T Fd_{k-1}^T - Fd_{k-1|N}X_{k-1}X_k^T + Fd_{k-1|N}X_{k-1}Xd_k^T$$
$$+ Fd_{k-1|N}X_{k-1}X_{k-1}^T Fd_{k-1}^T - Fd_{k-1|N}X_{k-1}Xd_k^T Fd_{k-1}^T + Fd_{k-1|N}Xd_{k-1}X_k^T$$
$$- Fd_{k-1|N}Xd_{k-1}Xd_k^T - Fd_{k-1|N}Xd_{k-1}X_{k-1}^T F_{k-1|N}^T + Fd_{k-1|N}Xd_{k-1}Xd_k^T Fd_{k-1}^T$$

The additional results apart from Eq-3.9 used in calculating the conditional expectation in Eq-3.6 are

$$
\left.\begin{aligned}
E[X_k Xd_k^T|Z_N] &= E[X_k|Z_N]E[Xd_k^T|Z_N] = X_{k|N}Xd_{k|N} \\
E[Xd_k Xd_k^T|Z_N] &= Xd_{k|N}Xd_{k|N}^T + Pd_{k|N} \\
E[Xd_k Xd_{k-1}^T|Z_N] &= Xd_{k|N}Xd_{k-1|N}^T + Pd_{k,k-1|N}
\end{aligned}\right\}
\tag{3.14}
$$

where $Xd_{k|N} = f(Xd_{k-1|N})$ is the predicted dynamical state trajectory without the measurement and process noise using the estimated parameter, $\Theta_{N|N}$. Using Eq-3.9 and Eq-3.14 we get

$$
\begin{aligned}
E\left[w_k w_k^T|Z_N\right] = {}&w2_{k|N}w2_{k|N}^T + P_{k|N} + Fd_{k-1|N}P_{k-1|N}Fd_{k-1|N} - P_{k,k-1|N}Fd_{k-1|N}^T \\
&- Fd_{k-1|N}P_{k,k-1|N}^T + Pd_{k|N} + Fd_{k-1|N}Pd_{k-1|N}Fd_{k-1|N} \\
&- Pd_{k,k-1|N}Fd_{k-1|N}^T - Fd_{k-1|N}Pd_{k,k-1|N}^T
\end{aligned}
\tag{3.15}
$$

where $w2_{k|N} = X_{k|N} - Xd_{k|N} - Fd_{k-1|N}(X_{k-1|N} - Xd_{k-1|N})$. Now consider the following term,

$$
\begin{aligned}
Xd_k - Xd_{k|N} &= f(Xd_{k-1}) - f(Xd_{k-1|N}) \\
&\approx f(Xd_{k-1|N}) + Fd_{k-1|N}(Xd_{k-1} - Xd_{k-1|N}) - f(Xd_{k-1|N}) \\
&\approx Fd_{k-1|N}(Xd_{k-1} - Xd_{k-1|N})
\end{aligned}
\tag{3.16}
$$

Using Eq-3.16, we get the covariance of the dynamical trajectory as

$$
\begin{aligned}
Pd_{k|N} &= E[(Xd_k - Xd_{k|N})(Xd_k - Xd_{k|N})^T] \\
&= Fd_{k-1|N}Pd_{k-1|N}Fd_{k-1|N}^T
\end{aligned}
$$

where $Pd_{0|N} = P_{0|N}$ since $Xd_{0|N} = X_{0|N}$. The lag one covariance of the dynamical trajectory is given by

$$
\begin{aligned}
Pd_{k,k-1|N} &= E[(Xd_k - Xd_{k|N})(Xd_{k-1} - Xd_{k-1|N})^T] \\
&= Fd_{k-1|N}Pd_{k-1|N}
\end{aligned}
$$

Substituting the value of $Pd_{k,k-1|N}$ and $Pd_{k|N}$ in Eq-3.15 and using Eq-3.6 we get

$$\mathbf{Q} = \frac{1}{N} \sum_{k=1}^{N} \{w2_{k|N}w2_{k|N}^T + P_{k|N} + Fd_{k-1|N}P_{k-1|N}Fd_{k-1|N}^T - P_{k,k-1|N}Fd_{k-1|N}^T - P_{k,k-1|N}^T Fd_{k-1|N}\} \quad (3.17)$$

If $\mathbf{Q} = 0$ then $X = Xd$ and assuming that $P_{0|N} \approx 0$, $\mathbf{R}$ can be estimated as

$$\mathbf{R} \approx \frac{1}{N} \sum_{k=1}^{N} \left\{ (Z_k - h(Xd_{k|N}))(Z_k - h(Xd_{k|N}))^T \right\} \quad (3.18)$$

3.8.5 Choice of R

The choice of $\mathbf{R}$ for the next filter pass can utilize one appropriate among the many that are possible. Bavdekar et al.[104] (2011) used the smoothed statistic $Z_k - h(X_{k|N})$ for the $\mathbf{R}$ estimation using extended EM method given by,

$$\mathbf{R} = \frac{1}{N} \sum_{k=1}^{N} \left\{ (Z_k - h(X_{k|N}))(Z_k - h(X_{k|N}))^T + H_{k|N}P_{k|N}H_{k|N}^T \right\}$$

We can also use the dynamical trajectory to estimate $\mathbf{R}$ exclusively for $\mathbf{Q}=0$ case using the sample covariance of dynamical residue based on Eq-3.18. The choice of Mohamed and Schwarz (MS) for the $\mathbf{R}$ estimation is filtered residue and is given by,

$$\mathbf{R} = \frac{1}{N} \sum_{k=1}^{N} \left\{ (Z_k - h(X_{k|k}))(Z_k - h(X_{k|k}))^T + H_{k|k}P_{k|k}H_{k|k}^T \right\} \quad (3.19)$$

The choice of Myers and Tapley (MT) for the $\mathbf{R}$ estimation is innovation given by,

$$\mathbf{R} = \frac{1}{N} \sum_{k=1}^{N} \left\{ (Z_k - h(X_{k|k-1}))(Z_k - h(X_{k|k-1}))^T - H_k P_{k|k-1} H_k^T \right\} \quad (3.20)$$

Thus the above three equations use respectively smoothed, after update, and before updated states, the measurement and their corresponding covariances. All the measurement noise statistics innovations, filtered residue, smoothed residue and dynamical residue are assumed to be zero mean. We note that the smoothed residue is the best statistic for $\mathbf{R}$ estimation for both $\mathbf{Q} = 0$ and $\mathbf{Q} > 0$ case.

3.8.6 Choice of Q

The choice of **Q** for the next filter pass can utilize one appropriate among the many that are possible. Bavdekar et al.[104](2011) used the smoothed statistic $X_{k|N} - f(X_{k-1|N})$ for the **Q** estimation using extended EM method given by,

$$\mathbf{Q} = \frac{1}{N} \sum_{k=1}^{N} \left\{ w1_{k|N} w1_{k|N}^T + P_{k|N} + F_{k-1|N} P_{k-1|N} F_{k-1|N}^T - P_{k,k-1|N} F_{k-1|N}^T - P_{k,k-1|N}^T F_{k-1|N} \right\}$$

where $w1_{k|N} = X_{k|N} - f(X_{k-1|N})$. The present work introduced the DSDT statistic for **Q** (section-3.8.4) and is given by

$$\mathbf{Q} = \frac{1}{N} \sum_{k=1}^{N} \left\{ w2_{k|N} w2_{k|N}^T + P_{k|N} + Fd_{k-1|N} P_{k-1|N} Fd_{k-1|N}^T - P_{k,k-1|N} Fd_{k-1|N}^T - P_{k,k-1|N}^T Fd_{k-1|N} \right\}$$

where $w2_{k|N} = X_{k|N} - Xd_{k|N} - Fd_{k-1|N}(X_{k-1|N} - Xd_{k-1|N})$. Mohamed and Schwarz (MS) used innovations and gain for estimating **Q** given by,

$$\mathbf{Q} = K_N \left\{ \frac{1}{N} \sum_{k=1}^{N} (Z_k - h(X_{k|k-1}))(Z_k - h(X_{k|k-1}))^T \right\} K_N^T \tag{3.21}$$

The choice of Myers and Tapley (MT) for **Q** is $w3_{k|k} = X_{k|k} - X_{k|k-1}$ and is given by,

$$\mathbf{Q} = \frac{1}{N} \sum_{k=1}^{N} \left\{ w3_{k|k} w3_{k|k}^T - \left(F_{k-1} P_{k|k-1} F_{k-1}^T - P_{k|k} \right) \right\} \tag{3.22}$$

All the process noise statistics, $w1_{k|N}, w2_{k|N}$ and $w3_{k|k}$ are assumed to be zero mean. We note that the smoothed statistics $w1_{k|N}$ and $w2_{k|N}$ give identical results and are the best for the **Q** estimation.

3.9 Adaptive Tuning Algorithm and the Reference Recursive Recipe

The different methods and options for filter tuning forms a part of sensitivity study on different simulated case studies which are discussed below,

- **P_0** can be estimated by Scale up, Inverse of Information Matrix = IIM or by Smoothing which will have to be scaled up and all of which has to be further modified as in the first option for obtaining proper results.

- Options for **P_0** which can be split as cov ([State-S;Parameter-P]) are, 1 - [0,0;0,✓],

2 -Diagonal matrix, 3 - Full matrix. The checkmark ($\checkmark$) represent a non-zero value at the indicated position. The 'cov (.)' represents covariance matrix.

- The process noise **Q** can be estimated by EM, DSDT, MT or MS method.

- Options for **Q** = cov ([S;P]) are, 1 - [$\checkmark$,0;0,0], 2 - Diagonal matrix, 3 - Full matrix.

- The measurement noise **R** can be estimated by EM, MT, MS or using Eq - 3.18.

The following steps explain the recursive or iterative algorithm for tuning the EKF.

1. Given the system model and the measurements the first iteration of EKF is carried out with guess values of $\mathbf{X_0}$, $\mathbf{P_0}$, Θ, **R** and **Q**.

2. Run the extended RTS smoother using the filtered data to get the smoothed state estimate $X_{k|N}$ and the corresponding smoothed covariance $P_{k|N}$.

3. The $\mathbf{P_0}$ can be estimated by Scale up (Eq - 3.3), IIM (Eq - 3.4) or by smoothing ($P_{0|N}$) which will have to be scaled up and modified using the the first option for obtaining proper results.

4. The measurement and process noise covariance can be estimated by any of the method and options as discussed in section-3.8.5 and 3.8.6.

5. EKF is run using the estimates of $\mathbf{X_0}$, $\mathbf{P_0}$, Θ, **Q** and **R** in the next few iterations until statistical equilibrium is reached.

6. Many simulation runs (say 50) are carried out by repeating the above steps using different injected measurement (v) and process noise (w) sequences.

7. Different cost functions (**J1** to **J8**) are checked for convergence.

For the **Q** = 0 case the value of **Q** is set at 10^{-10} or lower for all iterations to help the filter that would otherwise generate a pseudo **Q** and then slowly grind it to zero in hundreds of iterations. For the **Q** > 0 case if any of the states is known to have **Q** = 0 then it can be set at 10^{-10} or lower. For **Q** = 0 case one can even estimate **R** by ignoring the second order terms (assuming P$\rightarrow$ 0). It is of interest to note that for Q > 0 case unless the second order terms of the filter output covariance terms are also included in (i) the estimate for **R** and **Q** using the EM option and (ii) the estimate for **R** using the EM together with **Q** using the DSDT option the estimates for **R** and **Q** do not converge to the proper value. A comparative study is conducted among the different filter tuning methods and the best adaptive recipe is proposed as a reference method as shown below

Reference Recursive Recipe

$\mathbf{Q} = 0$	$\mathbf{Q} > 0$
$\mathbf{X_0}$: Given or $X_{0\|N}$	$\mathbf{X_0}$: Given or $X_{0\|N}$
$\Theta : \Theta_{N\|N}$	$\Theta : \Theta_{N\|N}$
$\mathbf{P_0}$: Scaled up-$[0,0;0,\checkmark]$	$\mathbf{P_0}$: Scaled up-$[0,0;0,\checkmark]$
$\mathbf{Q} : 10^{-10}$-$[\checkmark,0;0,0]$	$\mathbf{Q}$: EM/DSDT-$[\checkmark,0;0,0]$
$\mathbf{R}$: EM-diag	$\mathbf{R}$: EM-diag

Note : The $\mathbf{X_0}$ in all cases is either given or obtained by the smoother. The smoothed initial estimate which can be split as $X_{0|N} = (x_{0|N}, \Theta_{0|N})^T$ including both state and parameter. Hence the unknown parameters are not shown explicitly as Θ. The estimated parameter is taken as $\Theta_{N|N}$ obtained from $X_{N|N} = [x_{N|N}, \Theta_{N|N}]$ with covariance P_Θ obtained at the end of the final filter pass over the data, $P_{N|N} = [P_{xx}, P_{x\Theta}; P_{\Theta x}, P_\Theta]$.

Table 3.1: The Triplets Occurring in the Discussion on Kalman Filter

	Aspects	Three Levels or Approaches		
1	Triplet of any physical theory	Theory (T), Evolving truth, Model, Information	Experiments (E), Newer facts, Measurements, Innovation	Progress (P), Ultimate truth, Mind, Intuition
2	Contributions	Mathematicians	Engineers	Applied researchers
3	Cycle of progress	Analyze /Adapt / Predict By Theory	Accumulate / Accommodate / Capture By Experiments	Assimilate /Adjust / Correct by proper mix of T and E.
4	Strong and weak points	Information without Insight	Implementation without Innovation	Intelligent but Immature
5	Other possible Expressions	Nursery, Training, Writing/Working Karma, How?	Nature, Talent, Reading, Bhakthi, Why?	Nurture, Temperament, Thinking, Jnana, Who?
6	Treatment in Teaching	Examples like Babylonian	Rigorous as in Euclidian	Intuitive
7	Development of estimation theory	Empirical Correlations or corrections in mythological times.	Equations and Calculations during medieval times.	Computational approach using powerful computers.
8	Developments in handling model (**M**) process noise **Q**, and measurement noise **R**	Ancient Indian Astronomers: **M** is known, no **Q** but **R** exists	Medieval Period: Gauss: **M** is known, no **Q** but **R** exists	In 20th century for Kalman: **M**, **Q**, and **R** known. Present Day: Unknown or inaccurately known **M, Q**, and **R** at times even inputs.
9	Filter Tuning:	**R** is relatively easy. Fairly Objective **R** from data.	P_0 is somewhat tricky. Controls Initial transient.	**Q** is notoriously difficult. Controls the Filter learning and Steady state.
10	Algorithm and noise handled.	Least Squares or Equation Error Method handles **Q**.	Output Error Method handles **R**.	Method of Maximum Likelihood Estimation handles both **R** and **Q**.
11	Characteristics	Controllability	Observability	Identifiability
12	General types of data processing	Offline. Tune the filter parameters	Online. Use the offline tuned parameters	Real time. Efficient numerical Implementation

Chapter 4

Simulated Case Studies

In this Chapter we carry out simulation studies commencing from the simplest case having one or two unknown parameters and noise components and move on to more involved case of dynamical systems having around a dozen or more of unknown parameters and noise components. The dynamical systems are (i) a constant signal, (ii) a ramp, (iii) a spring, mass, damper system with a weak non linear spring constant, (iv) the longitudinal motion of an airplane and (v) the lateral directional motion of an airplane. We study all the above systems by including first only the measurement noise $\mathbf{R}$ and later include the process noise $\mathbf{Q}$ as well. Except in the simplest of cases the governing dynamical equations are integrated using ode23, ode45, or using the appropriately derived transition matrices in the MATLAB$^{®}$ all of which provided equally accurate state trajectories. Then for studies by including $\mathbf{R}$ and $\mathbf{Q}$ using the outputs from the default normal random number generator in MATLAB$^{®}$ the proper noise samples were added to the appropriate state and measurements. The accuracy of this approach in generating the stochastic trajectories with process noise input was also checked against the Van Loan's[39] (1978) procedure. The effect of higher order filters like Second Order Kalman Filter and Iterated Extended Kalman Filter (Jazwinski[18] 1970) on non linear complex systems (Longitudinal and Lateral dynamics of Aircraft system) are also studied. The detailed numerical inputs and figures used in the simulation of the various case studies are given in Table-4.1.

In all simulated system study, the guess value of $\mathbf{P_0}$ chosen is 10^{-1} for all states which is assumed to be a diagonal matrix in the first iteration. The guess value of $\mathbf{Q}$ chosen is 10^{-1} for all states and zero for the augmented parameters. The guess value of $\mathbf{R}$ chosen is 2^{-1} for all measurement channels. The initial parameters are chosen to be within $\pm 20\%$ error. A total of N = 100 measurement data are simulated with the time varying from 0 to

10 seconds in very small steps of $\delta t = 0.1$ s. For zero process noise case, the maximum number of iteration is set to 20 over 50 simulations and for non zero process noise case it is set to 100 over 50 simulations for obtaining generally four digit accuracy (though not necessary) in the results as presented in various tables in the report. In the present reference procedure it was noticed that generally even if the initial state covariance, initial process and measurement noise variances are varied over a wide range of powers from -3 to +3 together with the initial parameter values being set to zero one can reach the same estimation results for a given data. Thus there is stability with respect to far off initial conditions but sensitivity in the estimates with different data. Further detailed sensitivity studies were carried out and the results are provided in the various Tables of this Chapter along with brief remarks which are simple and self explanatory. One comment in most tables as unity means near about it. Such studies show that the present reference recursive recipe leads to a non diverging, and consistent filter performance over many simulations and provides better results when compared to earlier approaches.

4.1 Simulated Constant Signal System

Consider a constant signal system for discrete time instants k=1,2,...N is given by

$$x_k = \Theta x_{k-1} + w_k$$

where $x_0 = 10$ and $\Theta_{true} = 1$ is considered as an unknown parameter. The measurement equation is

$$Z_k = HX_k + v_k$$

where the augmented state, $X_k = [x_k, \Theta_k]$, H=[1 0] is the measurement matrix of size $m \times (n + p)$ and $m = n = p = 1$. The numerical values of the noise variances are shown in Table-4.1. All the figures are presented for only one run to prevent cluttering.

4.1.1 Details of Sensitivity Study

The system under consideration is first solved using the reference recipe. Later other sensitivity studies were carried out using other possible variations for $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$. Such studies lead to the conclusion that the reference recipe is just about the best possible and

others may not always help the filter operation without divergence and even if they did may not lead to results as good as the reference recipe. The tabulated results are averaged over 50 simulations which includes

- Θ ratio is the ratio of EKF estimated parameter ($\Theta_{N|N}$) to that of Θ_{true}.

- CRB ratio is the ratio of the square root of *CRB* estimated by the NR-[A.1] method to that of the square root of parameter covariance (P_Θ) estimated by EKF.

- PCRB ratio is the ratio of the square root of Posterior CRB-[A.2] to that of the square root of EKF estimated state covariance at the last time instant.

- Consistency check on the estimated parameters is done using the ratio defined as

$$\text{Consistency ratio} = \frac{\sigma_\Theta}{\text{SIGMA}_{avg}}$$

$$\sigma_\Theta = \left[\frac{1}{50} \sum_{s=1}^{50} (\Theta^s - \bar{\Theta})^2 \right]^{\frac{1}{2}}$$

$$\text{SIGMA}_{avg} = \frac{1}{50} \sum_{s=1}^{50} \sqrt{P_\Theta^s}$$

where 's' is the simulation number, $\bar{\Theta}$ is the sample mean of the estimated parameters over 50 simulations.

- Spread factor is a measure of percentage spread seen in the estimates using both first and second order moments which is defined as

$$\text{Spread factor} = \left[\frac{1}{50} \sum_{s=1}^{50} \sqrt{(\Theta - \Theta^s)^2 + P_{\Theta^s}} \right] \times \frac{100}{|\Theta|}$$

- **R** ratio EKF/True is the ratio of the EKF estimated **R** to that of the true reference value of **R**.

- **R** ratio EKF/NR is the ratio of the EKF estimated **R** to that of the **R** estimated by NR method.

- **Q** ratio is the ratio of the EKF estimated **Q** to that of the true **Q**.

- The mean (μ) and standard deviation (σ) of the cost functions (**J1-J8**) over many simulations.

Q = 0 CASE

1. If the **Q** is allowed to learn it takes thousands of iterations for it to tend to zero. Hence when studying cases with no process noise it is better to set it to a very small value such as 10^{-10}.

2. If **R** is estimated using the EM statistic using smoothed data (i) the Θ and (ii) the **R** estimate is good. It is the choice of $\mathbf{P_0}$ that controls the behaviour of the CRB.

3. The reference case with scaled up $\mathbf{P_0}$ with zero state covariance gives close CRB results obtainable from the modified Newton Raphson (NR) method (section-A.1). If a scaled up or diagonal $\mathbf{P_0}$ which have non zero initial state covariance is used then the filter CRB is higher. For this simple constant signal case, all other possible choices for **R** such as (i) dynamical residue, (ii) smoothed residue ignoring second order term, (iii) filtered residue ignoring second order term, (iv) innovation ignoring second order term, (v) MT and (vi) MS all give very good filter CRBs.

4. If the IIM is used instead of scaled up $\mathbf{P_0}$ with zero state covariance it gives good CRBs. If IIM with non zero initial state covariance is used then the filter estimated CRB is higher.

5. If the smoothed $\mathbf{P_0}$ is used instead of scaled up $\mathbf{P_0}$ then in all the above three combinations it leads to low filter estimated CRB. This is because the smoothed $\mathbf{P_0}$ is much lower than the scaled up $\mathbf{P_0}$ by a factor of the number of time points (N).

6. During smoothing the parameter covariances do not change but only the state covariance changes. Hence if the smoothed $\mathbf{P_0}$ is scaled up and the state covariance is set to zero then it becomes equivalent to the reference case.

7. The value of the cost functions **J1, J2** and **J3** are quite close to unity as is expected, **J4** is the trace of **R** and **J5** is the negative log likelihood cost function.

Q > 0 CASE

1. Further experiments were carried out for the non zero **Q** using the reference recursive recipe among many possible studies a few such as (i) **R** being known, (ii) Θ known, (ii) **R** and Θ and the like as shown in Table-4.3.

2. The next study is the procedure suggested by Bavdekar et al.[104] (2011). This had difficuly in providing consistent parameter estimates. The smoothed $\mathbf{P_0}$ used as the parameter covariance tend to decrease with iterations and the consistency ratio becomes greater than one. As the system complexity increases their procedure requires a stopping criterion using the cost function $\mathbf{J5}$ which may vary for different simulations. The cause of divergence is the process noise injected into the parameter augmented state equations which increase through the iterations as seen in the Fig. 4.68 and Fig. 4.69 referring to the SMD system studied later in the report. It is not desirable to inject process noise into the augmented states corresponding to the parameters in the filter operation. If only diagonal terms are used for $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$ then no stopping criterion is necessary but again the covariance keep on decreasing through iterations. Next if scaled up $\mathbf{P_0}$ is used and the state covariance is reset to zero then the results come very close to the reference approach. If the DSDT statistics is used for estimating $\mathbf{Q}$ then it also provides results very close to the reference case.

3. The next study is by using the MT statistics for $\mathbf{R}$ and $\mathbf{Q}$. Simultaneous estimation of $\mathbf{R, Q}$ is generally not possible. The diagonal terms can become negative at times which has to be forced to be a positive value. There is no proper convergence even if Θ is known. If $\mathbf{R}$ is assumed to be known then the results using only the later half of the filter outputs are somewhat close to the reference recipe. The procedure of Gemson using the MT statistics for $\mathbf{Q}$ and initial $\mathbf{R}$ close to the true value led to reasonably acceptable but not as close as the reference results.

4. We next ran the filter using the MS statistics of $\mathbf{Q}$ and $\mathbf{R}$. Here there was no systematic filter operation and convergence even if $\mathbf{R}$ and/or Θ is known. The procedure of Gemson but using the MS statistics for $\mathbf{Q}$ and initial $\mathbf{R}$ close to the true value led to reasonably acceptable but not as close as the reference results.

5. All the above studies show that even for the case of a constant signal only a proper combination of the choices for $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$ can lead systematically to a set of good results. The other choices appear to present some difficulties both in the filter operation and providing accurate estimates.

4.1.2 Remarks on the Results

We first run the filter assuming $\mathbf{Q} = 0$. It was found that about 20 iterations of the data would suffice. The Fig. 4.1 shows the various parameter estimates and its corresponding variances through cumulative time instants with iterations. The variation of the estimated initial parameters and their variances through iterations are shown in Fig. 4.2. The parameter and the uncertainty reach almost their final estimated values in about 2 and 5 iterations respectively. A similar plot in Fig. 4.3 shows the variation of the estimated measurement noise. The variation of different cost functions (**J1-J5**) through the iterations is shown in Fig. 4.4. The Fig. 4.5 shows the predicted dynamics, filtered and smoothed estimate at the last iteration. The Fig. 4.6 show the innovations, filtered residue and smoothed residue together with the square root of their variance ($\pm\sigma$ bound). In the EKF approach most of the quantities are Gaussian or approximated as quasi Gaussian and one would expect all the above quantities are close to being Gaussian and hence around one third of the total sample points to be outside the σ bound. The injected and estimated measurement noise distributions during the final iteration shown in Fig. 4.7 indicate that they are close to each other. Their autocorrelations are ideally expected to be close to the Kronecker delta function which provides confidence in the proposed algorithm.

The next step is to process the data with process noise ($\mathbf{Q} > 0$). The Fig. 4.9 shows the absolute difference between the iterated and final values with iterations which indicates the accuracy level that one needs and it was found that 100 iterations are required. The variation of the estimated initial parameters and their variances through iterations are shown in Fig. 4.10. The parameter and the uncertainty reach almost their final estimated values in about 5 and 20 iterations respectively. A similar plot in Fig. 4.11 shows the variation of the estimated $\mathbf{R}$ and $\mathbf{Q}$. The variation of different cost functions through the iterations are shown in Fig. 4.12. The cost functions **J1-J3** correspond to the number of measurement (m=1) and in presence of process noise, **J6-J8** correspond to the number of states (n=1). The **J4** in absence of process noise corresponds to the trace of $\mathbf{R}$. The **J5** is the negative log likelihood function whose absolute value is shown in the plot. There is a mismatch in the predicted dynamics and the measurement as seen in Fig. 4.13 indicating the presence of process noise. The subsequent Fig. 4.14 correspond to the earlier Fig. 4.6 of $\mathbf{Q} = 0$ case. The Fig. 4.15 and Fig. 4.16 shows respectively the injected and estimated measurement and process noise samples across time during the final iteration.

Table 4.1: Simulated System Noise inputs and Figures Numbers

Simulated system	State and process noise variance	Measurement and noise variance	Figures (Q = 0)	Figures (Q > 0)
Constant	Constant state (0.05)	Constant state (0.05)	Fig. 4.1, Fig. 4.2, Fig. 4.3, Fig. 4.4 , Fig. 4.5 , Fig. 4.6 , Fig. 4.7	Fig. 4.9 , Fig. 4.10 , Fig. 4.11 , Fig. 4.12 , Fig. 4.13 , Fig. 4.14 , Fig. 4.15 , Fig. 4.16
Ramp	Ramp state (0.05)	Ramp state (0.05)	Fig. 4.21 , Fig. 4.22 , Fig. 4.23 , Fig. 4.24 , Fig. 4.25 , Fig. 4.26 , Fig. 4.27	Fig. 4.28 , Fig. 4.29 , Fig. 4.30 , Fig. 4.31 , Fig. 4.32 , Fig. 4.33 , Fig. 4.34 , Fig. 4.35
SMD	Displacement (0.001) Velocity (0.002)	Displacement (0.001) Velocity (0.004)	Fig. 4.40-4.42 , Fig. 4.43 , Fig. 4.44 , Fig. 4.45 , Fig. 4.46-4.47 , Fig. 4.48-4.49 , Fig. 4.50-4.51	Fig. 4.52 , Fig. 4.53 , Fig. 4.54 , Fig. 4.55 , Fig. 4.56–4.57 , Fig. 4.58-4.59 , Fig. 4.60-4.61 , Fig. 4.62-4.63
Longitudinal motion of aircraft	Angle of attack (0.00005) Pitch rate (0.0001) Pitch angle (0.00005) Forward velocity (0.5)	Angle of attack (0.0001) Pitch rate (0.0001) Pitch angle (0.0001) Forward velocity (1) Normal acceleration (0.1)	Fig. 4.71-4.75 , Fig. 4.76 , Fig. 4.77 ,Fig. 4.78, Fig. 4.79-4.83 , Fig. 4.84-4.88, Fig. 4.89-4.93	Fig. 4.94 , Fig. 4.95 , Fig. 4.96 , Fig. 4.97, Fig. 4.98–4.102 , Fig. 4.103-4.107 , Fig. 4.108-4.112 , Fig. 4.113-4.116
Lateral motion of aircraft	Angle of sideslip (0.000005) Roll velocity (0.0002) Roll angle (0.00005) Yaw rate (0.0001)	Angle of sideslip (0.00001) Roll velocity (0.0001) Roll angle (0.0001) Yaw rate (0.0001) Lateral acceleration (0.005) Roll acceleration (0.004) Yaw acceleration (0.001)	Fig. 4.118-4.132 , Fig. 4.133 , Fig. 4.134 , Fig. 4.135 , Fig. 4.136-4.142 , Fig. 4.143-4.149 , Fig. 4.150-4.156	Fig. 4.157 , Fig. 4.158 , Fig. 4.159 , Fig. 4.160 , Fig. 4.161–4.167 , Fig. 4.168-4.174 , Fig. 4.175-4.181 , Fig. 4.182-4.185

4.1.3 Constant System Tables (Q = 0)

Table 4.2: Sensitivity Study : ($\mathbf{Q}$ = 0) : Constant system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Consistency ratio-NR	Spread factor EKF	Spread factor NR	$\mathbf{R}$ ratio EKF/True	$\mathbf{R}$ ratio EKF/NR	μ of J1-J5	σ of J1-J5	Remarks
Reference adaptive EKF used for non zero process noise case											
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: EM-diag	Extremely slow convergence of $\mathbf{Q}$ taking more than 100 iterations as seen in Fig. 4.8.										
Proposed reference adaptive EKF for zero process noise case, $\mathbf{Q}$ is diagonal matrix fixed to 10^{-10} for all states											
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0000	1.0042	0.9613	0.9577	0.0051	0.0051	0.9761	1.0013	0.9901 0.9901 0.9998 0.0483 -1.9933	0.0000 0.0000 0.0012 0.0065 0.1399	σ_{Θ}= 3.6257e-05 SIGMA$_{avg}$= 3.7718e-05
Using different method and options for $\mathbf{P_0}$											
$\mathbf{P_0}$: Scaled up-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0000	0.4992	0.7767	0.9577	0.0093	0.0051	0.9805	1.0057	0.9802 0.9802 0.9996 0.0481 -1.9392	0.0000 0.0000 0.0011 0.0065 0.1418	Higher filter CRB than that of the NR
$\mathbf{P_0}$: Scaled up-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0000	0.5272	0.8218	0.9577	0.0090	0.0051	0.9794	1.0047	0.9813 0.9813 0.9997 0.0481 -1.9529	0.0000 0.0000 0.0010 0.0065 0.1418	Larger filter spread than that of the NR

Table 4.2: Sensitivity Study : ($\mathbf{Q} = 0$) : Constant system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Consistency ratio-NR	Spread factor EKF	Spread factor NR	$\mathbf{R}$ ratio EKF/True	$\mathbf{R}$ ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
$\mathbf{P_0}$: IIM-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0000	1.0003	1.0186	0.9577	0.0052	0.0051	0.9770	1.0022	0.9900 0.9900 0.9998 0.0485 -1.8054	0.0000 0.0000 0.0012 0.0066 0.1378	Unity CRB ratio
$\mathbf{P_0}$: IIM-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0000	0.4939	0.7709	0.9577	0.0094	0.0051	0.9811	1.0064	0.9800 0.9800 0.9996 0.0484 -1.6845	0.0000 0.0000 0.0011 0.0067 0.1383	Higher filter CRB than the NR
$\mathbf{P_0}$: IIM-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0000	0.4986	0.7793	0.9577	0.0093	0.0051	0.9963	1.0220	0.9709 0.9709 0.9892 0.0487 -1.8111	0.0688 0.0688 0.0710 0.0070 0.1417	Higher filter CRB than that of the NR
$\mathbf{P_0}$: Smoothed-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0000	4.3549	4.1694	0.9577	0.0032	0.0051	0.9670	0.9919	0.9994 0.9994 0.9999 0.0483 -2.0388	0.0000 0.0000 0.0001 0.0065 0.1399	Lower filter CRB than that of the NR
$\mathbf{P_0}$: Smoothed-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0000	3.9982	5.6520	0.9577	0.0058	0.0051	0.9649	0.9898	0.9987 0.9987 0.9998 0.0482 -2.0413	0.0000 0.0000 0.0001 0.0066 0.1419	Lower filter CRB than that of the NR
$\mathbf{P_0}$: Smoothed-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0000	2.3390	3.4089	0.9577	0.0057	0.0051	0.9644	0.9892	0.9988 0.9988 0.9997 0.0482 -2.0420	0.0002 0.0002 0.0002 0.0066 0.1421	Lower filter CRB than that of the NR

Table 4.2: Sensitivity Study : ($\mathbf{Q} = 0$) : Constant system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Consistency ratio-NR	Spread factor EKF	Spread factor NR	$\mathbf{R}$ ratio EKF/True	$\mathbf{R}$ ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
Using different methods for estimating $\mathbf{R}$
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	1.0000	1.0048	0.9619	0.9577	0.0051	0.0051	0.9749	1.0000	0.9914 0.9914 1.0012 0.0483 -1.9933	0.0025 0.0025 0.0028 0.0065 0.1399	*Using dynamical residue as per Eq-3.18
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	1.0000	1.0092	0.9660	0.9577	0.0051	0.0051	0.9664	0.9913	1.0000 1.0000 1.0098 0.0483 -1.9932	0.0000 0.0000 0.0013 0.0065 0.1399	*Using smoothed residue ignoring 2nd order terms in Eq-3.7
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	1.0000	1.0316	0.9875	0.9577	0.0050	0.0051	0.9244	0.9483	1.0455 1.0455 1.0558 0.0483 -1.9918	0.0023 0.0023 0.0023 0.0065 0.1399	*Using filtered residue ignoring 2nd order terms in Eq-3.19
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	1.0000	0.9821	0.9402	0.9577	0.0051	0.0051	1.0268	1.0533	0.9643 0.9643 0.9736 0.0483 -1.9806	0.1599 0.1599 0.1603 0.0065 0.1400	*Using innovations ignoring 2nd order terms in Eq-3.20
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: MS-diag	1.0000	1.0088	0.9657	0.9577	0.0051	0.0051	0.9672	0.9921	0.9993 0.9993 1.0092 0.0483 -1.9933	0.0021 0.0021 0.0021 0.0065 0.1399	Using full expression in Eq-3.19
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: MT-diag	1.0000	1.0020	0.9592	0.9577	0.0051	0.0051	0.9871	1.0126	1.0075 1.0075 1.0172 0.0483 -1.9784	0.1834 0.1834 0.1839 0.0065 0.1407	Using full expression in Eq-3.20

4.1.4 Constant System Tables (Q > 0)

Table 4.3: Sensitivity Study : ($\mathbf{Q} > 0$) : Constant system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
Proposed reference adaptive EKF for non zero process noise									
P_0 : Scaled up-[0,0;0,✓] **Q** : EM-[✓,0;0,0] **R** : EM-diag	1.0000	1.0275	1.0270	0.2906	0.9348	1.0300	0.9898 0.9898 1.0003 0.9880 -1.0448 0.9778 0.9778 0.9898	0.0018 0.0018 0.0019 0.8767 0.1262 0.0089 0.0089 0.0018	σ_Θ= 0.00222 SIGMA$_{avg}$= 0.00216
P_0 : Scaled up-[0,0;0,✓] **Q** : EM-[✓,0;0,0] **R** : Known	1.0000	0.9936	1.0371	0.2907	NA	0.9986	0.9846 0.9846 0.9873 0.9791 -1.0368 0.9783 0.9783 0.9846	0.0586 0.0586 0.1059 0.8448 0.1247 0.0057 0.0057 0.0586	**R** is known

Table 4.3: Sensitivity Study : ($\mathbf{Q} > 0$) : Constant system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of J1-J8	σ of J1-J8	Remarks
$\mathbf{P_0}$: Zero $\mathbf{Q}$: EM-diag $\mathbf{R}$: EM-diag	NA	1.0309	NA	NA	0.9355	1.0280	1.0000 1.0000 1.0006 2.5377 -1.0804 1.0001 1.0001 1.0000	0.0001 0.0001 0.0051 3.5893 0.1198 0.0067 0.0067 0.0001	The parameters are assumed to be known
$\mathbf{P_0}$: Zero $\mathbf{Q}$: EM-diag $\mathbf{R}$: Known	NA	0.9963	NA	NA	NA	0.9989	0.9945 0.9945 0.9869 2.5377 -1.0718 1.0004 1.0004 0.9945	0.0618 0.0618 0.1119 3.5893 0.1180 0.0051 0.0051 0.0618	The parameters and $\mathbf{R}$ are assumed to be known

Table 4.3: Sensitivity Study : ($Q > 0$) : Constant system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**Q** ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
Extended EM algorithm (Bavdekar et al.[104](2011))									
P₀ : Smoothed-full **Q** : EM-full **R** : EM-full	There is no consistency in the parameter estimates since the smoothed **P₀** simply decreases through iterations.								
Modification-1									
P₀ : Smoothed-diag **Q** : EM-diag **R** : EM-diag	1.0000	1.0313	10.8083	0.1680	0.9745	0.9400	0.9993 0.9993 1.0004 1.0079 -1.1020 1.0041 1.0041 0.9993	0.0017 0.0017 0.0017 0.8743 0.1278 0.0030 0.0030 0.0017	No stoping condition. The low spread factor is deceptive since there is no consistency in the parameter estimates with its covariance
Modification-2									
P₀ : Scaled up-[0,0;0,✓] **Q** : EM-diag **R** : EM-diag	1.0000	1.0275	1.0270	0.2906	0.9348	1.0300	0.9898 0.9898 1.0003 0.9880 -1.0448 0.9778 0.9778 0.9898	0.0018 0.0018 0.0019 0.8767 0.1262 0.0089 0.0089 0.0018	The slight differences from the reference case is due to the small **Q** that is generated for the parameters
Modification-3									
P₀ : Scaled up-[0,0;0,✓] **Q** : DSDT-diag **R** : EM-diag	1.0000	1.0270	1.0280	0.2903	0.9360	1.0273	0.9902 0.9902 1.0003 0.9871 -1.0448 0.9785 0.9785 0.9902	0.0021 0.0021 0.0019 0.8753 0.1262 0.0098 0.0098 0.0021	Using DSDT statistics for **Q** estimation given by Eq-3.17

Table 4.3: Sensitivity Study : ($\mathbf{Q} > 0$) : Constant system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of J1-J8	σ of J1-J8	Remarks
MS method as suggested by Mohamed and Schwarz									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MS-diag $\mathbf{R}$: MS-diag	$\mathbf{R}$ can converge to a very large value and the cost functions diverge as seen in Fig. 4.17,4.18. There is no convergence in cost even if Θ is known.								
MS method with $\mathbf{Q}$ being estimated using only last N/2 samples									
$\mathbf{P_0}$: zero $\mathbf{Q}$: MS-diag $\mathbf{R}$: known	NA	1.0100	NA	NA	NA	0.9927	1.0354 1.0354 1.0071 2.5377 -1.0504 1.0983 1.0983 1.0354	0.1629 0.1629 0.1304 3.5893 0.1170 0.2726 0.2726 0.1629	$\mathbf{R}$, Θ is known
MS method assuming $\mathbf{R}$ is known and $\mathbf{Q}$ is estimated using only last N/2 samples									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MS-[✓,0;0,0] $\mathbf{R}$: Known	1.0000	1.0173	0.9643	0.3096	0.8703	1.1872	0.9837 0.9837 1.0067 0.9703 -0.9896 0.9709 0.9709 0.9837	0.1474 0.1474 0.1313 0.8348 0.1242 0.1866 0.1866 0.1474	Stable operation when $\mathbf{R}$ is known
Gemson's approach using MS method									
$\mathbf{P_0}$: IIM-full $\mathbf{Q}$: MS-[✓,0;0,0] $\mathbf{R}$: 0.9*True $\mathbf{R}$	1.0000	1.0249	0.9932	0.3038	0.8703	1.1185	1.0103 1.0103 1.0282 0.9683 -0.9909 1.0020 1.0020 1.0103	0.1469 0.1469 0.1330 0.8346 0.1248 0.1847 0.1847 0.1469	$\delta J = 10^{-6}$ with one loop of $\mathbf{R}$ estimation

Table 4.3: Sensitivity Study : ($Q > 0$) : Constant system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	R ratio EKF/True	Q ratio EKF/True	μ of J1-J8	σ of J1-J8	Remarks
Covariance matching as suggested by Myers and Tapley (MT)									
P_0 : Scaled up-[0,0;0,✓] Q : MT-diag R : MT-diag	Positive definiteness of Q,R is not gauranteed and the cost functions can diverge as seen in Fig. 4.19,-4.20. There is no proper convergence even if Θ is known.								
MT method with Q being estimated using only last N/2 samples									
P_0 : zero Q : MT-diag R : Known	NA	1.0087	NA	NA	NA	0.9933	1.0310 1.0310 1.0068 2.5377 -1.0527 1.0832 1.0832 1.0310	0.1634 0.1634 0.1383 3.5893 0.1182 0.2476 0.2476 0.1634	R, Θ is known
MT method assuming R is known and Q is estimated using only last N/2 samples									
P_0 : Scaled up-[0,0;0,✓] Q : MT-[✓,0;0,0] R : Known	1.0000	1.0053	1.0583	0.2888	NA	0.9884	1.0184 1.0184 1.0066 0.9770 -1.0205 1.0500 1.0500 1.0184	0.1518 0.1518 0.1300 0.8339 0.1258 0.2179 0.2179 0.1518	Stable Operation
Gemson's approach using MT method									
P_0 : IIM-full Q : MT-[✓,0;0,0] R : 0.9*True R	1.0000	1.0249	0.9932	0.3038	0.8703	1.1185	1.0103 1.0103 1.0282 0.9683 -0.9909 1.0020 1.0020 1.0103	0.1469 0.1469 0.1330 0.8346 0.1248 0.1847 0.1847 0.1469	$\delta J = 10^{-6}$ with one loop of R estimation

4.1.5 Constant System Figures (Q = 0)
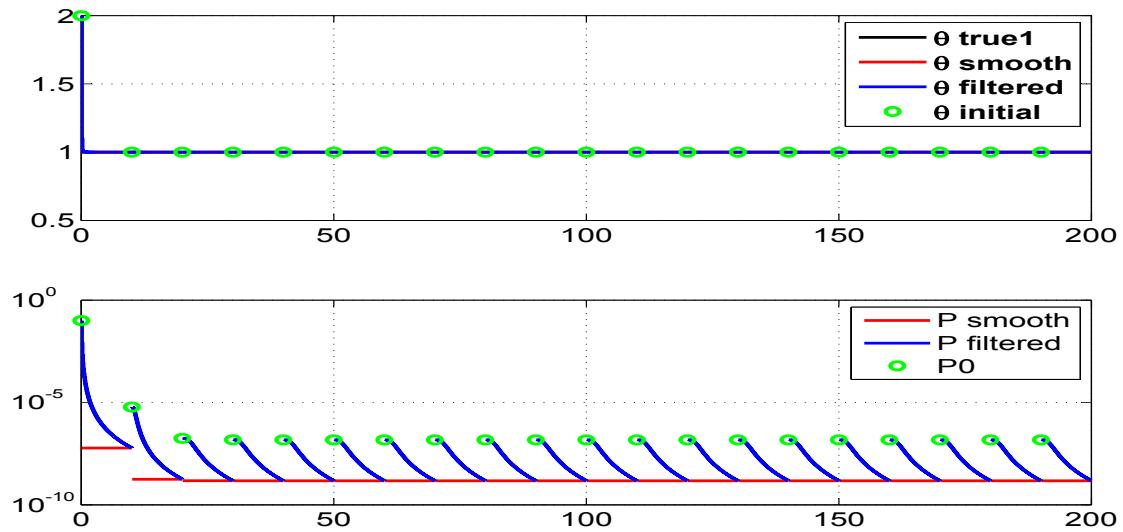
Figure 4.1: The variation of different parameter estimate and their filtered and smoothed covariances through (with the time cumulatively) the iterations
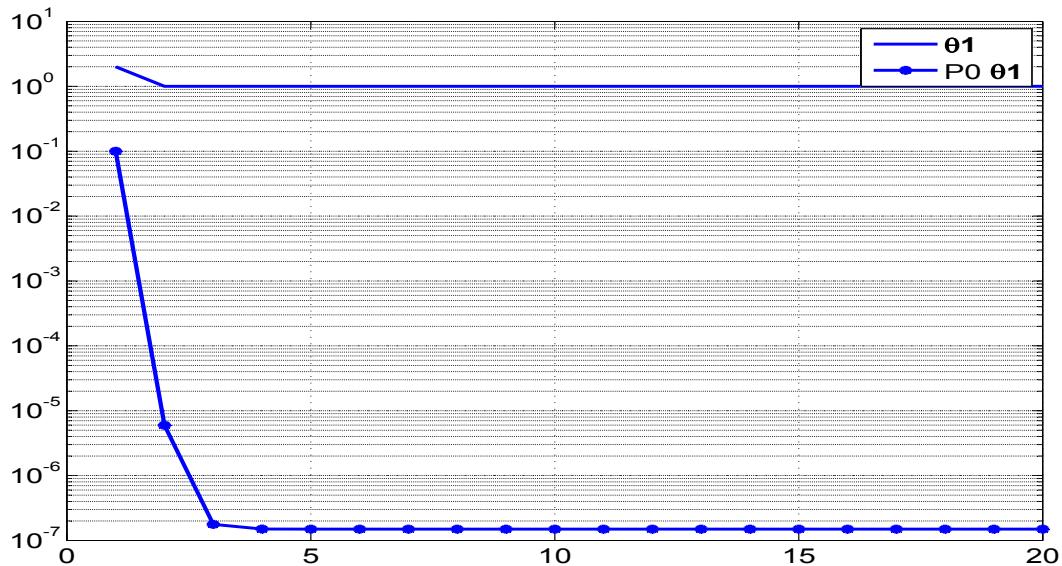
Figure 4.2: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
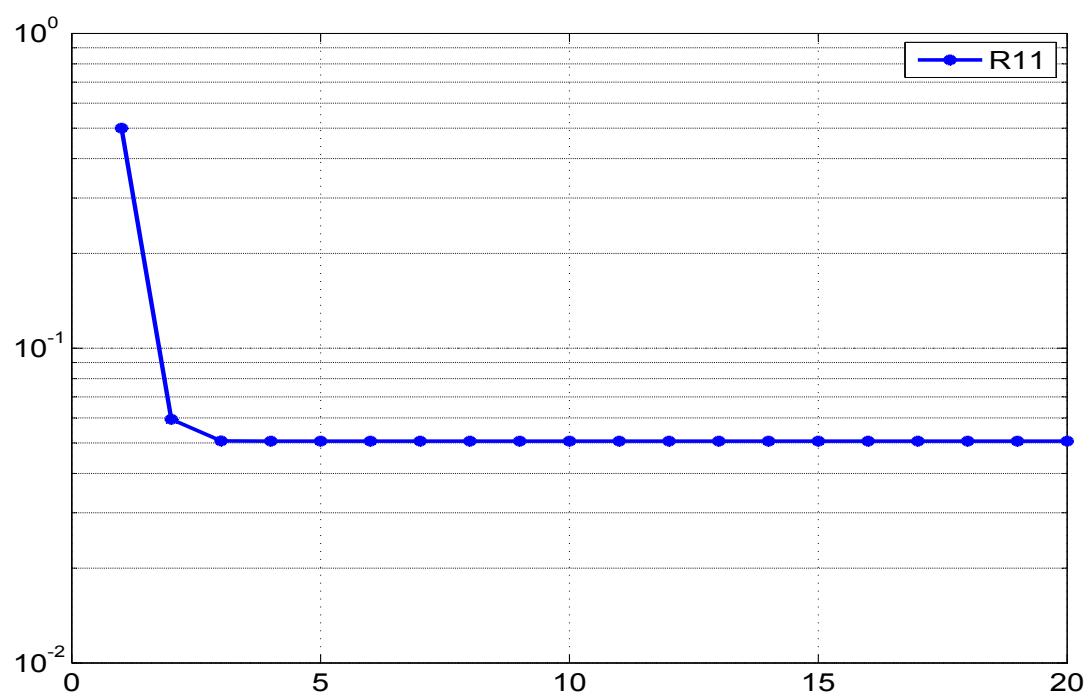
Figure 4.3: Variation of **R** with iterations
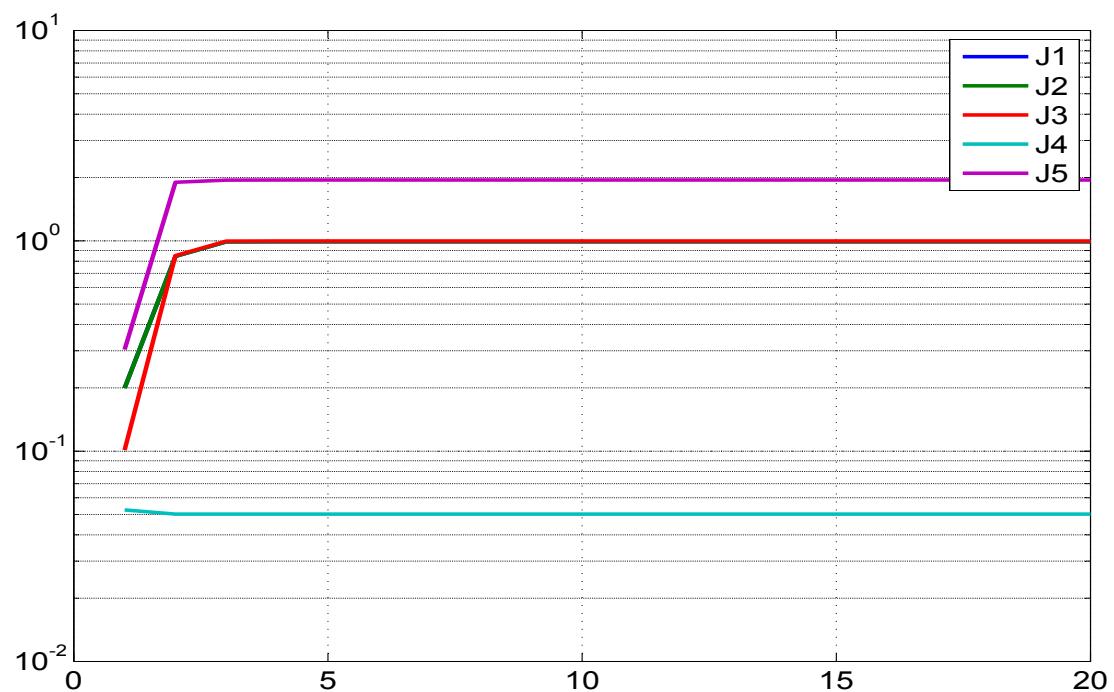
Figure 4.4: Variation of different costs (**J1-J5**) with iterations
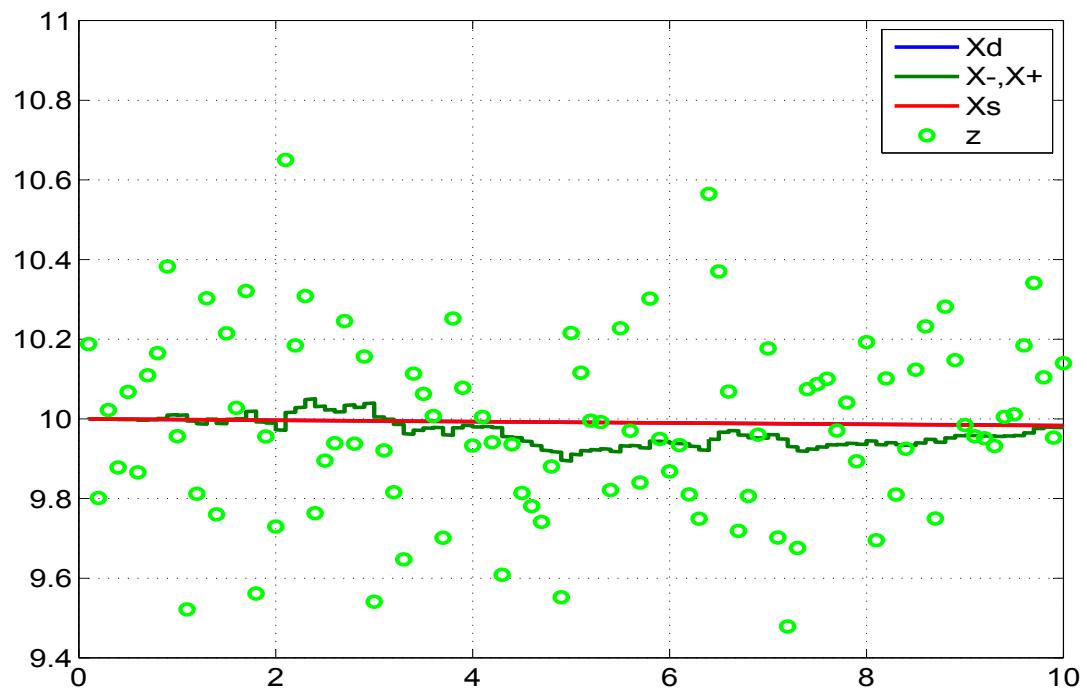
Figure 4.5: Comparison of the predicted dynamics, posterior, smoothed
and the measurement
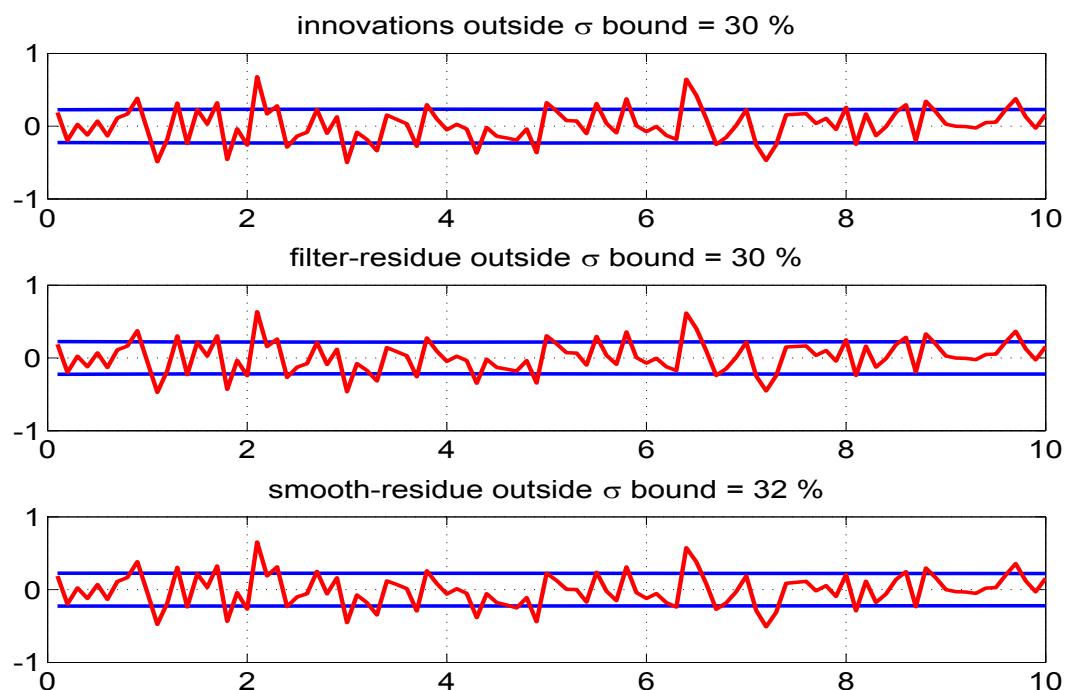
Figure 4.6: The innovations, filtered residue and smoothed residue
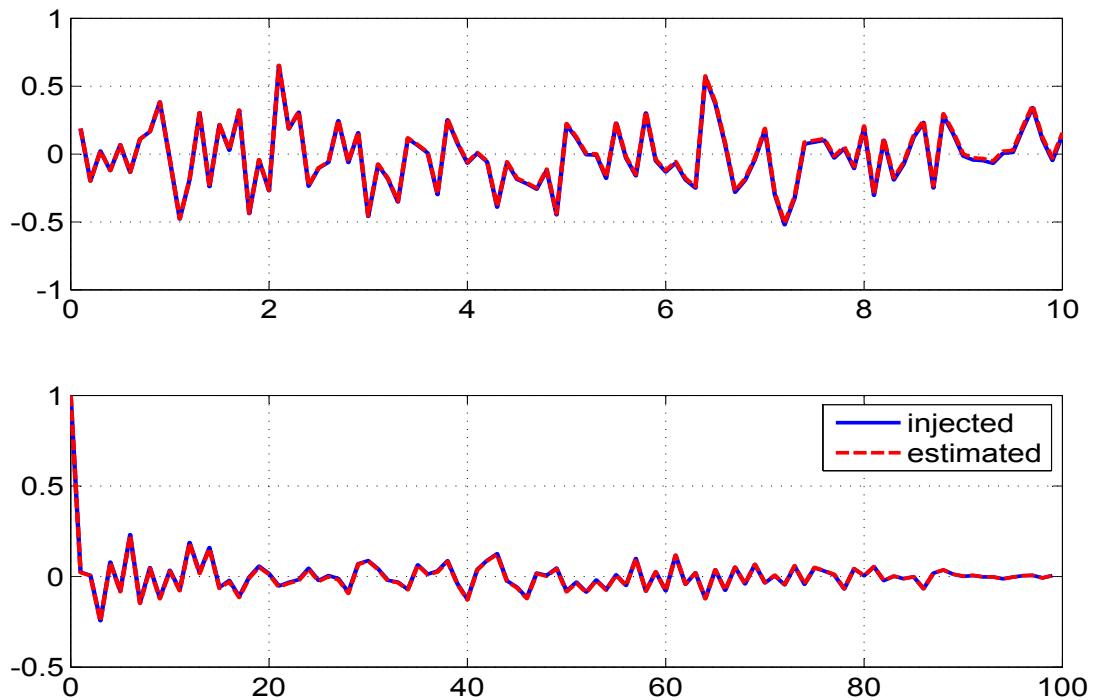
Figure 4.7: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom)
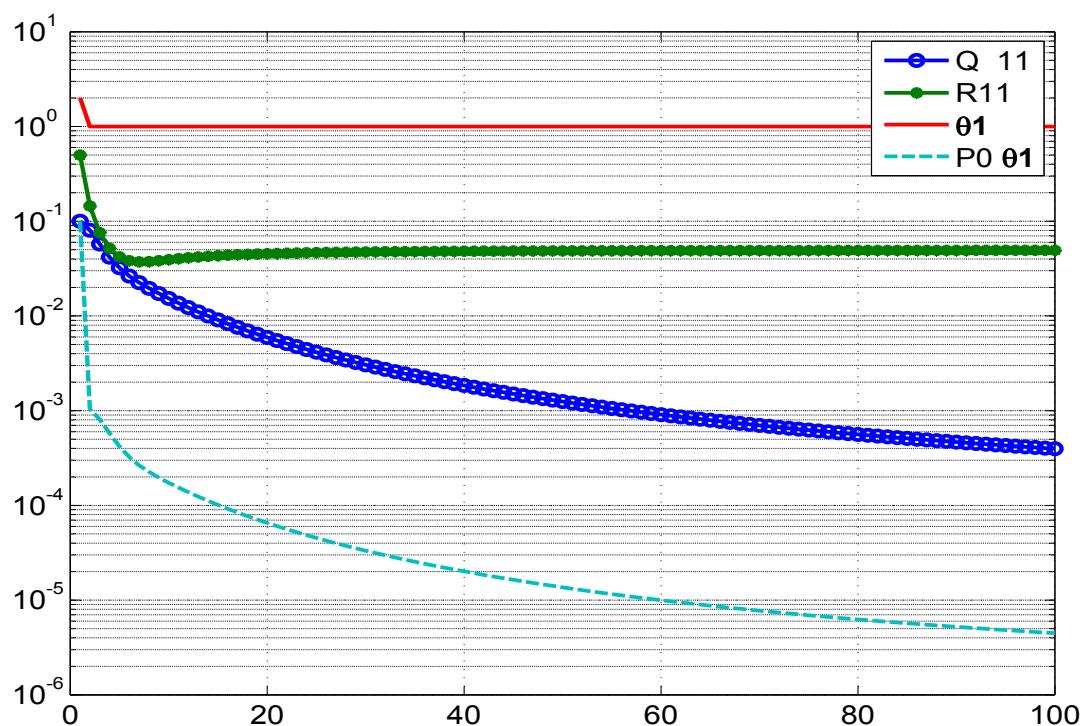
Figure 4.8: Variation of different estimates with iterations using

Reference EKF ($\mathbf{Q} > 0$)

4.1.6 Constant System Figures (Q > 0)
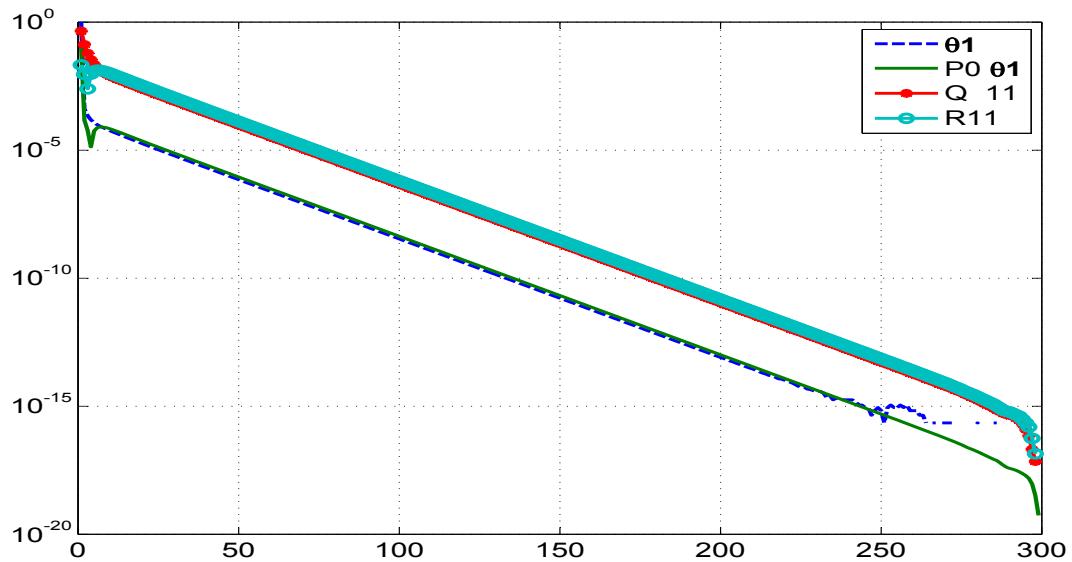
Figure 4.9: The absolute difference between the iterated and final values

with 500 iterations
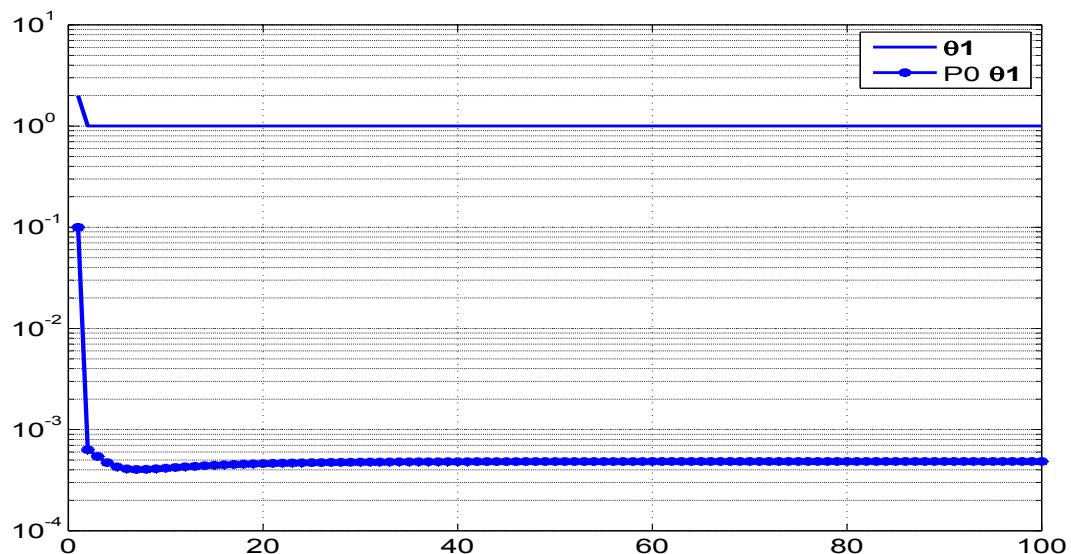
Figure 4.10: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
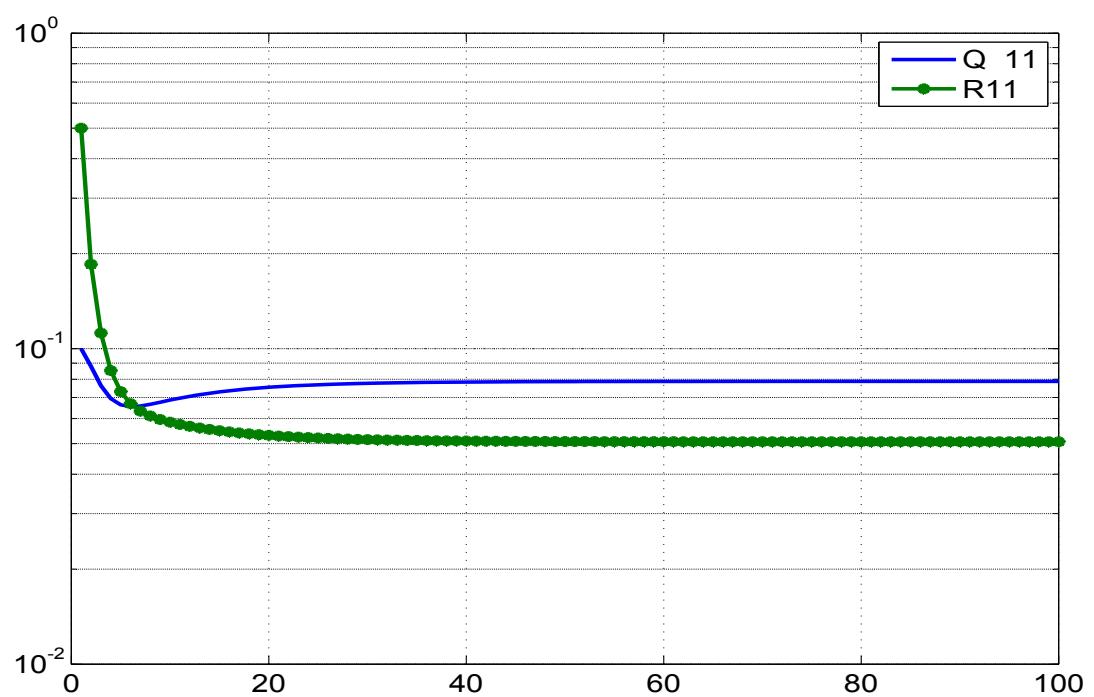
Figure 4.11: Variation of **Q** and **R** with iterations
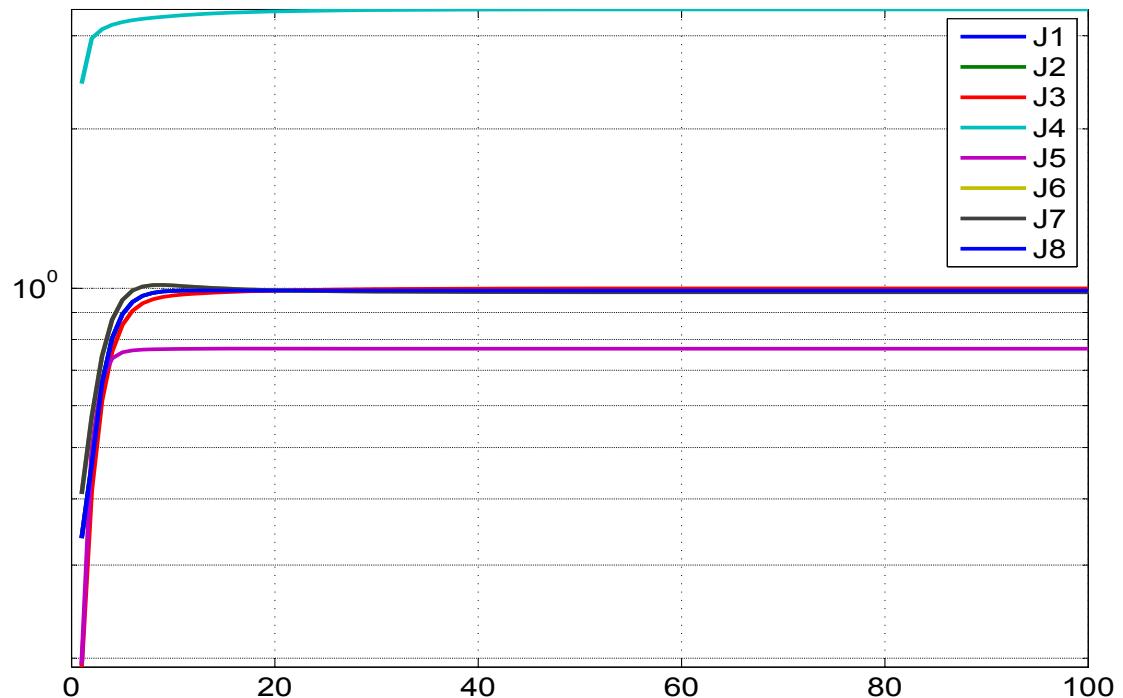
Figure 4.12: Variation of different costs (**J1-J8**) with iterations
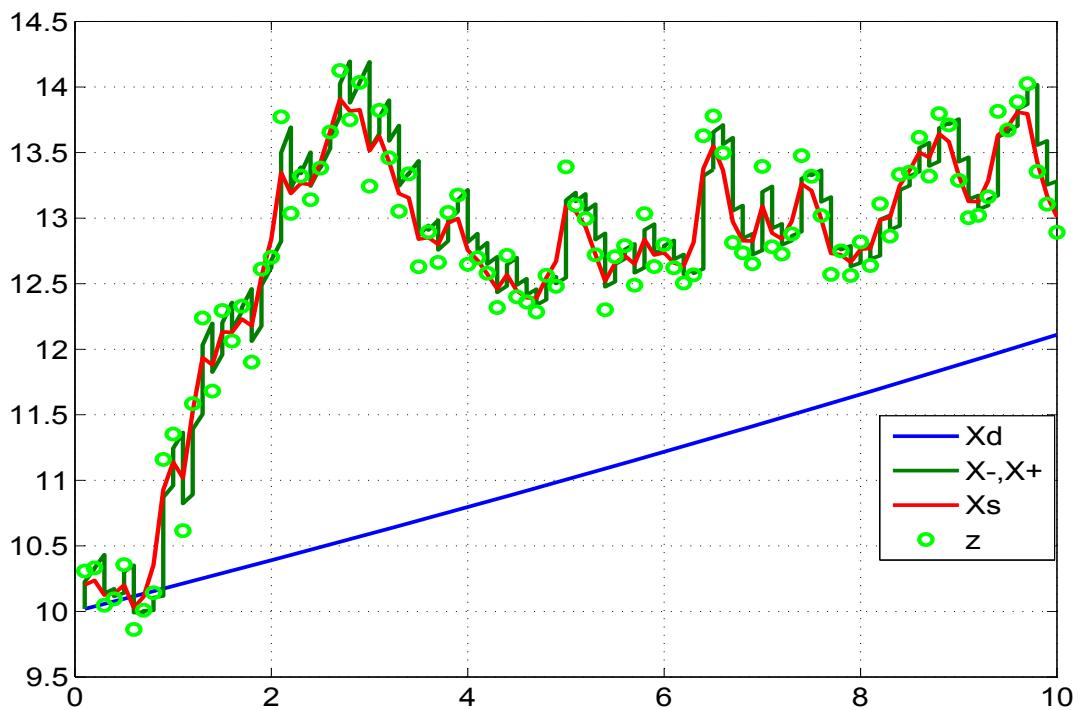
Figure 4.13: Comparison of the predicted dynamics, posterior, smoothed

and the measurement
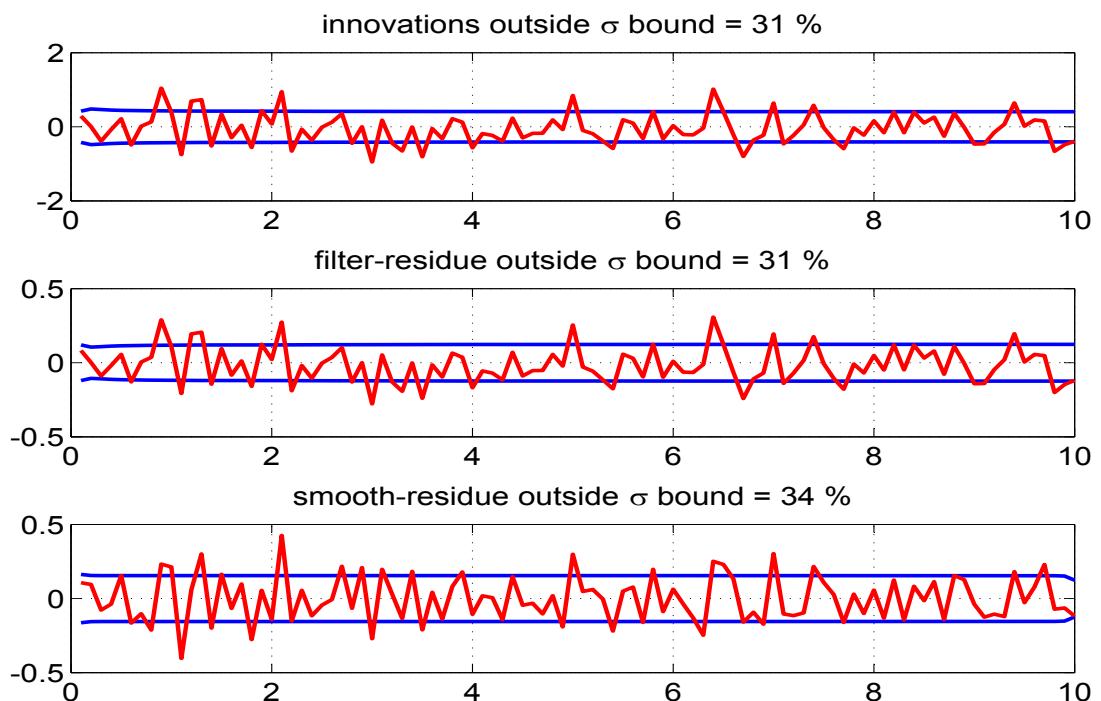
Figure 4.14: The innovations, filtered residue and smoothed residue
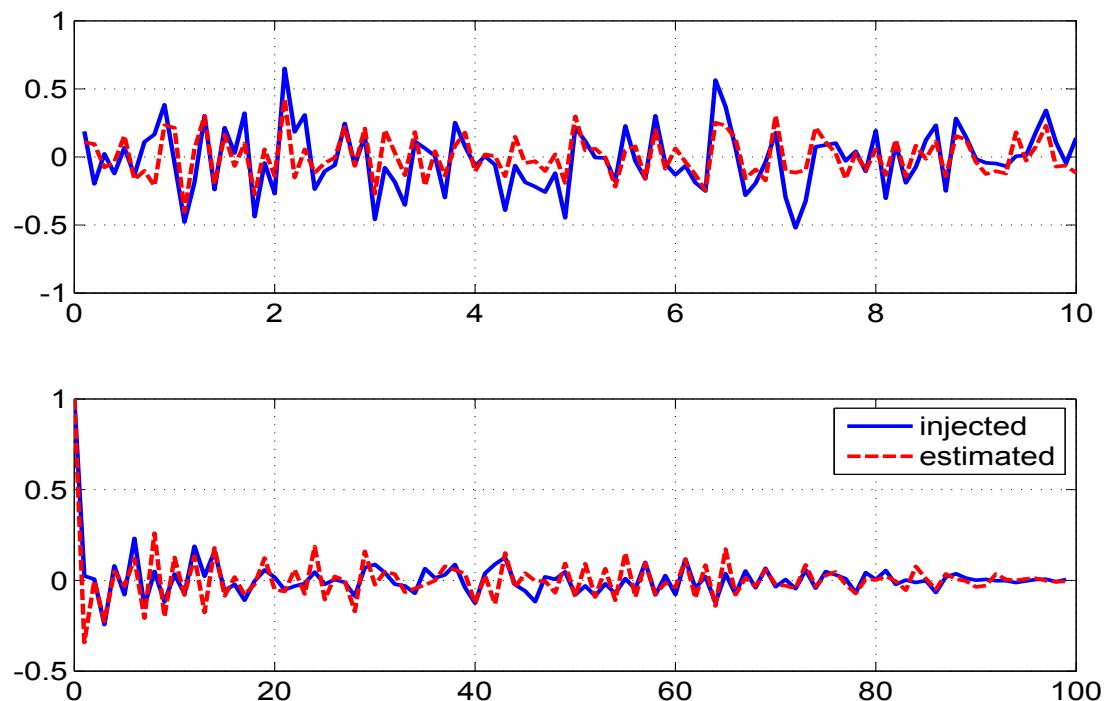
Figure 4.15: Time variation of injected and estimated measurement noise (top) and
their autocorrelation (bottom)
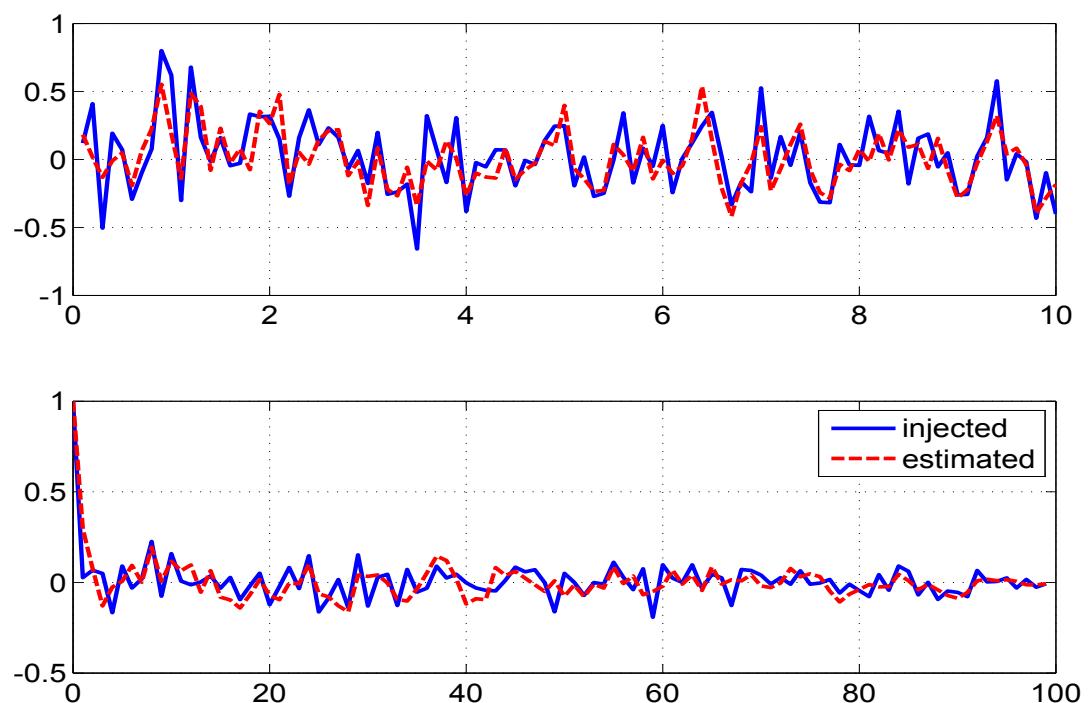
Figure 4.16: Time variation of injected and estimated process noise (top) and
their autocorrelation (bottom)
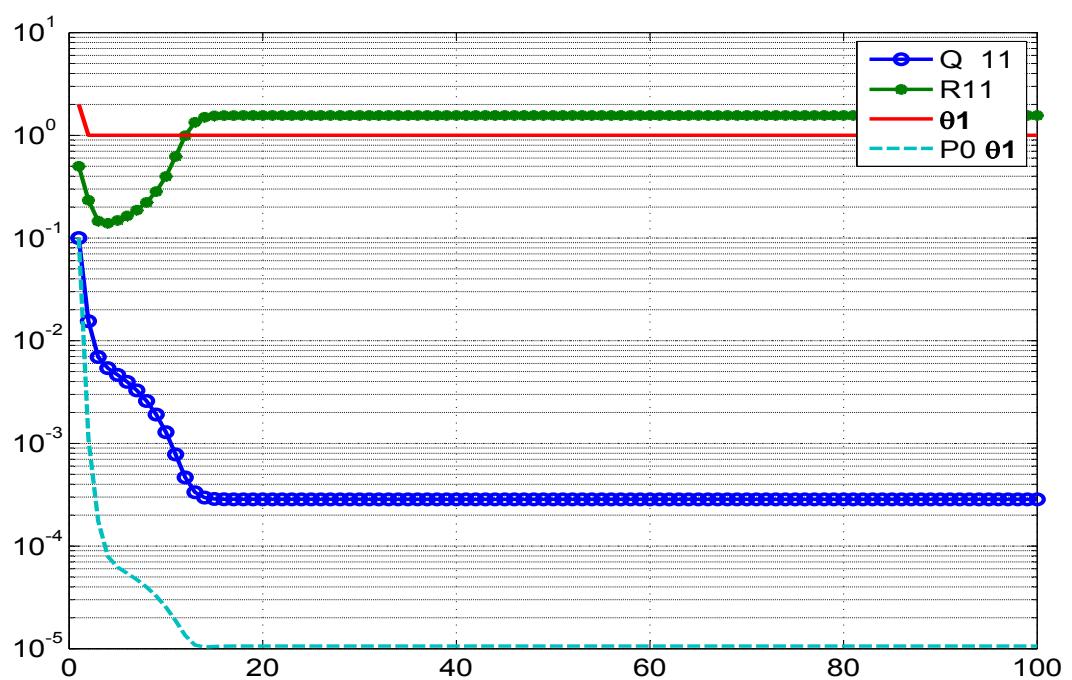
Figure 4.17: Variation of different estimates with iterations using MS method
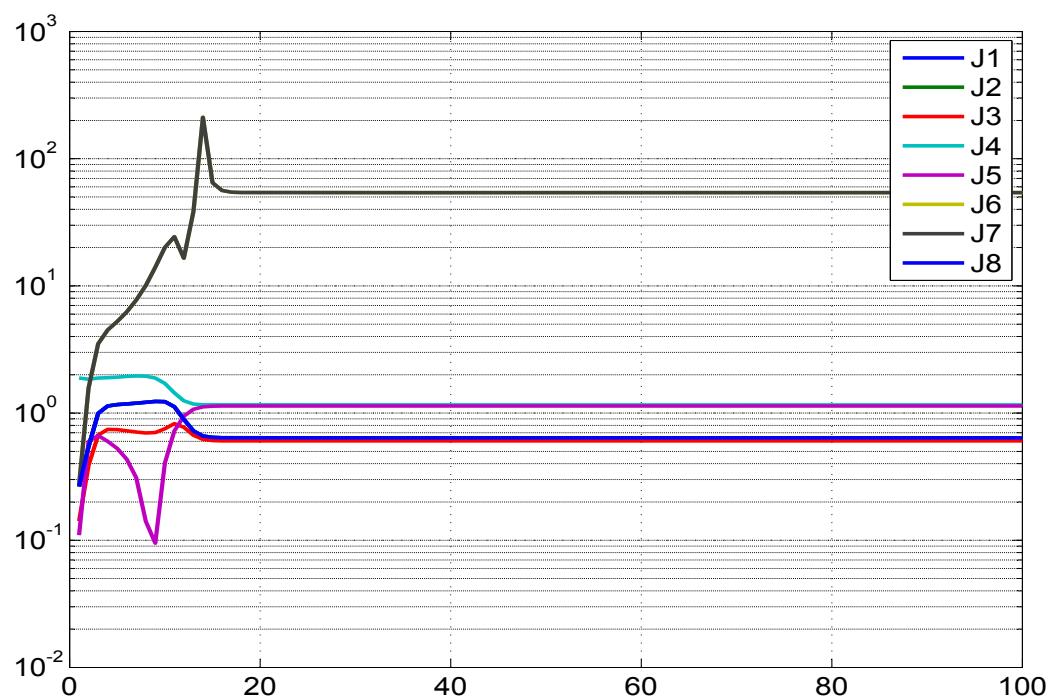
Figure 4.18: Variation of different costs with iterations using MS method
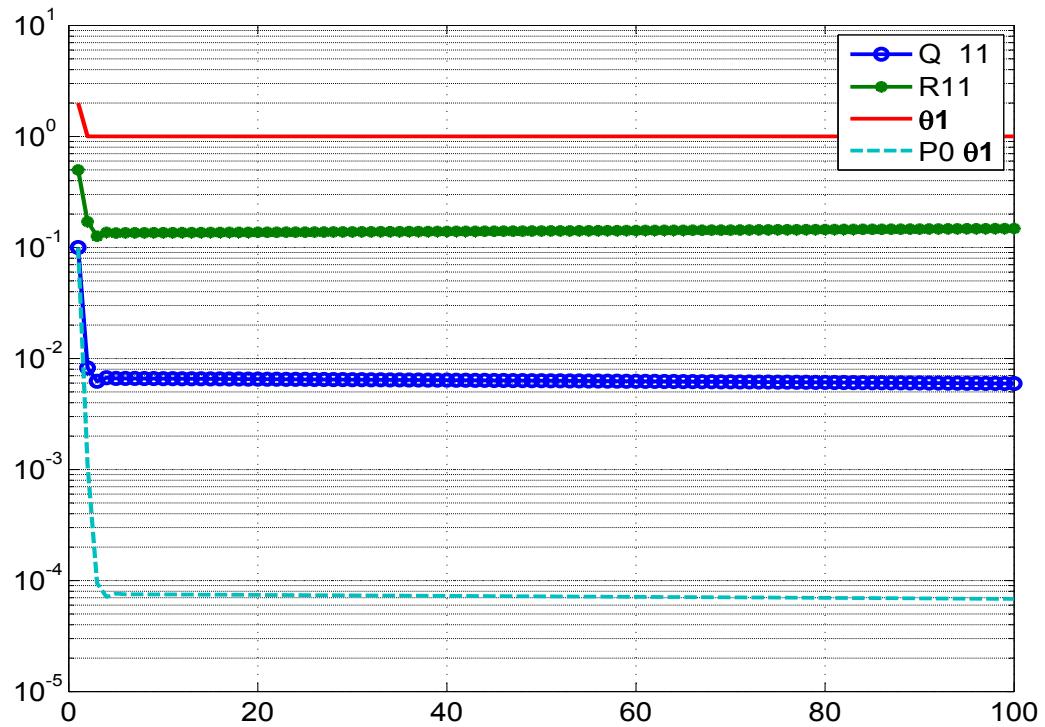
Figure 4.19: Variation of different estimates with iterations using MT method
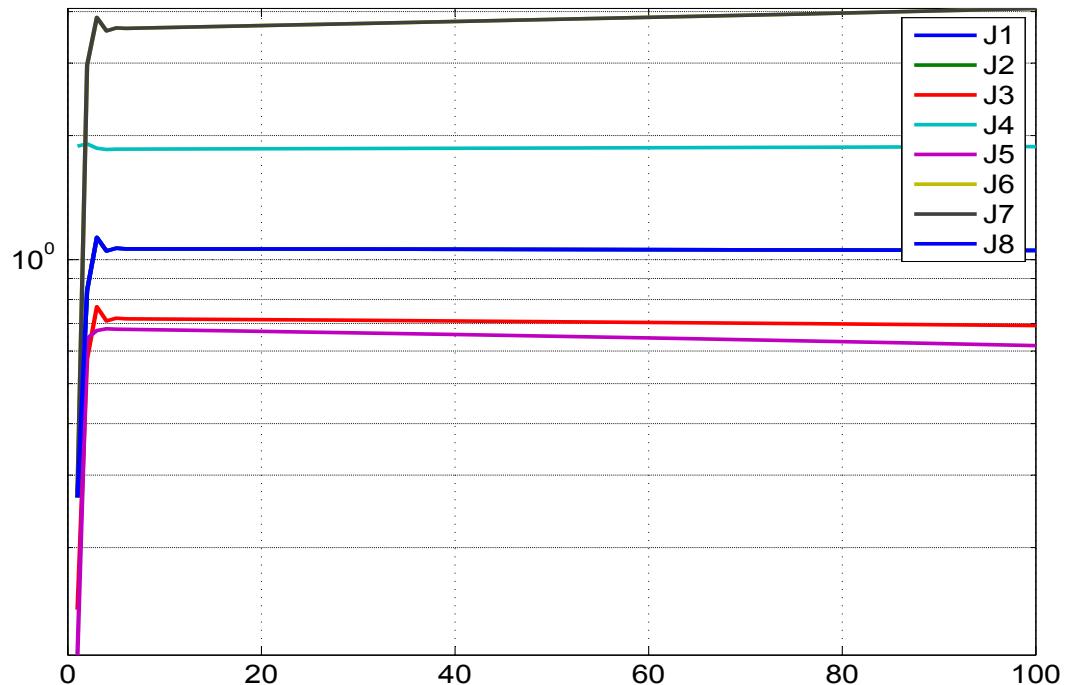
Figure 4.20: Variation of different costs with iterations using MT method

4.2 Simulated Ramp System

Consider a straight line (ramp) defined by, x=at+b where the unknowns are parameter 'a' which is the slope and parameter 'b' which is the intercept. The discretized state and measurement equations of a ramp signal system is given by

$$x_k = x_{k-1} + \Theta \delta t + w_k$$

where $b = x_0 = 10, \delta t$=0.1sec is the sampling time period and $a = \Theta_{true} = 2$ is considered as an unknown parameter. The measurement equation is given by

$$Z_k = HX_k + v_k$$

where the augmented state, $X_k = [x_k, \Theta_k]$, H=[1 0] is the measurement matrix of size $m \times (n + p)$ where $m = n = p = 1$. The numerical values of the noise variances are shown in Table-4.1. All the figures are presented for only one simulation run to prevent cluttering.

4.2.1 Remarks on the Results

We first run the filter assuming **Q** = 0. It was found that about 20 iterations of the data would suffice. The Fig. 4.21-4.27 refer to the **Q** = 0 case. The Fig. 4.21 shows the various parameter estimates and its corresponding variances through cumulative time instants with iterations. The variation of the estimated initial parameters and their variances through iterations are shown in Fig. 4.22. The parameter and the uncertainty reach almost their final estimated values in about 2 and 5 iterations respectively. A similar plot in Fig. 4.23 shows the variation of the estimated measurement noise. The variation of different cost functions (**J1-J5**) through the iterations is shown in Fig. 4.24. The Fig. 4.25 shows the predicted dynamics, filtered and smoothed estimate at the last iteration. The Fig. 4.26 show the innovations, filtered residue and smoothed residue together with the square root of their variance ($\pm\sigma$ bound). In the EKF approach most of the quantities are Gaussian or approximated as quasi Gaussian and one would expect all the above quantities are close to being Gaussian and hence around one third of the total sample points to be outside the σ bound. The injected and estimated measurement noise distributions during the final

iteration shown in Fig. 4.27 indicate that they are close to each other. Further even their autocorrelations are ideally expected to be close to the Kronecker delta function which provides confidence in the proposed filter algorithm.

The next step is to process the data with process noise ($\mathbf{Q} > 0$). The Fig. 4.28-4.35 refer to the $\mathbf{Q} > 0$ case. The Fig. 4.28 shows the absolute difference between the iterated and final values with iterations which indicates the accuracy level that one needs and it was found that 100 iterations are required. The variation of the estimated initial parameters and their variances through iterations are shown in Fig. 4.29. The parameter and the uncertainty reach almost their final estimated values in about 5 and 20 iterations respectively. A similar plot in Fig. 4.30 shows the variation of the estimated measurement and process noise. The variation of different cost functions ($\mathbf{J1}$-$\mathbf{J8}$) through the iterations are shown in Fig. 4.31. The cost functions $\mathbf{J1}$-$\mathbf{J3}$ correspond to the number of measurement (m=1) and in presence of process noise, $\mathbf{J6}$-$\mathbf{J8}$ correspond to the number of states (n=1). The $\mathbf{J4}$ in absence of process noise corresponds to the trace of the measurement noise $\mathbf{R}$. The $\mathbf{J5}$ is the negative log likelihood cost whose absolute value is shown in the plot. There is a mismatch in the predicted dynamics and the measurement as seen in Fig. 4.32 indicating the presence of process noise. The subsequent Fig. 4.33 correspond to the earlier Fig. 4.26 of $\mathbf{Q} = 0$ case. The Fig. 4.34 and Fig. 4.35 shows respectively the injected and estimated measurement and process noise samples across time during the final iteration.

4.2.2 Ramp System Tables (Q = 0)

Table 4.4: Sensitivity Study : ($\mathbf{Q} = 0$) : Ramp system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Consistency ratio-NR	Spread factor EKF	Spread factor NR	$\mathbf{R}$ ratio EKF/True	$\mathbf{R}$ ratio EKF/NR	μ of J1-J5	σ of J1-J5	Remarks
				Reference adaptive EKF used for non zero process noise case							
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: EM-diag				Extremely slow convergence of $\mathbf{Q}$ taking more than 100 iterations.							
				Proposed reference adaptive EKF for zero process noise case, $\mathbf{Q}$ is diagonal matrix fixed to 10^{-10} for all states							
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9996	1.0044	0.9623	0.9581	0.2538	0.2544	0.9761	1.0013	0.9901 0.9901 0.9998 0.0483 -1.9933	0.0000 0.0000 0.0012 0.0065 0.1399	σ_Θ= 0.0036 SIGMA$_{avg}$= 0.0038
				Using different method and options for $\mathbf{P_0}$							
$\mathbf{P_0}$: Scaled up-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9996	0.4993	0.7791	0.9581	0.4676	0.2544	0.9805	1.0058	0.9802 0.9802 0.9996 0.0481 -1.9392	0.0000 0.0000 0.0011 0.0065 0.1418	Higher filter CRB than that of the NR
$\mathbf{P_0}$: Scaled up-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9996	0.5272	0.8226	0.9581	0.4506	0.2544	0.9794	1.0047	0.9813 0.9813 0.9997 0.0481 -1.9529	0.0000 0.0000 0.0010 0.0065 0.1418	Larger spread than that of NR

Table 4.4: Sensitivity Study : ($\mathbf{Q} = 0$) : Ramp system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Consistency ratio-NR	Spread factor EKF	Spread factor NR	$\mathbf{R}$ ratio EKF/True	$\mathbf{R}$ ratio EKF/NR	μ of J1-J5	σ of J1-J5	Remarks
$\mathbf{P_0}$: IIM-diag* $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9996	0.5024	0.7839	0.9581	0.4656	0.2544	0.9803	1.0056	0.9804 0.9804 0.9996 0.0481 -1.9010	0.0000 0.0000 0.0011 0.0065 0.1404	IIM taken only for dynamic states and unity $\mathbf{P_0}$ for parameters
$\mathbf{P_0}$: Smoothed-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9996	4.3583	4.1745	0.9581	0.1599	0.2544	0.9670	0.9919	0.9994 0.9994 0.9999 0.0483 -2.0388	0.0000 0.0000 0.0001 0.0065 0.1399	Lower filter CRB than that of the NR
$\mathbf{P_0}$: Smoothed-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9997	3.9208	6.0621	0.9581	0.2353	0.2544	0.9624	0.9872	0.9986 0.9986 0.9999 0.0481 -2.0438	0.0000 0.0000 0.0001 0.0065 0.1418	Lower filter CRB than that of the NR
$\mathbf{P_0}$: Smoothed-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9996	2.1716	3.3859	0.9581	0.2560	0.2544	0.9622	0.9870	0.9989 0.9989 0.9999 0.0481 -2.0441	0.0000 0.0000 0.0001 0.0065 0.1418	Lower filter CRB than that of the NR

Table 4.4: Sensitivity Study : ($\mathbf{Q} = 0$) : Ramp system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Consistency ratio-NR	Spread factor EKF	Spread factor NR	$\mathbf{R}$ ratio EKF/True	$\mathbf{R}$ ratio EKF/NR	μ of J1-J5	σ of J1-J5	Remarks
					Using different methods for estimating $\mathbf{R}$						
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	0.9996	1.0050	0.9629	0.9581	0.2537	0.2544	0.9749	1.0000	0.9914 0.9914 1.0012 0.0483 -1.9933	0.0026 0.0026 0.0029 0.0065 0.1399	*Using dynamical residue as per Eq-3.18
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	0.9996	1.0093	0.9670	0.9581	0.2530	0.2544	0.9664	0.9913	1.0000 1.0000 1.0098 0.0483 -1.9932	0.0000 0.0000 0.0013 0.0065 0.1399	*Using smoothed residue ignoring 2nd order terms in Eq-3.7
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	0.9996	1.0318	0.9886	0.9581	0.2498	0.2544	0.9244	0.9483	1.0455 1.0455 1.0558 0.0483 -1.9918	0.0022 0.0022 0.0023 0.0065 0.1399	*Using filtered residue ignoring 2nd order terms in Eq-3.19
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	0.9996	0.9823	0.9412	0.9581	0.2575	0.2544	1.0268	1.0533	0.9643 0.9643 0.9736 0.0483 -1.9805	0.1599 0.1599 0.1603 0.0065 0.1399	*Using innovations ignoring 2nd order terms in Eq-3.20
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: MS-diag	0.9996	1.0090	0.9667	0.9581	0.2531	0.2544	0.9672	0.9921	0.9993 0.9993 1.0092 0.0483 -1.9932	0.0021 0.0021 0.0021 0.0065 0.1399	Using full expression in Eq-3.19
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: MT-diag	0.9996	1.0022	0.9602	0.9581	0.2545	0.2544	0.9871	1.0126	1.0075 1.0075 1.0172 0.0483 -1.9784	0.1834 0.1834 0.1839 0.0065 0.1407	Using full expression in Eq-3.20

4.2.3 Ramp System Tables (Q > 0)

Table 4.5: Sensitivity Study : **Q**> 0 : Ramp system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**Q** ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
Proposed reference adaptive EKF for non zero process noise									
P$_0$: Scaled up-[0,0;0,✓] **Q** : EM-[✓,0;0,0] **R** : EM-diag	1.0220	1.0275	0.9981	14.9505	0.9362	1.0279	0.9898 0.9898 1.0003 0.8370 -1.0451 0.9778 0.9778 0.9898	0.0018 0.0018 0.0018 0.5799 0.1263 0.0090 0.0090 0.0018	σ_Θ= 0.2219 SIGMA$_{avg}$= 0.2224
P$_0$: Scaled up-[0,0;0,✓] **Q** : EM-[✓,0;0,0] **R** : EM-diag	1.0221	0.9938	1.0070	14.9282	NA	0.9979	0.9848 0.9848 0.9876 0.8365 -1.0369 0.9783 0.9783 0.9848	0.0590 0.0590 0.1065 0.5726 0.1245 0.0056 0.0056 0.0590	**R** is known

Table 4.5: Sensitivity Study : $\mathbf{Q} > 0$: Ramp system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of J1-J8	σ of J1-J8	Remarks
$\mathbf{P_0}$: Zero $\mathbf{Q}$: EM-diag $\mathbf{R}$: EM-diag	NA	1.0309	NA	NA	0.9355	1.0280	1.0000 1.0000 1.0006 2.5377 -1.0804 1.0001 1.0001 1.0000	0.0001 0.0001 0.0051 3.5893 0.1198 0.0067 0.0067 0.0001	The parameters are assumed to be known
$\mathbf{P_0}$: Zero $\mathbf{Q}$: EM-diag $\mathbf{R}$: Known	NA	0.9963	NA	NA	0.9670	0.9989	0.9945 0.9945 0.9869 2.5377 -1.0718 1.0004 1.0004 0.9945	0.0618 0.0618 0.1119 3.5893 0.1180 0.0051 0.0051 0.0618	The parameters and $\mathbf{R}$ are assumed to be known

Table 4.5: Sensitivity Study : $\mathbf{Q} > 0$: Ramp system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of $\mathbf{J1\text{-}J8}$	σ of $\mathbf{J1\text{-}J8}$	Remarks
Extended EM algorithm (Bavdekar et al.[104] (2011))									
$\mathbf{P_0}$: Smoothed-full $\mathbf{Q}$: EM-full $\mathbf{R}$: EM-full	There is no consistency in the parameter estimates since the smoothed $\mathbf{P_0}$ simply decreases through iterations.								
Modification-1									
$\mathbf{P_0}$: Smoothed-diag $\mathbf{Q}$: EM-diag $\mathbf{R}$: EM-diag	1.0273	1.0314	10.3040	8.7111	0.9753	0.9389	0.9993 0.9993 1.0004 0.8505 -1.1022 1.0041 1.0041 0.9993	0.0017 0.0017 0.0017 0.5852 0.1277 0.0030 0.0030 0.0017	No stoping condition. The low spread factor is deceiving since there is no consistency in the parameter estimates with its covariance
Modification-2									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-diag $\mathbf{R}$: EM-diag	1.0220	1.0275	0.9981	14.9505	0.9362	1.0279	0.9898 0.9898 1.0003 0.8370 -1.0451 0.9778 0.9778 0.9898	0.0018 0.0018 0.0018 0.5799 0.1263 0.0090 0.0090 0.0018	The slight differences from the reference case is due to the small $\mathbf{Q}$ that is generated for the parameters
Modification-3									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: DSDT-diag $\mathbf{R}$: EM-diag	1.0220	1.0275	0.9981	14.9505	0.9362	1.0279	1.9634 1.9691 1.9982 0.0710 -8.7877 1.9334 1.9406 1.9565	0.0256 0.0247 0.0090 0.0402 0.2274 0.0374 0.0394 0.0411	Using DSDT statistics for $\mathbf{Q}$ estimation given by Eq-3.17

Table 4.5: Sensitivity Study : $\mathbf{Q} > 0$: Ramp system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
				MS method (Mohamed and Schwarz)					
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MS-diag $\mathbf{R}$: MS-diag				$\mathbf{R}$ can converge to a very large value and the cost functions diverge as seen in Fig. 4.36,4.37. There is no convergence in cost even if Θ is known.					
			MS method with $\mathbf{Q}$ being estimated using only last N/2 samples						
$\mathbf{P_0}$: Zero $\mathbf{Q}$: MS-diag $\mathbf{R}$: Known	NA	1.0100	NA	NA	NA	0.9927	1.0354 1.0354 1.0071 2.5377 -1.0504 1.0983 1.0983 1.0354	0.1629 0.1629 0.1304 3.5893 0.1170 0.2726 0.2726 0.1629	$\mathbf{R}$, Θ is known
			MS method assuming $\mathbf{R}$ is known and $\mathbf{Q}$ is estimated using only last N/2 samples						
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MS-[✓,0;0,0] $\mathbf{R}$: Known	1.0221	0.9804	0.9599	15.4046	NA	1.0968	0.9447 0.9447 0.9543 0.8383 -1.0321 0.9376 0.9376 0.9447	0.0101 0.0101 0.0590 0.5718 0.1237 0.0807 0.0807 0.0101	Stable operation when $\mathbf{R}$ is known
			Gemson's approach using MS method						
$\mathbf{P_0}$: IIM-full $\mathbf{Q}$: MS-[✓,0;0,0] $\mathbf{R}$: 0.9*True $\mathbf{R}$	1.0229	1.0183	0.9589	15.4820	0.8703	1.1794	0.9864 0.9864 1.0089 0.8438 -1.0348 0.9741 0.9741 0.9864	0.1453 0.1453 0.1295 0.5768 0.1230 0.1846 0.1846 0.1453	$\delta J = 10^{-6}$ with one loop of $\mathbf{R}$ estimation

Table 4.5: Sensitivity Study : $\mathbf{Q} > 0$: Ramp system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	R ratio EKF/True	Q ratio EKF/True	μ of J1-J8	σ of J1-J8	Remarks
MT method (Myers and Tapley)									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MT-diag $\mathbf{R}$: MT-diag	Positive definiteness of $\mathbf{Q}$, $\mathbf{R}$ is not gauranteed and the cost functions can diverge as seen in Fig. 4.38,-4.39. There is no proper convergence even if Θ is known.								
MT method with $\mathbf{Q}$ being estimated using only last N/2 samples									
$\mathbf{P_0}$: Zero $\mathbf{Q}$: MT-diag $\mathbf{R}$: Known	NA	1.0087	NA	NA	NA	0.9933	1.0310 1.0310 1.0068 2.5377 -1.0527 1.0832 1.0832 1.0310	0.1634 0.1634 0.1383 3.5893 0.1182 0.2476 0.2476 0.1634	$\mathbf{R}$, Θ is known
MT method assuming $\mathbf{R}$ is known and $\mathbf{Q}$ is estimated using only last N/2 samples									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MT-[✓,0;0,0] $\mathbf{R}$: Known	1.0222	1.0049	1.0261	14.8499	0.9362	0.9900	1.0171 1.0171 1.0058 0.8358 -1.0209 1.0478 1.0478 1.0171	0.1507 0.1507 0.1296 0.5691 0.1255 0.2155 0.2155 0.1507	Stable Operation
Gemson's approach using MT method									
$\mathbf{P_0}$: IIM-full $\mathbf{Q}$: MT-[✓,0;0,0] $\mathbf{R}$: 0.9*True $\mathbf{R}$	1.0230	1.0252	0.9832	15.2329	0.8703	1.1168	1.0107 1.0107 1.0287 0.8423 -1.0359 1.0021 1.0021 1.0107	0.1462 0.1462 0.1326 0.5757 0.1237 0.1836 0.1836 0.1462	$\delta J = 10^{-6}$ with one loop of $\mathbf{R}$ estimation

4.2.4 Ramp System Figures (Q = 0)
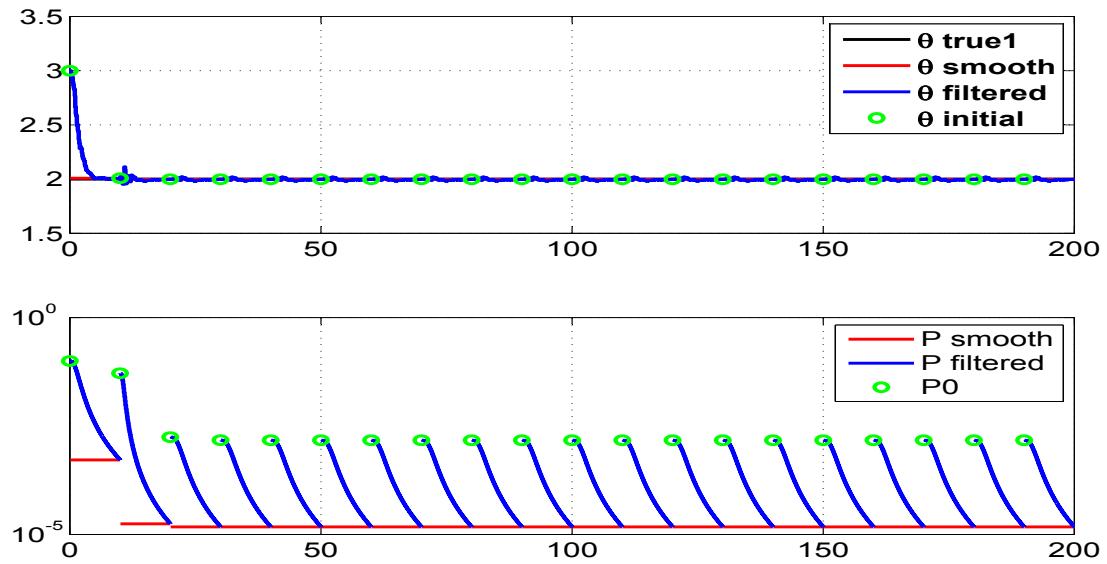
Figure 4.21: The variation of different parameter estimate and their filtered and smoothed covariances through (with the time cumulatively) the iterations
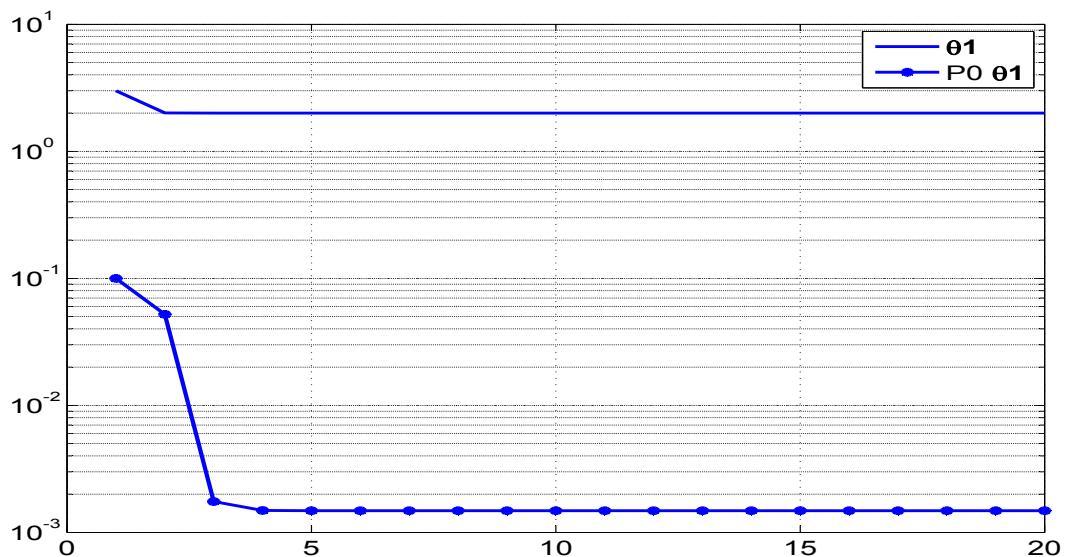
Figure 4.22: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
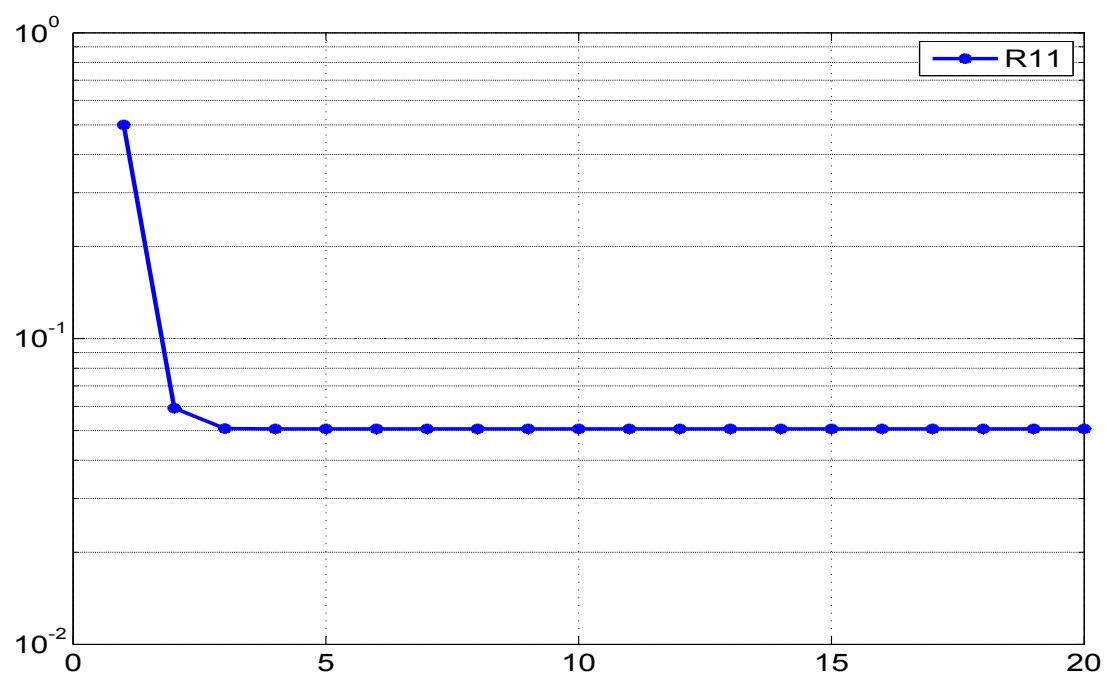
Figure 4.23: Variation of **R** with iterations
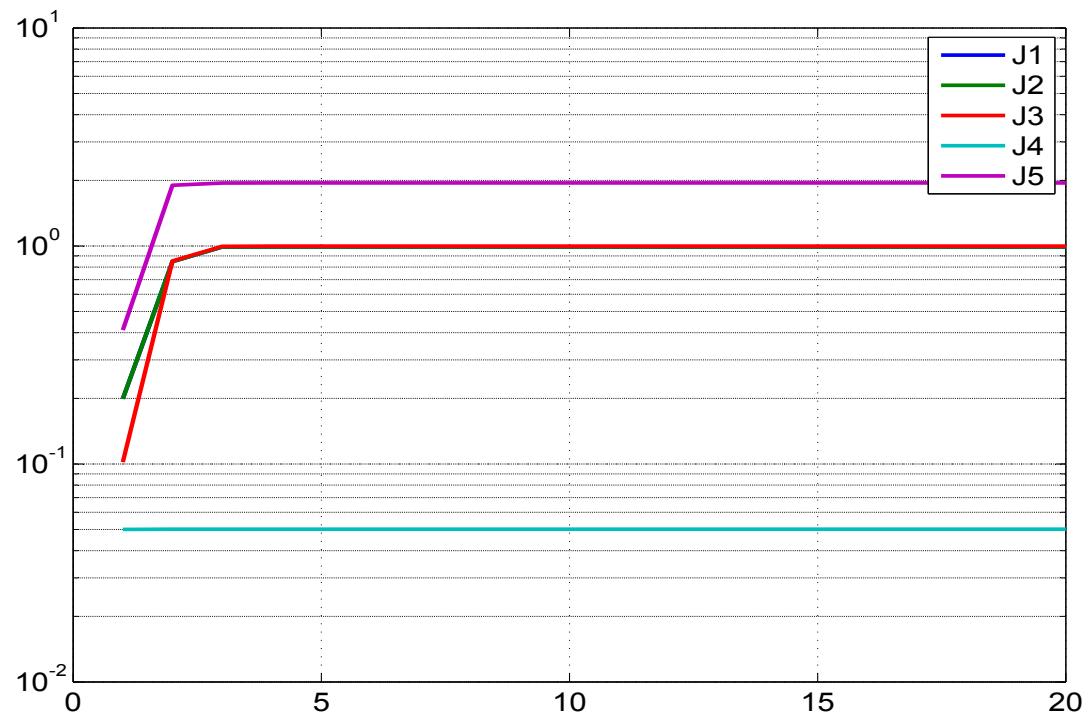
Figure 4.24: Variation of different costs (**J1-J5**) with iterations
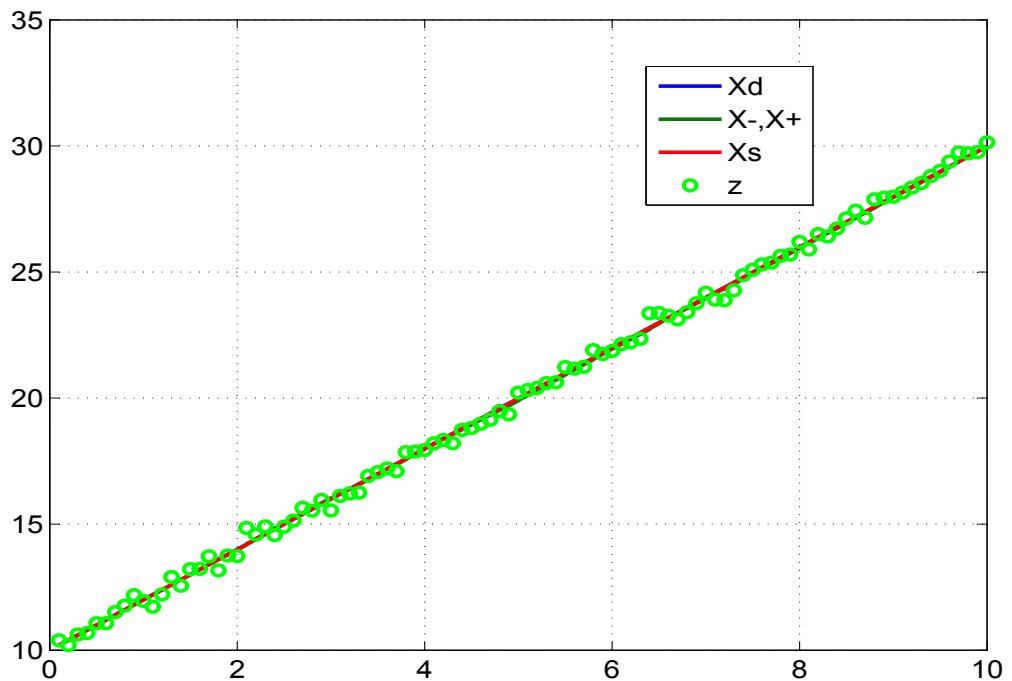
Figure 4.25: Comparison of the predicted dynamics, posterior, smoothed

and the measurement
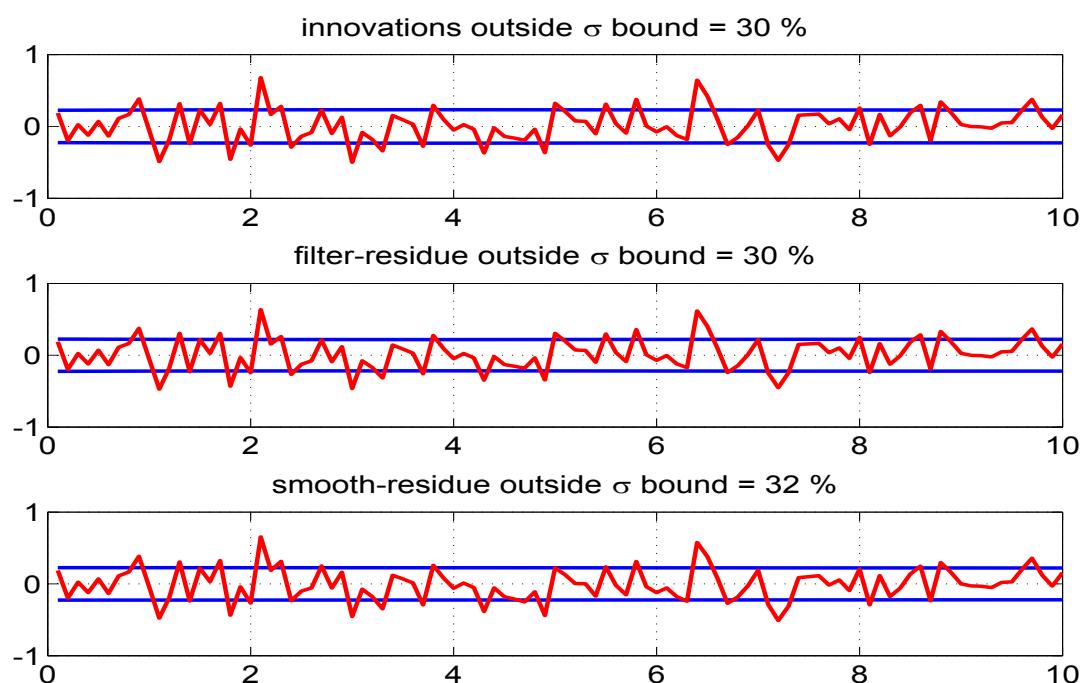
Figure 4.26: The innovations, filtered residue and smoothed residue
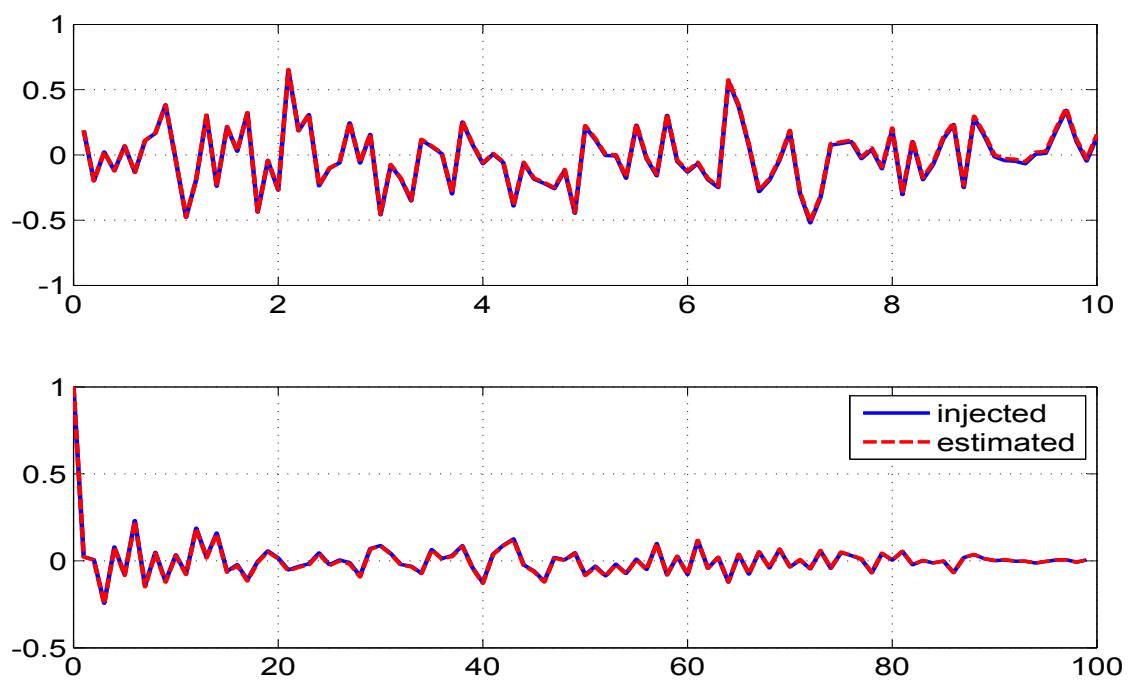
Figure 4.27: Time variation of injected and estimated measurement noise (top) and their autocorrelation (bottom)

4.2.5   Ramp System Figures (Q > 0)
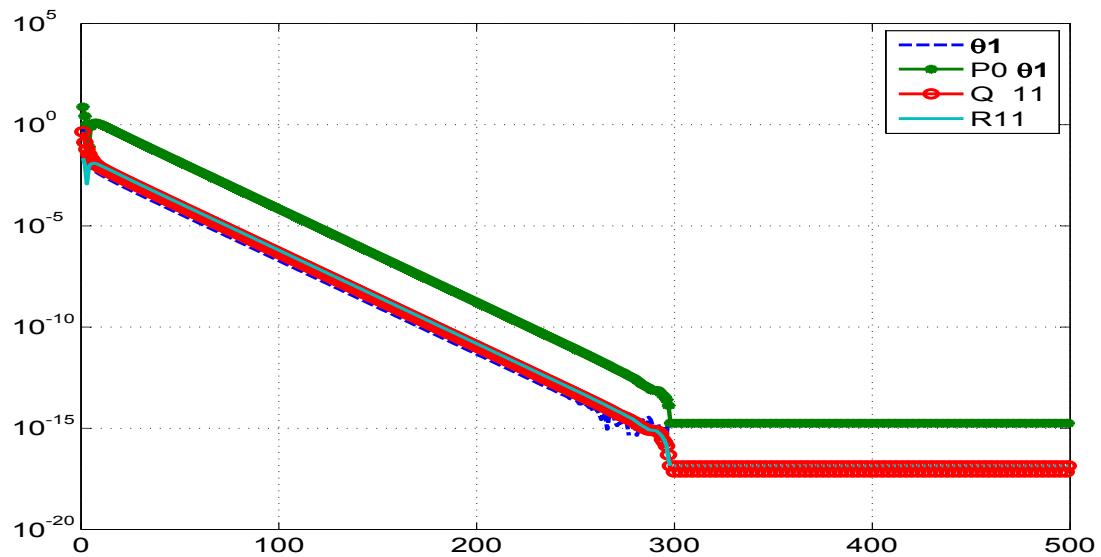
Figure 4.28: The absolute difference between the iterated and final values

with 500 iterations
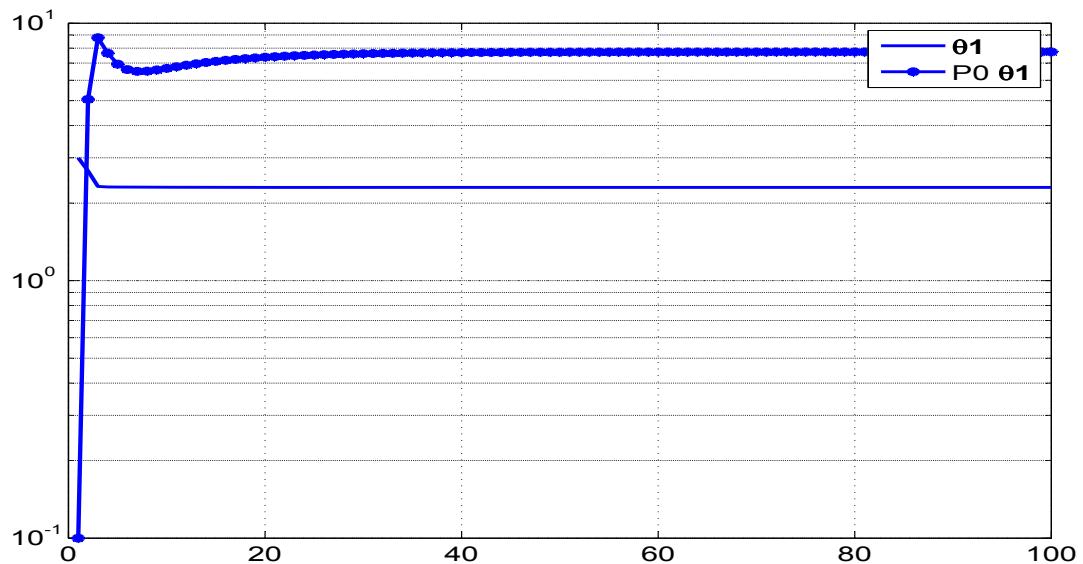
Figure 4.29: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
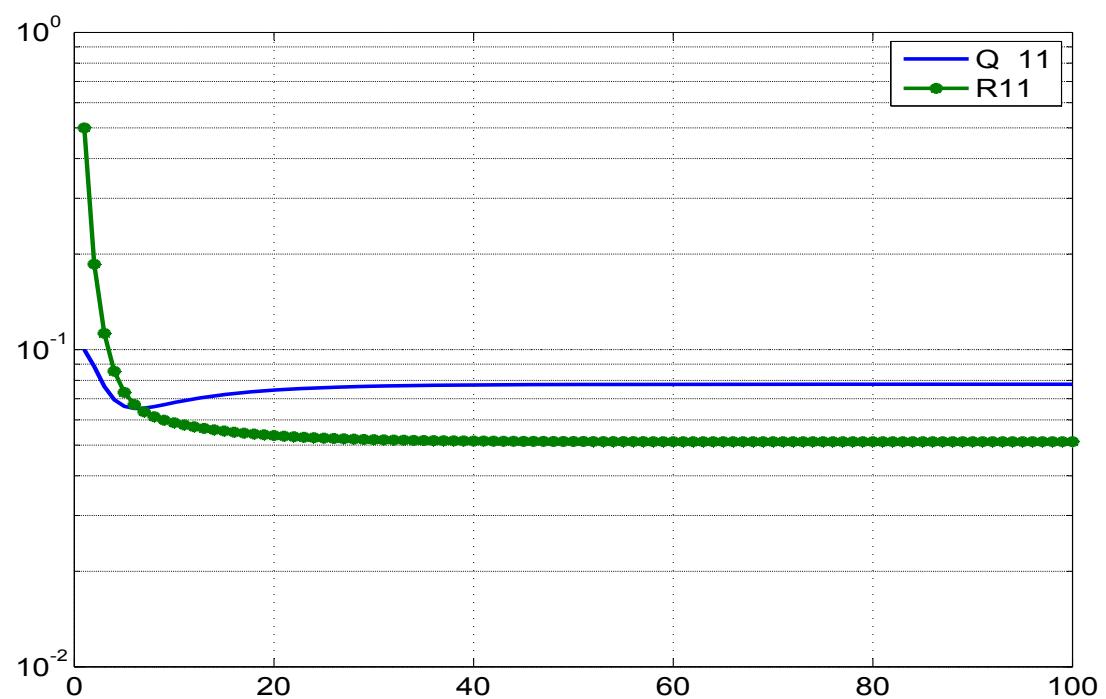
Figure 4.30: Variation of **Q** and **R** with iterations
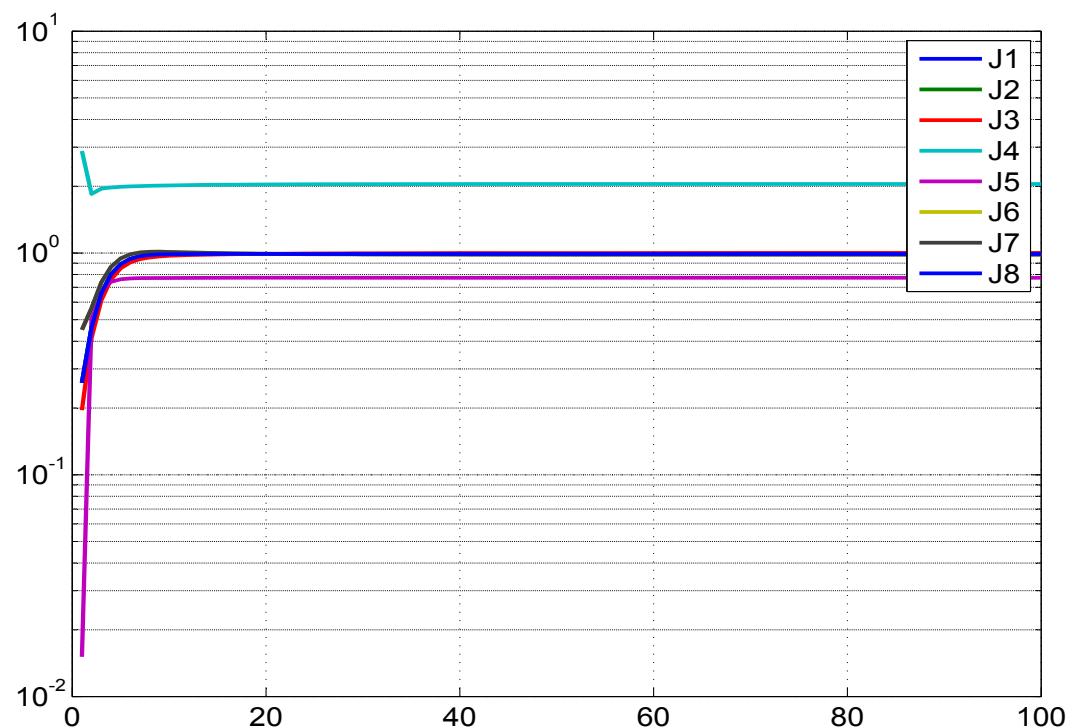
Figure 4.31: Variation of different costs (**J1-J8**) with iterations

103
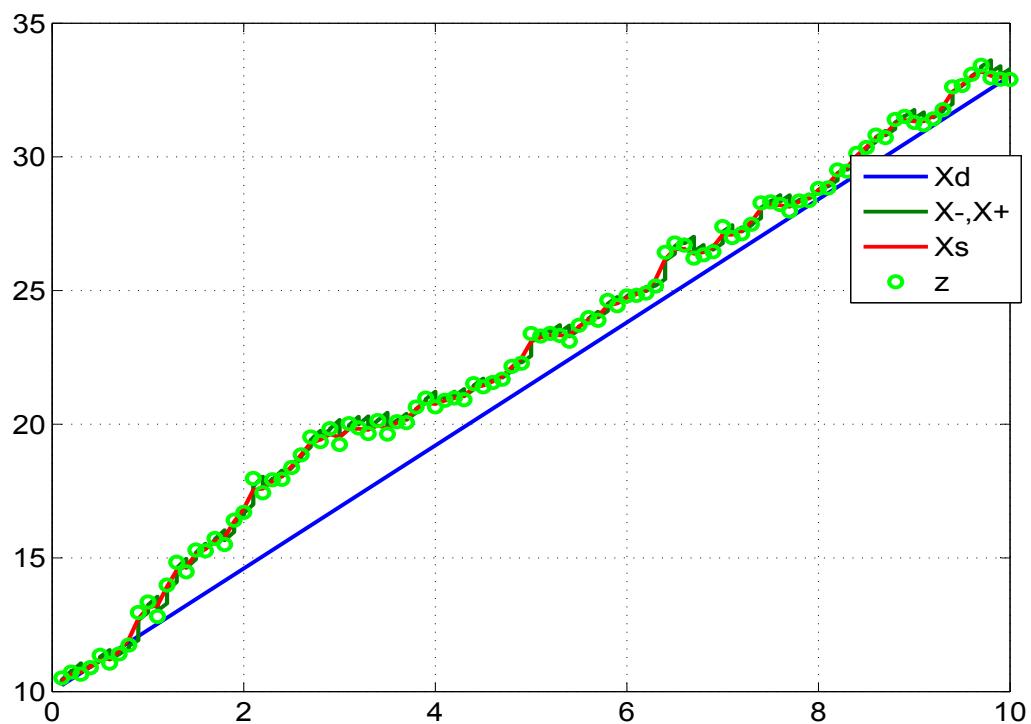
Figure 4.32: Comparison of the predicted dynamics, posterior, smoothed
and the measurement
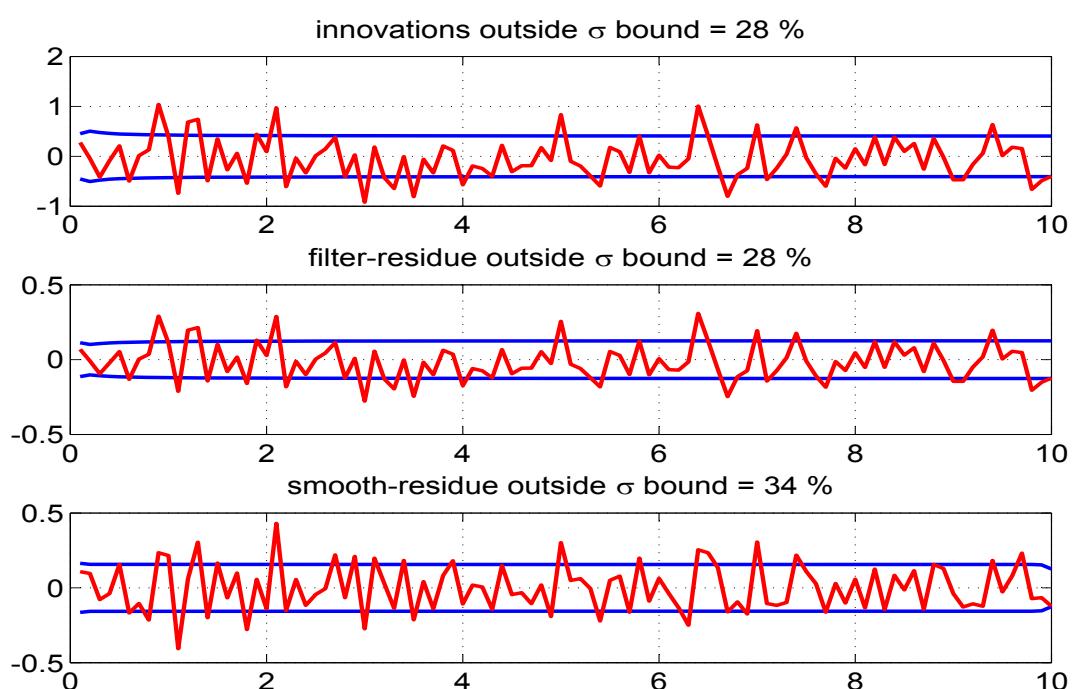
Figure 4.33: The innovations, filtered residue and smoothed residue
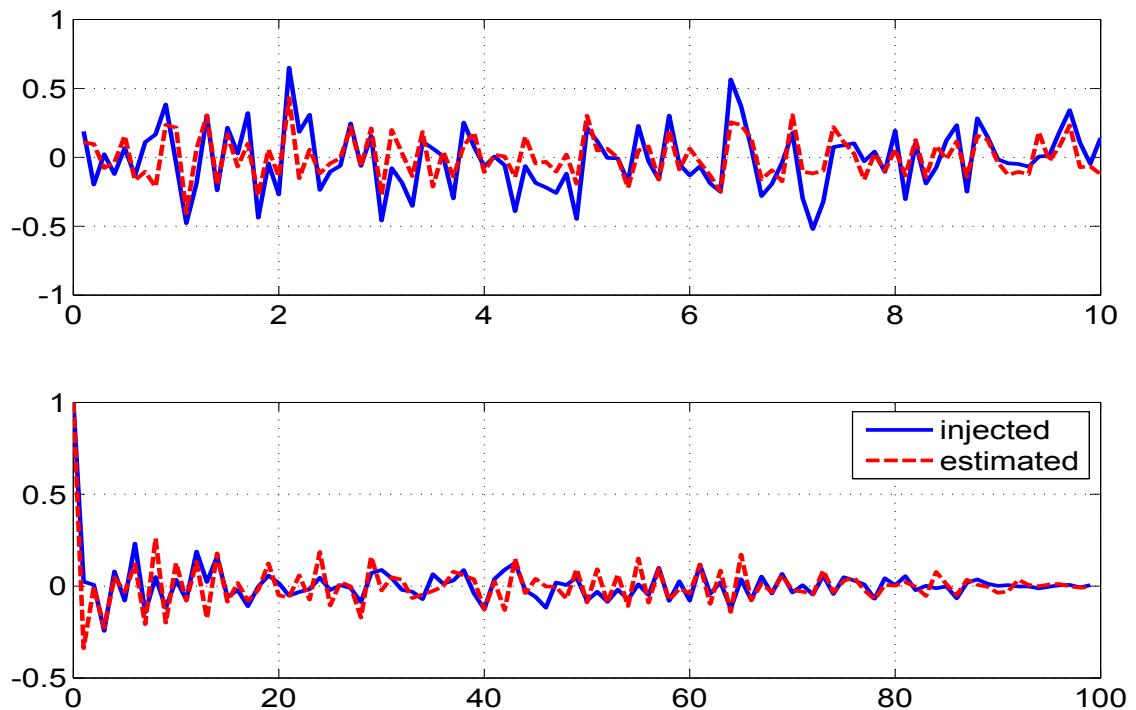
Figure 4.34: Time variation of injected and estimated measurement noise (top) and their autocorrelation (bottom)
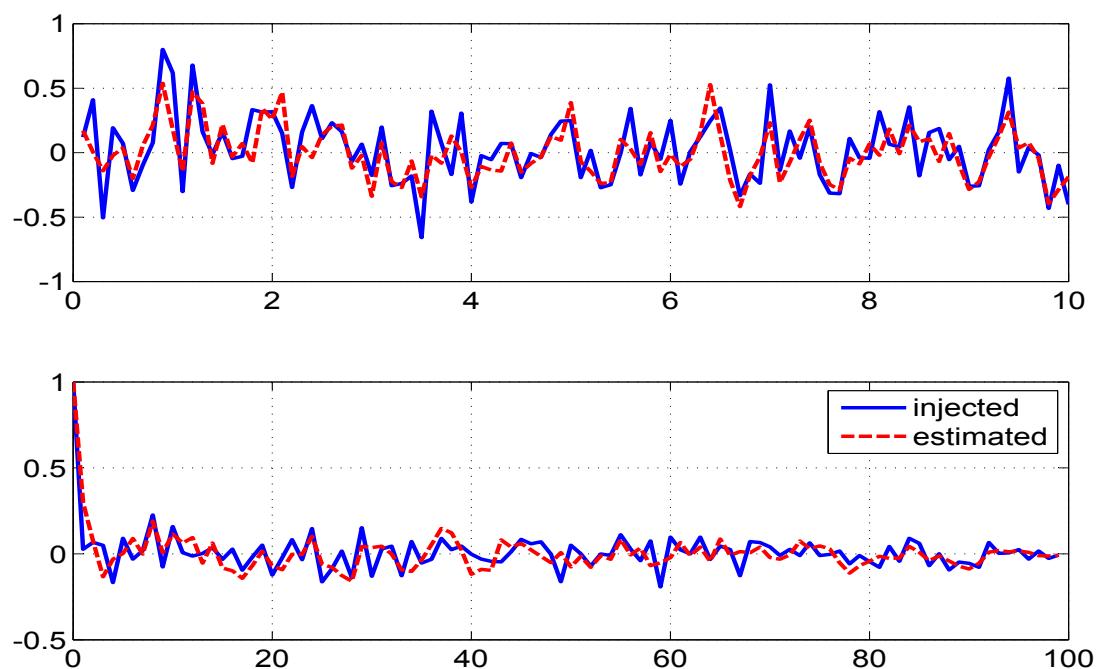
Figure 4.35: Time variation of injected and estimated process noise (top) and their autocorrelation (bottom)
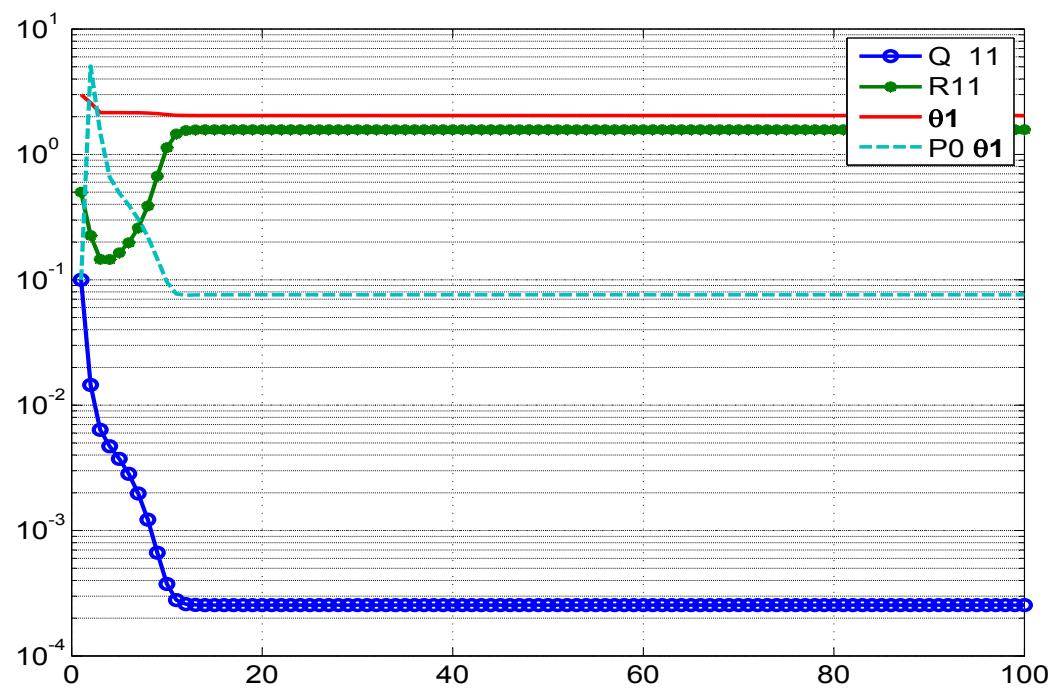
Figure 4.36: Variation of different estimates with iterations using MS method
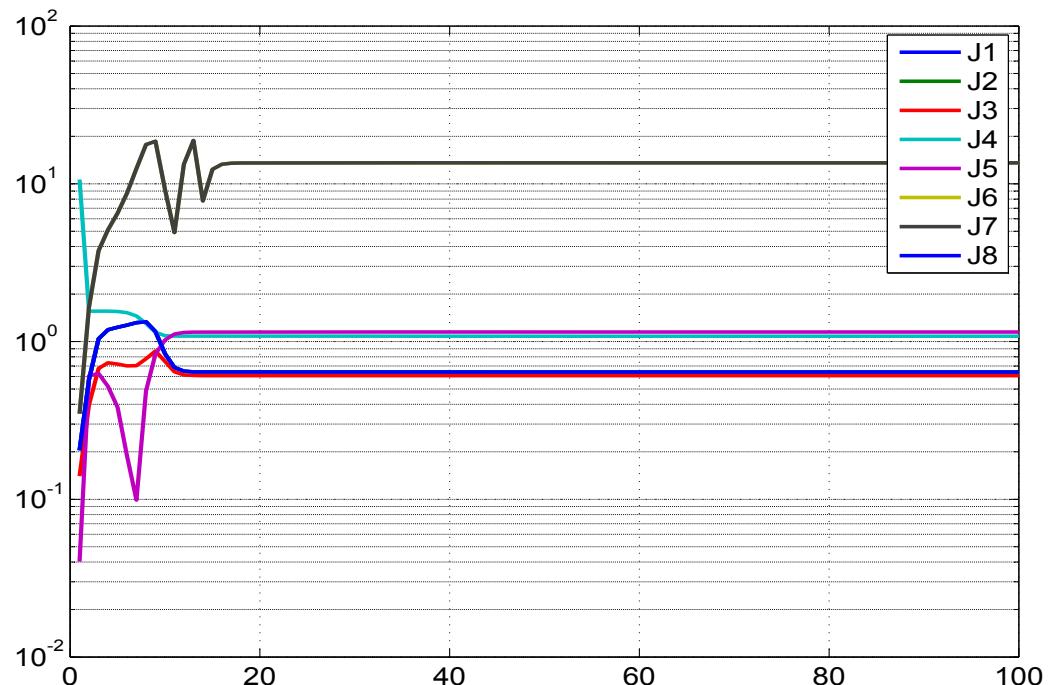
Figure 4.37: Variation of different costs with iterations using MS method
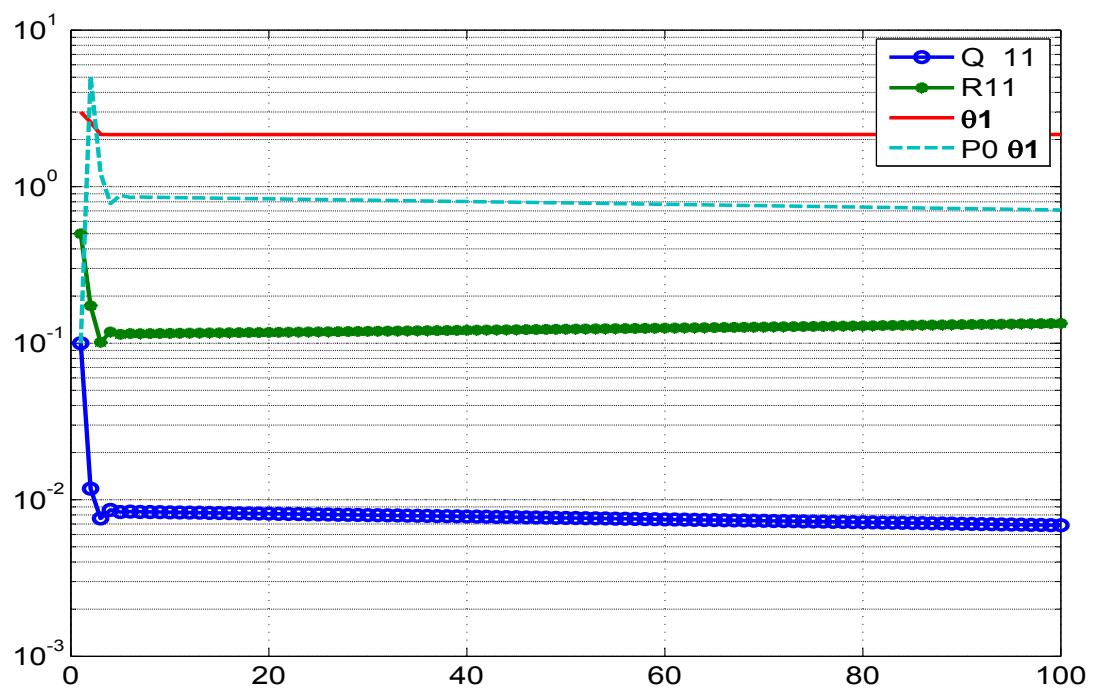
Figure 4.38: Variation of different estimates with iterations using MT method
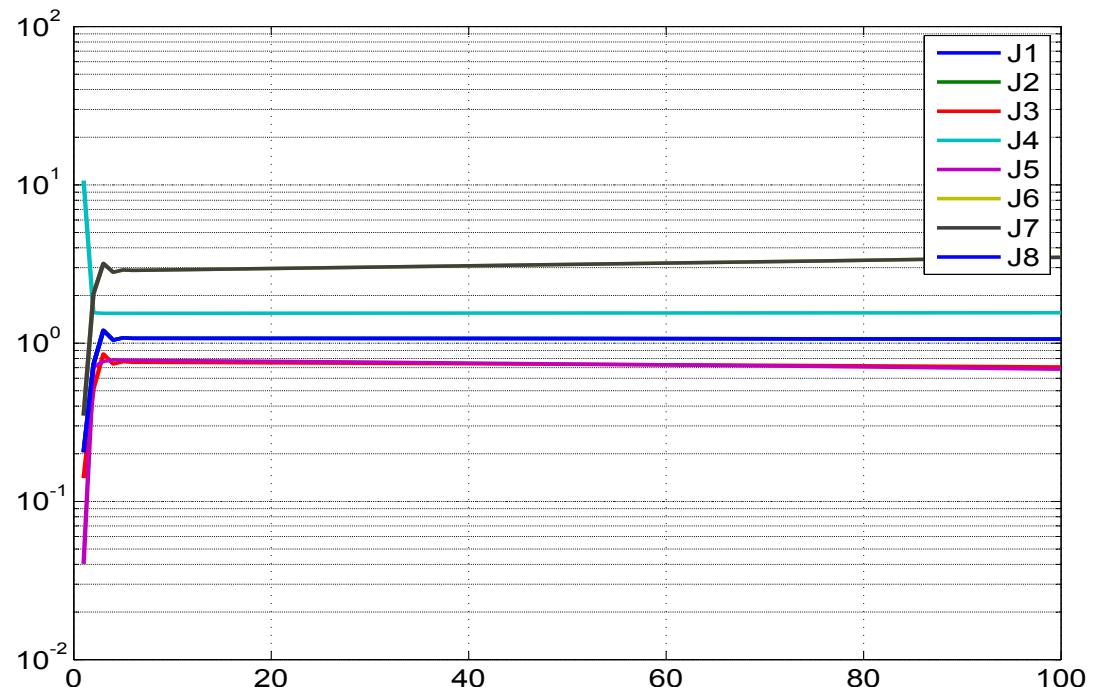
Figure 4.39: Variation of different costs with iterations using MT method

4.3   Simulated Spring, Mass, and Damper System

A spring mass damper (SMD) system with weak non-linearity in the continuous time (t) state space form is given by

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -\Theta_1 x_1(t) - \Theta_2 x_2(t) - \Theta_3 x_1^3(t)$$

where x_1, x_2 is the displacement and velocity state with initial condition 1,0 respectively. $\dot{x}$ represents differentiation with respect to time (t). The unknown parameter vector is $\Theta = [\Theta_1, \Theta_2, \Theta_3]^T$ with the true values being $\Theta_{true} = (4, 0.4, 0.6)^T$. The Θ_3 is a weak parameter since its value do not affect the system dynamics much. The complete state vector, $X = [x_1, x_2, \Theta_1, \Theta_2, \Theta_3]^T$ of size $(n + p) \times 1$. The measurement equation is given by

$$Z_k = HX_k + v_k$$

where $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$ is the measurement matrix of size $m \times (n + p)$ where $m = n = 2$ and $p = 3$. The numerical values of the noise variances are shown in Table-4.1. All the figures are presented for only one simulation run to prevent cluttering.

4.3.1   Remarks on the Results

We first run the filter assuming **Q** = 0. It was found that about 20 iterations of the data would suffice. The Fig. 4.40-4.51 refer to the **Q** = 0 case. The Fig. 4.40-4.42 shows the various parameter estimates and its corresponding variances through cumulative time instants with iterations. The variation of the estimated initial parameters and their variances through iterations are shown in Fig. 4.43. The parameter and the uncertainty reach almost their final estimated values in about 2 and 5 iterations respectively. A similar plot in Fig. 4.44 shows the variation of the estimated measurement noise. The variation of different cost functions (**J1-J5**) through the iterations is shown in Fig. 4.45. The Fig. 4.46-4.47 shows the predicted dynamics, filtered and smoothed estimate at the last iteration. The Fig. 4.48-4.49 show the innovations, filtered residue and smoothed residue together with the square root of their variance ($\pm\sigma$ bound). In the EKF approach most of the quantities

are Gaussian or approximated as quasi Gaussian and one would expect all the above quantities are close to being Gaussian and hence around one third of the total sample points to be outside the σ bound. The injected and estimated measurement noise distributions during the final iteration shown in Fig. 4.50-4.51 indicate that they are close to each other. Further even their autocorrelations are ideally expected to be close to the Kronecker delta function which provides confidence in the proposed filter algorithm.

The next step is to process the data with process noise ($\mathbf{Q} > 0$). The Fig. 4.52-4.63 refer to the $\mathbf{Q} > 0$ case. The Fig. 4.52 shows the absolute difference between the iterated and final values with iterations which indicates the accuracy level that one needs and it was found that 100 iterations are required. The variation of the estimated initial parameters and their variances through iterations are shown in Fig. 4.53. The parameter and the uncertainty reach almost their final estimated values in about 5 and 20 iterations respectively. A similar plot in Fig. 4.54 shows the variation of the estimated measurement and process noise. The variation of different cost functions through the iterations are shown in Fig. 4.55. The cost functions **J1-J3** correspond to the number of measurement (m=2) and in presence of process noise, **J6-J8** correspond to the number of states (n=2). The **J4** in absence of process noise corresponds to the trace of the measurement noise **R**. The **J5** is the negative log likelihood cost whose absolute value is shown in the plot. There is a mismatch in the predicted dynamics and the measurement as seen in Fig. 4.56-4.57 indicating the presence of process noise. The subsequent Fig. 4.58-4.59 correspond to the earlier Fig. 4.48-4.49 of $\mathbf{Q} = 0$ case. The Fig. 4.60-4.61 and Fig. 4.62-4.63 shows respectively the injected and estimated measurement and process noise samples across time during the final iteration.

4.3.2 SMD System Tables (Q = 0)

Table 4.6: Sensitivity Study : ($\mathbf{Q} = 0$) : SMD system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Consistency ratio-NR	Spread factor EKF	Spread factor NR	**R** ratio EKF/True	**R** ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
Reference adaptive EKF used for non zero process noise case											
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: EM-diag					Extremely slow convergence of $\mathbf{Q}$ taking more than 100 iterations.						
Proposed reference adaptive EKF for zero process noise case, $\mathbf{Q}$ is diagonal matrix fixed to 10^{-10} for all states											
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag RTS-Smoothing	1.0007 1.0000 0.9869	0.9952 1.0171 1.0173	1.0140 1.3764 1.0919	1.0533 1.3138 1.0817	0.8128 1.5488 14.5838	0.8126 1.5079 14.5843	0.9807 0.9743	1.0139 1.0128	1.9704 1.9702 1.9999 0.0048 -10.3911	0.0512 0.0512 0.0013 0.0008 0.2243	σ_Θ= 0.0240, 0.0055, 0.0670 $SIGMA_{avg}$= 0.0236, 0.0040, 0.0614
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag MBF-Smoothing	1.0007 1.0000 0.9869	0.9952 1.0171 1.0173	1.0141 1.3767 1.0918	1.0533 1.3138 1.0817	0.8128 1.5490 14.5835	0.8126 1.5079 14.5843	0.9807 0.9743	1.0139 1.0128	1.9704 1.9702 1.9998 0.0048 -10.3911	0.0512 0.0512 0.0013 0.0008 0.2243	σ_Θ= 0.0240, 0.0055, 0.0670 $SIGMA_{avg}$= 0.0236, 0.0040, 0.0614
Using different method and options for $\mathbf{P_0}$											
$\mathbf{P_0}$: Scaled up-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9998 1.0024 1.0255	0.8039 0.7206 0.5706	1.0518 1.2966 1.1761	1.0533 1.3138 1.0817	1.0223 2.1729 26.6449	0.8126 1.5079 14.5843	0.9786 0.9679	1.0117 1.0061	1.9530 1.9534 1.9998 0.0047 -10.3252	0.0570 0.0570 0.0050 0.0008 0.2308	Higher filter CRB than that of the NR
$\mathbf{P_0}$: Scaled up-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9991 1.0023 1.0471	1.0541 0.8635 0.9337	1.3638 1.5696 1.8936	1.0533 1.3138 1.0817	0.8922 2.0313 21.6826	0.8126 1.5079 14.5843	0.9716 0.9626	1.0045 1.0006	1.9668 1.9670 1.9999 0.0047 -10.4051	0.0438 0.0438 0.0014 0.0008 0.2293	Unity CRB ratio but with larger filter spread than that of NR

Table 4.6: Sensitivity Study : ($\mathbf{Q} = 0$) : SMD system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Consistency ratio-NR	Spread factor EKF	Spread factor NR	**R** ratio EKF/True	**R** ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
$\mathbf{P_0}$: IIM-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0014 0.9990 0.9667	0.9934 1.0058 1.0212	1.1296 2.2338 1.3654	1.0533 1.3138 1.0817	0.8997 2.3360 17.3808	0.8126 1.5079 14.5843	0.9894 0.9816	1.0228 1.0204	1.9690 1.9710 1.9998 0.0050 -10.2078	0.0776 0.0776 0.0015 0.0009 0.2349	Unity CRB ratio
$\mathbf{P_0}$: IIM-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9995 1.0023 1.0375	0.8007 0.7215 0.5713	1.0476 1.5883 1.1881	1.0533 1.3138 1.0817	1.0205 2.4209 26.8883	0.8126 1.5079 14.5843	0.9828 0.9718	1.0160 1.0102	1.9530 1.9542 1.9998 0.0049 -10.1918	0.0655 0.0655 0.0049 0.0009 0.2403	Higher filter CRB than that of the NR
$\mathbf{P_0}$: IIM-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9997 1.0025 1.0340	0.8004 0.7202 0.5714	1.0479 1.6662 1.1851	1.0533 1.3138 1.0817	1.0185 2.4988 26.7966	0.8126 1.5079 14.5843	0.9846 0.9732	1.0178 1.0116	1.9540 1.9532 1.9999 0.0050 -10.2001	0.0656 0.0656 0.0050 0.0009 0.2420	Higher filter CRB than that of the NR
$\mathbf{P_0}$: Smoothed-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0006 1.0002 0.9911	4.2358 4.3336 4.3190	4.2997 5.7348 4.5352	1.0533 1.3138 1.0817	0.5240 1.0375 9.5986	0.8126 1.5079 14.5843	0.9663 0.9617	0.9990 0.9997	1.9982 1.9982 1.9999 0.0048 -10.5276	0.0009 0.0009 0.0001 0.0008 0.2240	Lower filter CRB than that of the NR
$\mathbf{P_0}$: Smoothed-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0021 0.9999 0.9560	10.0871 4.2013 10.0720	10.8182 7.4621 16.1755	1.0533 1.3138 1.0817	0.5565 1.5098 14.2577	0.8126 1.5079 14.5843	0.9580 0.9497	0.9904 0.9872	1.9960 1.9960 1.9997 0.0047 -10.5493	0.0016 0.0016 0.0002 0.0008 0.2294	Lower filter CRB than that of the NR
$\mathbf{P_0}$: Smoothed-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9997 1.0011 1.0260	3.2374 3.0566 2.1902	4.1863 5.4617 4.3984	1.0533 1.3138 1.0817	0.6813 1.5528 18.0659	0.8126 1.5079 14.5843	0.9568 0.9485	0.9891 0.9859	1.9970 1.9971 1.9998 0.0047 -10.5523	0.0011 0.0011 0.0003 0.0008 0.2290	Lower filter CRB than that of the NR

Table 4.6: Sensitivity Study : ($\mathbf{Q} = 0$) : SMD system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Consistency ratio-NR	Spread factor EKF	Spread factor NR	**R** ratio EKF/True	**R** ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
					Using different methods for estimating **R**						
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	1.0007 1.0000 0.9868	1.0015 1.0234 1.0237	1.0191 1.3877 1.0984	1.0533 1.3138 1.0817	0.8094 1.5462 14.5332	0.8126 1.5079 14.5843	0.9699 0.9615	1.0027 0.9994	1.9950 1.9948 2.0248 0.0048 -10.3911	0.0589 0.0589 0.0271 0.0008 0.2244	*Using dynamical residue as per Eq-3.18
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	1.0007 1.0000 0.9869	1.0028 1.0248 1.0250	1.0218 1.3873 1.1004	1.0533 1.3138 1.0817	0.8094 1.5436 14.5273	0.8126 1.5079 14.5843	0.9665 0.9598	0.9992 0.9977	2.0001 1.9999 2.0300 0.0048 -10.3913	0.0522 0.0522 0.0037 0.0008 0.2243	*Using smoothed residue ignoring 2nd order terms in Eq-3.7
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	1.0007 1.0001 0.9871	1.0361 1.0588 1.0593	1.0531 1.4343 1.1331	1.0533 1.3138 1.0817	0.7939 1.5203 14.2471	0.8126 1.5079 14.5843	0.9257 0.8859	0.9570 0.9209	2.1324 2.1321 2.1643 0.0048 -10.3857	0.0264 0.0237 0.0704 0.0008 0.2241	*Using filtered residue ignoring 2nd order terms in Eq-3.19
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag $\mathbf{R}$*	1.0006 1.0003 0.9890	0.9927 1.0146 1.0148	0.9992 1.3854 1.0673	1.0533 1.3138 1.0817	0.8093 1.5590 14.4567	0.8126 1.5079 14.5843	0.9925 0.9874	1.0261 1.0264	1.9800 1.9797 2.0098 0.0048 -10.3709	0.2788 0.2792 0.2882 0.0008 0.2286	*Using innovations ignoring 2nd order terms in Eq-3.20
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: MS-diag	1.0007 1.0001 0.9870	1.0020 1.0239 1.0243	1.0189 1.3864 1.0958	1.0533 1.3138 1.0817	0.8090 1.5437 14.5024	0.8126 1.5079 14.5843	0.9777 0.9552	1.0108 0.9929	1.9959 1.9957 2.0258 0.0048 -10.3904	0.0198 0.0151 0.0579 0.0008 0.2241	Using full expression in Eq-3.19
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: MT-diag	1.0006 0.9997 0.9860	1.0074 1.0296 1.0298	1.0286 1.3862 1.1048	1.0533 1.3138 1.0817	0.8339 1.6746 15.0101	0.8126 1.5079 14.5843	0.9639 0.9605	0.9965 0.9984	2.0385 2.0382 2.0692 0.0048 -10.3702	0.2893 0.2897 0.2995 0.0008 0.2293	Using full expression in Eq-3.20

4.3.3 SMD System Tables ($\mathbf{Q} > 0$)

Table 4.7: Sensitivity Study : ($\mathbf{Q} > 0$) : SMD system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
Proposed reference adaptive EKF for non zero process noise									
P$_0$: Scaled up-[0,0;0,$\checkmark$] **Q** : EM-[$\checkmark$,0;0,0] **R** : EM-diag RTS-Smoothing	0.9933 1.0201 1.0764	1.0179 1.0286	1.0327 1.0772 0.9128	7.9434 23.5655 98.4334	0.9450 0.9135	1.0648 1.0922	1.9650 1.9700 1.9982 0.0709 -8.7886 1.9439 1.9533 1.9585	0.0224 0.0217 0.0091 0.0396 0.2281 0.0422 0.0618 0.0373	σ_Θ= 0.2418, 0.0720, 0.4089 $SIGMA_{avg}$= 0.2342, 0.0668, 0.4480
P$_0$: Scaled up-[0,0;0,$\checkmark$] **Q** : EM-[$\checkmark$,0;0,0] **R** : EM-diag MBF-Smoothing	0.9933 1.0203 1.0764	1.0178 1.0309	1.0314 1.0762 0.9113	7.9494 23.5909 98.4953	0.9445 0.9116	1.0648 1.0965	1.9656 1.9707 1.9994 0.0709 -8.7886 1.9441 1.9535 1.9592	0.0221 0.0219 0.0029 0.0396 0.2281 0.0429 0.0623 0.0366	σ_Θ= 0.2418, 0.0720, 0.4087 $SIGMA_{avg}$= 0.2345, 0.0669, 0.4485

113

Table 4.7: Sensitivity Study : ($\mathbf{Q} > 0$) : SMD system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**Q** ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
P$_0$: Scaled up-[0,0;0,✓] **Q** : EM-[✓,0;0,0] **R** : Known	0.9942 1.0143 1.0705	0.9914 0.9926	1.0052 1.0596 0.9034	7.8903 23.2708 97.1068	NA	1.0343 1.0112	1.9489 1.9516 1.9670 0.0691 -8.7675 1.9335 1.9395 1.9456	0.1121 0.1136 0.1862 0.0357 0.2236 0.0350 0.0451 0.1118	**R** is known
P$_0$: Zero **Q** : EM-diag **R** : EM-diag	NA	1.0136 1.0214	NA	NA	0.9484 0.9248	1.0582 1.0601	1.9997 2.0011 1.9981 0.0819 -8.8857 1.9978 2.0067 1.9982	0.0216 0.0216 0.0083 0.0508 0.2244 0.0158 0.0403 0.0246	The parameters are assumed to be known
P$_0$: Zero **Q** : EM-diag **R** : EM-diag	NA	0.9917 0.9965	NA	NA	NA	1.0336 0.9985	1.9870 1.9881 1.9735 0.0819 -8.8731 1.9972 2.0049 1.9860	0.1136 0.1153 0.1893 0.0508 0.2220 0.0181 0.0347 0.1124	The parameters and **R** are assumed to be known

Table 4.7: Sensitivity Study : ($\mathbf{Q} > 0$) : SMD system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**Q** ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
Extended EM algorithm (Bavdekar et al.[104] (2011))									
P₀ : Smoothed-full **Q** : EM-full **R** : EM-full	Cost functions diverges after few iterations. There is a need for a precise stopping condition without which unity ratios cannot be achieved.								
Modification-1									
P₀ : Smoothed-diag **Q** : EM-diag **R** : EM-diag	1.0210 1.0018 0.6806	1.0164 1.0473	13.3560 10.0828 13.6464	3.9686 13.6349 51.6473	0.9502 0.9703	1.0286 0.8676	1.9988 1.9996 1.9998 0.0656 -8.9374 2.0031 2.0027 1.9976	0.0241 0.0241 0.0076 0.0325 0.2290 0.0147 0.0239 0.0282	No stoping condition. The low spread factor is deceiving since there is no consistency in the parameter estimates with its covariance.
Modification-2									
P₀ : Scaled up-[0,0;0,✓] **Q** : EM-diag **R** : EM-diag	0.9938 1.0185 1.0794	1.0133 1.0221	0.9788 1.0371 0.8933	7.9586 23.5949 98.7913	0.9485 0.9144	1.0577 1.0883	1.9658 1.9690 1.9979 0.0692 -8.7796 1.9346 1.9427 1.9613	0.0250 0.0250 0.0091 0.0357 0.2278 0.0608 0.0629 0.0351	The slight differences from the reference case is due to the small **Q** that is generated for the parameters
Modification-3									
P₀ : Scaled up-[0,0;0,✓] **Q** : DSDT-diag **R** : EM-diag	0.9931 1.0205 1.0739	1.0178 1.0258	1.0329 1.0701 0.9015	7.9837 23.6494 98.4487	0.9446 0.9100	1.0658 1.1028	1.9634 1.9691 1.9982 0.0710 -8.7877 1.9334 1.9406 1.9565	0.0256 0.0247 0.0090 0.0402 0.2274 0.0374 0.0394 0.0411	Using DSDT statistics for **Q** estimation given by Eq-3.17

Table 4.7: Sensitivity Study : ($\mathbf{Q} > 0$) : SMD system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
MS method (Mohamed and Schwarz)									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MS-diag $\mathbf{R}$: MS-diag	$\mathbf{R}$ can converge to a very large value and the cost functions diverge as seen in Fig. 4.64,4.65. There is no convergence in cost even if Θ is known.								
MS method with $\mathbf{Q}$ being estimated using last N/2 samples									
$\mathbf{P_0}$: Zero $\mathbf{Q}$: MS-diag $\mathbf{R}$: MS-diag	NA	1.0084 1.0437	NA	NA	NA	0.9510 0.8739	2.1281 2.1309 2.0729 0.0819 -8.8300 2.3141 2.3369 2.1230	0.3269 0.3290 0.2541 0.0508 0.2363 0.6473 0.6726 0.3213	$\mathbf{R}$, Θ is known
MS method assuming $\mathbf{R}$ is known and $\mathbf{Q}$ estimated using last N/2 samples									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MS-[✓,0;0,0] $\mathbf{R}$: Known	0.9926 1.0258 1.0883	0.9989 0.9756	1.0005 1.0762 0.8684	8.1555 24.2527 99.2637	NA	1.0085 1.1562	1.9341 1.9419 1.9466 0.0698 -8.7546 1.9178 1.9220 1.9260	0.0842 0.0902 0.1191 0.0381 0.2185 0.2016 0.2007 0.0836	Stable operation when $\mathbf{R}$ is known
Gemson's approach using MS method									
$\mathbf{P_0}$: IIM-full $\mathbf{Q}$: MS-[✓,0;0,0] $\mathbf{R}$: 0.9*True $\mathbf{R}$	0.9923 1.0252 1.1360	1.0156 0.9709	0.8727 0.9292 0.7758	8.8231 25.6543 115.4840	0.8778 0.8708	1.1406 1.4149	1.9072 1.9090 1.9881 0.0689 -8.7388 1.8263 1.8290 1.9062	0.0586 0.0625 0.1190 0.0351 0.2182 0.1641 0.1626 0.0591	δ J = 10^{-6} with one loop of $\mathbf{R}$ estimation

Table 4.7: Sensitivity Study : ($\mathbf{Q} > 0$) : SMD system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
Covariance matching method suggested by Myers and Tapley (MT)									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MT-diag $\mathbf{R}$: MT-diag	Positive definiteness of $\mathbf{Q}$, $\mathbf{R}$ is not gauranteed and the cost functions can diverge as seen in Fig. 4.66,-4.67. There is no proper convergence even if Θ is known.								
MT method with $\mathbf{Q}$ being estimated using only last N/2 samples									
$\mathbf{P_0}$: Zero $\mathbf{Q}$: MS-diag $\mathbf{R}$: MS-diag	NA	1.0058 1.0145	NA	NA	NA	0.9716 1.0205	2.0620 2.0643 2.0266 0.0819 -8.8372 2.2149 2.7665 2.0569	0.3138 0.3153 0.2537 0.0508 0.2300 0.6658 4.2955 0.3092	$\mathbf{R}$, Θ is known
MT method assuming $\mathbf{R}$ is known and $\mathbf{Q}$ is estimated using only last N/2 samples									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MT-[✓,0;0,0] $\mathbf{R}$: Known	0.9915 1.0148 1.1158	1.0091 1.0223	1.1623 1.1677 1.1151	7.9841 23.0947 100.5373	NA	0.9940 1.0208	2.0247 2.0656 2.0439 0.0698 -8.7304 2.0215 2.0197 1.9923	0.2809 0.2941 0.2493 0.0382 0.2290 0.4419 0.4271 0.2666	Stable operation
Gemson's approach using MT method									
$\mathbf{P_0}$: IIM-full $\mathbf{Q}$: MT-[✓,0;0,0] $\mathbf{R}$: 0.9*True $\mathbf{R}$	0.9913 1.0186 1.1403	1.0262 1.0151	0.9376 1.0095 0.8315	8.3555 24.4327 110.5494	0.8778 0.8708	1.0928 1.1996	2.0236 2.0260 2.0794 0.0682 -8.7237 1.9761 1.9810 2.0207	0.2895 0.2915 0.2576 0.0342 0.2231 0.6273 0.6333 0.2867	δ J=10^{-6} with one loop of $\mathbf{R}$ estimation

4.3.4 SMD System Figures (Q = 0)
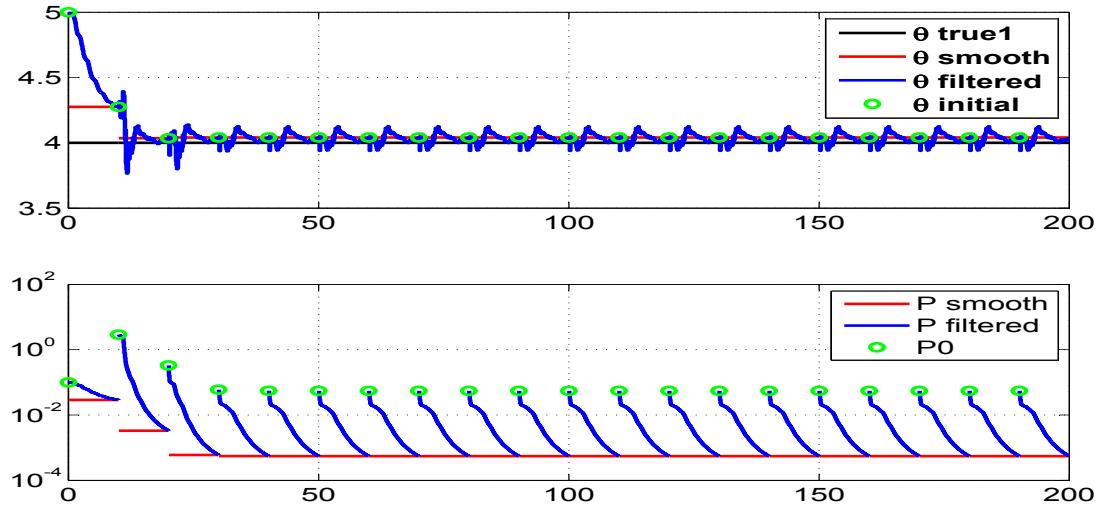
Figure 4.40: The variation of parameter estimate 1 and their filtered and

smoothed covariances through (with the time cumulatively) the iterations
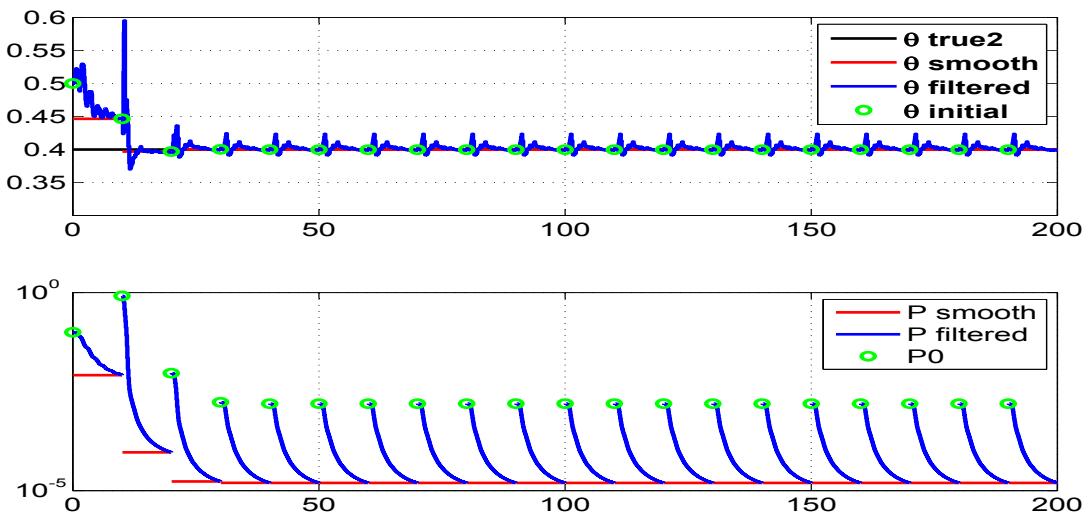
Figure 4.41: The variation of parameter estimate 2 and their filtered and

smoothed covariances through (with the time cumulatively) the iterations
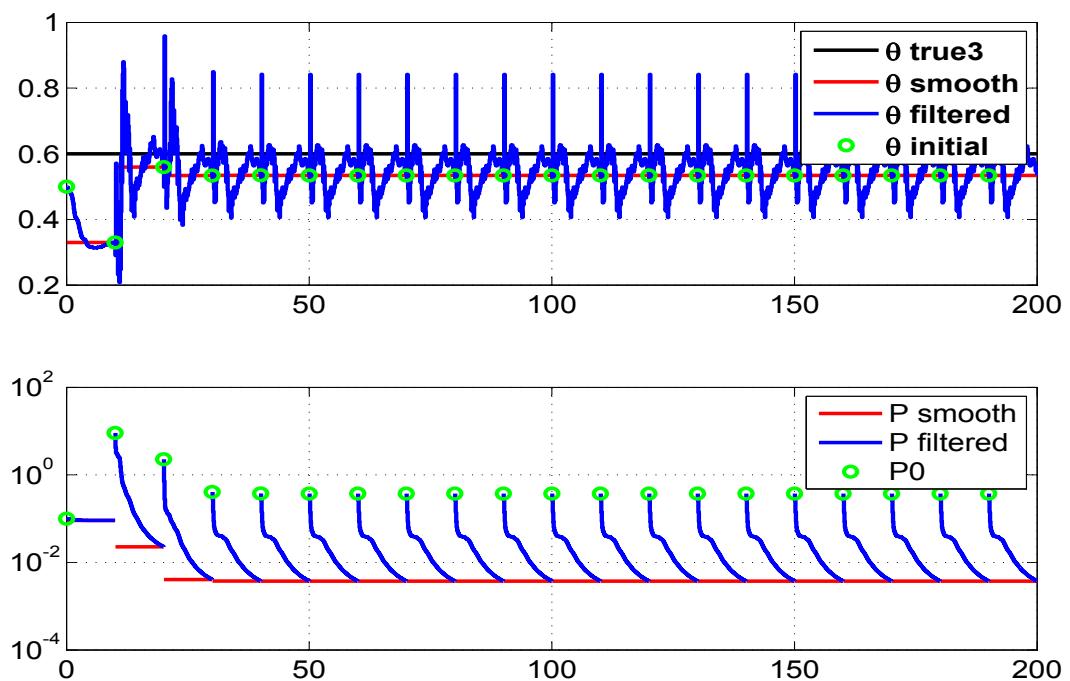
Figure 4.42: The variation of parameter estimate 3 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
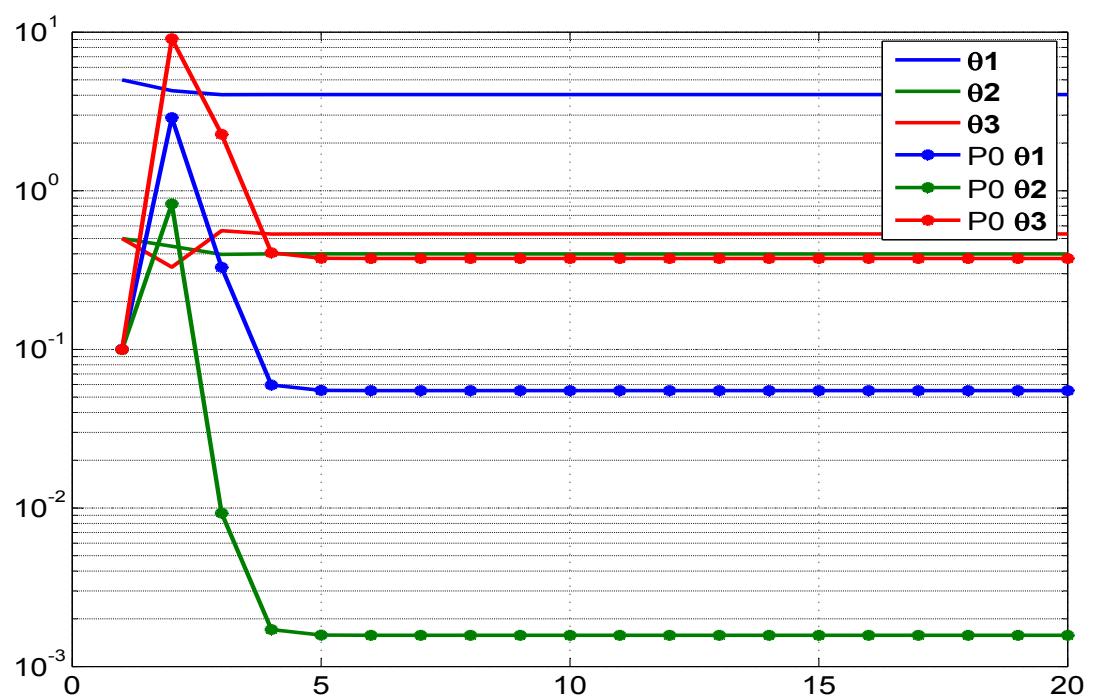
Figure 4.43: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations

119
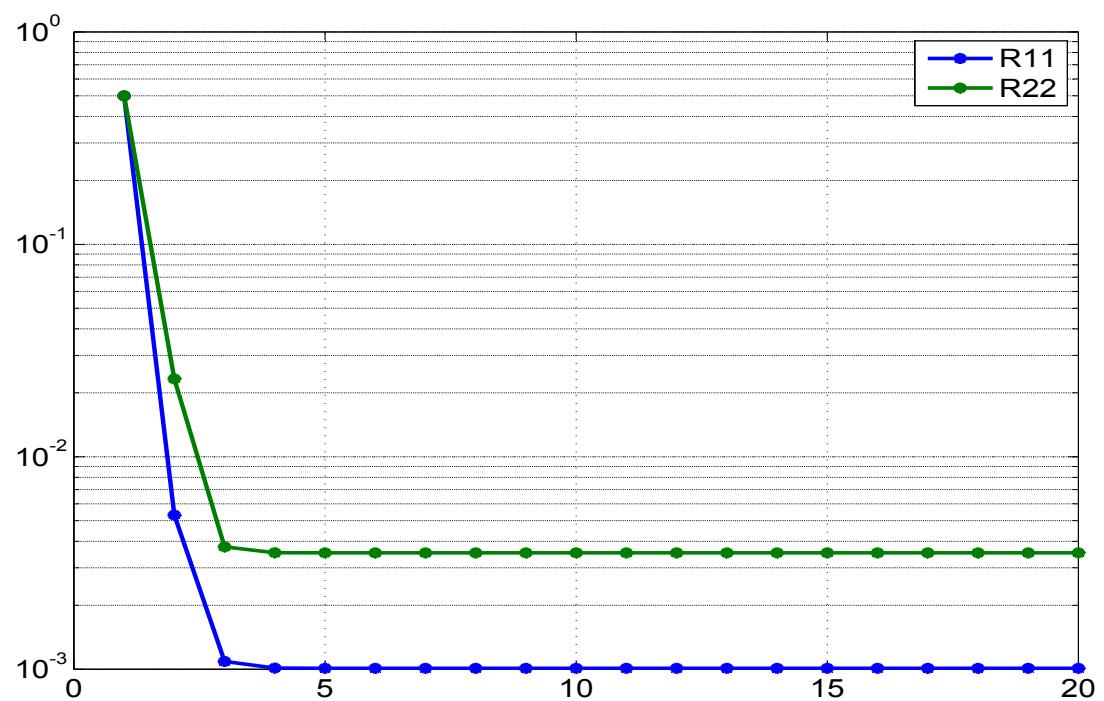
Figure 4.44: Variation of **R** with iterations
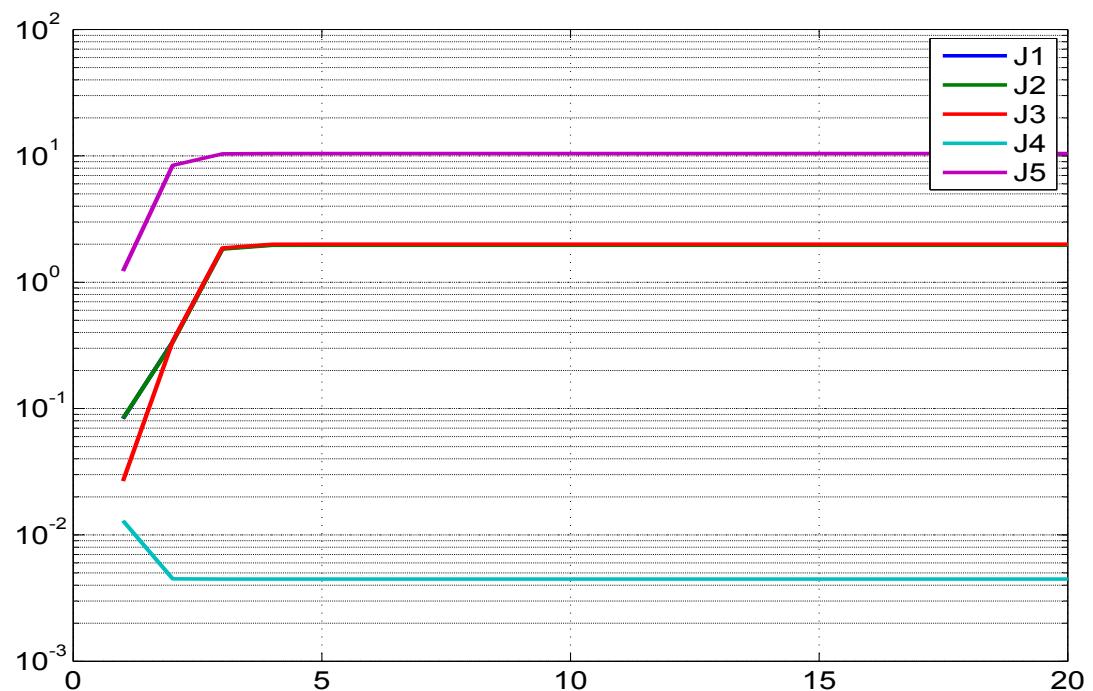
Figure 4.45: Variation of different costs (**J1-J5**) with iterations
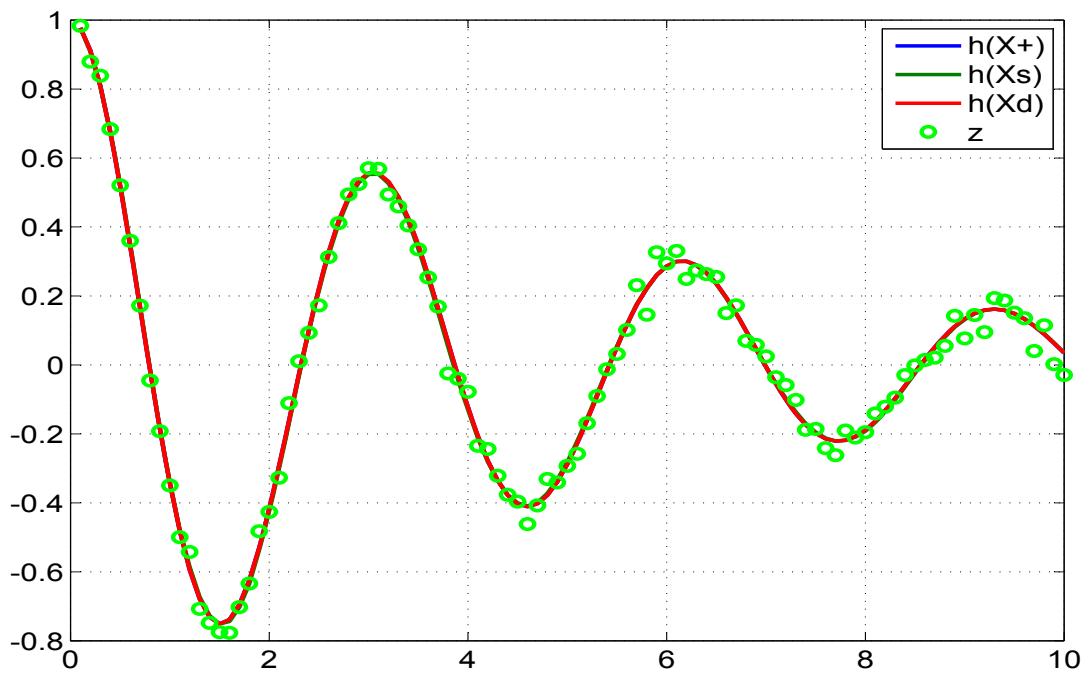
Figure 4.46: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1 (displacement)
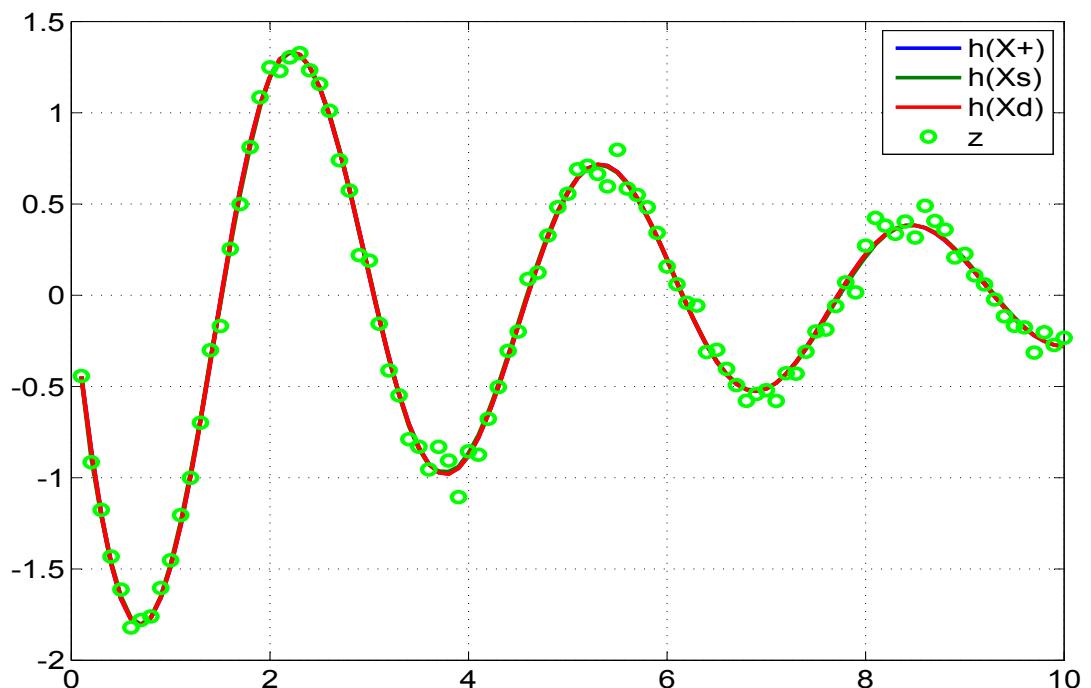
Figure 4.47: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 2 (velocity)
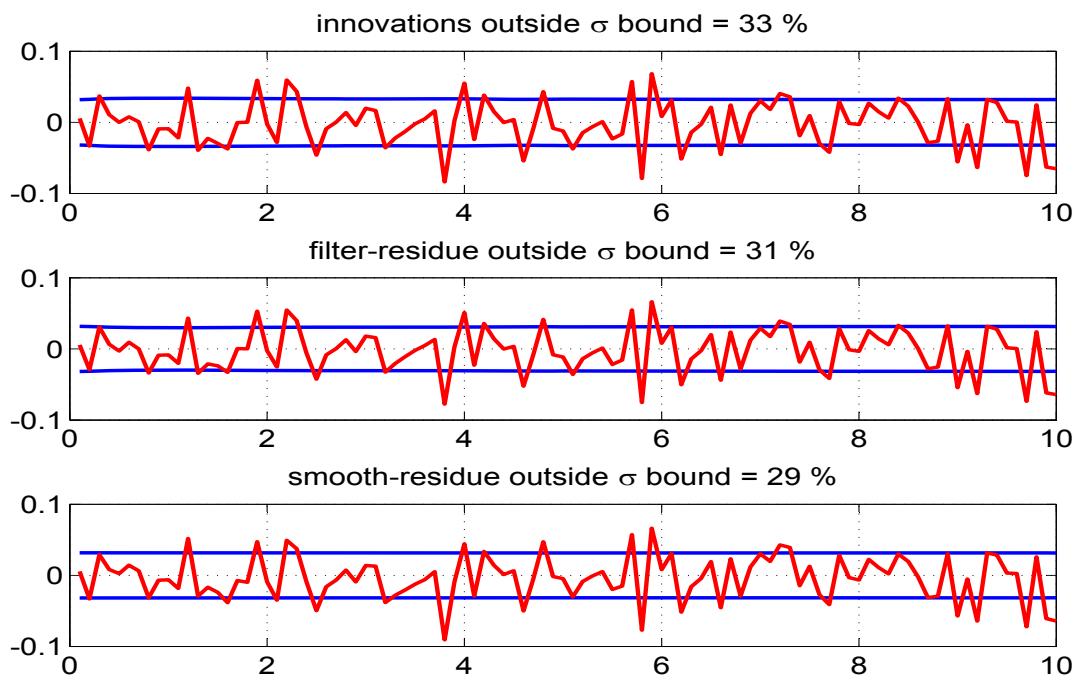
Figure 4.48: The innovations, filtered residue and smoothed residue

corresponding to measurement 1 (displacement)
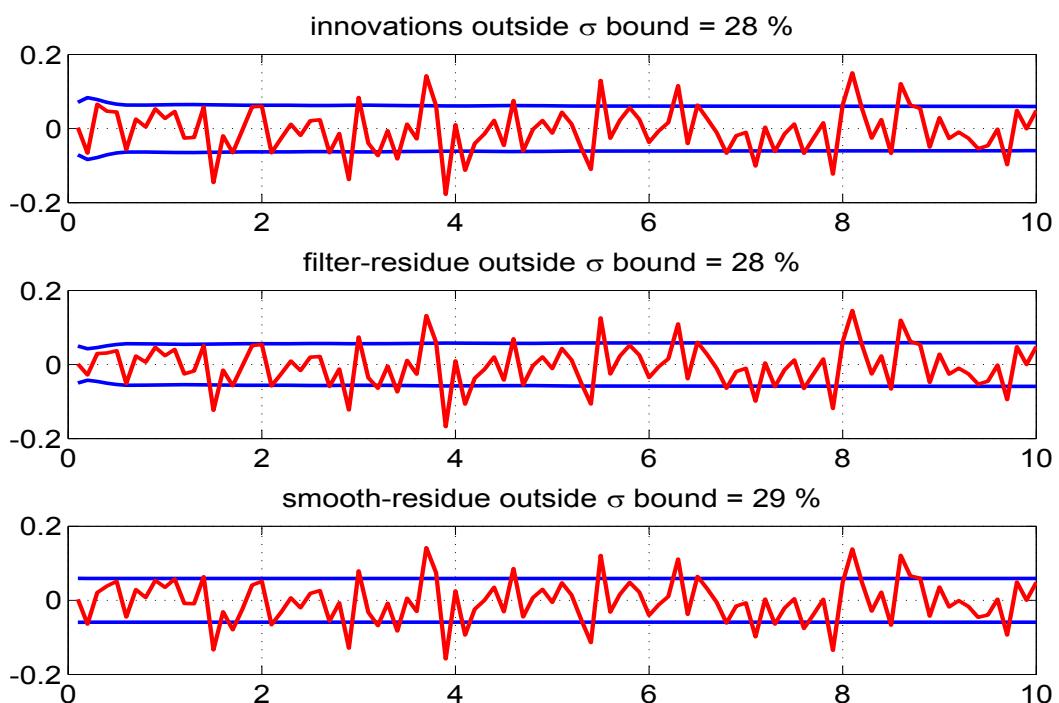
Figure 4.49: The innovations, filtered residue and smoothed residue

corresponding to measurement 2 (velocity)
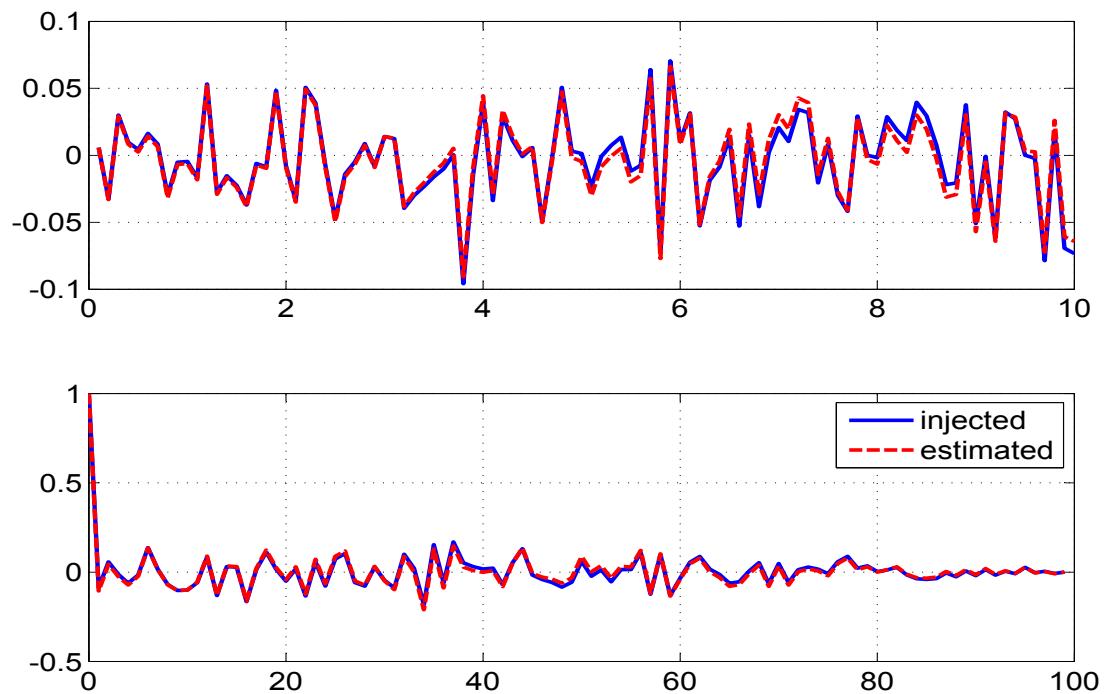
Figure 4.50: Time variation of injected and estimated measurement noise (top) and
their autocorrelation (bottom) for measurement 1 (displacement)
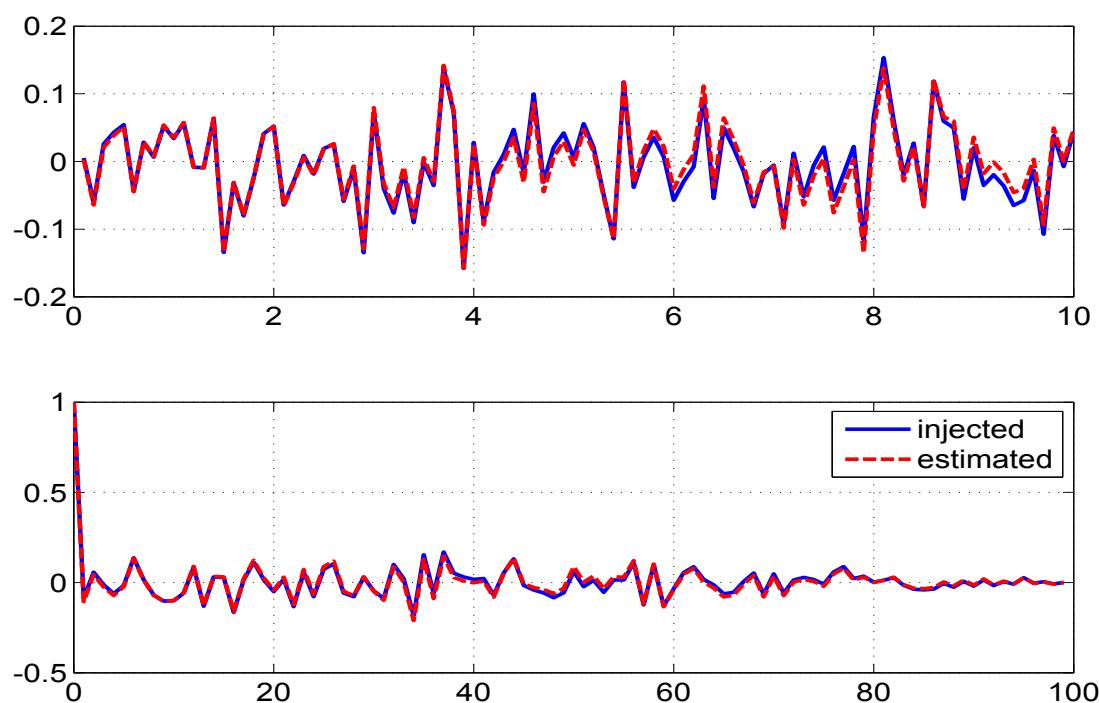
Figure 4.51: Time variation of injected and estimated measurement noise (top) and
their autocorrelation (bottom) for measurement 2 (velocity)

4.3.5 SMD System Figures (Q > 0)
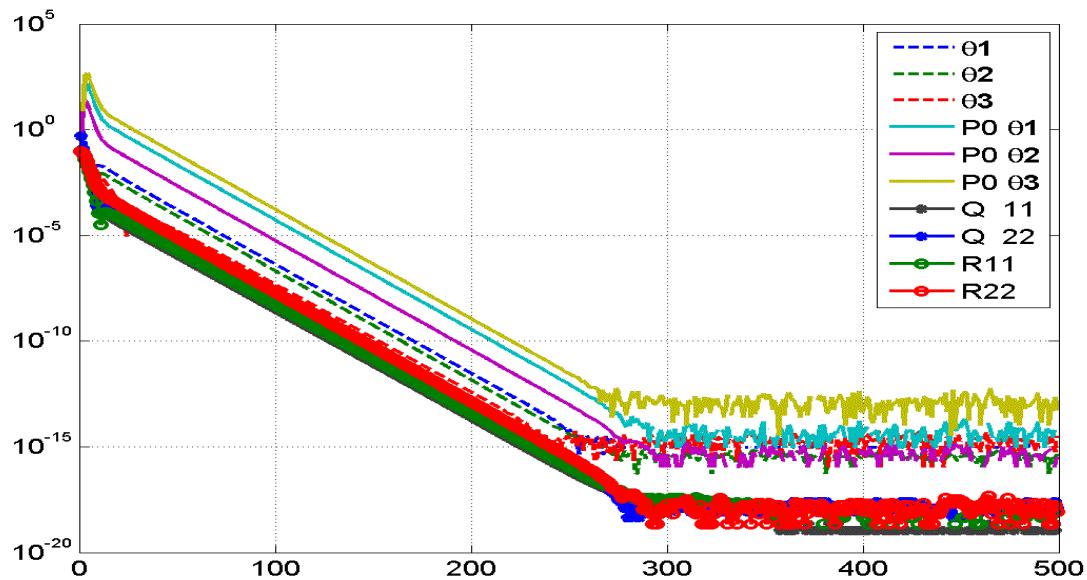
Figure 4.52: The absolute difference between the iterated and final values

with 500 iterations
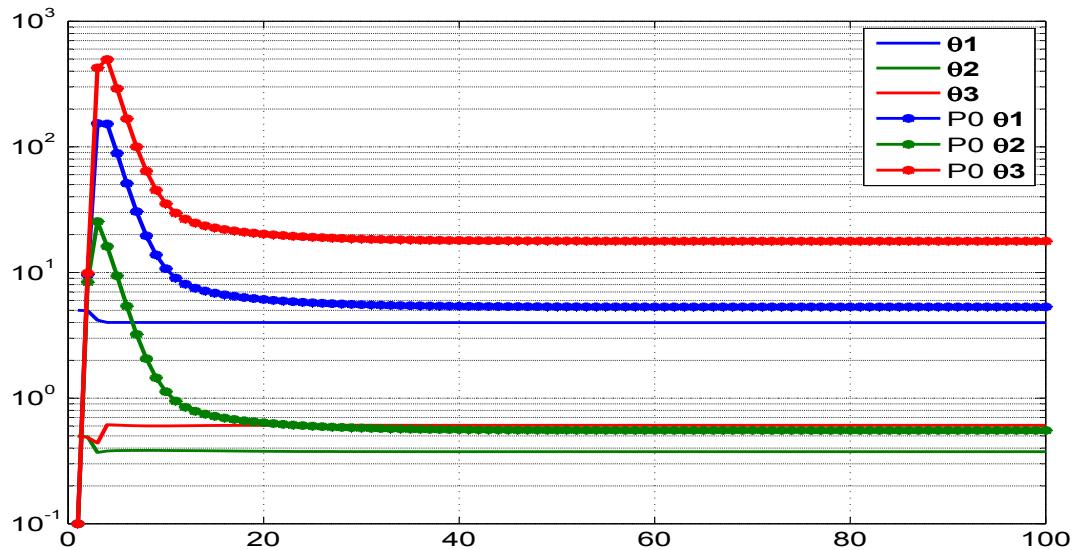
Figure 4.53: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
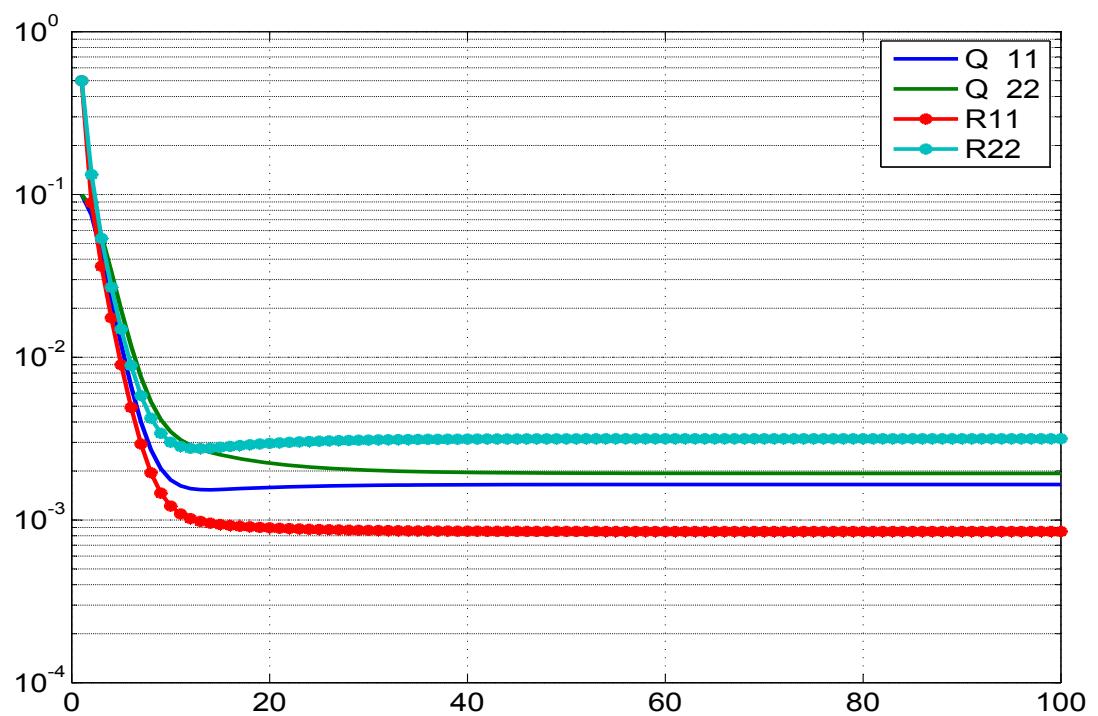
Figure 4.54: Variation of **Q** and **R** with iterations
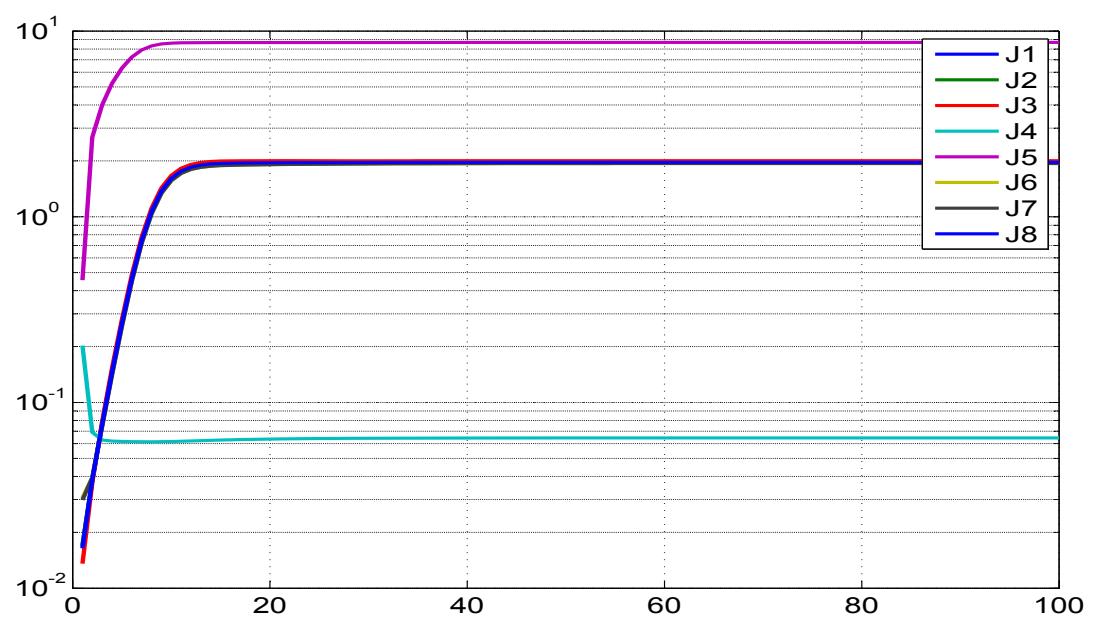
Figure 4.55: Variation of different costs (**J1-J8**) with iterations

125
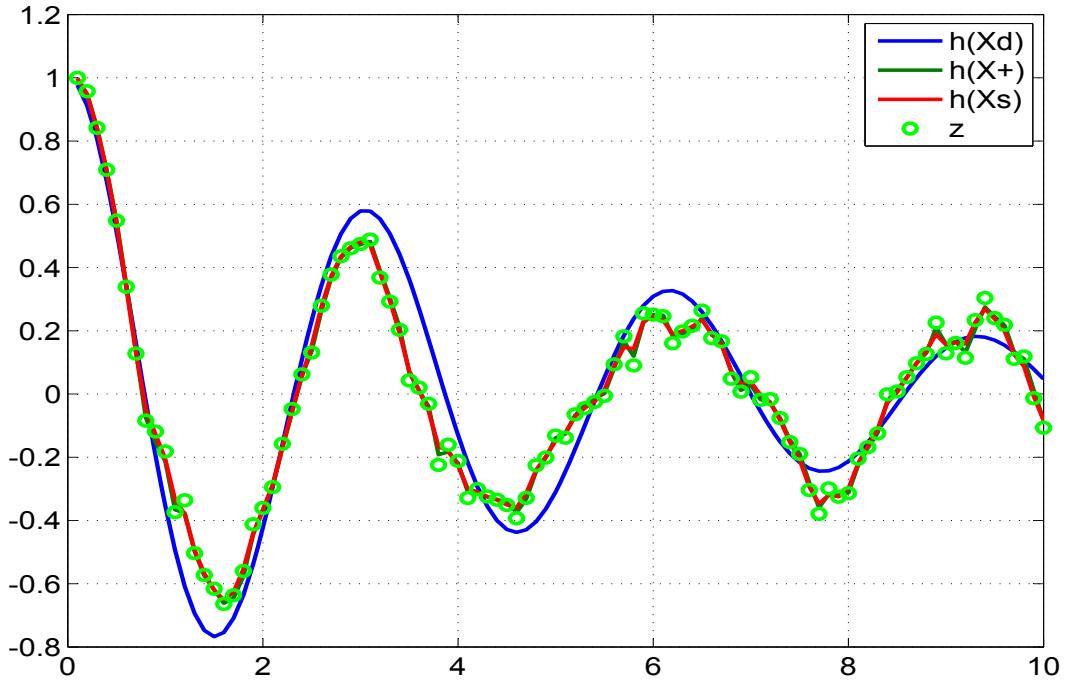
Figure 4.56: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1 (displacement)
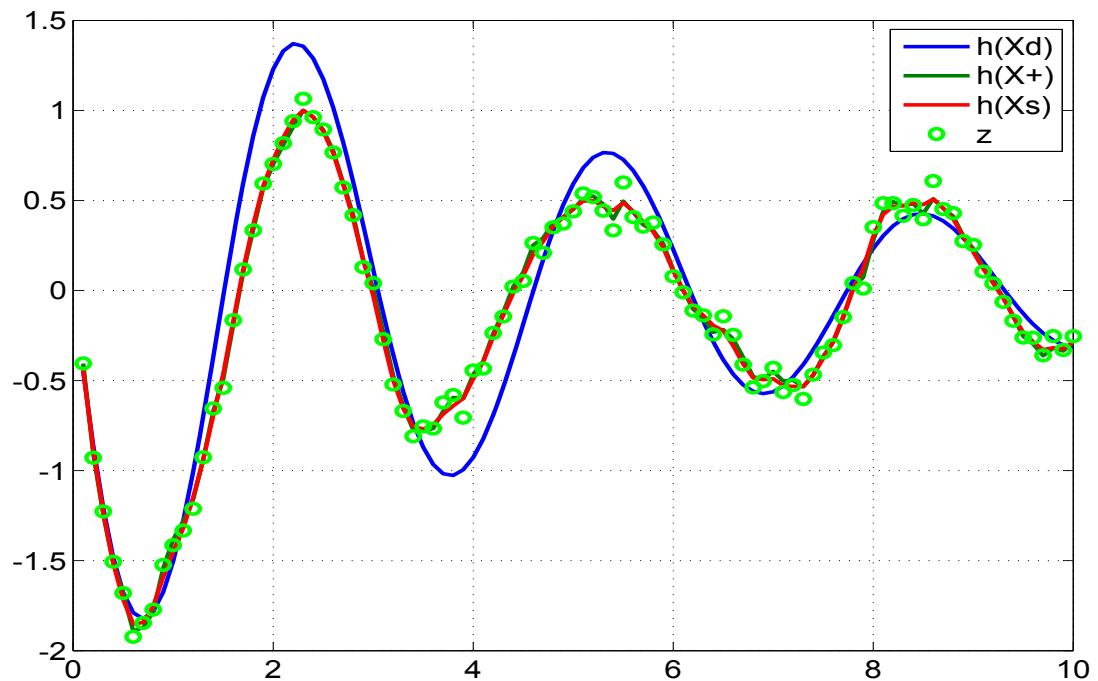
Figure 4.57: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 2 (velocity)
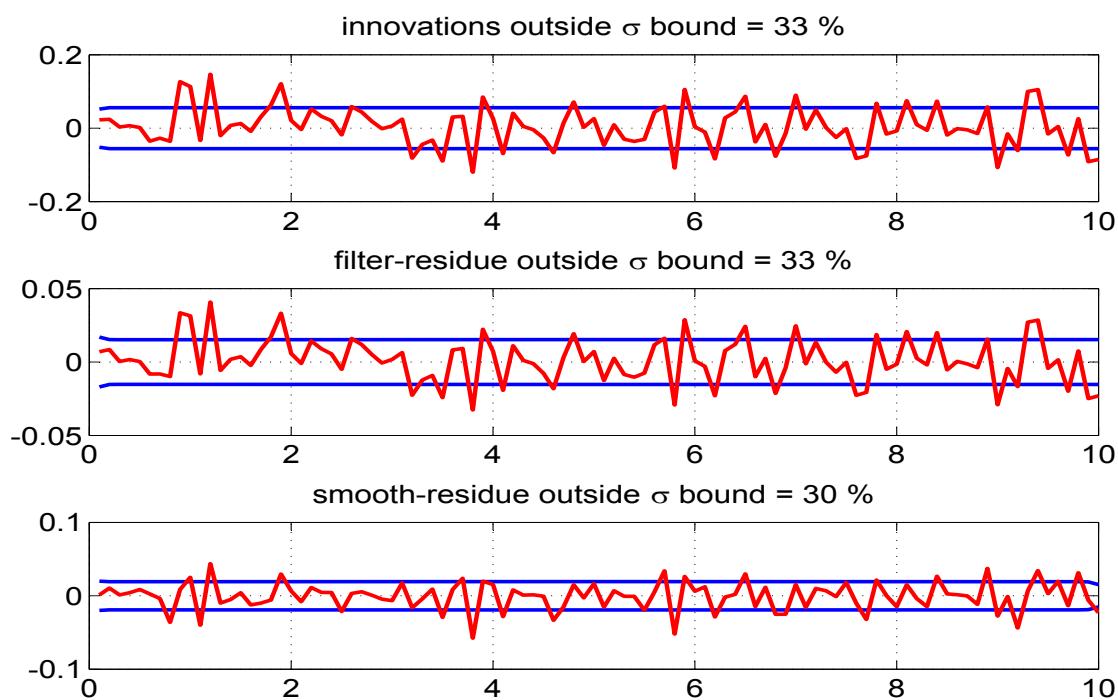
Figure 4.58: The innovations, filtered residue and smoothed residue

corresponding to measurement 1 (displacement)
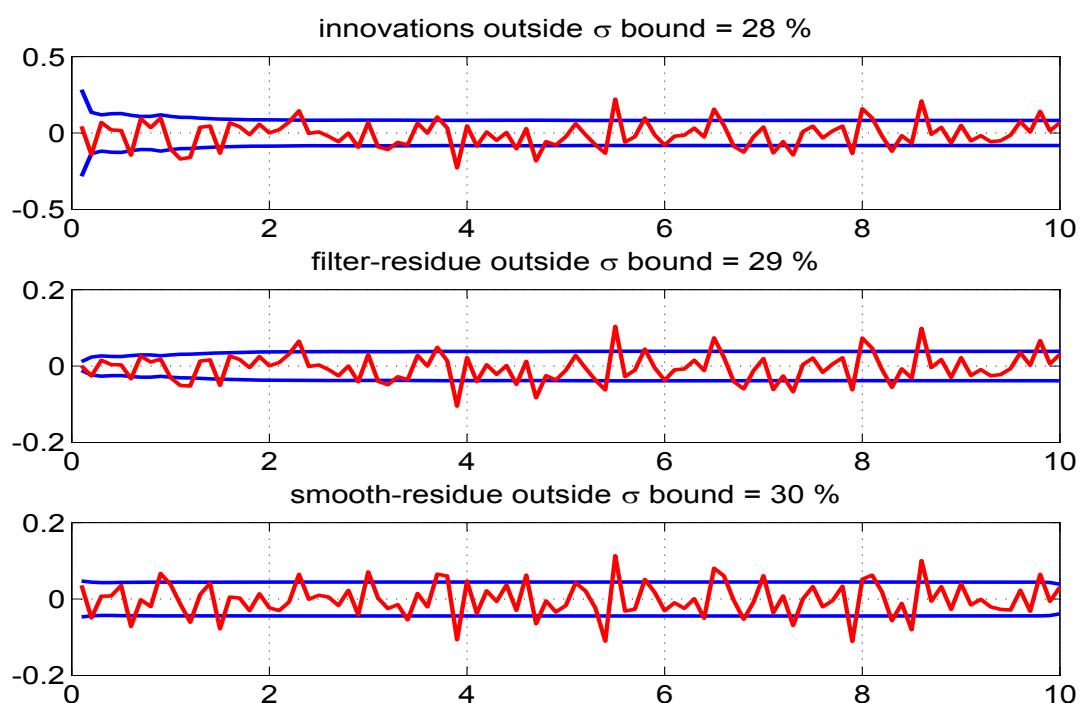
Figure 4.59: The innovations, filtered residue and smoothed residue

corresponding to measurement 2 (velocity)
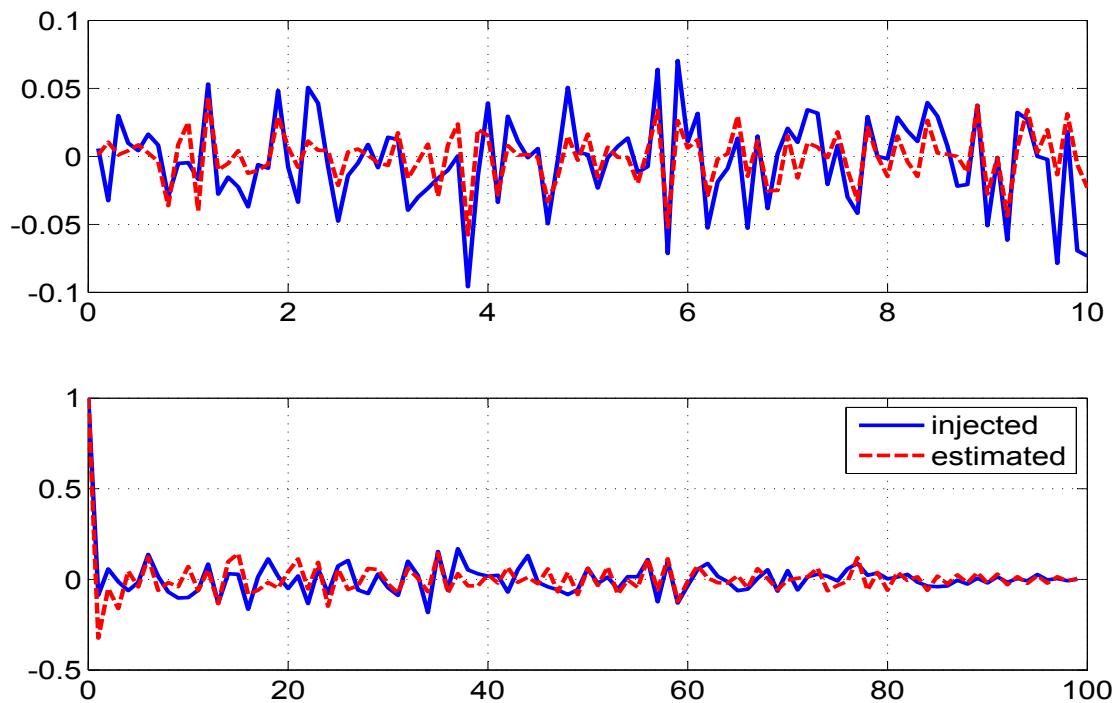
Figure 4.60: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 1 (displacement)
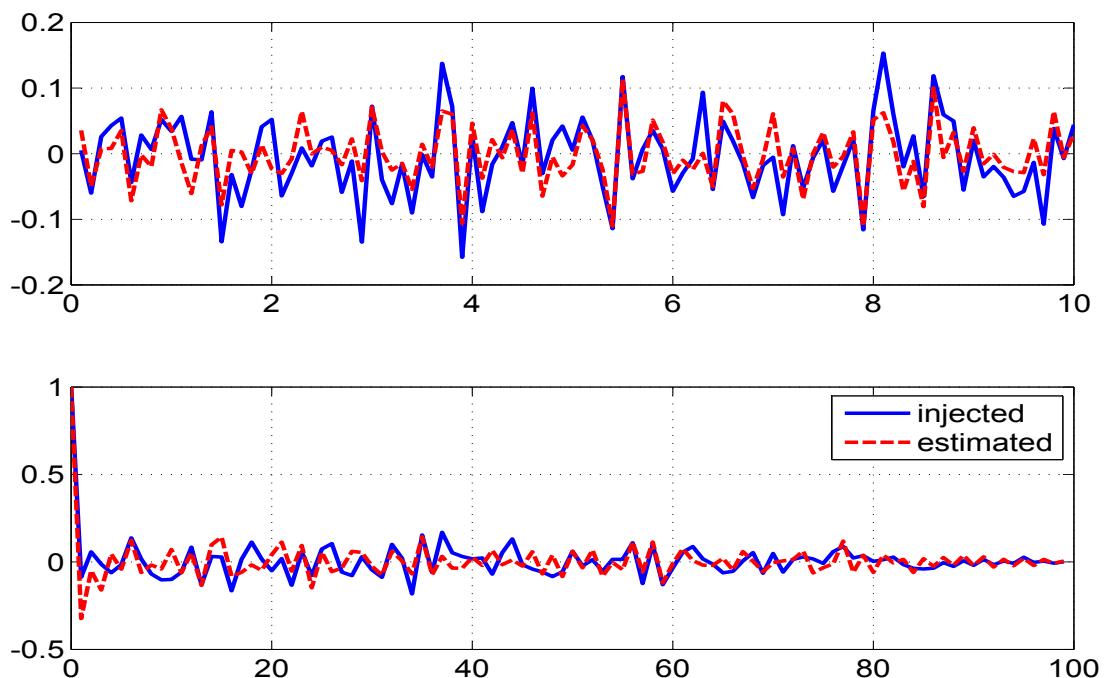
Figure 4.61: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 2 (velocity)
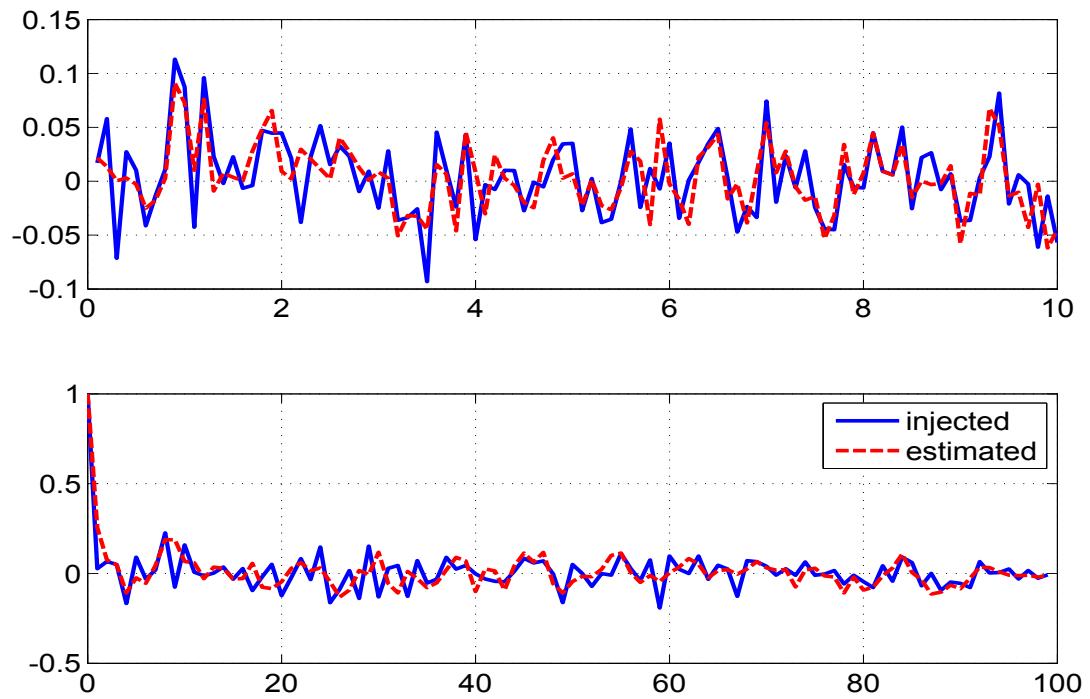
Figure 4.62: Time variation of injected and estimated process noise (top) and

their autocorrelation (bottom) for state 1 (displacement)
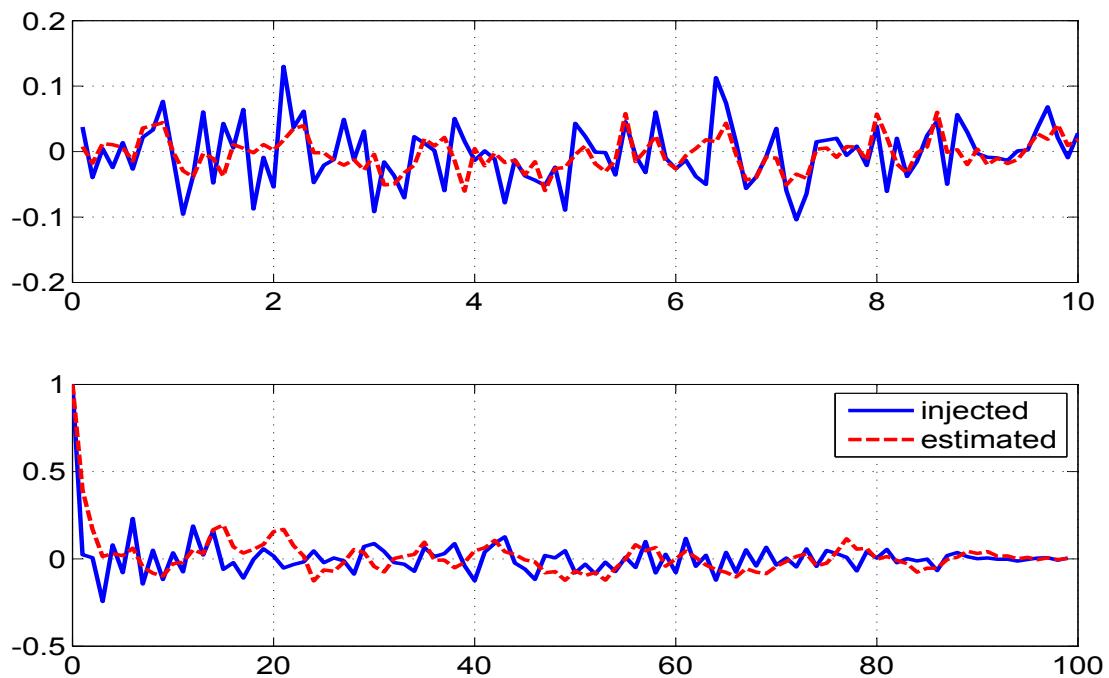
Figure 4.63: Time variation of injected and estimated process noise (top) and

their autocorrelation (bottom) for state 2 (velocity)
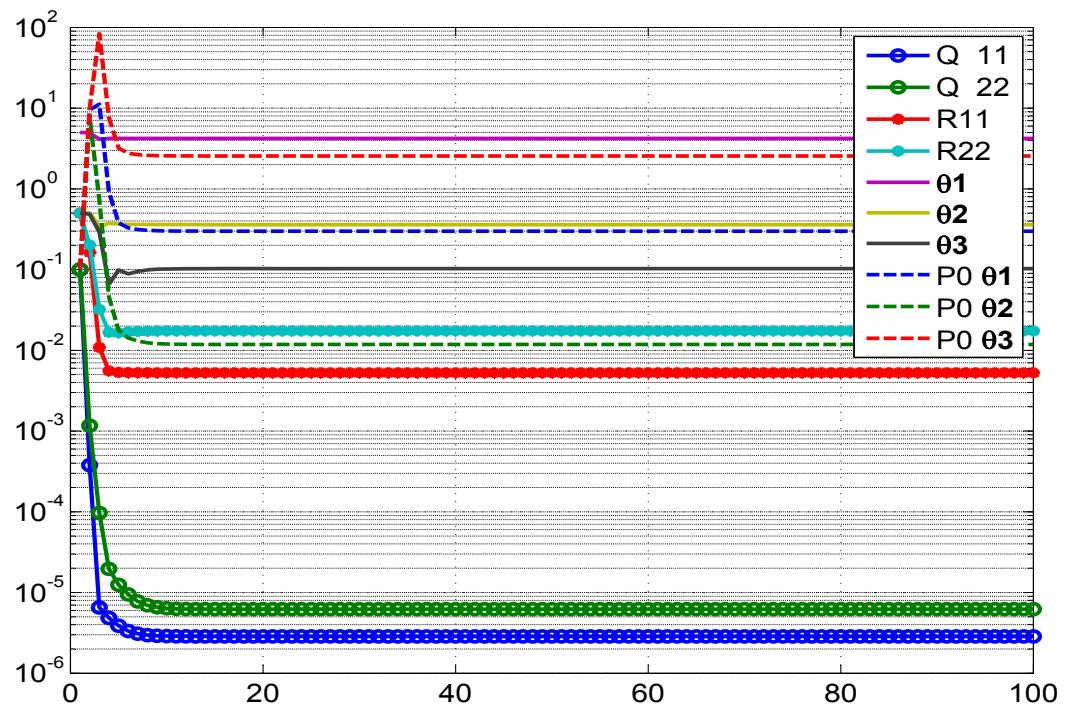
Figure 4.64: Variation of different estimates with iterations using MS method
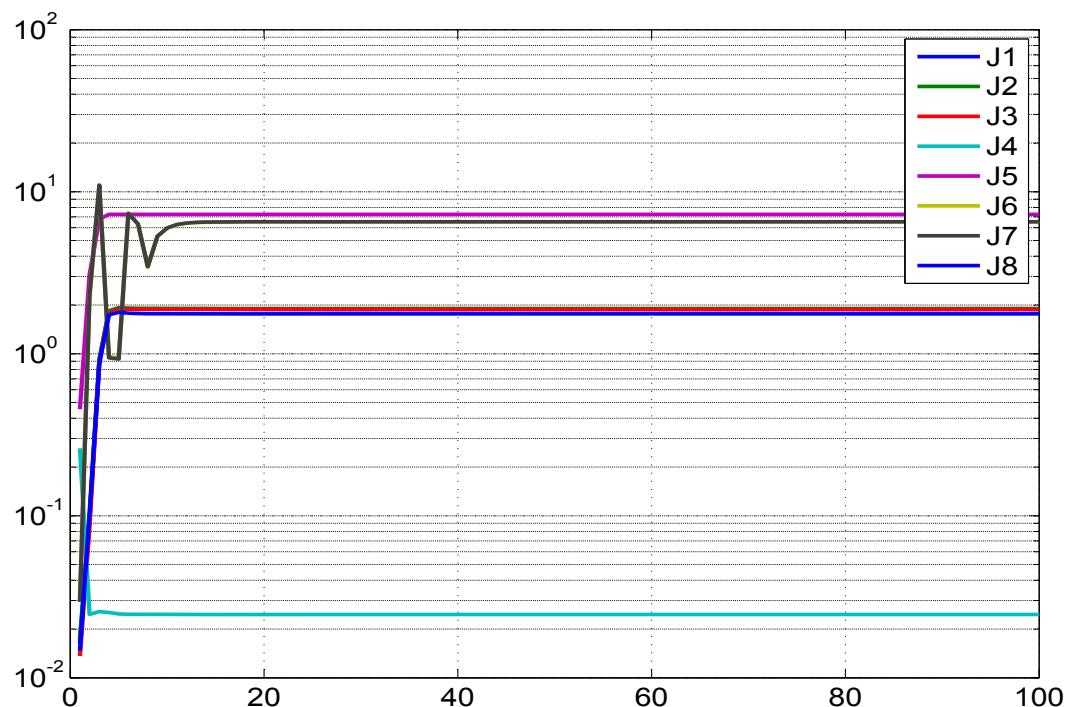
Figure 4.65: Variation of different costs with iterations using MS method

130
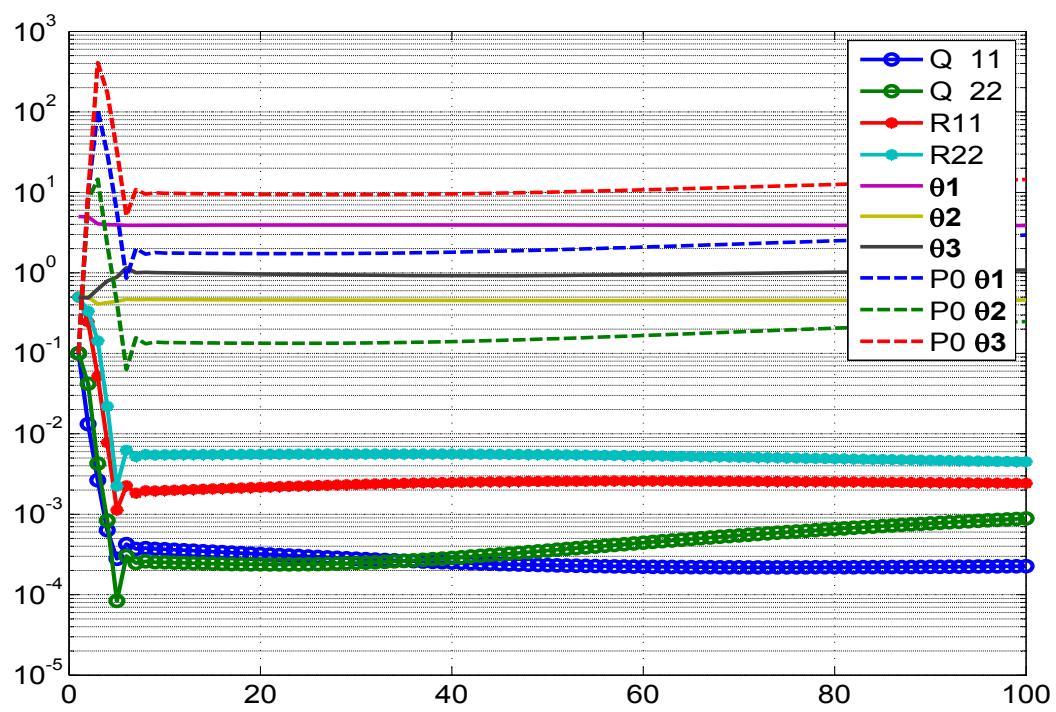
Figure 4.66: Variation of different estimates with iterations using MT method
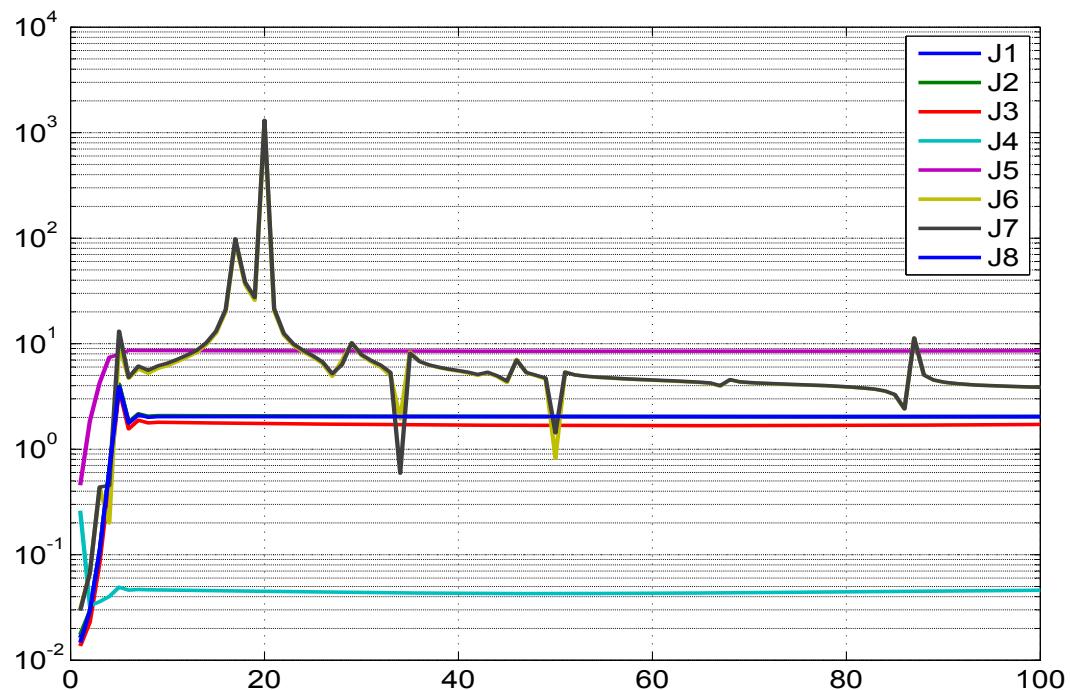
Figure 4.67: Variation of different costs with iterations using MT method

131
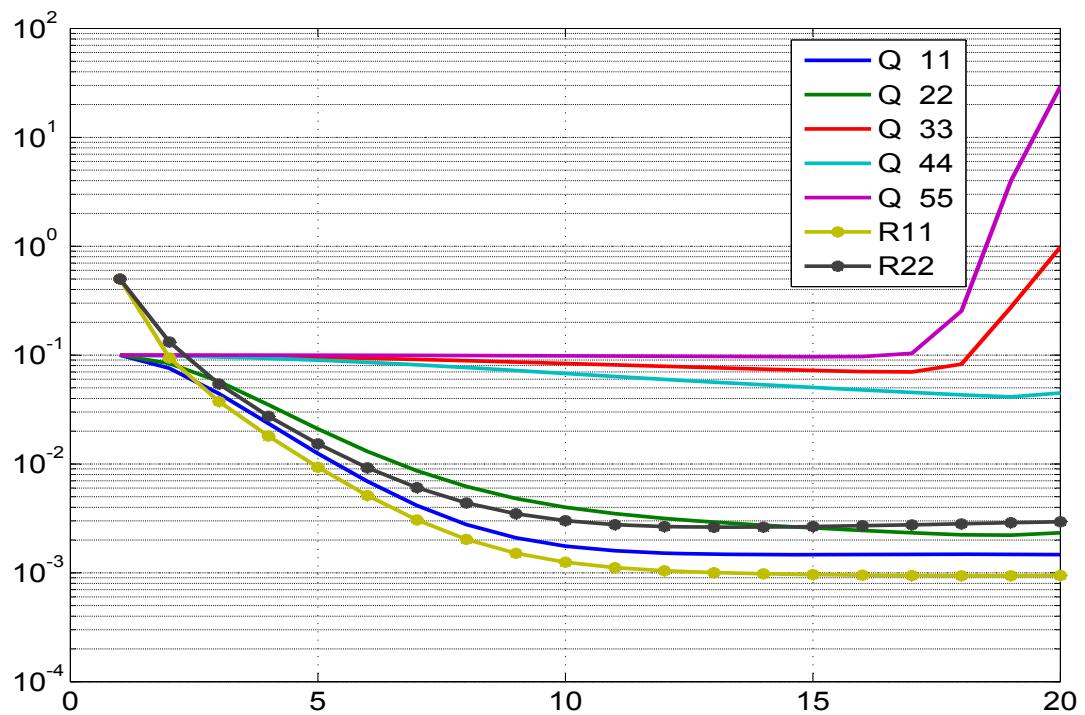
Figure 4.68: Variation of **Q, R** with iterations using the

procedure suggested by Bavdekar et al.[104] (2011)
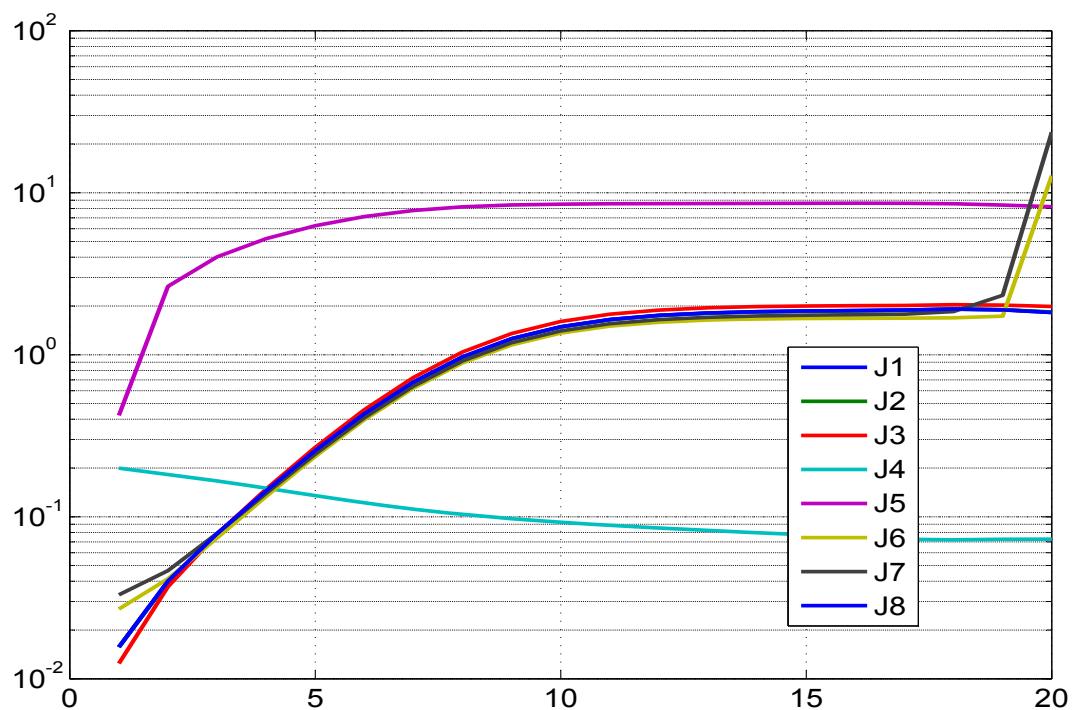
Figure 4.69: Variation of different costs with iterations using the

procedure suggested by Bavdekar et al.[104] (2011)

4.4 Simulated Longitudinal Motion of Aircraft

Consider the Longitudinal motion of an aircraft system excited by control input as shown in Fig. 4.70. The state equations ($n = 4$) for the angle of attack (α), pitch rate (q), pitch angle (θ) and downward velocity (V) are respectively given by

$$\dot{\alpha} = Z_\alpha \alpha + q - 0.0021\theta + Z_{\delta_e}\delta_e$$

$$\dot{q} = m_\alpha \alpha + m_q q + m_{\delta_e}\delta_e$$

$$\dot{\theta} = q$$

$$\dot{V} = 15.67\alpha - 32.16\theta + 8.354\delta_e$$

All the states have zero initial conditions. The measured quantities are indicated with subscript 'm' which includes the angle of attack (α), pitch rate (q), pitch angle (θ), downward velocity (V) and normal acceleration (a_z). The measurement equations (m=5) are given by

$$\alpha_m = \alpha + v_\alpha$$

$$q_m = q + v_q$$

$$\theta_m = \theta + v_\theta$$

$$V_m = V + v_V$$

$$a_{z_m} = U_0 \times (Z_\alpha \alpha + Z_{\delta_e}\delta_e) + v_{a_z}$$

The unknown parameter ($p = 5$) set is $\Theta = (Z_\alpha, m_\alpha, m_q, z_{\delta_e}, m_{\delta_e})^T$ with the true values being $\Theta_{true} = (-0.42, -3.7943, -0.3632, -0.006489, -6.2807)^T$ and $U_0 = 100$. The time indices are not shown for brevity in the state and measurement equations. The numerical values of the noise variances are shown in Table-4.1. All the figures are presented for only one simulation run to prevent cluttering.

4.4.1 Remarks on the Results

We first run the filter assuming $\mathbf{Q} = 0$. It was found that about 20 iterations of the data would suffice. The Fig. 4.71-4.93 refer to the $\mathbf{Q} = 0$ case. The Fig. 4.71-4.75 shows the various parameter estimates and its corresponding variances through cumulative time in-

stants with iterations. The variation of the estimated initial parameters and their variances through iterations are shown in Fig. 4.76. The parameter and the uncertainty reach almost their final estimated values in about 2 and 5 iterations respectively. A similar plot in Fig. 4.77 shows the variation of the estimated measurement noise. The variation of different cost functions (**J1-J5**) through the iterations is shown in Fig. 4.78. The Fig. 4.79-4.83 shows the predicted dynamics, filtered and smoothed estimate at the last iteration. The Fig. 4.84-4.88 show the innovations, filtered residue and smoothed residue together with the square root of their variance ($\pm\sigma$ bound). In the EKF approach most of the quantities are Gaussian or approximated as quasi Gaussian and one would expect all the above quantities are close to being Gaussian and hence around one third of the total sample points to be outside the σ bound. The injected and estimated measurement noise distributions during the final iteration shown in Fig. 4.89-4.93 indicate that they are close to each other. Further even their autocorrelations are ideally expected to be close to the Kronecker delta function which provides confidence in the proposed filter algorithm.

The next step is to process the data with process noise ($\mathbf{Q} > 0$). The Fig. 4.94-4.114 refer to the $\mathbf{Q} > 0$ case. The Fig. 4.94 shows the absolute difference between the iterated and final values with iterations which indicates the accuracy level that one needs and it was found that 100 iterations are required. The variation of the estimated initial parameters and their variances through iterations are shown in Fig. 4.95. The parameter and the uncertainty reach almost their final estimated values in about 5 and 20 iterations respectively. A similar plot in Fig. 4.96 shows the variation of the estimated measurement and process noise. The variation of different cost functions through the iterations are shown in Fig. 4.97. The cost functions **J1-J3** correspond to the number of measurement (m=5) and in presence of process noise, **J6-J8** correspond to the number of states (n=4). The **J4** in absence of process noise corresponds to the trace of the measurement noise $\mathbf{R}$. The **J5** is the negative log likelihood cost whose absolute value is shown in the plot. There is a mismatch in the predicted dynamics and the measurement as seen in Fig. 4.98-4.102 indicating the presence of process noise. The subsequent Fig. 4.103-4.107 correspond to the earlier Fig. 4.84-4.88 of $\mathbf{Q} = 0$ case. The Fig. 4.108-4.112 and Fig. 4.113-4.116 shows respectively the injected and estimated measurement and process noise samples across time during the final iteration.

4.4.2 Longitudinal Motion of Aircraft System Tables (Q = 0)

Table 4.8: Sensitivity Study : ($\mathbf{Q} = 0$) : Longitudinal Aircraft system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**R** ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
Reference adaptive EKF used for non zero process noise case									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: EM-diag	Extremely slow convergence of $\mathbf{Q}$ taking more than 100 iterations.								
Proposed reference adaptive EKF for zero process noise case, $\mathbf{Q}$ is diagonal matrix fixed to 10^{-10} for all states Consistency ratio NR= 1.1320, 1.0367, 0.9417, 1.1147, 0.9322, Spread Factor NR= 3.5427, 0.8768, 6.5086, 233.9268, 1.9185									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0042 1.0017 0.9914 0.7854 0.9975	1.0067 1.0281 0.9550 1.0208 0.9522	1.1653 1.3887 0.9912 1.1302 0.9463	3.5691 0.9673 6.9775 230.7863 2.0266	1.0070 1.0211 0.9736 0.9803 0.9763	1.0020 1.0162 1.0051 1.0029 1.0231	4.9484 4.9527 4.9988 1.0724 -24.8497	0.0702 0.0657 0.0088 0.1532 0.3506	σ_Θ= 0.0119, 0.0326, 0.0182, 0.0116, 0.0898 $SIGMA_{avg}$= 0.0102, 0.0235, 0.0184, 0.0102, 0.0949
$\mathbf{P_0}$: Scaled up-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0056 1.0032 0.9806 0.5947 0.9998	0.9978 1.0907 0.9660 1.0203 0.9330	1.1570 1.9983 1.0561 1.1503 1.7221	3.6190 1.1857 7.2503 234.4404 2.7593	1.0129 1.0859 0.9824 0.9777 0.9640	1.0079 1.0807 1.0141 1.0002 1.0102	4.9263 4.9313 4.9957 1.0730 -24.7175	0.0926 0.0904 0.0107 0.1519 0.3479	Unity CRB ratio
$\mathbf{P_0}$: Scaled up-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	1.0039 1.0011 0.9927 0.7996 0.9986	1.0000 0.9347 0.9576 1.0120 0.8639	1.1456 1.2660 0.9667 1.1483 0.9863	3.5690 1.0036 6.8720 234.9046 2.2701	1.0055 1.0173 0.9737 0.9782 0.9757	1.0004 1.0124 1.0051 1.0007 1.0224	4.9158 4.9192 4.9992 1.0670 -24.7485	0.0824 0.0795 0.0112 0.1547 0.3560	Unity CRB ratio

Table 4.8: Sensitivity Study : ($\mathbf{Q} = 0$) : Longitudinal Aircraft system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{R}$ ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
P$_0$: IIM-[0,0;0,✓] **Q** : 10^{-10}-[✓,0;0,0] **R** : EM-diag	1.0043 1.0005 0.9865 0.9496 1.0000	0.9974 1.0207 0.9433 1.0112 0.9233	1.2091 2.6391 1.0989 1.1716 1.0806	3.6684 1.5370 7.3022 236.3277 2.1750	1.0132 1.0569 0.9890 1.0271 0.9942	1.0082 1.0518 1.0210 1.0507 1.0418	4.9498 4.9532 5.0001 1.0732 -24.6034	0.1284 0.1245 0.0095 0.1525 0.3909	Unity CRB ratio
P$_0$: IIM-diag **Q** : 10^{-10}-[✓,0;0,0] **R** : EM-diag	1.0060 0.9969 0.9929 0.7579 1.0091	0.9856 0.9120 0.9345 0.9957 0.8027	1.1812 2.2915 1.0050 1.1052 1.1132	3.6970 1.5035 7.0722 233.7360 2.6459	1.0227 1.0539 0.9800 0.9781 1.0041	1.0176 1.0489 1.0116 1.0007 1.0522	4.9116 4.9132 5.0017 1.1739 -24.5304	0.1555 0.1497 0.0135 0.2161 0.3722	Unity CRB ratio
P$_0$: IIM-full **Q** : 10^{-10}-[✓,0;0,0] **R** : EM-diag	1.0043 1.0011 0.9893 0.8365 1.0028	0.9872 0.9132 0.9379 0.9989 0.8046	1.1973 2.2625 0.9890 1.1164 1.1235	3.6971 1.4986 7.0395 233.6142 2.5264	1.0187 1.0500 0.9792 0.9781 0.9985	1.0136 1.0450 1.0109 1.0006 1.0463	4.9164 4.9179 5.0061 1.1561 -24.5444	0.1484 0.1430 0.0359 0.1802 0.3839	Unity CRB ratio

Table 4.8: Sensitivity Study : ($Q = 0$) : Longitudinal Aircraft system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**R** ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
P₀ : Smoothed-[0,0;0,✓] **Q** : 10^{-10}-[✓,0;0,0] **R** : EM-diag	1.0023	4.4108	5.0072	2.4008	1.0032	0.9982	4.9971	0.0014	Lower filter CRB than that of the NR
	1.0011	4.4306	4.6519	0.5516	1.0074	1.0026	4.9971	0.0014	
	0.9943	4.1146	4.1359	4.4445	0.9684	0.9997	4.9997	0.0005	
	0.8240	4.4680	4.9952	160.2168	0.9767	0.9991	1.0722	0.1532	
	0.9980	4.0746	4.0162	1.2668	0.9528	0.9984	-25.0816	0.3498	
P₀ : Smoothed-diag **Q** : 10^{-10}-[✓,0;0,0] **R** : EM-diag	0.9962	4.9542	5.5242	2.3346	0.9992	0.9942	4.9935	0.0025	Lower filter CRB than that of the NR
	0.9973	4.6358	5.3890	0.6331	1.0085	1.0037	4.9935	0.0025	
	1.0225	6.4682	6.3287	4.4581	0.9650	0.9962	4.9993	0.0008	
	0.6821	4.5531	5.1484	163.9454	0.9655	0.9878	1.0585	0.1515	
	1.0149	5.8173	6.3994	1.8093	0.9492	0.9946	-25.1038	0.3557	
P₀ : Smoothed-full **Q** : 10^{-10}-[✓,0;0,0] **R** : EM-diag	1.0013	4.3915	4.9936	2.4231	0.9984	0.9935	4.9947	0.0017	Lower filter CRB than that of the NR
	1.0005	3.8644	4.4205	0.5917	0.9996	0.9948	4.9947	0.0017	
	0.9947	4.1147	4.0574	4.3767	0.9637	0.9949	4.9995	0.0008	
	0.8386	4.4230	5.0333	163.9541	0.9650	0.9872	1.0592	0.1520	
	0.9994	3.4979	4.0582	1.4796	0.9486	0.9941	-25.1165	0.3575	

Table 4.8: Sensitivity Study : ($\mathbf{Q} = 0$) : Longitudinal Aircraft system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{R}$ ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag*	1.0042 1.0017 0.9914 0.7846 0.9975	1.0172 1.0366 0.9636 1.0324 0.9598	1.1779 1.3998 1.0002 1.1434 0.9534	3.5512 0.9631 6.9442 229.5023 2.0168	1.0033 1.0025 0.9697 0.9799 0.9539	0.9982 0.9977 1.0010 1.0025 0.9996	4.9982 5.0025 5.0495 1.0724 -24.8501	0.0709 0.0664 0.0100 0.1532 0.3506	*Using smoothed residue ignoring 2nd order terms in Eq-3.7
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag*	1.0042 1.0017 0.9916 0.7845 0.9976	1.0461 1.0655 0.9935 1.0621 0.9906	1.2128 1.4342 1.0285 1.1764 0.9823	3.5027 0.9483 6.8233 226.4157 1.9777	0.9934 0.9268 0.9613 0.9796 0.9008	0.9884 0.9224 0.9924 1.0022 0.9439	5.1606 5.1648 5.2151 1.0724 -24.8442	0.0557 0.0432 0.0963 0.1532 0.3503	*Using filtered residue ignoring 2nd order terms in Eq-3.19
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag*	1.0042 1.0017 0.9912 0.7908 0.9973	1.0026 1.0251 0.9509 1.0166 0.9478	1.1714 1.3874 0.9865 1.1284 0.9421	3.5856 0.9698 7.0068 231.4841 2.0412	1.0288 1.0398 0.9924 0.9920 0.9916	1.0237 1.0348 1.0245 1.0148 1.0391	4.9580 4.9621 5.0094 1.0724 -24.7982	0.7232 0.7200 0.7374 0.1532 0.3521	*Using innovations ignoring 2nd order terms in Eq-3.20
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag*	1.0043 1.0017 0.9915 0.7843 0.9975	1.0163 1.0348 0.9616 1.0321 0.9571	1.1790 1.4012 0.9986 1.1445 0.9501	3.5566 0.9654 6.9529 229.6816 2.0195	1.0063 1.0111 0.9700 0.9776 0.9544	1.0013 1.0063 1.0013 1.0001 1.0001	4.9906 4.9949 5.0417 1.0724 -24.8494	0.1029 0.1019 0.0876 0.1532 0.3504	*Using dynamical residue as per Eq-3.18
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: MT-diag	1.0042 1.0017 0.9912 0.7908 0.9974	1.0062 1.0292 0.9551 1.0200 0.9523	1.1759 1.3916 0.9902 1.1321 0.9462	3.5796 0.9674 6.9885 231.0834 2.0351	1.0255 1.0277 0.9890 0.9914 0.9851	1.0204 1.0228 1.0210 1.0142 1.0323	4.9843 4.9883 5.0360 1.0724 -24.7981	0.7297 0.7265 0.7441 0.1532 0.3523	Using full expression in Eq-3.20
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: MS-diag	1.0042 1.0017 0.9916 0.7862 0.9975	1.0102 1.0320 0.9588 1.0244 0.9562	1.1694 1.3967 0.9944 1.1341 0.9507	3.5613 0.9666 6.9589 230.4433 2.0211	1.0037 1.0116 0.9726 0.9801 0.9694	0.9987 1.0068 1.0040 1.0027 1.0158	4.9709 4.9752 5.0219 1.0724 -24.8492	0.0509 0.0368 0.0889 0.1532 0.3505	Using full expression in Eq-3.19

4.4.3 Longitudinal Motion of Aircraft System Tables (Q > 0)

Table 4.9: Sensitivity Study : ($\mathbf{Q} > 0$) : Longitudinal Aircraft system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of $\mathbf{J1}$-$\mathbf{J8}$	σ of $\mathbf{J1}$-$\mathbf{J8}$	Remarks
Proposed reference adaptive EKF for non zero process noise								
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: EM-diag EKF	1.0014 0.9890 1.0643 1.1928 1.0248	0.9966 1.0106 1.0060 0.9964	1.0047 1.0752 0.9955 1.1647 0.9464	4.4845 12.7867 55.2270 322.6663 9.5713	1.0100 0.9565 0.9802 1.0048 0.9730	0.9997 1.1642 0.9912 0.9892	4.9322 5.2814 5.0007 97.8400 -21.4188 3.9676 3.9680 3.9527	0.1543 1.1192 0.0210 120.7487 0.3619 0.0809 0.0806 0.1038	σ_Θ= 0.0140 , 0.3692, 0.1488 , 0.0164, 0.4272 $SIGMA_{avg}$= 0.0139 , 0.3434, 0.1495, 0.0141, 0.4514
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: EM-diag IEKF	1.0516 1.0157 1.0520 0.9003 1.0245	1.0279 1.0108 1.0059 0.9964	1.4154 1.0607 0.9880 1.1382 0.9398	6.8077 13.1297 54.9221 330.3530 9.5744	1.0436 0.9586 0.9801 1.0048 0.9629	0.9427 1.1603 0.9914 0.9889	4.9348 4.9223 5.0021 97.8200 -21.4066 3.9651 3.9656 3.9593	0.1536 0.1537 0.0226 120.5984 0.3589 0.0687 0.0685 0.1041	σ_Θ= 0.0204 , 0.3711 , 0.1474 , 0.0167 , 0.4241 $SIGMA_{avg}$= 0.0144 , 0.3499 , 0.1492 , 0.0147, 0.4513
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: EM-diag SOKF	1.0042 0.9911 1.0733 1.2108 1.0253	0.9942 1.0107 1.0060 0.9964	0.9971 1.0739 0.9977 1.1310 0.9469	4.5369 12.8096 55.4131 326.3170 9.5916	1.0085 0.9565 0.9801 1.0048 0.9909	0.9847 1.1649 0.9912 0.9892	4.9220 5.3169 5.0021 97.8363 -21.4173 4.0213 4.0215 3.9505	0.1549 1.3630 0.0232 120.7395 0.3627 0.4561 0.4550 0.1039	σ_Θ= 0.0140, 0.3697, 0.1493, 0.0163, 0.4278 $SIGMA_{avg}$= 0.0141 , 0.3442, 0.1496, 0.0144, 0.4518

Table 4.9: Sensitivity Study : ($\mathbf{Q} > 0$) : Longitudinal Aircraft system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**Q** ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
$\mathbf{P_0}$: Scaled up-[0,0;0,$\checkmark$] $\mathbf{Q}$: EM-[$\checkmark$,0;0,0] **R** : Known IEKF	1.0483 1.0108 1.0387 0.8982 1.0245	1.0196 0.9786 0.9964 0.9908	1.3351 1.0833 0.9738 1.1246 0.9305	6.4478 12.9885 54.1652 326.9580 9.5244	NA	0.9569 1.1234 0.9934 1.0086	4.9662 4.9670 5.0405 97.7912 -21.3807 3.9569 3.9572 3.9745	0.3800 0.4077 0.5043 120.6386 0.3628 0.0306 0.0304 0.2672	**R** is known
$\mathbf{P_0}$: Zero $\mathbf{Q}$: EM-diag **R** : EM-diag IEKF	NA	0.9997 1.0040 1.0066 0.9964	NA	NA	1.0113 1.0128 0.9813 1.0048 0.9678	1.0064 1.0327 0.9868 0.9893	4.9980 5.0022 5.0000 97.7558 -21.6081 4.0025 4.0027 4.0034	0.1402 0.1378 0.0132 120.7395 0.3570 0.0277 0.0277 0.0823	Θ is known
$\mathbf{P_0}$: Zero $\mathbf{Q}$: EM-diag **R** : Known IEKF	NA	0.9930 0.9906 0.9969 0.9908	NA	NA	NA	1.0117 1.0431 0.9913 1.0089	5.0271 5.0360 5.0401 97.7558 -21.5797 4.0022 4.0022 4.0286	0.3665 0.3956 0.4920 120.7395 0.3617 0.0254 0.0254 0.2492	Θ and **R** is known

Table 4.9: Sensitivity Study : $\mathbf{Q} > 0$: Longitudinal Aircraft system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
MS method (Mohamed and Schwarz)								
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MS-diag $\mathbf{R}$: MS-diag	**R** can converge to a very large value with far away initial conditions. There is no proper convergence even if Θ is known.								
MS method with **Q** being estimated using only last N/2 samples								
$\mathbf{P_0}$: Zero $\mathbf{Q}$: MS-diag $\mathbf{R}$: Known IEKF	NA	1.0152 1.0082 0.9854 0.9851	NA	NA	NA	0.9306 0.9767 1.1479 1.1491	5.1784 5.1293 5.0846 97.7558 -21.4882 4.3548 4.3546 4.1503	0.7942 0.7294 0.5682 120.7395 0.3742 1.2636 1.2629 0.6592	$\mathbf{R}, \Theta$ is known
MS method assuming **R** is known and **Q** is estimated using only last N/2 samples								
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MS-[✓,0;0,0] $\mathbf{R}$: Known IEKF	1.0496 1.0120 1.0207 0.8931 1.0217	1.0438 1.0043 0.9853 0.9851	1.3070 1.1829 1.0406 1.1690 0.9896	6.5109 12.7676 51.9801 328.3617 9.2482	NA	0.8745 0.9949 1.1489 1.1493	5.1386 5.0867 5.1101 97.7763 -21.2961 4.3513 4.3515 4.1159	0.7844 0.7295 0.5842 120.6334 0.3703 1.3097 1.3093 0.6521	Stable operation
Gemsons approach using MS method								
$\mathbf{P_0}$: IIM-Full $\mathbf{Q}$: MS-[✓,0;0,0] $\mathbf{R}$: 0.9*True $\mathbf{R}$	1.032 0.9991 1.0183 1.1305 1.0511	1.0491 1.0211 0.9991 0.9874	1.2785 1.0331 0.9433 1.1835 0.8350	5.8240 11.595 51.881 325.941 9.4834	0.9001 0.8901 0.9125 0.9305 0.9261	1.065 1.1782 1.233 1.2914	5.115 5.214 5.3262 96.5612 -21.2033 4.0755 4.0756 4.0711	0.7567 0.7601 0.7511 119.1911 0.3640 1.0123 1.0122 0.6110	$\delta J = 10^-6$, with one loop of $\mathbf{R}$ estimation

Table 4.9: Sensitivity Study : $\mathbf{Q} > 0$: Longitudinal Aircraft system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**Q** ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
MT method (Myers and Tapley)									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MT-diag $\mathbf{R}$: MT-diag	**Q** and **R** cannot be estimated simultaneously . There is no proper convergence even if Θ is known.								
MT method with **Q** being estimated using only last N/2 samples									
$\mathbf{P_0}$: Zero $\mathbf{Q}$: MT-diag $\mathbf{R}$: Known IEKF	NA	0.9930 0.9906 0.9969 0.9908	NA	NA	NA	1.0117 1.0431 0.9913 1.0089	5.0271 5.0360 5.0401 97.7558 -21.5797 4.0022 4.0022 4.0286	0.3665 0.3956 0.4920 120.7395 0.3617 0.0254 0.0254 0.2492	**R**, Θ is known
MT method assuming **R** is known and **Q** is estimated using only last N/2 samples									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MT-[✓,0;0,0] $\mathbf{R}$: Known IEKF	1.0483 1.0105 1.0268 0.9000 1.0227	1.0256 0.9975 0.9981 0.9857	1.2950 1.1640 1.0281 1.1549 0.9734	6.4271 12.8682 52.7727 329.5367 9.3336	NA	0.9764 1.0329 1.0819 1.1350	5.0554 5.0509 5.0801 97.7781 -21.3068 4.2598 4.2599 4.0573	0.7582 0.7099 0.5780 120.6202 0.3724 1.2297 1.2292 0.6253	**R** is known
Gemsons approach using MT method									
$\mathbf{P_0}$: IIM-full $\mathbf{Q}$: MT-[✓,0;0,0] $\mathbf{R}$: 0.9*True **R**	1.0409 0.9992 1.0173 1.1313 1.0538	1.0399 1.0202 0.9982 0.9850	1.2786 1.0330 0.9430 1.1827 0.8347	5.8238 11.5975 51.8872 325.9407 9.4833	0.9006 0.8914 0.9320 0.9300 0.9262	1.0608 1.1781 1.2229 1.2964	5.1140 5.2137 5.3295 96.5690 -21.2019 4.0772 4.0770 4.0720	0.7574 0.7668 0.7536 119.1909 0.3641 1.0157 1.0160 0.6108	$\delta J = 10^{-}6$, with one loop of **R** estimation

4.4.4 Longitudinal Motion of Aircraft System Figures (Q = 0)
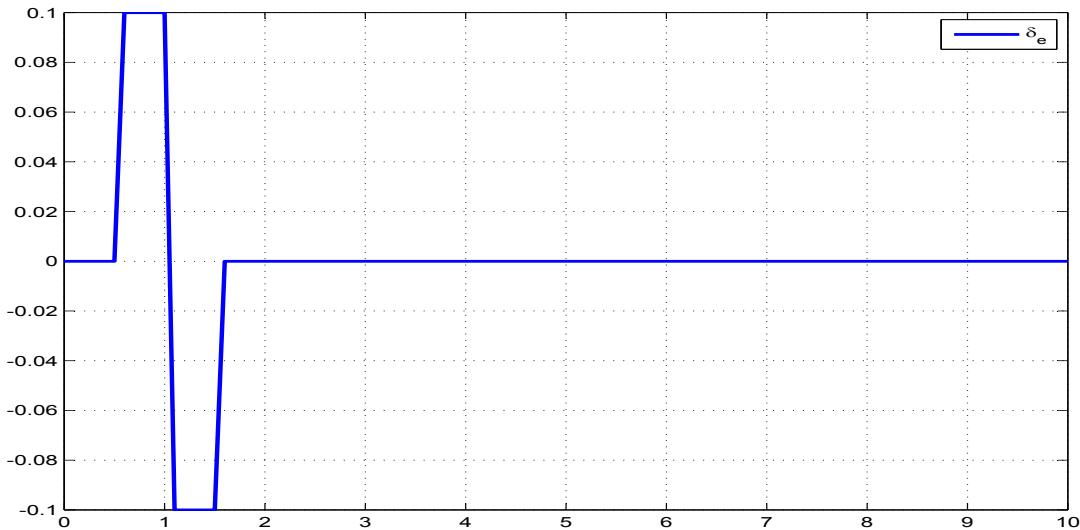
Figure 4.70: Control input of simulated Longitudinal aircraft system
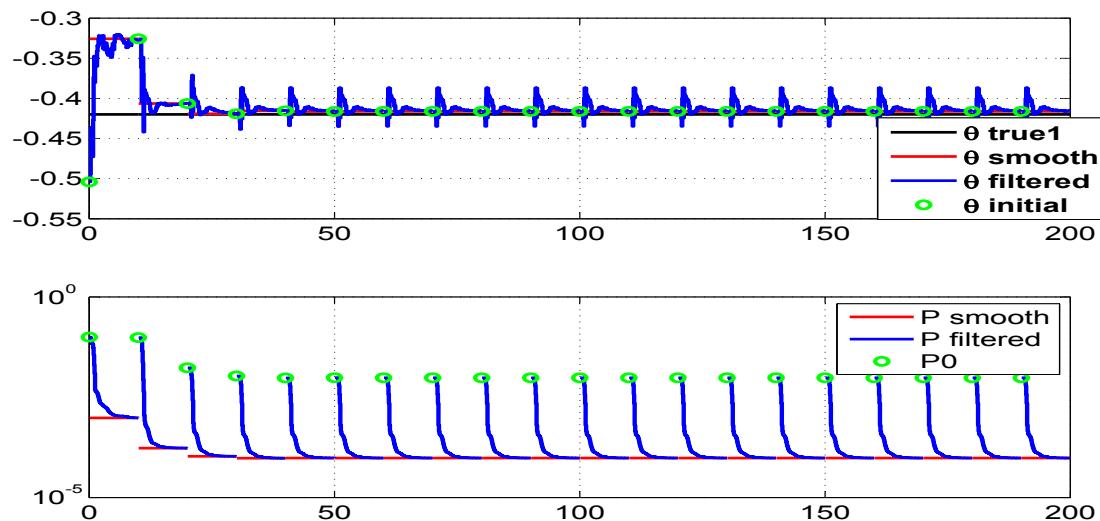
Figure 4.71: The variation of parameter estimate 1 and their filtered and

smoothed covariances through (with the time cumulatively) the iterations
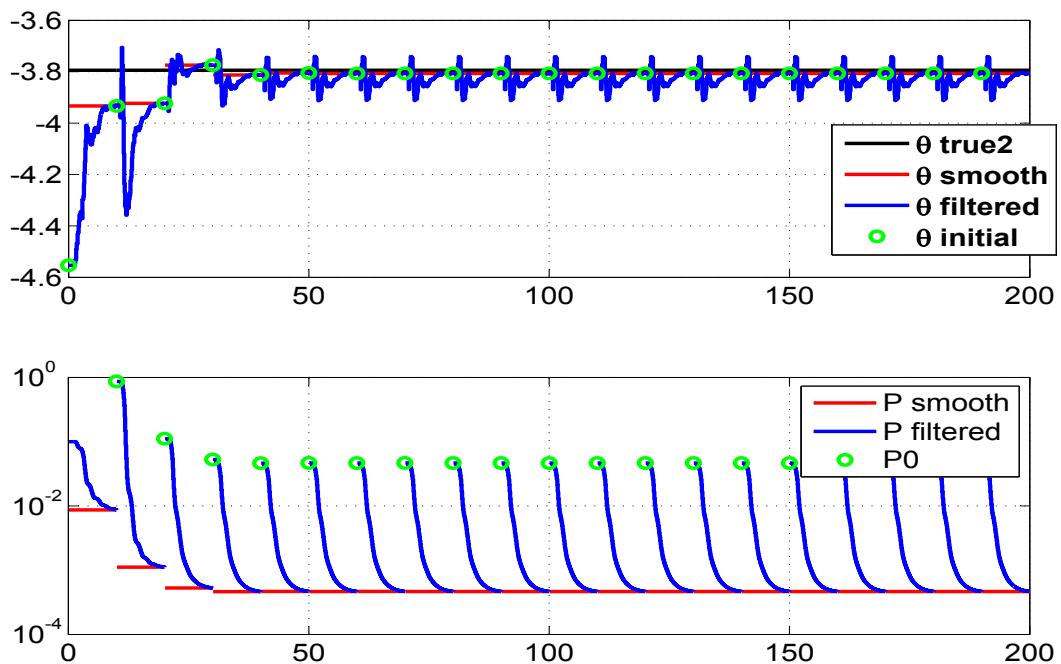
Figure 4.72: The variation of parameter estimate 2 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
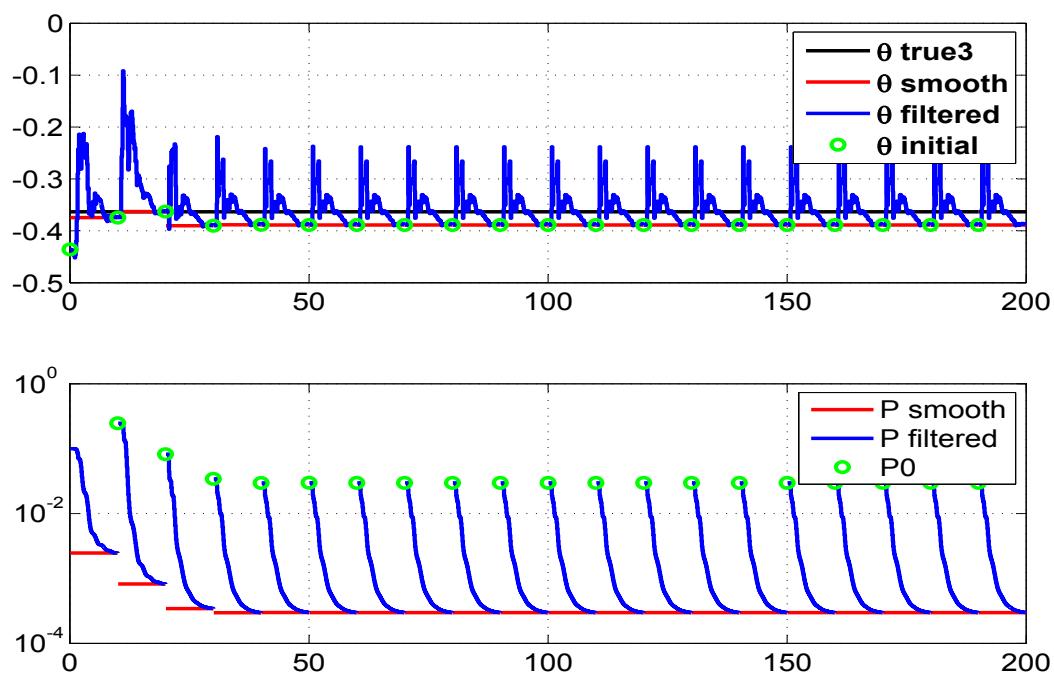
Figure 4.73: The variation of parameter estimate 3 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
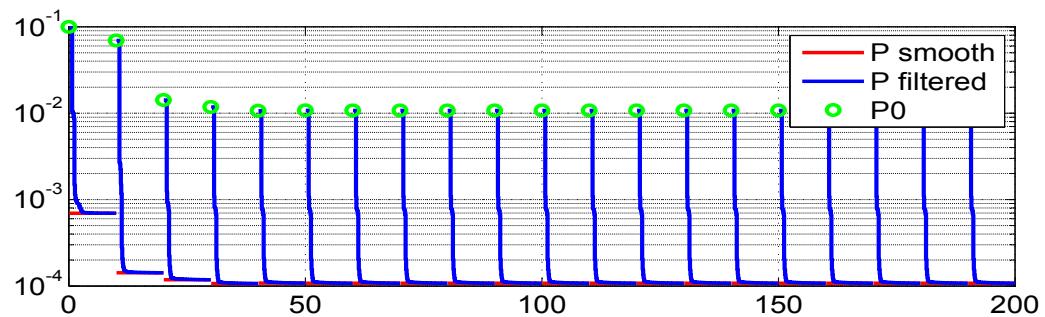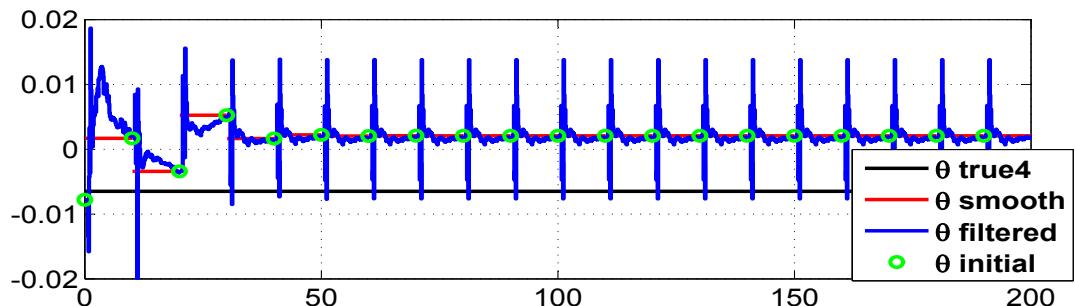
Figure 4.74: The variation of parameter estimate 4 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
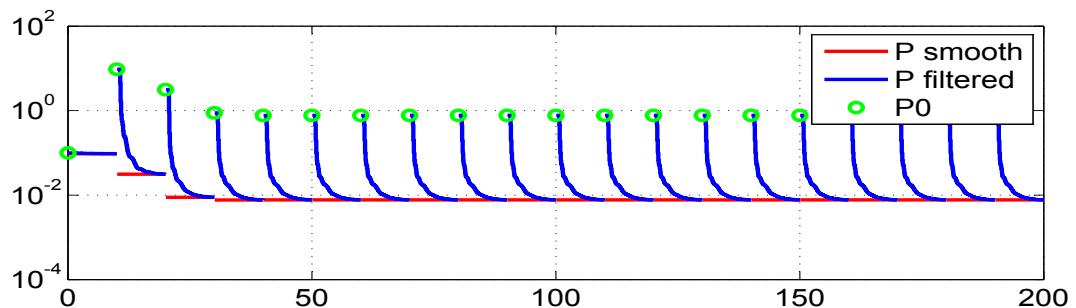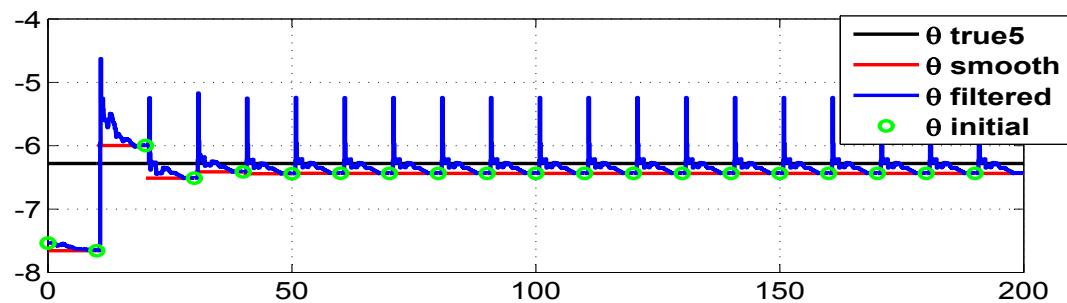
Figure 4.75: The variation of parameter estimate 5 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
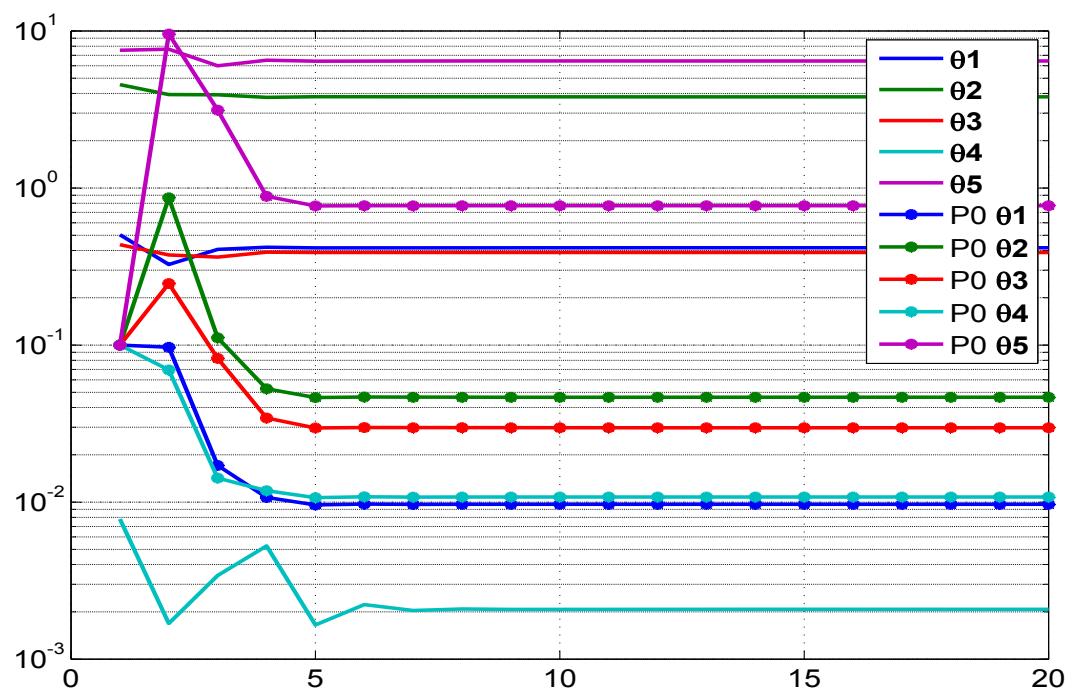
Figure 4.76: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
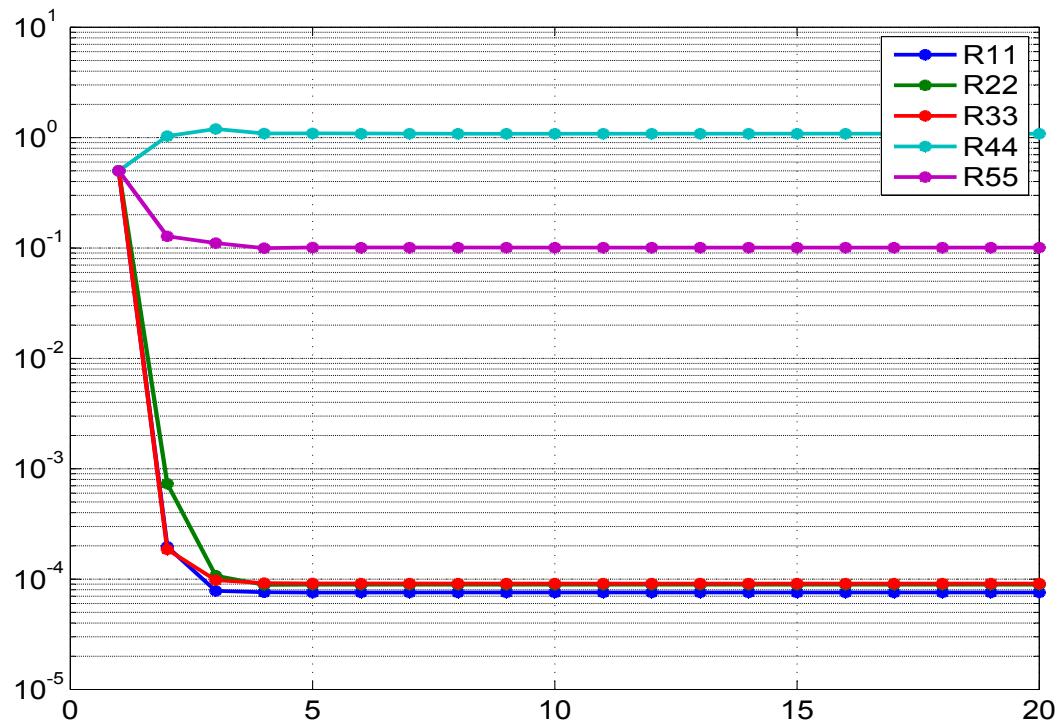
Figure 4.77: Variation of $\mathbf{R}$ with iterations

146
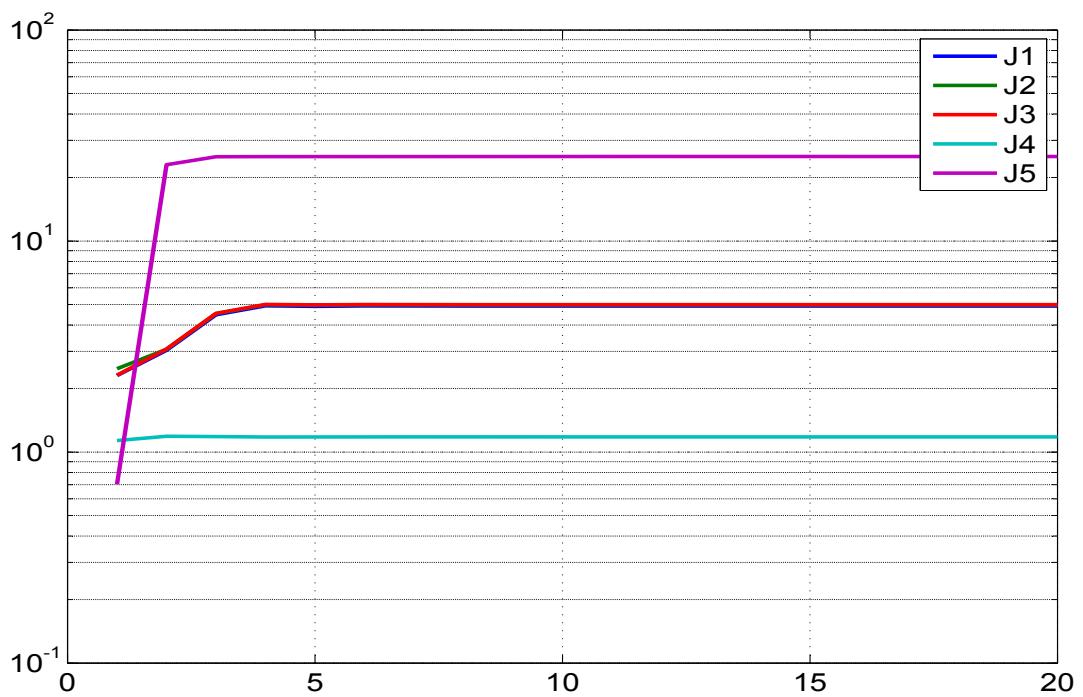
Figure 4.78: Variation of different costs (**J1-J5**) with iterations
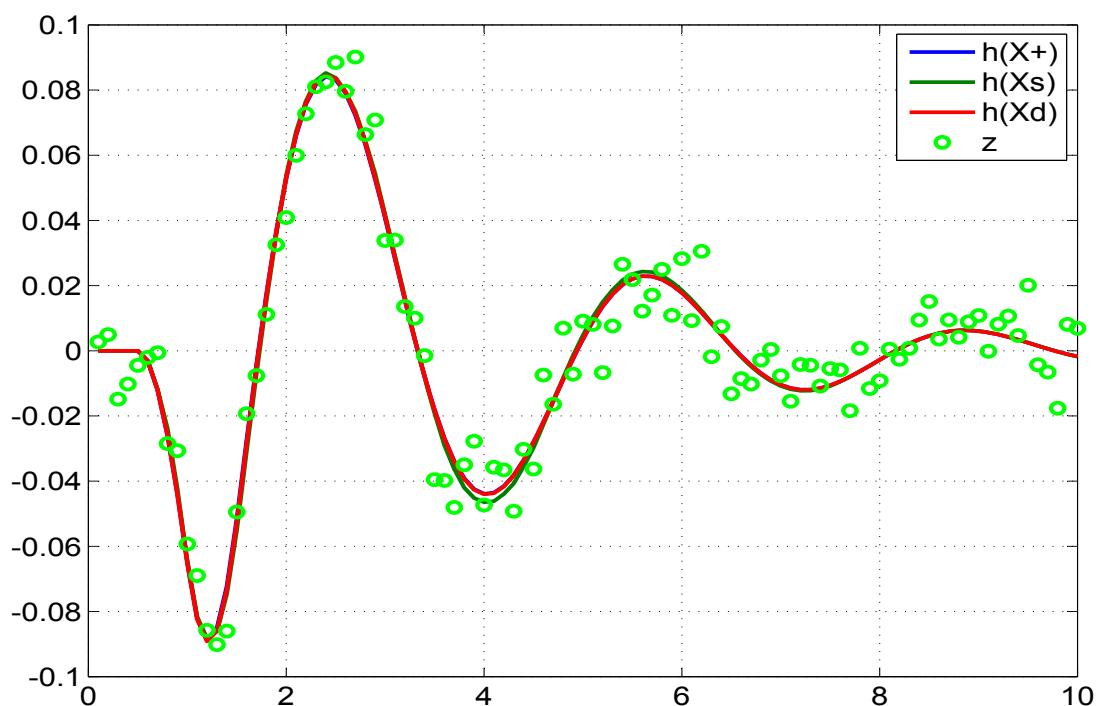
Figure 4.79: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1
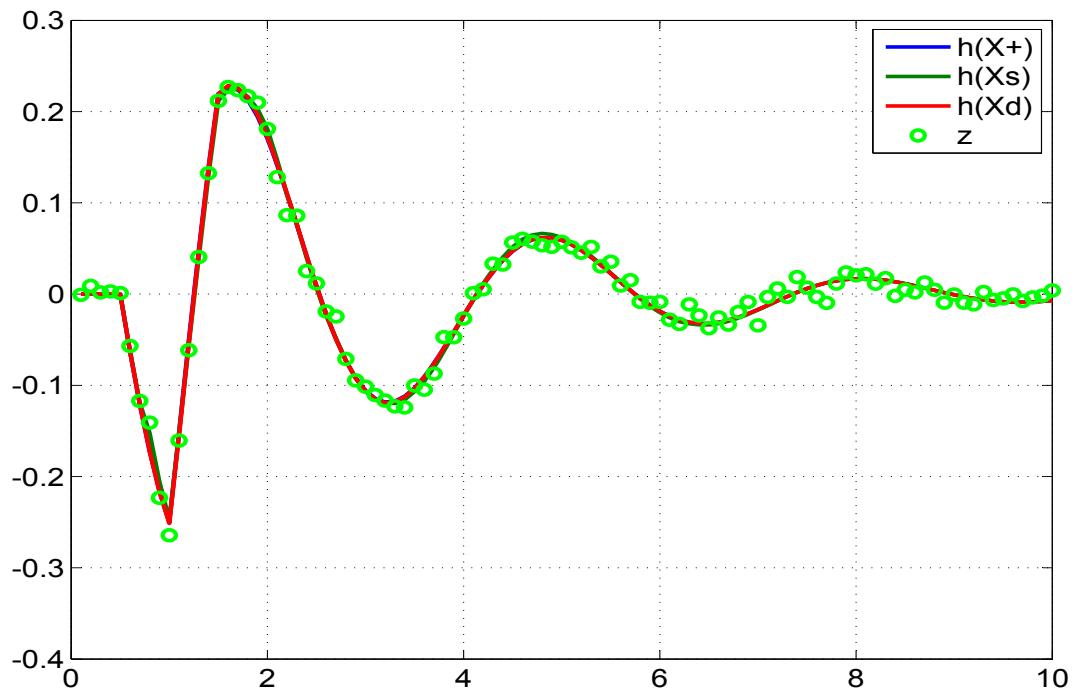
Figure 4.80: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 2
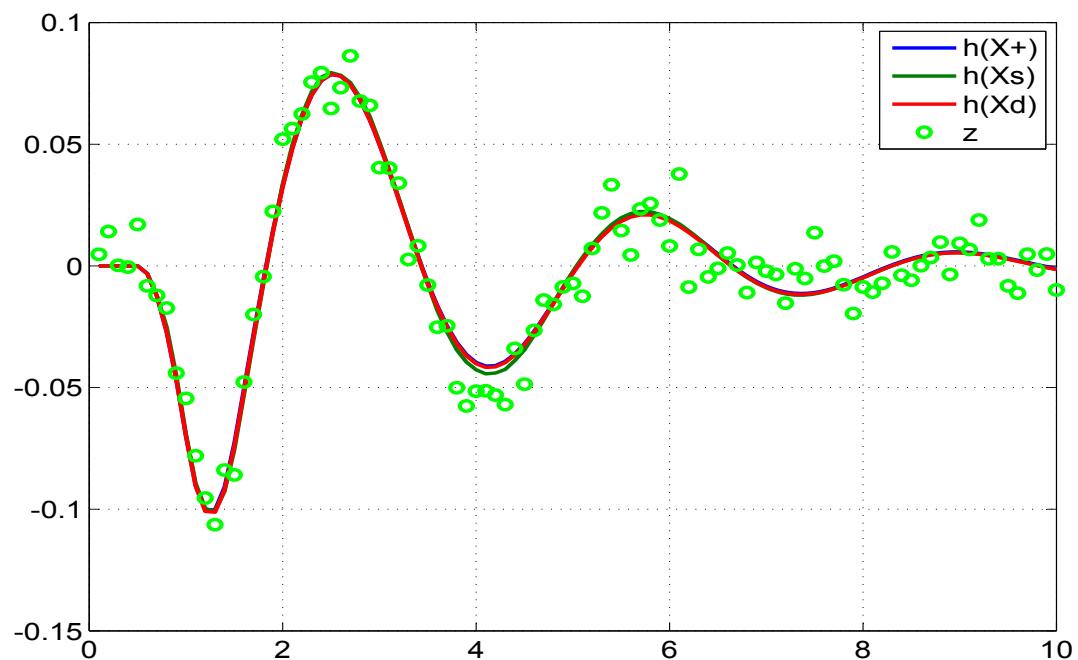
Figure 4.81: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 3
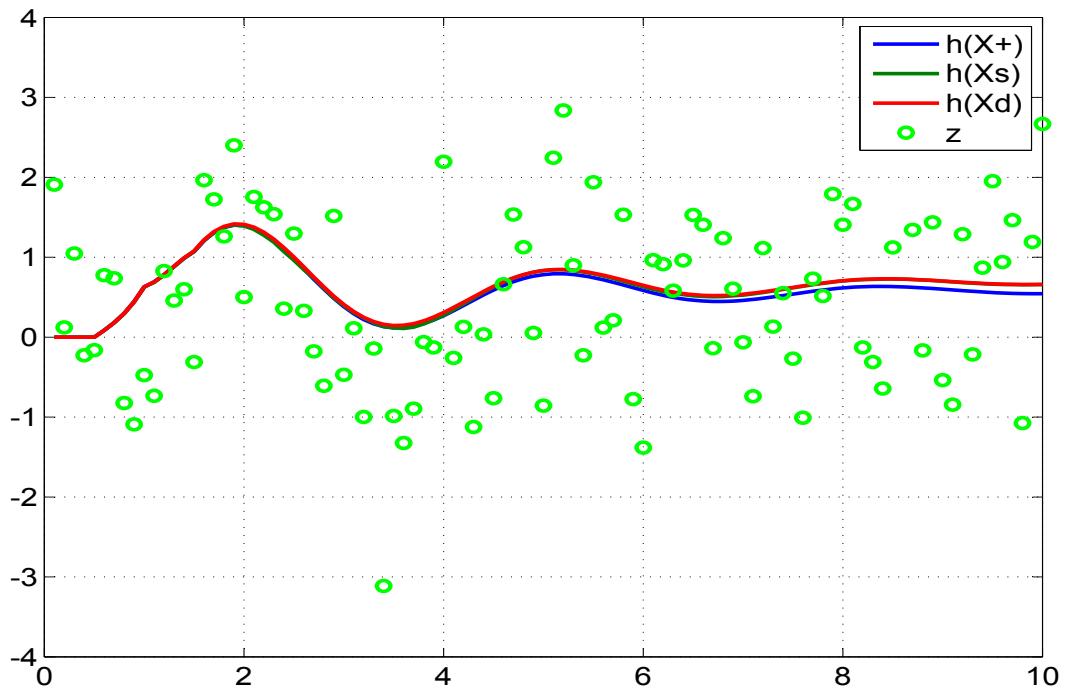
Figure 4.82: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 4
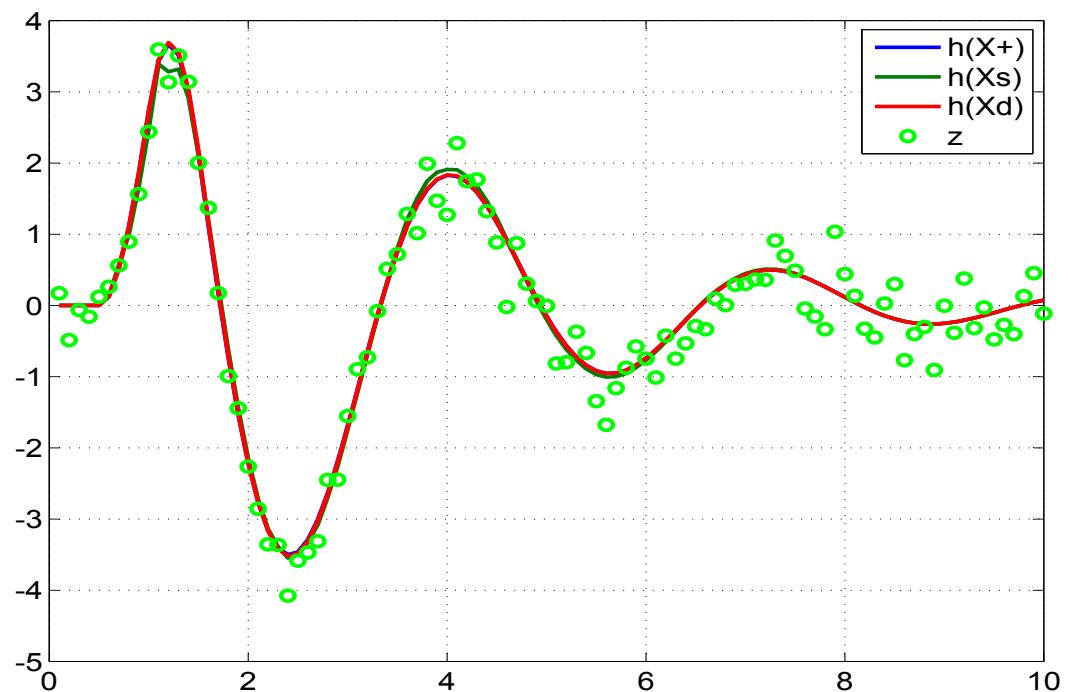
Figure 4.83: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 5

149
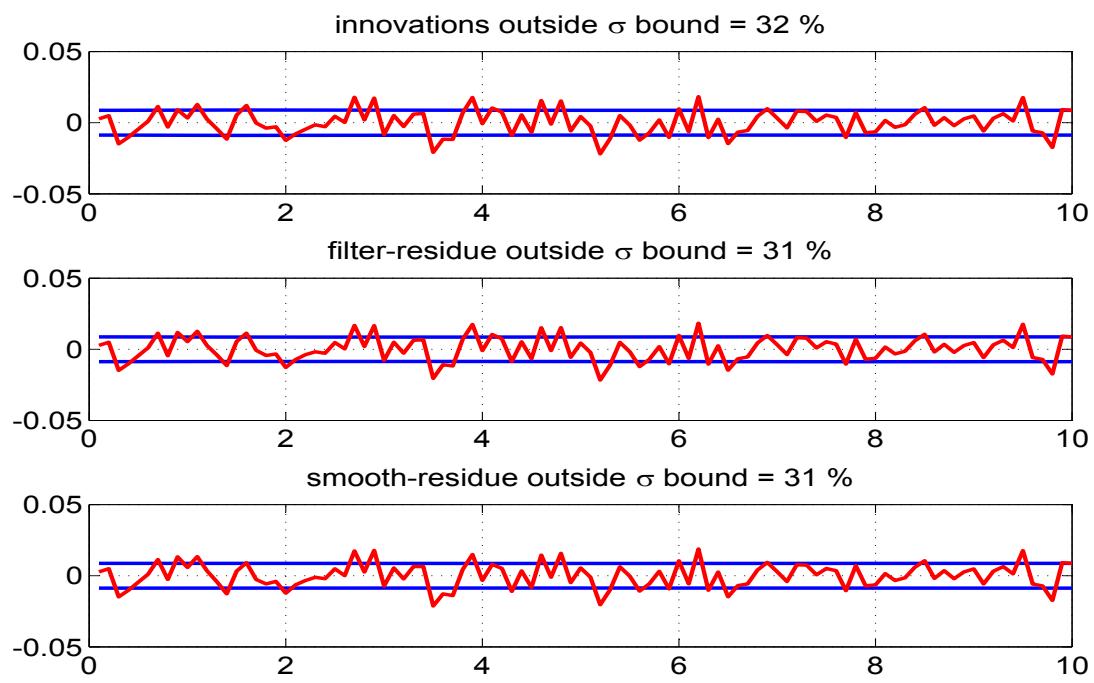
Figure 4.84: The innovations, filtered residue and smoothed residue

corresponding to measurement 1
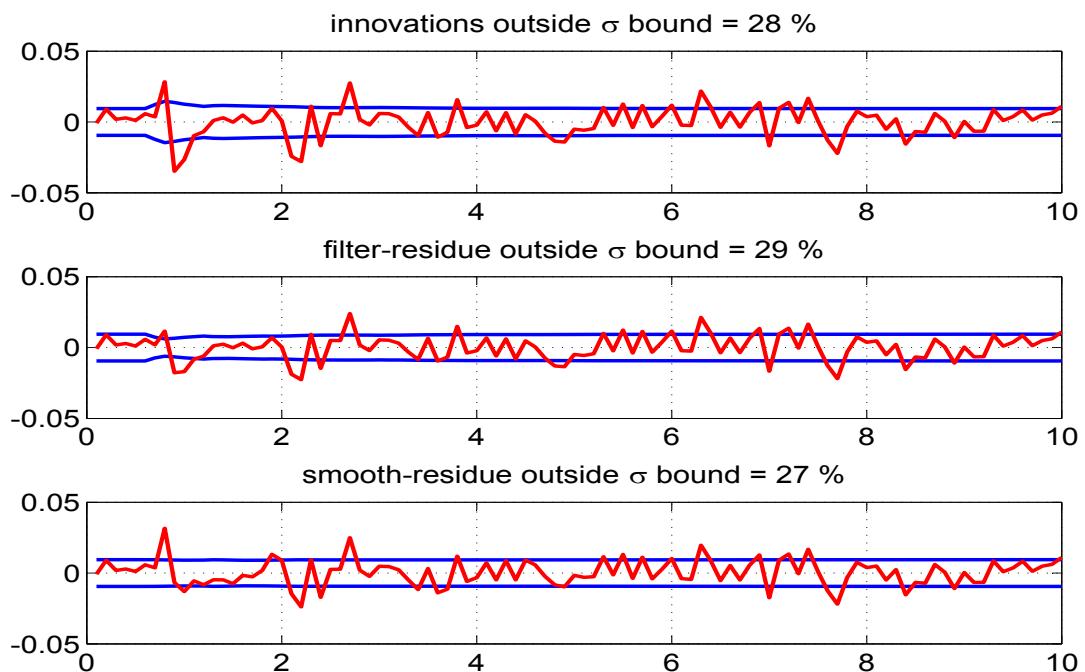
Figure 4.85: The innovations, filtered residue and smoothed residue

corresponding to measurement 2
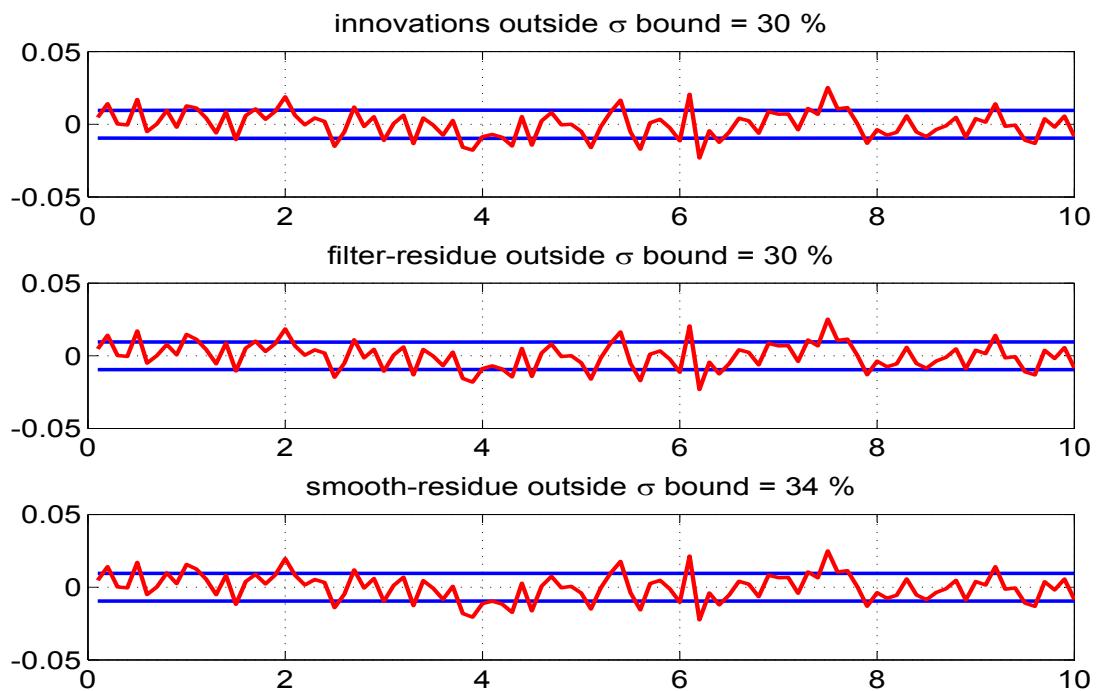
Figure 4.86: The innovations, filtered residue and smoothed residue

corresponding to measurement 3
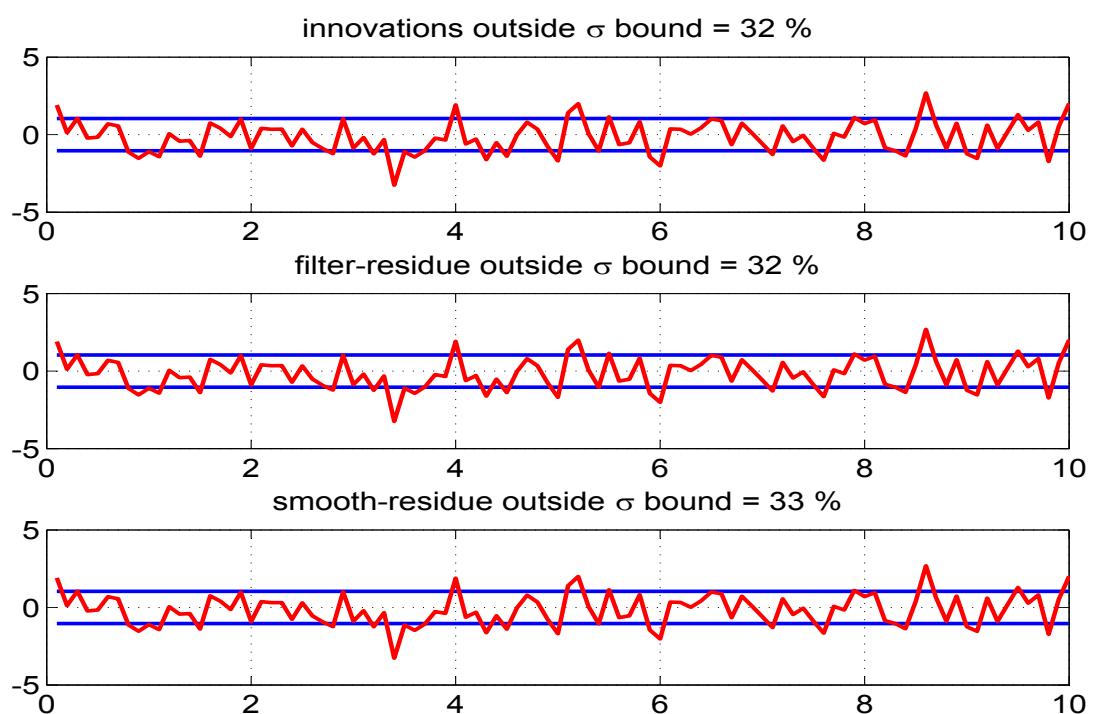
Figure 4.87: The innovations, filtered residue and smoothed residue

corresponding to measurement 4
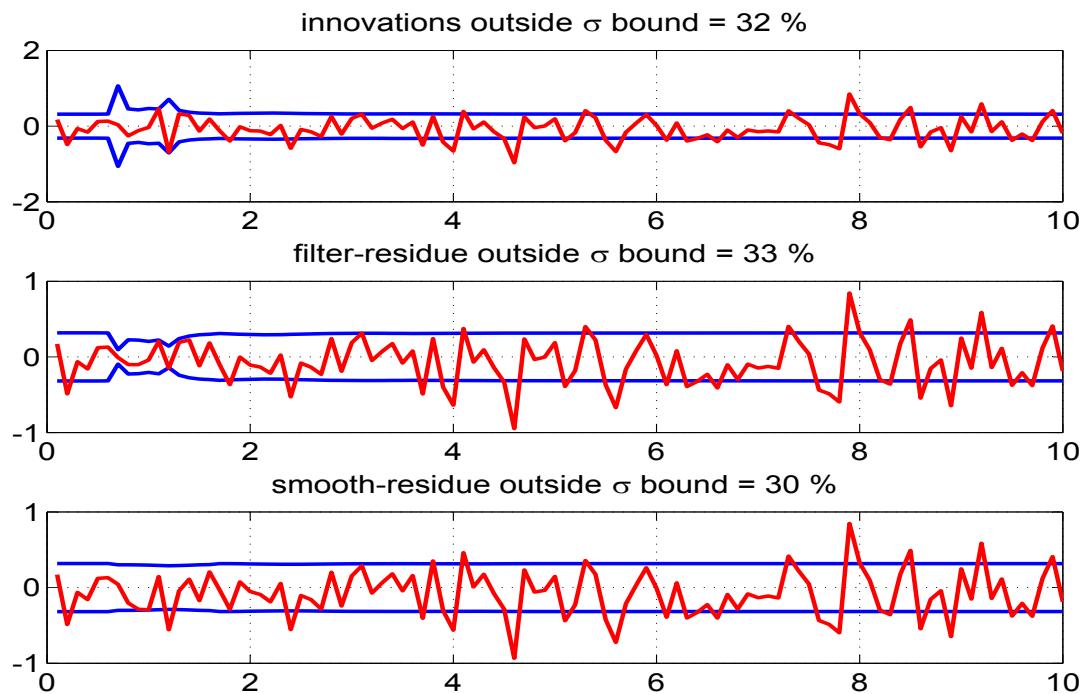
Figure 4.88: The innovations, filtered residue and smoothed residue

corresponding to measurement 5
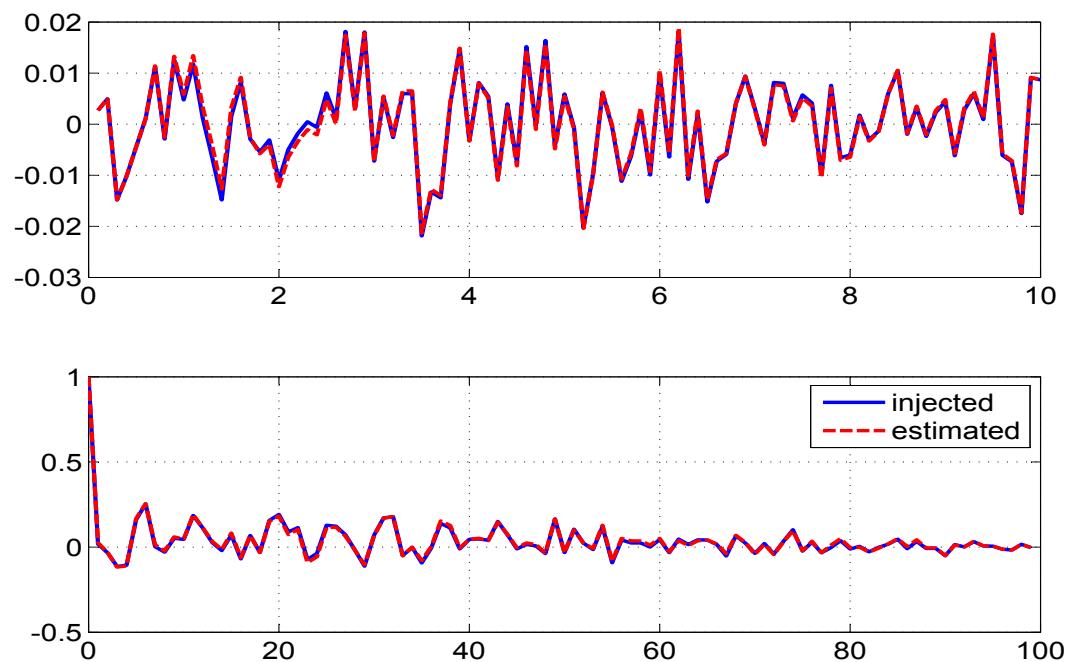
Figure 4.89: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 1
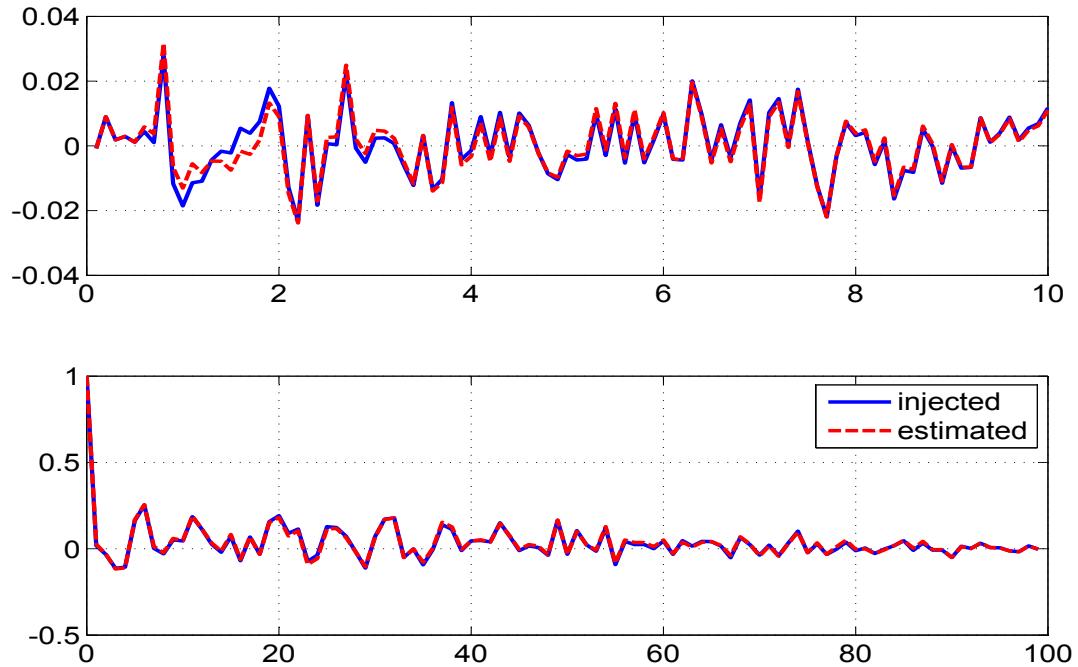
Figure 4.90: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 2
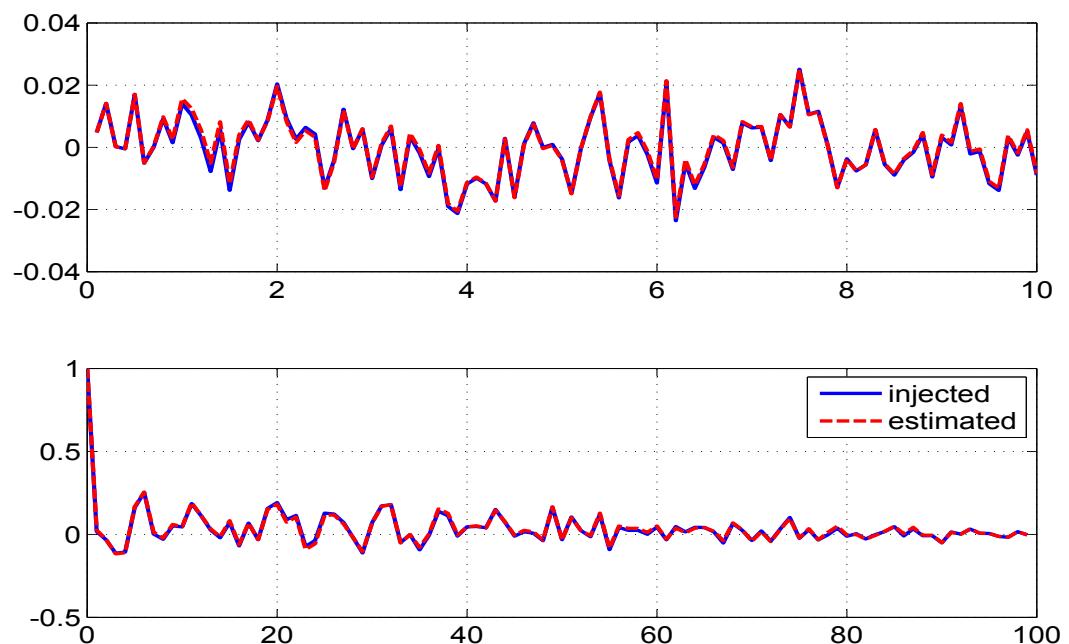
Figure 4.91: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 3
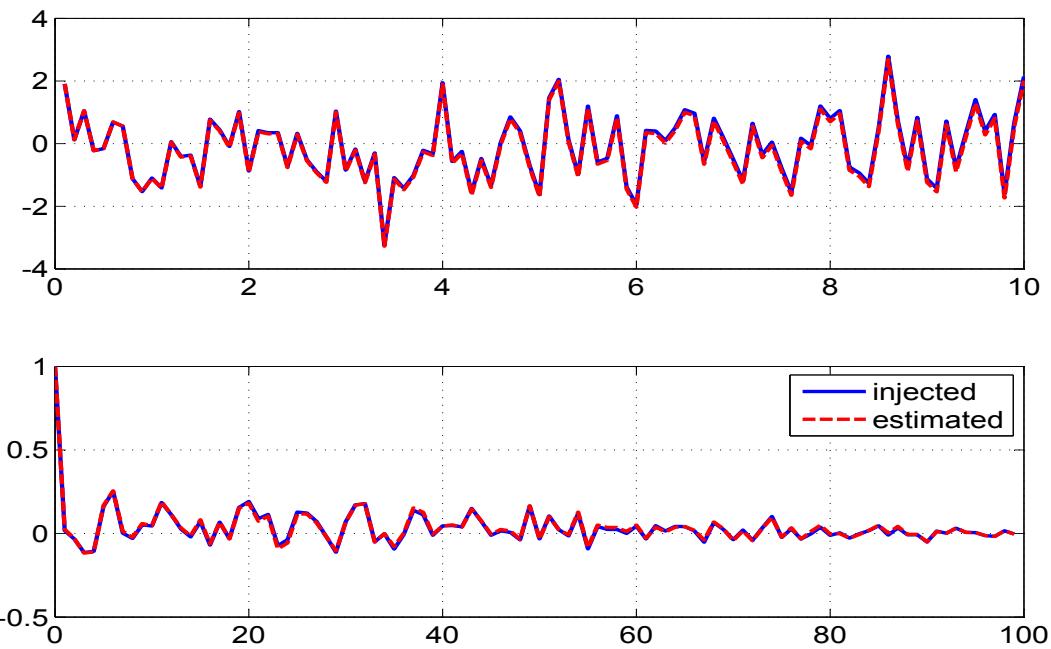
Figure 4.92: Time variation of injected and estimated measurement noise (top) and their autocorrelation (bottom) for measurement 4
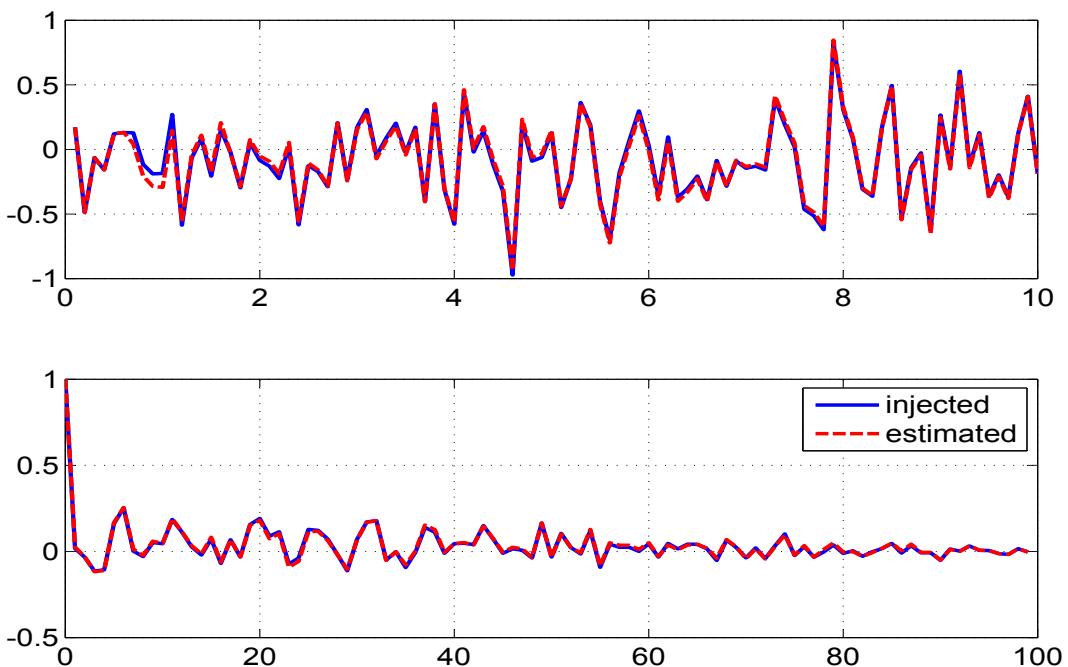
Figure 4.93: Time variation of injected and estimated measurement noise (top) and their autocorrelation (bottom) for measurement 5

154

4.4.5 Longitudinal Motion of Aircraft System Figures (Q > 0)
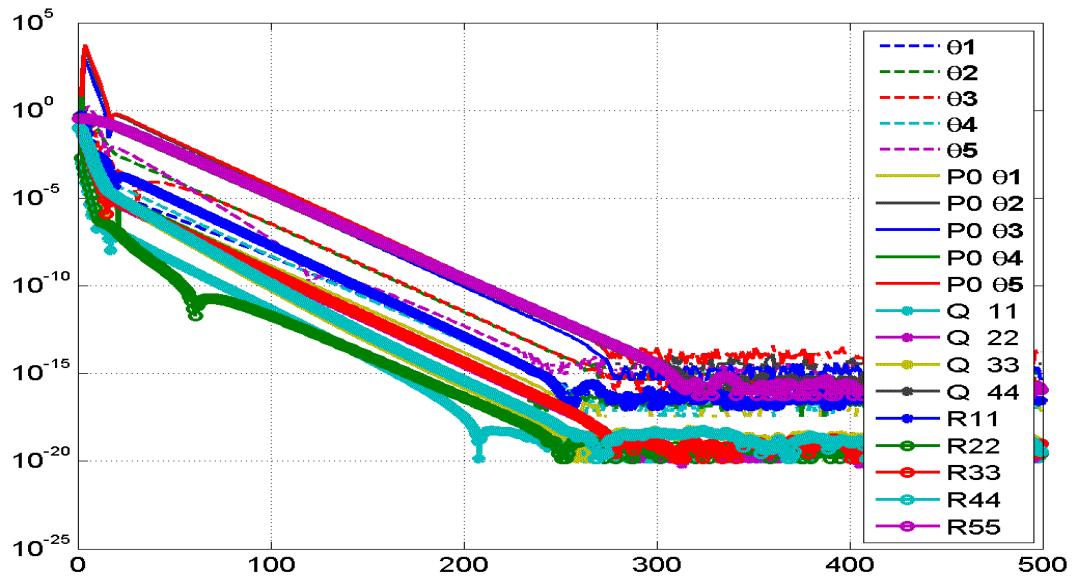
Figure 4.94: The absolute difference between the iterated and final values with 500 iterations
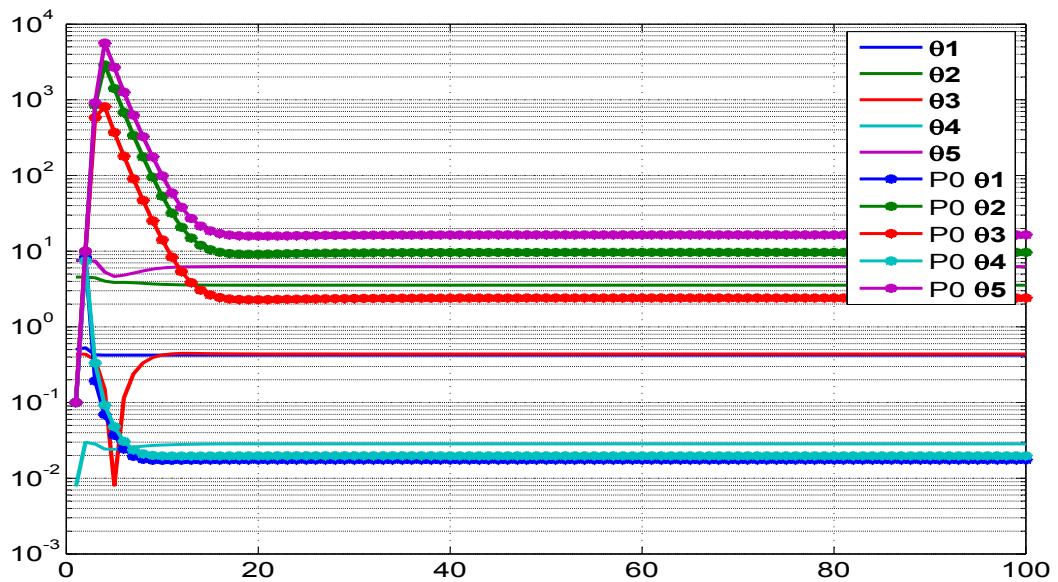
Figure 4.95: Variation of parameter and its initial covariance (P0) with iterations
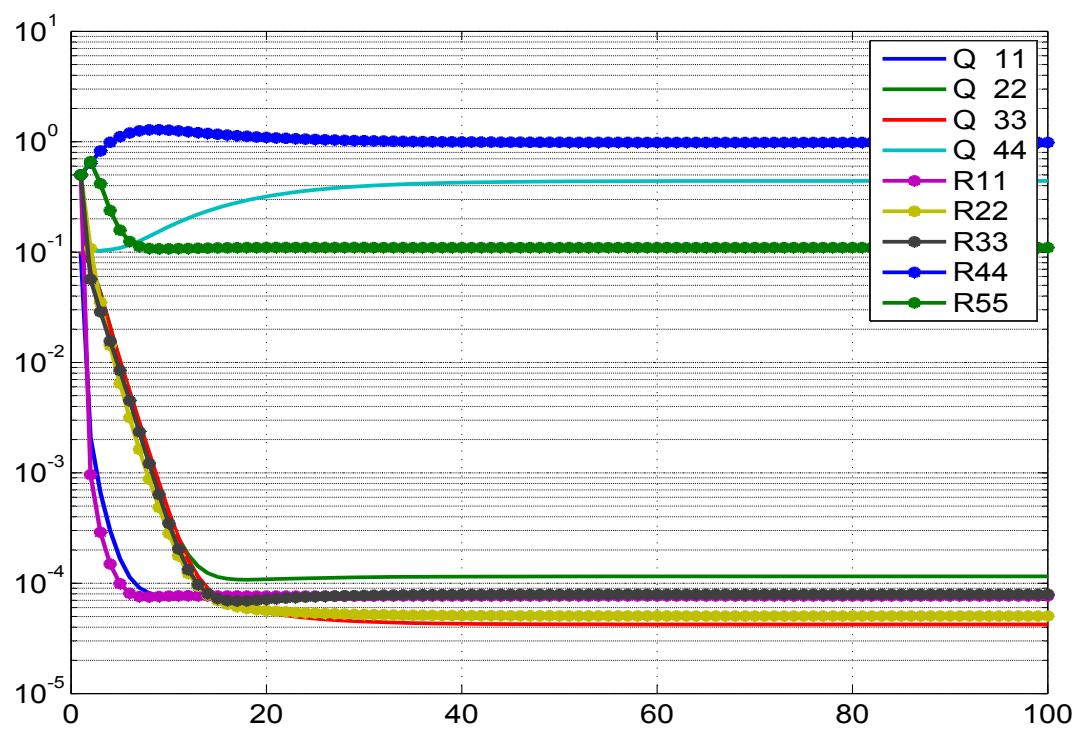
Figure 4.96: Variation of **Q** and **R** with iterations
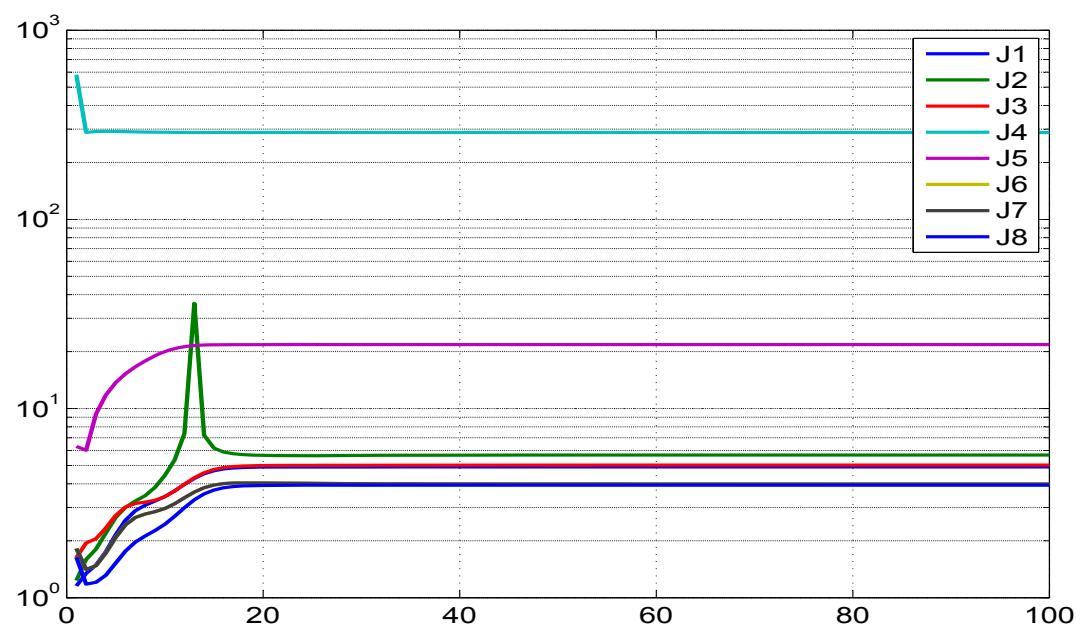
Figure 4.97: Variation of different costs (**J1-J8**) with iterations
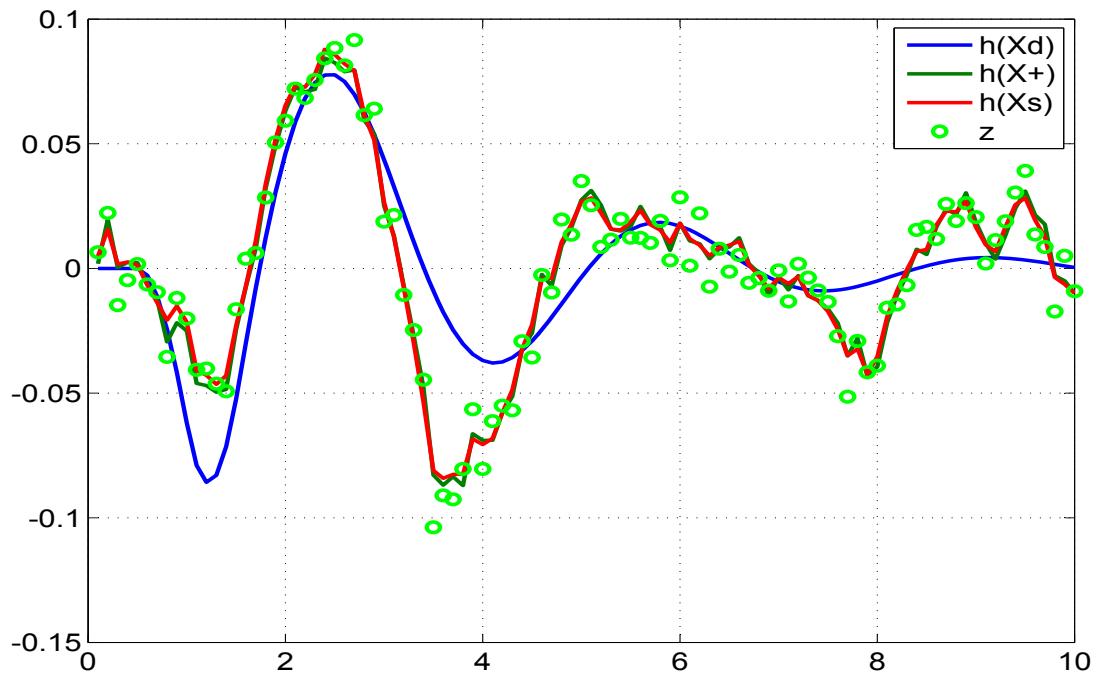
Figure 4.98: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1
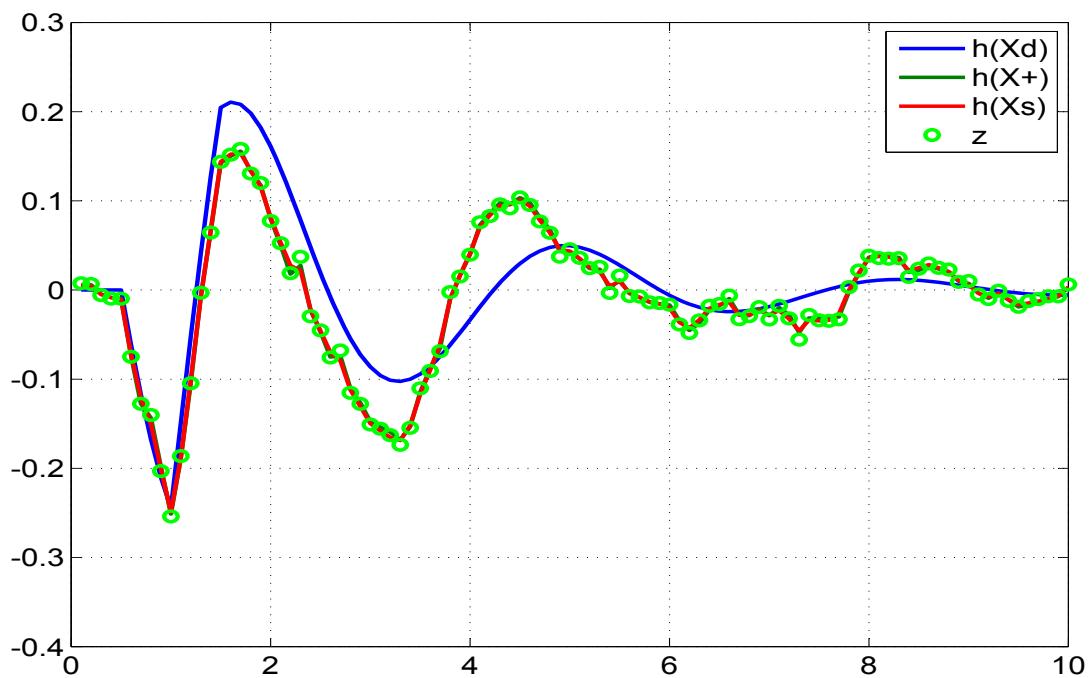
Figure 4.99: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 2
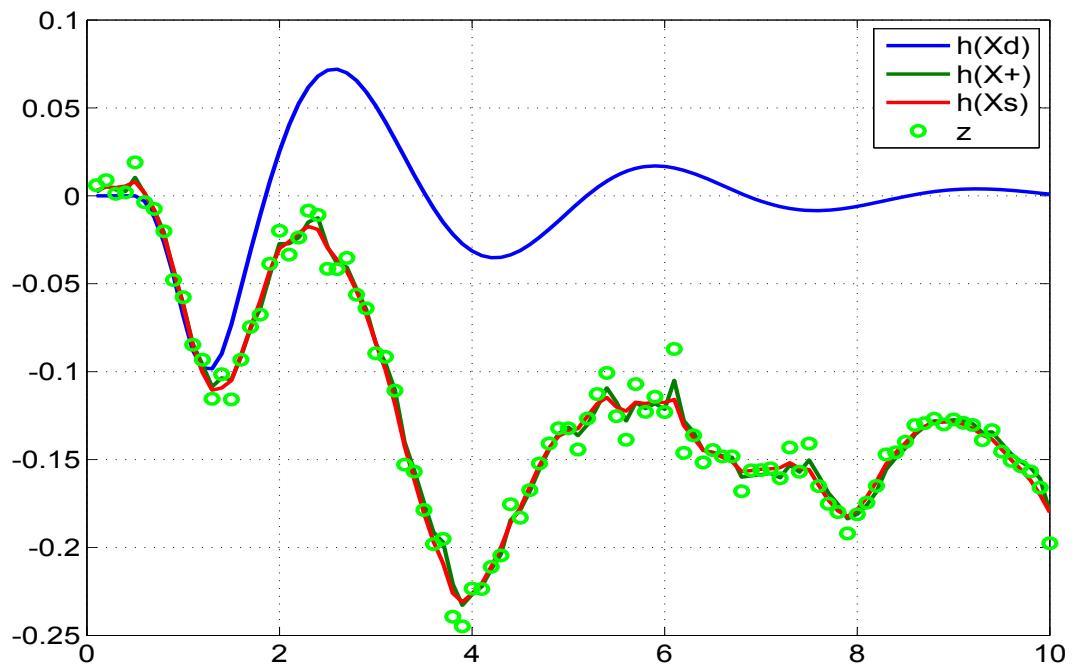
Figure 4.100: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 3
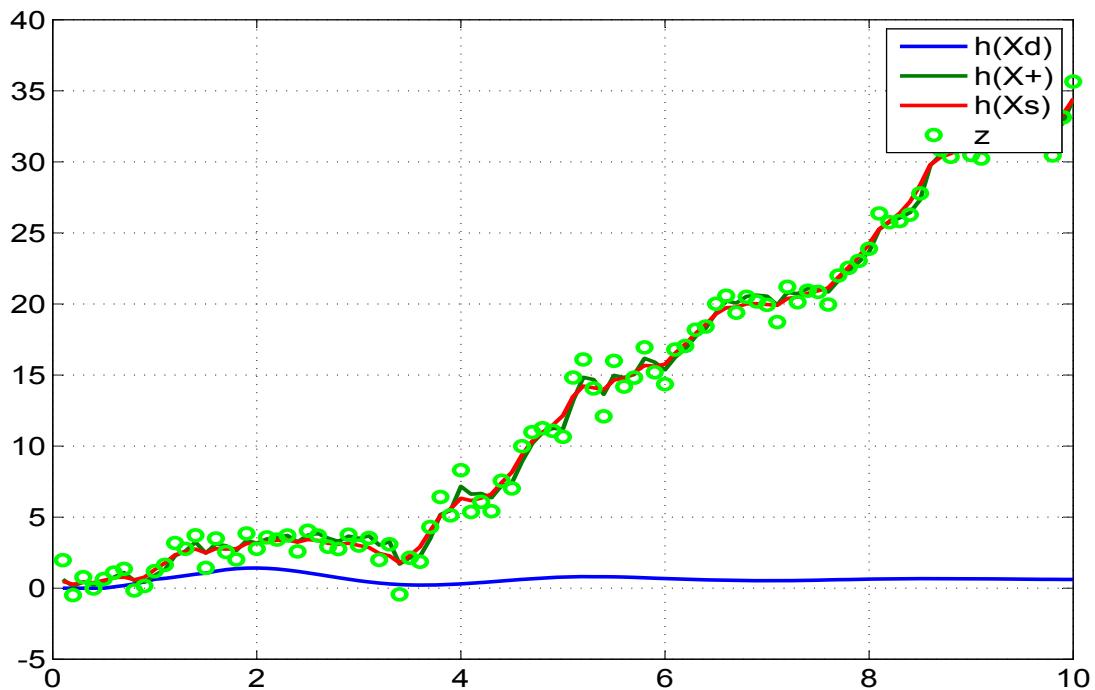
Figure 4.101: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 4

158
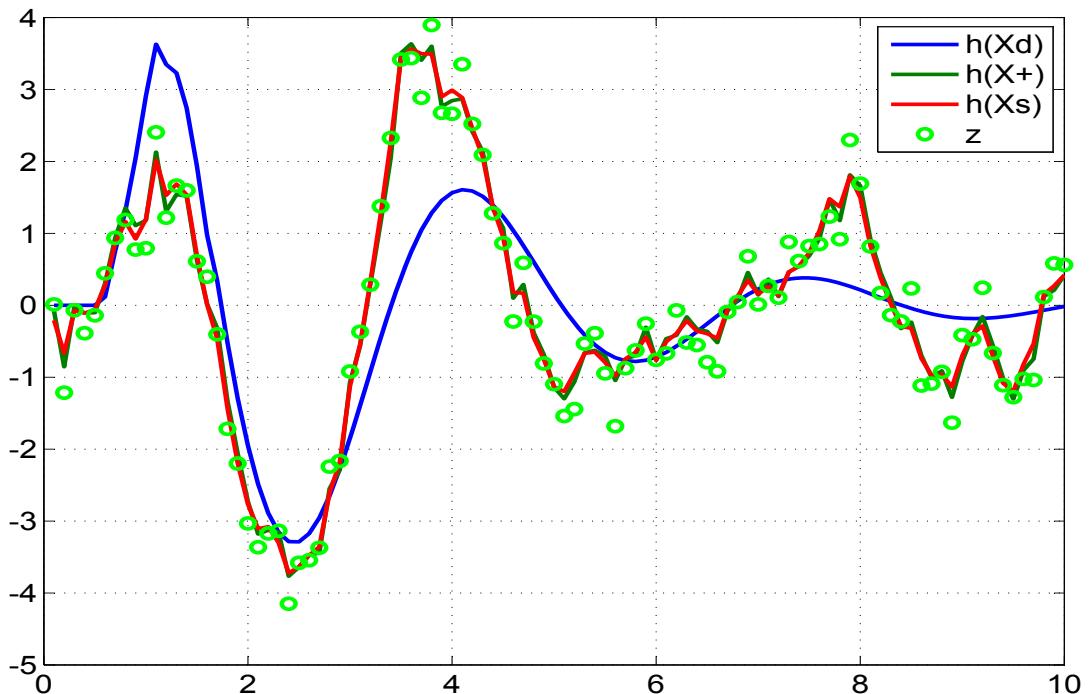
Figure 4.102: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 5
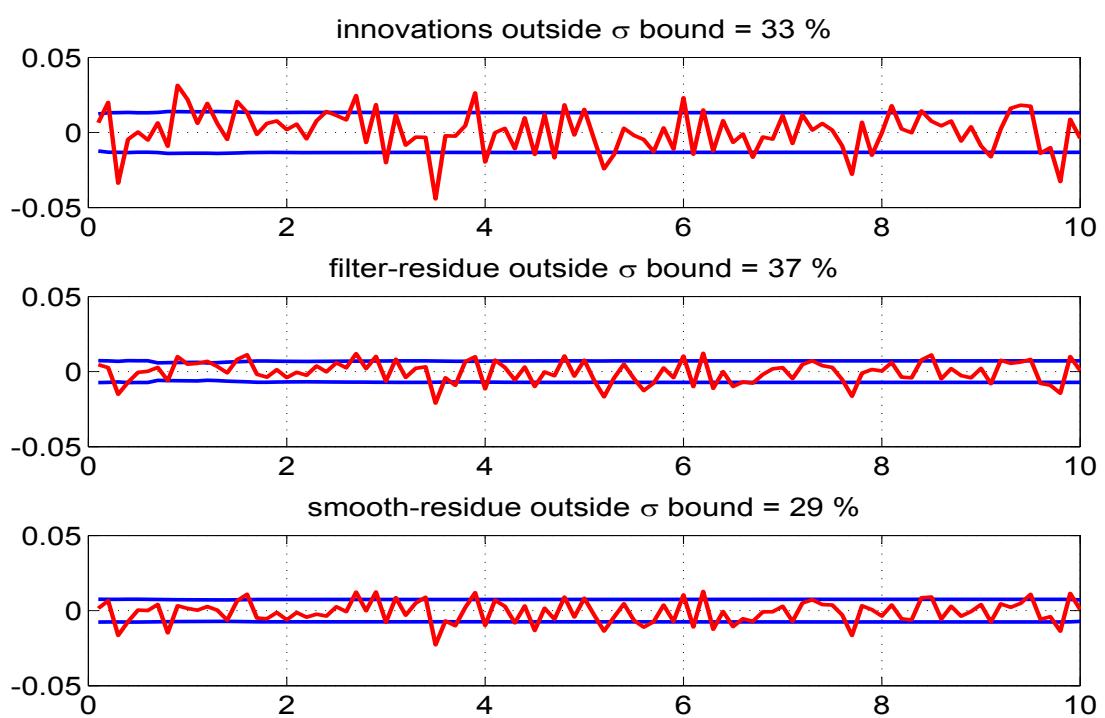
Figure 4.103: The innovations, filtered residue and smoothed residue

corresponding to measurement 1
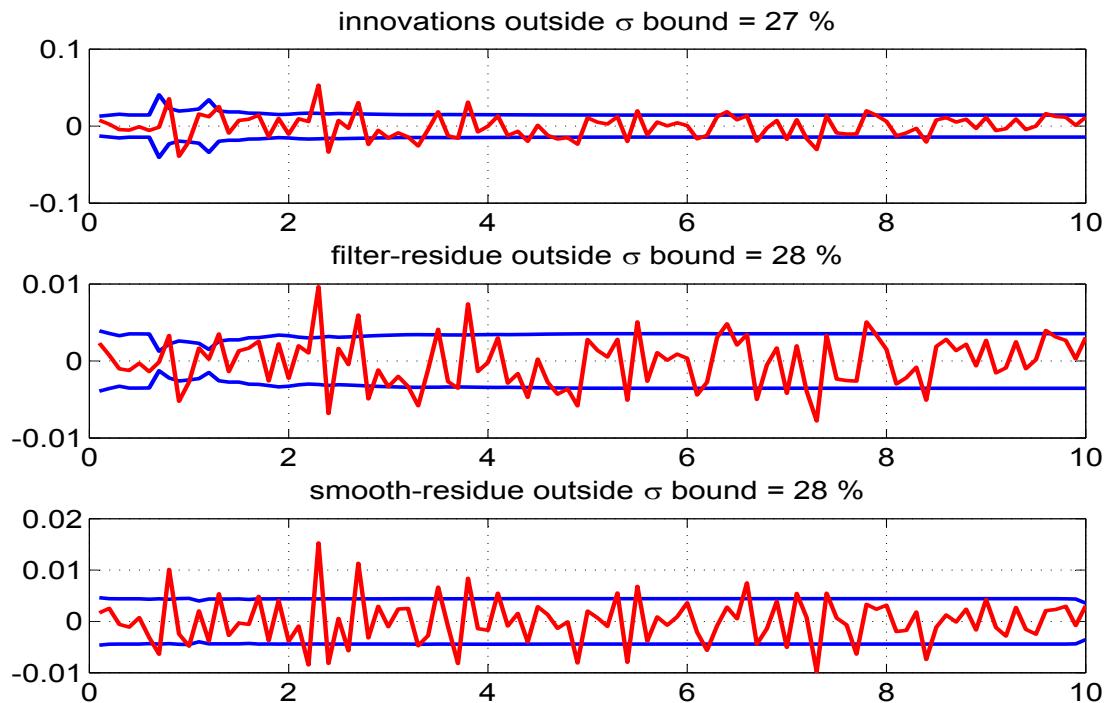
Figure 4.104: The innovations, filtered residue and smoothed residue

corresponding to measurement 2
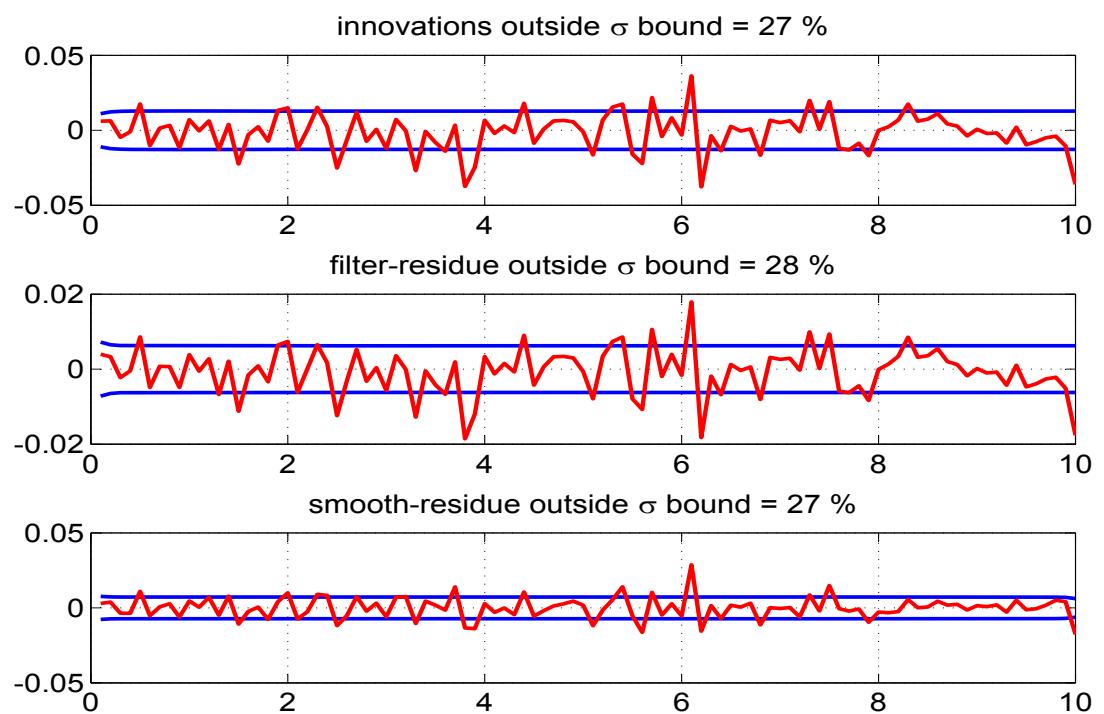
Figure 4.105: The innovations, filtered residue and smoothed residue

corresponding to measurement 3
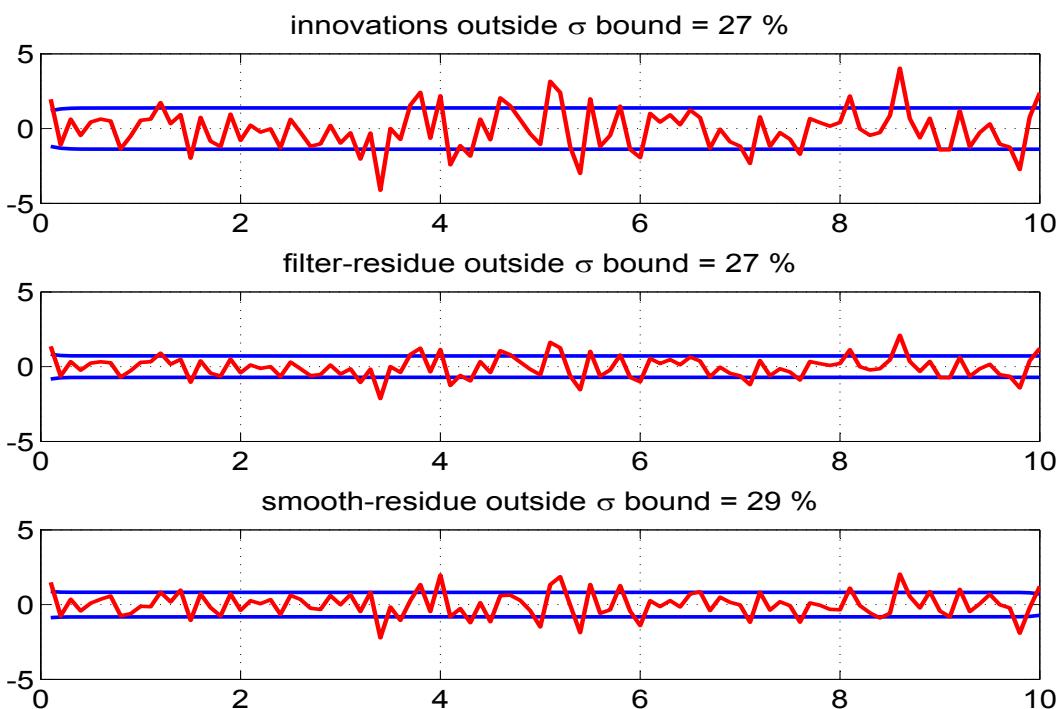
Figure 4.106: The innovations, filtered residue and smoothed residue

corresponding to measurement 4
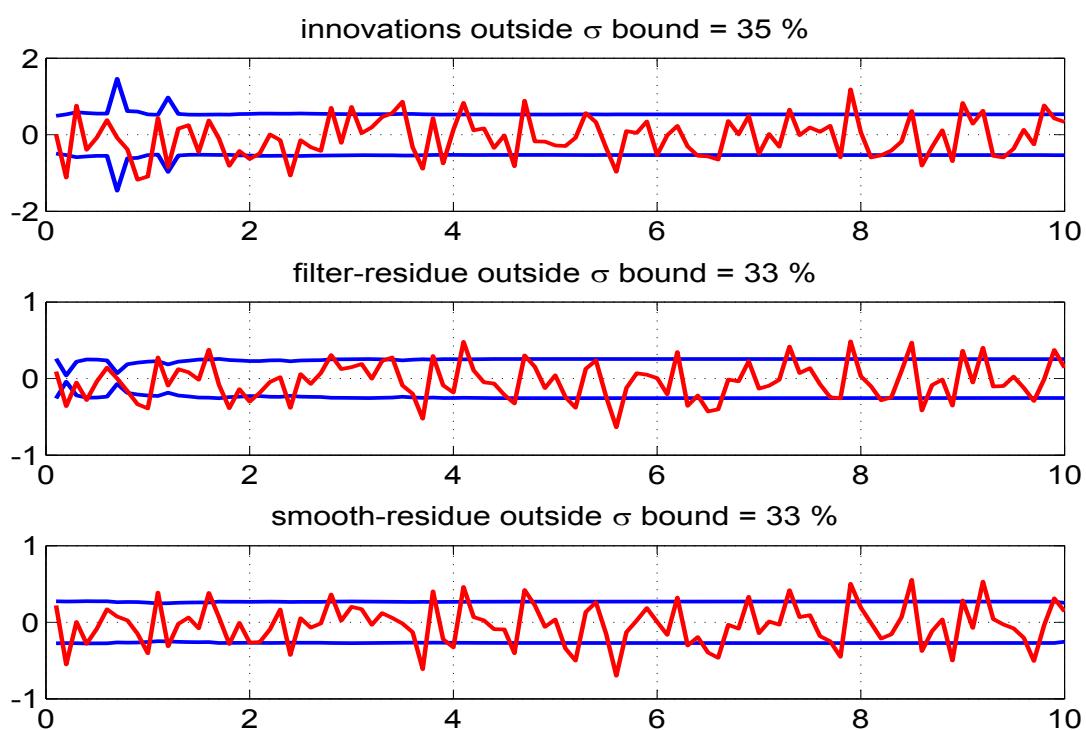
Figure 4.107: The innovations, filtered residue and smoothed residue

corresponding to measurement 5
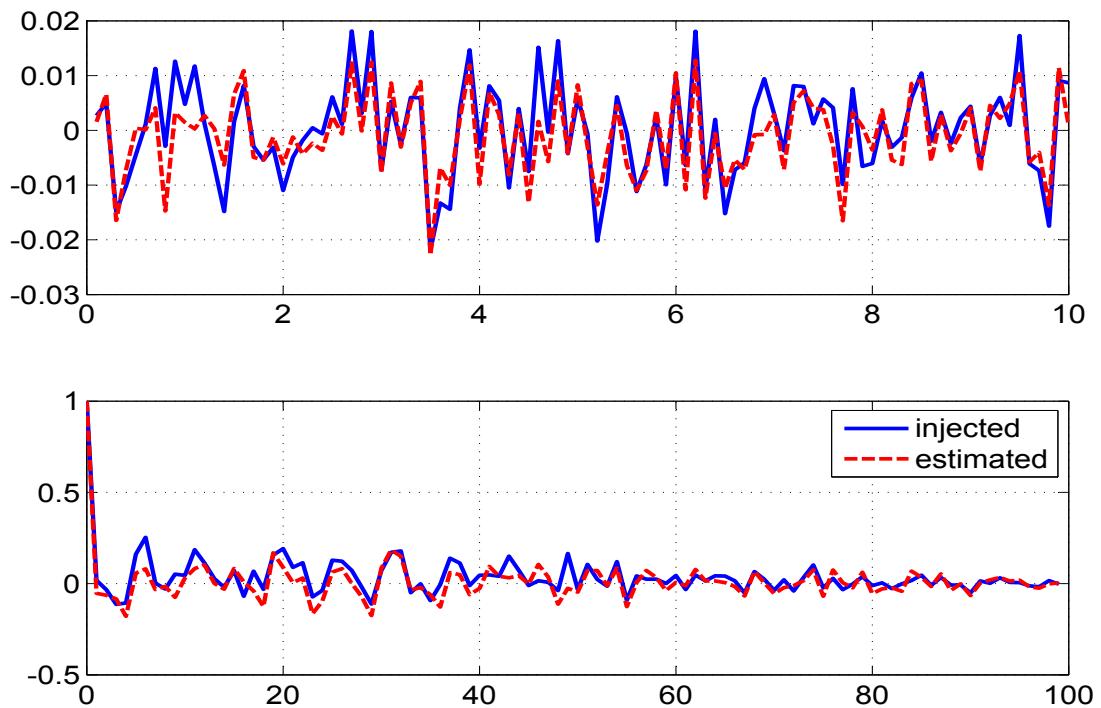
Figure 4.108: Time variation of injected and estimated measurement noise (top) and their autocorrelation (bottom) for measurement 1
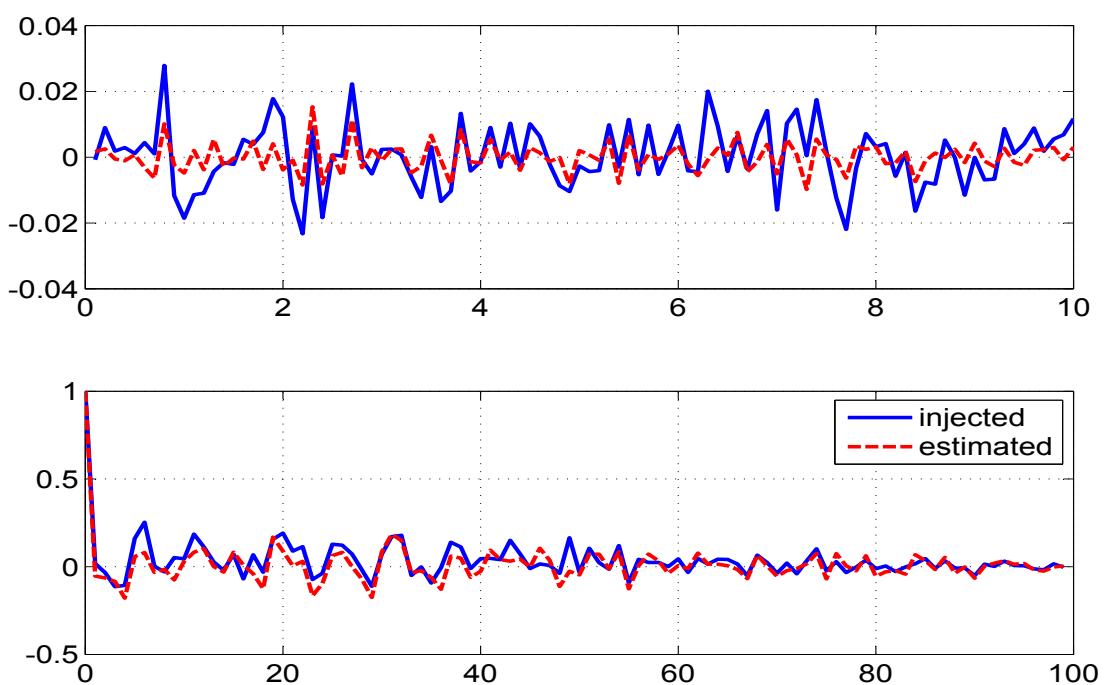
Figure 4.109: Time variation of injected and estimated measurement noise (top) and their autocorrelation (bottom) for measurement 2
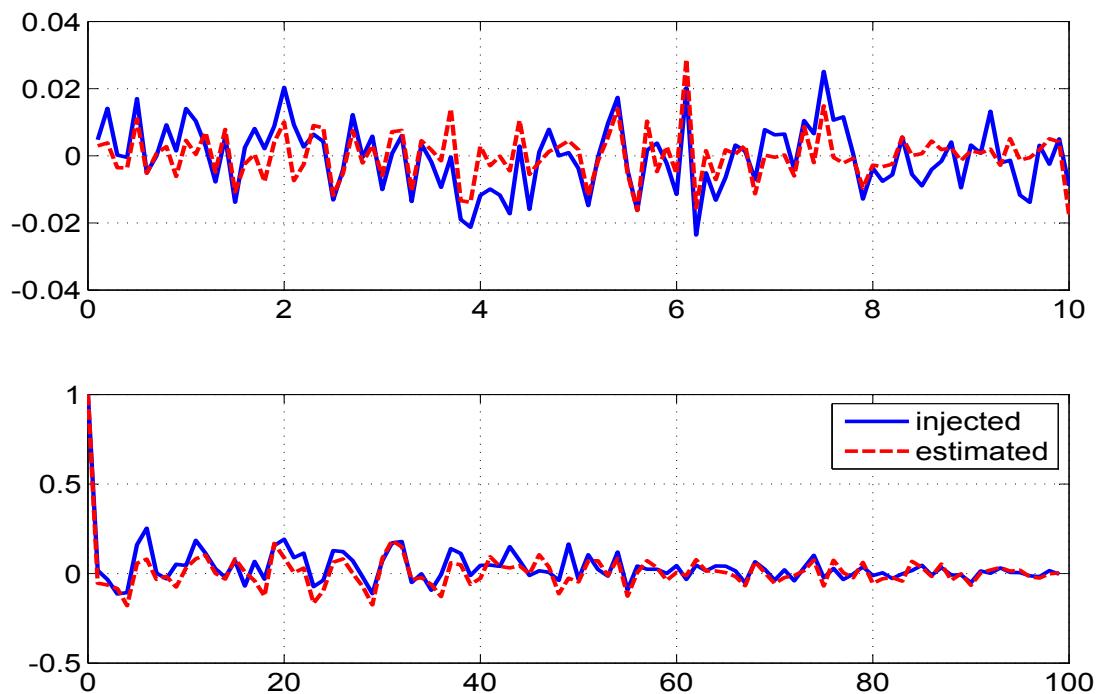
Figure 4.110: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 3
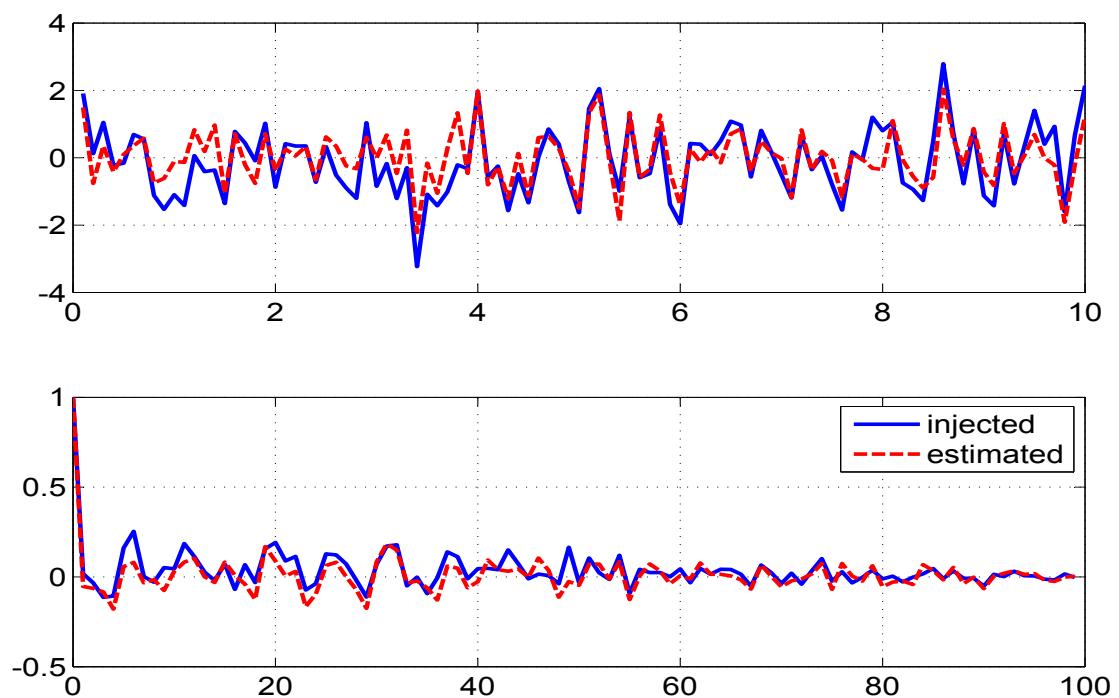
Figure 4.111: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 4
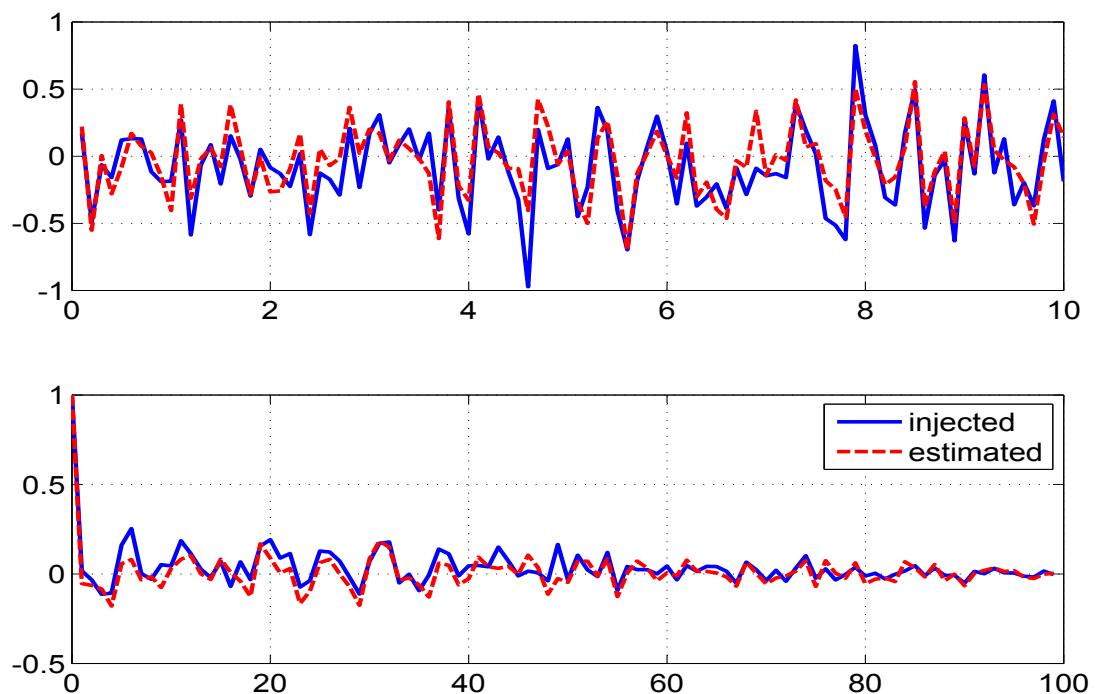
Figure 4.112: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 5
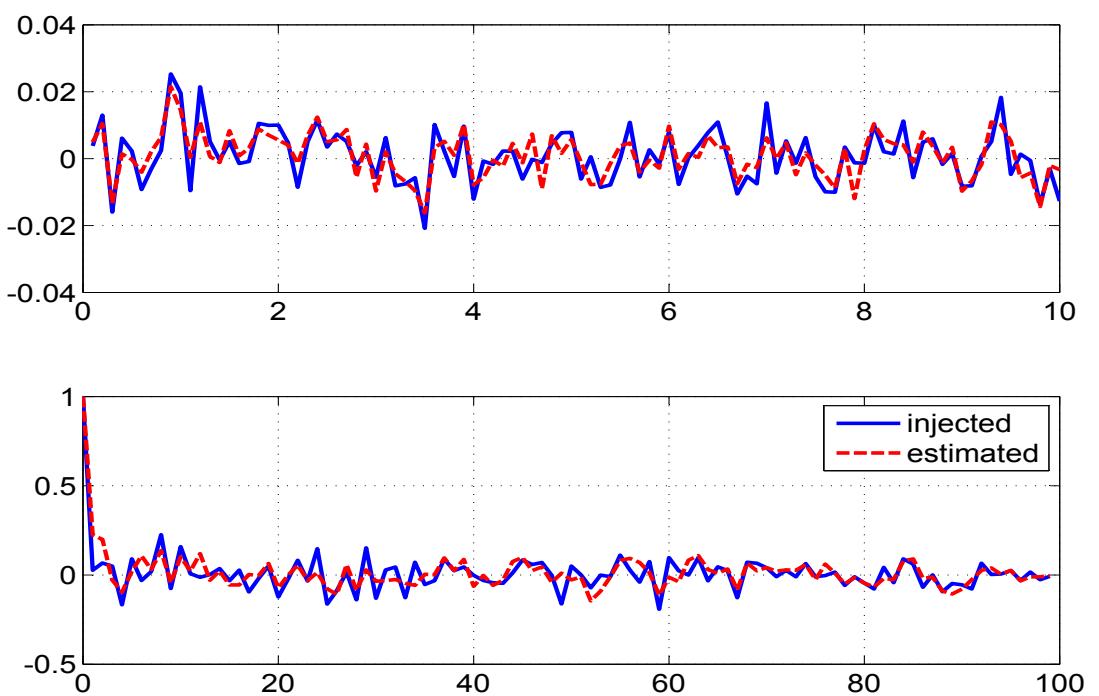
Figure 4.113: Time variation of injected and estimated process noise (top) and

their autocorrelation (bottom) for state 1
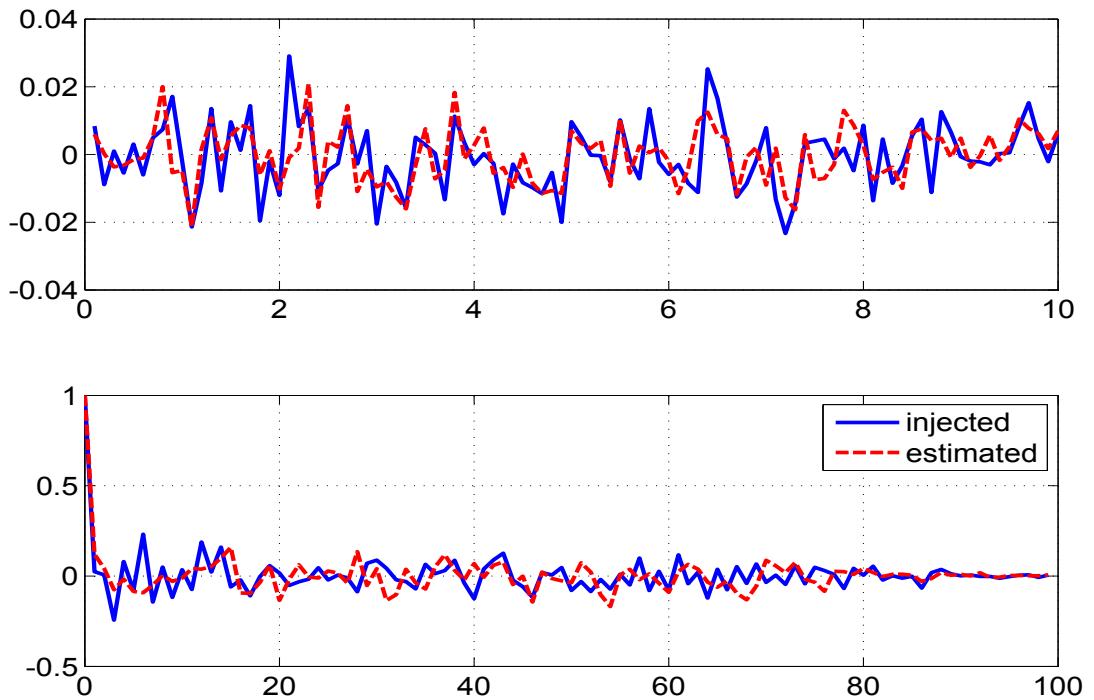
Figure 4.114: Time variation of injected and estimated process noise (top) and

their autocorrelation (bottom) for state 2
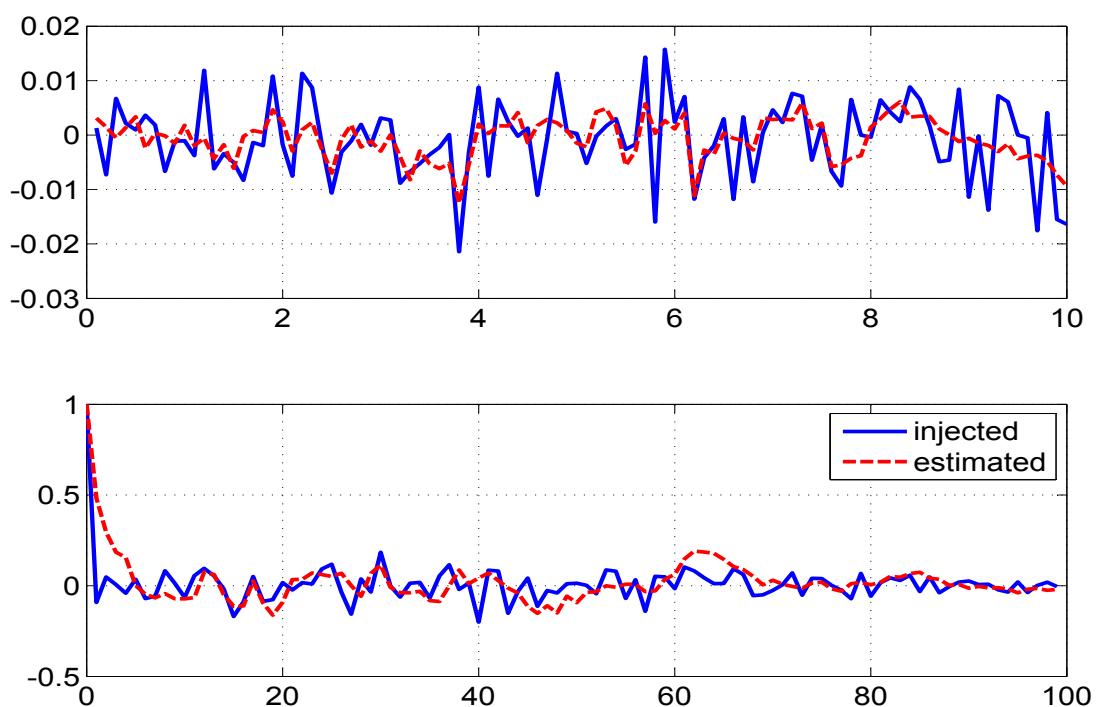
Figure 4.115: Time variation of injected and estimated process noise (top) and

their autocorrelation (bottom) for state 3
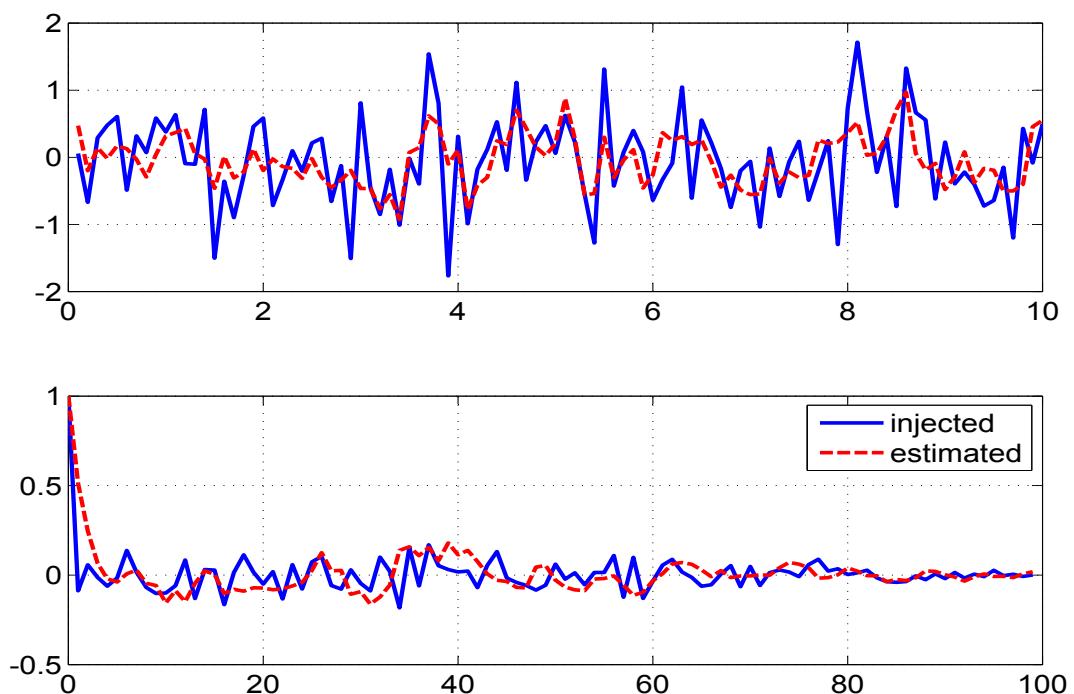
Figure 4.116: Time variation of injected and estimated process noise (top) and

their autocorrelation (bottom) for state 4

4.5 Simulated Lateral Motion of Aircraft System

Consider the Lateral motion of an aircraft system excited by control input as shown in Fig. 4.117. The state equations ($n = 4$) for the angle of sideslip (β), roll rate (p), roll angle (ϕ) and yaw rate (r) are respectively given by

$$\dot{\beta} = Y_\beta \beta + Y_p p + Y_\phi \phi - r + Y_{\delta_a} \delta_a + Y_{\delta_r} \delta_r$$

$$\dot{p} = L_\beta \beta + L_p p + L_r r + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r$$

$$\dot{\phi} = p$$

$$\dot{r} = N_\beta \beta + N_p p + N_r r + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r$$

All the states have zero initial conditions. The measured quantites are indicated with subscript 'm' which includes the angle of sideslip (β), roll rate (p), roll angle (ϕ), yaw rate (r), lateral acceleration (a_y), roll acceleration ($\dot{p}$) and yaw acceleration ($\dot{r}$). The measurement equations ($m = 7$) are given by

$$\beta_m = \beta + v_\beta$$

$$p_m = p + v_p$$

$$\phi_m = \phi + v_\phi$$

$$r_m = r + v_r$$

$$a_{y_m} = U_0 \times (Y_\beta \beta + Y_{\delta_a} \delta_a + Y_{\delta_r} \delta_r) + v_{a_y}$$

$$\dot{p}_m = L_\beta \beta + L_p p + L_r r + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r + v_p$$

$$\dot{r}_m = N_\beta \beta + N_p p + N_r r + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r + v_r$$

The unknown parameter set(p=15) is $\Theta = (Y_\beta, Y_p, Y_r, L_\beta, L_p, L_r, N_\beta, N_p, N_r, Y_{\delta_a}, Y_{\delta_r}, L_{\delta_a}, L_{\delta_r}, N_{\delta_a}, N_{\delta_r})^T$ with the true values being $(-0.18, -0.00278, 0.14, -0.097, -5.82, 1.782, 0.0084, -0.665, -0.712, -0.00447, 0.02657, 16.434, 0.434, -0.428, -2.824)^T$ and $U_0 = 100$. The numerical values of the noise variances are shown in Table-4.1. All the figures are presented for only one simulation run to prevent cluttering.

4.5.1 Remarks on the Results

We first run the filter assuming $\mathbf{Q} = 0$. It was found that about 20 iterations of the data would suffice. The Fig. 4.118-4.132 shows the various parameter estimates and its corresponding variances through cumulative time instants with iterations. The variation of the estimated initial parameters and their variances through iterations are shown in Fig. 4.133. The parameter and the uncertainty reach almost their final estimated values in about 2 and 5 iterations respectively. Similar plots in Fig. 4.134 and Fig. 4.135 shows the variation of the estimated $\mathbf{R}$ and cost functions ($\mathbf{J1}$-$\mathbf{J5}$) through the iterations. The Fig. 4.136-4.142 shows the predicted dynamics, filtered and smoothed estimate at the last iteration. The Fig. 4.143-4.149 show the innovations, filtered residue and smoothed residue together with the square root of their variance ($\pm\sigma$ bound). In the EKF approach most of the quantities are Gaussian or approximated as quasi Gaussian and one would expect all the above quantities are close to being Gaussian and hence around one third of the total sample points to be outside the σ bound. The injected and estimated measurement noise distributions during the final iteration shown in Fig. 4.150-4.156 indicate that they are close to each other. Further their autocorrelations are ideally expected to be close to the Kronecker delta function which provides confidence in the proposed algorithm.

The next step is to process the data with process noise ($\mathbf{Q} > 0$). The Fig. 4.157 shows the absolute difference between the iterated and final values with iterations which indicates the accuracy level that one needs and it was found that 100 iterations are required. The variation of the estimated initial parameters and their variances through iterations are shown in Fig. 4.158. The parameter and the uncertainty reach almost their final estimated values in about 5 and 20 iterations respectively. A similar plot in Fig. 4.159 shows the variation of the estimated $\mathbf{R}$ and $\mathbf{Q}$. The variation of different cost functions through the iterations are shown in Fig. 4.160. The cost functions $\mathbf{J1}$-$\mathbf{J3}$ correspond to the number of measurement (m=7) and in presence of process noise, $\mathbf{J6}$-$\mathbf{J8}$ correspond to the number of states (n=4). The $\mathbf{J4}$ in absence of process noise corresponds to the trace of $\mathbf{R}$. The $\mathbf{J5}$ is the negative log likelihood cost whose absolute value is shown in the plot. The mismatch in the predicted dynamics and the measurements in Fig. 4.161-4.167 indicates the presence of $\mathbf{Q}$. The subsequent Fig. 4.168-4.174 correspond to the earlier Fig. 4.143-4.149 of $\mathbf{Q} = 0$ case. The Fig. 4.175-4.181 and Fig. 4.182-4.185 shows respectively the injected and estimated measurement and process noise samples during the final iteration.

4.5.2 Lateral Motion of Aircraft System Tables (Q = 0)

Table 4.10: Sensitivity Study : ($\mathbf{Q} = 0$) : Lateral Aircraft system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**R** ratio EKF/NR	μ of J1-J5	σ of J1-J5	Remarks
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: EM-diag	Reference adaptive EKF ($\mathbf{Q} > 0$) gives extremely slow convergence of $\mathbf{Q}$ taking more than 100 iterations.								
Proposed reference adaptive EKF for zero process noise case, $\mathbf{Q}$ is diagonal matrix fixed to 10^{-10} for all states Consistency ratio NR= 1.1692, 0.9759, 1.0174, 0.9121, 1.0440, 0.7969, 0.9606, 1.1951, 1.0610 , 1.0618, 0.9218, 1.0657, 0.9702, 1.1985, 1.2637 Spread Factor NR= 1.0150, 153.8666, 1.2514, 60.5416, 1.7035, 2.0545, 199.2658, 8.8451, 2.1629, 32.2073, 5.1724, 1.5438, 22.8668, 36.5847, 1.4793									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag	0.9990, 1.2045, 0.9974, 1.0143, 1.0028, 0.9983, 1.0100, 1.0014, 0.9975, 1.0102 , 0.9940, 1.0020, 0.9670, 0.9926 , 1.0001	0.9925, 0.9973, 0.9935, 0.9928, 0.7217, 0.8653, 0.9796,0.8510, 0.8466, 1.0136 , 1.0040, 0.6704, 0.9818, 0.8549, 1.0354	1.1619, 0.9749, 0.9986, 1.0639, 0.8768, 0.7658, 1.1737,1.0488, 1.0403, 1.0719, 0.9147, 0.8447, 1.0134, 1.0203, 1.3072	1.0192, 154.0626, 1.2466, 65.0897, 2.1798, 2.3535, 225.2347, 9.7853, 2.5230, 31.8982 , 5.1314, 2.0881, 23.9045, 39.6036, 1.4385	0.9773 1.0361 0.9982 0.9934 1.0008 1.0406 0.9878	1.0131 1.0336 1.0183 1.0394 1.0299 1.0256 1.0152	6.8448 6.8572 7.0020 0.0103 -49.3730	0.1658 0.1651 0.0196 0.0015 0.4259	$\sigma_\Theta=$ 0.0015, 0.0031, 0.0013, 0.0488, 0.0861, 0.0262, 0.0150, 0.0488, 0.0133, 0.0011, 0.0009, 0.2267, 0.0758, 0.1258, 0.0349 $SIGMA_{avg}=$ 0.0013, 0.0032, 0.0013, 0.0458, 0.0981, 0.0342, 0.0128, 0.0465, 0.0128, 0.0010, 0.0010, 0.2684, 0.0748, 0.1233, 0.0267
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag IEKF	0.9990, 1.2132, 0.9975, 1.0330, 1.0027, 0.9984, 1.0112,1.0025, 0.9972, 1.0110, 0.9941, 1.0018, 0.9807, 0.9878, 1.0003	0.9926, 0.9975, 0.9937, 0.9930, 0.7218, 0.8650, 0.9799, 0.8507, 0.8457, 1.0136, 1.0041, 0.6706, 0.9827, 0.8546, 1.0355	1.1632, 0.9739, 1.0027, 1.0690, 0.8763, 0.7694, 1.1837, 1.0449 , 1.0453, 1.0727, 0.9140, 0.8448, 1.0220, 1.0185, 1.3045	1.0199, 154.0575, 1.2463, 65.1652, 2.1788, 2.3588, 226.6207,9.7739, 2.5339, 31.9160 , 5.1280, 2.0873, 23.8021, 39.6100, 1.4374	0.9773 1.0362 0.9981 0.9945 1.0008 1.0410 0.9874	1.0130 1.0337 1.0182 1.0405 1.0299 1.0260 1.0149	6.8470 6.8583 7.0020 0.0103 -49.3701	0.1674 0.1665 0.0196 0.0015 0.4260	$\sigma_\Theta=$ 0.0015, 0.0031, 0.0013, 0.0490, 0.0860, 0.0263, 0.0152, 0.0486, 0.0134, 0.0011, 0.0009, 0.2266, 0.0763, 0.1256, 0.0348 $SIGMA_{avg}=$ 0.0013, 0.0032, 0.0013, 0.0458, 0.0981, 0.0342, 0.0128, 0.0465, 0.0128, 0.0010, 0.0010, 0.2683, 0.0747, 0.1234, 0.0267

Table 4.10: Sensitivity Study : ($\mathbf{Q} = 0$) : Lateral Aircraft system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**R** ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
$\mathbf{P_0}$: Scaled up-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag IEKF	0.9989, 1.2459, 0.9971, 1.0381, 1.0031, 0.9975, 0.6238, 0.9996, 0.9974, 1.0129 , 0.9946, 1.0018, 0.9963, 0.9982, 0.9979	0.9942, 0.9951, 0.9584, 0.9715, 0.7238, 0.8645, 1.0667, 0.8463, 0.8419, 1.0143, 0.9893, 0.6702, 0.9037, 0.8486, 1.0203	1.1584, 0.9868, 1.1616, 1.1822, 0.8613, 0.7929, 1.5266, 1.0295, 1.0027, 1.0762, 0.9345, 0.8370, 1.0839, 1.0057, 1.4594	1.0186, 155.5837, 1.3714, 69.5064, 2.1606, 2.4004, 237.5212, 9.7509, 2.4920, 32.0314, 5.2582, 2.0830, 26.5464, 39.6918, 1.5538	0.9816 1.0463 1.0012 1.0575 0.9960 1.0348 0.9962	1.0175 1.0438 1.0214 1.1064 1.0250 1.0199 1.0239	6.8244 6.8473 7.0003 0.0103 -49.2573	0.2049 0.2031 0.0187 0.0016 0.4732	Unity CRB ratio
$\mathbf{P_0}$: Scaled up-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag IEKF	0.9990, 1.2454, 0.9981, 1.0206, 1.0027, 0.9990, 1.0533, 1.0022, 0.9974, 1.0096 , 0.9959, 1.0021, 0.9775, 0.9878, 1.0004	0.9934, 0.9907, 0.9623, 0.9426, 0.7300, 0.8692, 0.9550, 0.8581, 0.8524, 1.0135, 0.9810, 0.6788, 0.9072, 0.8627, 1.0141	1.1560, 0.9669, 0.9419, 1.0905, 0.8751, 0.7410, 1.2805, 1.0497, 0.9990, 1.0673 , 0.9254, 0.8451, 1.0474, 1.0218, 1.3249	1.0170, 155.1119, 1.2465, 69.0018, 2.1502, 2.3241, 242.0594, 9.7080, 2.4557, 31.8713, 5.2662, 2.0633, 26.0829, 39.2891, 1.4710	0.9762 1.0223 0.9993 0.9848 1.0016 1.0306 0.9885	1.0120 1.0198 1.0194 1.0304 1.0308 1.0158 1.0160	6.8045 6.8453 7.0019 0.0102 -49.1906	0.1908 0.1888 0.0183 0.0015 0.4351	Unity CRB ratio

Table 4.10: Sensitivity Study : ($\mathbf{Q} = 0$) : Lateral Aircraft system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**R** ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
P$_0$: IIM-[0,0;0,$\checkmark$] **Q** : 10^{-10}-[$\checkmark$,0;0,0] **R** : EM-diag IEKF	0.9989, 1.2449, 0.9981, 0.9983, 1.0028, 0.9993, 1.1540, 1.0028, 0.9965 , 1.0127 , 0.9945, 1.0021, 0.9740, 0.9844, 1.0002	0.9915, 0.9899, 0.9847, 0.9861, 0.7193, 0.8613, 0.9730, 0.8488, 0.8440, 1.0125, 1.0029, 0.6687, 0.9791, 0.8536, 1.0355	1.1625, 0.9752, 0.9424, 1.1079, 0.8821, 0.7978, 1.2250, 1.0373, 1.0470, 1.0636 , 0.9251, 0.8490, 1.0283, 1.0113, 1.3048	1.0233, 155.4587, 1.2141, 67.0329, 2.1895, 2.3980, 231.3071, 9.7648, 2.5407, 31.8428, 5.1540, 2.0977, 24.0242, 39.5683, 1.4396	0.9818 1.0400 0.9991 0.9979 1.0030 1.0469 0.9884	1.0177 1.0374 1.0192 1.0441 1.0322 1.0319 1.0159	6.8445 6.8623 7.0025 0.0103 -49.0731	0.2190 0.2173 0.0200 0.0015 0.4340	Unity CRB ratio
P$_0$: IIM-diag **Q** : 10^{-10}-[$\checkmark$,0;0,0] **R** : EM-diag IEKF	0.9988, 1.2716, 0.9984, 1.0193, 1.0026, 0.9991, 1.0849, 1.0023, 0.9969, 1.0117, 0.9960, 1.0021, 0.9822, 0.9859, 1.0001	0.9912, 0.9806, 0.9511, 0.9353, 0.7242, 0.8616, 0.9457, 0.8517, 0.8457, 1.0109, 0.9802, 0.6731, 0.9022, 0.8564, 1.0075	1.1643, 0.9709, 0.9159, 1.1186, 0.8850, 0.7733, 1.3434, 1.0395, 1.0076, 1.0651, 0.9051, 0.8516, 1.0544, 1.0133, 1.3310	1.0258, 157.0830, 1.2409, 70.5333, 2.1731, 2.3751, 251.0510, 9.7345, 2.4824, 31.9001, 5.2199, 2.0850, 26.3485, 39.4385, 1.4884	0.9784 1.0304 0.9999 0.9908 1.0031 1.0385 0.9895	1.0142 1.0279 1.0200 1.0366 1.0323 1.0235 1.0170	6.7974 6.8444 7.0017 0.0102 -48.9684	0.2314 0.2284 0.0195 0.0016 0.4389	Unity CRB ratio
P$_0$: IIM-full **Q** : 10^{-10}-[$\checkmark$,0;0,0] **R** : EM-diag IEKF	0.9989, 1.2681, 0.9984, 1.0221, 1.0025, 0.9991, 1.1118, 1.0028, 0.9967, 1.0112 , 0.9962, 1.0019, 0.9897, 0.9835, 1.0001	0.9933, 0.9809, 0.9503, 0.9341, 0.7196, 0.8592, 0.9446, 0.8482, 0.8435, 1.0108 , 0.9792, 0.6691, 0.9014, 0.8532, 1.0069	1.1643, 0.9696, 0.9194, 1.1918, 0.8820, 0.7672, 1.3024, 1.0281, 1.0402, 1.0595 , 0.9049, 0.8512, 1.0677, 1.0034, 1.3466	1.0230, 156.8616, 1.2443, 72.8894, 2.1838, 2.3816, 245.7278, 9.7346, 2.5236, 31.8495, 5.2267, 2.0969, 26.4839, 39.4765, 1.5006	0.9783 1.0338 0.9995 0.9931 1.0038 1.0486 0.9910	1.0141 1.0312 1.0196 1.0391 1.0330 1.0335 1.0186	6.8049 6.8352 7.0019 0.0103 -49.0281	0.2177 0.2163 0.0195 0.0016 0.4310	Unity CRB ratio

Table 4.10: Sensitivity Study : ($\mathbf{Q} = 0$) : Lateral Aircraft system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**R** ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
$\mathbf{P_0}$: Smoothed-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag IEKF	0.9991, 1.1701, 0.9974, 1.0332, 1.0020, 1.0004, 0.8688, 0.9942, 0.9987, 1.0108, 0.9936, 1.0044, 0.8758, 1.0115, 1.0009	4.2097, 4.2225, 4.2202, 4.1573, 3.0842, 3.6371, 4.1848, 3.6626, 3.6094, 4.3177, 4.2761, 2.8470, 4.0742, 3.6461, 4.3865	4.9086, 4.1081, 4.3202, 3.7738, 3.6153, 2.7622, 3.9802, 4.2684, 3.8233, 4.5531 , 3.9190, 3.3175, 4.0104, 4.2435, 5.3230	0.6829, 93.7791, 0.8005, 34.8745, 1.2827, 1.1357, 124.4322, 6.1603, 1.4765, 20.7795 , 3.1263, 1.2059, 17.5755, 24.4412, 0.9789	0.9598 1.0052 0.9743 0.9551 0.9680 1.0140 0.9717	0.9949 1.0027 0.9939 0.9993 0.9961 0.9994 0.9987	6.9905 6.9904 6.9990 0.0102 -50.1023	0.0025 0.0025 0.0014 0.0015 0.4227	Lower filter CRB than that of the NR
$\mathbf{P_0}$: Smoothed-diag $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag IEKF	0.9994, 1.1773, 0.9988, 1.0742, 1.0452, 1.0473, 0.7050, 0.9653, 0.9991, 1.0129, 0.9974, 1.0599, 0.4335, 1.0664, 1.0060	4.8848, 7.1513, 7.7761, 6.6641, 9.0458, 9.4296, 6.9313, 16.7962, 12.6527, 5.0318 , 4.9523, 12.1572, 5.5411, 24.6483, 7.0731	5.6485, 7.0356, 7.7960, 7.0054, 7.9707, 6.7037, 7.0583, 16.3888, 12.1193, 5.2630, 4.6454, 8.2965, 5.6760, 24.1527, 8.5493	0.6699, 89.2043, 0.7296, 38.5294, 4.5225, 4.7347, 125.5112, 5.6088, 1.2154, 20.3946, 3.1094, 5.9887, 56.7594, 19.7969, 1.0682	0.9539 1.0900 0.9826 0.9482 0.9666 1.1917 0.9733	0.9888 1.0873 1.0024 0.9921 0.9947 1.1745 1.0003	6.9820 6.9821 6.9975 0.0109 -49.8671	0.0061 0.0061 0.0015 0.0017 0.4260	Lower filter CRB than that of the NR
$\mathbf{P_0}$: Smoothed-full $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: EM-diag IEKF	0.9990, 1.2152, 0.9982, 1.0454, 1.0018, 1.0006, 0.9862, 0.9939, 0.9989, 1.0079 , 0.9962, 1.0043, 0.8751, 1.0118, 1.0016	4.2119, 4.1823, 4.0611, 3.9365, 3.0908, 3.6328, 4.0566, 3.6656, 3.6125, 4.3119 , 4.1689, 2.8531, 3.7416, 3.6488, 4.2656	4.8754, 4.1032, 4.0720, 4.0456, 3.6276, 2.7046, 4.1937, 4.2643, 3.8290, 4.5156 , 3.9087, 3.3351, 4.1927, 4.2411, 5.3009	0.6794, 94.4882 , 0.7723, 39.5495, 1.2815 , 1.1268, 133.7273, 6.1593 , 1.4714, 20.7010 , 3.1897, 1.2068, 19.5474, 24.4637, 0.9863	0.9520 0.9930 0.9682 0.9433 0.9660 1.0095 0.9728	0.9869 0.9906 0.9876 0.9869 0.9940 0.9950 0.9998	6.9878 6.9881 6.9987 0.0101 -50.1493	0.0028 0.0028 0.0015 0.0015 0.4253	Lower filter CRB than that of the NR

Table 4.10: Sensitivity Study : ($\mathbf{Q} = 0$) : Lateral Aircraft system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{R}$ ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag*	0.9990, 1.2054, 0.9974, 1.0148, 1.0028, 0.9983, 1.0058, 1.0012, 0.9975, 1.0102, 0.9939, 1.0021, 0.9669, 0.9930, 1.0001	1.0072, 1.0106, 1.0074, 1.0042, 0.7264, 0.8734, 0.9927, 0.8591, 0.8558, 1.0309, 1.0216, 0.6749, 0.9910, 0.8626, 1.0476	1.1791, 0.9878, 1.0126, 1.0762, 0.8836, 0.7736, 1.1896, 1.0608, 1.0530, 1.0903 , 0.9309, 0.8517, 1.0226, 1.0312, 1.3245	1.0114, 152.8164, 1.2368, 64.6694, 2.1720, 2.3390, 223.8159, 9.7431, 2.5102, 31.5948 , 5.0773, 2.0803, 23.7799, 39.4302, 1.4315	0.9571 1.0169 0.9759 0.9639 0.9663 1.0314 0.9749	0.9921 1.0145 0.9955 1.0085 0.9944 1.0165 1.0020	6.9932 7.0060 7.1543 0.0103 -49.3749	0.1692 0.1687 0.0216 0.0015 0.4259	*Using smoothed residue ignoring 2nd order terms in Eq-3.7
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag*	0.9990, 1.2021, 0.9974, 1.0191, 1.0028, 0.9984, 1.0124, 1.0011, 0.9975, 1.0102, 0.9940, 1.0021, 0.9682, 0.9938, 1.0001	1.0643, 1.0623, 1.0602, 1.0423, 0.7518, 0.9114, 1.0419, 0.8992, 0.8967, 1.0979, 1.0896, 0.7009, 1.0363, 0.9023, 1.0998	1.2449, 1.0400, 1.0664, 1.1228, 0.9235, 0.8126, 1.2554, 1.1119, 1.1084, 1.1595, 0.9922, 0.8936, 1.0654, 1.0811, 1.3881	0.9830, 148.4543, 1.2018, 63.5143, 2.1335, 2.2781, 219.2865, 9.5198, 2.4555, 30.5047, 4.8847, 2.0408, 23.1390, 38.5133, 1.4004	0.8818 0.9056 0.9348 0.8669 0.8485 0.9858 0.8985	0.9141 0.9034 0.9537 0.9070 0.8732 0.9716 0.9234	7.6354 7.6486 7.8126 0.0103 -49.3405	0.1281 0.1109 0.2424 0.0015 0.4250	*Using filtered residue ignoring 2nd order terms in Eq-3.19
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag*	0.9991, 1.1882, 0.9974, 1.0153, 1.0030, 0.9983, 1.0006, 1.0007, 0.9977, 1.0105 , 0.9938, 1.0020, 0.9724, 0.9955, 1.0002	0.9607, 0.9665, 0.9620, 0.9606, 0.7042, 0.8394, 0.9515, 0.8360, 0.8283, 0.9848, 0.9761, 0.6535, 0.9527, 0.8410, 1.0109	1.1194, 0.9534, 0.9672, 1.0442, 0.8552, 0.7480, 1.1466, 1.0283, 1.0135, 1.0380, 0.8831, 0.8327, 0.9724, 0.9972, 1.2802	1.0339, 157.6258, 1.2725, 66.4180, 2.2192, 2.4109, 229.4332, 9.8697, 2.5480, 32.3891 , 5.2102, 2.1333, 24.1328, 39.8876, 1.4614	1.0556 1.1108 1.0788 1.0697 1.0648 1.0969 1.0242	1.0943 1.1081 1.1005 1.1192 1.0958 1.0811 1.0526	6.5499 6.5631 6.7007 0.0103 -49.2789	0.9452 0.9574 1.0203 0.0015 0.4140	*Using innovations ignoring 2nd order terms in Eq-3.20
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: diag*	0.9990, 1.2052, 0.9974, 1.0124, 1.0029, 0.9983, 1.0134, 1.0013, 0.9975, 1.0102, 0.9939, 1.0021, 0.9676, 0.9931, 1.0000	1.0008, 1.0040, 1.0020, 0.9988, 0.7299, 0.8744, 0.9880, 0.8586, 0.8551, 1.0254, 1.0163, 0.6785, 0.9923, 0.8625, 1.0463	1.1706, 0.9809, 1.0067, 1.0754, 0.8906, 0.7770, 1.1871, 1.0581, 1.0533, 1.0834 , 0.9262, 0.8549, 1.0147, 1.0304, 1.3276	1.0142, 153.3664, 1.2401, 64.9841, 2.1690, 2.3400, 224.6858, 9.7391, 2.5126, 31.6725, 5.0932, 2.0727, 23.6377, 39.4288, 1.4354	0.9745 1.0053 0.9971 0.9686 0.9762 1.0186 0.9731	1.0102 1.0029 1.0171 1.0134 1.0046 1.0039 1.0001	6.9668 6.9795 7.1271 0.0103 -49.3719	0.2258 0.2239 0.1717 0.0015 0.4262	*Using dynamical residue as per Eq-3.18

Table 4.10: Sensitivity Study : ($\mathbf{Q} = 0$) : Lateral Aircraft system.
No. of iterations=20, No. of simulations=50.

Study	Θ ratio	CRB ratio NR/EKF	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**R** ratio EKF/NR	μ of **J1-J5**	σ of **J1-J5**	Remarks
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: MT-diag	0.9992, 1.2063, 0.9974, 1.0556, 1.0021, 0.9978, 0.9917, 1.0030, 0.9970, 1.0120, 0.9934, 1.0010, 0.9904, 0.9831, 1.0007	1.0099, 1.0370, 1.0369, 1.0494, 0.7713, 0.9133, 1.0375, 0.8915, 0.8880, 1.0196, 1.0081, 0.7138, 1.0363, 0.8934, 1.0940	1.1573, 1.0155, 1.0558, 1.0967, 0.9008, 0.8586, 1.3394, 1.1279, 1.1102, 1.0721, 0.9083, 0.8897, 1.0563, 1.0710, 1.3553	0.9959, 151.6909, 1.2296, 62.3387, 2.0734, 2.3346, 227.0889, 9.7392, 2.4880, 31.7424, 5.1082, 2.0011, 22.8644, 38.9936, 1.3925	0.9294 0.9975 0.9202 0.8984 1.0148 0.9408 0.9569	0.9634 0.9950 0.9387 0.9400 1.0443 0.9272 0.9834	7.9279 7.9437 8.1171 0.0103 -48.9504	2.7564 2.7666 2.9365 0.0015 0.5994	Using full expression in Eq-3.20
$\mathbf{P_0}$: Scale up-[0,0;0,✓] $\mathbf{Q}$: 10^{-10}-[✓,0;0,0] $\mathbf{R}$: MS-diag	0.9990, 1.1995, 0.9974, 1.0170, 1.0028, 0.9984, 1.0247, 1.0016, 0.9975, 1.0101, 0.9940, 1.0020, 0.9690, 0.9925, 1.0001	1.0033, 1.0065, 1.0019, 0.9990, 0.7276, 0.8731, 0.9874, 0.8600, 0.8546, 1.0269, 1.0175, 0.6765, 0.9914, 0.8643, 1.0447	1.1742, 0.9860, 1.0079, 1.0772, 0.8862, 0.7761, 1.1888, 1.0566, 1.0506, 1.0847, 0.9263, 0.8553, 1.0231, 1.0299, 1.3125	1.0131, 153.3755, 1.2409, 65.0359, 2.1703, 2.3429, 224.8038, 9.7192, 2.5113, 31.6440, 5.0876, 2.0788, 23.7417, 39.3523, 1.4313	0.9621 1.0088 0.9941 0.9811 0.9741 1.0276 0.9643	0.9973 1.0063 1.0141 1.0264 1.0024 1.0128 0.9911	6.9706 6.9829 7.1308 0.0103 -49.3711	0.1122 0.0923 0.2109 0.0015 0.4248	Using full expression in Eq-3.19

4.5.3 Lateral Motion of Aircraft System Tables (Q > 0)

Table 4.11: Sensitivity Study : (**Q** > 0) : Lateral Aircraft system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**Q** ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
				Proposed reference adaptive EKF for non zero process noise					
P$_0$: Scaled up-[0,0;0,✓] **Q** : EM-[✓,0;0,0] **R** : EM-diag EKF	0.9988, 1.3069, 0.9875 , 1.0070, 1.0000, 1.0018, 1.0287,1.0104 , 1.0038, 1.0117 , 0.9955, 1.0011, 0.9693, 0.9810 1.0027	1.0035 1.0158 1.0157 1.0029	1.1851, 1.0470, 0.9742, 1.0537, 0.9017, 0.8138, 0.9498,0.8835, 0.9044 , 1.0751 , 0.9403, 0.8743, 0.9852, 0.8719 1.1659	1.1353, 862.0811, 11.1119, 232.8240, 3.4730, 4.5492, 966.6156,12.3076, 4.4412, 36.9012 , 6.0104, 2.8582, 39.0526, 44.6277, 2.4897	0.9638 0.9718 0.9800 0.9870 0.9914 1.0204 0.9870	1.0129 1.0273 1.0347 0.9984	6.8895 6.9973 7.0020 0.0624 -45.3321 3.9490 3.9491 3.9838	0.2670 0.4450 0.0311 0.0525 0.4042 0.0325 0.0329 0.1524	σ_{Θ}= 0.0017, 0.0184, 0.0112, 0.1717, 0.1402, 0.0523, 0.0577,0.0557, 0.0221 , 0.0013, 0.0011, 0.3217, 0.1241, 0.1288, 0.0553 $SIGMA_{avg}$= 0.0014, 0.0176, 0.0115, 0.1630, 0.1555, 0.0643, 0.0608, 0.0630, 0.0244, 0.0012, 0.0012, 0.3680, 0.1259, 0.1478, 0.0475
P$_0$: Scaled up-[0,0;0,✓] **Q** : EM-[✓,0;0,0] **R** : EM-diag IEKF	0.9996, 1.2879, 0.9877, 0.8382, 1.0141, 1.0088, 1.3180,1.0209, 1.0043, 1.0183 , 0.9960 , 1.0120, 1.0043, 0.9520, 1.0029	1.0035 1.0265 1.0155 1.0047	1.1840, 1.0443, 0.9697, 1.0387, 0.9657, 0.8136, 0.9370,0.8498, 0.8999, 1.0774, 0.9398, 0.9257, 0.9700, 0.8729 1.1680	1.1314, 864.5130, 11.1336, 232.9577, 3.7173, 4.5938, 962.3419,11.9090, 4.4143, 36.9535, 6.0049, 3.0622, 38.8536 , 43.7752, 2.4894	0.9612 0.9906 0.9795 0.9804 0.9886 0.9708 0.9839	1.0209 1.0358 1.0372 1.0079	6.8994 6.8505 7.0054 0.0625 -45.3093 3.9485 3.9485 3.9852	0.2654 0.2513 0.0328 0.0525 0.4092 0.0324 0.0327 0.1558	σ_{Θ}= 0.0017, 0.0185, 0.0112, 0.1696, 0.1451, 0.0520, 0.0570, 0.0521, 0.0218 , 0.0013 , 0.0011, 0.3308, 0.1226, 0.1261, 0.0554 $SIGMA_{avg}$= 0.0014, 0.0177, 0.0116, 0.1633, 0.1503, 0.0639, 0.0609, 0.0613, 0.0243 , 0.0012, 0.0012, 0.3573, 0.1264, 0.1445, 0.0474

Table 4.11: Sensitivity Study : ($\mathbf{Q} > 0$) : Lateral Aircraft system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	**R** ratio EKF/True	**Q** ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: EM-diag SOKF	0.9990, 1.3116, 0.9875, 0.9928, 1.0001, 1.0016, 1.1360, 1.0106, 1.0044, 1.0098 , 0.9959 , 1.0012, 0.9715, 0.9825, 1.0026	1.0036 1.0146 1.0156 1.0036	1.1840, 1.0472, 0.9747, 1.0427, 0.8956, 0.8116, 0.9482,0.8817, 0.9024, 1.0751, 0.9391, 0.8705, 0.9744, 0.8694, 1.1594	1.1343, 861.8517, 11.1096, 233.4512, 3.5111, 4.5793, 969.0388,12.3551, 4.4568, 36.9050, 6.0082 , 2.8932, 39.2475, 44.7999, 2.4979	0.9637 0.9411 0.9802 0.9845 0.9919 1.0698 0.9898	1.0121 1.0255 1.0344 0.9981	6.8541 7.0614 7.0025 0.0624 -45.3300 3.9483 3.9484 3.9856	0.2661 0.5166 0.0303 0.0525 0.4045 0.0321 0.0324 0.1508	σ_Θ= 0.0017, 0.0184, 0.0112 , 0.1710, 0.1410, 0.0526, 0.0578, 0.0558, 0.0221, 0.0013, 0.0011, 0.3245, 0.1239, 0.1291, 0.0554 $SIGMA_{avg}$= 0.0014, 0.0176, 0.0115, 0.1640, 0.1575, 0.0647, 0.0610, 0.0633, 0.0245, 0.0012, 0.0012 , 0.3728, 0.1272, 0.1485, 0.0478
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: Known IEKF	0.9996, 1.3012, 0.9875, 0.8827, 1.0140, 1.0094, 1.2659,1.0225, 1.0039, 1.0203 , 0.9965, 1.0121, 1.0062, 0.9461, 1.0028	0.9959 0.9981 0.9979 0.9901	1.1773, 1.0468, 0.9686, 1.0017, 0.8430, 0.8016, 0.9293, 0.8375, 0.8931, 1.0713 , 0.9197, 0.8189, 0.9518, 0.8664, 1.1606	1.1318, 866.1140, 11.1359, 231.5518 , 3.6018, 4.6441, 955.3048,11.8487, 4.3813, 37.0358, 5.9755, 2.9910 , 39.1211, 43.5260 , 2.4730	NA	1.0147 1.0285 1.0107 1.0018	6.8999 6.8179 6.9696 0.0625 -45.2718 3.9478 3.9479 3.9894	0.5458 0.5965 0.6564 0.0522 0.4007 0.0285 0.0287 0.2640	**R** is known

Table 4.11: Sensitivity Study : ($\mathbf{Q} > 0$) : Lateral Aircraft system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of J1-J8	σ of J1-J8	Remarks
P₀ : Zero **Q** : EM-diag **R** : EM-diag IEKF	NA	1.0045 1.0174 1.0154 1.0053	NA	NA	0.9746 0.9753 0.9809 0.9878 0.9945 0.9938 0.9774	0.9904 1.0288 1.0326 0.9929	6.9921 7.0063 7.0009 0.0717 -45.9025 4.0002 4.0003 3.9976	0.2624 0.2531 0.0113 0.0508 0.3919 0.0248 0.0255 0.1342	Θ is known
P₀ : Zero **Q** : EM-diag **R** : Known IEKF	NA	0.9996 0.9944 0.9979 0.9925	NA	NA	NA	0.9930 1.0258 1.0085 0.9912	6.9978 6.9844 6.9781 0.0717 -45.8670 3.9996 3.9998 4.0042	0.5168 0.5837 0.6240 0.0508 0.3878 0.0232 0.0240 0.2254	Θ and **R** is known

Table 4.11: Sensitivity Study : $\mathbf{Q} > 0$: Lateral Aircraft system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	R ratio EKF/True	Q ratio EKF/True	μ of J1-J8	σ of J1-J8	Remarks
MS method (Mohamed and Schwarz)									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MS-diag $\mathbf{R}$: MS-diag	**R** can converge to a very large value with far away initial conditions. There is no proper convergence even if Θ is known.								
MS method with **Q** being estimated using only last N/2 samples									
$\mathbf{P_0}$: Zero $\mathbf{Q}$: MS-diag $\mathbf{R}$: Known IEKF	NA	1.0100 1.0503 0.9981 1.0746	NA	NA	NA	0.9909 0.7928 1.1609 0.7904	7.7731 7.4569 7.3698 0.0717 -45.6042 5.4816 5.5710 4.5418	1.5827 1.4457 1.2896 0.0508 0.6309 4.9955 5.6072 0.9964	**R**, Θ is known
MS method assuming **R** is known and **Q** is estimated using only last N/2 samples									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MS-[✓,0;0,0] $\mathbf{R}$: Known IEKF	0.9996, 1.2676, 0.9879, 0.8834, 1.0127, 1.0100, 0.8569, 1.0233, 1.0033, 1.0189 , 0.9965, 1.0113, 0.9920, 0.9433 1.0037	0.9929 1.0606 0.9983 1.0582	1.1807, 1.0334, 0.9466, 1.0746, 0.9835, 0.8701, 0.9828,0.8449 , 1.0180, 1.0574, 0.9013, 0.9084, 1.1272, 0.9094, 1.1855	1.1335, 882.9809, 11.3537, 234.4577, 3.5932, 4.6951, 949.7402,11.6920, 4.4946, 36.8293, 5.9314, 2.9841, 40.1616, 43.9798, 2.4277	NA	1.0844 0.8148 1.1623 0.8176	7.5463 7.2900 7.3631 0.0620 -45.0151 4.2031 4.2025 4.3836	1.4701 1.5206 1.4066 0.0518 0.6604 3.4211 3.4256 0.7897	Stable operation
Gemson's approach using MS method									
$\mathbf{P_0}$: IIM-full $\mathbf{Q}$: MSMS-[✓,0;0,0] $\mathbf{R}$: 0.9*True **R**	0.9991, 1.2576, 0.9971, 0.8934, 1.0527, 1.0100, 0.8662, 1.0333, 1.0033, 1.0191 , 0.9946, 1.0120 0.9910, 0.9423 1.0765	0.9954 1.0611 0.9979 1.0662	1.1928, 1.0444, 0.9666, 1.0842, 1.0985, 0.8801, 0.9900,0.8912 , 1.0190, 1.1880, 0.9113, 0.9045, 1.1289, 0.9234, 1.2823	1.2367, 883.809, 11.3937, 235.865, 3.7911, 4.8967, 949.8912,11.920, 5.0991, 37.8293, 6.0317, 2.843, 40.2611, 44.098, 2.564	NA	1.0944 0.8141 1.1681 0.8576	7.5461 7.2910 7.3625 0.0621 -45.0151 4.204 4.2005 4.3842	1.4405 1.5301 1.4115 0.0567 0.6625 3.4223 3.4252 0.791	$\delta J = 10^-6$, with one loop of **R** estimation

Table 4.11: Sensitivity Study : $\mathbf{Q} > 0$: Lateral Aircraft system.
No. of iterations=100, No. of simulations=50.

Study	Θ ratio	PCRB ratio	Consistency ratio-EKF	Spread factor EKF	$\mathbf{R}$ ratio EKF/True	$\mathbf{Q}$ ratio EKF/True	μ of **J1-J8**	σ of **J1-J8**	Remarks
MT method (Myers and Tapley)									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MT-diag $\mathbf{R}$: MT-diag	**R and Q** cannot be estimated simultaneously . There is no proper convergence even if Θ is known.								
MT method with $\mathbf{Q}$ being estimated using only last N/2 samples									
$\mathbf{P_0}$: Zero $\mathbf{Q}$: MT-diag $\mathbf{R}$: Known IEKF	NA	0.9992 0.9978 1.0044 1.0000	NA	NA	NA	1.0449 1.0256 1.1214 1.0038	7.0697 7.0434 6.9995 0.0717 -45.7955 4.2887 4.2893 4.0697	0.9002 0.7497 0.6406 0.0508 0.3936 1.3455 1.3473 0.5942	$\mathbf{R}$, Θ is known
MT method assuming $\mathbf{R}$ is known and $\mathbf{Q}$ is estimated using only last N/2 samples									
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MT-[✓,0;0,0] $\mathbf{R}$: Known IEKF	0.9996, 1.2697, 0.9877, 0.8809, 1.0134, 1.0095, 1.0605,1.0215, 1.0036, 1.0182, 0.9963, 1.0116, 1.0020, 0.9499, 1.0032	0.9936 1.0030 1.0046 1.0012	1.1800, 1.0349, 0.9491, 0.9976, 0.8459, 0.8165, 0.9296,0.8253, 0.9154, 1.0535 , 0.9046, 0.8227, 0.9771, 0.8740, 1.1287	1.1337, 880.2806, 11.3298, 230.5343, 3.5689, 4.6652, 947.3631, 11.7144, 4.3840, 36.7781 , 5.9385, 2.9682, 39.3087, 43.4827, 2.4210	NA	1.0738 1.0189 1.1224 1.0079	6.9885 6.8976 7.0068 0.0623 -45.1920 4.2643 4.2650 4.0554	0.8959 0.7827 0.6746 0.0522 0.4080 1.4922 1.4959 0.5820	$\mathbf{R}$ is known
Gemson's approach using MT method									
$\mathbf{P_0}$: IIM-full $\mathbf{Q}$: MT-[✓,0;0,0] $\mathbf{R}$: 0.9*True $\mathbf{R}$	0.9989, 1.2178, 0.9888, 0.8934, 1.0303, 1.0269, 1.0714, 1.0192, 1.0033 , 1.0263, 0.9957, 1.0296, 0.9951, 0.9529, 1.0037	1.0037 1.0468 0.9981 1.0128	1.2361, 0.9820, 0.8870, 1.0153, 0.8965, 0.8345, 0.9365,0.8385, 0.9359, 1.1701, 0.9896, 0.8484, 0.9817, 0.8796, 1.1674	1.1190, 912.7317, 11.7061, 223.6725, 4.2567, 5.0563, 924.6135, 11.4621, 4.3242 , 37.6355, 5.9893, 3.8010, 38.4115, 42.4877, 2.4264	0.9320 0.8948 0.9138 0.9128 0.9062 0.8871 0.9258	1.2069 1.0567 1.3170 1.1136	7.2819 7.5925 7.5429 0.0635 -44.3797 4.1397 4.1397 4.0843	1.1025 1.2536 1.1724 0.0538 0.5381 1.0205 1.0210 0.5908	$\delta J = 10^-6$, with one loop of $\mathbf{R}$ estimation

4.5.4 Lateral Motion of Aircraft System Figures (Q = 0)
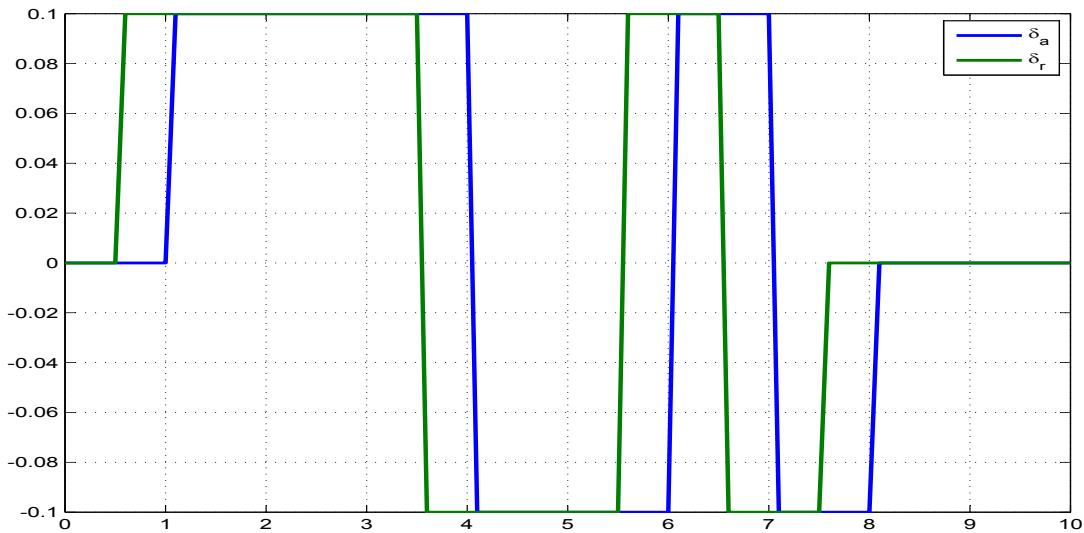
Figure 4.117: Control input of simulated Lateral aircraft system
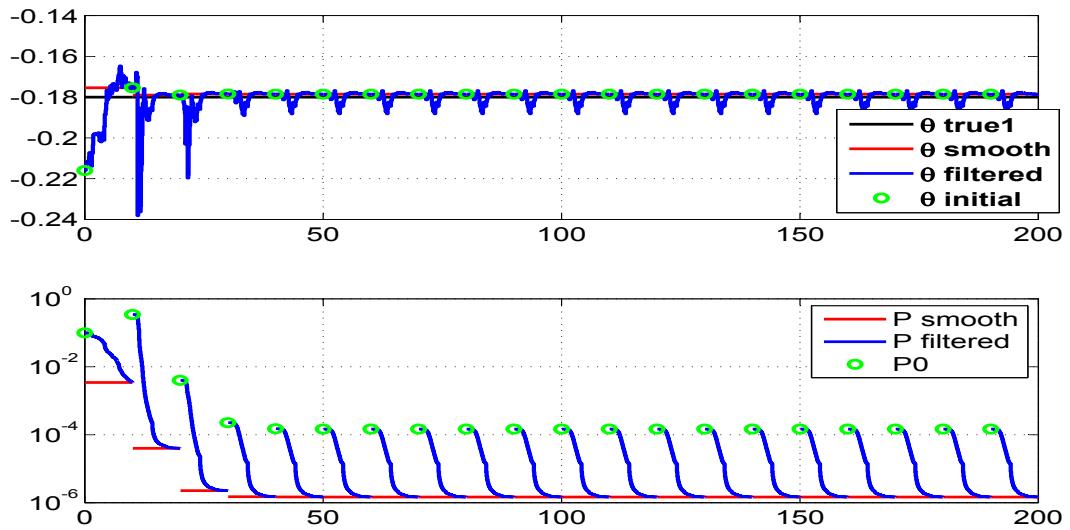
Figure 4.118: The variation of parameter estimate 1 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
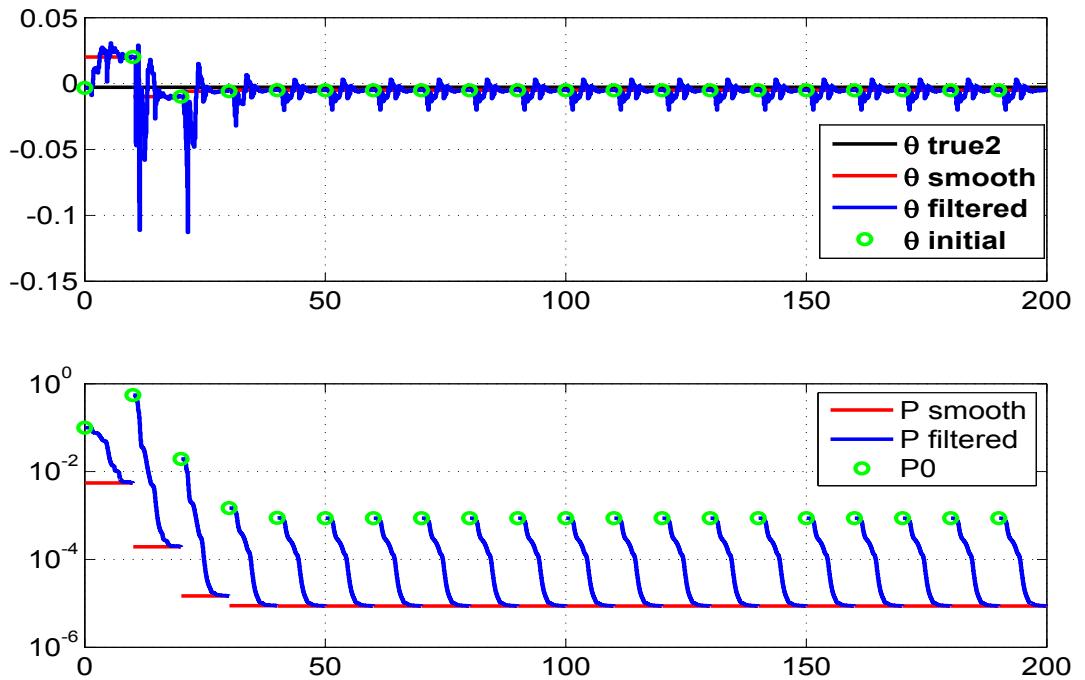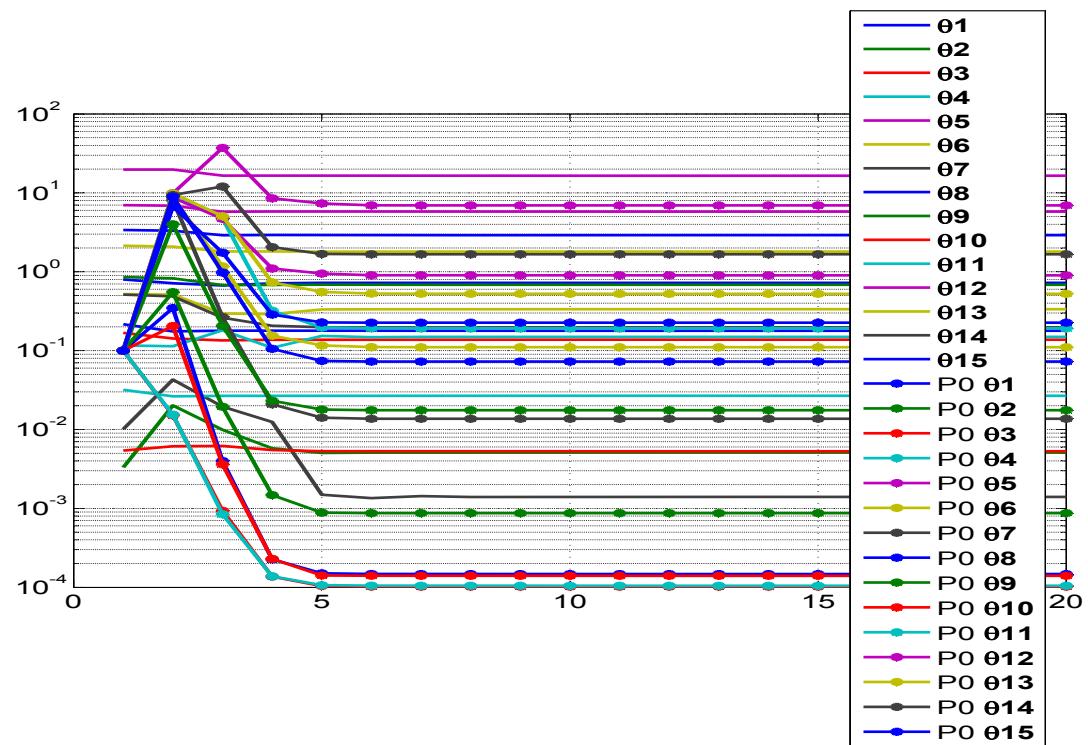
Figure 4.119: The variation of parameter estimate 2 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
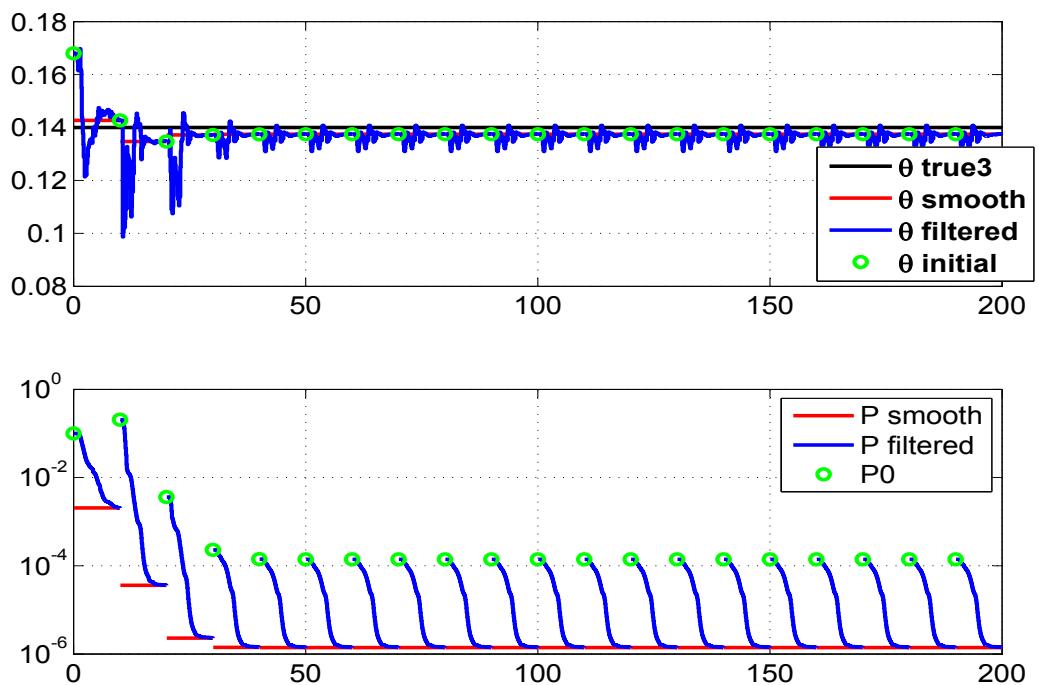
Figure 4.120: The variation of parameter estimate 3 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
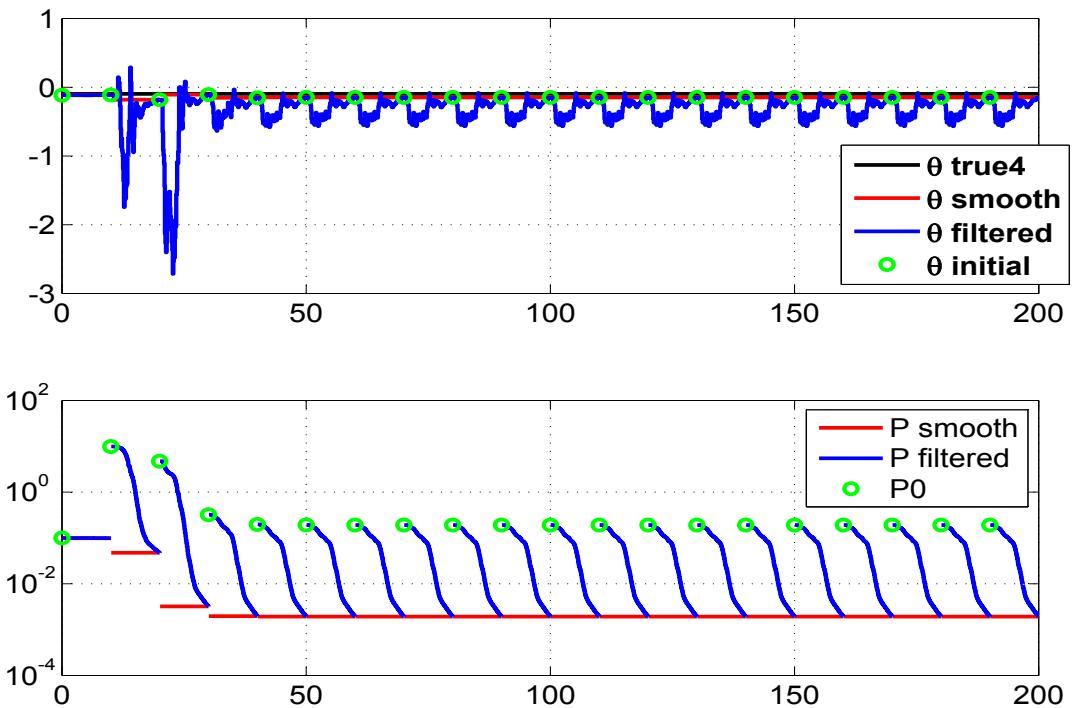
Figure 4.121: The variation of parameter estimate 4 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
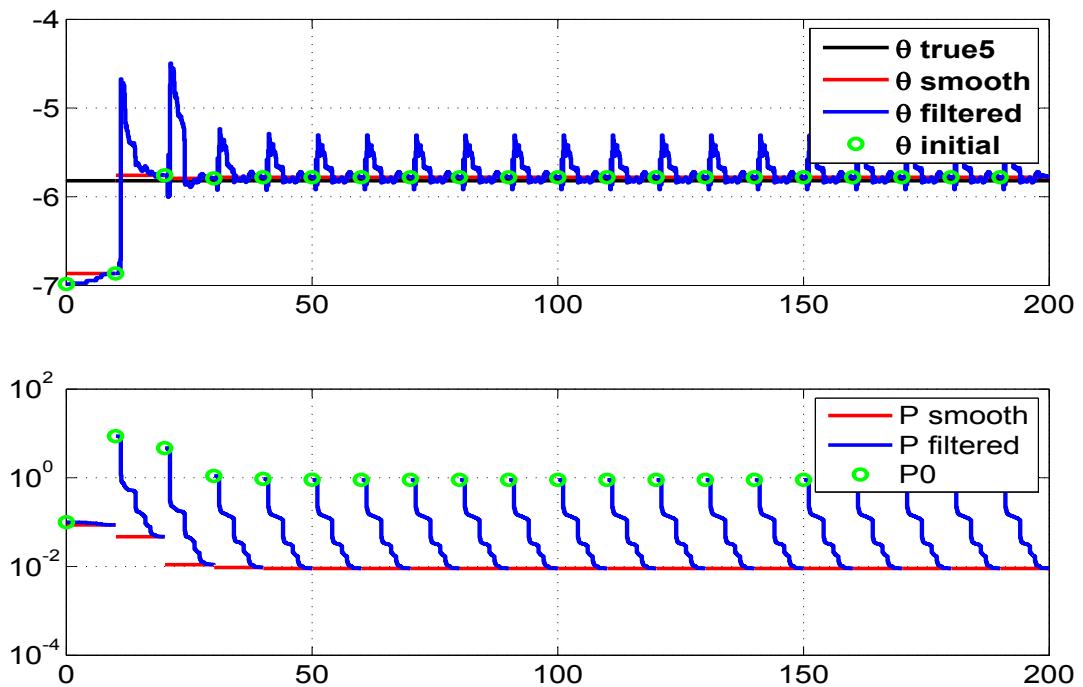
Figure 4.122: The variation of parameter estimate 5 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
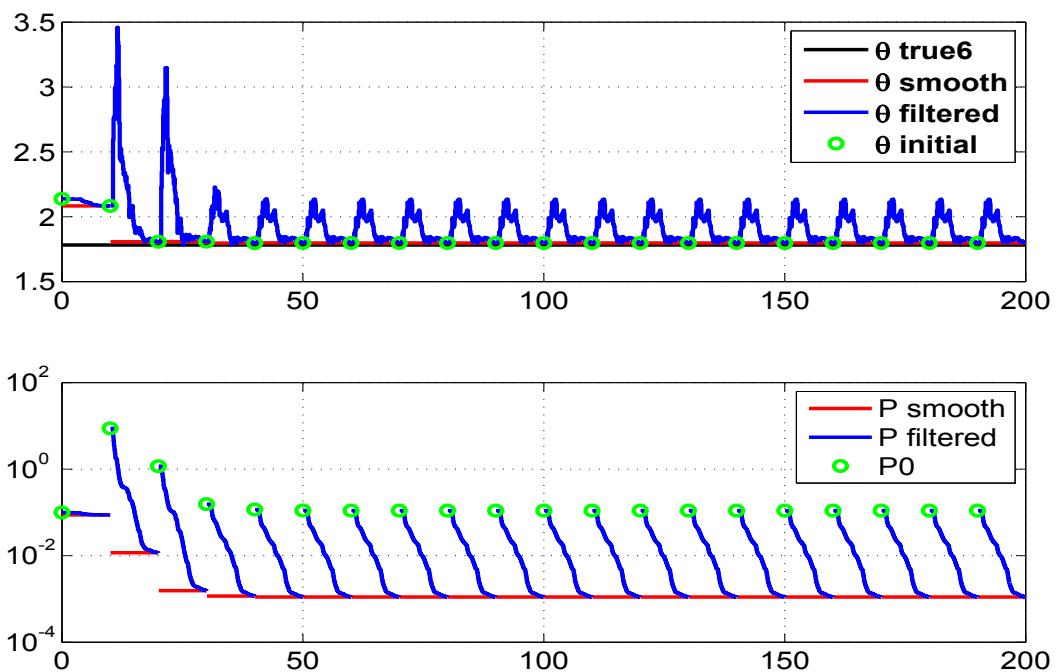
Figure 4.123: The variation of parameter estimate 6 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
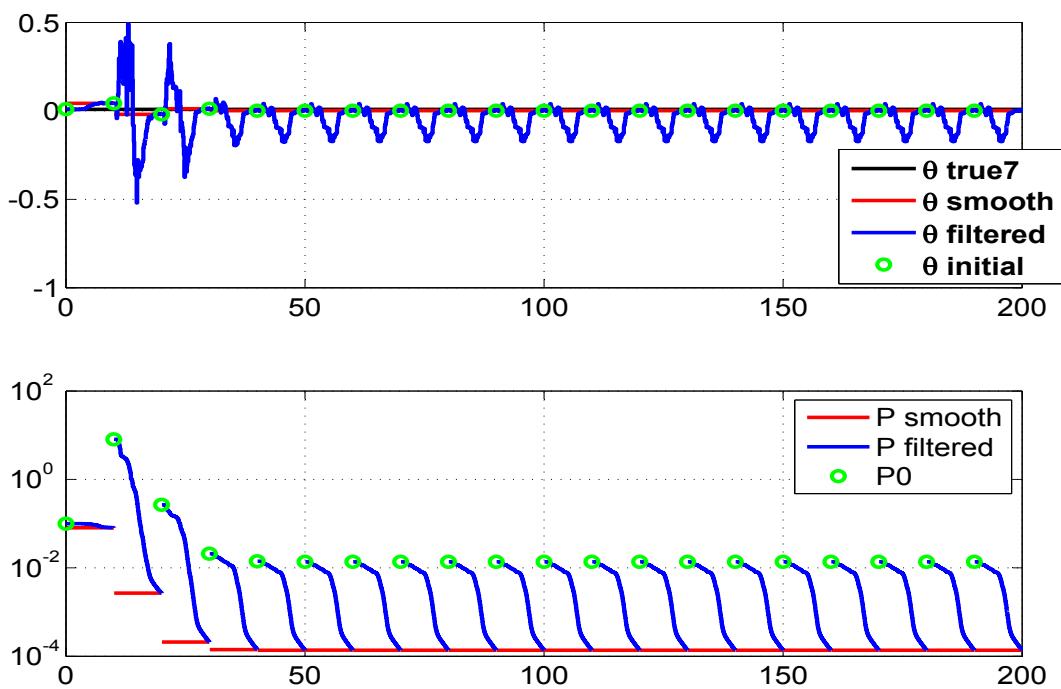
Figure 4.124: The variation of parameter estimate 7 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
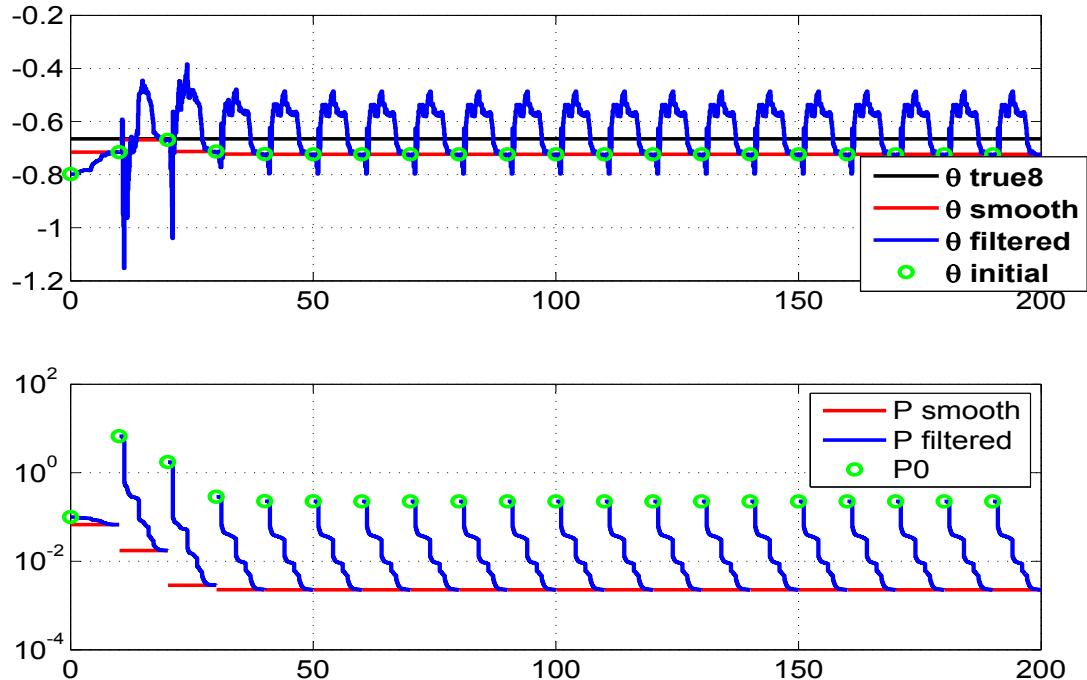
Figure 4.125: The variation of parameter estimate 8 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
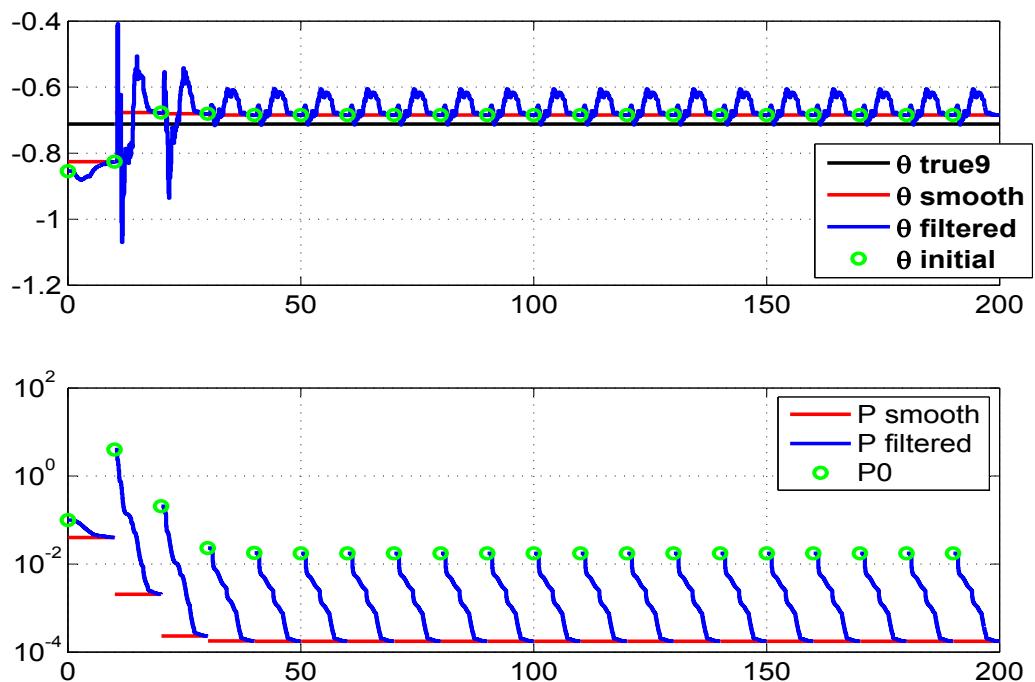
Figure 4.126: The variation of parameter estimate 9 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
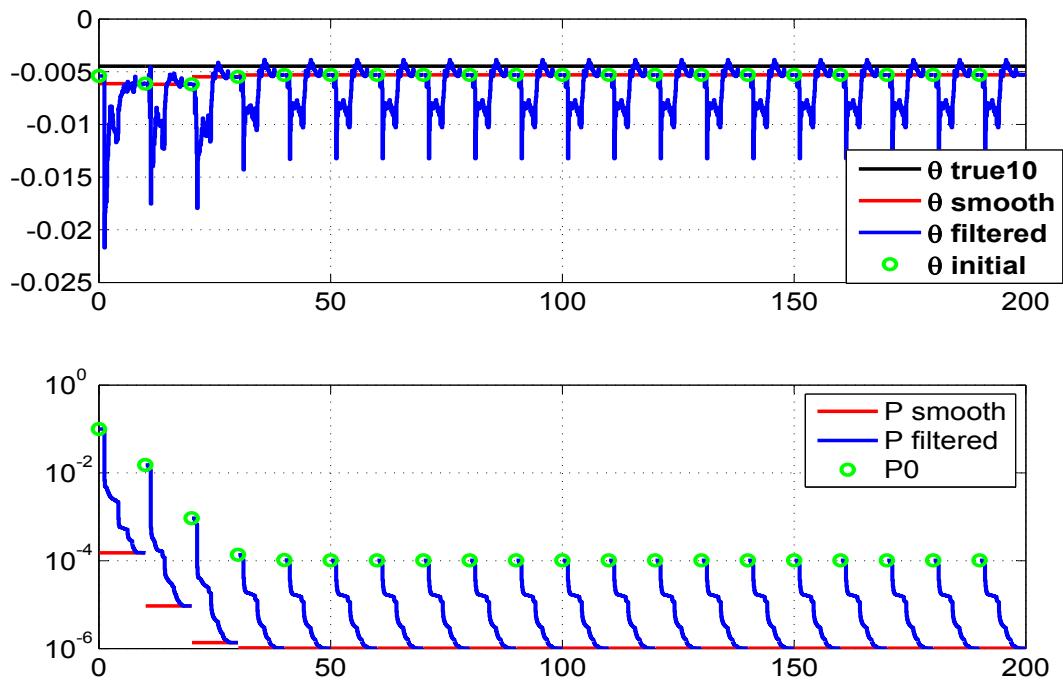
Figure 4.127: The variation of parameter estimate 10 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
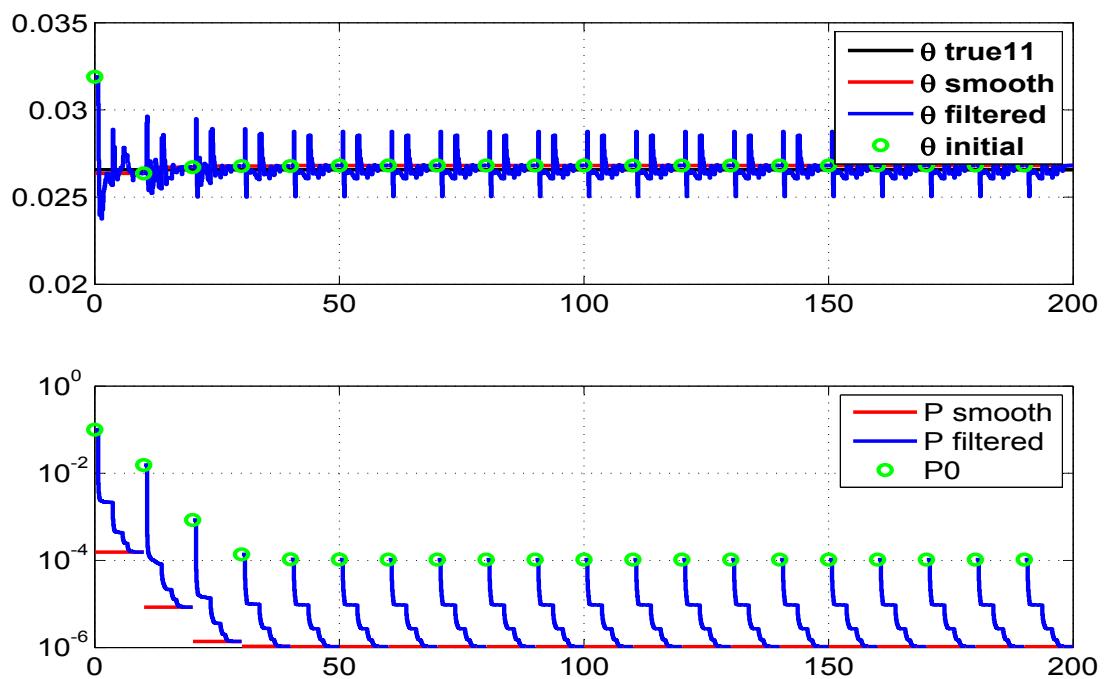
Figure 4.128: The variation of parameter estimate 11 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
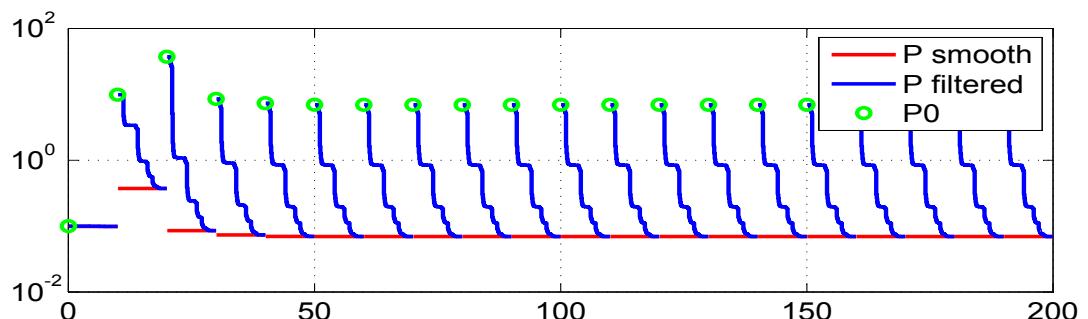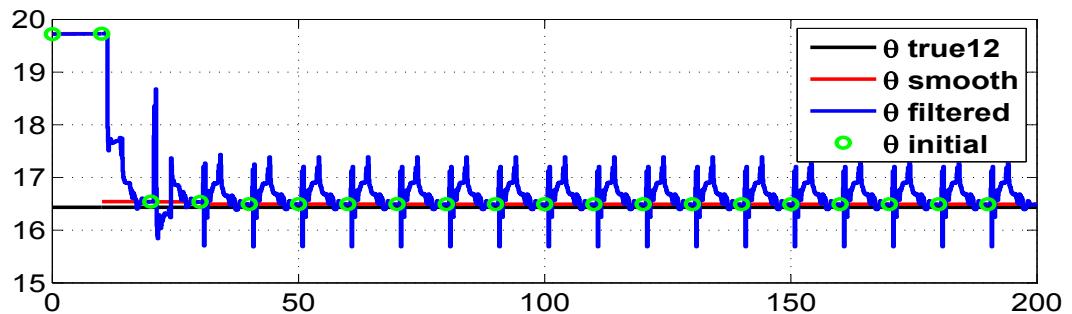
Figure 4.129: The variation of parameter estimate 12 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
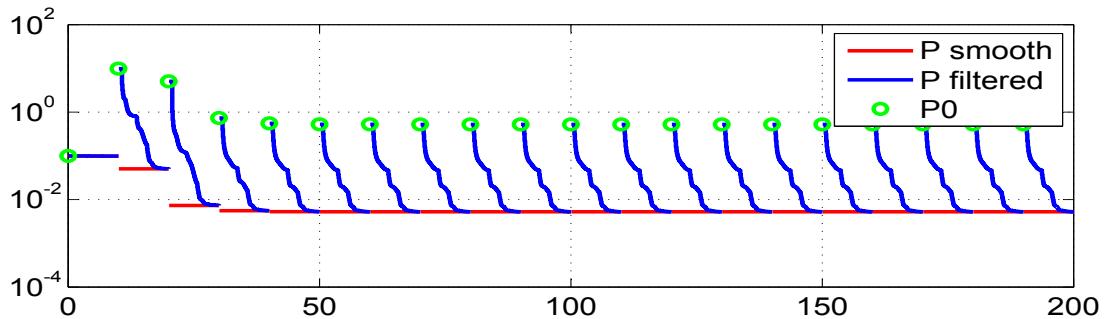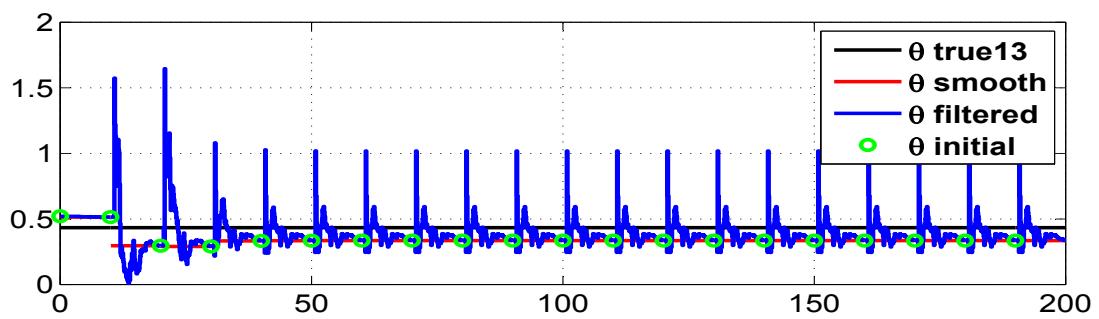
Figure 4.130: The variation of parameter estimate 13 and their filtered and smoothed covariances through (with the time cumulatively) the iterations

186
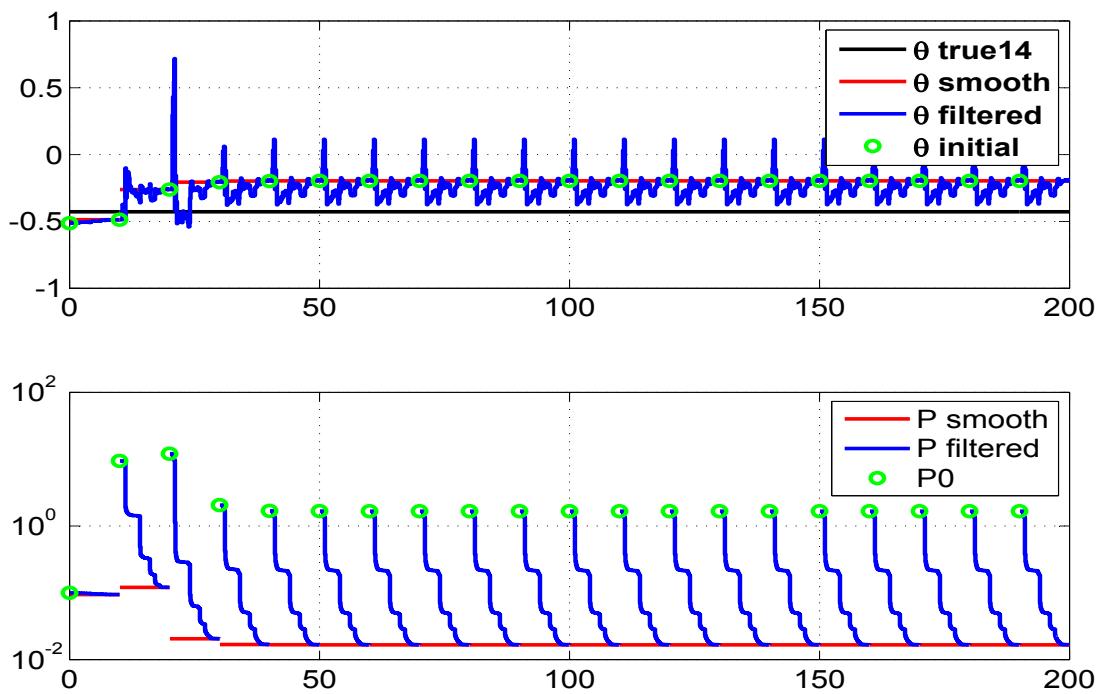
Figure 4.131: The variation of parameter estimate 14 and their filtered and smoothed covariances through (with the time cumulatively) the iterations
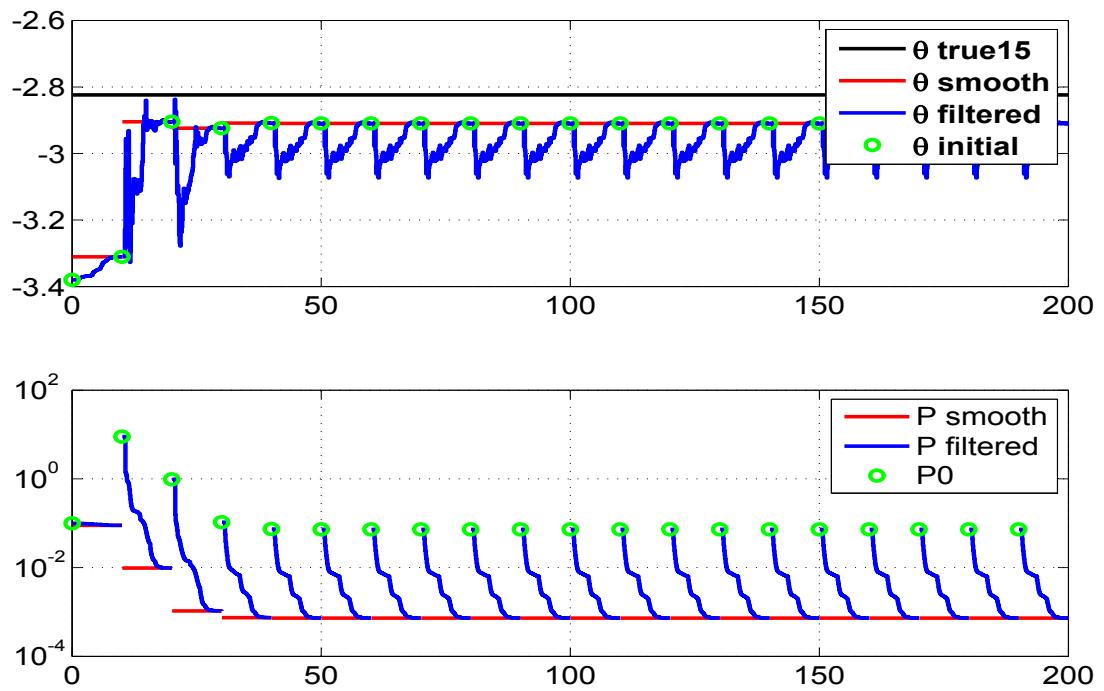
Figure 4.132: The variation of parameter estimate 15 and their filtered and smoothed covariances through (with the time cumulatively) the iterations

Figure 4.133: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
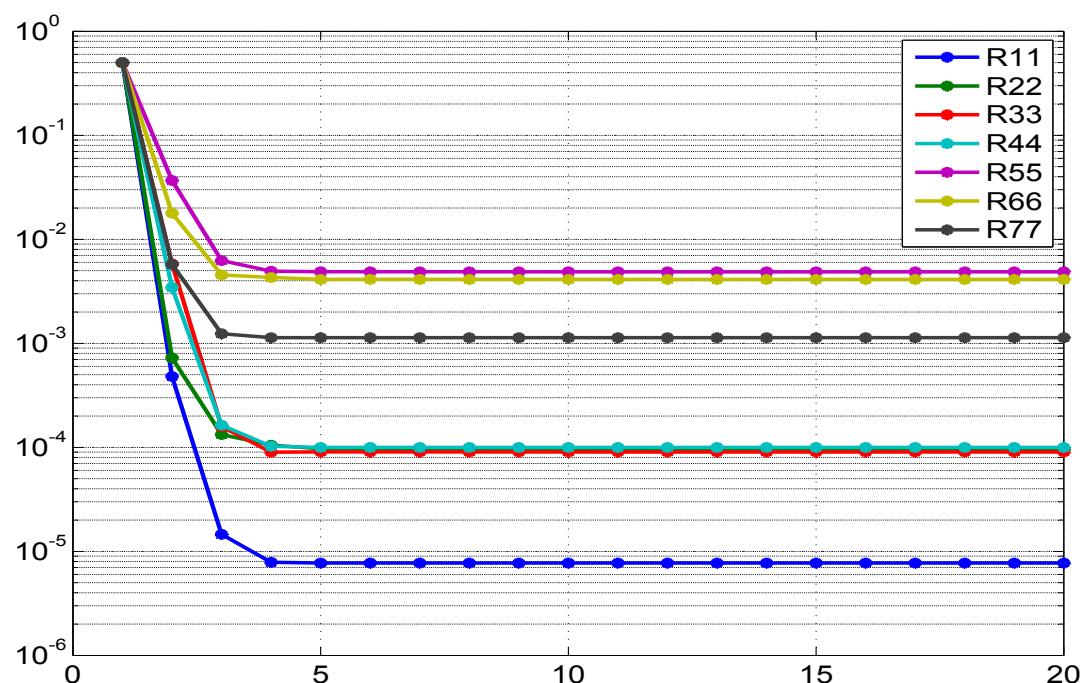
Figure 4.134: Variation of $\mathbf{R}$ with iterations
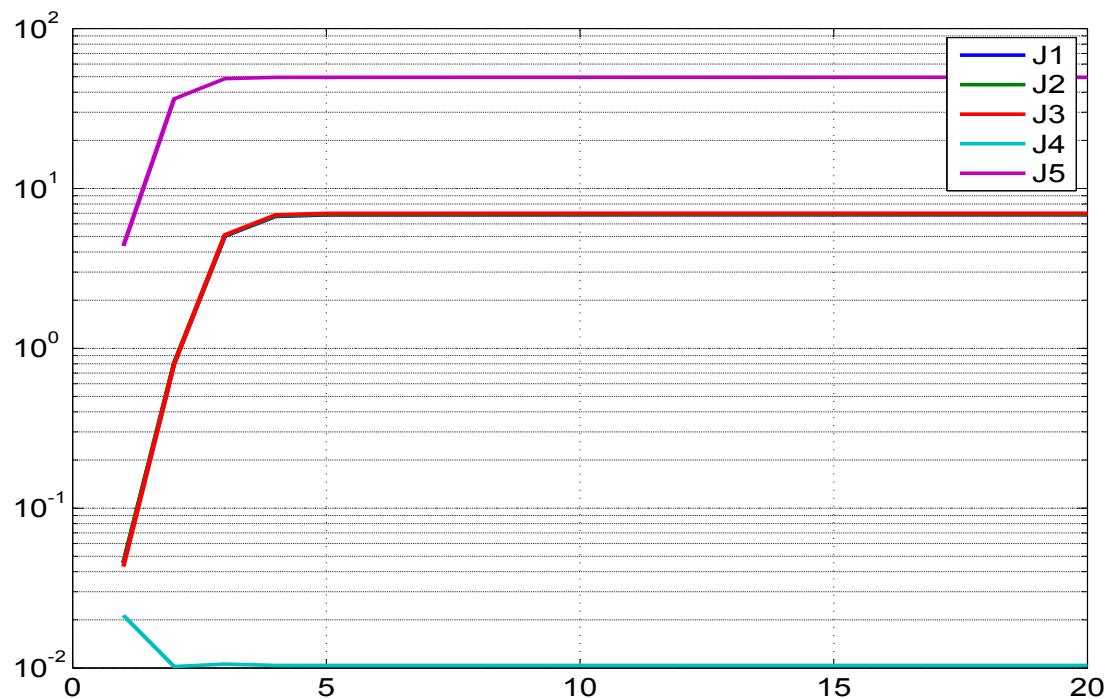
Figure 4.135: Variation of different costs (**J1-J5**) with iterations
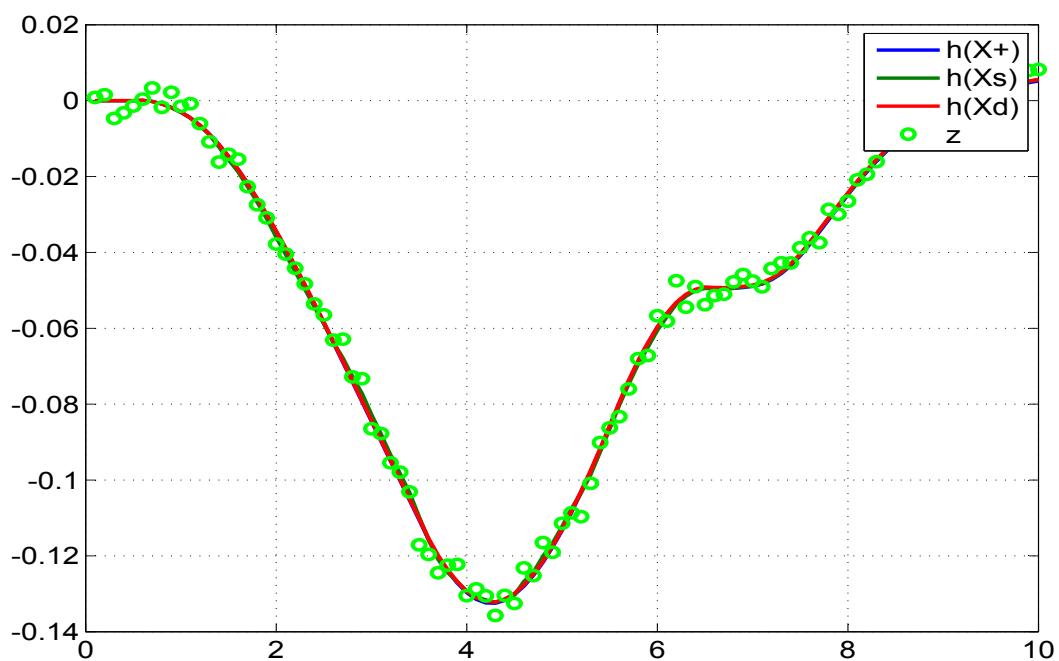
Figure 4.136: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1
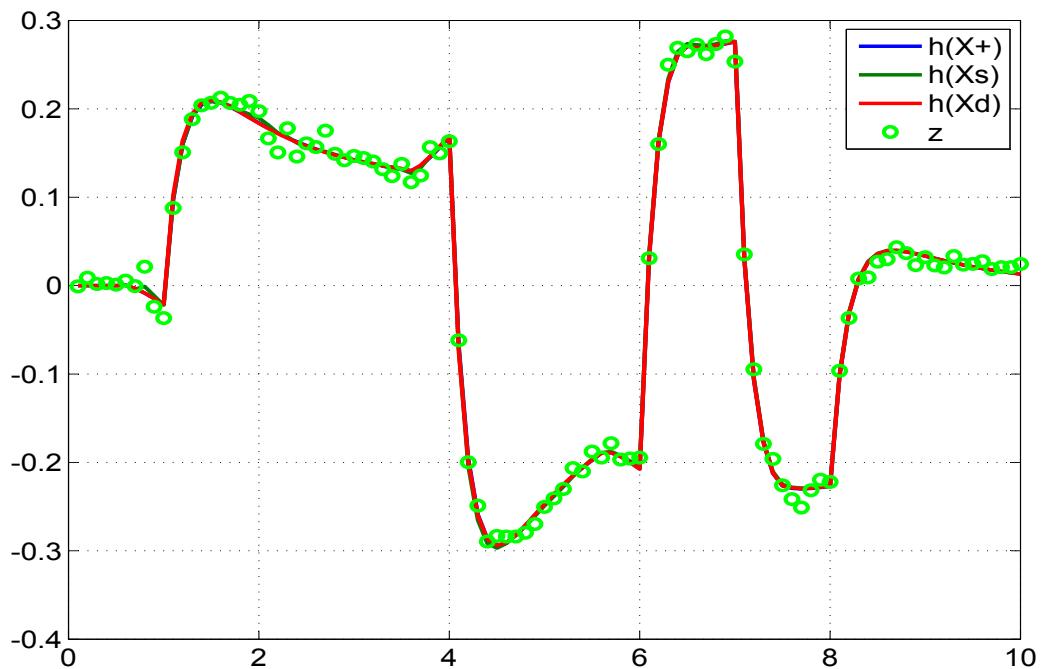
Figure 4.137: Comparison of the predicted dynamics, posterior, smoothed and the measurement 2
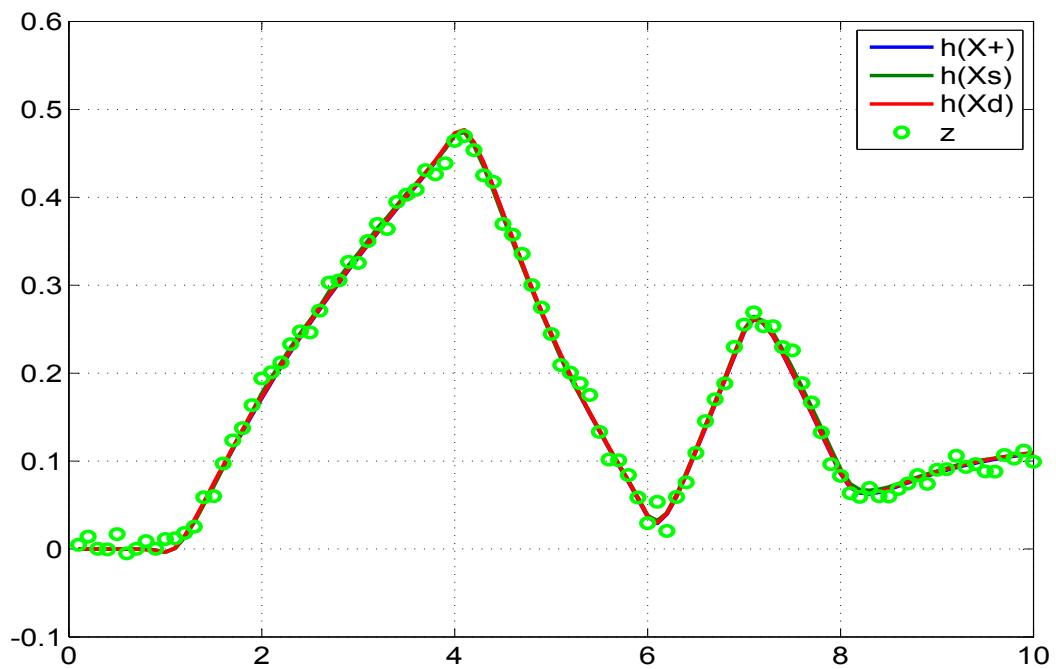
Figure 4.138: Comparison of the predicted dynamics, posterior, smoothed and the measurement 3
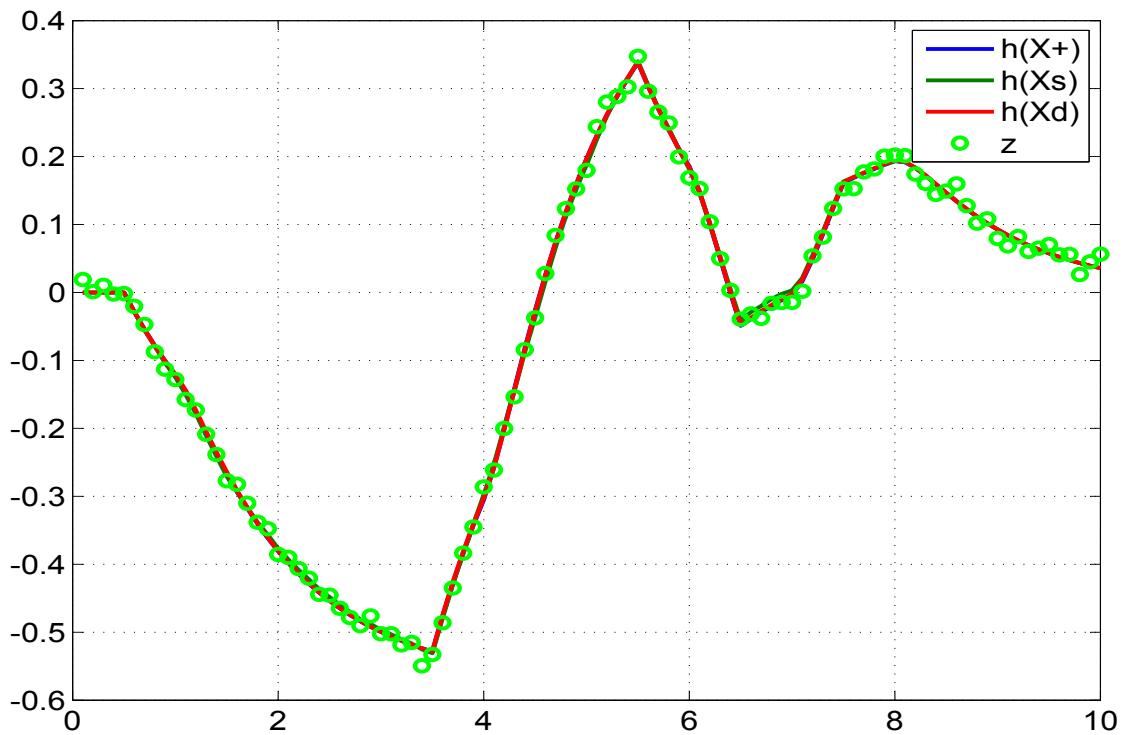
Figure 4.139: Comparison of the predicted dynamics, posterior, smoothed
and the measurement 4
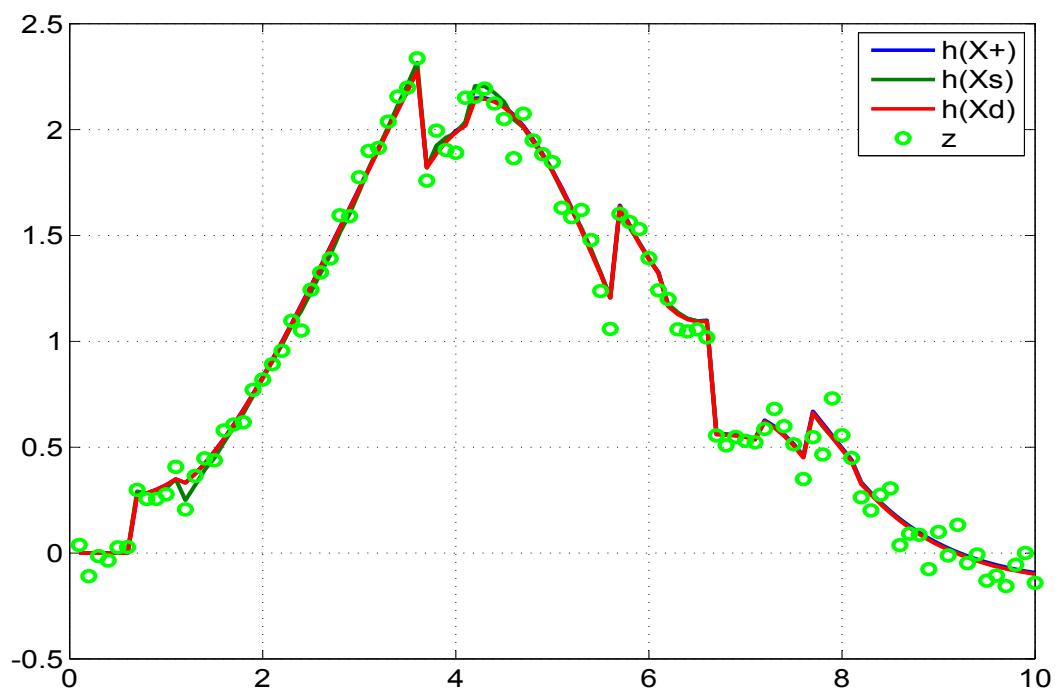
Figure 4.140: Comparison of the predicted dynamics, posterior, smoothed
and the measurement 5
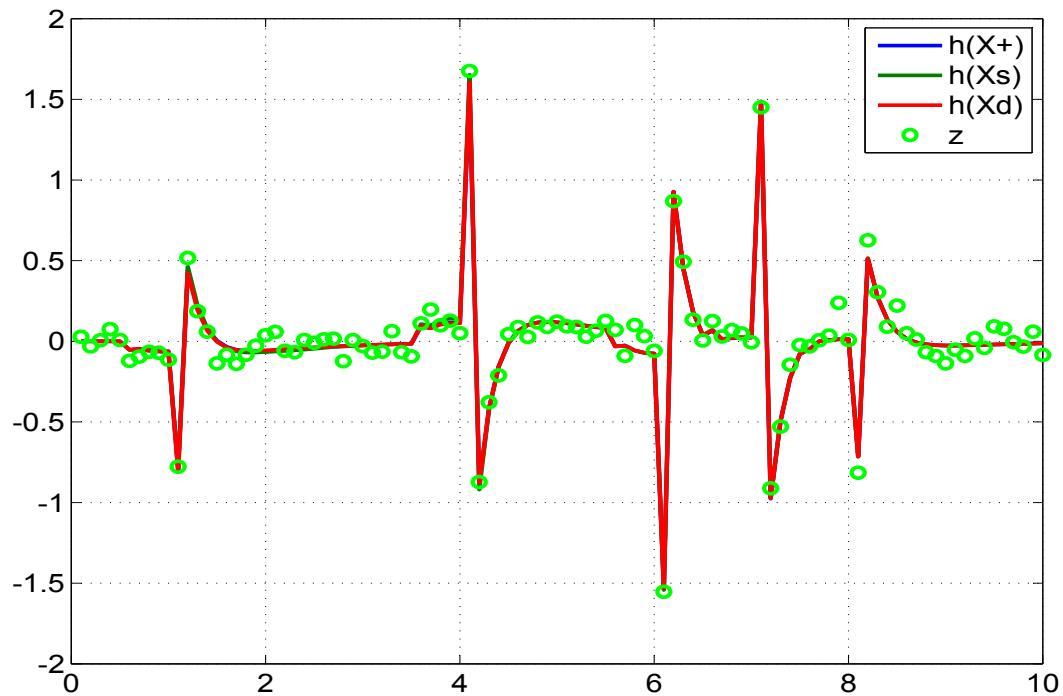
Figure 4.141: Comparison of the predicted dynamics, posterior, smoothed
and the measurement 6
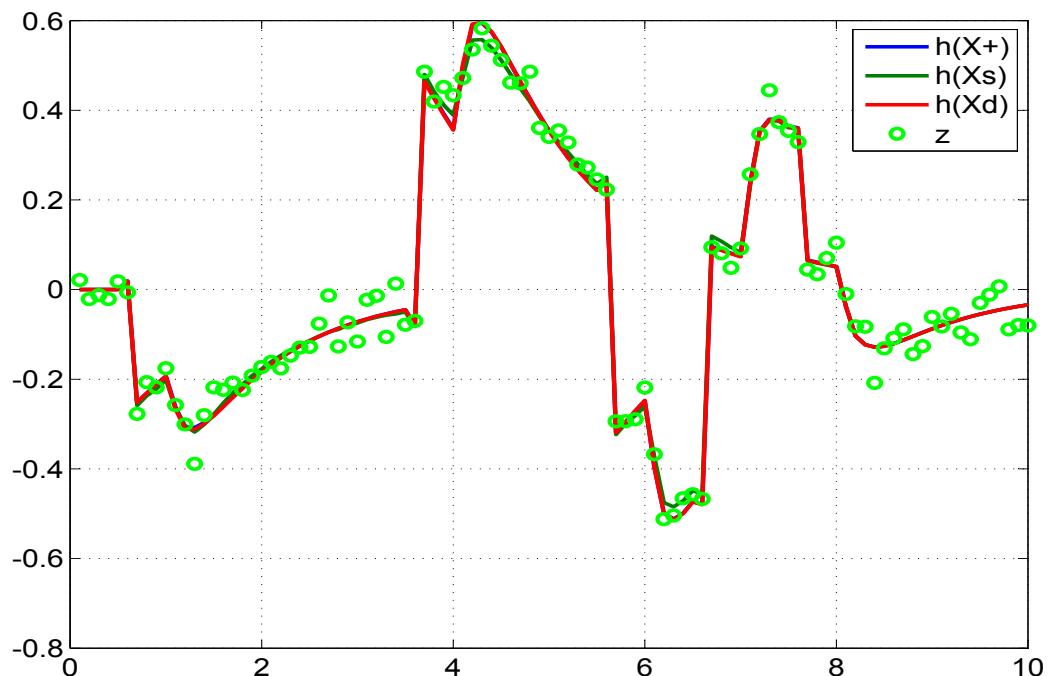
Figure 4.142: Comparison of the predicted dynamics, posterior, smoothed
and the measurement 7
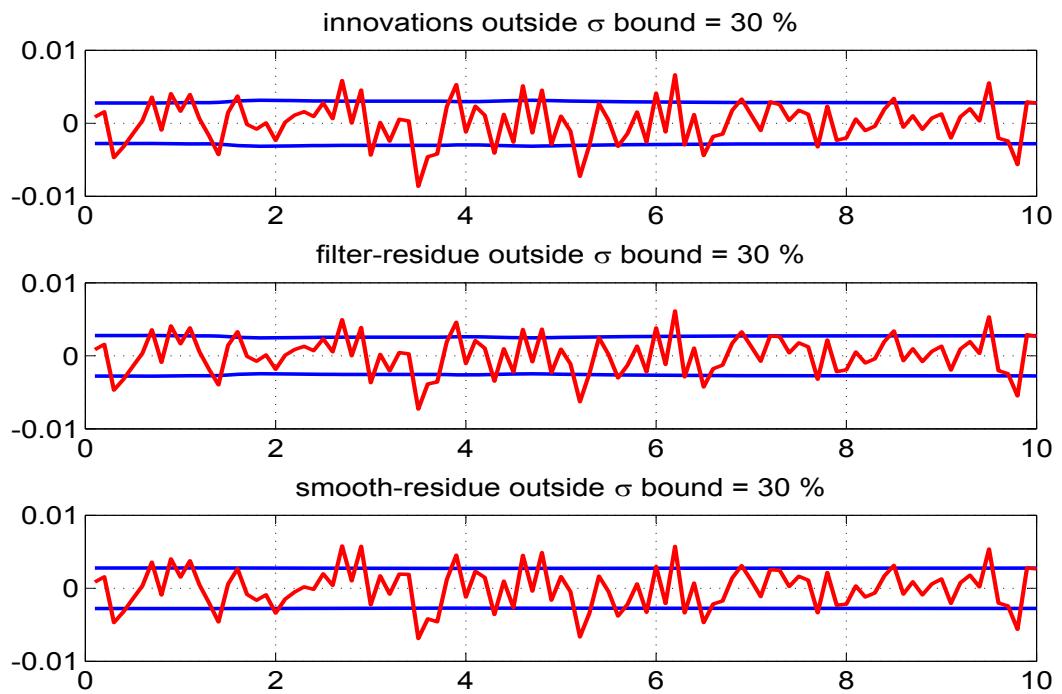
Figure 4.143: The innovations, filtered residue and smoothed residue

corresponding to measurement 1
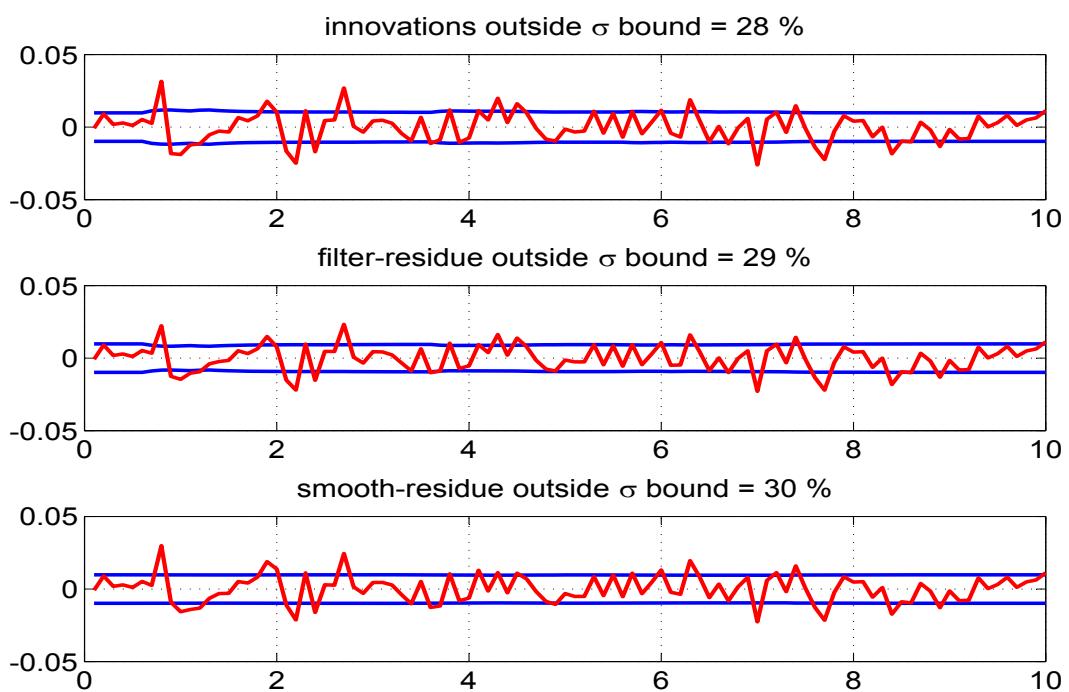
Figure 4.144: The innovations, filtered residue and smoothed residue

corresponding to measurement 2
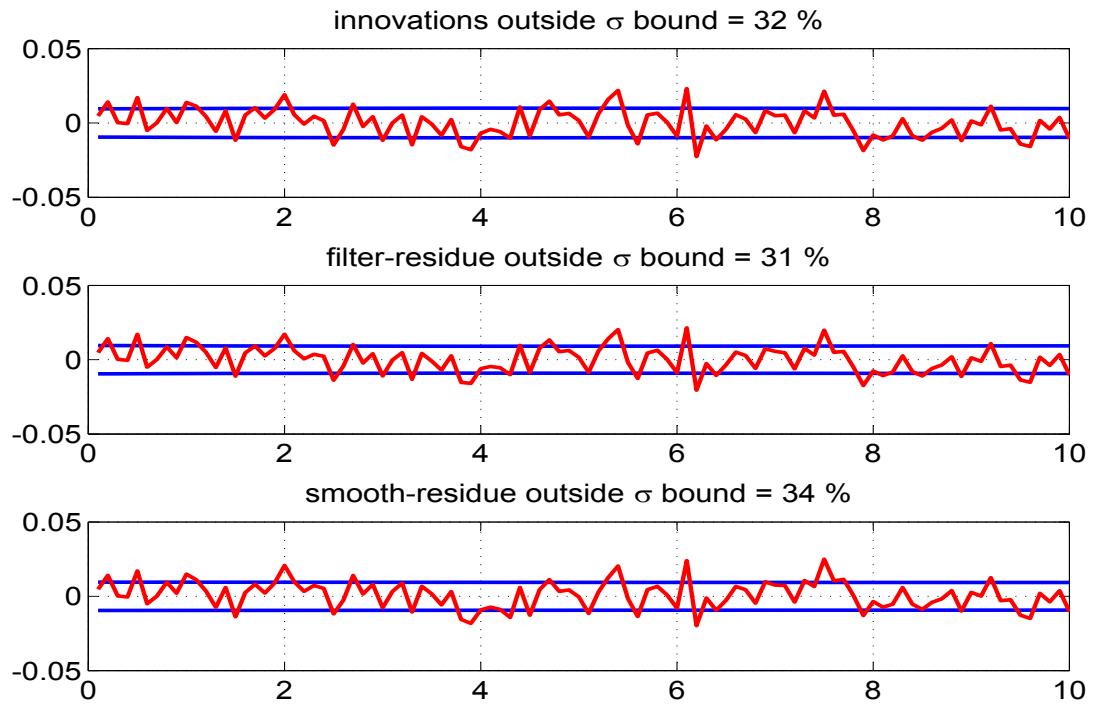
Figure 4.145: The innovations, filtered residue and smoothed residue

corresponding to measurement 3
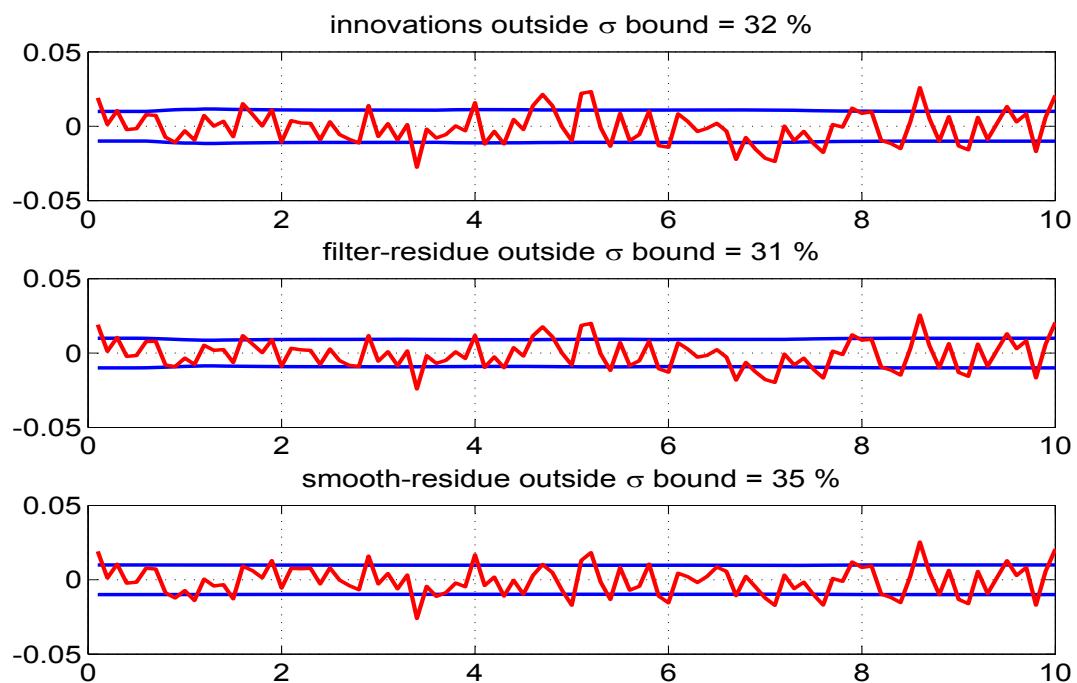
Figure 4.146: The innovations, filtered residue and smoothed residue

corresponding to measurement 4
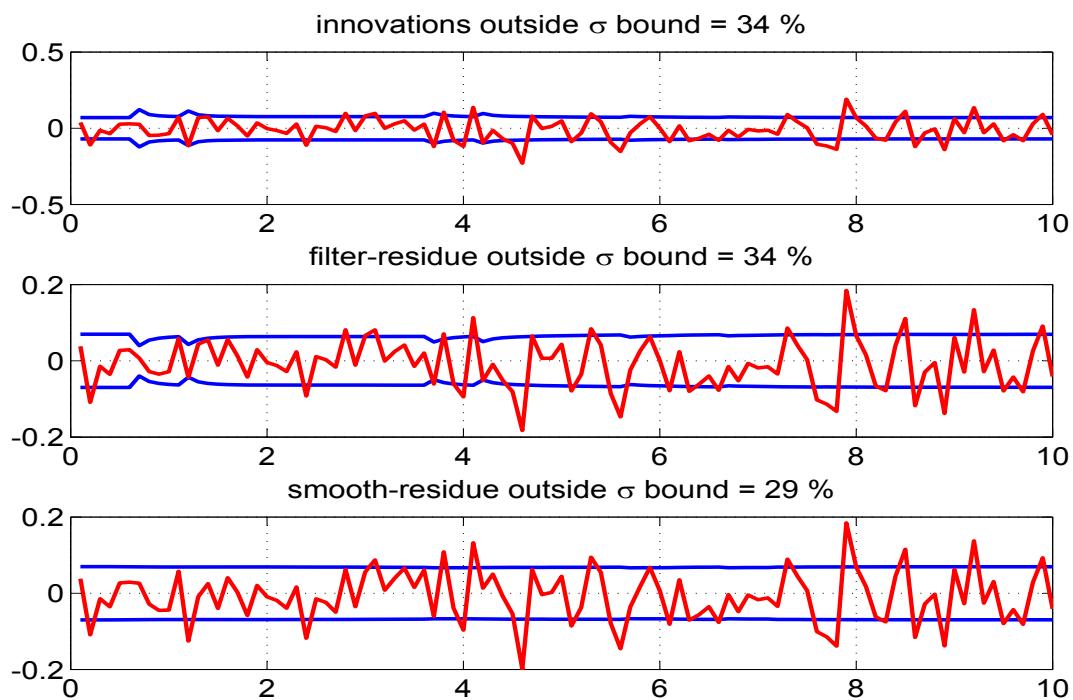
Figure 4.147: The innovations, filtered residue and smoothed residue

corresponding to measurement 5
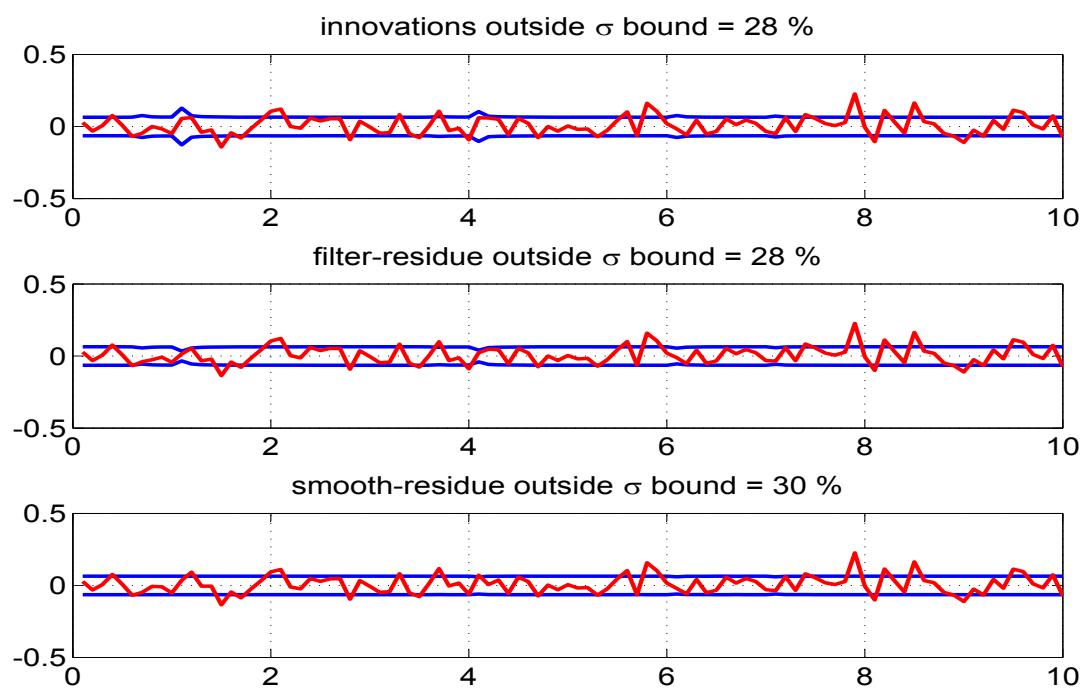
Figure 4.148: The innovations, filtered residue and smoothed residue

corresponding to measurement 6
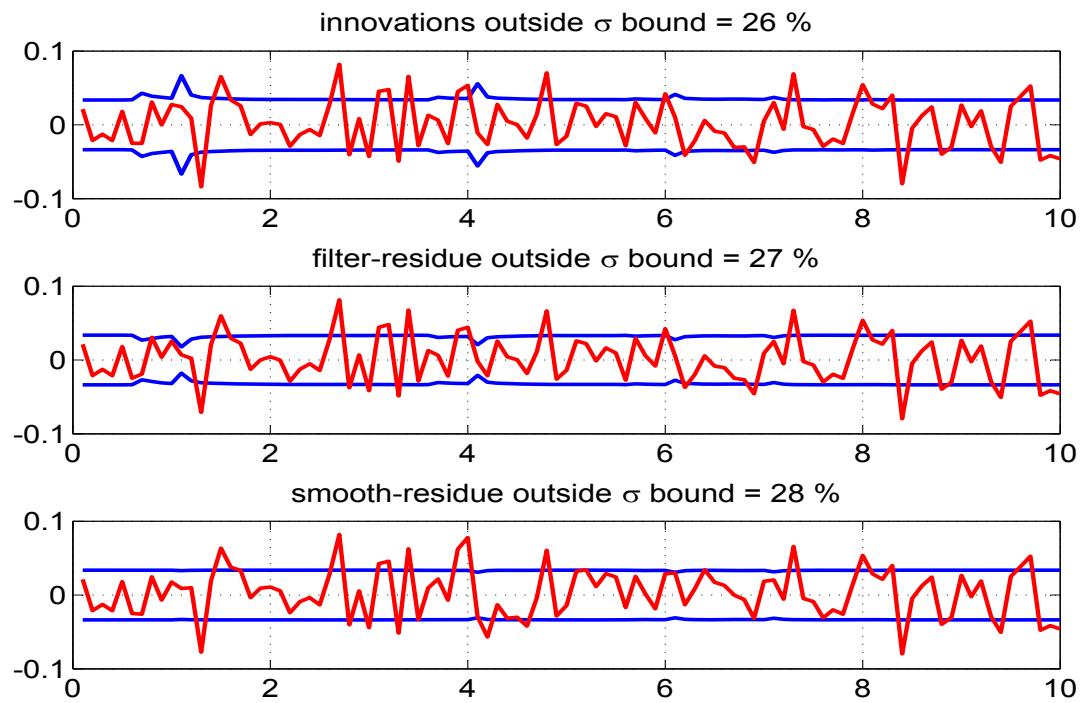
Figure 4.149: The innovations, filtered residue and smoothed residue

corresponding to measurement 7
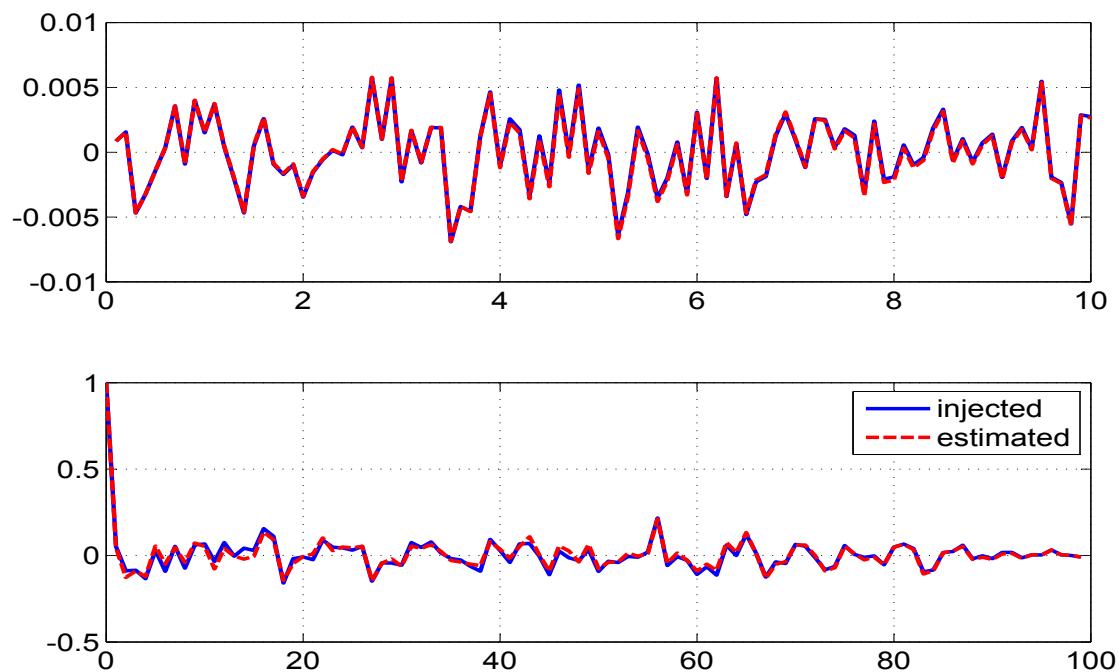
Figure 4.150: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 1
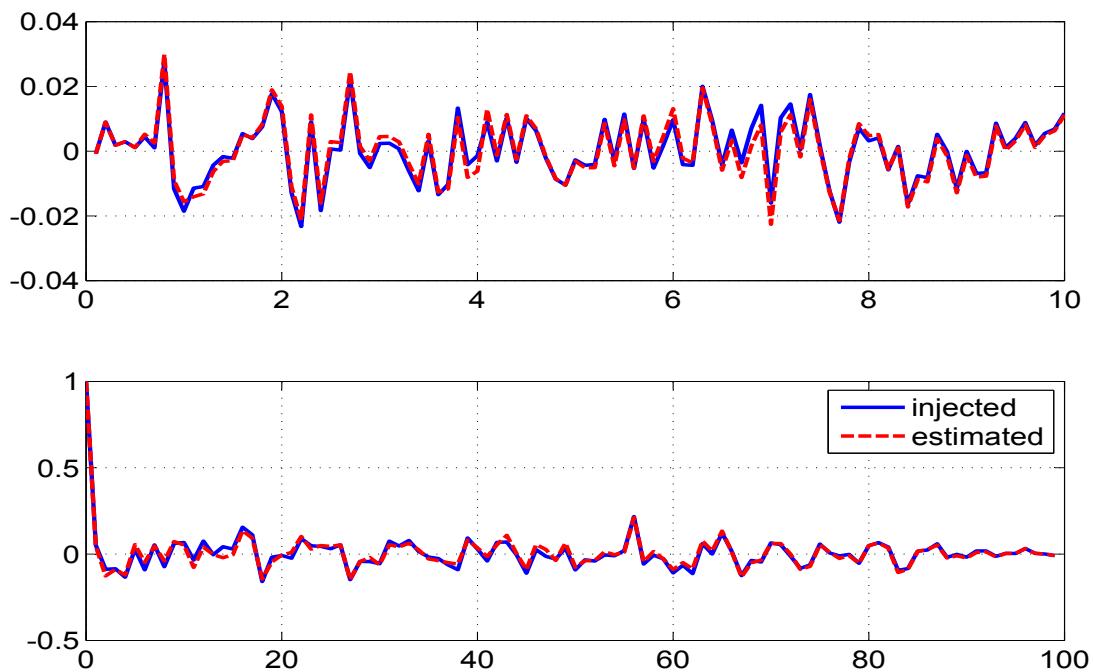
Figure 4.151: Time variation of injected and estimated measurement noise (top) and their autocorrelation (bottom) for measurement 2
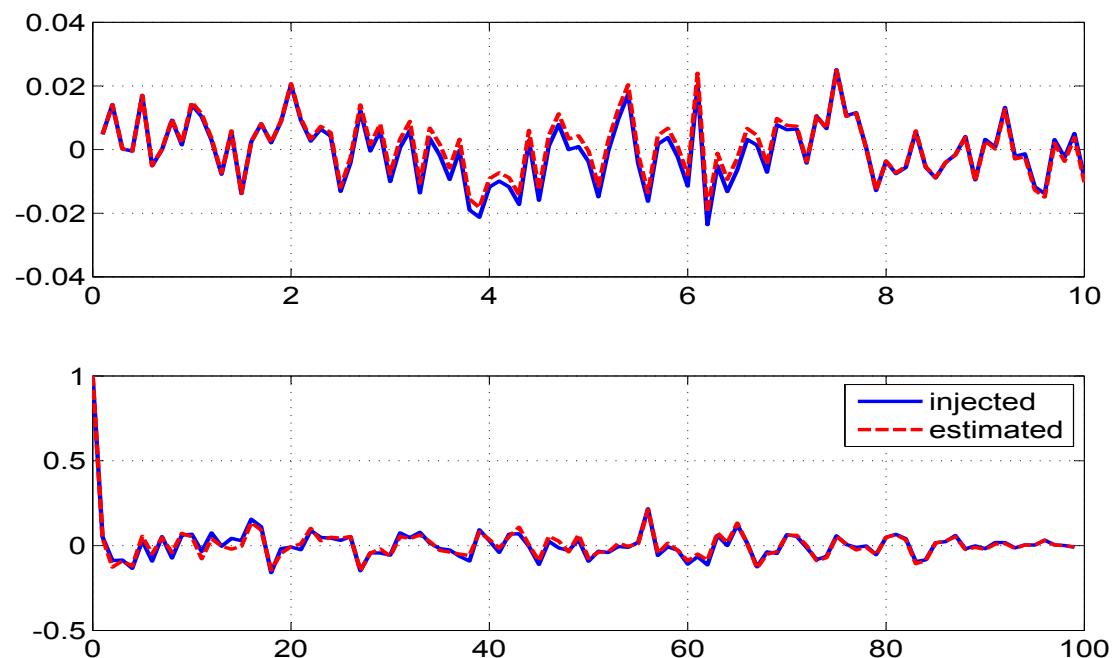
Figure 4.152: Time variation of injected and estimated measurement noise (top) and their autocorrelation (bottom) for measurement 3
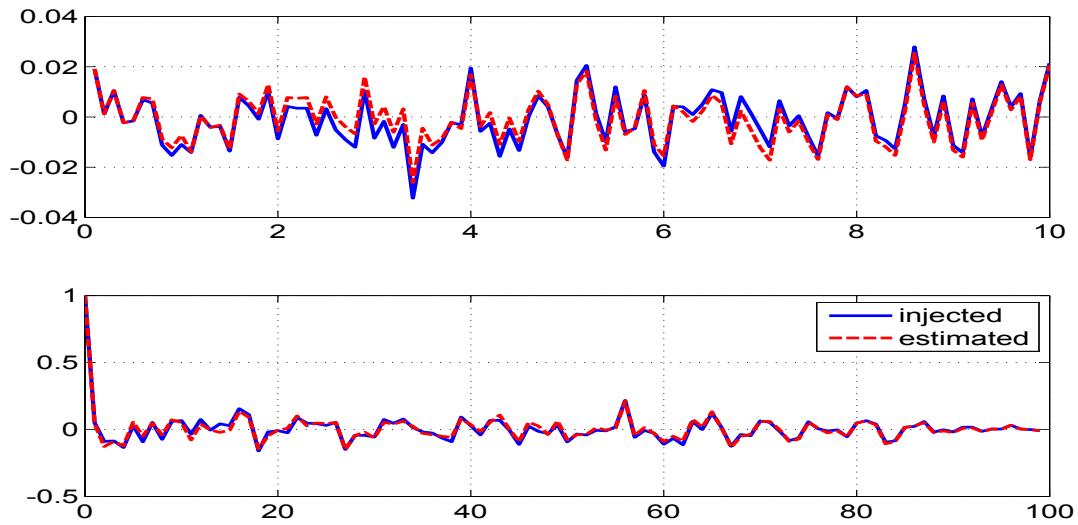
Figure 4.153: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 4
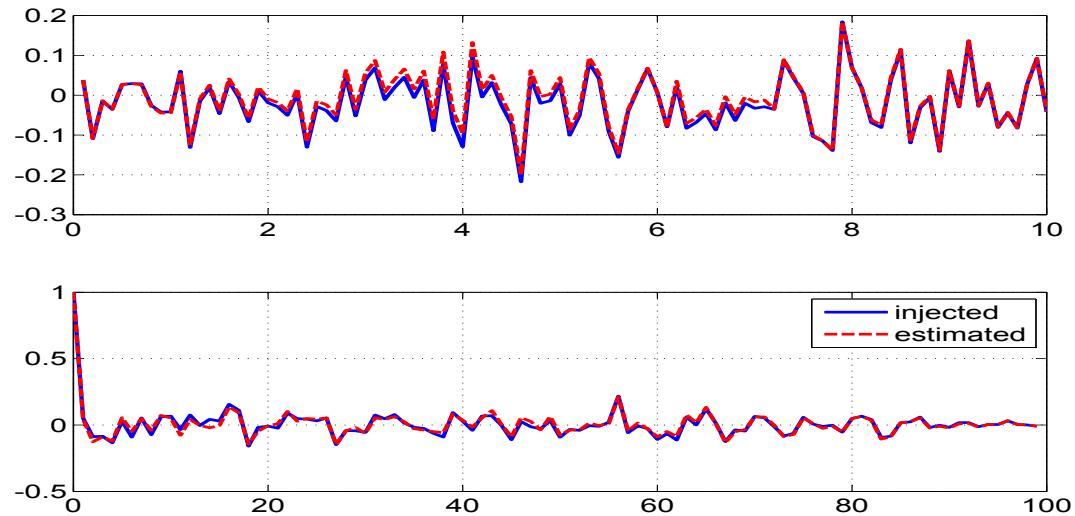
Figure 4.154: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 5
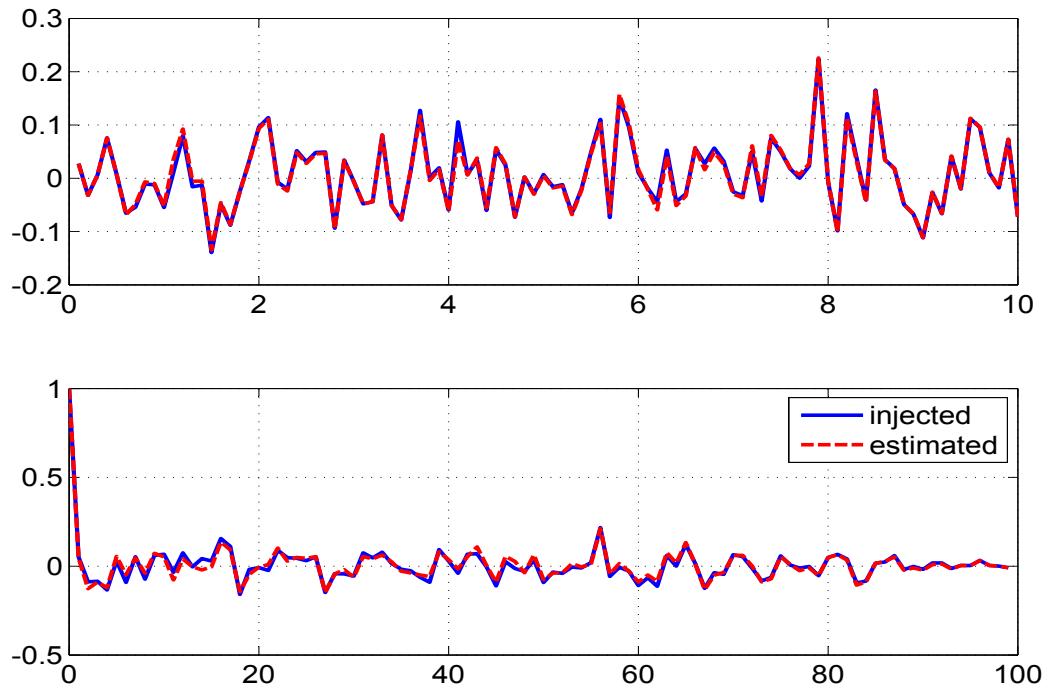
Figure 4.155: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 6
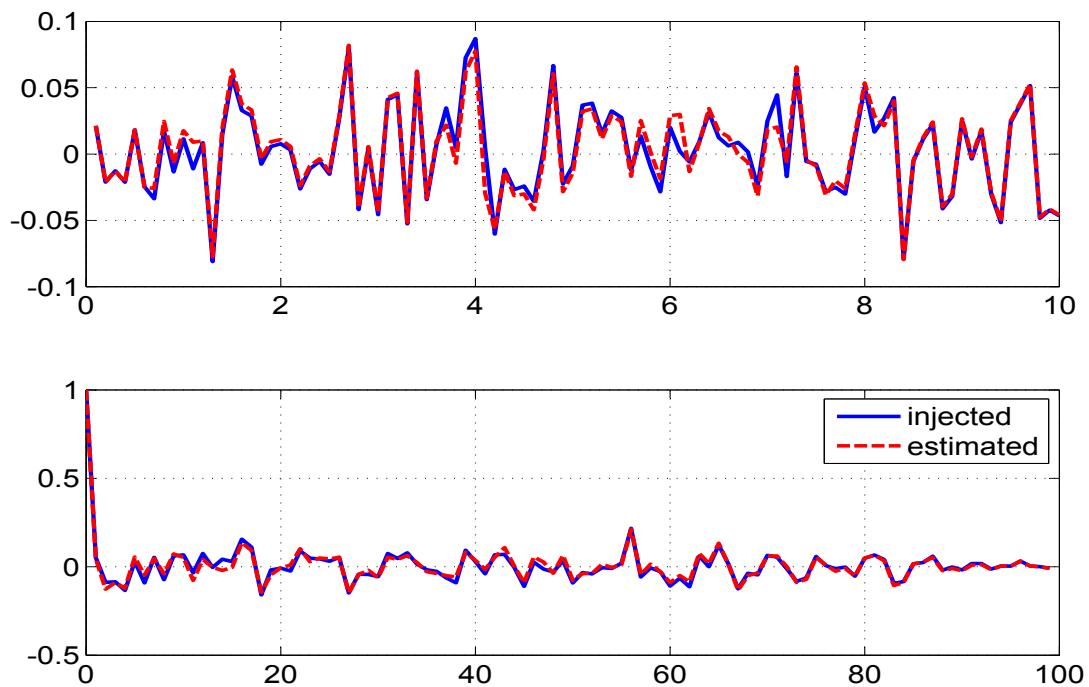
Figure 4.156: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 7

4.5.5 Lateral Motion of Aircraft System Figures (Q > 0)
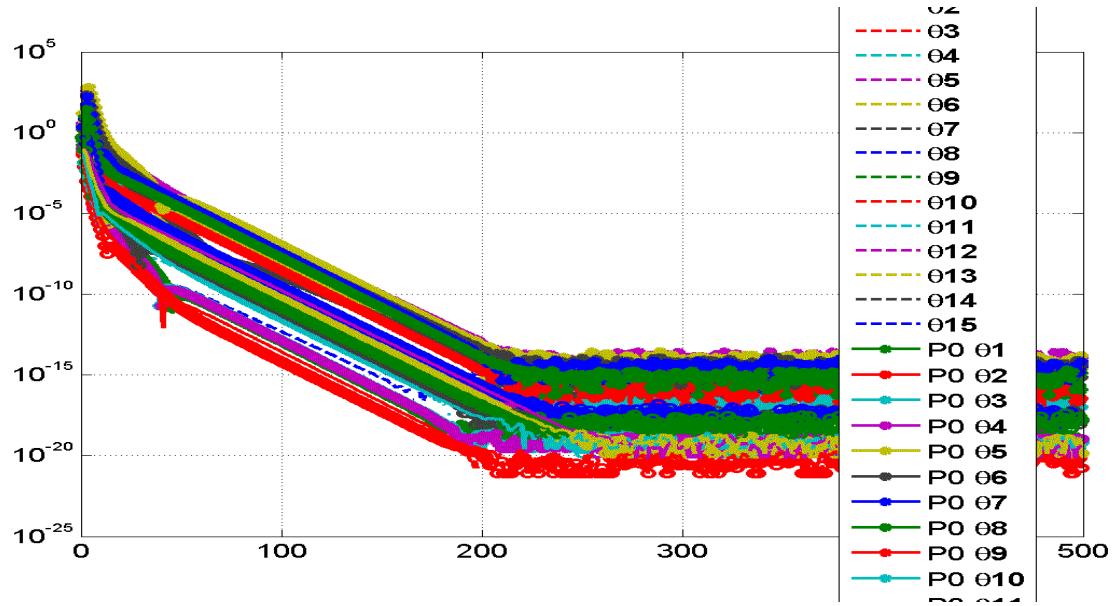
Figure 4.157: The absolute difference between the iterated and final values with 500 iterations
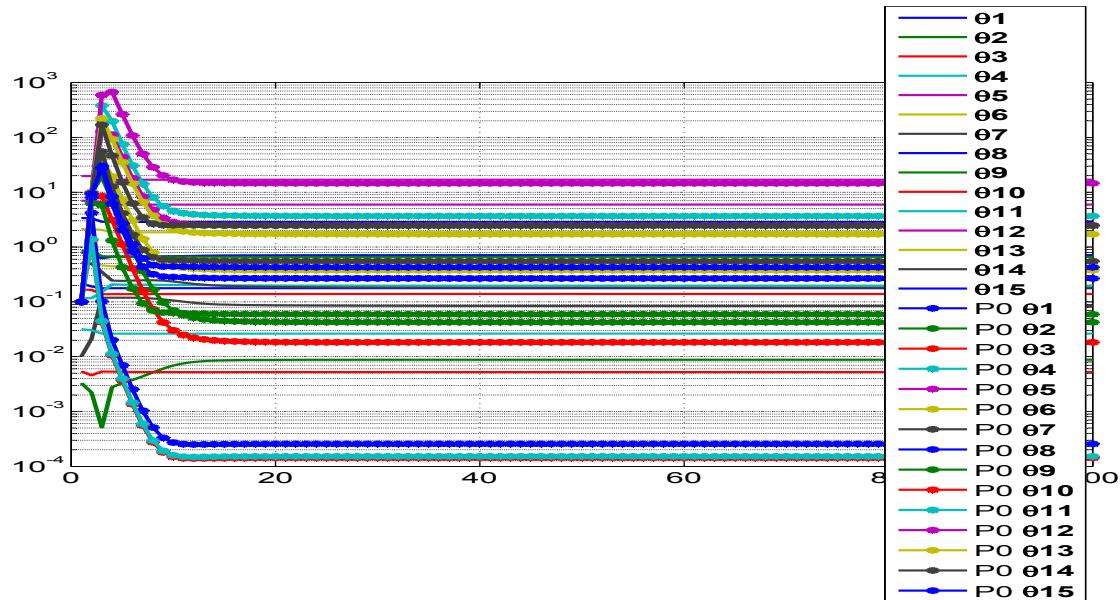
Figure 4.158: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
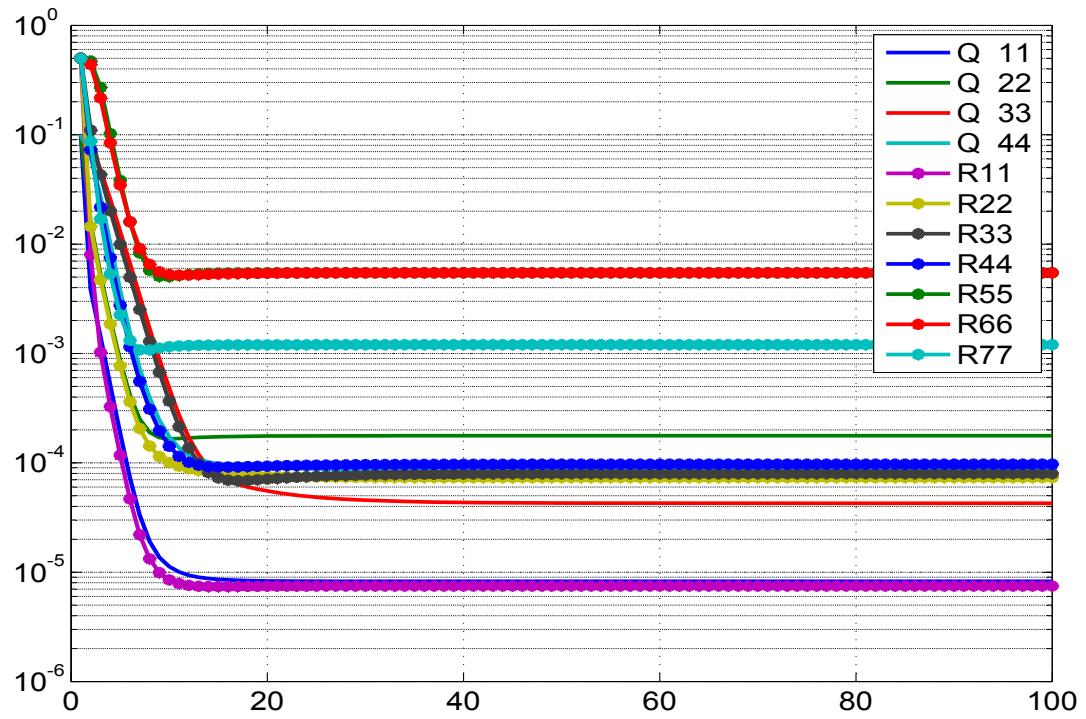
Figure 4.159: Variation of **Q** and **R** with iterations
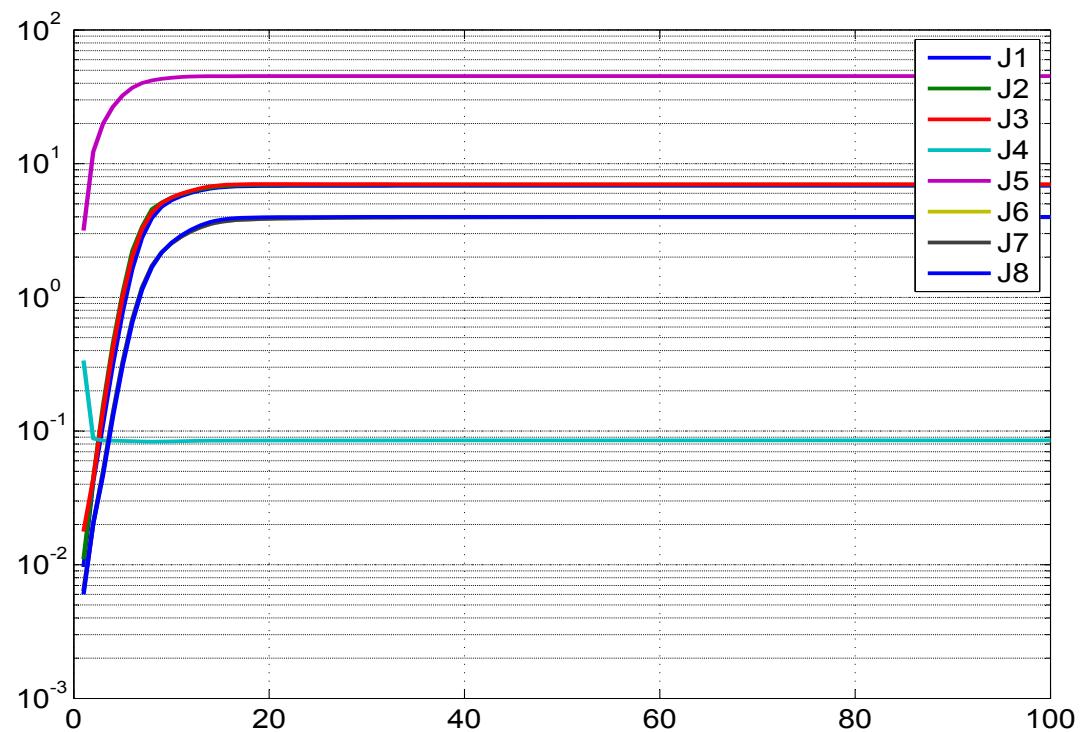
Figure 4.160: Variation of different costs (**J1-J8**) with iterations
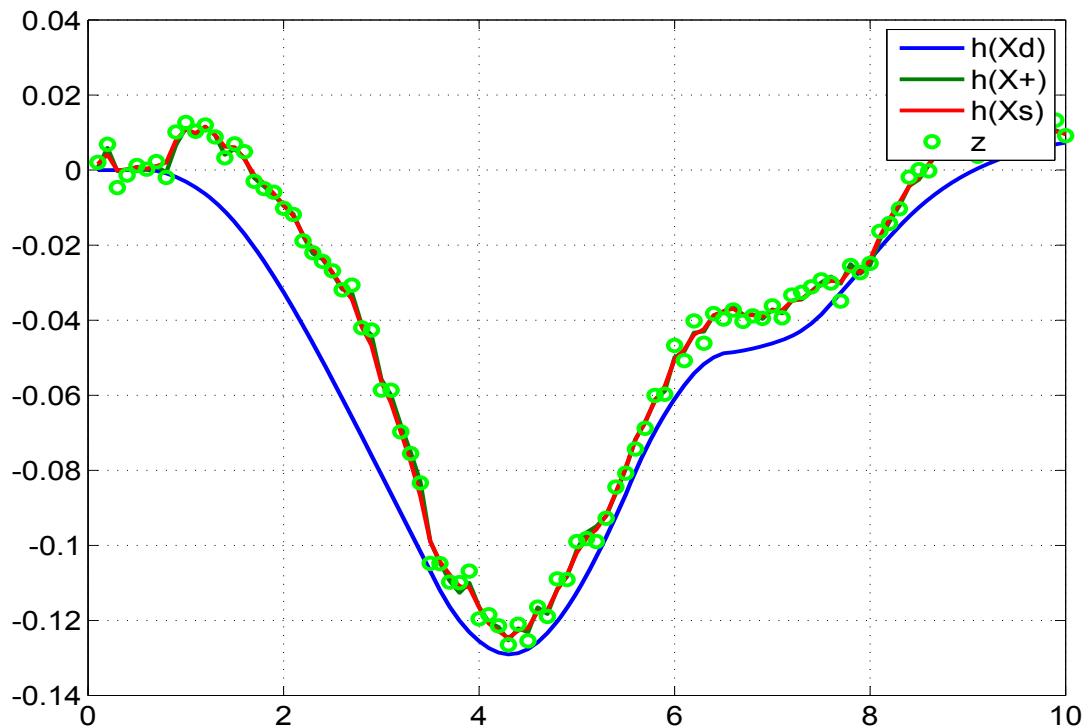
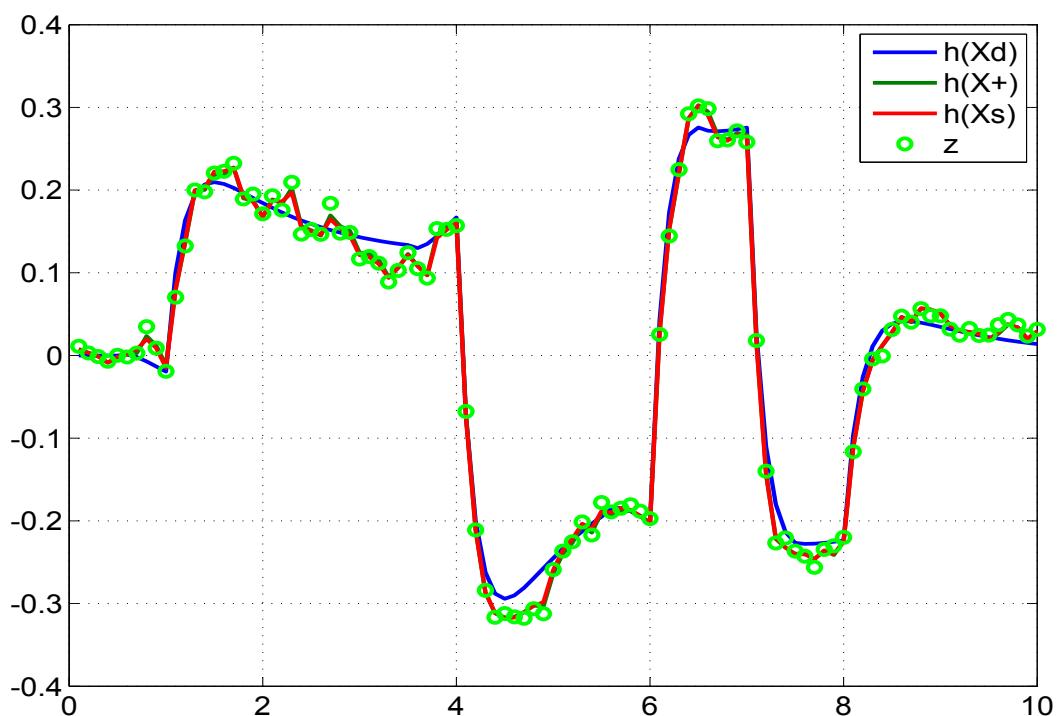Figure 4.161: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1

Figure 4.162: Comparison of the predicted dynamics, posterior, smoothed
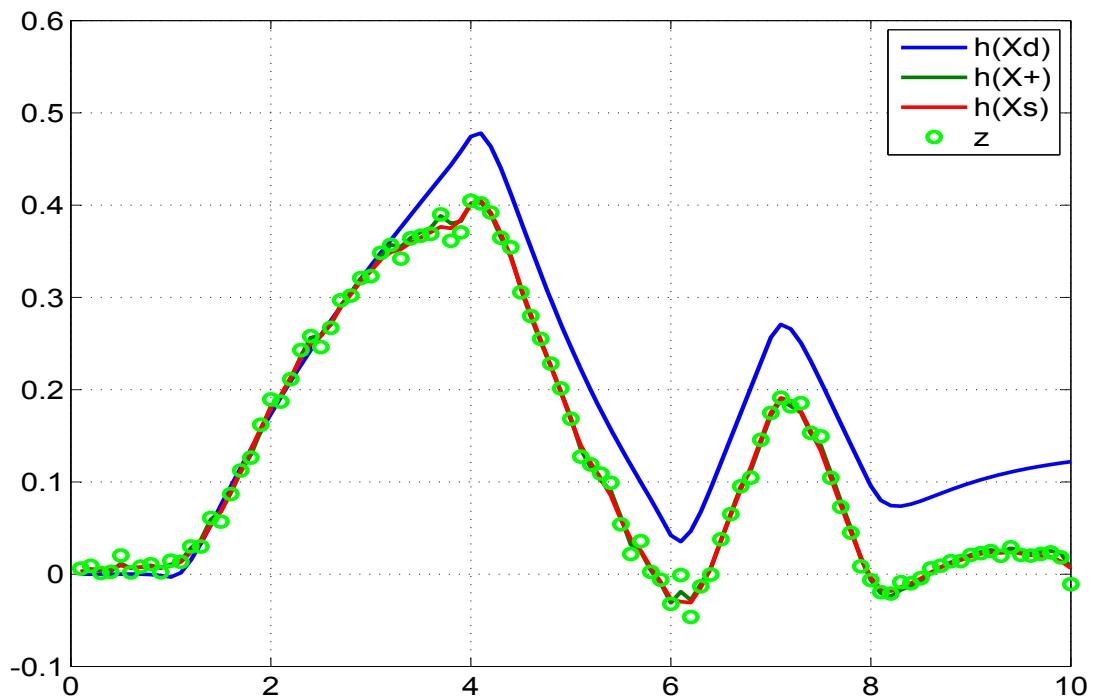
and the measurement 2

Figure 4.163: Comparison of the predicted dynamics, posterior, smoothed
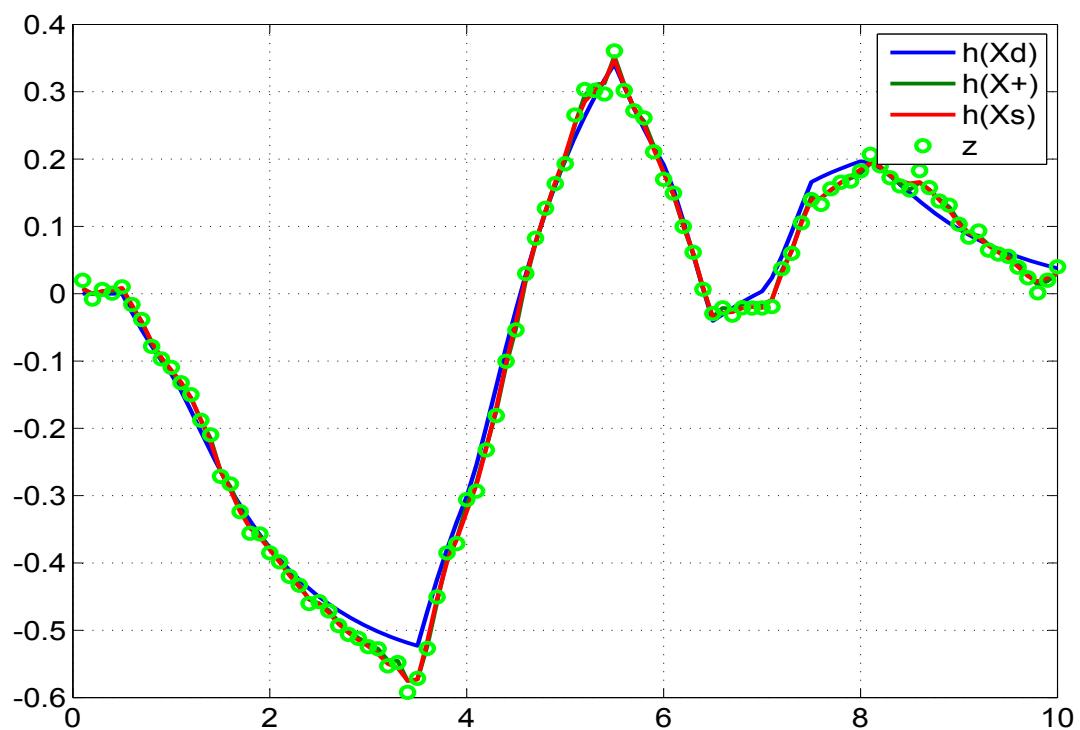
and the measurement 3

Figure 4.164: Comparison of the predicted dynamics, posterior, smoothed
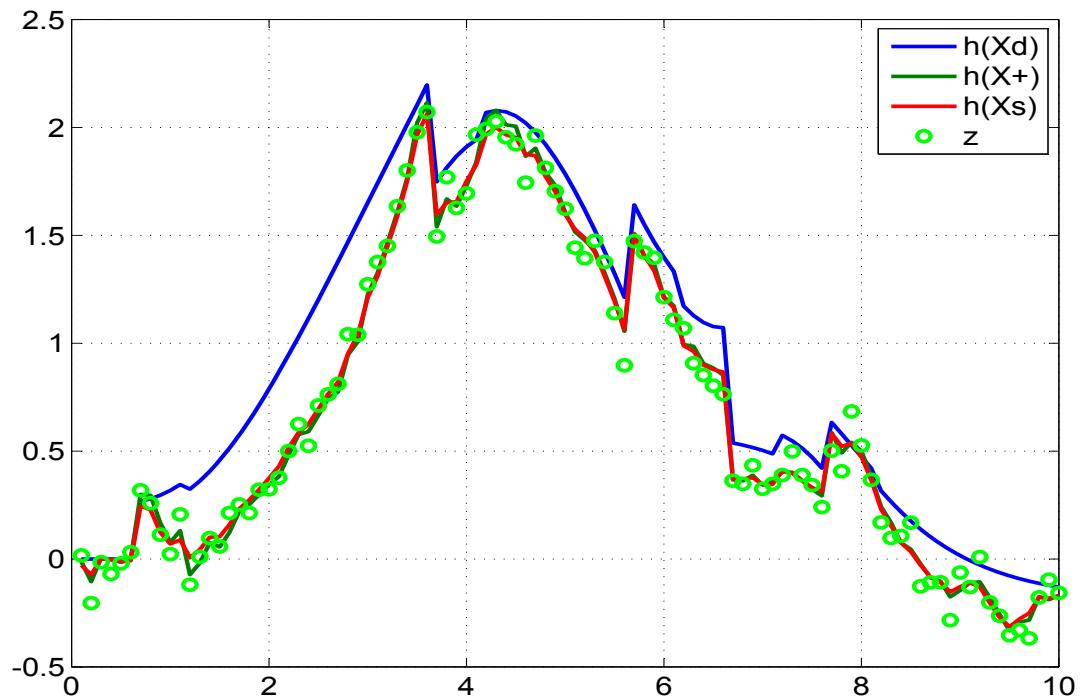
and the measurement 4

Figure 4.165: Comparison of the predicted dynamics, posterior, smoothed
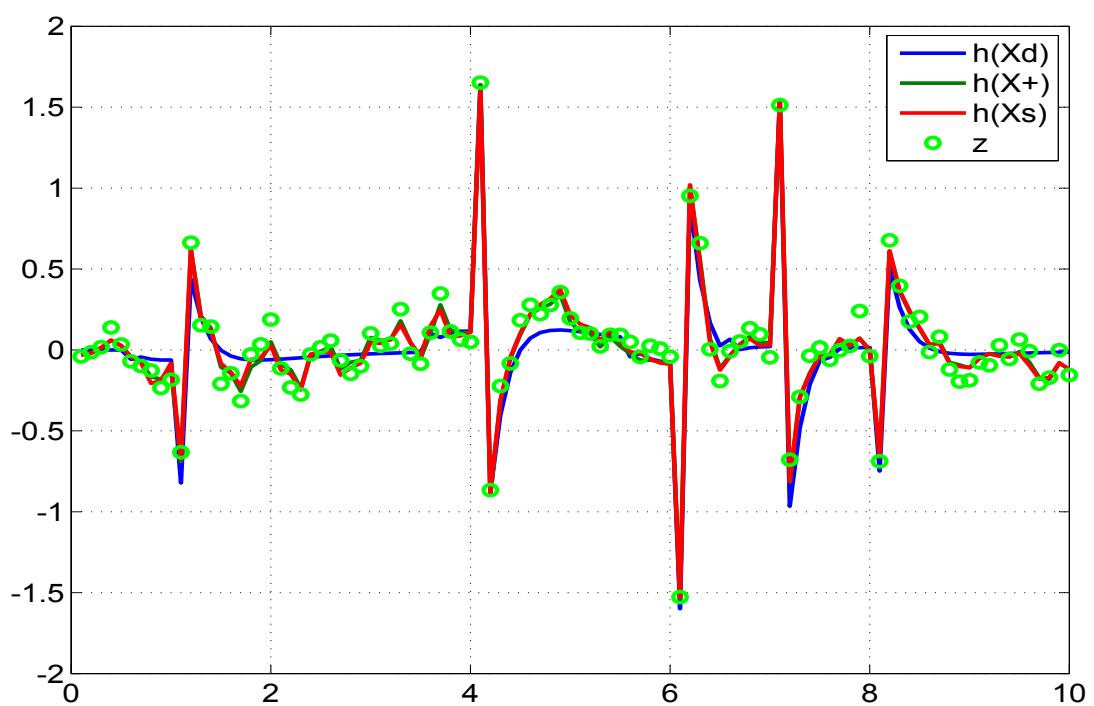
and the measurement 5

Figure 4.166: Comparison of the predicted dynamics, posterior, smoothed
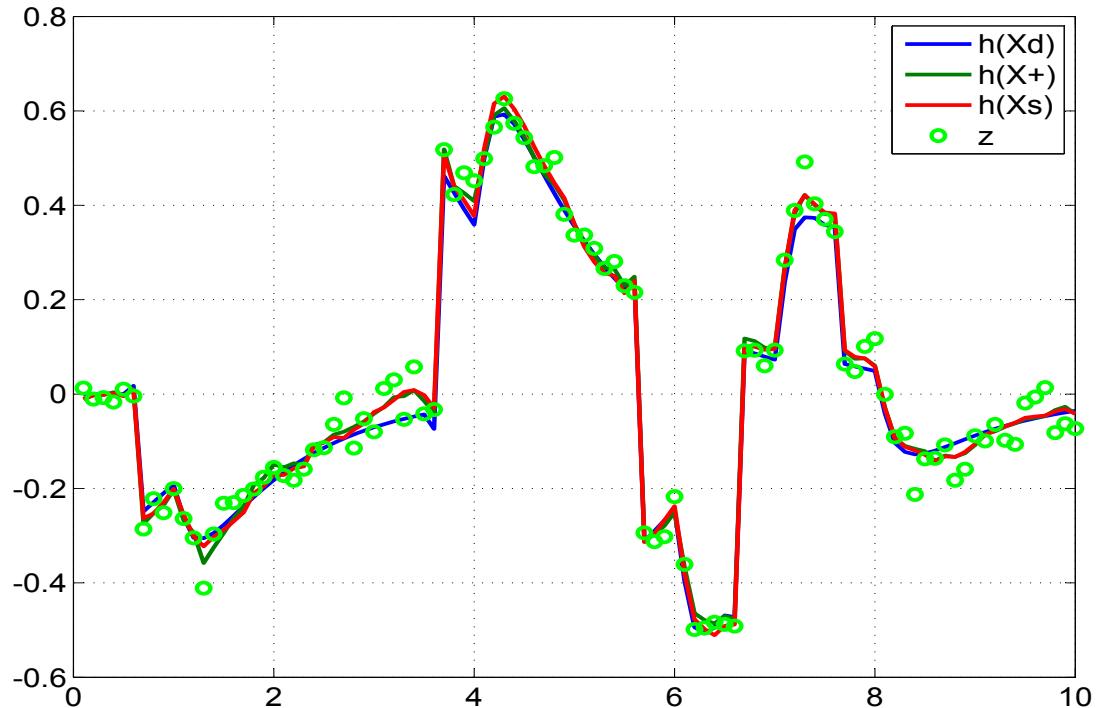
and the measurement 6

Figure 4.167: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 7

Figure 4.168: The innovations, filtered residue and smoothed residue

corresponding to measurement 1

Figure 4.169: The innovations, filtered residue and smoothed residue

corresponding to measurement 2

Figure 4.170: The innovations, filtered residue and smoothed residue

corresponding to measurement 3

Figure 4.171: The innovations, filtered residue and smoothed residue

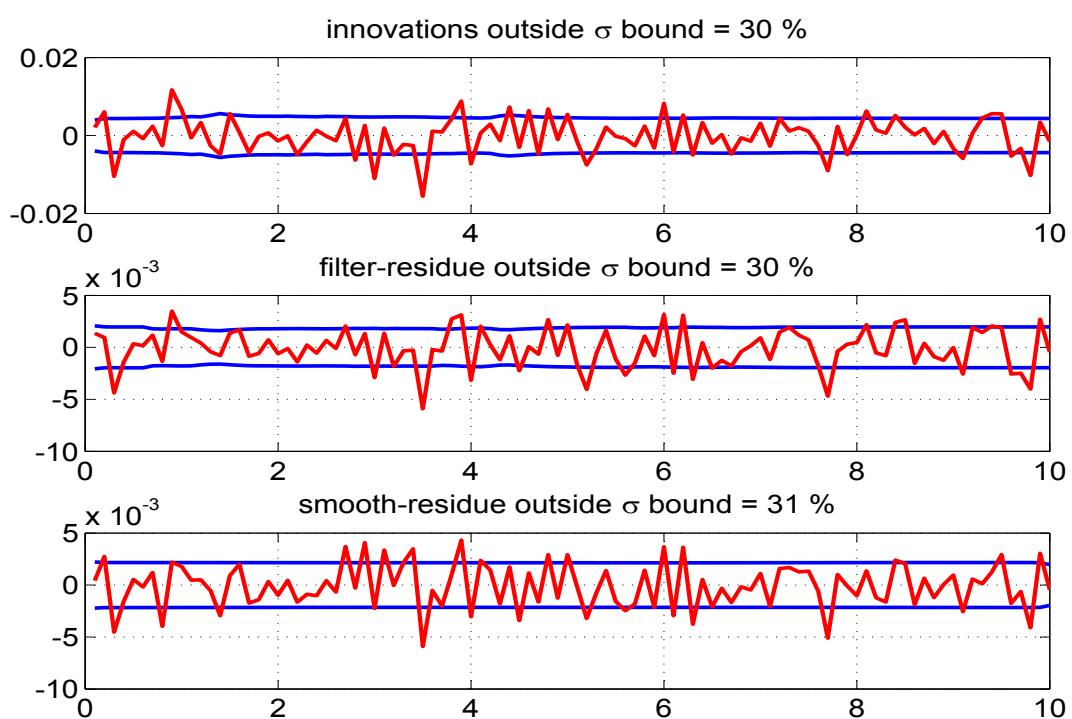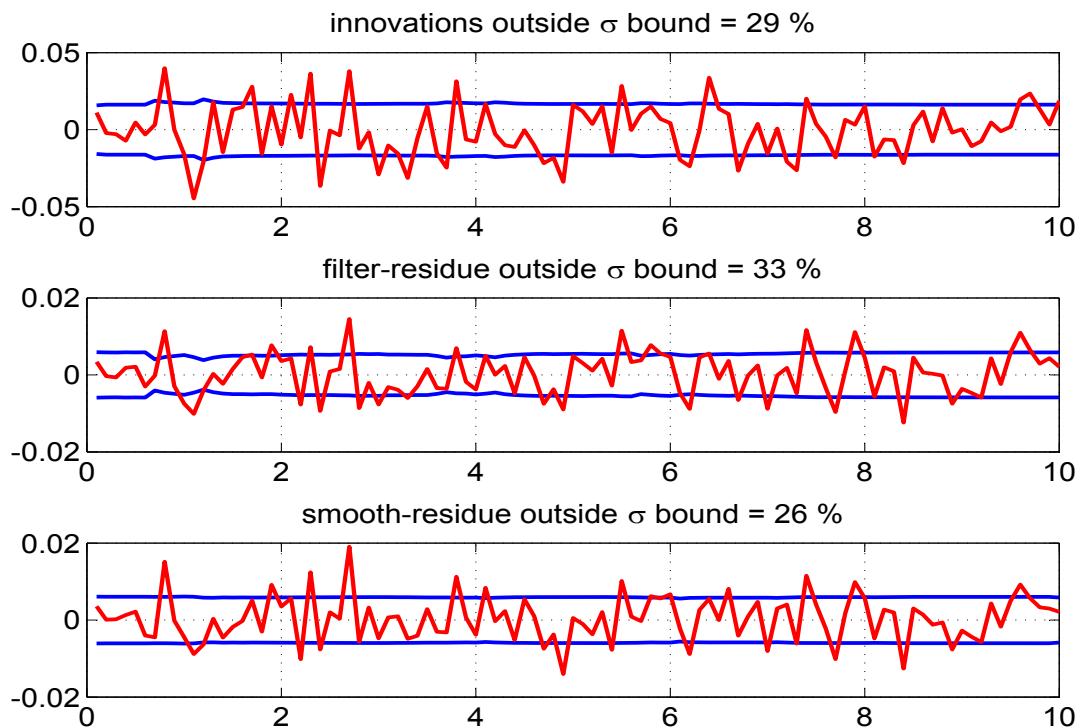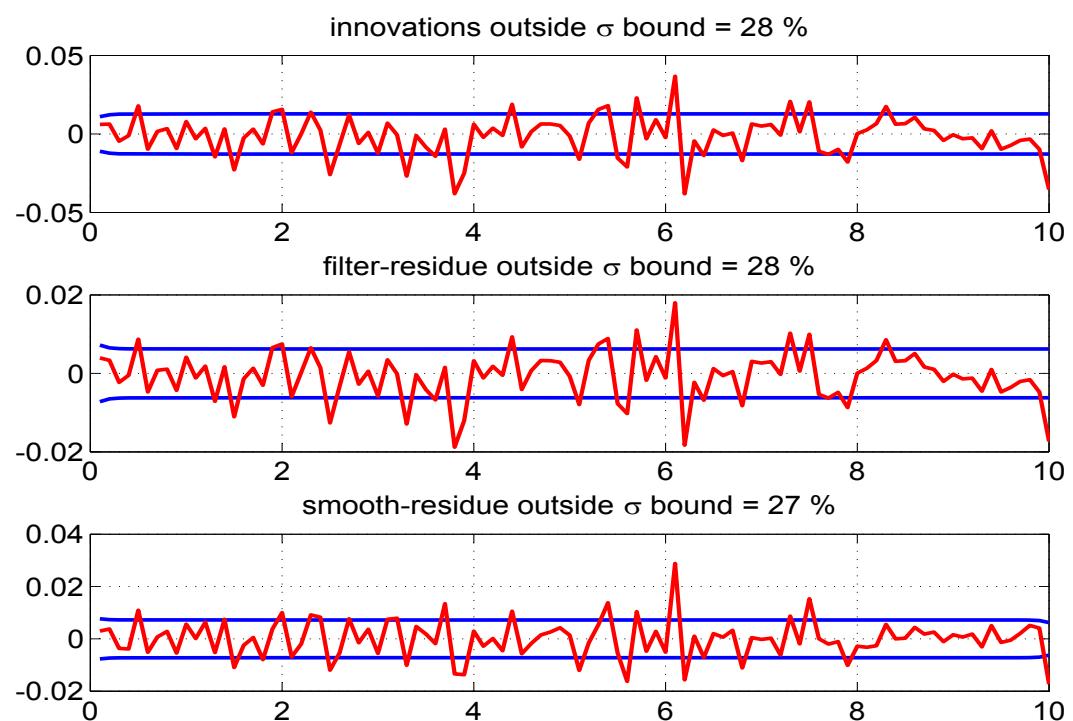corresponding to measurement 4
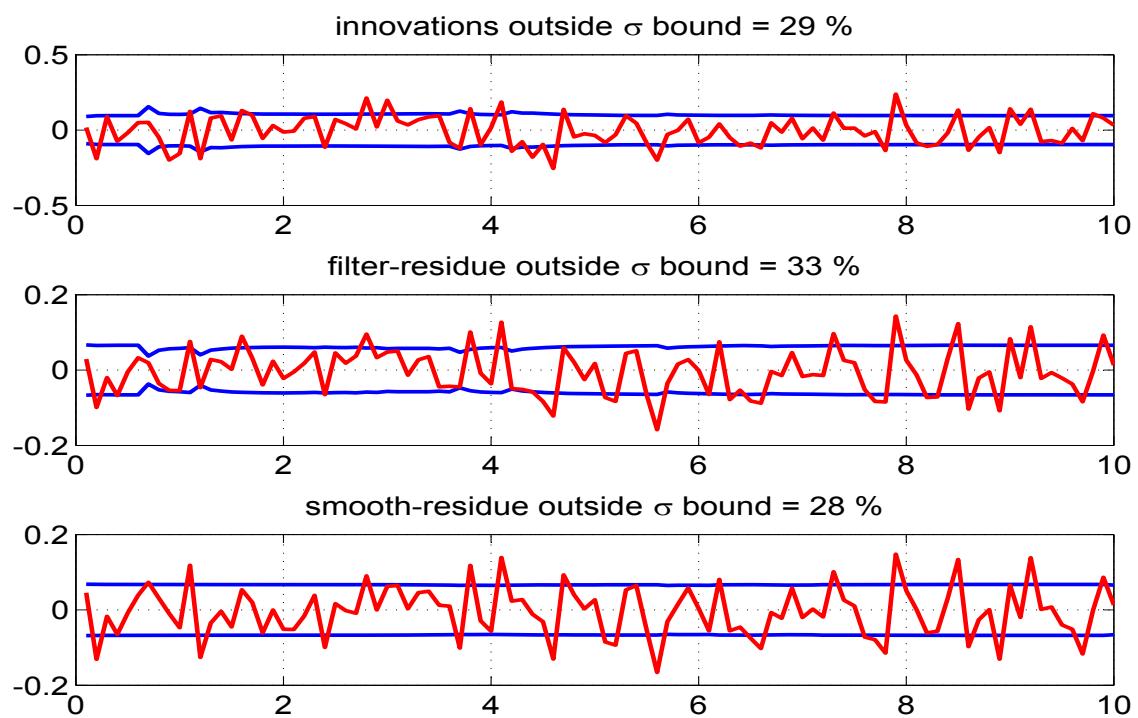
Figure 4.172: The innovations, filtered residue and smoothed residue

corresponding to measurement 5
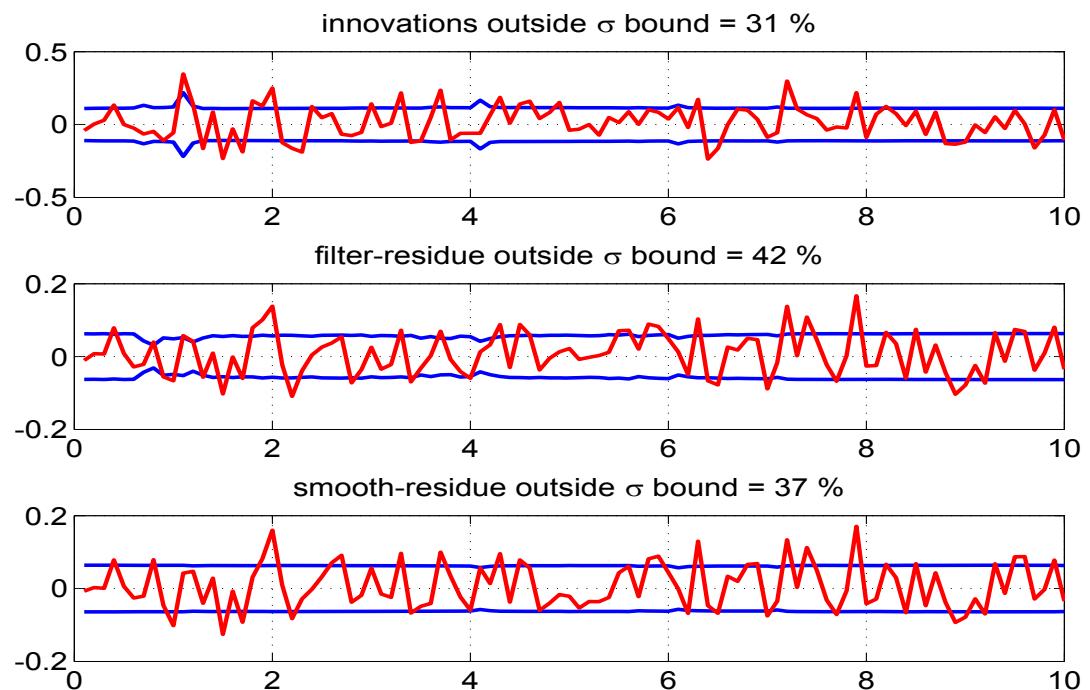
Figure 4.173: The innovations, filtered residue and smoothed residue

corresponding to measurement 6
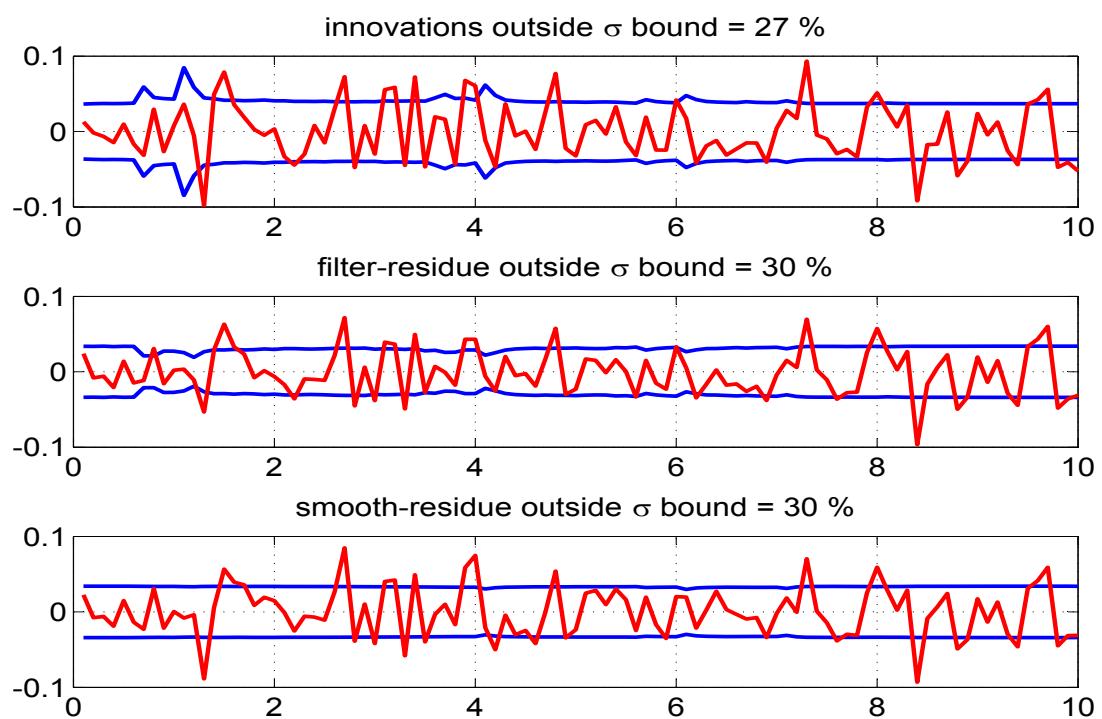
Figure 4.174: The innovations, filtered residue and smoothed residue

corresponding to measurement 7

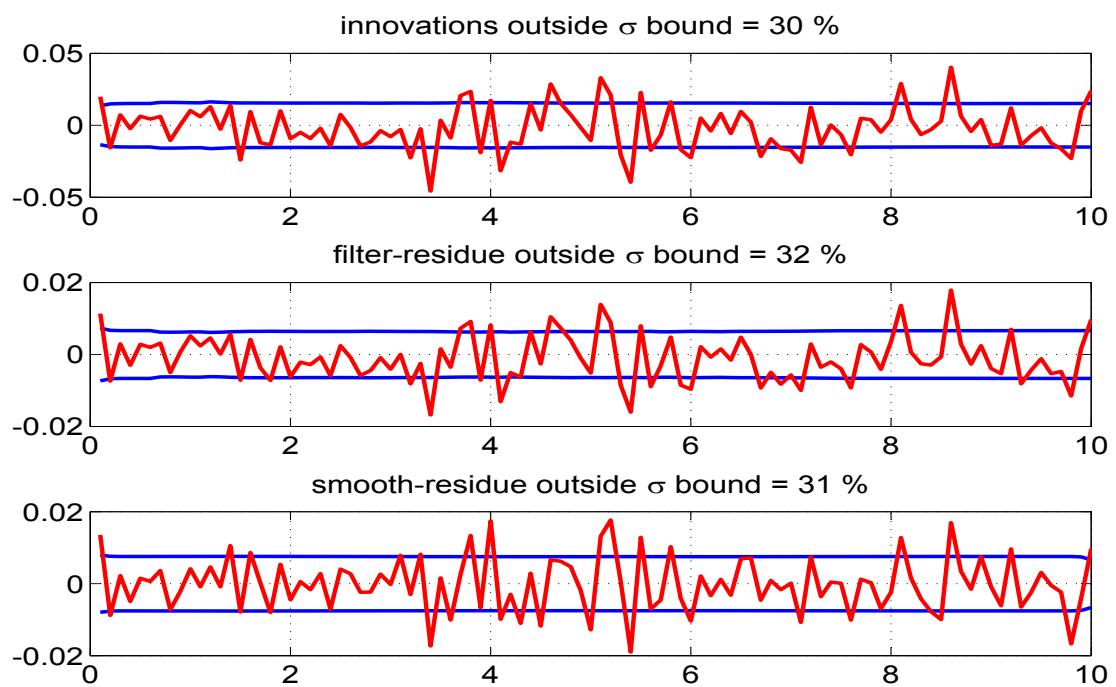Figure 4.175: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 1

Figure 4.176: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 2

Figure 4.177: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 3

Figure 4.178: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 4

Figure 4.179: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 5

Figure 4.180: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 6

Figure 4.181: Time variation of injected and estimated measurement noise (top) and

their autocorrelation (bottom) for measurement 7
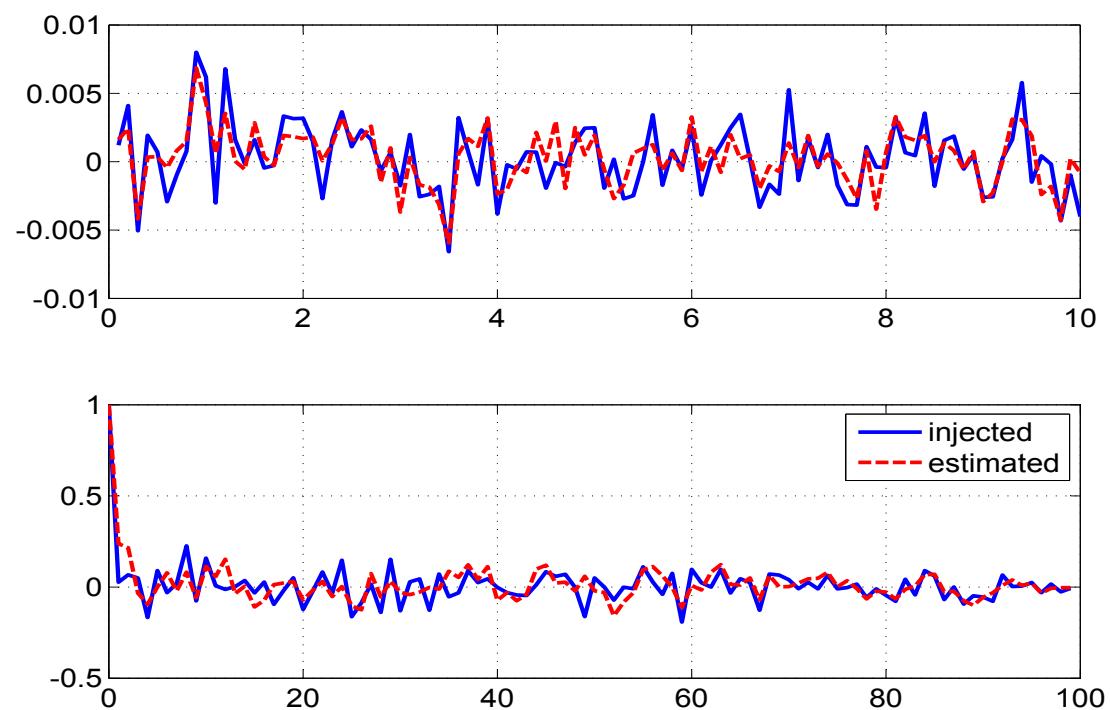
Figure 4.182: Time variation of injected and estimated process noise (top) and

their autocorrelation (bottom) for state 1
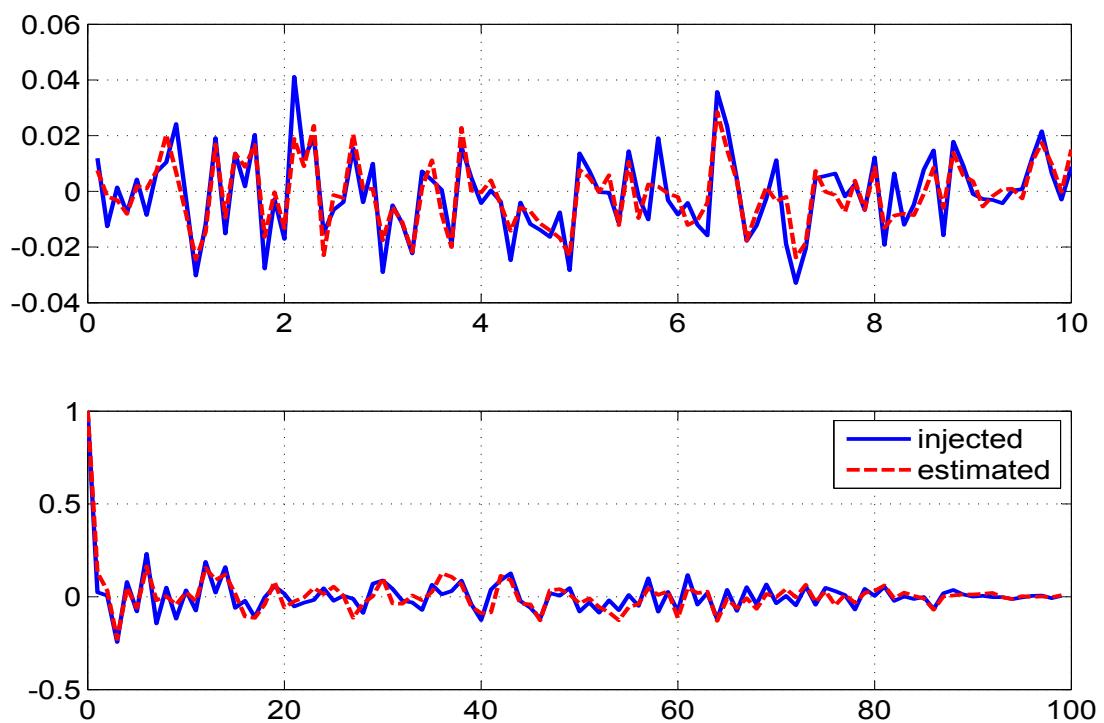
Figure 4.183: Time variation of injected and estimated process noise (top) and

their autocorrelation (bottom) for state 2

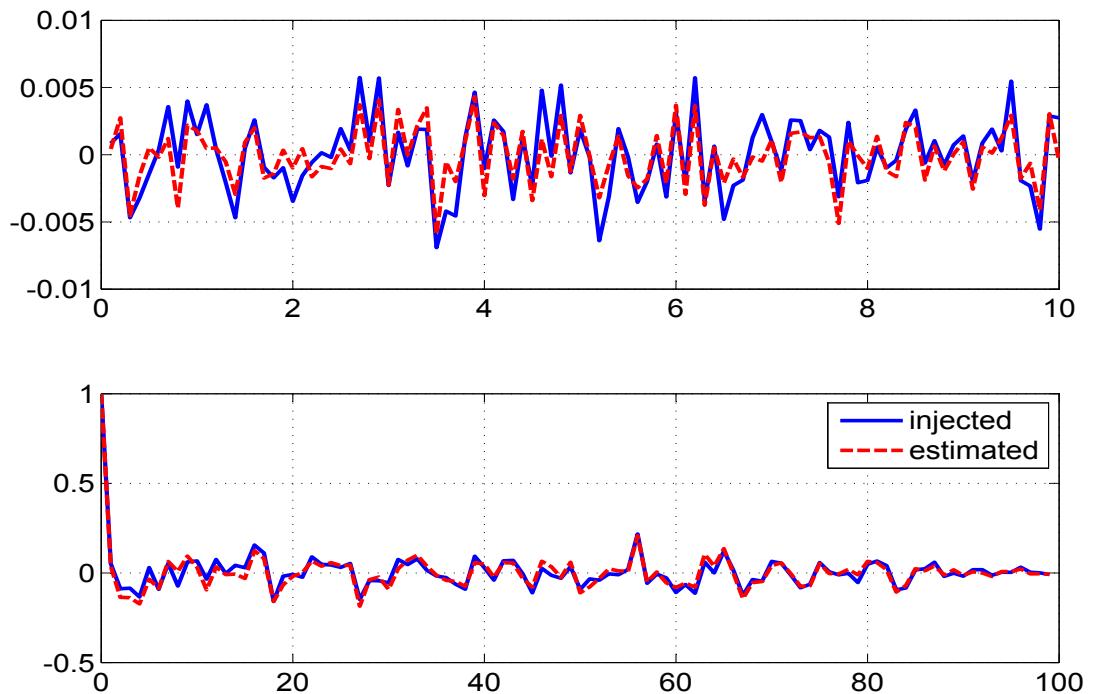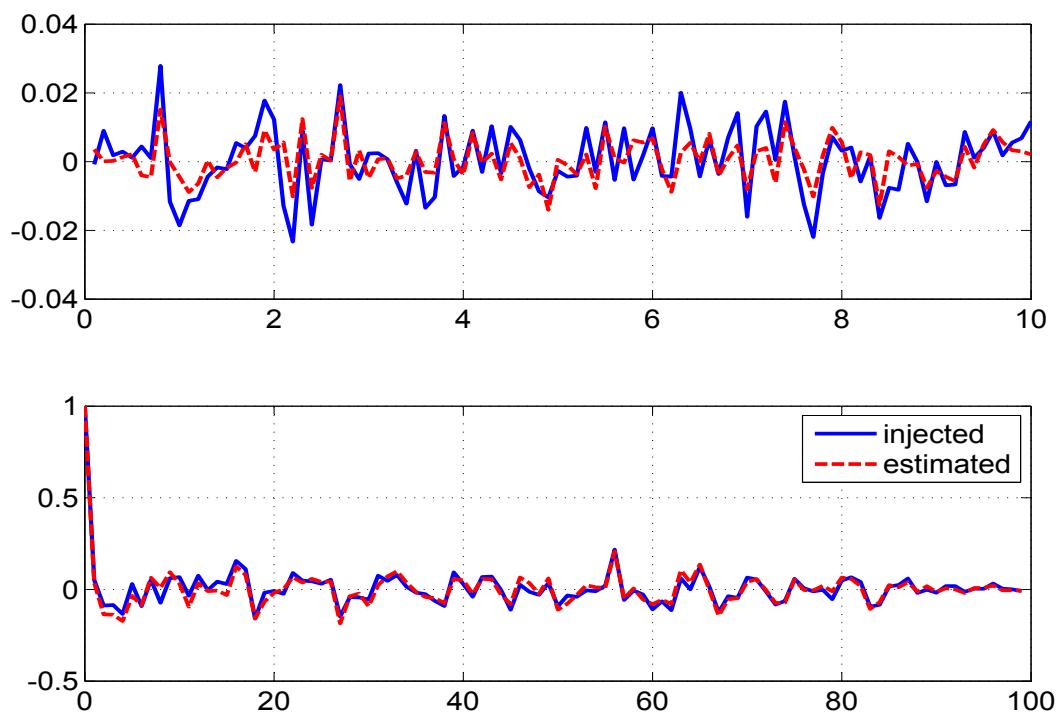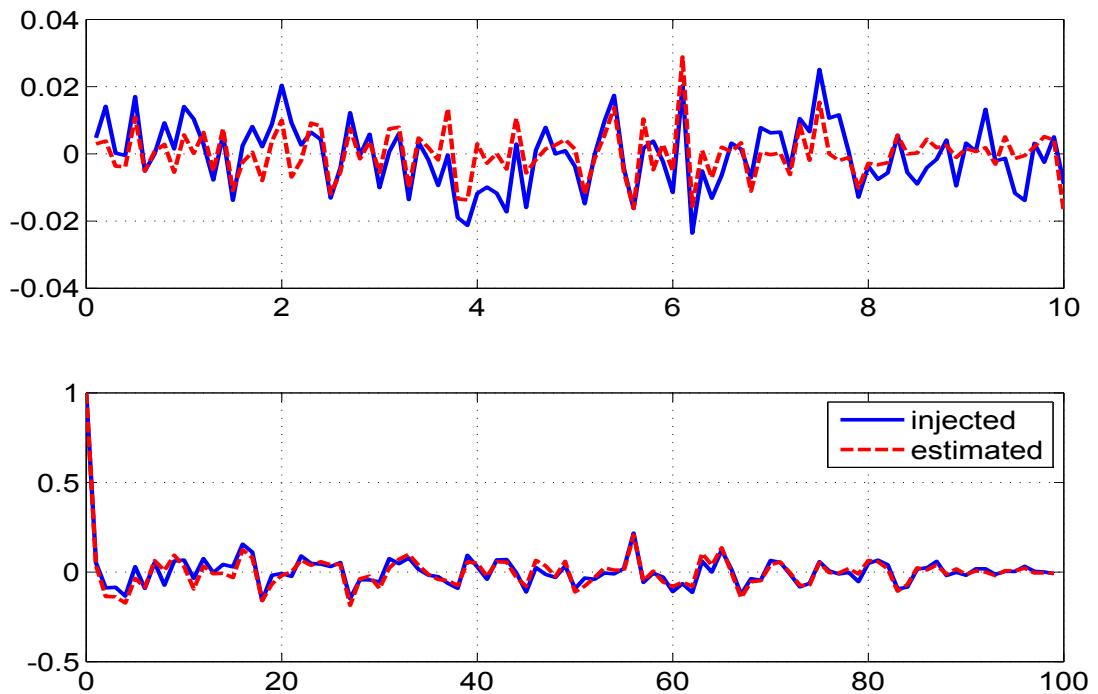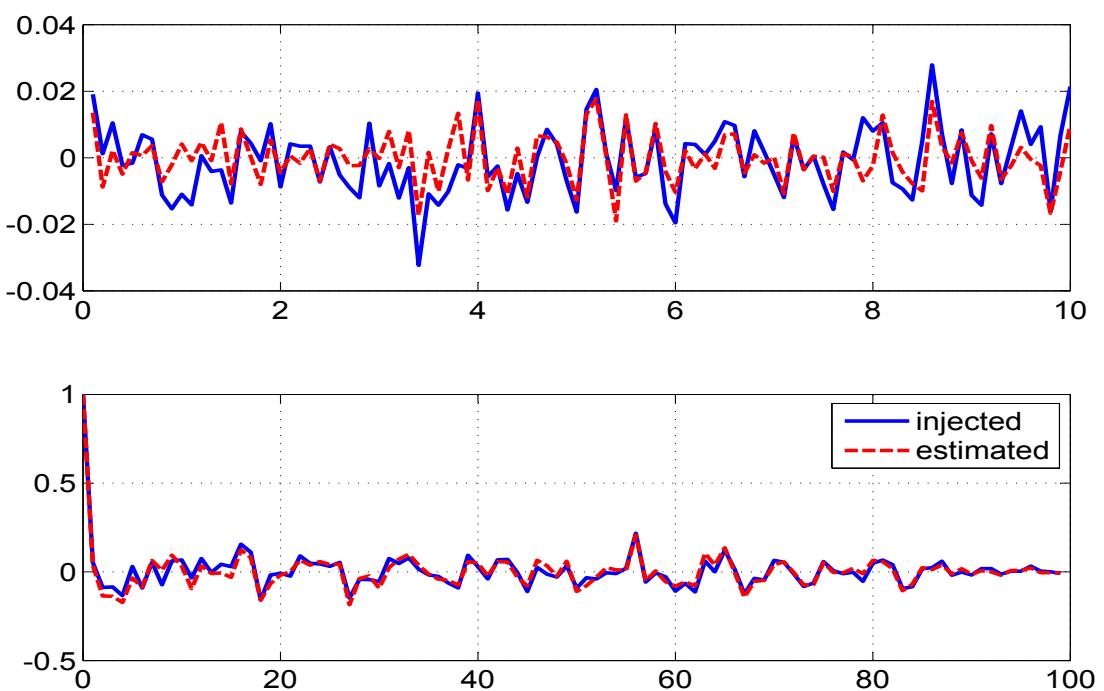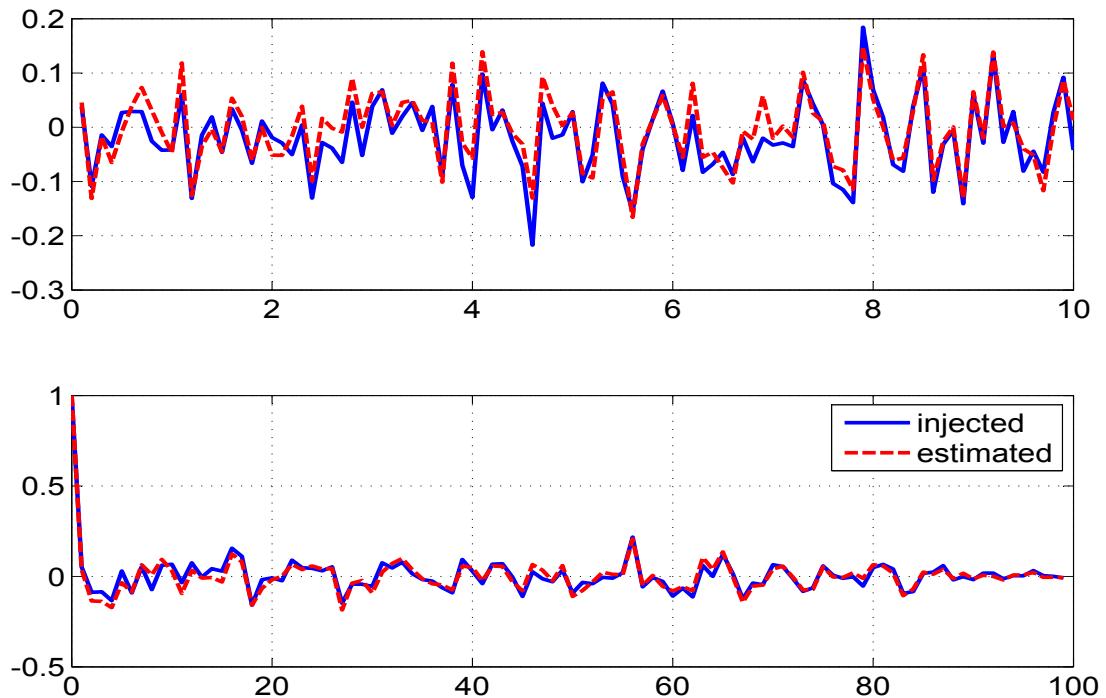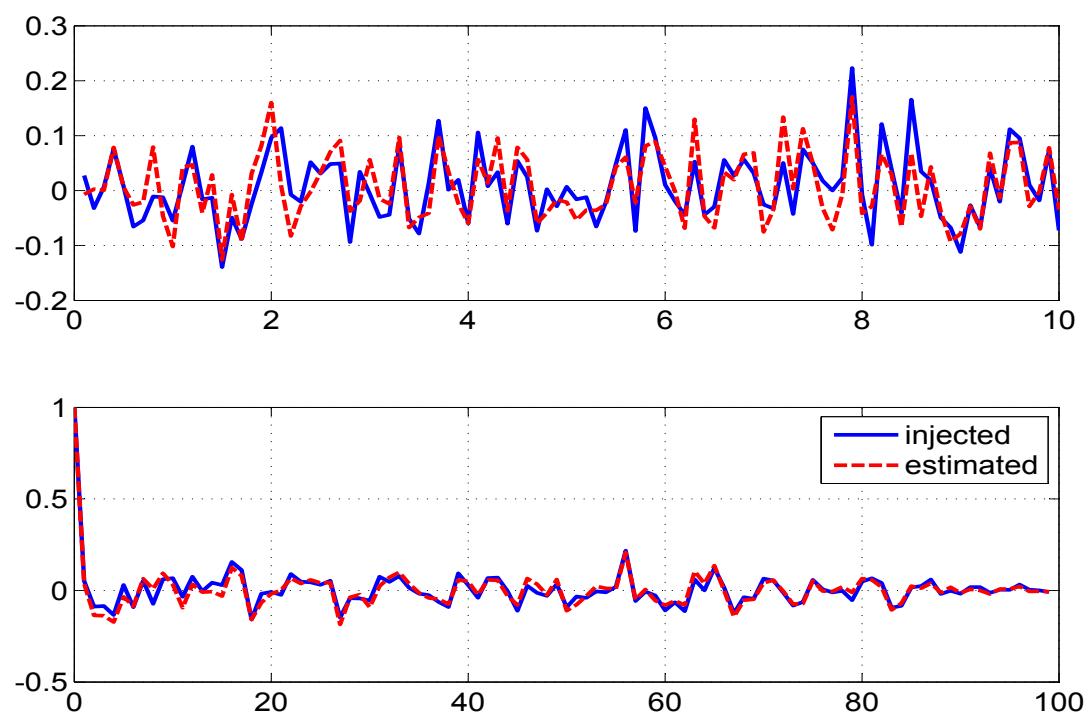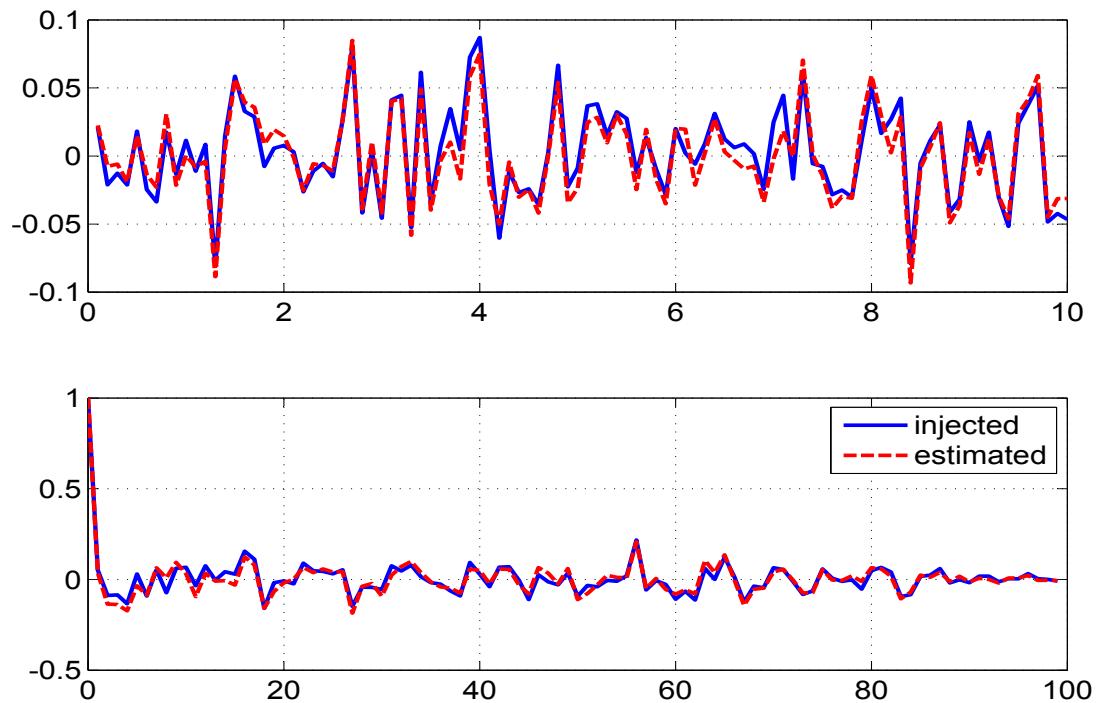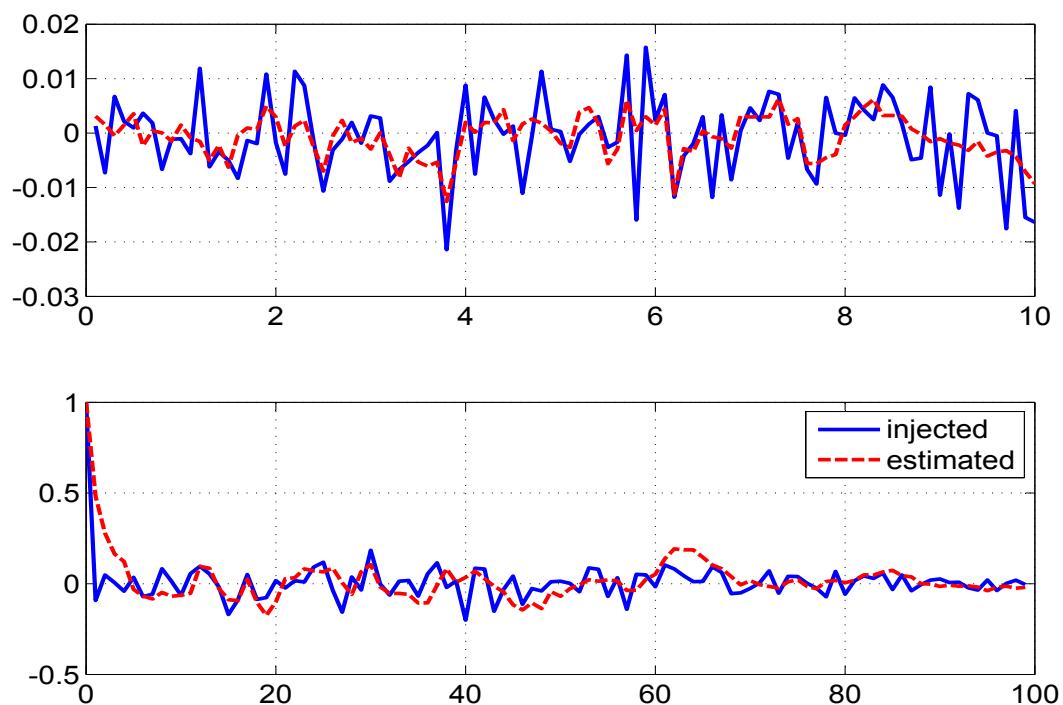Figure 4.184: Time variation of injected and estimated process noise (top) and

their autocorrelation (bottom) for state 3
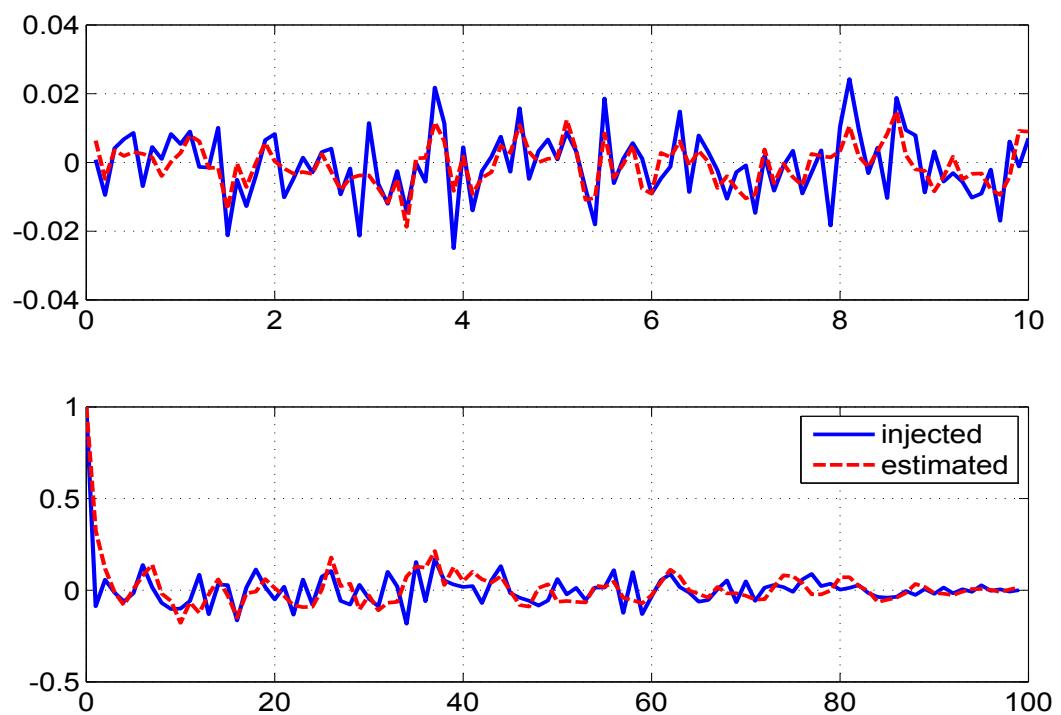
Figure 4.185: Time variation of injected and estimated process noise (top) and

their autocorrelation (bottom) for state 4

Chapter 5

Real Flight Test Data Analysis

After having studied simulated cases for simple systems and the longitudinal and lateral motion of an airplane it was felt useful to handle more involved real life cases. It turns out that a good choice once again would be the flight test data analysis of an airplane systems involving longitudinal and lateral motion. These generally have many states and measurements and more so a larger number of aerodynamic parameters to be determined. Further the flight tests cannot always be conducted in an ideal situation of the process noise and the measurement noises being white and Gaussian as is generally needed in the Kalman filter. The measurements are not available with respect to the center of gravity, possess scale and bias factors which have to be modelled and accounted for. The coupling between the longitudinal and lateral motion brings in added difficulty making the problem more interesting. At times the noisy measurements from the lateral motion are fed into the longitudinal states and thus are input as process noise. This is another example of introducing subjectivity in estimation theory. However the final results should be meaningful, reasonable, acceptable and useful no matter whatever subjective inputs are introduced into the problem formulation and solution.

In the previous Chapter we considered simulated data and established the efficacy of the present reference recursive recipe both for the system parameter estimates and the statistical characteristics of the process and the measurement noises. As in the analysis of simulated data the following filter outputs are studied which provides an insight into the filter performance. The following quantites converges through the iterations

1. The parameter estimates Θ and their covariances P_Θ.

2. The noise covariance of $\mathbf{Q}$ and R.

3. The state dynamics without measurement and process noises based on the estimated parameter after the filter pass through the data (Xd), the posterior state estimate, the smoothed state and the measurement.

4. The sample innovation, filtered residue and the smoothed residue along with their bounds which is the square root of the predicted covariances given respectively by $(\mathbf{R} + H_k P_{k|k-1} H_k^T)$, $(\mathbf{R} - H_{k|k} P_{k|k} H_{k|k}^T)$ and $(\mathbf{R} - H_{k|N} P_{k|N} H_{k|N}^T)$ by the filter.

5. The estimated measurement and process noise distribution with time as well as their autocorrelations.

6. The cost functions (**J1-J8**) in the final iteration as seen in the tables.

Since the following examples are from aerospace application we have retained their notation as it is and avoided cluttering non aeronautical readers with aeronautical terminology. There are many standard text books such as Roskam[22] (1973), or the thesis of Gemson[55] (1991) which can be referred to for the notations used here. Only the control inputs and biases are defined in the text. The initial values of $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$ were chosen as 10^{-1}, 10^{-1} and 2^{-1} as used in the simulated case study. All the results have stability over a wide range of the initial values of $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$.

5.1 Real Flight Test Case-1

The data set pertinent to case-1 is the Aircraft B check case in NASA TN D 7831 (Maine[31] 1975). The short period motion is excited by a elevator control input (δ_e in radians) as shown in Fig. 5.1. The total number of data points are N = 352 at regular interval of 20 millisecs. The analysis is carried out in FPS units. The data is processed using linear model with dimensional form of stability and control derivatives. The state equations ($n = 3$) for the angle of attack (α), pitch rate (q) and the pitch angle (Θ) respectively are

$$\dot{\alpha} = Z_\alpha \alpha + q - Z_\Theta \theta + Z_{\delta_e} \delta_e + Z_0$$
$$\dot{q} = M_\alpha \alpha + M_q q + M_{\delta_e} \delta_e + M_0$$
$$\dot{\Theta} = Cq + \Theta_0$$

The angle of attack (α), pitch rate (q), the pitch angle (Θ) and the normal acceleration (a_n) are measured (indicated with subscript 'm') in units of rad, rad/s, rad and ft/s^2 respectively. The measurement equations (m=4) are given by

$$\alpha_m = \alpha$$

$$q_m = q$$

$$\Theta_m = \Theta$$

$$a_{n_m} = -\frac{U_0}{g}(Z_\alpha \alpha + Z_{\delta_e} \delta_e + Z_0) + \frac{x_{a_n}}{g}\dot{q} + a_{n_b}$$

The unknown parameter set ($p = 9$) is $\Theta = (Z_\alpha, M_\alpha, M_q, Z_{\delta_e}, M_{\delta_e}, Z_0, M_0, \Theta_0, a_{n_b})^T$. The first three are aerodynamic parameters, the next two are the control derivatives and the last three are the biases in the state and measurements. The initial states are taken as initial measurement and the initial parameter values are $(-0.46, -3.5, -0.47, -0.05,$ $-5.74, 0.08, 0.16, -0.001, 1)^T$.

<div align="center">

Other constant values used for case-1

</div>

g=32.2	U_0=415.2	x_{a_n}=-0.01	Z_Θ=0.00221	C=0.9916

We first run the filter assuming Q = 0 followed by $\mathbf{Q} > 0$. It was found that about 20 iterations of the data by the filter would suffice. The Fig. 5.1 shows the elevator control input. The Fig. 5.2-5.22 refer to the $\mathbf{Q} = 0$. The Fig. 5.2 to 5.10 show the evolution of the parameter estimates as well as their covariances with iterations. It may be noted that in about 5 to 10 iterations the estimates converge and by about 10 iterations the co-variances converge to their final estimates. The Fig. 5.11 summarizes the above results. The parameters that control the dynamics strongly converge rapidly with the weaker ones perhaps with more iterations after some fluctuations. The next Fig. 5.12 and 5.13 show respectively the convergence of $\mathbf{R}$ and the various cost functions $\mathbf{J}$. The subsequent figures 5.14-5.17 shows (i) the state dynamics based on the estimated parameter after the filter pass through the data, (ii) the estimated state after measurement update, (iii) the smoothed state and (iv) the measurement. It may be seen that the measurements are generally away from all the others quantities. If the filter was tuned properly then all the quantities except the state based on the estimated parameters (without including the process noise) should be close to each other. Thus running the filter assuming zero process noise providing such an above behaviour indicates the compulsive necessity of running the filter by assuming

nonzero $\mathbf{Q}$ and estimating it as well. The following Fig. 5.18 to 5.21 show the variation of the sample innovation, filtered residue and the smoothed residue along with their σ bounds. Since most quantities in the extended Kalman filter follow Gaussian or approximated as a Gaussian distribution the sample values can be expected to be within the square root of the covariances for about two thirds of the time. Though in the above figures this appears to be true the above sample statistics appear to be non white in this real flight test data. The next Fig. 5.22 shows the estimated measurement noise distribution from different channels to be non white. If the filter works well then the cost functions **J1-J3** should be close to the number of measurements and **J6-J8** should be close to the number of states. However the Table-5.1 shows the various cost functions among them **J1-J3** are close to the number of measurements but the costs **J6, J7** are far different from the number of states with only **J8** being close to the number of states. Such a behaviour of the cost functions points to the fact the filter can work better and hence we operate the filter by assuming $\mathbf{Q} > 0$.

Thus the next step is to process the data by assuming $\mathbf{Q} > 0$ and here 100 iterations was used. This was inferred based on a plot in Fig. 5.23 of the difference between the iterated and final values over 100 iterations. The next Fig. 5.24 shows the variation of parameter estimates and its initial covariance $\mathbf{P_0}$ with iterations and a similar Fig. 5.25 for $\mathbf{Q}$ and $\mathbf{R}$ over 100 iterations. The variation of the different costs **J1-J8** is more revealing and informative. The values of **J1-J3** are close to the number of measurements (m=4) with **J6-J8** are close to the number of states (n=3) as shown in Fig. 5.26 and Table-5.2. This means the measurement and state equations are well balanced. The **J5** is the negative log likelihood cost function. Hence the filter results with $\mathbf{Q} > 0$ provide more confidence in the parameter Θ and noise covariance $\mathbf{Q}$ and $\mathbf{R}$ estimates. The later Fig. 5.27-5.30 compares (i) the state dynamics based on the estimated parameter after the filter pass through the data, (ii) the state after measurement update, (iii) the smoothed state and (iv) the measurement. Unlike the case of $\mathbf{Q} = 0$ except the first one as is to be expected the other three are close to each other thus providing more confidence in all the filter results. The Fig. 5.31-5.34 correspond to Fig. 5.18-5.21 of the $\mathbf{Q} = 0$ case. Fig. 5.35 shows that the estimated measurements do not appear to have constant statistical characteristics across time and may not be Additive White Gaussian Noise (AWGN). A similar plot for the estimated process noise is shown in Fig. 5.36. Another experiment was carried out by generating a typical data set by using the estimated theta and injecting the estimated

Q and **R** as additive white Gaussian noise. This is to determine the effect of non additive, non White and non Gaussian noise distribution in the real data on the CRBs. After each iteration in the reference recipe the Θ, **Q** and **R** were reset as from the real data. Similar experiment was also conducted by updating Θ as well. It turned out that there is not much of a difference in the final CRBs as can be seen from the Table-5.3.

Finally two other filter runs were carried out using the Myers and Tapley[34] (1976) as well as Mohamed and Schwarz[70] (1999) statistics for the estimation of **Q** and **R**. The behaviour of the various cost function and in particular **J6** and **J7** in Table-5.4 shows that the choice of the filter statistics for estimating **Q** and **R** in the proposed reference approach is the best possible when compared to other approaches presently considered.

5.1.1 Remarks on Case-1

The NASA results have been generated assuming **Q** = 0 and are comparable with reference procedure for the parameter estimates and their CRBs assuming **Q** = 0. There are only slight differences in most of the parameter estimates from the reference procedure run with **Q** = 0 and **Q** > 0. However it may be noted that the important parameter M_q representing the pitching moment coefficient with respect to pitch is only one fourth of the value and its CRB is also about 6 times than for **Q** = 0 case. The weakest parameter in terms of controlling the dynamics parameter Θ_0 has sign changes among the different estimation procedures. It may be noted that its CRB is of the same order as its parameter value which explains the sign change at times. Regarding the noise estimates in the reference case with **Q** > 0 the **R** values are lower than in the **Q** = 0 case perhaps at the cost of the emerging **Q**. Further the MT and MS methods give quite different estimates for the **R** values and much lower values for **Q** than in the reference **Q** > 0 case. Hence we believe that the present reference procedure provides the best possible parameter estimates and their uncertainties. From the plot of the parameter estimates and their %CRB in Fig 5.37, it can be seen that the parameters 1, 2, 3, 4, 5, 6, 7, 9, 10 are strong and the parameter 8 is the weak one. The %CRBs as estimated by different methods generally appear to vary widely. However what is interesting is that even the estimate of the strong parameter such as 5 varies widely among the methods. Such a behaviour of the filter estimates shows how important is the tuning of the filter statistics namely $\mathbf{P_0}$, **Q** and **R** in parameter estimation and their uncertainties.

The correlation co-efficient matrix (C) for i, j = 1, 2,... p is defined as

$$C_{ij} = \frac{P_{\Theta_{ij}}}{\sqrt{P_{\Theta_{ii}} \times P_{\Theta_{jj}}}}$$

where P_{Θ} is the parameter covariance matrix estimated by the EKF at the last time instant of the final iteration. The rounded off 100×C matrix for case-1 ($\mathbf{Q} = 0$) is given by

$$\begin{bmatrix}
100 & -79 & 57 & -91 & 80 & 100 & -79 & -73 & -66 \\
-79 & 100 & -76 & 72 & -99 & -79 & 99 & 73 & 42 \\
57 & -76 & 100 & -51 & 73 & 56 & -78 & -52 & -30 \\
-91 & 72 & -51 & 100 & -73 & -91 & 72 & 66 & 61 \\
80 & -99 & 73 & -73 & 100 & 81 & -99 & -74 & -43 \\
100 & -79 & 56 & -91 & 81 & 100 & -80 & -74 & -65 \\
-79 & 99 & -78 & 72 & -99 & -80 & 100 & 73 & 42 \\
-73 & 73 & -52 & 66 & -74 & -74 & 73 & 100 & 1 \\
-66 & 42 & -30 & 61 & -43 & -65 & 42 & 1 & 100
\end{bmatrix}$$

The rounded off 100×C matrix for case-1 ($\mathbf{Q} > 0$) is given by

$$\begin{bmatrix}
100 & 0 & 0 & -84 & 0 & 100 & 0 & 0 & -99 \\
0 & 100 & -66 & 0 & -84 & 0 & 100 & 1 & 0 \\
0 & -66 & 100 & 0 & 53 & 0 & -67 & -1 & 0 \\
-84 & 0 & 0 & 100 & 0 & -85 & 0 & 0 & 85 \\
0 & -84 & 53 & 0 & 100 & 0 & -85 & -1 & 0 \\
100 & 0 & 0 & -85 & 0 & 100 & 0 & 0 & -100 \\
0 & 100 & -67 & 0 & -85 & 0 & 100 & 1 & 0 \\
0 & 1 & -1 & 0 & -1 & 0 & 1 & 100 & 0 \\
-99 & 0 & 0 & 85 & 0 & -100 & 0 & 0 & 100
\end{bmatrix}$$

Ideally the above matrix is expected to be a diagonal matrix with 100 as its diagonal value. The correlation co-efficient matrix for $\mathbf{Q} > 0$ is more closer to the ideal values. The above matrix comparison indicates the usefullness of the right choice of $\mathbf{Q}$ in a real case scenario where one cannot model the system accurately. Hence we will be running only the reference EKF suggested for $\mathbf{Q} > 0$ case for the analysis of all the subsequent real cases.

5.1.2 Case-1 Tables

Table 5.1: Real flight test data case-1 results using the reference adaptive EKF ($\mathbf{Q} = 0$)

No of iterations=20

Study	Θ (Ref)	Θ (NASA)	Θ (Gemson)	σ_Θ (Ref)	σ_Θ (NASA)	σ_Θ (Gemson)	$\mathbf{R}$ $\times 10^{-3}$ (Ref)	**J1-J8** (Ref)	Remarks
				Case-1, Longitudinal case in dimensional form					
	-0.4604	-0.4502	-0.4616	0.0054	0.004742	0.0040		4.0886	
	-3.2725	-3.192	-3.5093	0.0151	0.01515	0.0845		4.0647	
	-0.4964	-0.5003	-0.4737	0.0103	0.01104	0.0360		3.9989	
$\mathbf{P_0}$: Scaled up-[0,0;0,✓]	-0.0530	-0.05197	-0.0534	0.0018	0.001529	0.0012	0.0056	0.0003	Improper fit as seen in
$\mathbf{Q}$: 10^{-10}-[✓,0;0,0]	-6.0924	-6.264	-5.7407	0.0454	0.04719	0.0372	0.0205	-39.3360	Fig. 5.14-5.17 due
$\mathbf{R}$: EM-diag	0.0871	0.08429	0.0852	0.0008	0.0007219	0.0007	0.0051	21.6246	to the absence of $\mathbf{Q}$
	0.1138	0.09206	0.1653	0.0028	0.002837	0.0124	0.2026	20.5621	
	0.0010	0.001517	-0.0009	0.0002	0.0002	0.0004		3.9875	
	1.0280	1.012	1.0038	0.0029	0.0029	0.0062			

Table 5.2: Real flight test data case-1 results using the reference adaptive EKF ($\mathbf{Q} > 0$)

No of iterations=100

Study	Θ (Ref)	Θ (NASA)	Θ (Gemson)	σ_Θ (Ref)	σ_Θ (NASA)	σ_Θ (Gemson)	$\mathbf{R}$ $\times 10^{-6}$ (Ref)	$\mathbf{Q}$ $\times 10^{-6}$ (Ref)	**J1-J8** (Ref)	Remarks
				Case-1, Longitudinal case in dimensional form						
	-0.4571	-0.4502	-0.4616	0.0065	0.004742	0.0040			3.7982	
	-3.5397	-3.192	-3.5093	0.1141	0.01515	0.0845			3.9098	
	-0.1227	-0.5003	-0.4737	0.0453	0.01104	0.0360			2.2523	
$\mathbf{P_0}$: Scaled up-[0,0;0,✓]	-0.0507	-0.05197	-0.0534	0.0022	0.001529	0.0012	0.0025	0.1876	0.0008	Cost functions
$\mathbf{Q}$: EM-[✓,0;0,0]	-5.6628	-6.264	-5.7407	0.0396	0.04719	0.0372	0.0022	0.2408	-49.9638	converge to the
$\mathbf{R}$: EM-diag	0.0839	0.08429	0.0852	0.0015	0.0007219	0.0007	0.0031	0.2293	2.9822	expected value
	0.1829	0.09206	0.1653	0.0164	0.002837	0.0124	328.6067		2.9871	
	-0.0010	0.001517	-0.0009	0.0013	0.0002	0.0004			2.8604	
	0.9941	1.012	1.0038	0.0149	0.0029	0.0062				

Table 5.3: Case-1 results using simulated Additive White Gaussian Noise

Study	σ_Θ (Simulated-without updating Θ)	σ_Θ (Simulated-with updating Θ)	σ_Θ (Ref)	Remarks
Case-1 data generated using simulated measurement and process noise (AWGN) of variance $\mathbf{Q}$ and $\mathbf{R}$ estimated by Reference EKF ($\mathbf{Q} > 0$)				
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: $\mathbf{Q}$ (Ref) $\mathbf{R}$: $\mathbf{R}$ (Ref)	0.0053 0.0941 0.0416 0.0022 0.0397 0.0014 0.0136 0.0013 0.0149	0.0053 0.0941 0.0416 0.0022 0.0397 0.0014 0.0136 0.0013 0.0149	0.0065 0.1141 0.0453 0.0022 0.0396 0.0015 0.0164 0.0013 0.0149	No Significant change in σ_Θ

Table 5.4: Real flight test data case-1 results using the MT and MS method.
No of iterations=100

Study	Θ (MT)	σ_Θ (MT)	$\mathbf{R}$ (MT) $\times 10^{-6}$	$\mathbf{Q}$ (MT) $\times 10^{-6}$	J1-J8 (MT)	Θ (MS)	σ_Θ (MS)	$\mathbf{R}$ (MS) $\times 10^{-6}$	$\mathbf{Q}$ (MS) $\times 10^{-6}$	J1-J8 (MS)	Remarks
Case-1, Longitudinal case in dimensional form											
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: MT/MS-[✓,0;0,0] $\mathbf{R}$: MT/MS-diag	-0.4511 -3.1810 -0.4088 -0.0541 -5.9706 0.0837 0.1088 -0.0008 1.0039	0.0053 0.0487 0.0257 0.0016 0.0450 0.0008 0.0075 0.0005 0.0023	8.3 9.7 0.5 148.2	0.00023 0.03652 0.02931	3.2977 3.2896 2.7542 0.0003 -42.8774 21.2035 20.9052 2.3735	-0.4552 -3.2115 -0.4934 -0.0531 -6.1559 0.0853 0.1011 0.0003 1.0163	0.0067 0.0215 0.0136 0.0020 0.0563 0.0010 0.0039 0.0007 0.0035	8.2906 30.0566 0.4620 259.4142	0.0001 0.0013 0.0523	2.9032 2.8957 2.5550 0.0003 -41.5716 21.5763 21.3320 2.3743	Cost functions are not close to their expected values in MT and MS method

5.1.3 Case-1 Figures (Q = 0)
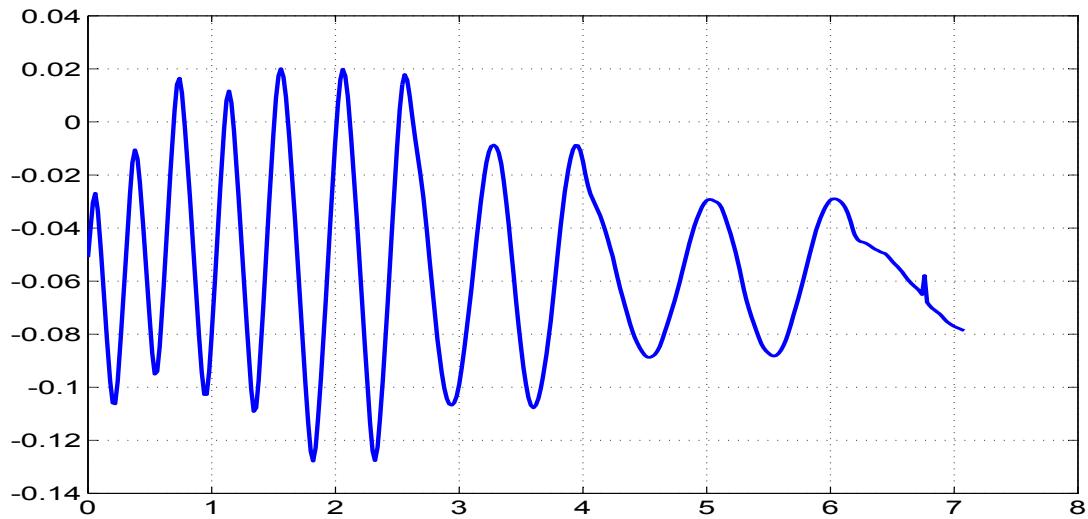
Figure 5.1: Control input (δ_e) versus time (s)

Figure 5.2: The variation of parameter estimate 1 and their filtered

and smoothed covariances through (with the time cumulatively) the iterations

Figure 5.3: The variation of parameter estimate 2 and their filtered

and smoothed covariances through (with the time cumulatively) the iterations

Figure 5.4: The variation of parameter estimate 3 and their filtered

and smoothed covariances through (with the time cumulatively) the iterations
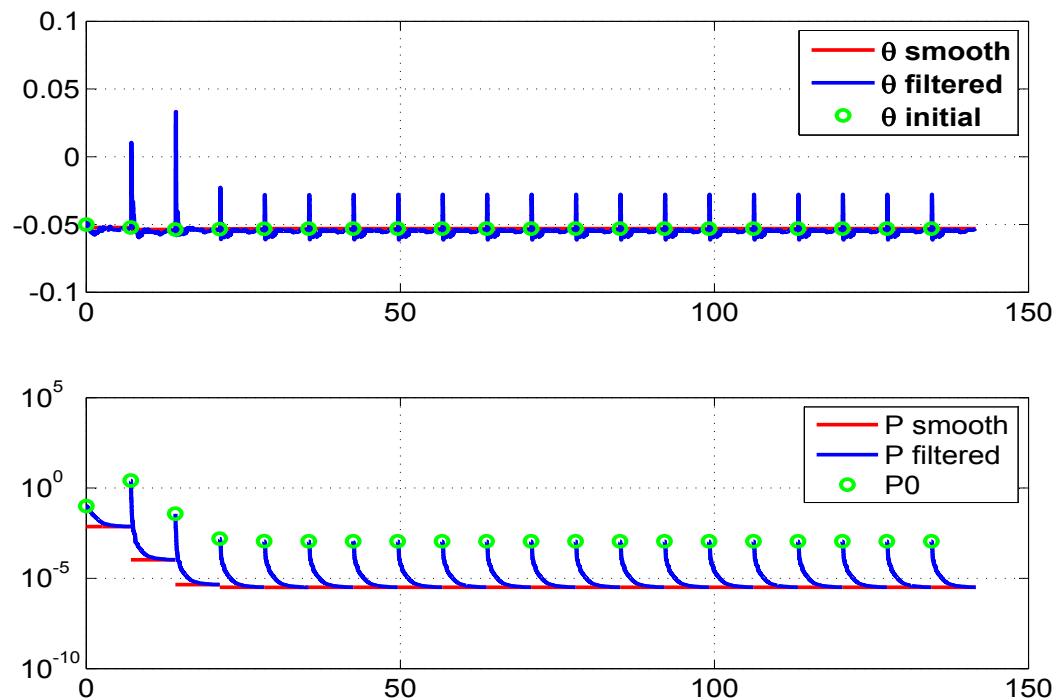
Figure 5.5: The variation of parameter estimate 4 and their filtered

and smoothed covariances through (with the time cumulatively) the iterations
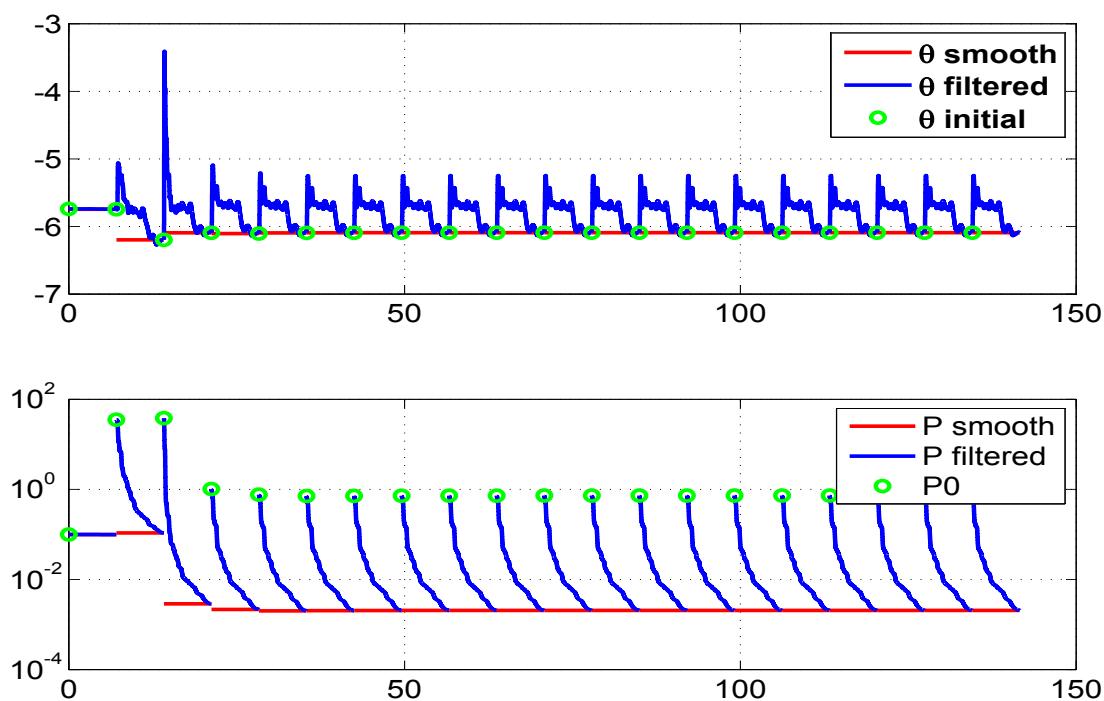
Figure 5.6: The variation of parameter estimate 5 and their filtered

and smoothed covariances through (with the time cumulatively) the iterations
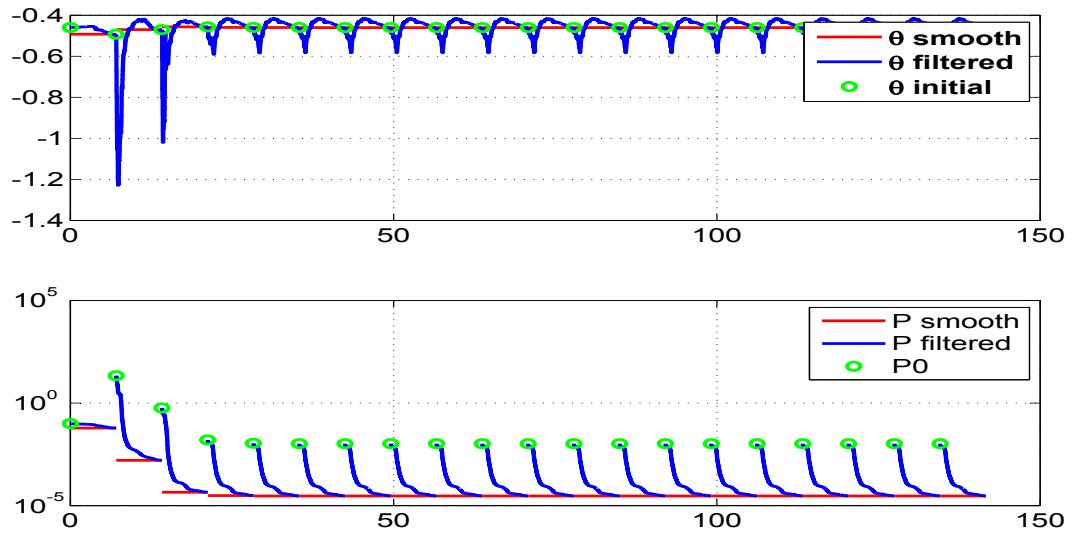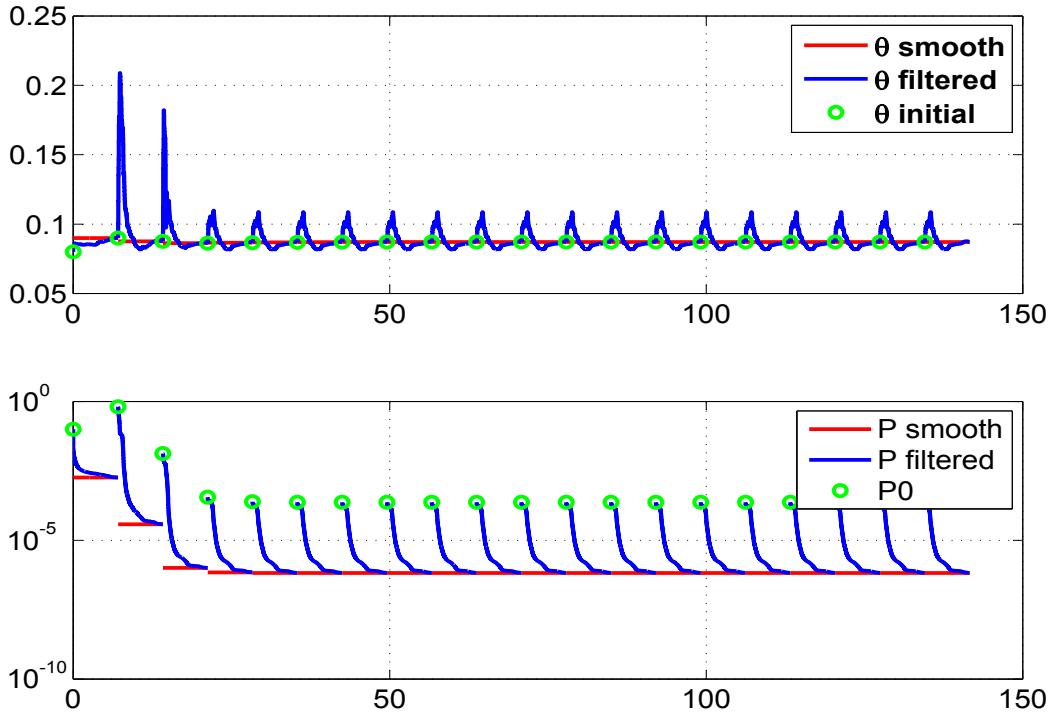
Figure 5.7: The variation of parameter estimate 6 and their filtered
and smoothed covariances through (with the time cumulatively) the iterations
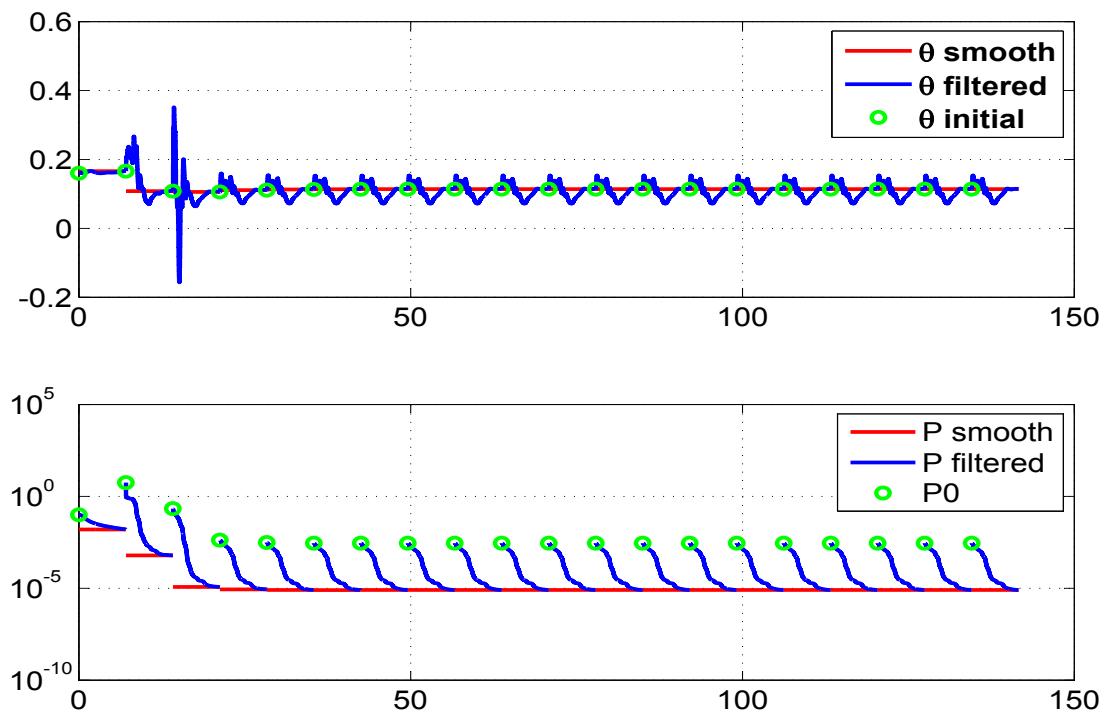
Figure 5.8: The variation of parameter estimate 7 and their filtered
and smoothed covariances through (with the time cumulatively) the iterations
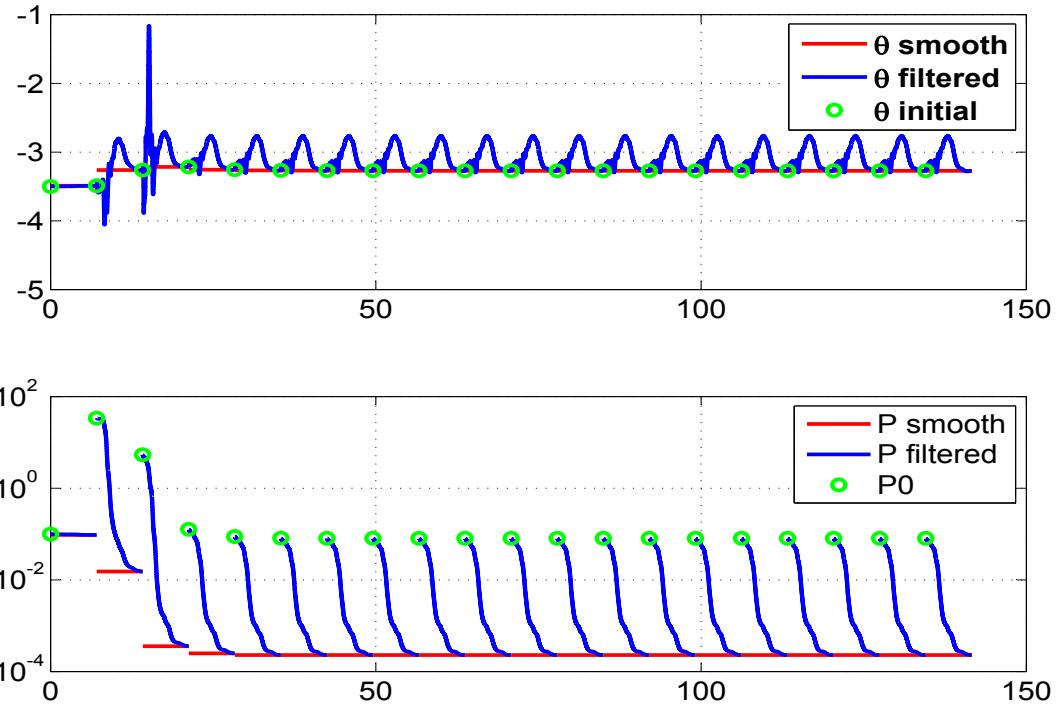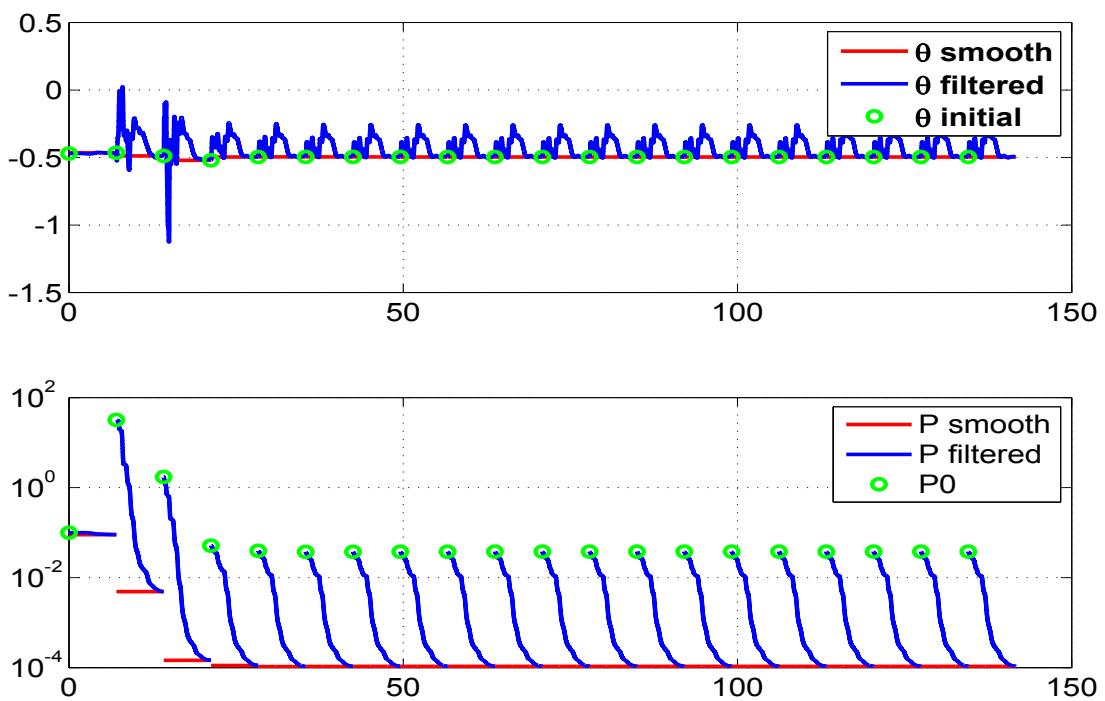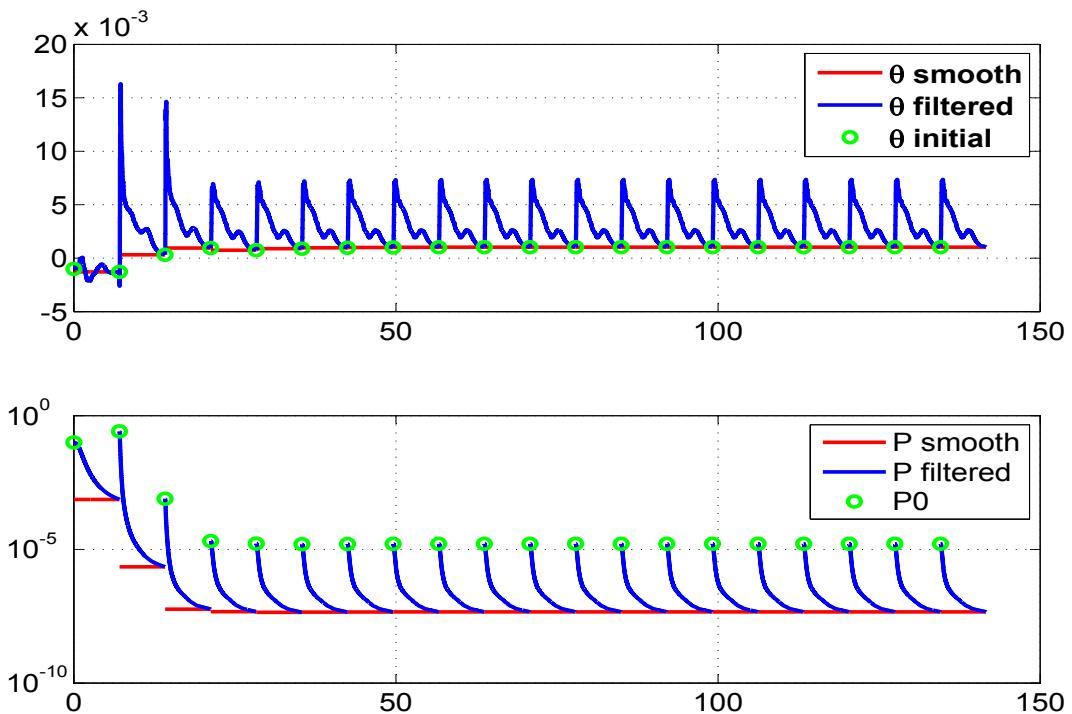
Figure 5.9: The variation of parameter estimate 8 and their filtered

and smoothed covariances through (with the time cumulatively) the iterations
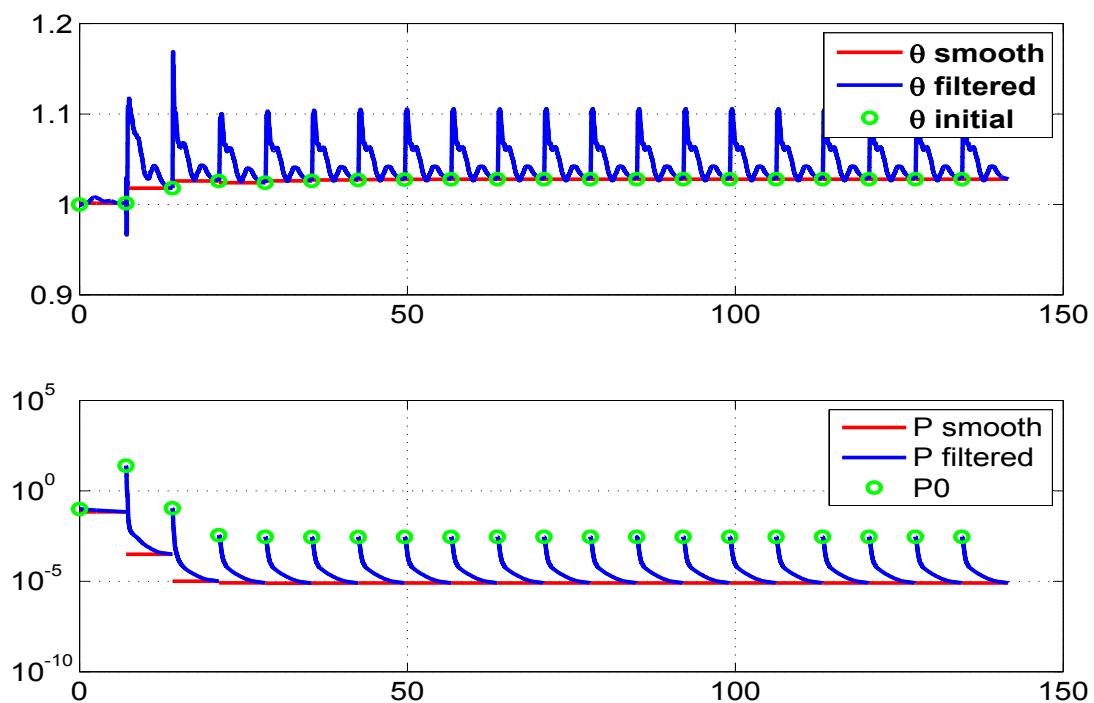
Figure 5.10: The variation of parameter estimate 9 and their filtered

and smoothed covariances through (with the time cumulatively) the iterations
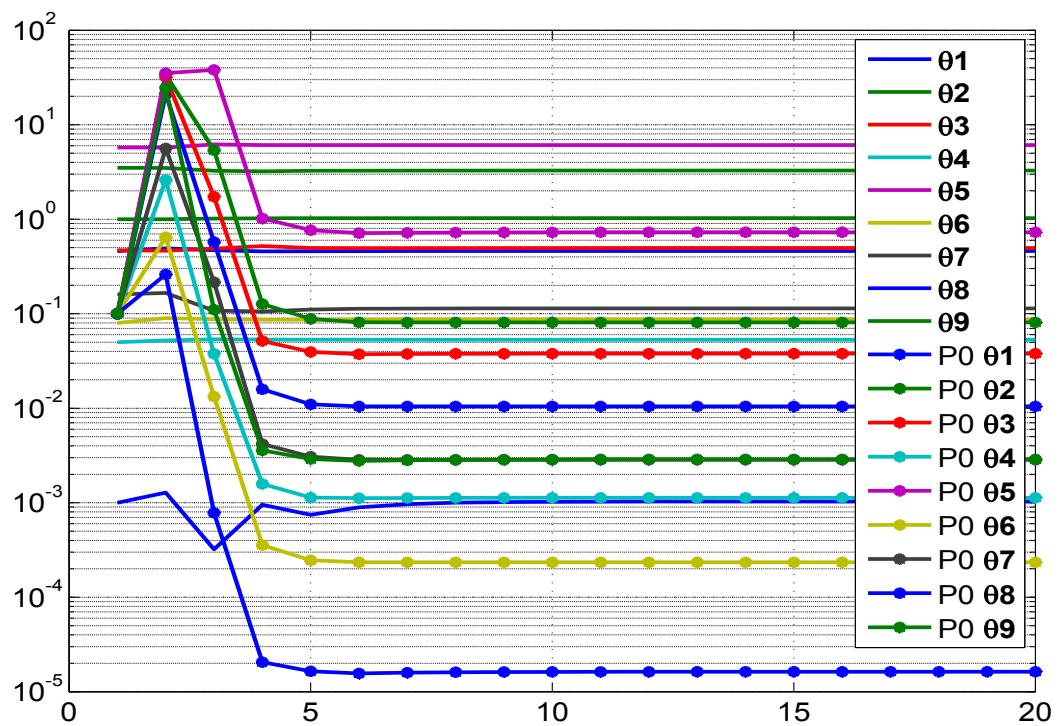
Figure 5.11: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
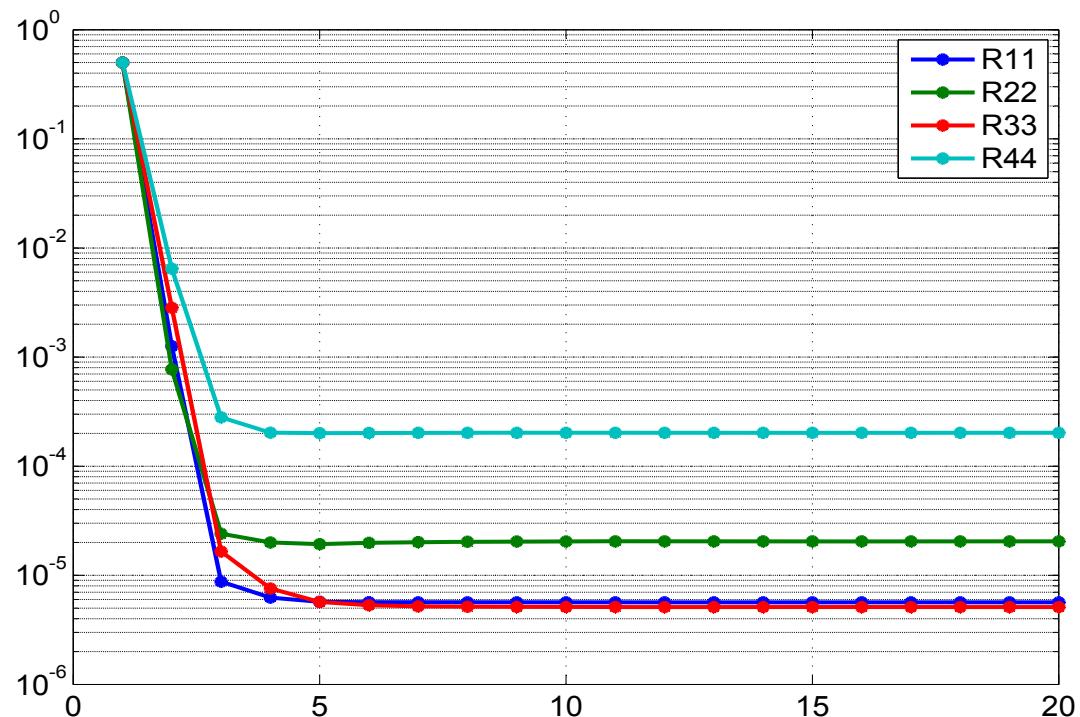
Figure 5.12: Variation of $\mathbf{R}$ with iterations
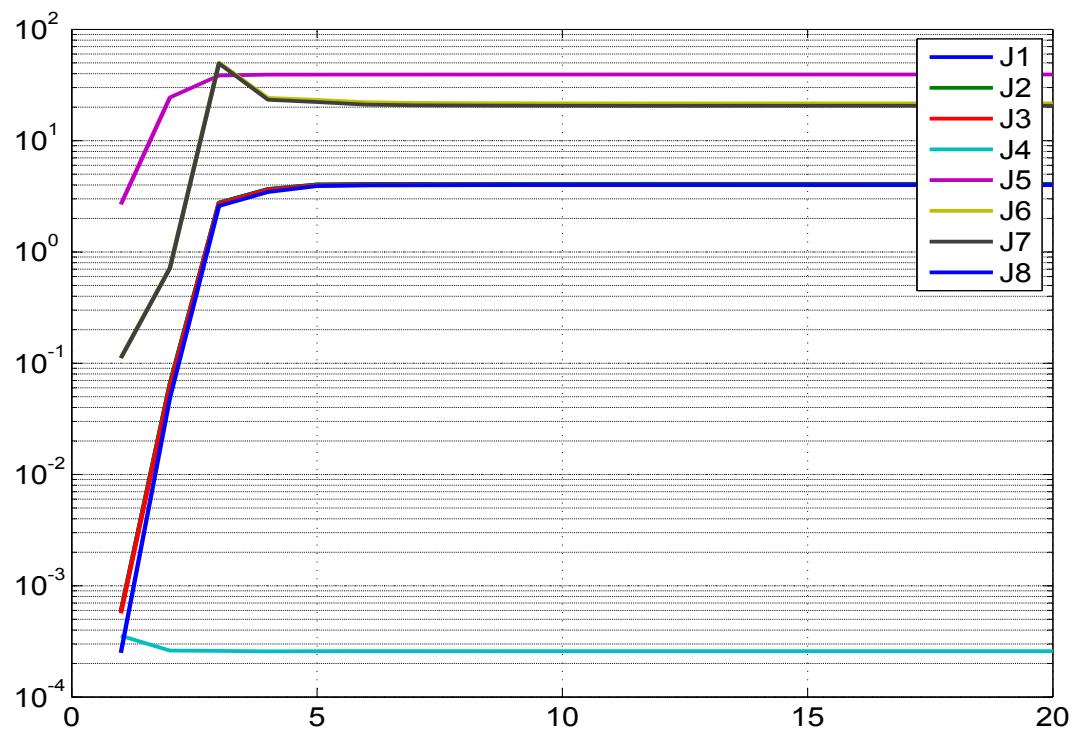
Figure 5.13: Variation of different costs (**J1-J8**) with iterations
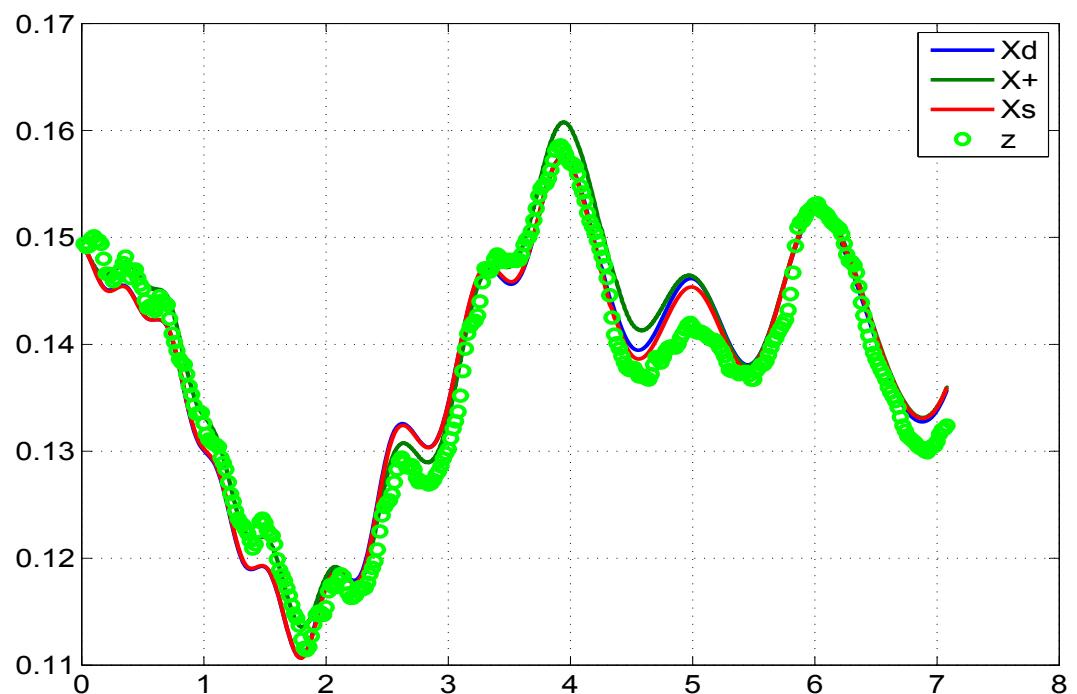
Figure 5.14: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1
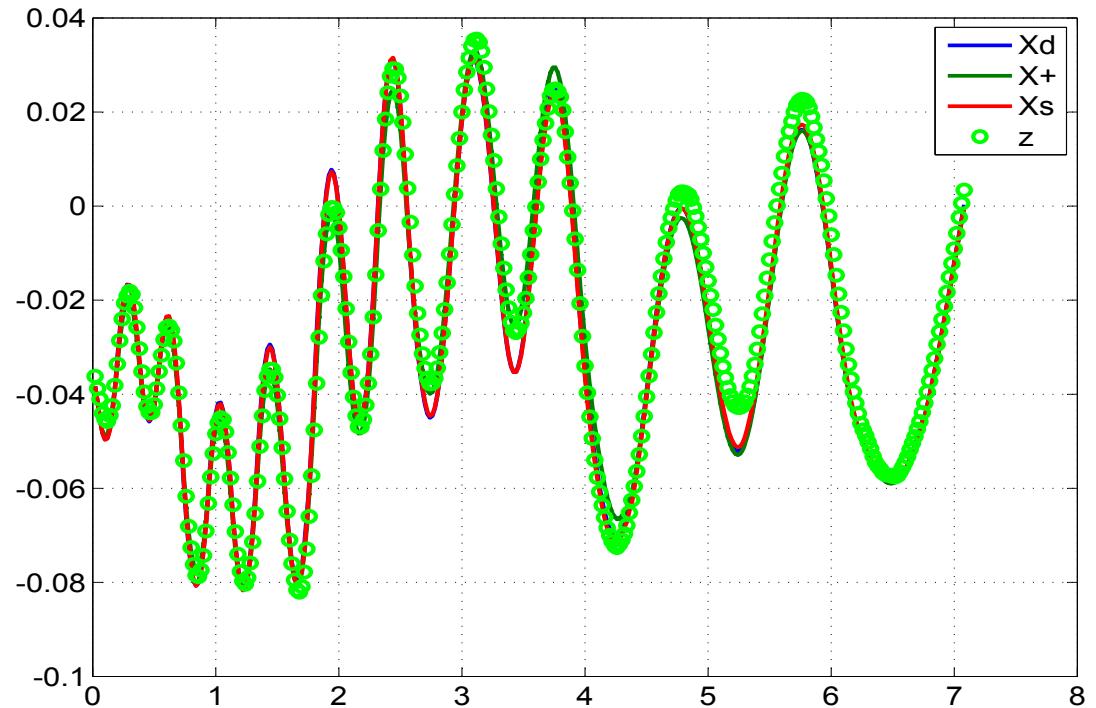
Figure 5.15: Comparison of the predicted dynamics, posterior, smoothed and the measurement 2
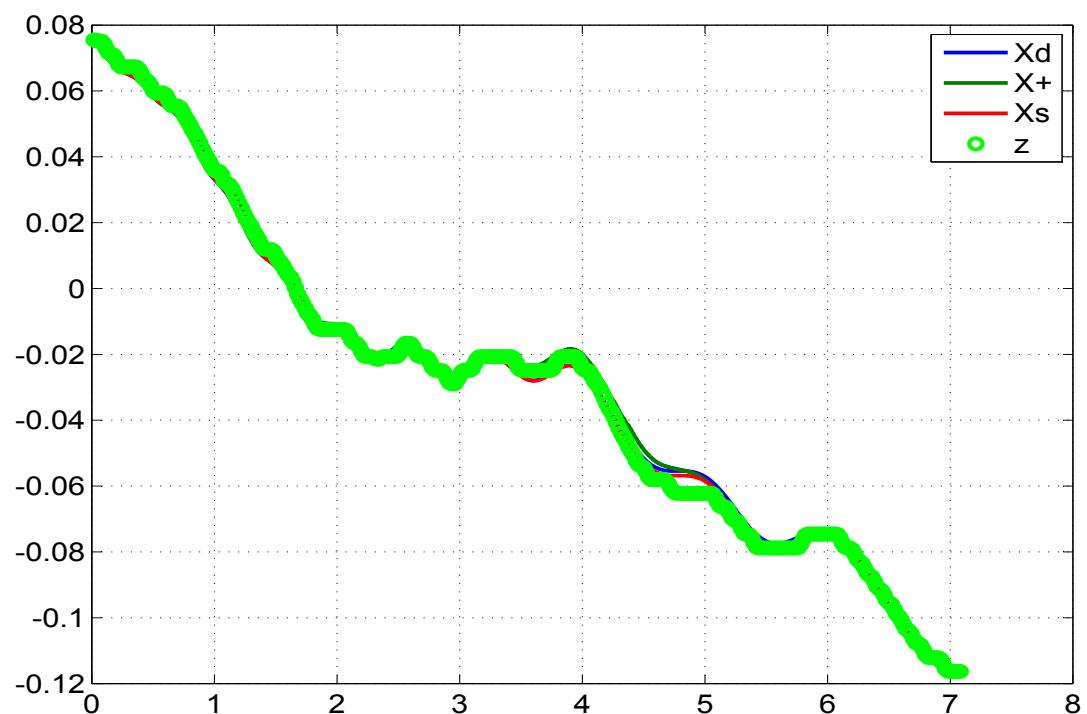
Figure 5.16: Comparison of the predicted dynamics, posterior, smoothed and the measurement 3
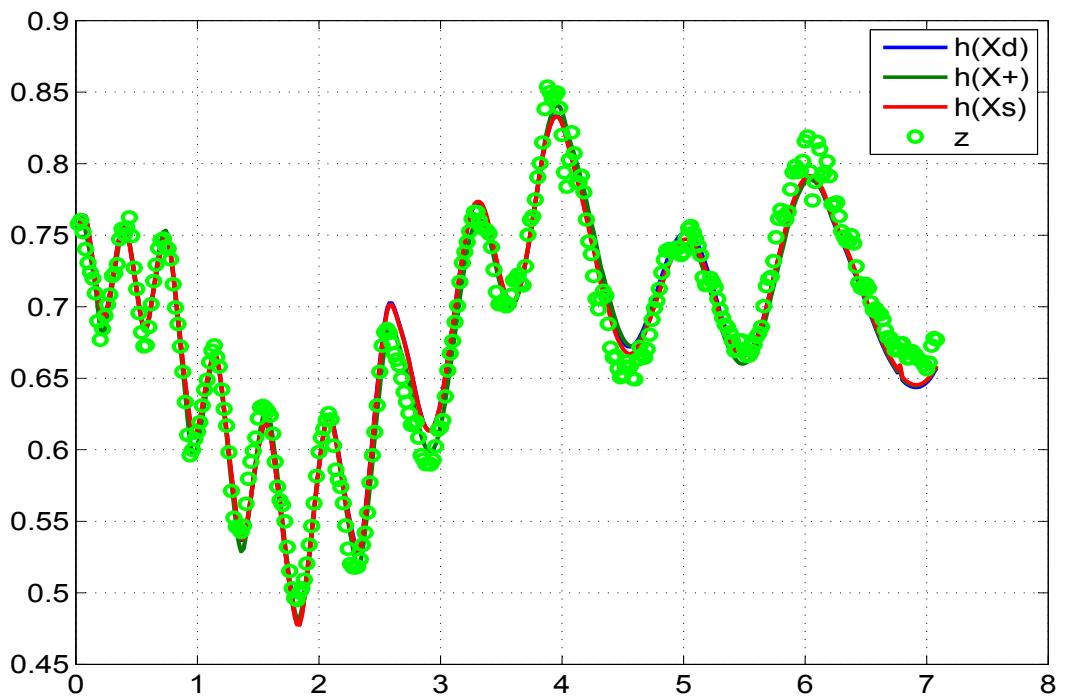
Figure 5.17: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 4
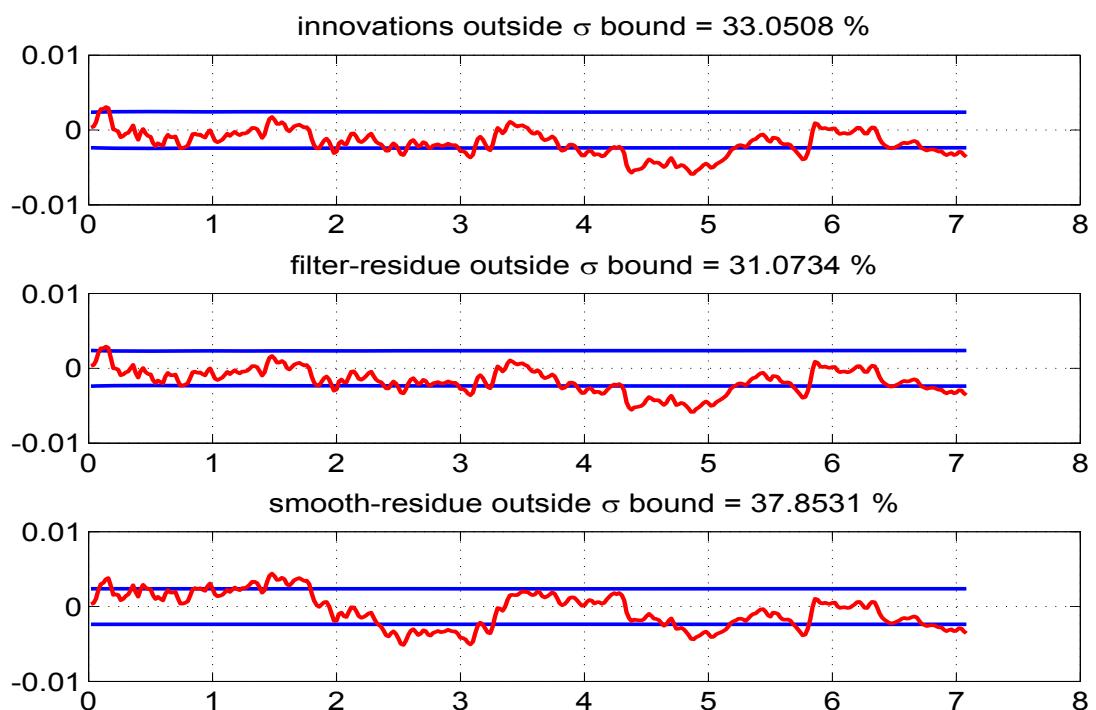
Figure 5.18: The innovations, filtered residue and smoothed residue of measurement 1
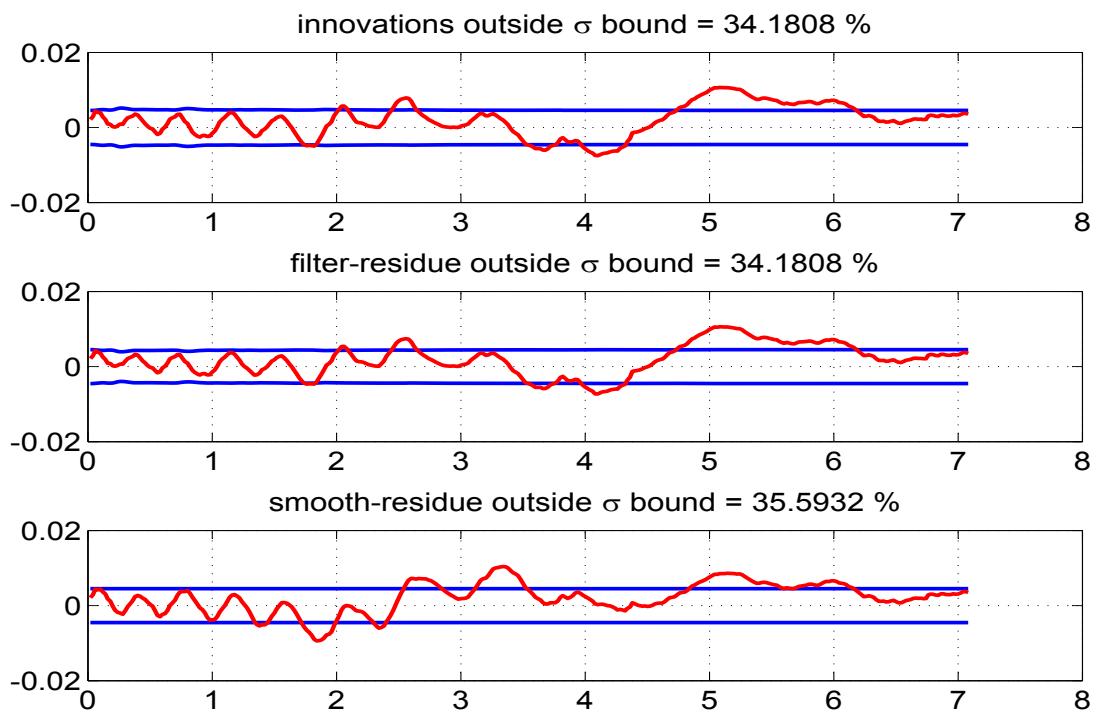
Figure 5.19: The innovations, filtered residue and smoothed residue of measurement 2
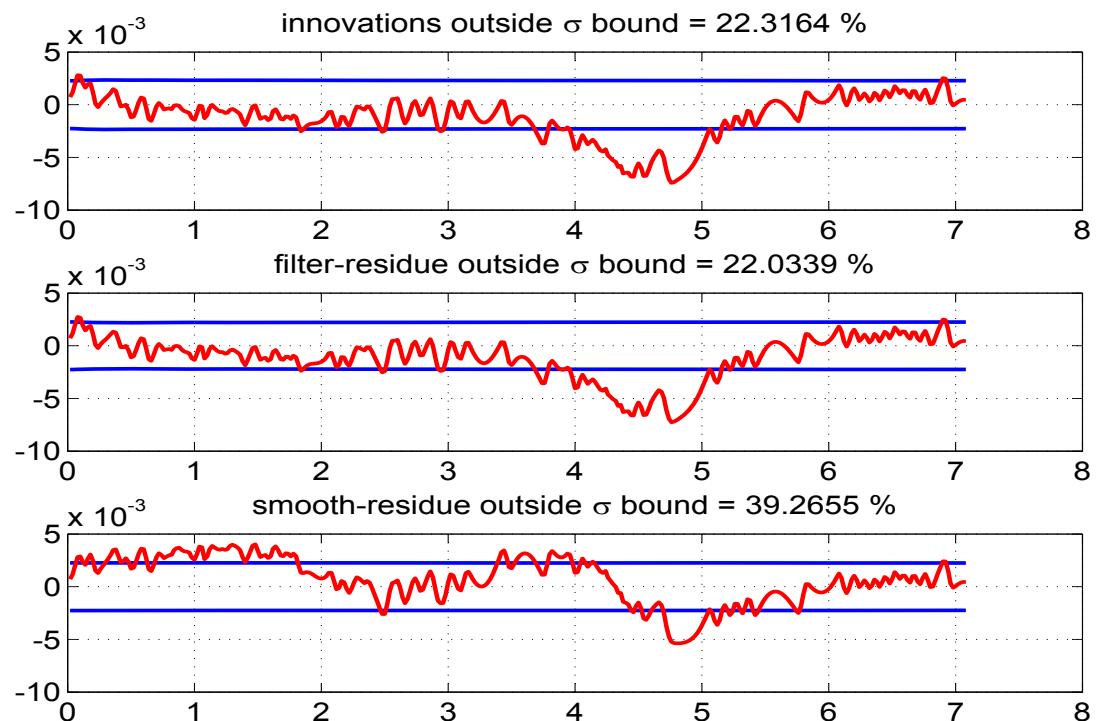
Figure 5.20: The innovations, filtered residue and smoothed residue of measurement 3
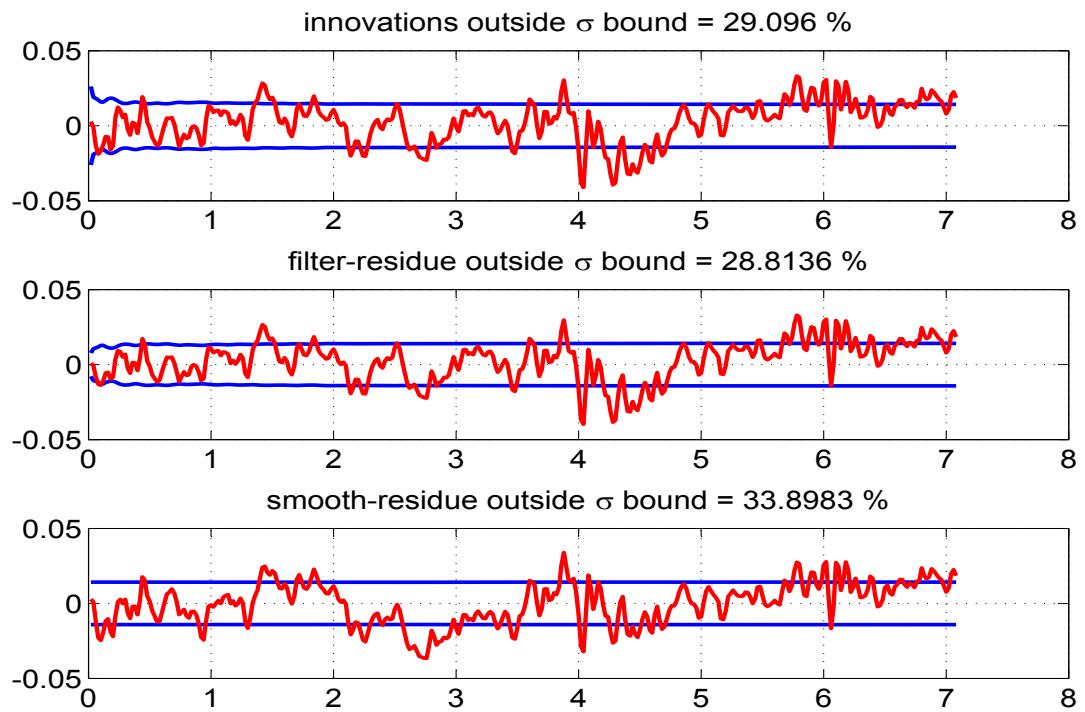
Figure 5.21: The innovations, filtered residue and smoothed residue of measurement 4
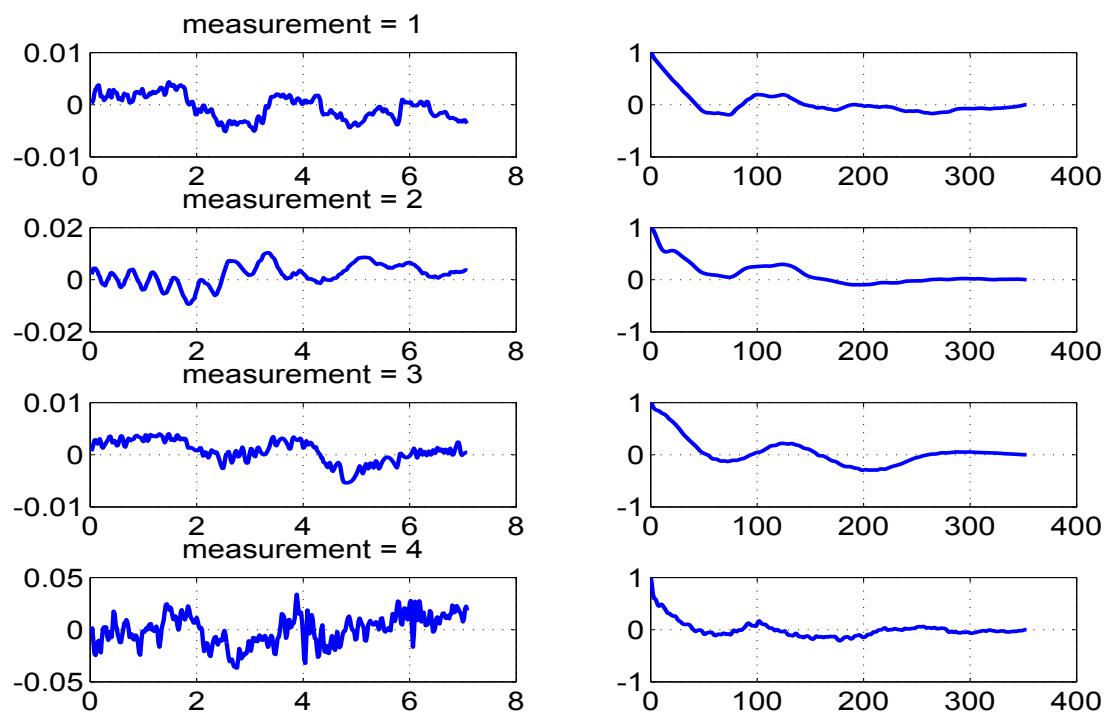
Figure 5.22: Time variation of estimated measurement noise (left) and

their autocorrelation (right)

5.1.4 Case-1 Figures (Q > 0)
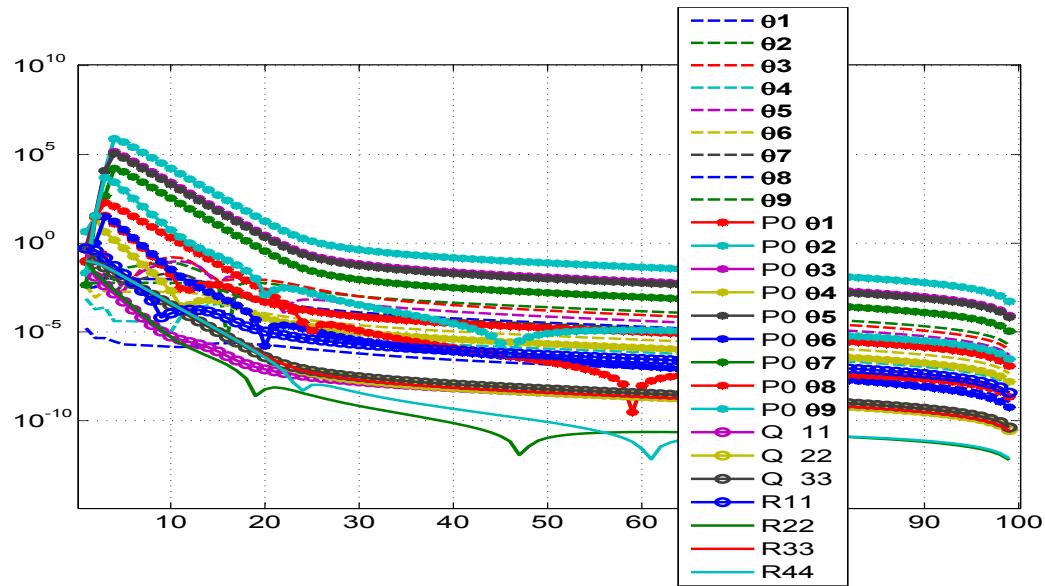
Figure 5.23: The absolute difference between the iterated and final values

with 100 iterations
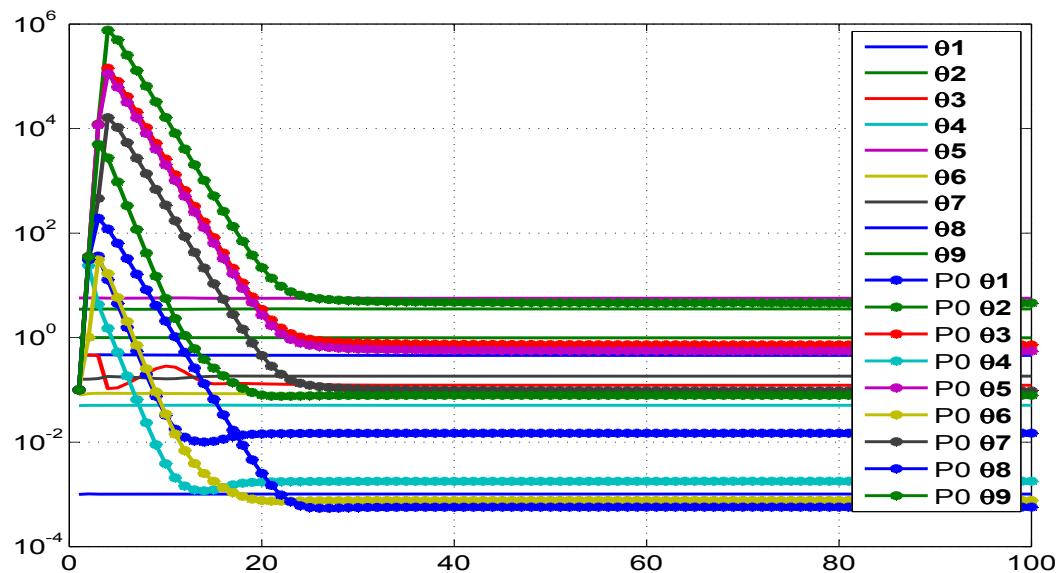
Figure 5.24: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
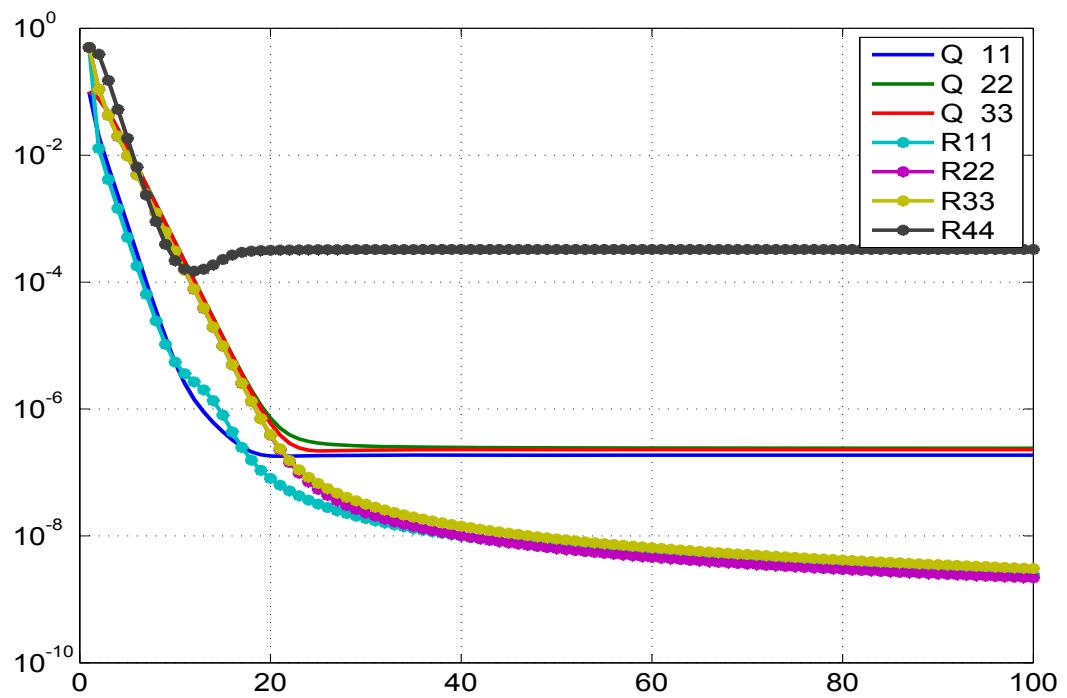
Figure 5.25: Variation of **R** and **Q** with iterations
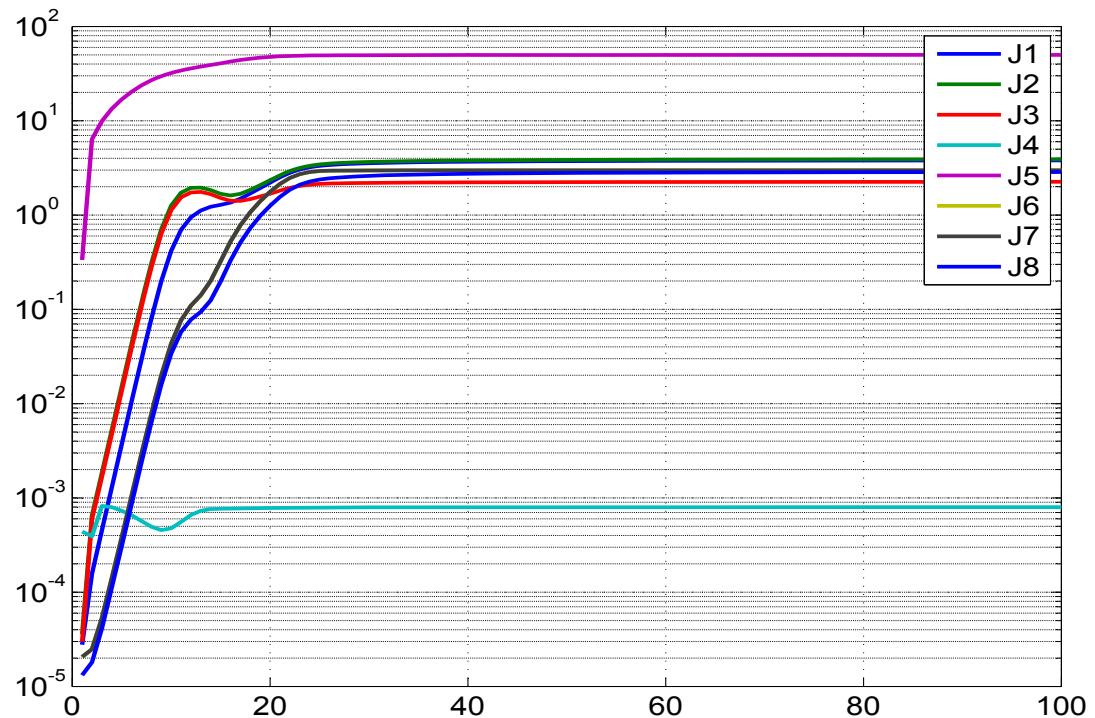
Figure 5.26: Variation of different costs (**J1-J8**) with iterations

235
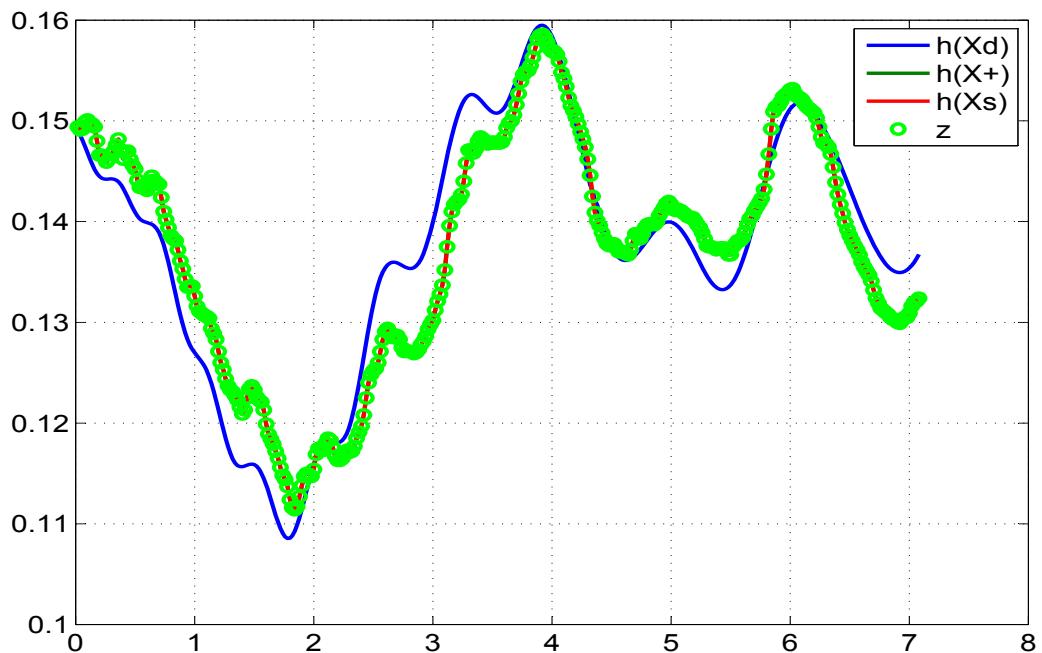
Figure 5.27: Comparison of the predicted dynamics, posterior, smoothed
and the measurement 1
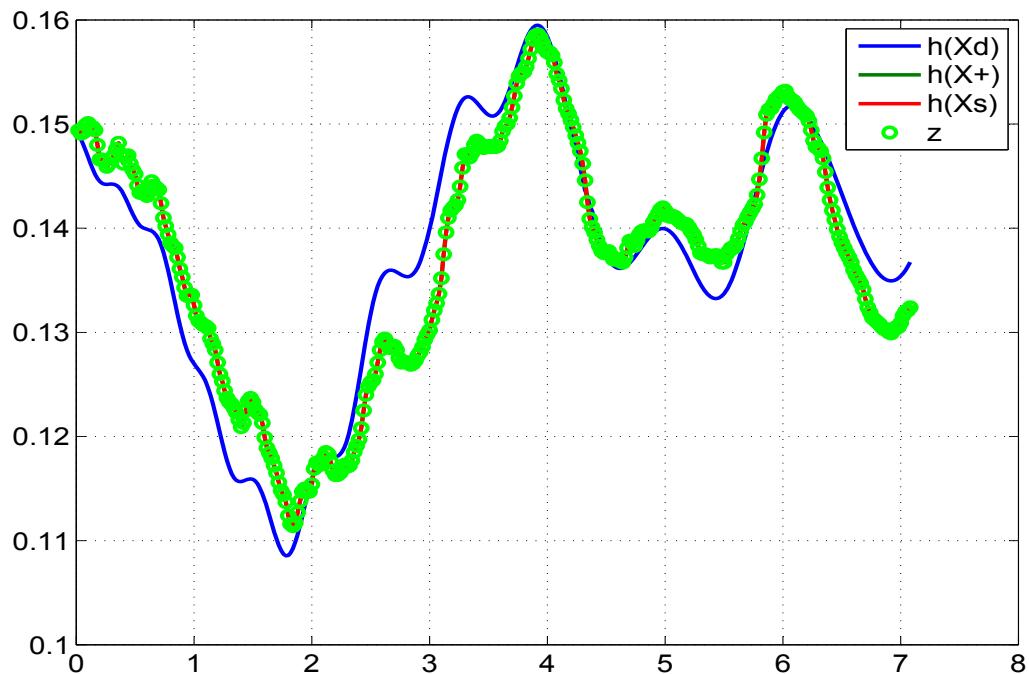
Figure 5.28: Comparison of the predicted dynamics, posterior, smoothed
and the measurement 2
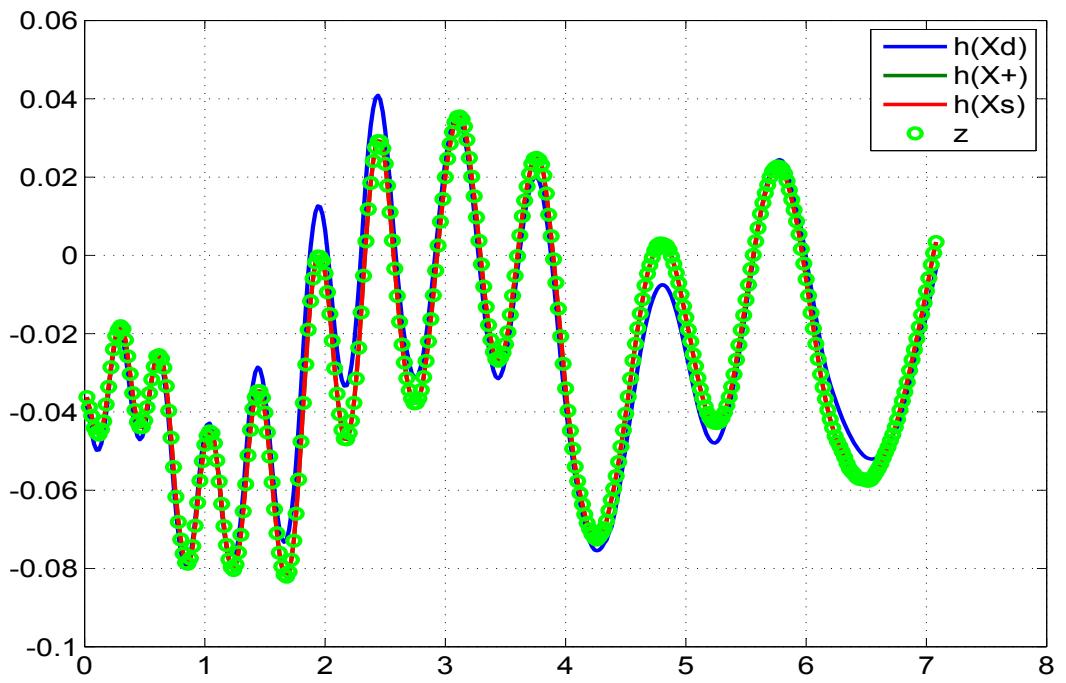
Figure 5.29: Comparison of the predicted dynamics, posterior, smoothed and the measurement 3
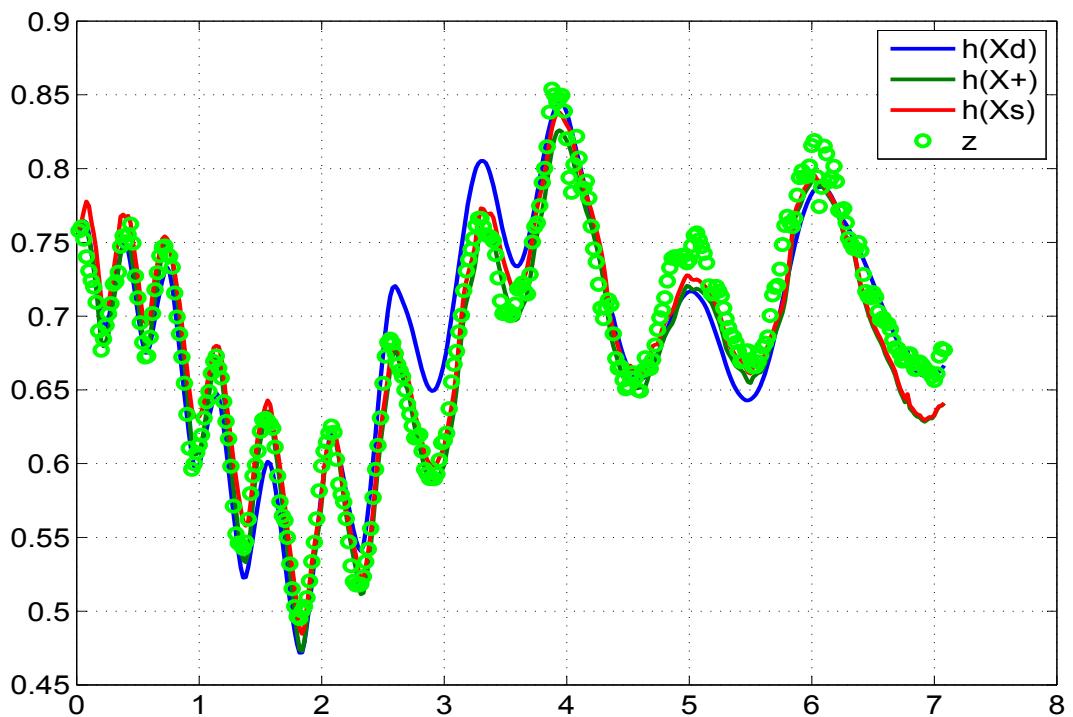
Figure 5.30: Comparison of the predicted dynamics, posterior, smoothed and the measurement 4

237
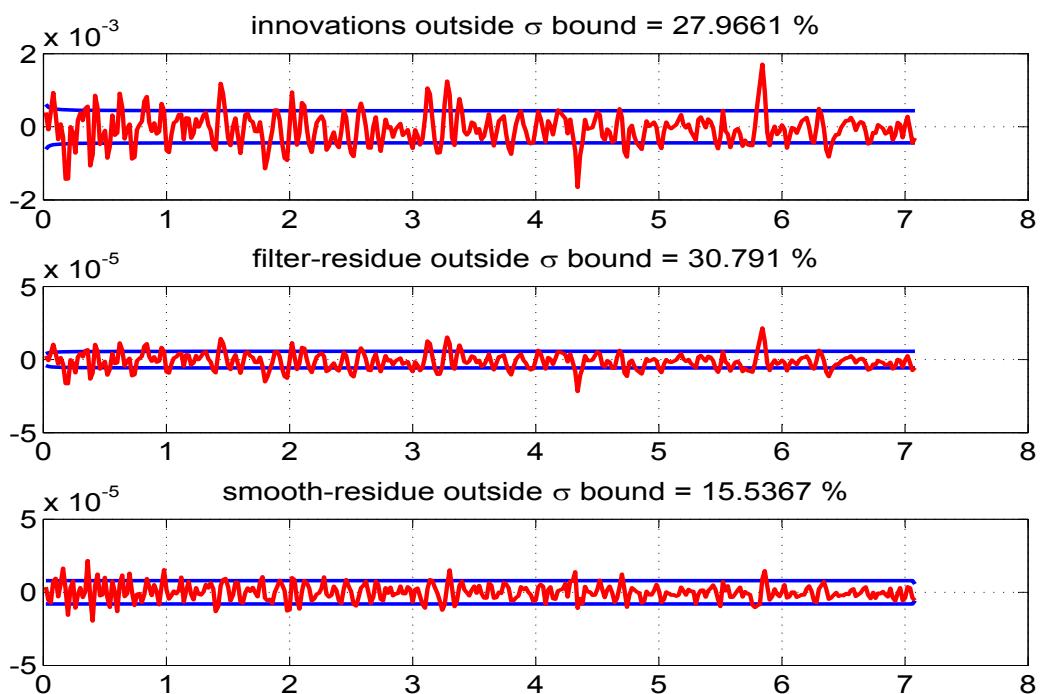
Figure 5.31: The innovations, filtered residue and smoothed residue of measurement 1
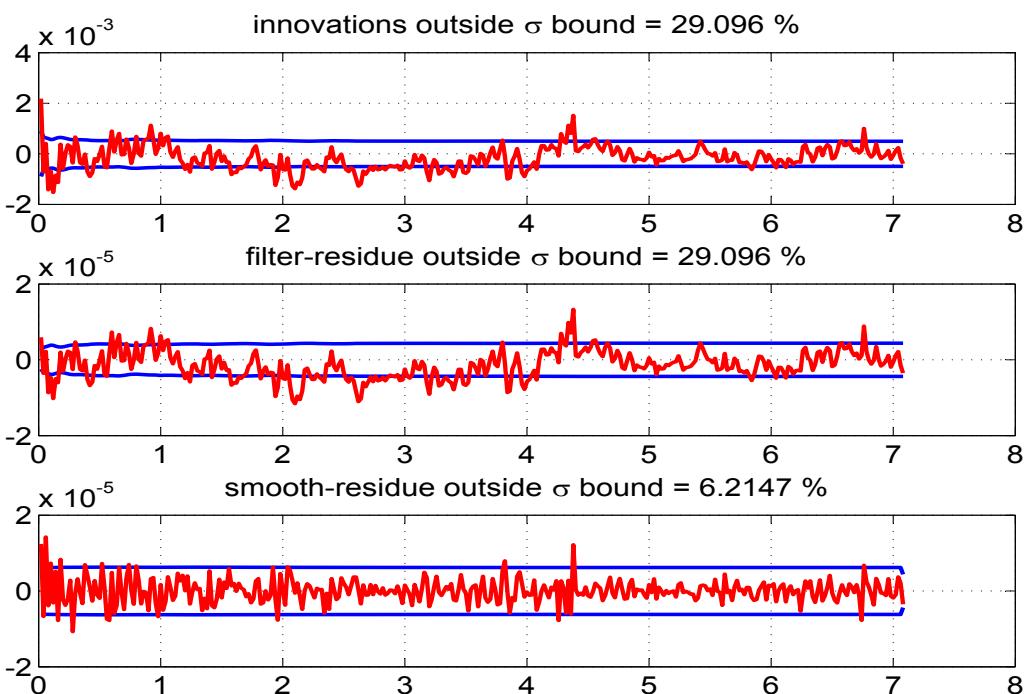
Figure 5.32: The innovations, filtered residue and smoothed residue of measurement 2
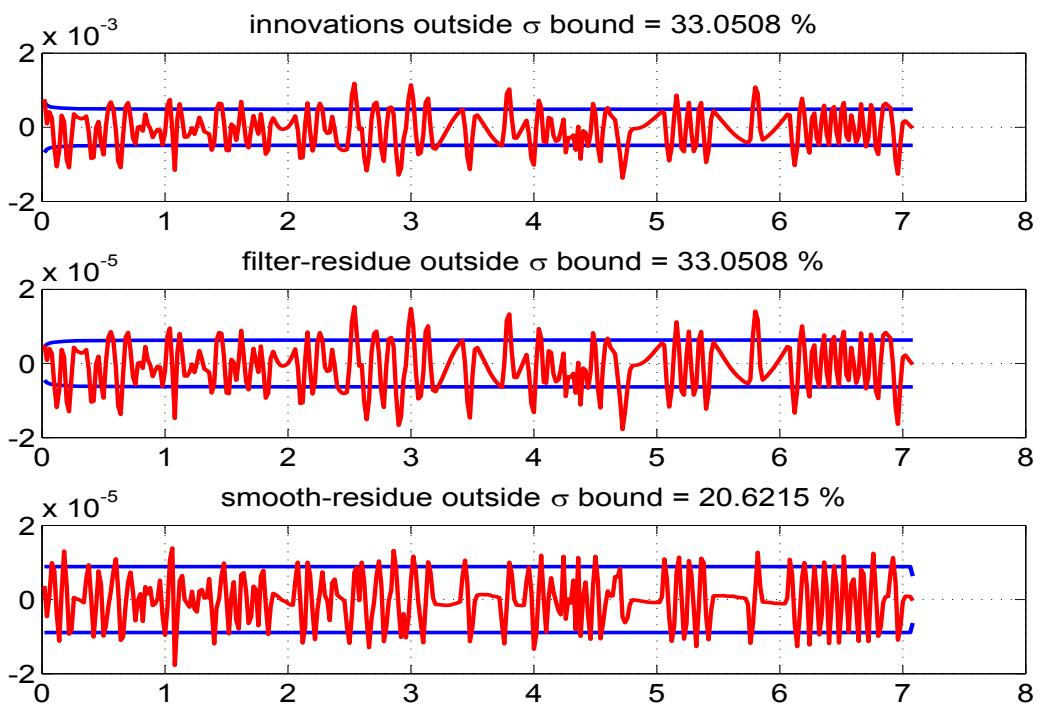
Figure 5.33: The innovations, filtered residue and smoothed residue of measurement 3
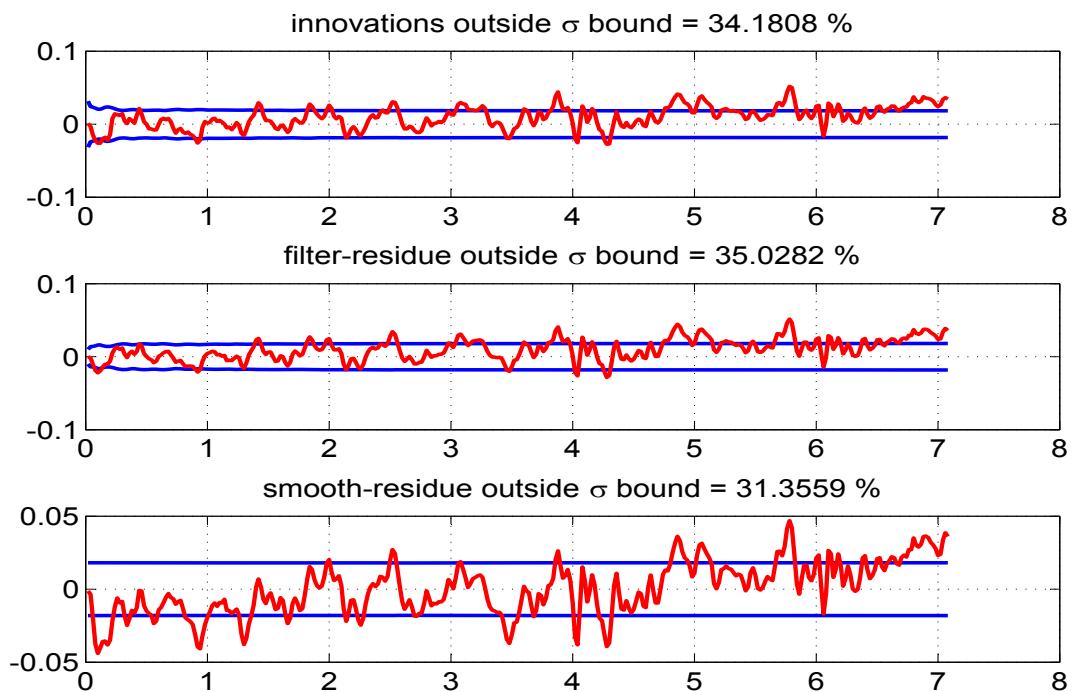
Figure 5.34: The innovations, filtered residue and smoothed residue of measurement 4
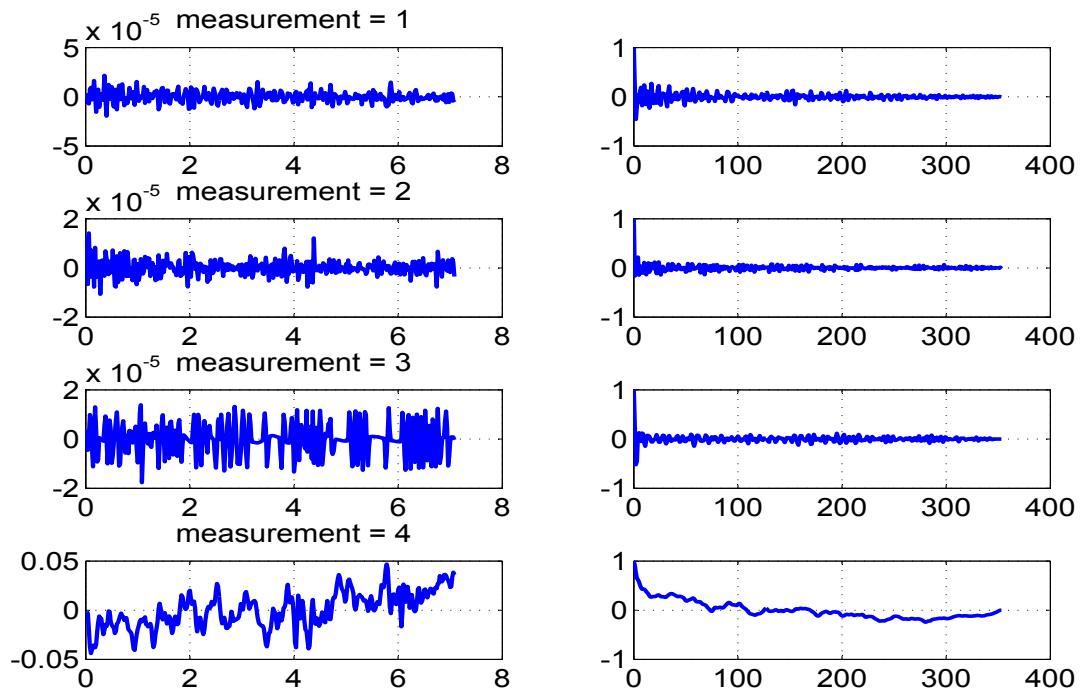
Figure 5.35: Time variation of estimated measurement noise (left) and
their autocorrelation (right)
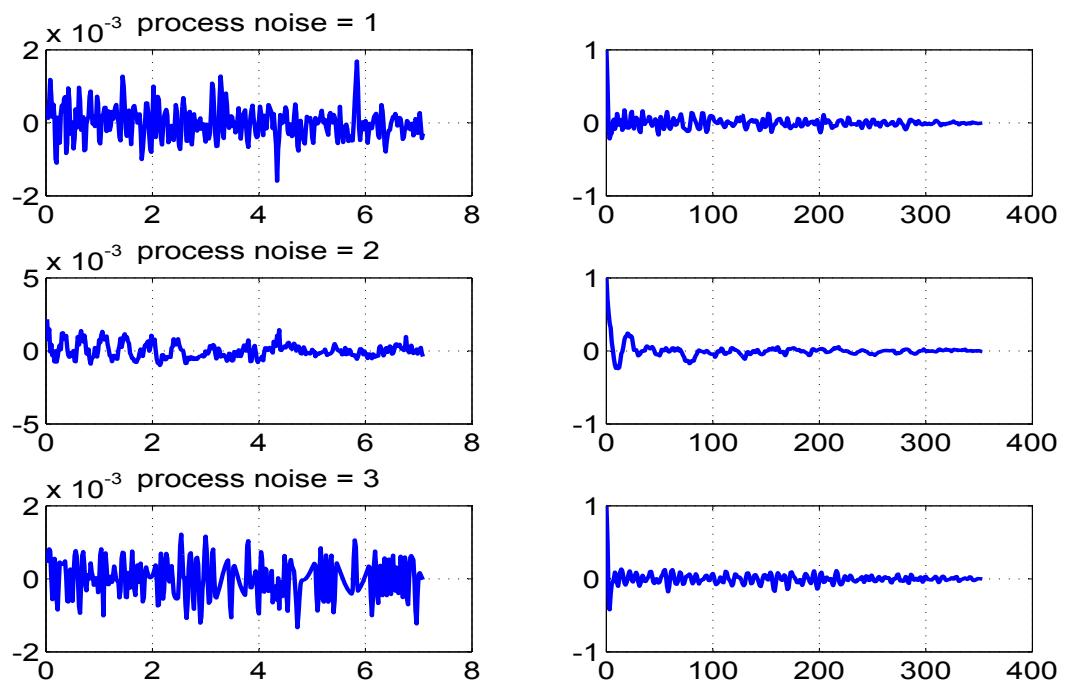
Figure 5.36: Time variation of estimated process noise (left) and
their autocorrelation (right)

240
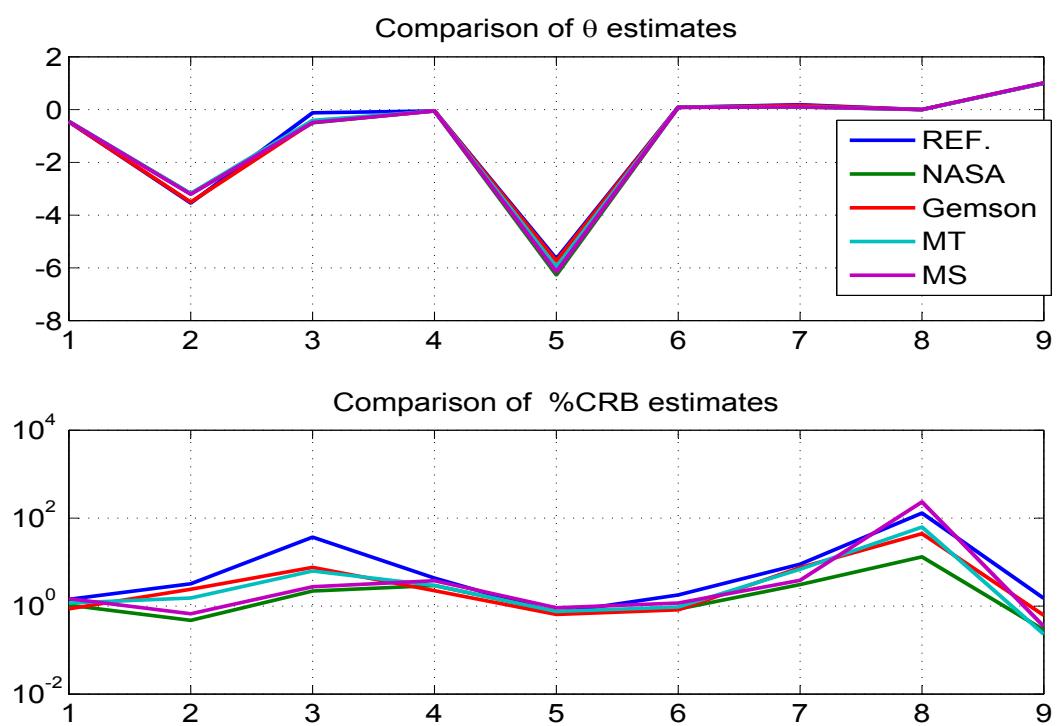
Figure 5.37: Comparison of the parameter estimates and %CRBs by different methods

5.2 Real Flight Test Case-2

The salient features of this aircraft are available in NASA TM-X 56036 (Shafer[30] 1975) and in NASA TP 1690 (Maine and Iliff[46] 1981). The parameters are estimated in dimensionless form. The data set is obtained is for a short period motion excited by the up and down elevator control input (δ_e in radians) as shown in Fig. 5.38. Some of the available measurements can also be used as inputs in the state equations which includes roll angle (ϕ_m), sideslip (β_m), roll rate (p_m), yaw rate (r_m) and the angle of attack (α_m) are shown in Fig. 5.39, Fig. 5.40, Fig. 5.41, Fig. 5.42 and Fig. 5.46 respectively. The state equations ($n = 3$) for the angle of attack (α), pitch rate (q) and the pitch angle (Θ) respectively are

$$\dot{\alpha} = -\frac{\bar{q}S}{mV}C_L + q + \frac{g}{V}(cos(\phi_m)cos(\alpha_m)cos(\theta) + sin(\alpha_m)sin(\theta)) -$$
$$\beta_m(p_m cos(\alpha_m) + r_m sin(\alpha_m))$$
$$\dot{q} = \frac{\bar{q}S\bar{c}}{Iyy}(C_{m_\alpha}\alpha + C_{m_q}\frac{\bar{c}}{2V}q + C_{m_{\delta_e}}\delta_e + C_{m_0}) + \frac{Izz - Ixx}{Iyy}r_m p_m$$
$$\dot{\theta} = q cos(\phi_m) - r_m sin(\phi_m) + \theta_0$$

The angle of attack (α), pitch rate (q), the pitch angle (θ), normal acceleration (a_n) and the longitudinal acceleration (a_x) are measured (indicated with subscript 'm') in units of rad, rad/s, rad, ft/s^2 and ft/s^2 respectively. The measurement equations (m=5) are

$$\alpha_m = \alpha - K_\alpha x_\alpha \frac{q}{V}$$
$$q_m = q$$
$$\theta_m = \theta$$
$$a_{n_m} = \frac{\bar{q}S}{mg}C_N + \frac{x_{a_n}}{g}\dot{q}$$
$$a_{x_m} = -\frac{\bar{q}S}{mg}C_A + \frac{z_{a_x}}{g}\dot{q}$$

where

$$C_L = C_N cos(\alpha) - C_A sin(\alpha) + C_{L_0}$$
$$C_N = C_{N_\alpha}\alpha + C_{N_{\delta_e}}\delta_e + C_{N_0}$$
$$C_A = C_{A_\alpha}\alpha + C_{A_{\alpha^2}}\alpha^2 + C_{A_{\delta_e}}\delta_e + C_{A_0}$$

The unknown parameter set ($p = 13$) is $\Theta = (C_{N_\alpha}, C_{N_{\delta_e}}, C_{L_0}, C_{m_\alpha}, C_{m_q}, C_{m_{\delta_e}}, C_{m_0}, \theta_0,$ $C_{N_0}, C_{A_\alpha}, C_{A_{\alpha^2}}, C_{A_{\delta_e}}, C_{A_0})^T$. The ones with suffix '$\delta_e$' are the control derivatives, the ones with suffix zero are the biases and all others are aerodynamic derivatives. The initial states are taken as initial measurement and the initial parameter values are taken as $(4, 0.24, 0.17, -0.48, -17, -0.9, -0.05, -0.02, 0.175, -0.3, 0.03, -0.083, -0.015)^T$.

Other constant values used for case-2

$\bar{c}$=5.58	S=184	m=172.667	Ixx=4142.9	Iyy=3922.4	Izz=7642.5
g=32.2	V=403.1	$\bar{q}$=83.08	$K_\alpha x_\alpha$=-0.0279	x_{a_n}=0.101	z_{a_x}=-1.17

Case-2 real data is run using the reference EKF ($\mathbf{Q} > 0$) with 100 iterations. The Fig. 5.38-5.42 are the inputs used in state equations. The Fig. 5.43 shows the variation of parameter estimates and its initial covariance $\mathbf{P_0}$ with iterations and a similar Fig. 5.44 for $\mathbf{Q}$ and $\mathbf{R}$. The values of **J1-J3** are close to the number of measurements ($m = 5$) with **J6-J8** are close to the number of states ($n = 3$) as shown in Fig. 5.45 and Table-5.5. This means the measurement and state equations are well balanced. The **J5** is the negative log likelihood cost function. The later Fig. 5.46-5.50 compares (i) the state dynamics based on the estimated parameter after the filter pass through the data, (ii) the state after measurement update, (iii) the smoothed state and (iv) the measurement. The Fig. 5.51-5.55 shows the confidence in the innovations, filtered residue and smoothed residue. The estimated measurement and process noise do not appear to have constant statistical characteristics across time as seen in Fig. 5.56 and Fig. 5.57. Another experiment was carried out by generating a typical data set by using the estimated theta and injecting the estimated $\mathbf{Q}$ and $\mathbf{R}$ as additive white Gaussian noise. This is to determine the effect of non additive, non White and non Gaussian noise distribution in the real data on the CRBs. After each iteration in the reference recipe the Θ, $\mathbf{Q}$ and $\mathbf{R}$ were reset as from the real data. Similar experiment was also conducted by updating Θ as well. It turned out that there is not much of a difference in the final CRBs as can be seen from the Table-5.6.

Finally two other filter runs were carried out using the MT and MS statistics for the estimation of $\mathbf{Q}$ and $\mathbf{R}$. The behaviour of the various cost function and in particular **J6** and J7 in Table-5.7 shows that the choice of the filter statistics for estimating $\mathbf{Q}$ and $\mathbf{R}$ in the proposed reference approach is the best possible when compared to other approaches presently considered. Another feature of recursive parameter estimation is that it can vary through time instants and point to two distinct values as reflected in the estimation of C_{N_α} in Fig. 5.58. This feature of tracking time varying parameters by the EKF brings in

clearly another advantage of sequential processing instead of batch processing of the data by a least squares (LS) procedure. If LS had been used then for the parameter 'C_{N_α}' only an average value would have been obtained. If one persists in using the LS procedure then to get the varying parameter such a feature should be modelled which would include where to change and one might have to use the data in blocks by splitting them and all such exercises have to be carried out which are not easy. Thus another way of looking at the filter it seems to be doing a better job of using a more appropriate range of data to estimate the parameters. It is doing a good job of time averaging the state estimates in a fluctuating situation created by the measurement noise. However it should be cautioned that the filter has to be tuned very well for such a situation. This may not be an easy task if the system dynamics is extremely fast.

5.2.1 Remarks on Case-2

The NASA results have been generated assuming $\mathbf{Q} = 0$ and are comparable with reference procedure for the parameter estimates and their CRBs. Further the MT and MS methods give quite different estimates for the $\mathbf{Q}$ and $\mathbf{R}$ values than in the reference case. We believe that the reference procedure provides the best possible parameter estimates and their uncertainties. From the plot of the parameter estimates and their %CRB in Fig 5.59, it can be seen that the parameters 1-7, 10, 11 are strong and the parameters 8 and 12 are the weak ones. The CRBs as estimated by different methods generally appear to vary widely. However what is interesting is that even the estimate of the strong parameter such as 5 varies widely among the methods. Such a behaviour of the filter across the parameter estimates shows how important is the tuning of the filter statistics namely $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$ in parameter estimation and their uncertainties.

The rounded off 100×C matrix for case-2 is given by

$$
\begin{bmatrix}
100 & -91 & 0 & 26 & 6 & -23 & 21 & 0 & 86 & 26 & 26 & -22 & 20 \\
-91 & 100 & 0 & -23 & -4 & 25 & -24 & 0 & -98 & -23 & -20 & 22 & -21 \\
0 & 0 & 100 & -2 & 0 & 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\
26 & -23 & -2 & 100 & 18 & -90 & 86 & 2 & 22 & 32 & 29 & -29 & 27 \\
6 & -4 & 0 & 18 & 100 & -14 & -2 & 0 & 0 & 6 & 8 & -5 & -1 \\
-23 & 25 & 2 & -90 & -14 & 100 & -98 & -3 & -24 & -29 & -23 & 32 & -31 \\
21 & -24 & -2 & 86 & -2 & -98 & 100 & 3 & 24 & 27 & 21 & -31 & 32 \\
0 & 0 & 0 & 2 & 0 & -3 & 3 & 100 & 0 & 0 & 0 & 0 & 0 \\
86 & -98 & 0 & 22 & 0 & -24 & 24 & 0 & 100 & 21 & 18 & -21 & 21 \\
26 & -23 & 0 & 32 & 6 & -29 & 27 & 0 & 21 & 100 & 91 & -91 & 86 \\
26 & -20 & 0 & 29 & 8 & -23 & 21 & 0 & 18 & 91 & 100 & -73 & 67 \\
-22 & 22 & 0 & -29 & -5 & 32 & -31 & 0 & -21 & -91 & -73 & 100 & -98 \\
20 & -21 & 0 & 27 & -1 & -31 & 32 & 0 & 21 & 86 & 67 & -98 & 100
\end{bmatrix}
$$

5.2.2 Case-2 Tables

Table 5.5: Real flight test data case-2 results using the reference adaptive EKF.
No of iterations=100

Study	Θ (Ref)	Θ (NASA)	Θ (Gemson)	σ_Θ (Ref)	σ_Θ (NASA)	σ_Θ (Gemson)	R $\times 10^{-6}$ (Ref)	Q $\times 10^{-6}$ (Ref)	J1-J8 (Ref)	Remarks
	4.6469	4.9584	4.7073	0.0179	0.1168	0.039				
	0.0555	0.3023	0.1292	0.0277	0.1550	0.0523				
	0.0162	0.2189	-0.0064	0.0032	0.009344	0.0048			4.4752	
	-0.5468	-0.6125	-0.63	0.0093	0.00953	0.0188			5.1532	
	-19.8027	-22.27	-20.8623	0.6692	0.7713	1.1908	0.49		4.6432	
$\mathbf{P_0}$: Scaled up-[0,0;0,✓]	-1.1229	-1.2193	-1.2763	0.0218	0.02881	0.0442	0.04	0.134	0.0004	Cost functions converge
$\mathbf{Q}$: EM-[✓,0;0,0]	-0.0495	-0.0532	-0.0561	0.0012	0.00165	0.0023	0.40	2.287	-56.2206	to the expected values
$\mathbf{R}$: EM-diag	0.0007	0.0273	0.0007	0.0021	0.04518	0.0135	15.98	1.204	2.9551	
	0.2195	0.2254	0.2225	0.0014	0.008725	0.0029	17.70		2.9303	
	-0.1398	-0.3639	-0.1023	0.0153	0.05328	0.0214			2.5161	
	-3.2088	–	-3.2397	0.1702	–	0.2430				
	-0.0651	-0.07	-0.0267	0.0134	0.08084	0.0191				
	-0.0155	-0.0131	-0.0144	0.0007	0.004088	0.0010				

Table 5.6: Case-2 results using simulated Additive White Gaussian Noise

Study	σ_Θ (Simulated-without updating Θ)	σ_Θ (Simulated-with updating Θ)	σ_Θ (Ref)	Remarks
Case-2 data generated using simulated measurement and process noise (AWGN) of variance $\mathbf{Q}$ and $\mathbf{R}$ estimated by Reference EKF ($\mathbf{Q} > 0$)				
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: $\mathbf{Q}$ (Ref) $\mathbf{R}$: $\mathbf{R}$ (Ref)	0.0178	0.0178	0.0179	No Significant change in σ_Θ
	0.0278	0.0278	0.0277	
	0.0032	0.0032	0.0032	
	0.0095	0.0095	0.0093	
	0.6887	0.6889	0.6692	
	0.0237	0.0238	0.0218	
	0.0013	0.0013	0.0012	
	0.0021	0.0021	0.0021	
	0.0014	0.0014	0.0014	
	0.0140	0.0140	0.0153	
	0.1670	0.1670	0.1702	
	0.0136	0.0136	0.0134	
	0.0007	0.0007	0.0007	

Table 5.7: Real flight test data case-2 results using the MT and MS method.
No of iterations=100

Study	Θ (MT)	σ_Θ (MT)	R (MT) $\times 10^{-6}$	Q (MT) $\times 10^{-6}$	J1-J8 (MT)	Θ (MS)	σ_Θ (MS)	R (MS) $\times 10^{-6}$	Q (MS) $\times 10^{-6}$	J1-J8 (MS)	Remarks
	4.6978	0.0229				4.9141	0.0422				
	0.1225	0.0357				0.4691	0.0517				
	0.0160	0.0018				0.0184	0.0021				
	-0.5560	0.0098			4.0090	-0.5885	0.0036			3.3893	
	-19.7062	0.7286	0.4107		3.9630	-20.2395	0.2937	3.2046		3.3866	
$\mathbf{P_0}$: Scaled up-[0,0;0,✓]	-1.1396	0.0236	0.0312	0.0393	2.9764	-1.1503	0.0111	37.6770	0.0001	3.2057	Cost functions are
$\mathbf{Q}$: MT/MS-[✓,0;0,0]	-0.0502	0.0013	3.9381	2.6418	0.0004	-0.0497	0.0006	7.5509	0.0015	0.0002	not close to their
$\mathbf{R}$: MT/MS-diag	0.0008	0.0011	94.5086	0.3231	-54.7596	0.0003	0.0012	198.2716	0.3456	-49.6223	expected values in
	0.2218	0.0018	26.3511		6.6681	0.2358	0.0028	28.9841		3.8921	MT and MS method
	-0.1401	0.0185			6.4985	-0.1265	0.0197			4.7110	
	-3.2088	0.2070			2.4562	-3.8625	0.2376			2.6369	
	-0.0633	0.0160				-0.1178	0.0167				
	-0.0154	0.0008				-0.0182	0.0008				

5.2.3 Case-2 Figures

Figure 5.38: Control input (δ_e) versus time (s)
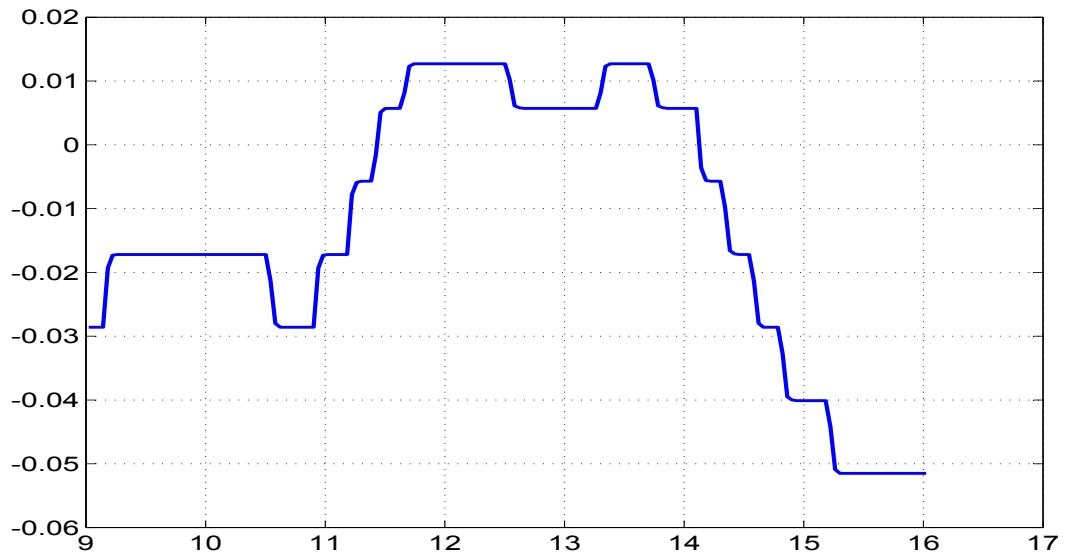
Figure 5.39: Measurement input (ϕ_m) versus time (s)

Figure 5.40: Measurement input (β_m) versus time (s)
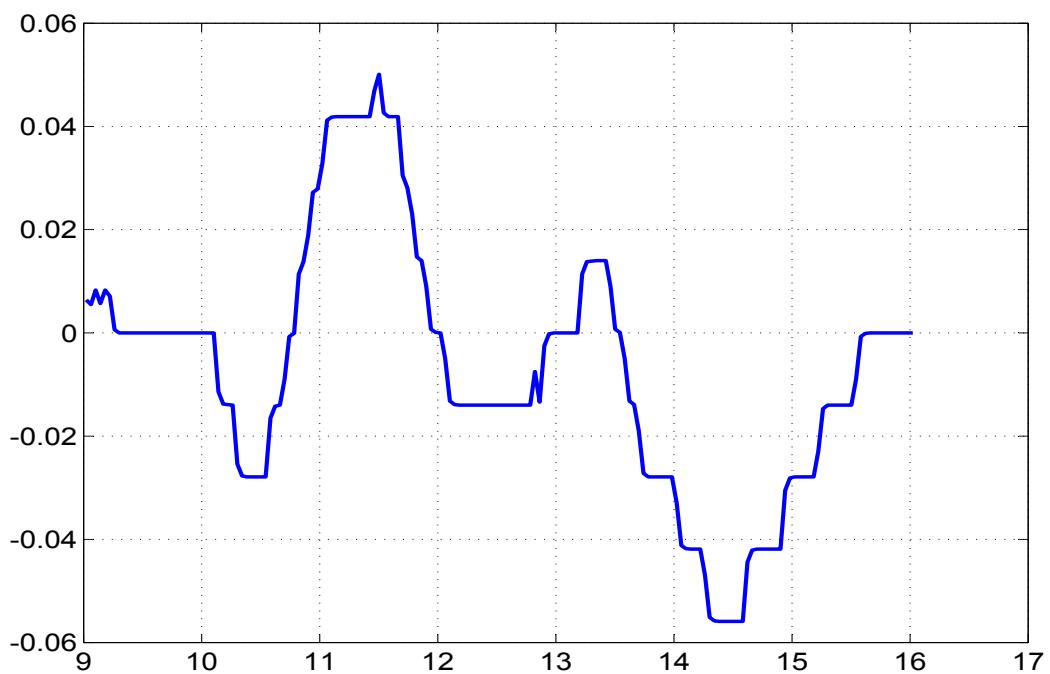
Figure 5.41: Measurement input (p_m) versus time (s)
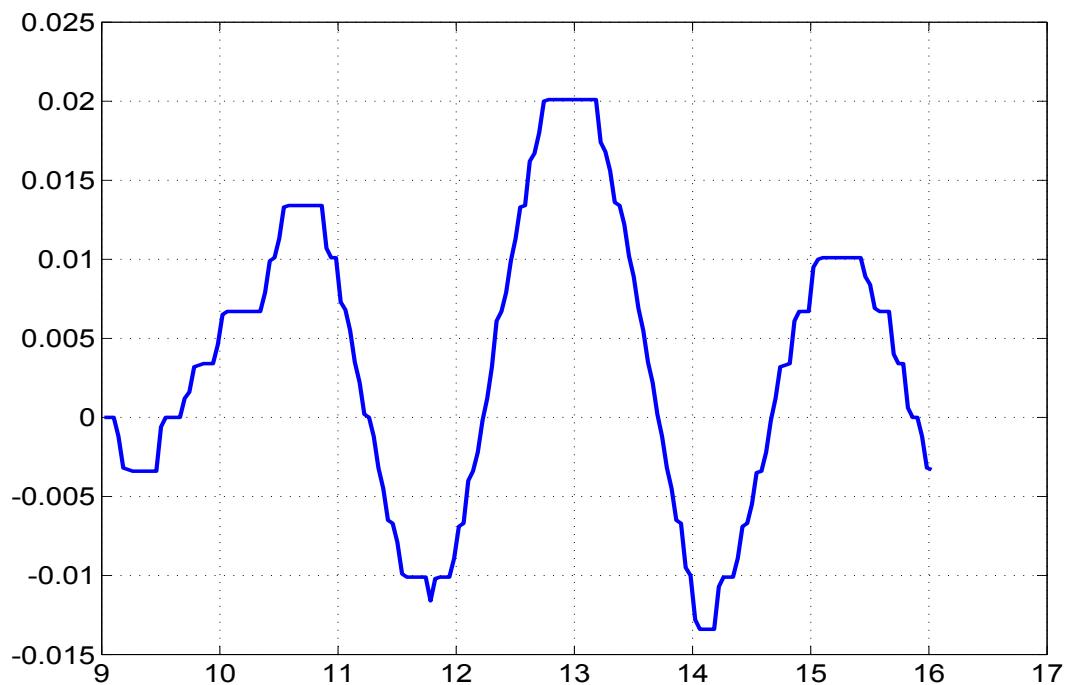
Figure 5.42: Measurement input (r_m) versus time (s)
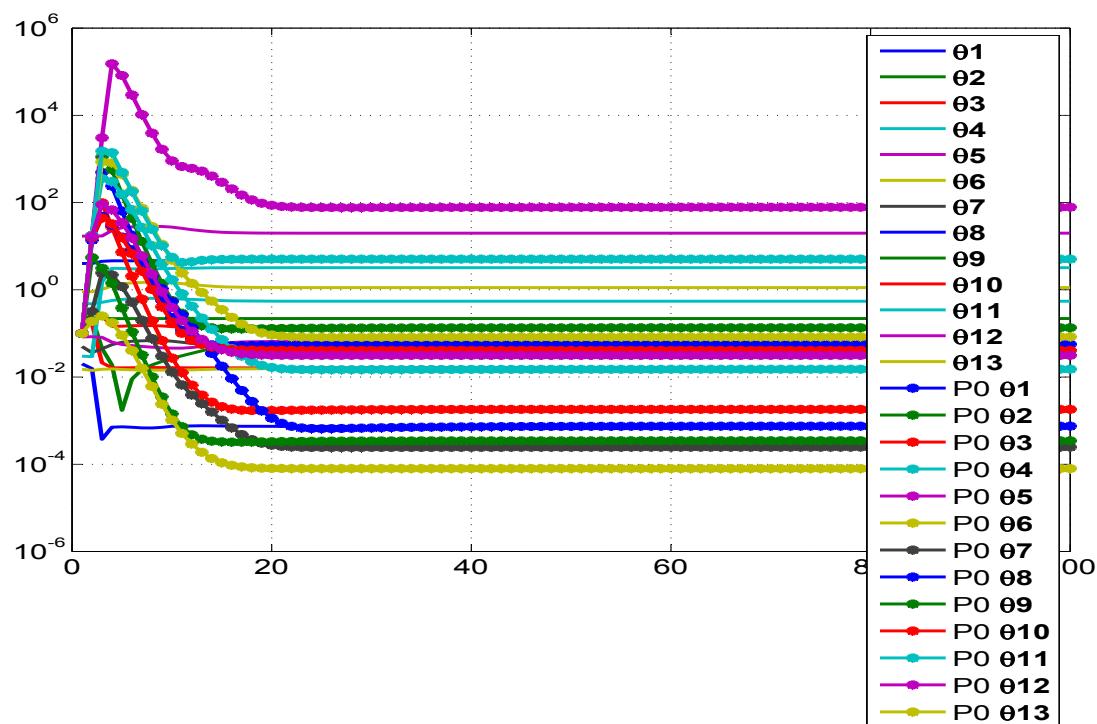
Figure 5.43: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
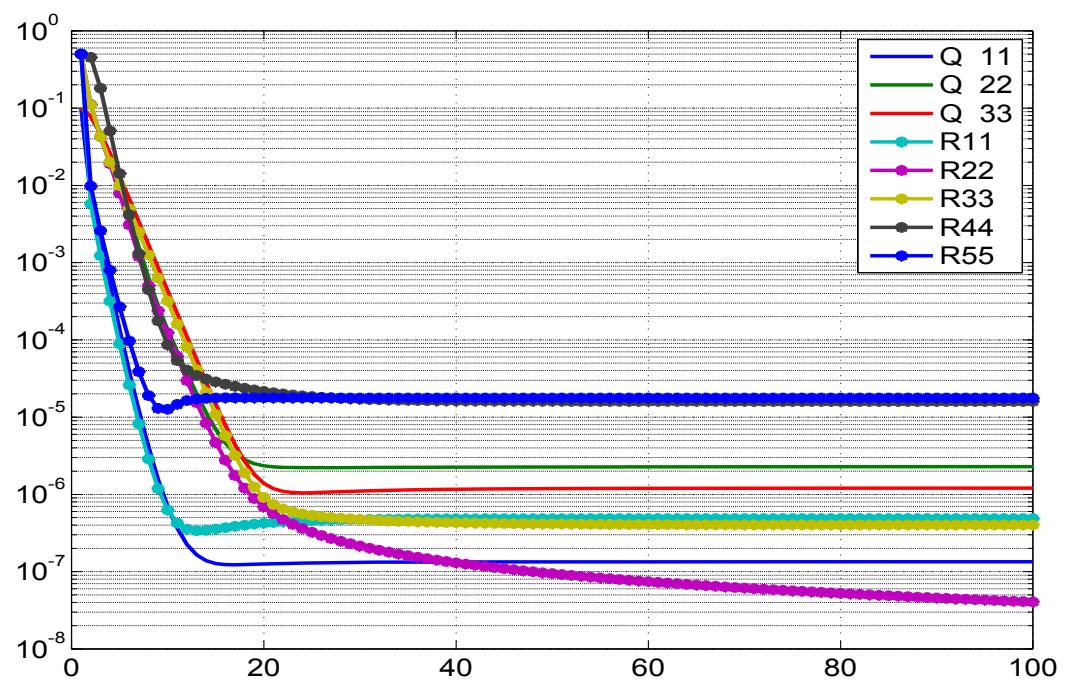
Figure 5.44: Variation of **Q** and **R** with iterations
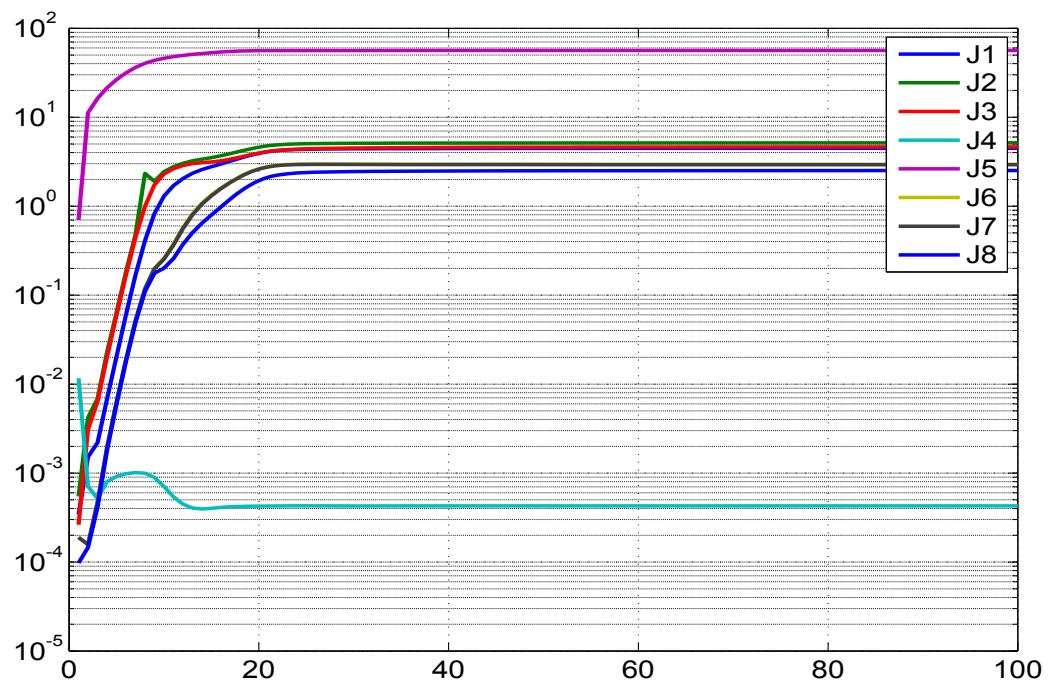
Figure 5.45: Variation of different costs (**J1-J8**) with iterations

252
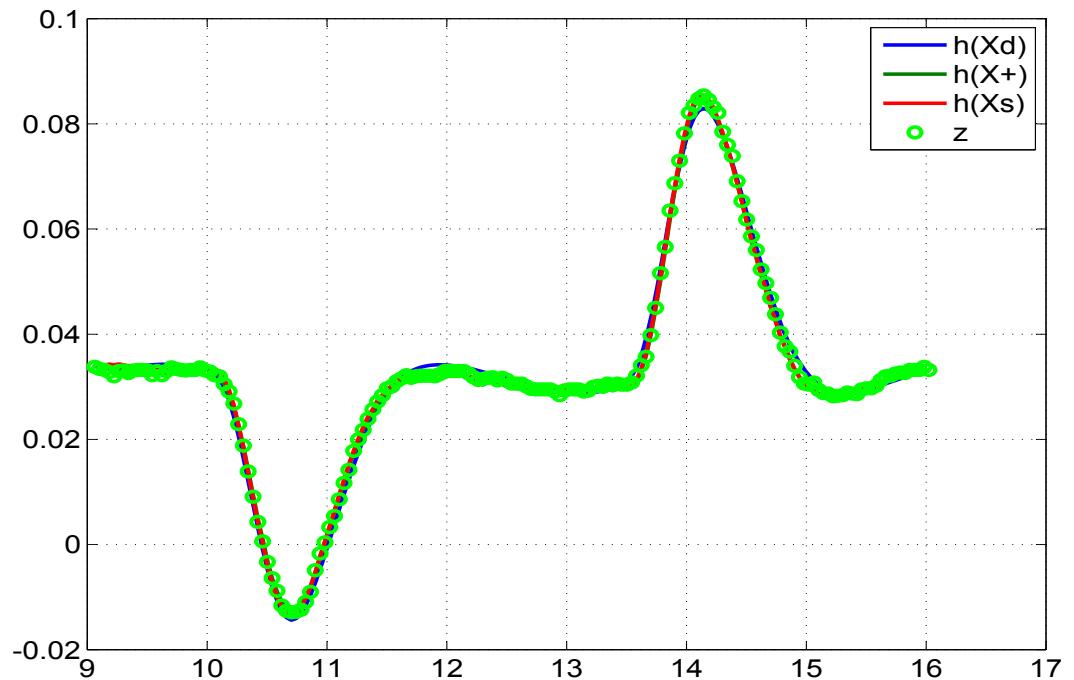
Figure 5.46: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1
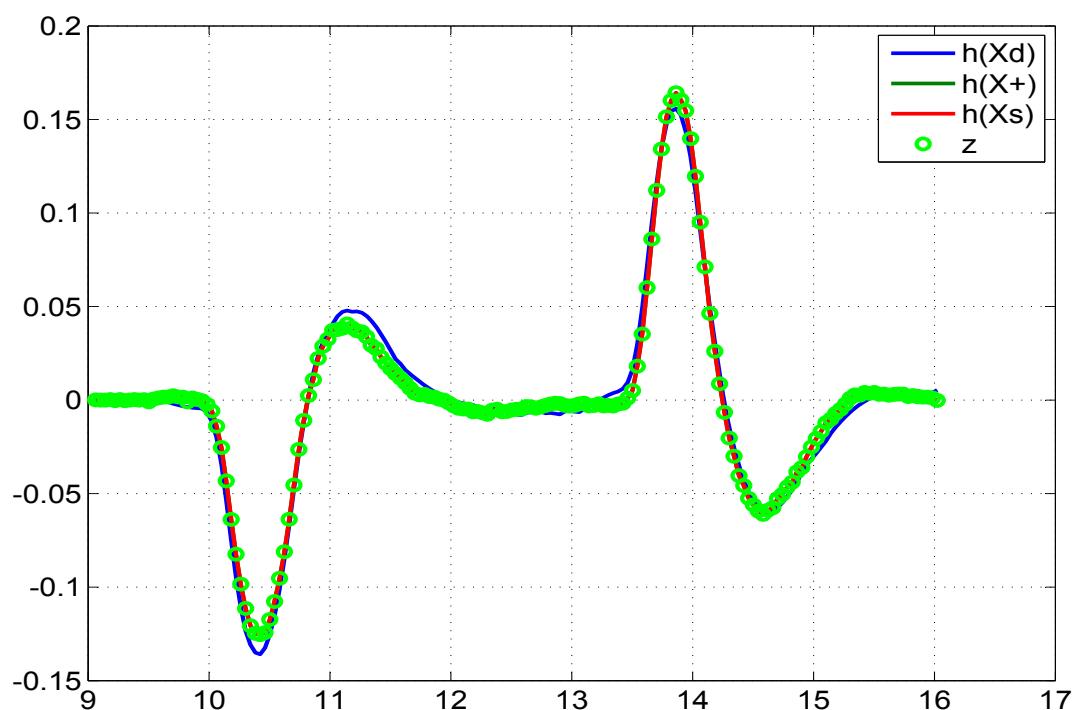
Figure 5.47: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 2
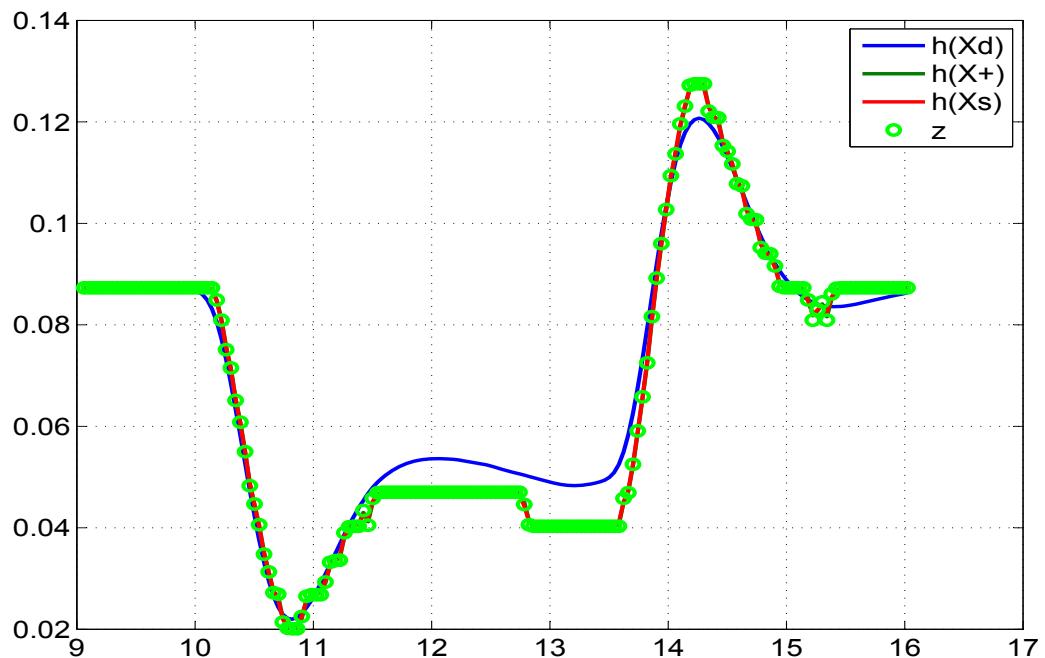
Figure 5.48: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 3
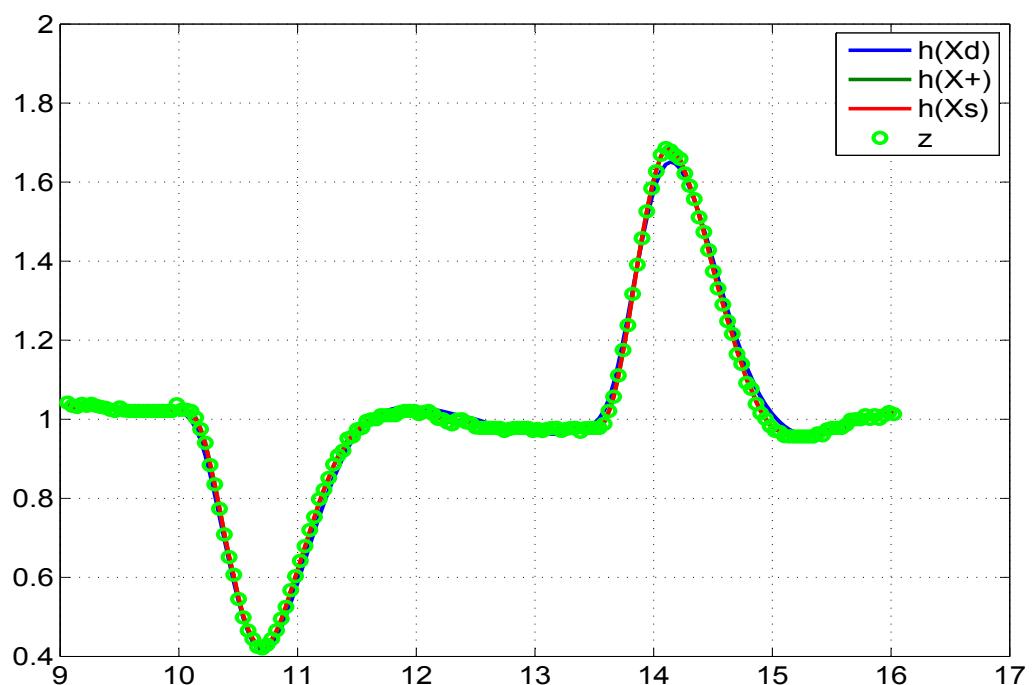
Figure 5.49: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 4
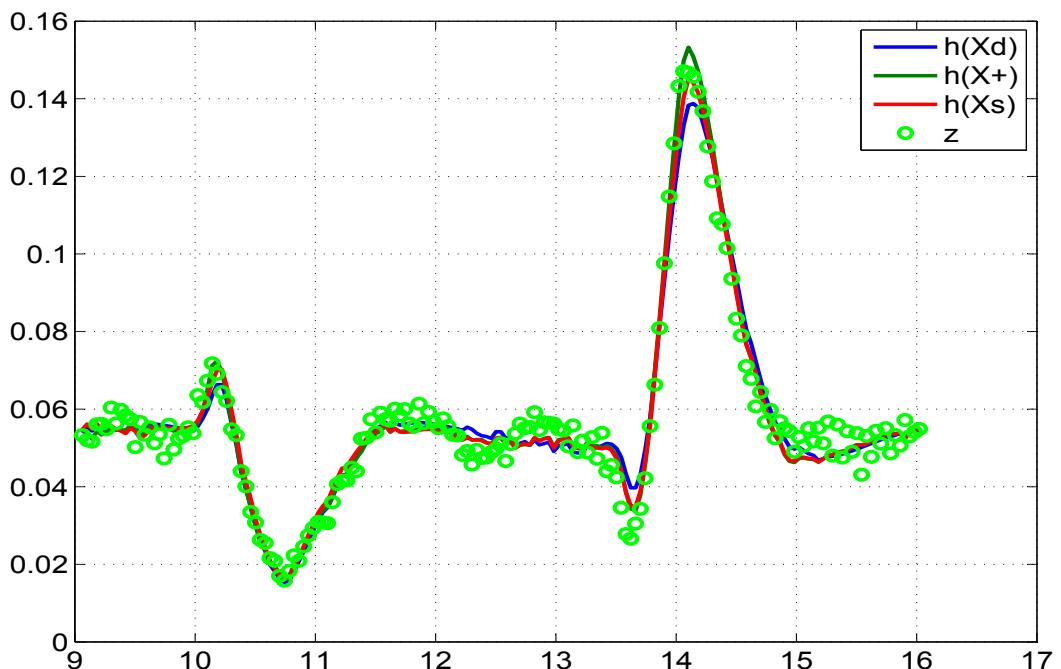
Figure 5.50: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 5
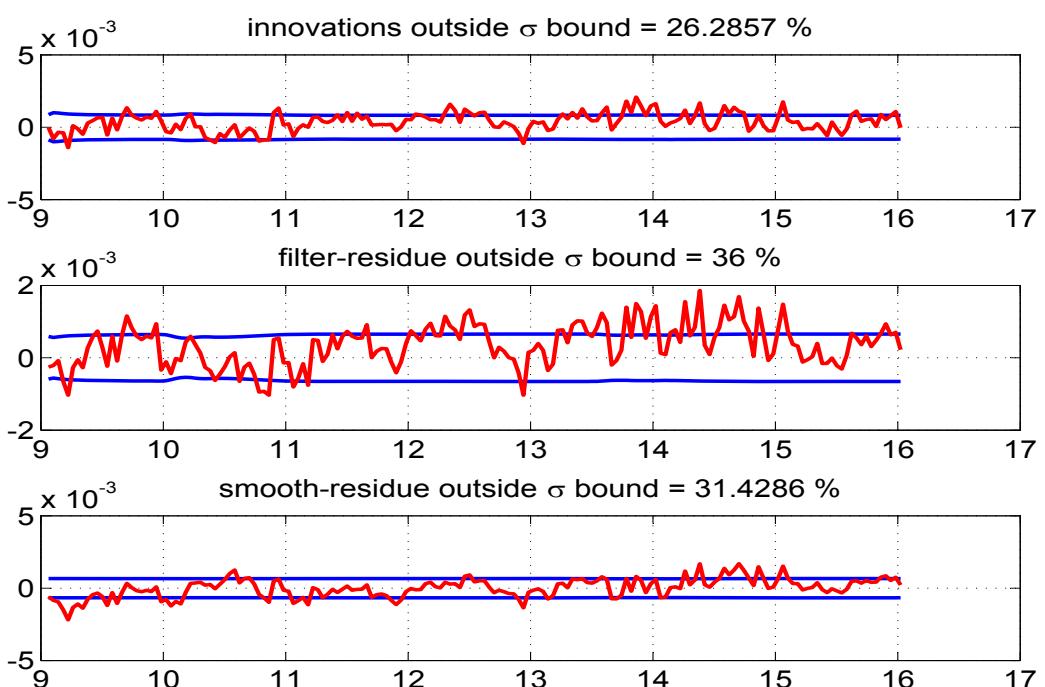
Figure 5.51: The innovations, filtered residue and smoothed residue of measurement 1
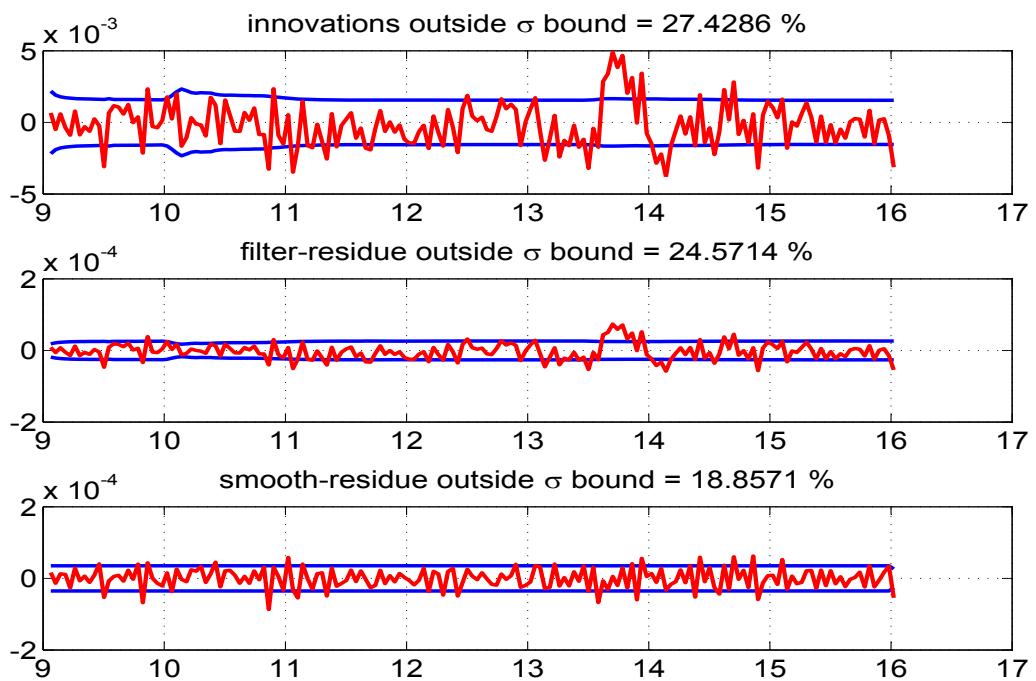
Figure 5.52: The innovations, filtered residue and smoothed residue of measurement 2
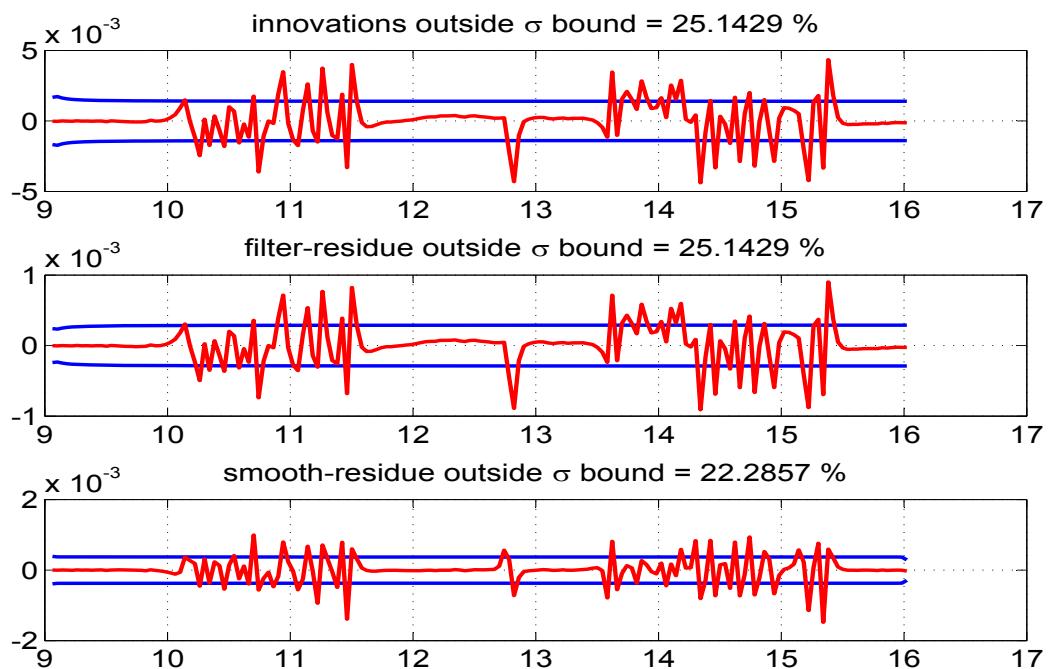
Figure 5.53: The innovations, filtered residue and smoothed residue of measurement 3
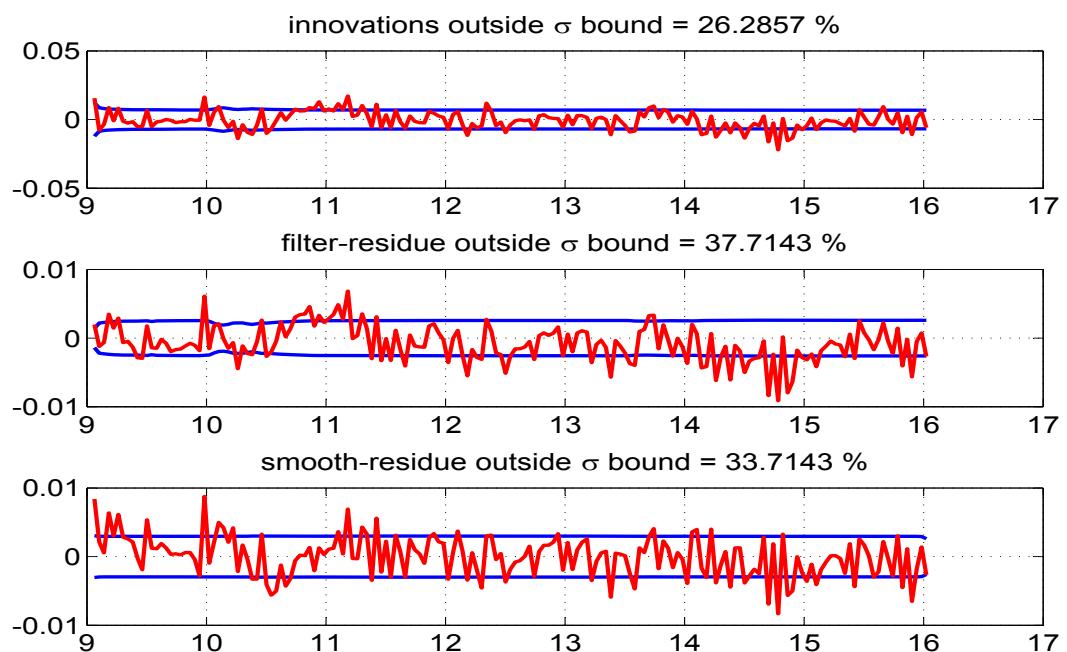
Figure 5.54: The innovations, filtered residue and smoothed residue of measurement 4
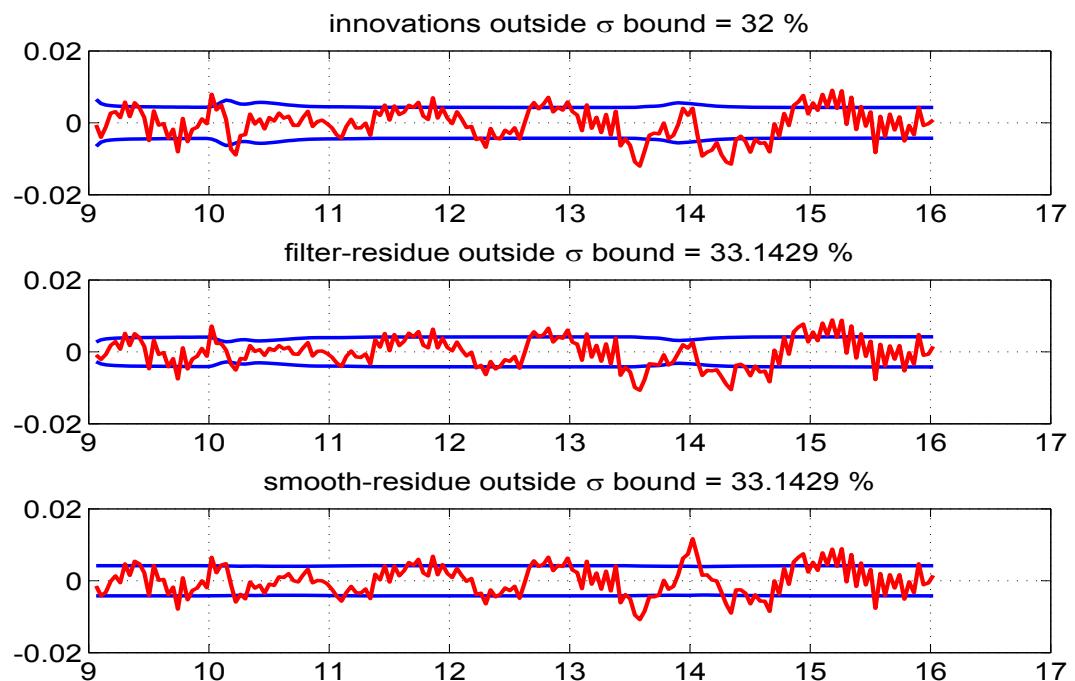
Figure 5.55: The innovations, filtered residue and smoothed residue of measurement 5
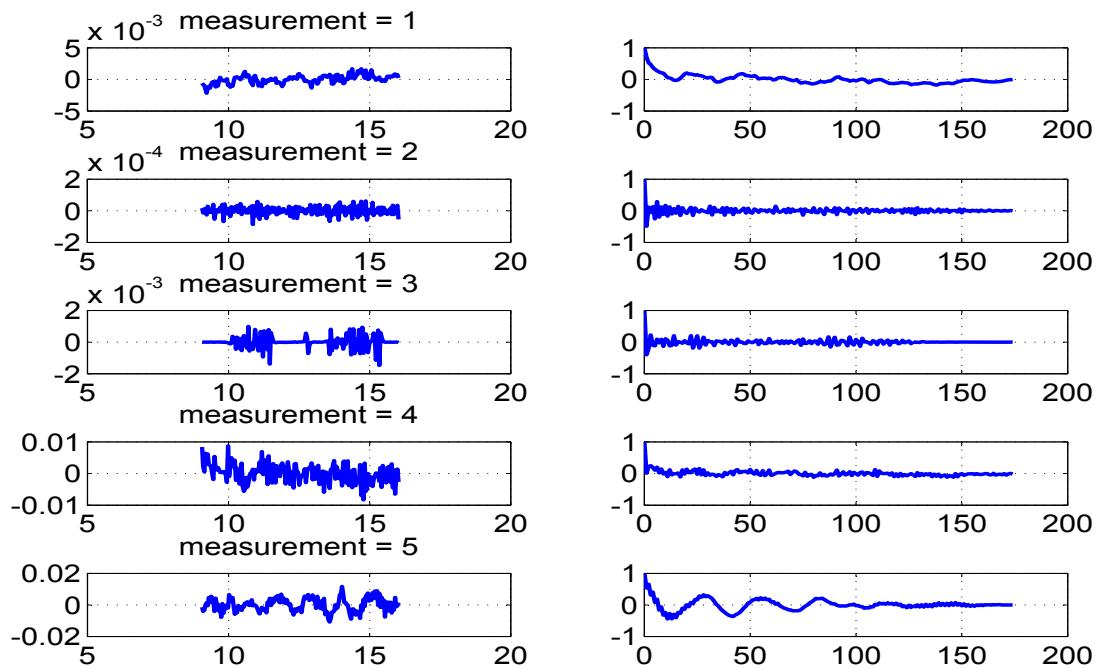
Figure 5.56: Time variation of estimated measurement noise (left) and

their autocorrelation (right)
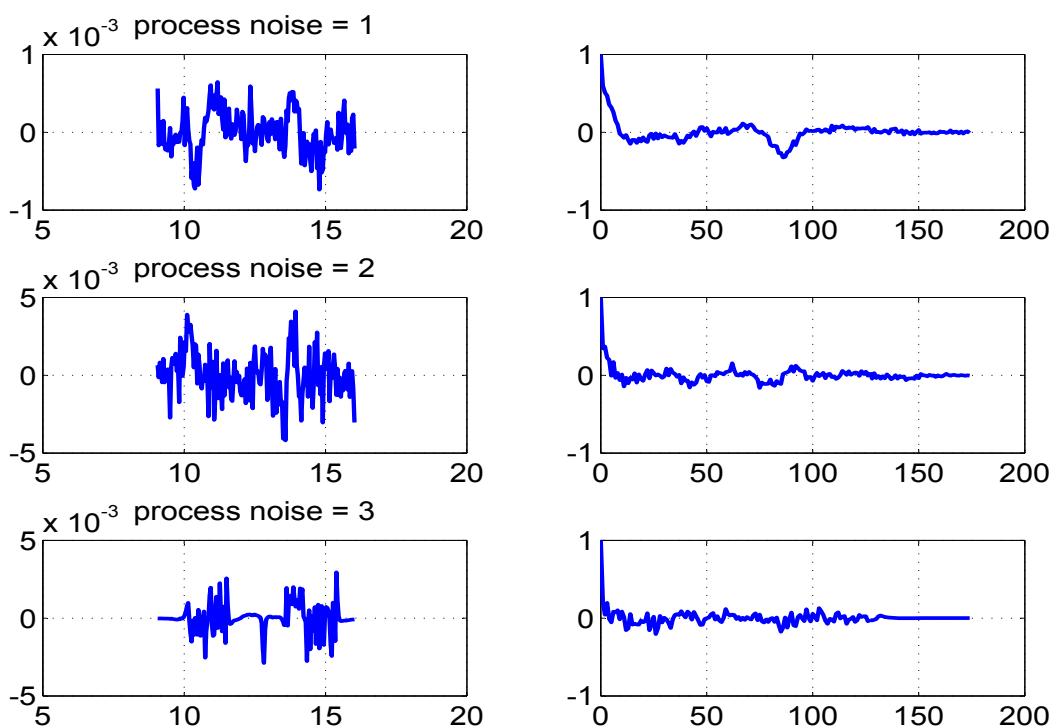
Figure 5.57: Time variation of estimated process noise (left) and

their autocorrelation (right)
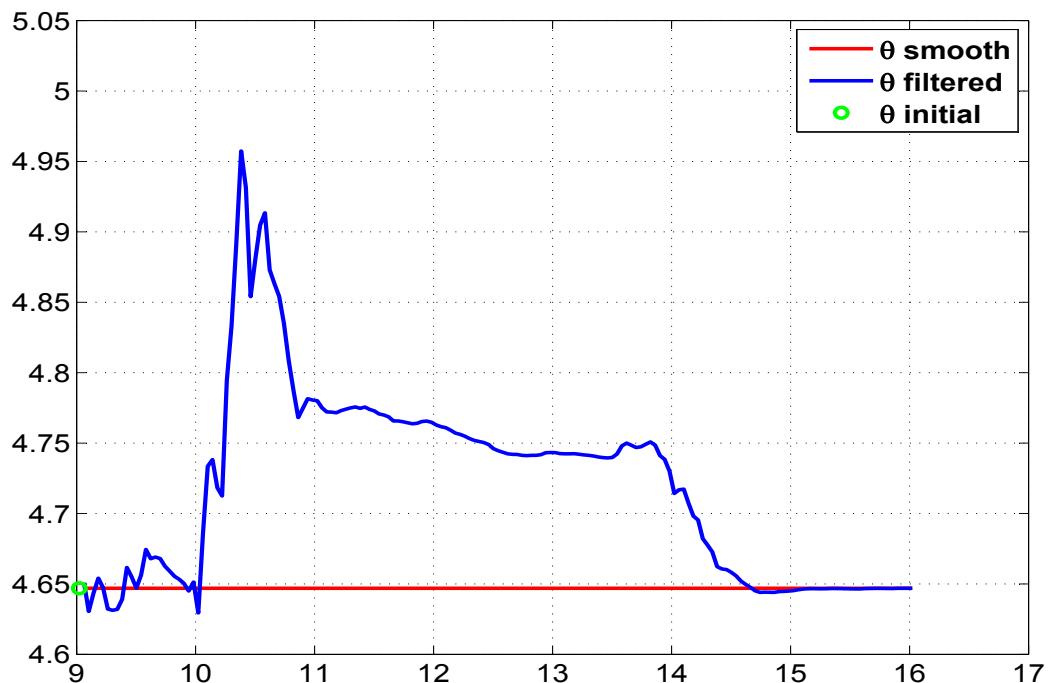
Figure 5.58: Variation of the parameter estimate (C_{N_α}) through time instants
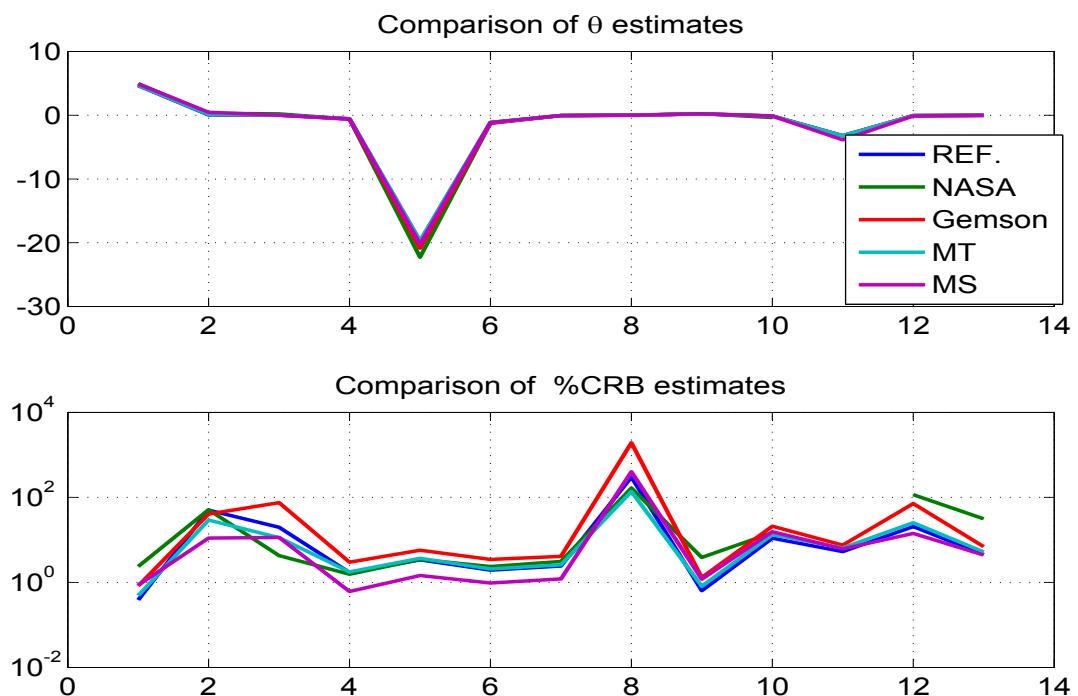
Figure 5.59: Comparison of the parameter estimates and %CRBs by different methods

259

5.3   Real Flight Test Case-3

The data set is obtained from NASA TP 1690 (Maine[46] 1981) by employing a peculiar manoeuvre where elevator control input (δ_e in rad) shown in Fig. 5.60 is imparted when the aircraft (T 37 B) is rolling through a full rotation about its x-axis during aileron roll. Some of the available measurements can also be used as inputs in the state equations which includes roll angle (ϕ_m in rad), sideslip (β_m in rad), velocity (V_m in ft/s), roll rate (p_m in rad/s), yaw rate (r_m in rad/s) and the angle of attack (α_m in rad) are shown in Fig. 5.62, Fig. 5.63, Fig. 5.64, Fig. 5.65, Fig. 5.61 and Fig. 5.69 respectively. The state equations ($n = 3$) for the angle of attack (α), pitch rate (q) and the pitch angle (θ) respectively are

$$\dot{\alpha} = -\frac{\bar{q}S}{mV_m cos(\beta_m)}(C_{L_\alpha}\alpha + C_{L_{\delta_e}}\delta_e + C_{L_0}) + q + \frac{g}{V_m cos(\beta_m)}(cos(\phi_m)cos(\alpha_m)cos(\theta) +$$

$$sin(\alpha_m)sin(\theta)) - tan(\beta_m)(p_m cos(\alpha_m) + r_m sin(\alpha_m))$$

$$\dot{q} = \frac{\bar{q}S\bar{c}}{Iyy}(C_{m_\alpha}\alpha + C_{m_q}\frac{\bar{c}}{2V}q + C_{m_{\dot{\alpha}}}\frac{\bar{c}}{2V}\dot{\alpha} + C_{m_{\delta_e}}\delta_e + C_{m_0}) + \frac{Izz - Ixx}{Iyy}r_m p_m$$

$$\dot{\theta} = qcos(\phi_m) - r_m sin(\phi_m) + \theta_0$$

The angle of attack (α), pitch rate (q), the pitch angle (θ) and the normal acceleration (a_n) are measured (indicated with subscript 'm') in units of rad, rad/s, rad and ft/s^2 respectively. The measurement equations (m=4) are given by

$$\alpha_m = K_\alpha \alpha - K_\alpha x_\alpha \frac{q}{V}$$

$$q_m = q$$

$$\theta_m = \theta$$

$$a_{n_m} = \frac{\bar{q}S}{mg}(C_{N_\alpha}\alpha + C_{N_{\delta_e}}\delta_e + C_{N_0}) + \frac{x_{a_n}}{g}\dot{q}$$

The unknown parameters ($p = 10$) is $(C_{L_\alpha}, C_{L_{\delta_e}}, C_{L_0}, C_{m_\alpha}, C_{m_q}, C_{m_{\dot{\alpha}}}, C_{m_{\delta_e}}, C_{m_0}, \theta_0, C_{N_0})^T$ with an approximation $C_{N_\alpha} = C_{L_\alpha}$ and $C_{N_{\delta_e}} = C_{L_{\delta_e}}$. The ones with suffix '$\delta_e$' are the control derivatives, the ones with suffix zero are the biases and all others are aerodynamic derivatives. The initial states are taken as initial measurement and the initial parameter values are taken as $(4, 0.15, 0.2, -0.5, -11.5, -5, -1.38, -0.06, -0.01, 0.2)^T$.

Case-3 real data is run using the reference EKF ($\mathbf{Q} > 0$) with 100 iterations. The Fig. 5.60-

Other constant values used for case-3

S=184	m=196	Ixx=6892.7	Iyy=3953.2	Izz=10416.4
g=32.2	$\bar{c}$=5.58	$K_\alpha x_\alpha$=-0.0279	x_{a_n}=0.101	$K_\alpha = 1$

5.65 are the inputs used in state equations. The Fig. 5.66 shows the variation of parameter estimates and its initial covariance $\mathbf{P_0}$ with iterations and a similar Fig. 5.67 for $\mathbf{Q}$ and $\mathbf{R}$. The values of **J1-J3** are close to the number of measurements ($m = 4$) with **J6-J8** are close to the number of states ($n = 3$) as shown in Fig. 5.68 and Table-5.8. This means the measurement and state equations are well balanced. The **J5** is the negative log likelihood cost function. The later Fig. 5.69-5.72 compares (i) the state dynamics based on the estimated parameter after the filter pass through the data, (ii) the state after measurement update, (iii) the smoothed state and (iv) the measurement. The Fig. 5.73-5.76 shows the confidence in the innovations, filtered residue and smoothed residue. The estimated measurement and process noise do not appear to have constant statistical characteristics across time as seen in Fig. 5.77 and Fig. 5.78. Another experiment was carried out by generating a typical data set by using the estimated theta and injecting the estimated $\mathbf{Q}$ and $\mathbf{R}$ as additive white Gaussian noise. This is to determine the effect of non additive, non White and non Gaussian noise distribution in the real data on the CRBs. After each iteration in the reference recipe the Θ, $\mathbf{Q}$ and $\mathbf{R}$ were reset as from the real data. Similar experiment was also conducted by updating Θ as well. It turned out that there is not much of a difference in the final CRBs as can be seen from the Table-5.9.

Finally two other filter runs were carried out using the MT and MS statistics for the estimation of $\mathbf{Q}$ and $\mathbf{R}$. The behaviour of the various cost function and in particular **J6** and **J7** in Table-5.10 shows that the choice of the filter statistics for estimating $\mathbf{Q}$ and $\mathbf{R}$ in the proposed reference approach is the best possible when compared to other approaches presently considered.

5.3.1 Remarks on Case-3

The NASA results have been generated assuming $\mathbf{Q} = 0$ and are comparable with reference procedure for the parameter estimates and their CRBs. Further the MT and MS methods give quite different estimates for the $\mathbf{Q}$ and $\mathbf{R}$ values than in the reference case. We believe that the reference procedure provides the best possible parameter estimates and their uncertainties. From the plot of the parameter estimates and their %CRB in Fig

5.79, it can be seen that the parameters 1, 3, 4, 5, 6, 7, 8, 10 are strong and the parameters 2 and 9 are the weak ones. The CRBs as estimated by different methods generally appear to vary widely. However what is interesting is that even the estimate of the strong parameter such as 5 varies widely among the methods. Such a behaviour of the filter across the parameter estimates shows how important is the tuning of the filter statistics namely $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$ in parameter estimation and their uncertainties. It was also observed that for this particular case, the cost $\mathbf{J2}$ using filtered residue went negative at some iteration as seen in Fig. 5.68 with a spike whose absolute value was plotted. However the cost $\mathbf{J3}$ using smoothed residue that was used for tuning the filter did not show any such peculiarity.

The rounded off 100×C matrix for case-3 is given by

$$
\begin{bmatrix}
100 & -44 & 6 & 41 & 10 & 1 & -22 & 16 & 1 & 28 \\
-44 & 100 & -14 & -22 & -34 & -6 & 42 & -41 & 0 & -95 \\
6 & -14 & 100 & 14 & 26 & 1 & -32 & 32 & 0 & -3 \\
41 & -22 & 14 & 100 & 23 & -2 & -53 & 41 & 1 & 14 \\
10 & -34 & 26 & 23 & 100 & 50 & -79 & 73 & 0 & 27 \\
1 & -6 & 1 & -2 & 50 & 100 & -14 & -1 & 0 & 0 \\
-22 & 42 & -32 & -53 & -79 & -14 & 100 & -97 & -1 & -35 \\
16 & -41 & 32 & 41 & 73 & -1 & -97 & 100 & 1 & 36 \\
1 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 100 & 0 \\
28 & -95 & -3 & 14 & 27 & 0 & -35 & 36 & 0 & 100
\end{bmatrix}
$$

5.3.2 Case-3 Tables

Table 5.8: Real flight test data case-3 results using the reference adaptive EKF.
No of iterations=100

Study	Θ (Ref)	Θ (NASA)	Θ (Gemson)	σ_Θ (Ref)	σ_Θ (NASA)	σ_Θ (Gemson)	R $\times 10^{-6}$ (Ref)	Q $\times 10^{-6}$ (Ref)	J1-J8 (Ref)	Remarks
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: EM-[✓,0;0,0] $\mathbf{R}$: EM-diag	4.9235 0.1554 0.2409 -0.5293 -11.8596 -6.8959 -0.9731 -0.0425 0.0003 0.2538	5.1068 0.1909 0.2448 -0.6474 -14.26 -8.27 -1.1614 -0.0505 -0.01177 0.2541	4.9028 0.0879 0.2529 -0.6174 -18.8339 -7.1290 -1.1841 -0.0507 -0.0037 0.2503	0.0164 0.0271 0.0021 0.0079 0.2402 0.4891 0.0177 0.0009 0.0021 0.0014	0.1322 0.1602 0.009215 0.02339 0.6528 1.296 0.05371 0.002655 0.02528 0.008935	0.0168 0.0267 0.0018 0.0211 0.8379 1.544 0.471 0.0024 0.001 0.0014	1.241 0.051 0.460 5.668	0.180 2.954 2.646	3.9336 4.2225 3.6162 0.0008 -44.1347 2.9752 2.9760 2.9070	Cost functions converge to the expected values

Table 5.9: Case-3 results using simulated Additive White Gaussian Noise

Study	σ_Θ (Simulated-without updating Θ)	σ_Θ (Simulated-with updating Θ)	σ_Θ (Ref)	Remarks
Case-3 data generated using simulated measurement and process noise (AWGN) of variance $\mathbf{Q}$ and $\mathbf{R}$ estimated by Reference EKF ($\mathbf{Q} > 0$)				
$\mathbf{P_0}$: Scaled up-[0,0;0,✓] $\mathbf{Q}$: $\mathbf{Q}$ (Ref) $\mathbf{R}$: $\mathbf{R}$ (Ref)	0.0173	0.0172	0.0164	No Significant change in σ_Θ
	0.0277	0.0276	0.0271	
	0.0022	0.0022	0.0021	
	0.0082	0.0082	0.0079	
	0.2441	0.2445	0.2402	
	0.5068	0.5086	0.4891	
	0.0182	0.0182	0.0177	
	0.0009	0.0009	0.0009	
	0.0021	0.0021	0.0021	
	0.0014	0.0014	0.0014	

Table 5.10: Real flight test data case-3 results using the MT and MS method.
No of iterations=100

Study	Θ (MT)	σ_Θ (MT)	R (MT) $\times 10^{-6}$	Q (MT) $\times 10^{-6}$	J1-J8 (MT)	Θ (MS)	σ_Θ (MS)	R (MS) $\times 10^{-6}$	Q (MS) $\times 10^{-6}$	J1-J8 (MS)	Remarks
	4.9260	0.0184				5.0620	0.0323				
	0.1587	0.0302			3.7662	0.3594	0.0508			3.1621	
	0.2408	0.0023			4.5191	0.2517	0.0027			3.1507	
$\mathbf{P_0}$: Scaled up-[0,0;0,✓]	-0.5285	0.0082	1.6135		3.8384	-0.5590	0.0055	3.1599		2.5900	Cost functions are
Q : MT/MS-[✓,0;0,0]	-11.8255	0.2483	0.2395	0.2025	0.0008	-12.5965	0.1400	37.2424	0.00005	0.0007	not close to their
R : MT/MS-diag	-6.8798	0.5062	2.3155	3.1532	-43.7340	-6.6713	0.3021	9.3413	0.0003	-38.0517	expected values in
	-0.9711	0.0184	2.9290	0.6666	4.2266	-1.0247	0.0129	841.5496	0.2386	8.4768	MT and MS method
	-0.0424	0.0009			4.2284	-0.0447	0.0006			8.4655	
	0.0002	0.0011			2.9489	-0.0006	0.0007			3.0215	
	0.2540	0.0016				0.2635	0.0026				

5.3.3 Case-3 Figures
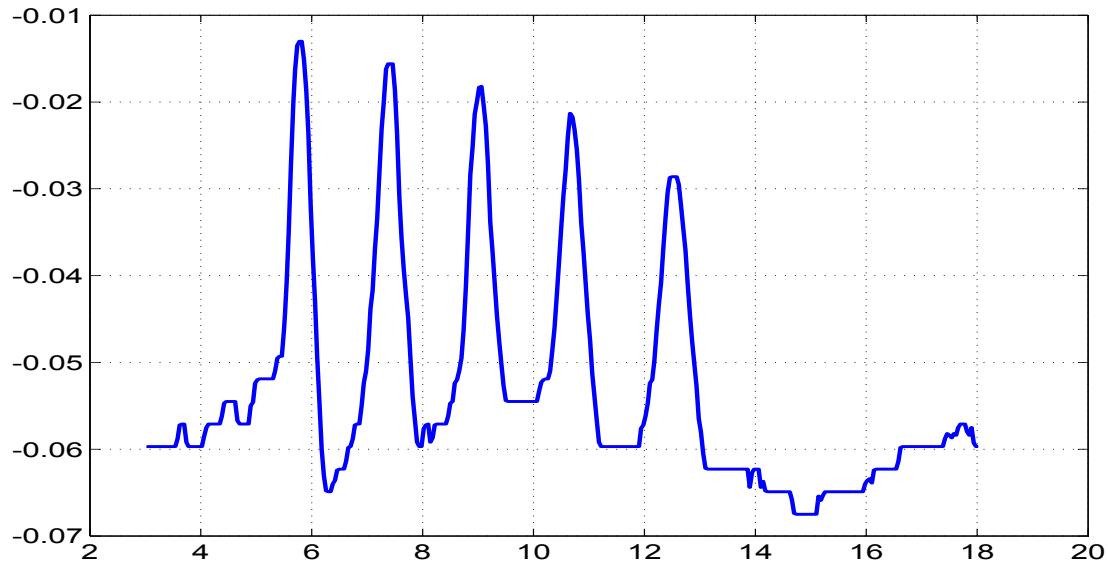
Figure 5.60: Control input (δ_e) versus time (s)
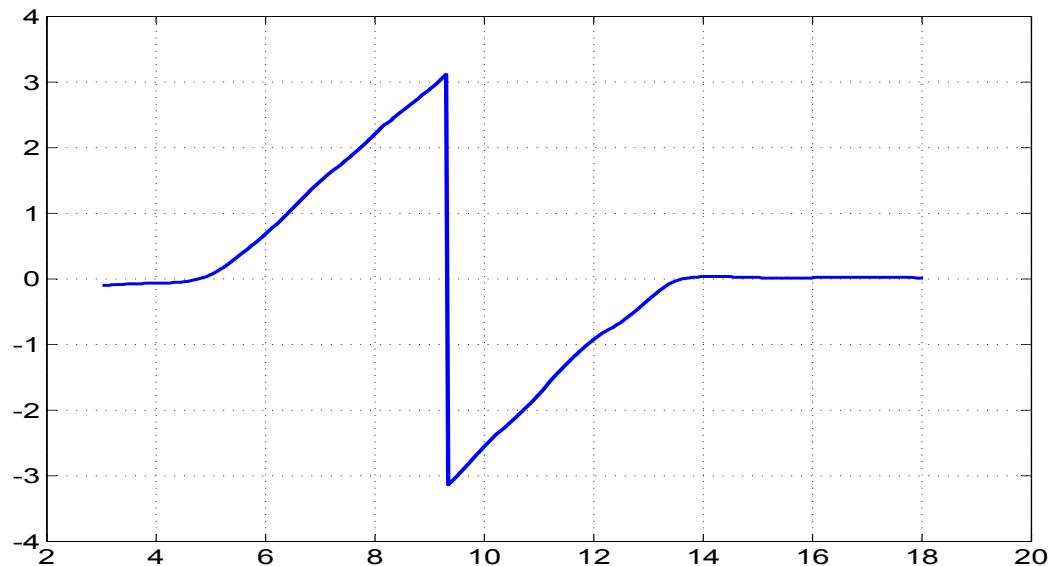
Figure 5.61: Measurement input (ϕ_m) versus time (s)
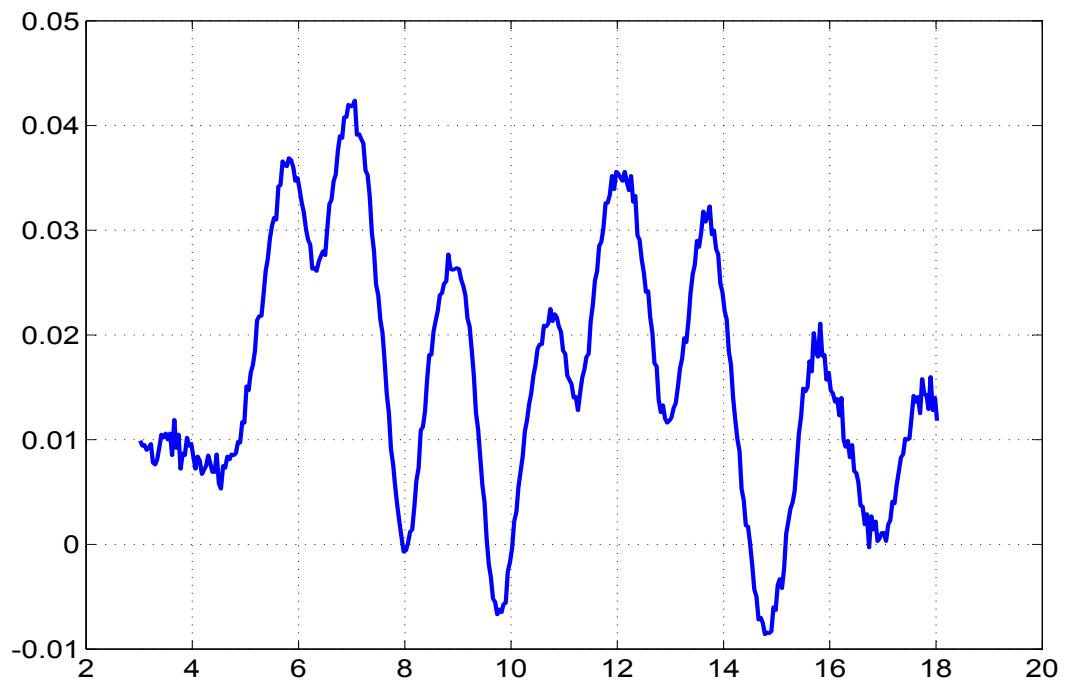
Figure 5.62: Measurement input (β_m) versus time (s)
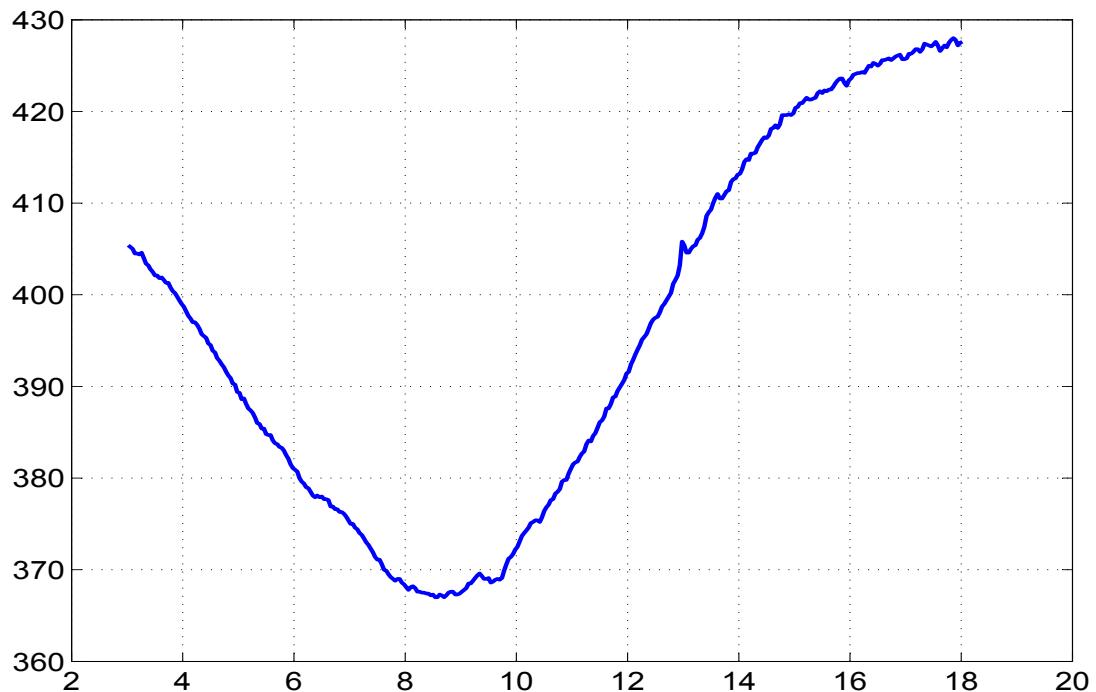
Figure 5.63: Measurement input (V_m) versus time (s)

267
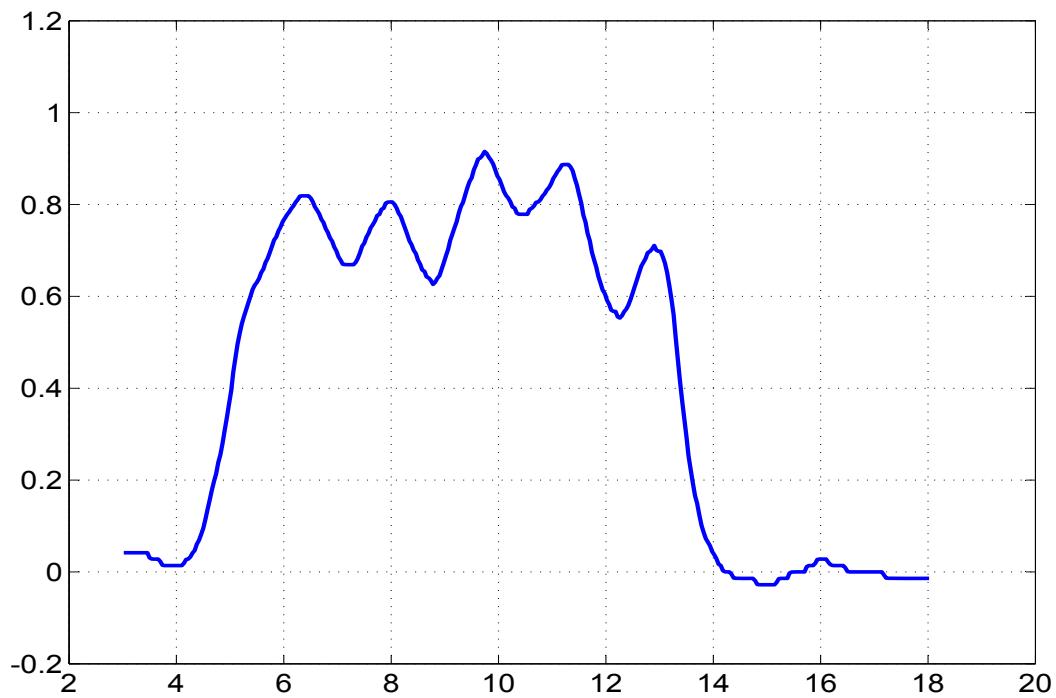
Figure 5.64: Measurement input (p_m) versus time (s)
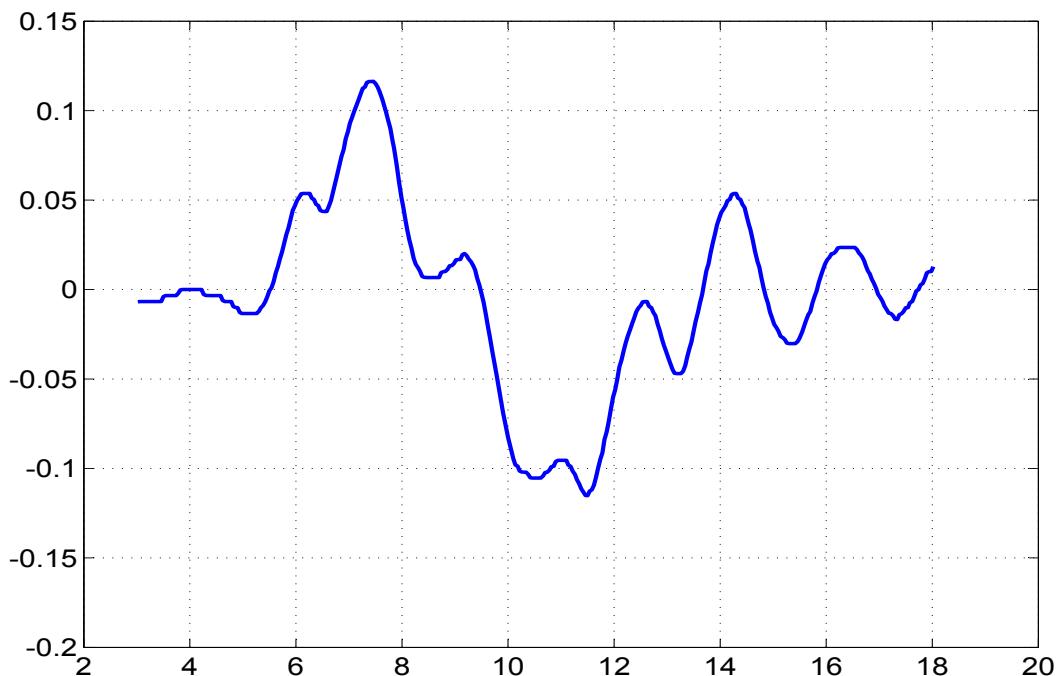
Figure 5.65: Measurement input (r_m) versus time (s)
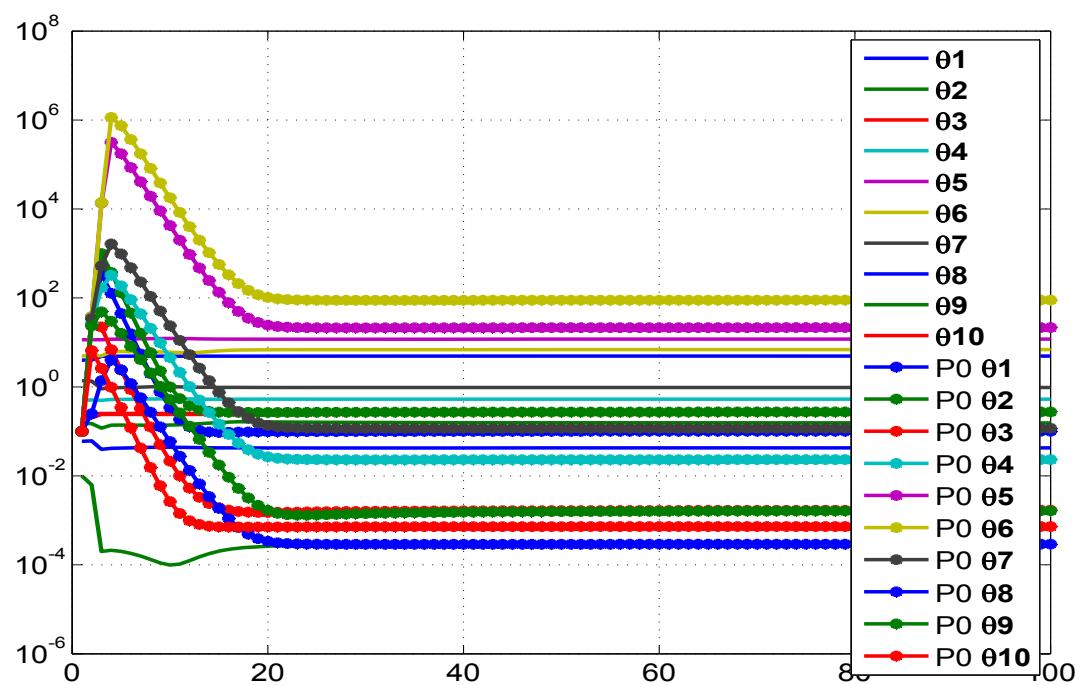
Figure 5.66: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
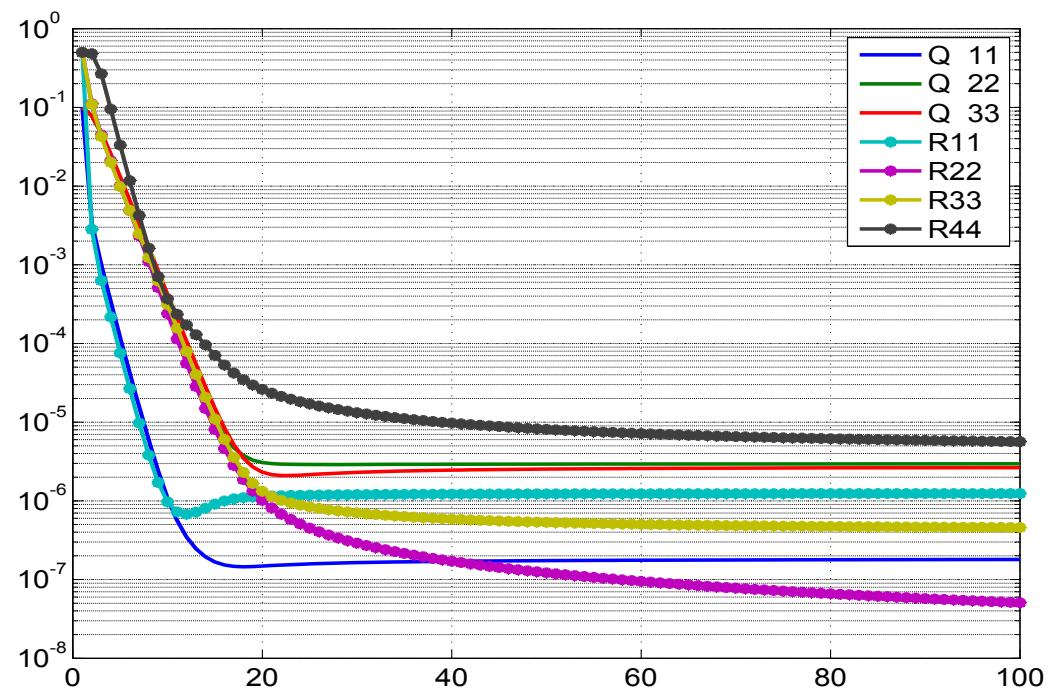
Figure 5.67: Variation of $\mathbf{R}$ with iterations of case 3
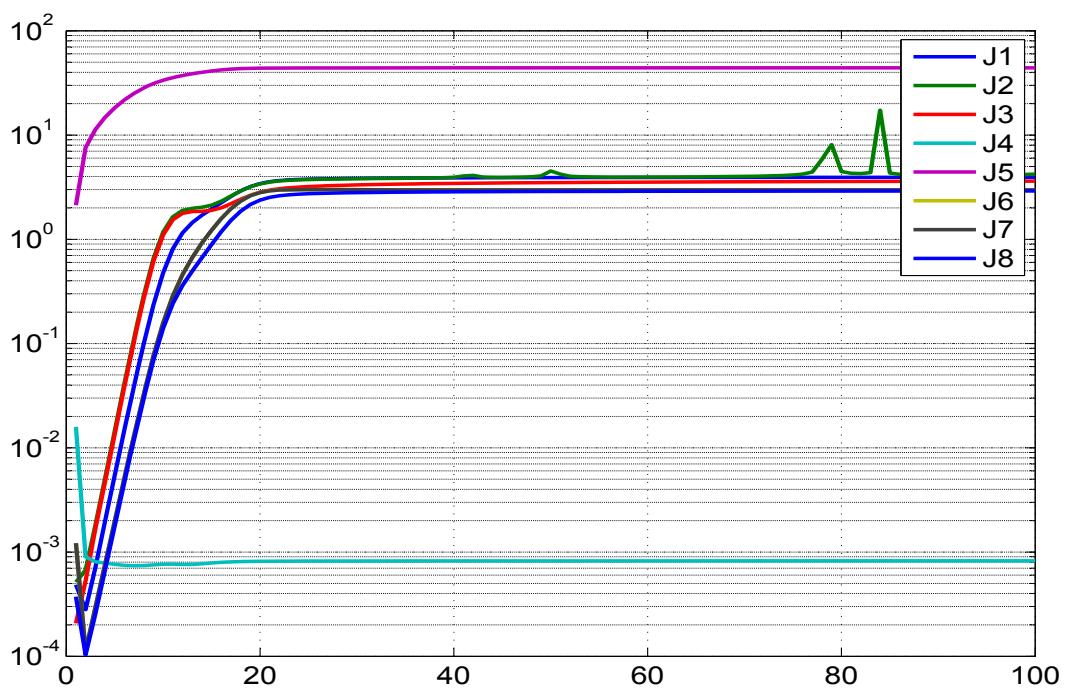
Figure 5.68: Variation of different costs (**J1-J8**) with iterations
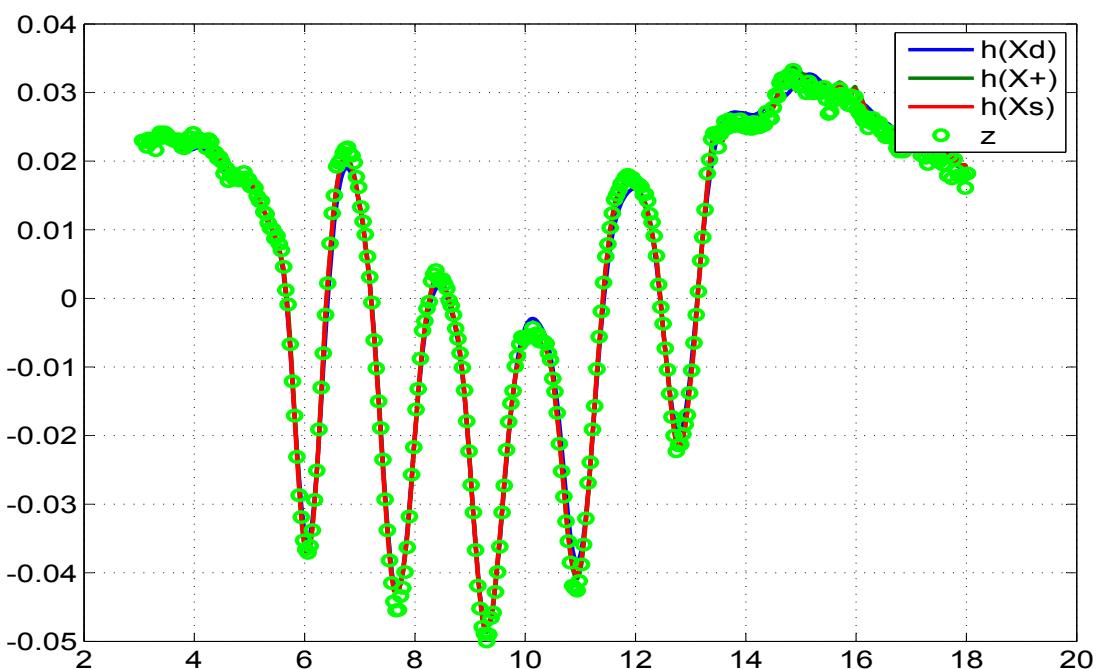
Figure 5.69: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1
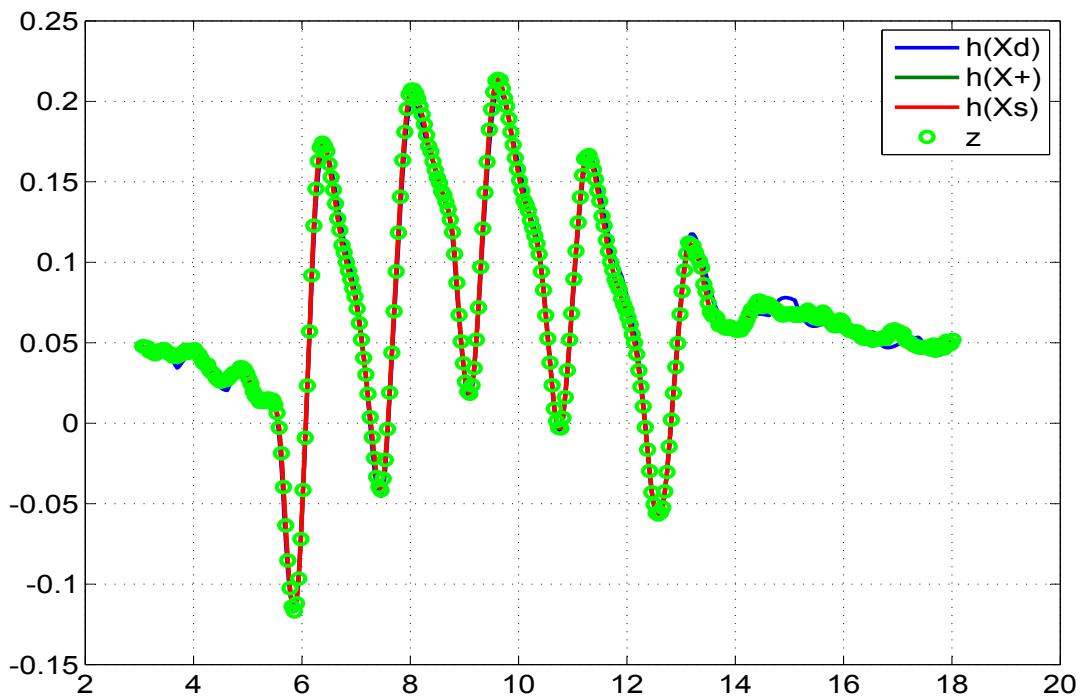
Figure 5.70: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 2
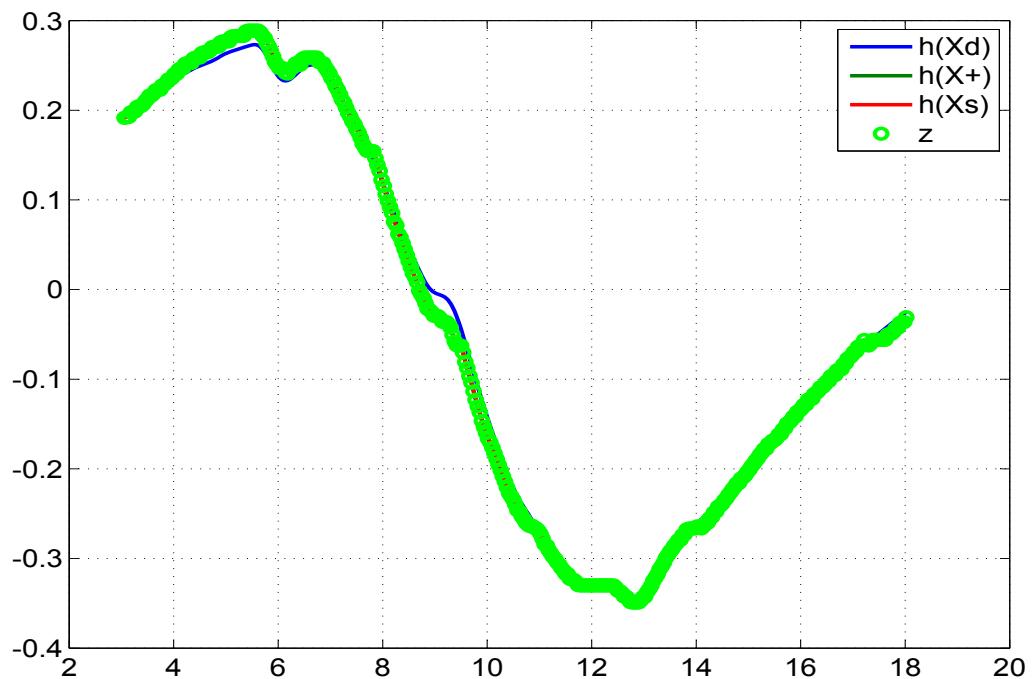
Figure 5.71: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 3
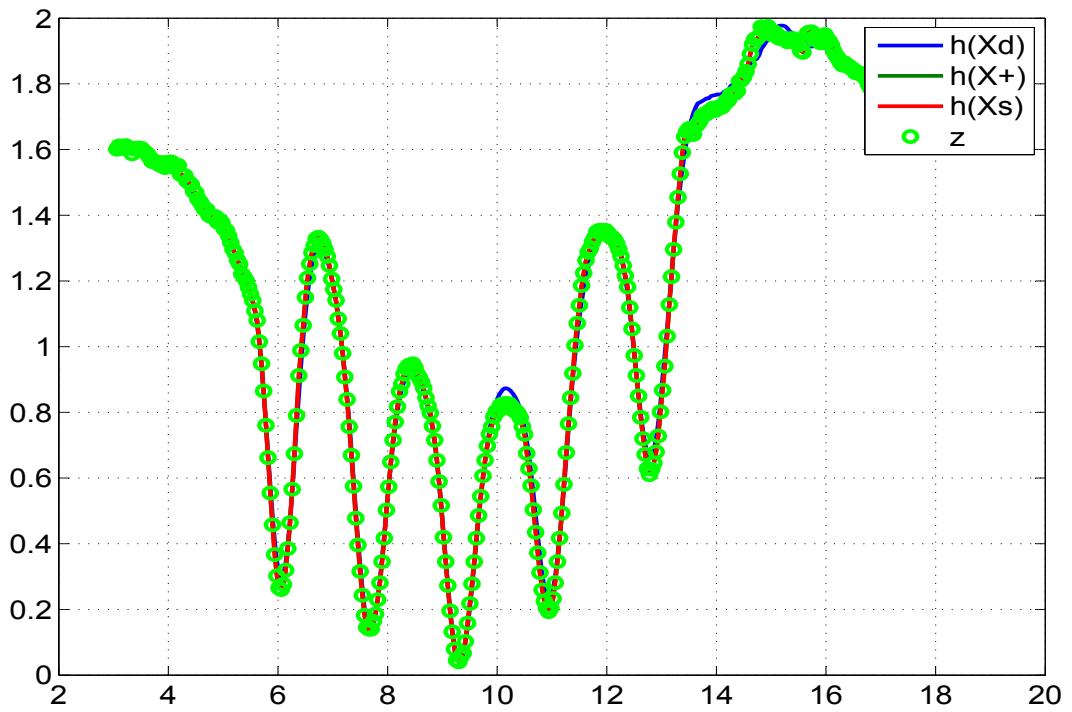
Figure 5.72: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 4
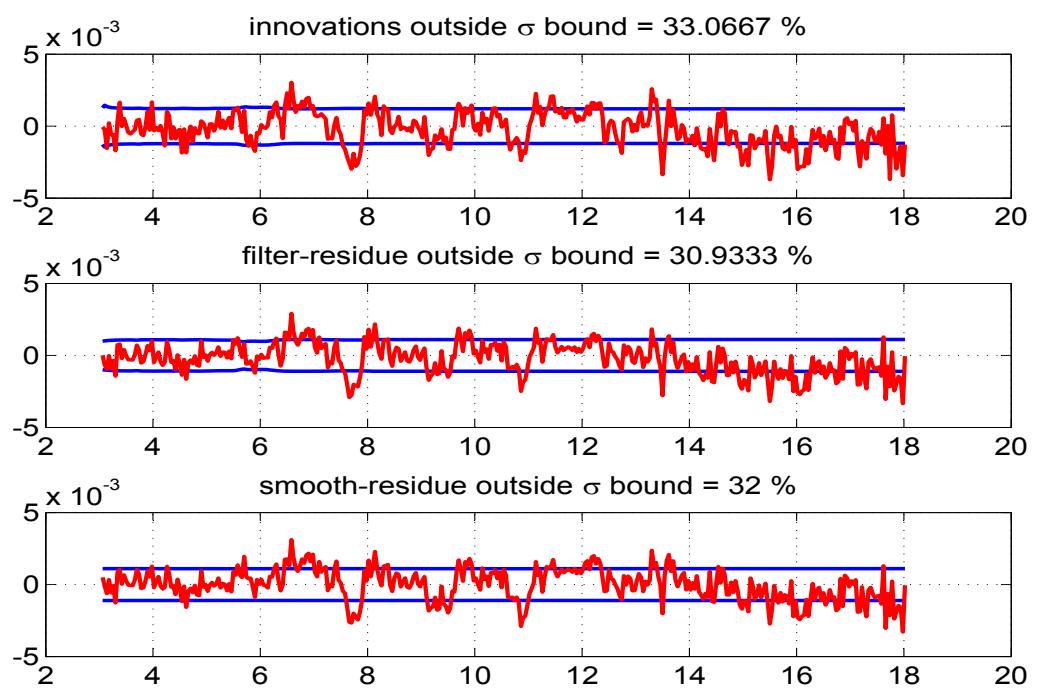
Figure 5.73: The innovations, filtered residue and smoothed residue of measurement 1
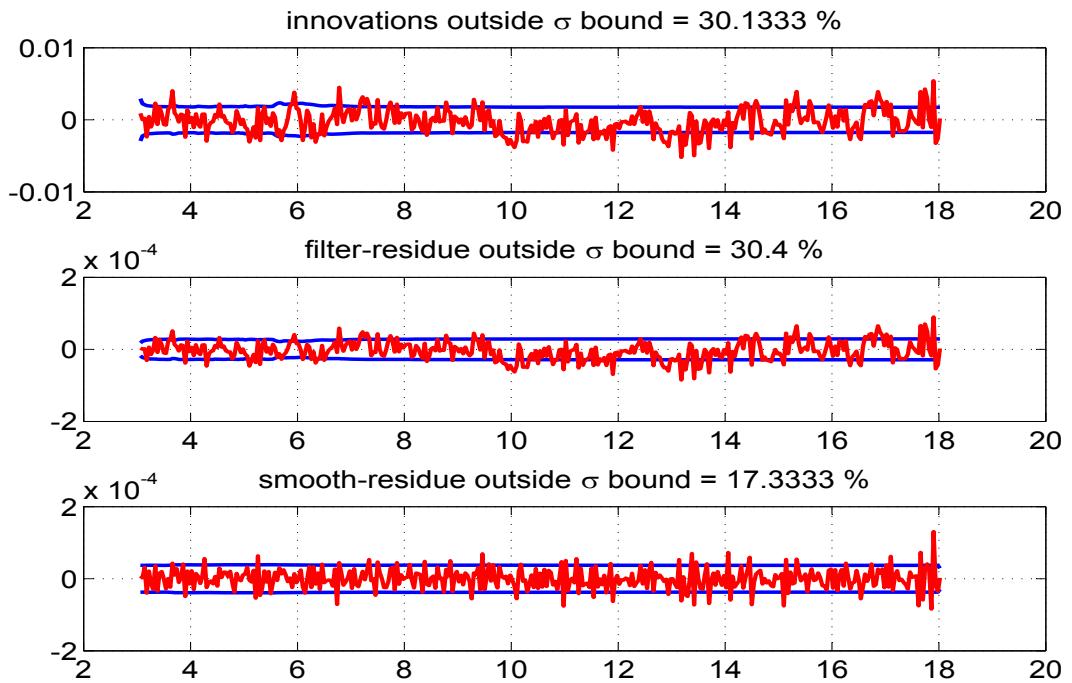
Figure 5.74: The innovations, filtered residue and smoothed residue of measurement 2
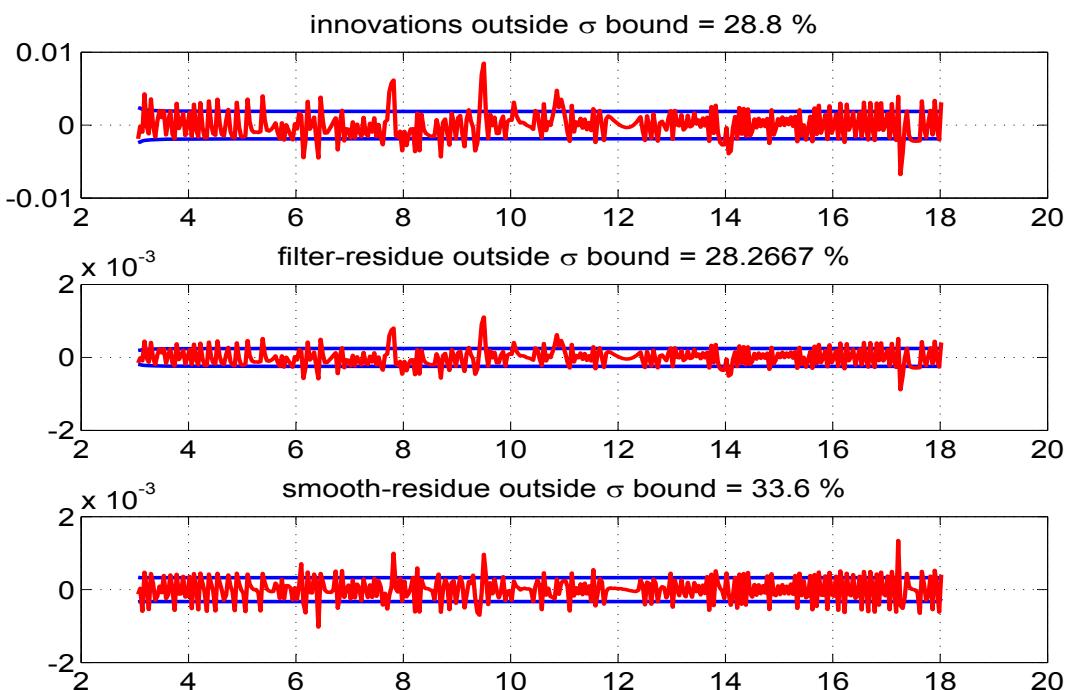
Figure 5.75: The innovations, filtered residue and smoothed residue of measurement 3
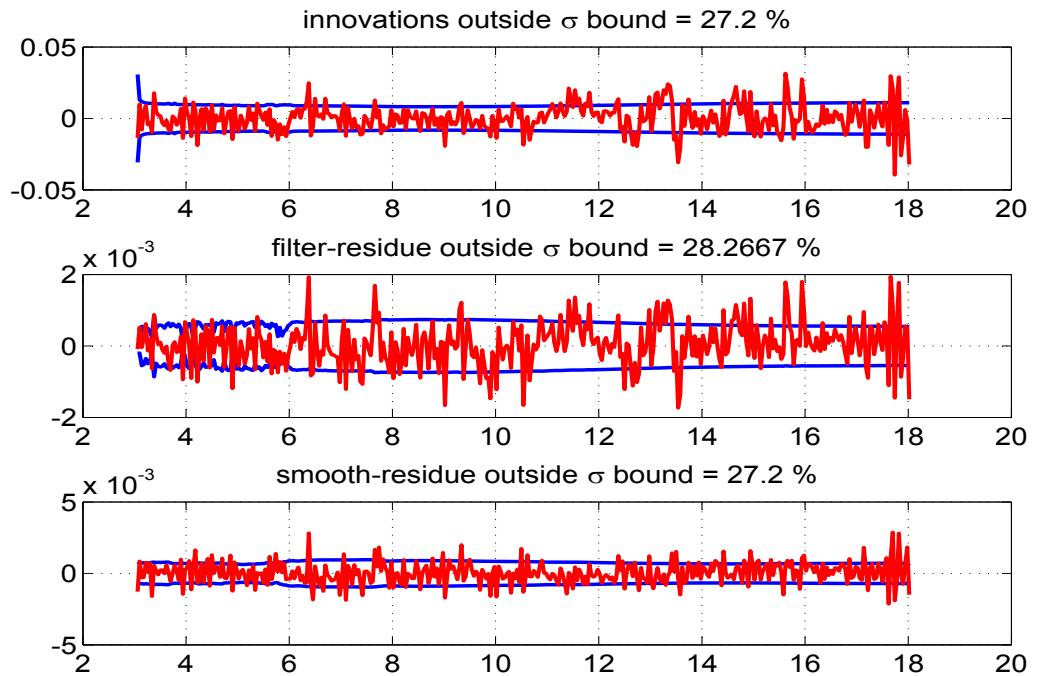
Figure 5.76: The innovations, filtered residue and smoothed residue of measurement 4
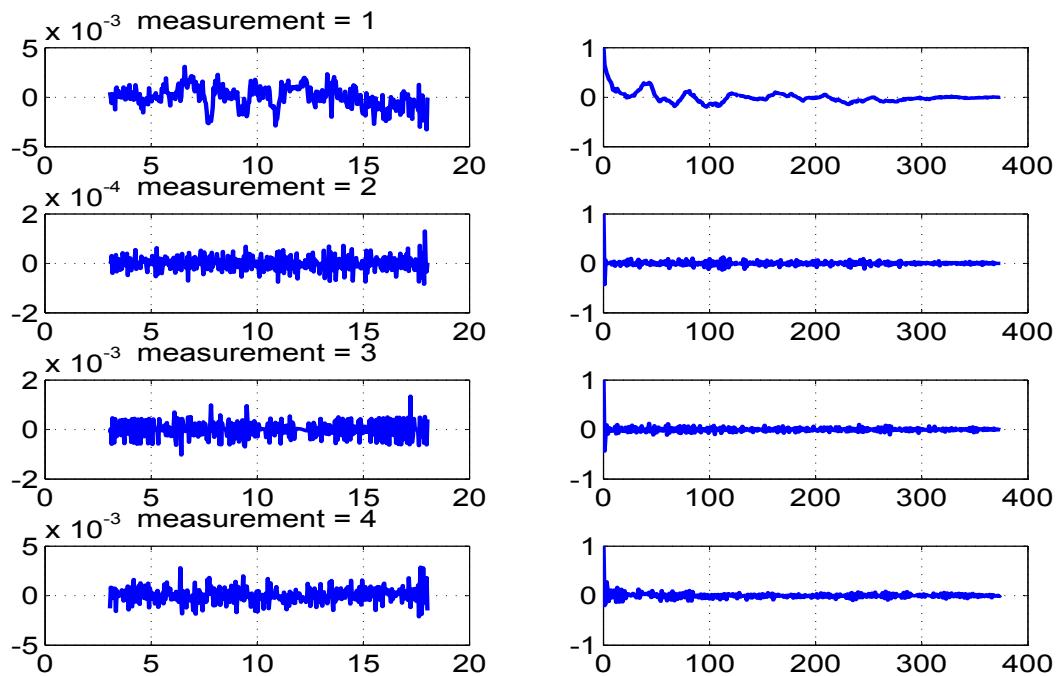
Figure 5.77: Time variation of estimated measurement noise (left) and

their autocorrelation (right)
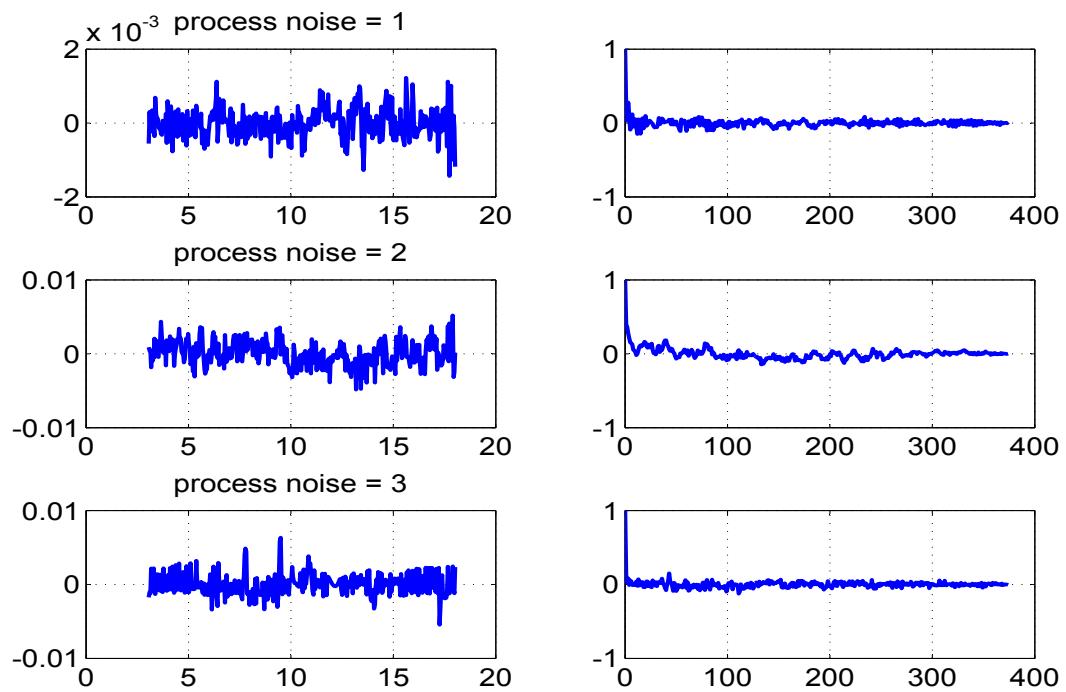
Figure 5.78: Time variation of estimated process noise (left) and

their autocorrelation (right)
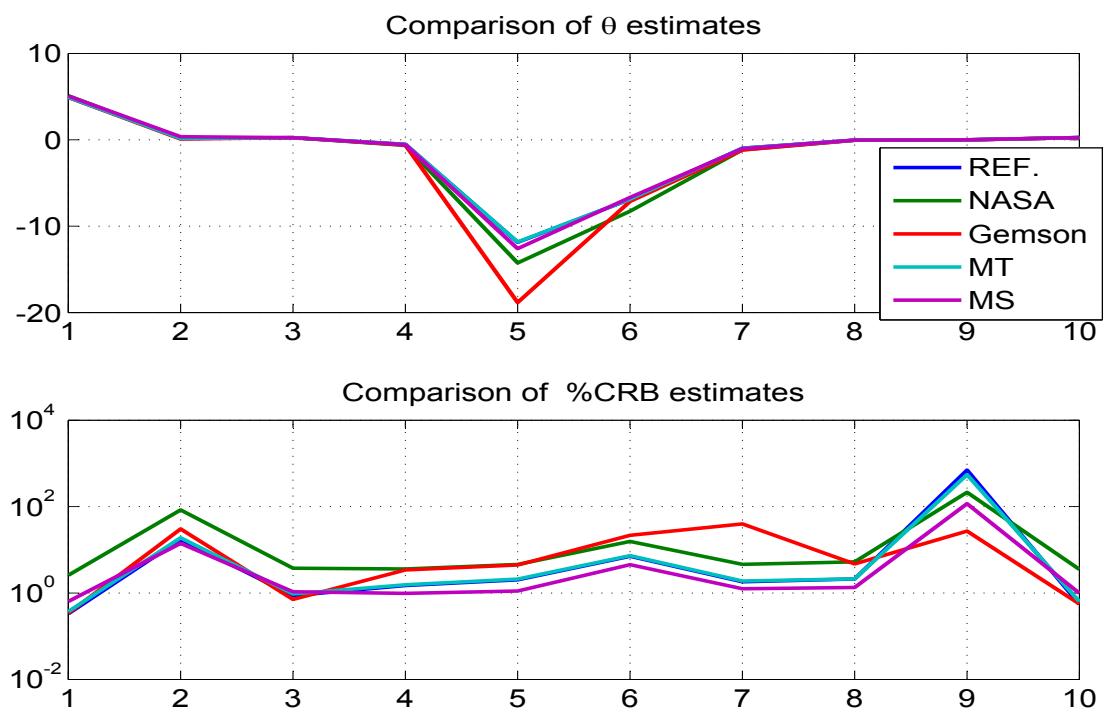
Figure 5.79: Comparison of the parameter estimates and %CRBs by different methods

5.4 Real Flight Test Case-4

The data set is obtained from NASA TN D-7831 (Maine[31] 1975) which pertains to a high speed (4665.0 ft/s) flight in lateral motion excited by a rudder (δ_r in rad) and aileron (δ_a in rad) pulse as shown in Fig. 5.80. The state equations ($n = 4$) for the sideslip (β), roll rate (p), roll angle (ϕ) and yaw rate (r) respectively are

$$\dot{\beta} = Y_\beta \beta + Y_p p + Y_\phi \phi - r + Y_{\delta_a} \delta_a + Y_{\delta_r} \delta_r + Y_0$$

$$\dot{p} = L_\beta \beta + L_p p + L_r r + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r + L_0$$

$$\dot{\phi} = p + \phi_0$$

$$\dot{r} = N_\beta \beta + N_p p + N_r r + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r + N_0$$

The angle of sideslip (β), roll rate (p), roll angle (ϕ), yaw rate (r), lateral acceleration (a_y), roll acceleration ($\dot{p}$) and yaw acceleration ($\dot{r}$) are measured (indicated with subscript 'm') in units of rad, rad/s, rad, rad/s, ft/s^2, rad/s^2 and rad/s^2 respectively. The measurement equations (m=7) are given by

$$\beta_m = \beta$$

$$p_m = p$$

$$\phi_m = \phi$$

$$r_m = r$$

$$a_{y_m} = \frac{U_0}{g}(Y_\beta \beta + Y_{\delta_a} \delta_a + Y_{\delta_r} \delta_r + Y_0) + a_{y_b}$$

$$\dot{p}_m = L_\beta \beta + L_p p + L_r r + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r + p_b$$

$$\dot{r}_m = N_\beta \beta + N_p p + N_r r + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r + r_b$$

The unknown parameter set ($p = 21$) is $\theta = (Y_\beta, Y_p, Y_{\delta_a}, Y_{\delta_r}, Y_0, L_\beta, L_p, L_r, L_{\delta_a}, L_{\delta_r}, L_0,$ $\phi_0, N_\beta, N_p, N_r, N_{\delta_a}, N_{\delta_r}, N_0, a_{y_b}, p_b, r_b)^T$. The ones with suffix '$\delta_a$' or '$\delta_r$' are the control derivatives, the ones with suffix zero or b are the biases and all others are aerodynamic derivatives. The initial states are taken as initial measurement and the initial parameter values are taken as $(-0.04, 0.01, 0.0001, 0.01, 0.0001, -16.8, -0.24, 0.4, 12, 20, 0.0001,$ $0.0001, 1.5, -0.003, -0.04, 0.35, -2.5, 0.0001, 0.2, 0.01, 0.001)^T$.

Other constant values used for case-4

$Y_\phi = 0.0069$	$U_0 = 4665$	g=32.2

Case-4 real data is run using the reference EKF ($\mathbf{Q} > 0$) with 100 iterations. The Fig. 5.80 is the control input. The Fig. 5.81 shows the variation of parameter estimates and its initial covariance $\mathbf{P_0}$ with iterations and a similar Fig. 5.82 for $\mathbf{Q}$ and $\mathbf{R}$. The values of **J1-J3** are close to the number of measurements ($m = 7$) with **J6-J8** are close to the number of states ($n = 4$) as shown in Fig. 5.83 and Table-5.11. This means the measurement and state equations are well balanced. The **J5** is the negative log likelihood cost function. The later Fig. 5.84-5.90 compares (i) the state dynamics based on the estimated parameter after the filter pass through the data, (ii) the state after measurement update, (iii) the smoothed state and (iv) the measurement. The Fig. 5.91-5.97 shows the confidence in the innovations, filtered residue and smoothed residue. The estimated measurement and process noise do not appear to have constant statistical characteristics across time as seen in Fig. 5.98 and Fig. 5.99. Another experiment was carried out by generating a typical data set by using the estimated theta and injecting the estimated $\mathbf{Q}$ and $\mathbf{R}$ as additive white Gaussian noise. This is to determine the effect of non additive, non White and non Gaussian noise distribution in the real data on the CRBs. After each iteration in the reference recipe the Θ, $\mathbf{Q}$ and $\mathbf{R}$ were reset as from the real data. Similar experiment was also conducted by updating Θ as well. It turned out that there is not much of a difference in the final CRBs as can be seen from the Table-5.12.

Finally two other filter runs were carried out using the MT and MS statistics for the estimation of $\mathbf{Q}$ and $\mathbf{R}$. The behaviour of the various cost function and in particular **J6** and **J7** in Table-5.13 shows that the choice of the filter statistics for estimating $\mathbf{Q}$ and $\mathbf{R}$ in the proposed reference approach is the best possible when compared to other approaches presently considered.

5.4.1 Remarks on Case-4

The NASA results have been generated assuming $\mathbf{Q} = 0$ and are comparable with reference procedure for the parameter estimates and their CRBs. Further the MT and MS methods give quite different estimates for the $\mathbf{Q}$ and $\mathbf{R}$ values than in the reference case. We believe that the reference procedure provides the best possible parameter estimates and their uncertainties. From the plot of the parameter estimates and their %CRB in Fig 5.100, it can be seen that the parameters 1, 2, 4, 6, 9-11, 13, 17, 20 and 21 are strong and the parameters 3, 8, 12, 14-16 and 19 are the weak ones and others can be considered as modestly controlling the dynamics of the system. The CRBs as estimated by different methods generally appear to vary widely. However what is interesting is that even the estimate of the strong parameters such as 6, 9 and 10 varies widely among the methods. Such a behaviour of the filter across the parameter estimates shows how important is the tuning of the filter statistics namely $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$ in parameter estimation and their uncertainties.

The rounded off 100×C matrix for case-4 is given by

$$
\begin{bmatrix}
100 & -9 & -24 & 36 & 69 & -77 & 21 & 31 & 17 & -29 & -9 & 0 & 29 & -7 & -12 & -7 & 12 & 7 & 69 & -53 & 19 \\
-9 & 100 & 3 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-24 & 3 & 100 & -58 & -16 & 16 & 0 & -44 & -75 & 43 & 2 & 0 & -7 & -1 & 18 & 30 & -18 & -1 & -16 & 9 & -4 \\
36 & 1 & -58 & 100 & 52 & -27 & -5 & 31 & 42 & -73 & -7 & 0 & 11 & 3 & -12 & -17 & 29 & 5 & 52 & -39 & 15 \\
69 & -1 & -16 & 52 & 100 & -53 & 23 & 44 & 10 & -42 & -14 & 0 & 20 & -7 & -16 & -4 & 16 & 10 & 100 & -77 & 27 \\
-77 & -1 & 16 & -27 & -53 & 100 & -27 & -40 & -23 & 37 & 12 & 0 & 3 & -1 & -1 & -1 & 1 & 1 & -53 & 68 & 2 \\
21 & 0 & 0 & -5 & 23 & -27 & 100 & -3 & 0 & 7 & -5 & 0 & -1 & 3 & 0 & 0 & 0 & 0 & 23 & -27 & -1 \\
31 & 0 & -44 & 31 & 44 & -40 & -3 & 100 & 58 & -42 & -10 & 0 & -1 & 0 & 3 & 2 & -1 & -1 & 44 & -55 & -2 \\
17 & 0 & -75 & 42 & 10 & -23 & 0 & 58 & 100 & -57 & -2 & 0 & -1 & 0 & 2 & 3 & -2 & 0 & 10 & -13 & 0 \\
-29 & 0 & 43 & -73 & -42 & 37 & 7 & -42 & -57 & 100 & 9 & 0 & 1 & 0 & -1 & -2 & 3 & 1 & -42 & 53 & 2 \\
-9 & 0 & 2 & -7 & -14 & 12 & -5 & -10 & -2 & 9 & 100 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & -13 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
29 & 0 & -7 & 11 & 20 & 3 & -1 & -1 & -1 & 1 & 0 & 0 & 100 & -27 & -40 & -22 & 37 & 24 & 20 & 2 & 65 \\
-7 & 0 & -1 & 3 & -7 & -1 & 3 & 0 & 0 & 0 & 0 & 0 & -27 & 100 & -3 & -1 & 7 & -10 & -7 & -1 & -26 \\
-12 & 0 & 18 & -12 & -16 & -1 & 0 & 3 & 2 & -1 & 0 & 0 & -40 & -3 & 100 & 58 & -42 & -20 & -16 & -2 & -52 \\
-7 & 0 & 30 & -17 & -4 & -1 & 0 & 2 & 3 & -2 & 0 & 0 & -22 & -1 & 58 & 100 & -57 & -5 & -4 & 0 & -13 \\
12 & 0 & -18 & 29 & 16 & 1 & 0 & -1 & -2 & 3 & 0 & 0 & 37 & 7 & -42 & -57 & 100 & 19 & 16 & 2 & 50 \\
7 & 0 & -1 & 5 & 10 & 1 & 0 & -1 & 0 & 1 & 1 & 0 & 24 & -10 & -20 & -5 & 19 & 100 & 10 & 1 & 0 \\
69 & 0 & -16 & 52 & 100 & -53 & 23 & 44 & 10 & -42 & -13 & 0 & 20 & -7 & -16 & -4 & 16 & 10 & 100 & -77 & 27 \\
-53 & 0 & 9 & -39 & -77 & 68 & -27 & -55 & -13 & 53 & -1 & 0 & 2 & -1 & -2 & 0 & 2 & 1 & -77 & 100 & 3 \\
19 & 0 & -4 & 15 & 27 & 2 & -1 & -2 & 0 & 2 & 0 & 0 & 65 & -26 & -52 & -13 & 50 & 0 & 27 & 3 & 100
\end{bmatrix}
$$

5.4.2 Case-4 Tables

Table 5.11: Real flight test data case-4 results using the reference adaptive EKF
No of iterations=100

Study	Θ (Ref)	Θ (NASA)	Θ (Gemson)	σ_Θ (Ref)	σ_Θ (NASA)	σ_Θ (Gemson)	$\mathbf{R}$ $\times 10^{-6}$ (Ref)	$\mathbf{Q}$ $\times 10^{-6}$ (Ref)	J1-J8 (Ref)	Remarks
	-0.0465	-0.04670	-0.04590	0.0003	0.0005401	0.0003				
	0.1017	0.1026	0.10430	0.0036	0.0008	0.0018				
	-0.0018	0.002753	-0.0013	0.0005	0.0009128	0.0006				
	0.0125	0.01594	0.0121	0.0006	0.001019	0.0007				
	-0.0028	-0.003035	-0.0028	0.0010	0.0002101	0.0005				
	-22.8656	-24.32	-22.5872	0.1902	0.1500	0.1672				
	-0.2335	-0.1505	-0.1346	0.0084	0.009149	0.0127			6.6805	
	1.7855	2.4640	0.7565	0.1554	0.2088	0.2344	4.6243		7.0957	
	11.6289	14.47	12.7465	0.3795	0.3688	0.4221	0.2728	0.14	6.6025	
$\mathbf{P_0}$: Scaled up-[0,0;0,✓]	15.6179	17.87	15.1983	0.3697	0.3906	0.4004	0.2611	17.23	0.0324	
$\mathbf{Q}$: EM-[✓,0;0,0]	0.3920	0.4092	0.3859	0.0125	0.007108	0.0119	0.0086	5.12	-72.6142	Cost functions converge
$\mathbf{R}$: EM-diag	-0.0029	-0.008423	-0.0019	0.0059	–	0.0032	6.3789	0.10	4.0024	to the expected values.
	1.3150	1.2900	1.2708	0.0172	0.01056	0.0131	930.0747		4.0024	
	0.0137	0.0004483	0.0090	0.0012	0.0007281	0.0010	21.7747		3.5376	
	-0.0698	-0.1514	0.0299	0.0219	0.01637	0.0194				
	0.5326	0.5062	0.4043	0.0428	0.3197	0.0355				
	-1.9137	-2.125	-1.9494	0.0412	0.03726	0.0343				
	-0.0079	-0.007553	-0.006	0.0010	0.0005759	0.0008				
	0.4982	0.5179	0.4908	0.1466	–	0.0739				
	0.3703	-0.02581	0.3617	0.0065	–	0.0072				
	-0.0128	-0.004472	-0.0110	0.0007	–	0.0006				

Table 5.12: Case-4 results using simulated Additive White Gaussian Noise

Study	σ_Θ (Simulated-without updating Θ)	σ_Θ (Simulated-with updating Θ)	σ_Θ (Ref)	Remarks
Case-4 data generated using simulated measurement and process noise (AWGN) of variance $\mathbf{Q}$ and $\mathbf{R}$ estimated by Reference EKF ($\mathbf{Q} > 0$)				
$\mathbf{P_0}$: Scaled up-$[0,0;0,\checkmark]$ $\mathbf{Q}$: $\mathbf{Q}$ (Ref) $\mathbf{R}$: $\mathbf{R}$ (Ref)	0.0004	0.0004	0.0003	No Significant change in σ_Θ
	0.0043	0.0043	0.0036	
	0.0006	0.0006	0.0005	
	0.0006	0.0006	0.0006	
	0.0011	0.0011	0.0010	
	0.2225	0.2370	0.1902	
	0.0108	0.0108	0.0084	
	0.1642	0.1644	0.1554	
	0.3668	0.3724	0.3795	
	0.3775	0.3813	0.3697	
	0.0130	0.0132	0.0125	
	0.0059	0.0059	0.0059	
	0.0204	0.0214	0.0172	
	0.0015	0.0015	0.0012	
	0.0232	0.0232	0.0219	
	0.0395	0.0398	0.0428	
	0.0419	0.0420	0.0412	
	0.0011	0.0011	0.0010	
	0.1550	0.1550	0.1466	
	0.0074	0.0078	0.0065	
	0.0008	0.0009	0.0007	

Table 5.13: Real flight test data case-4 results using the MT and MS method.
No of iterations=100

Study	Θ (MT)	σ_Θ (MT)	R (MT) $\times 10^{-6}$	Q (MT) $\times 10^{-6}$	J1-J8 (MT)	Θ (MS)	σ_Θ (MS)	R (MS) $\times 10^{-6}$	Q (MS) $\times 10^{-6}$	J1-J8 (MS)	Remarks
P₀ : Scaled up-[0,0;0,✓] **Q** : MT/MS-[✓,0;0,0] **R** : MT/MSMS-diag	-0.0460	0.0004				-0.0461	0.0005				Cost functions are not close to their expected values in MT and MS method
	0.1040	0.0027				0.1051	0.0009				
	-0.0007	0.0008				-0.0012	0.0010				
	0.0130	0.0010				0.0126	0.0011				
	-0.0026	0.0008				-0.0026	0.0002				
	-23.2819	0.1526				-24.2547	0.1558				
	-0.1308	0.0118			7.8309	-0.1537	0.0126			7.1837	
	0.7872	0.2170	0.00002		7.9831	1.2883	0.2398	0.0005		7.2237	
	13.2291	0.3960	0.00310		5.5817	13.7752	0.4337	0.2160		5.0069	
	16.1444	0.3826	0.09153	0.0802	0.0187	17.3261	0.4870	0.0026	0.0056	0.0092	
	0.3926	0.0077	0.00008	3.4875	-68.6021	0.4087	0.0079	0.0005	0.8155	-64.7950	
	-0.0020	0.0003	0.35034	0.0048	36.8377	-0.0031	0.0040	0.4559	2.1818	58.5572	
	1.2867	0.0144	2.85722	0.0495	36.8379	1.2826	0.0129	7.0779	0.0194	58.5577	
	0.0074	0.0011	0.02140		5.1064	0.0074	0.0010	0.0212		4.2938	
	0.0006	0.0204				-0.0058	0.0186				
	0.4469	0.0374				0.4704	0.0335				
	-1.9271	0.0360				-1.9647	0.0333				
	-0.0070	0.0008				-0.0067	0.0006				
	0.4652	0.1118				0.4598	0.0348				
	0.3710	0.0068				0.3864	0.0091				
	-0.0119	0.0006				-0.0115	0.0006				

5.4.3 Case-4 Figures
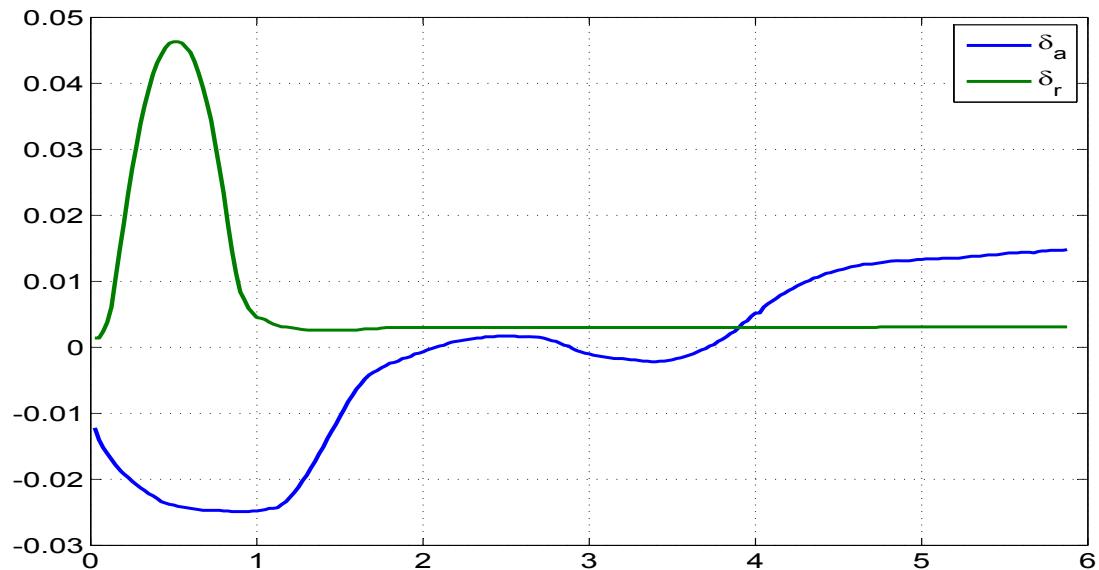
Figure 5.80: Control input versus time (s)
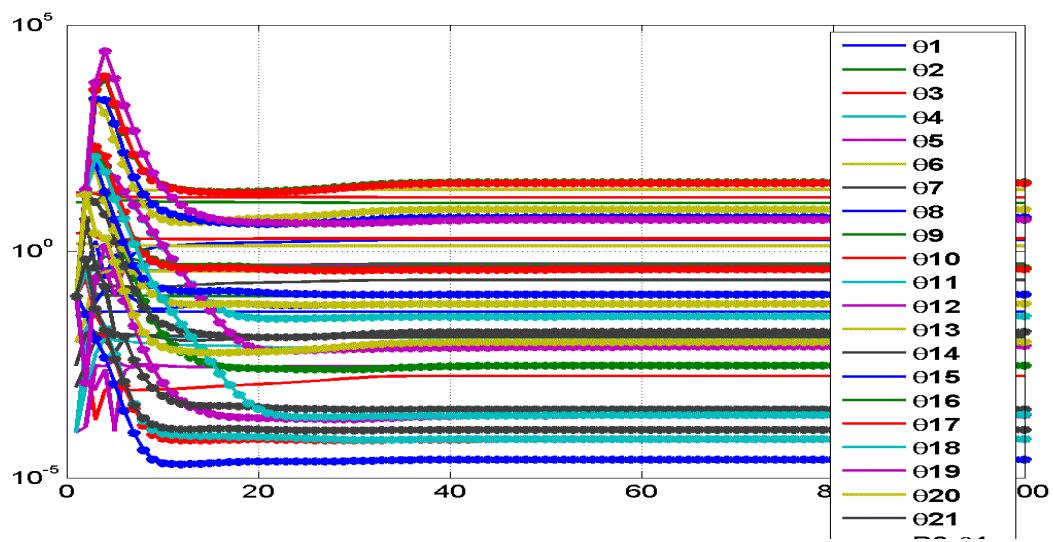
Figure 5.81: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
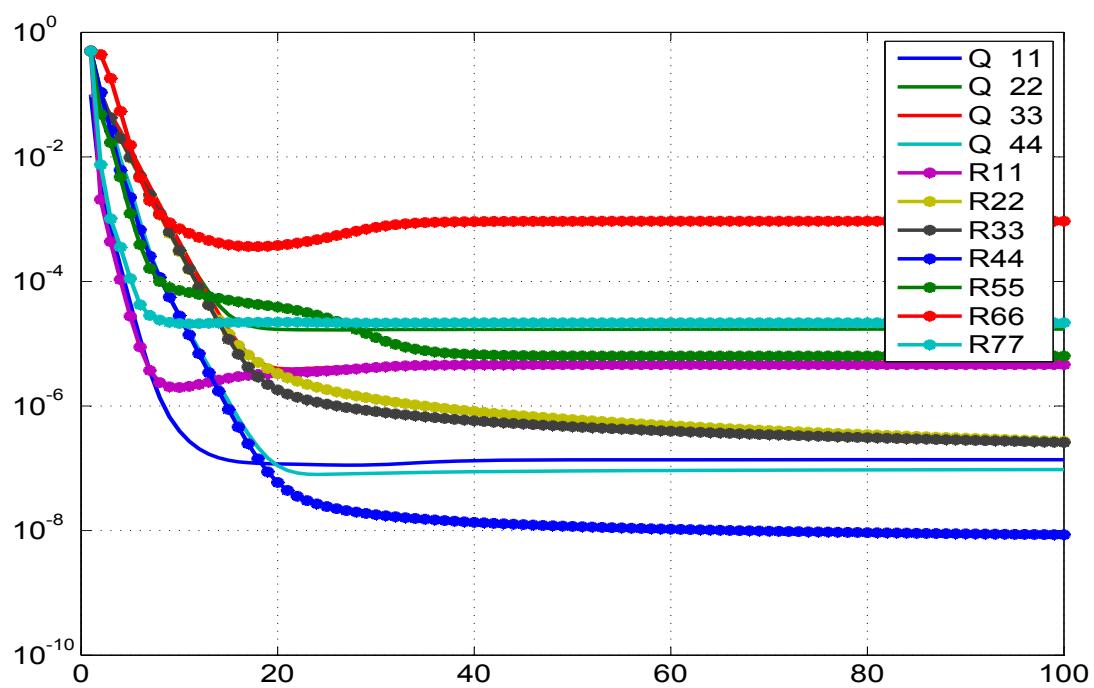
Figure 5.82: Variation of **Q** and **R** with iterations
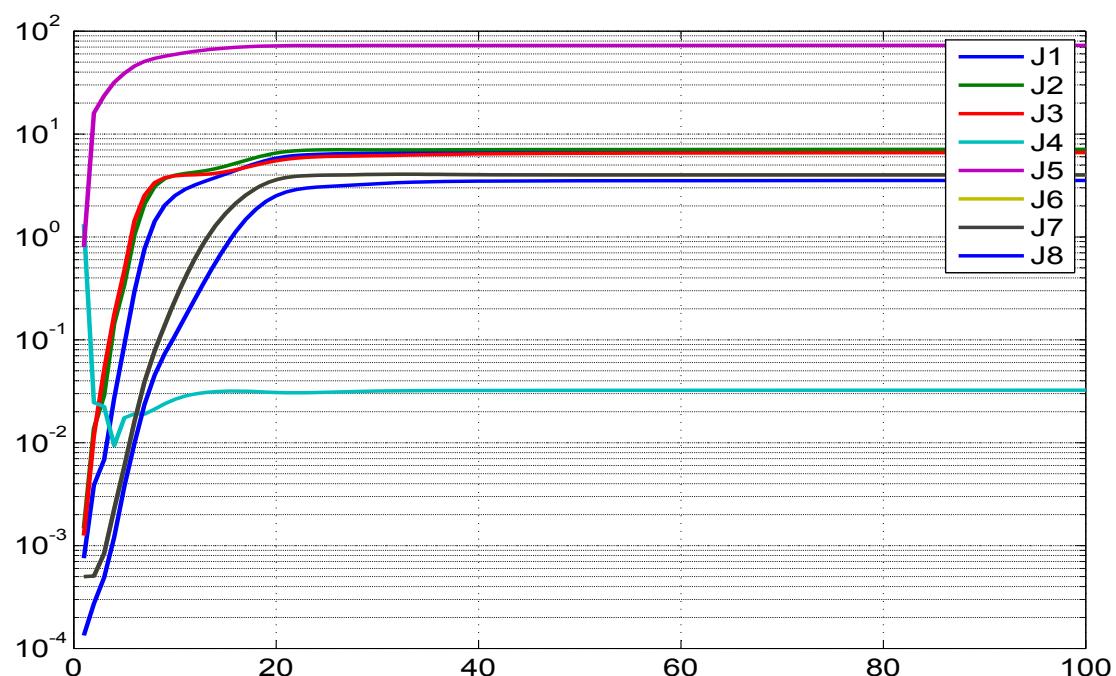
Figure 5.83: Variation of different costs (**J1-J8**) with iterations
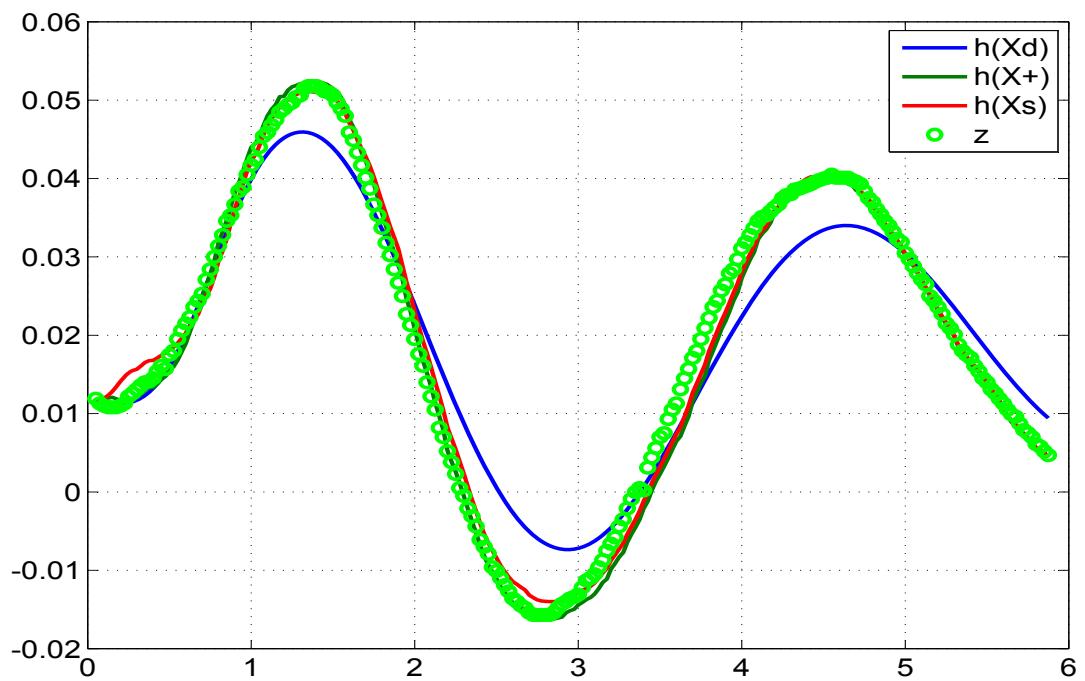
Figure 5.84: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1
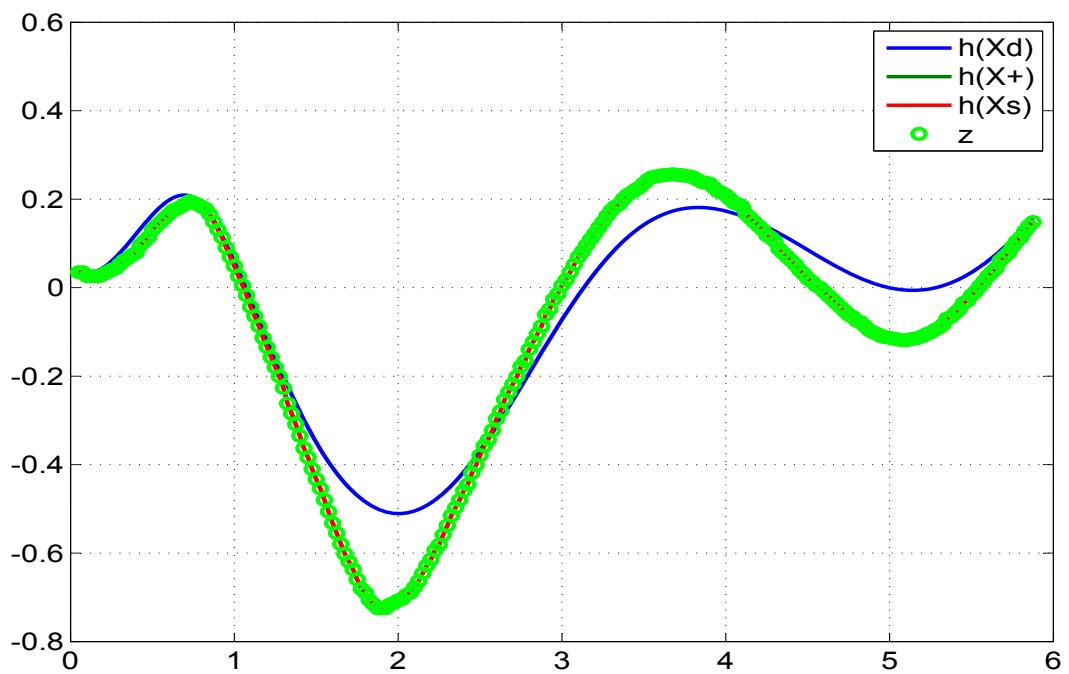
Figure 5.85: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 2
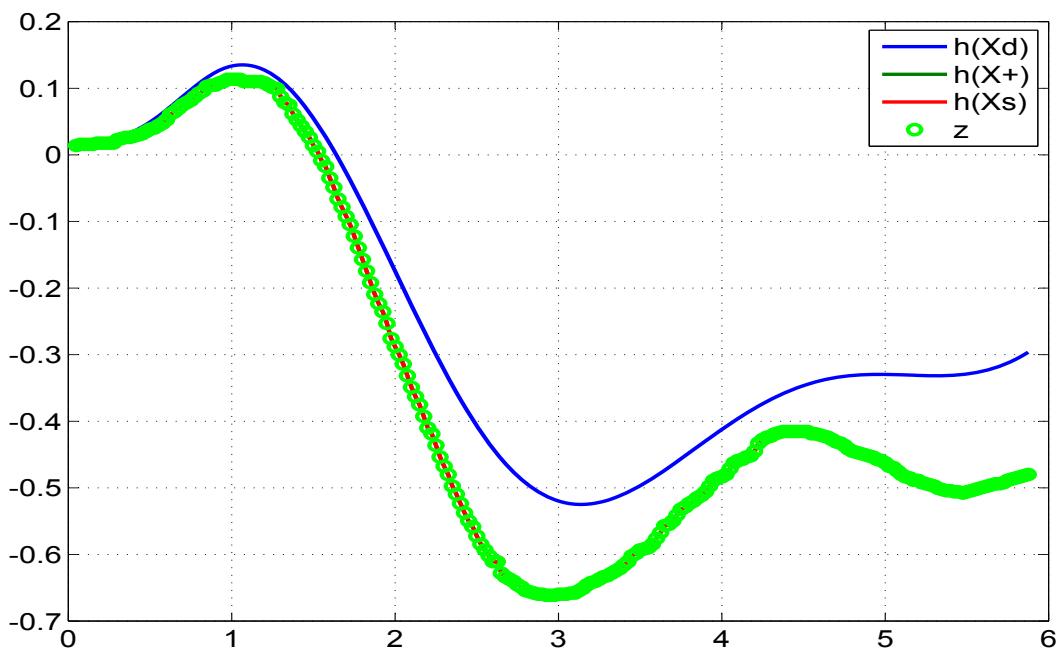
Figure 5.86: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 3
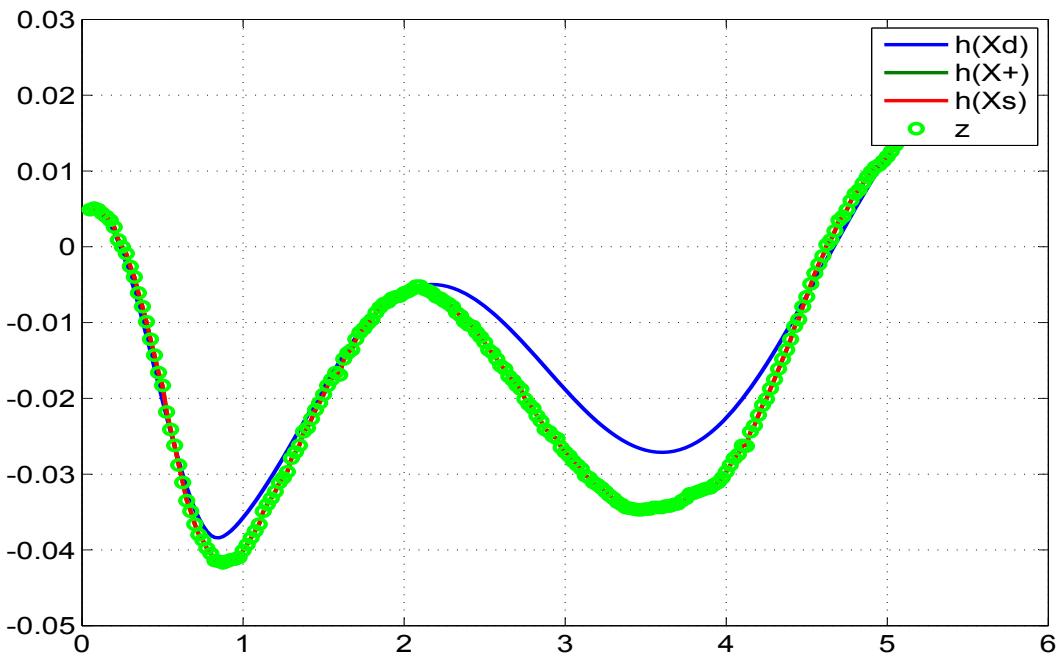
Figure 5.87: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 4
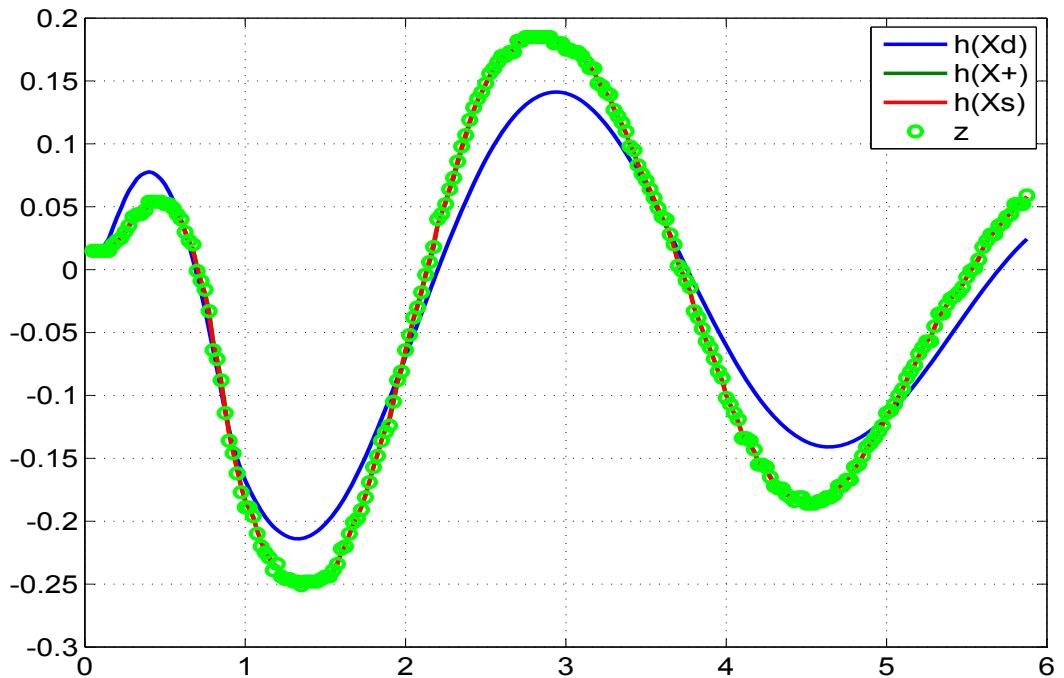
Figure 5.88: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 5
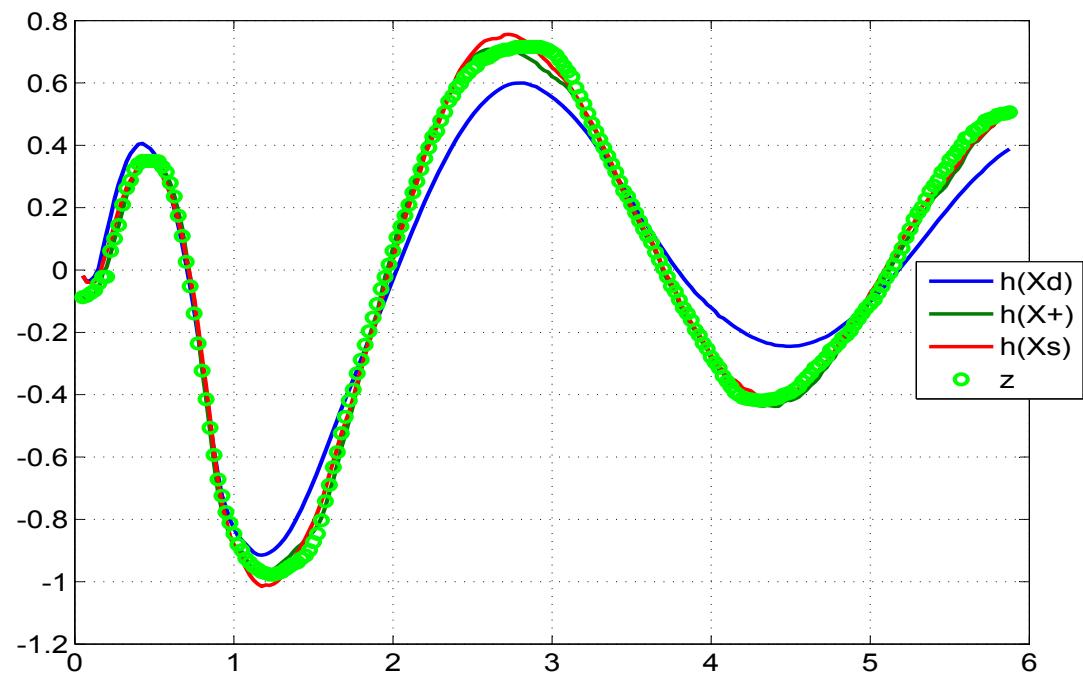
Figure 5.89: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 6
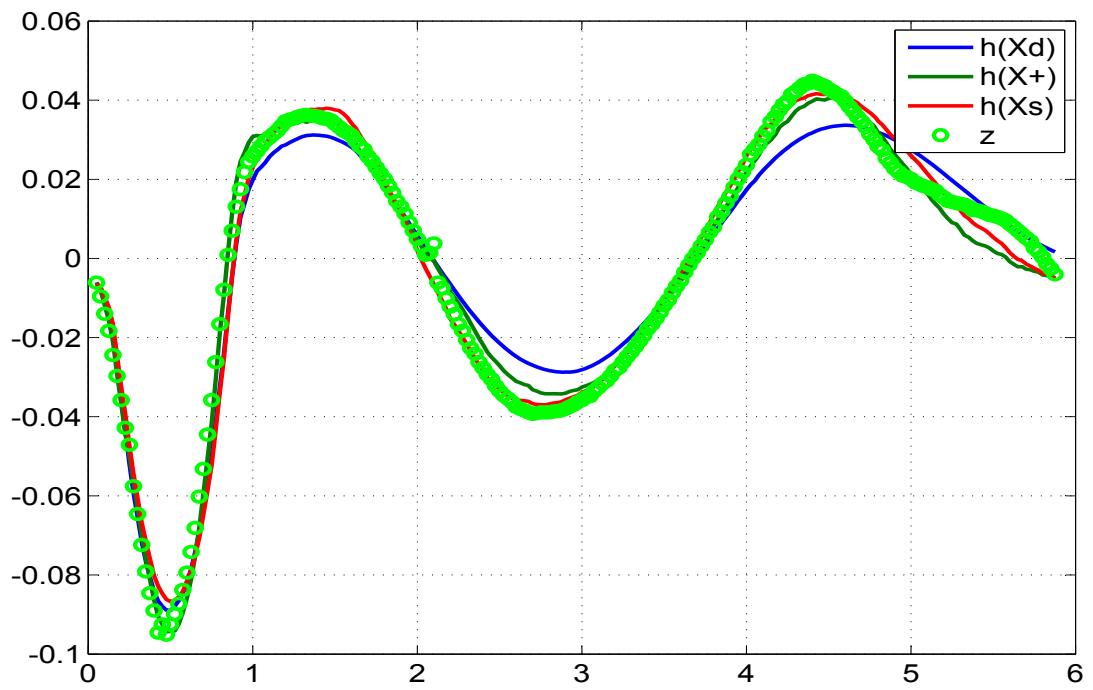
Figure 5.90: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 7
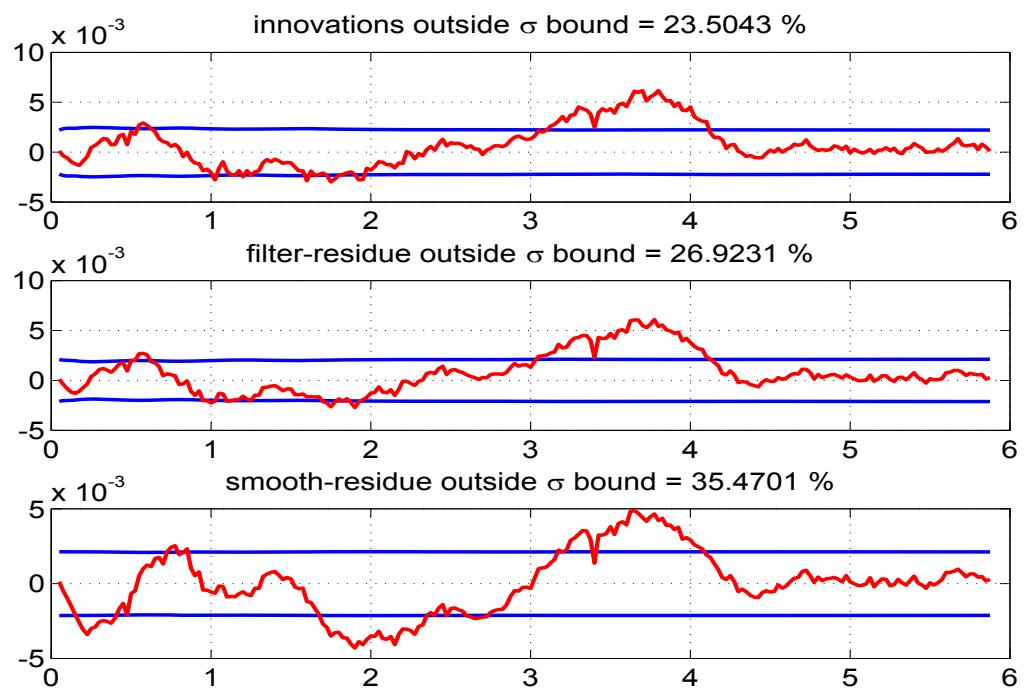
Figure 5.91: The innovations, filtered residue and smoothed residue of measurement 1
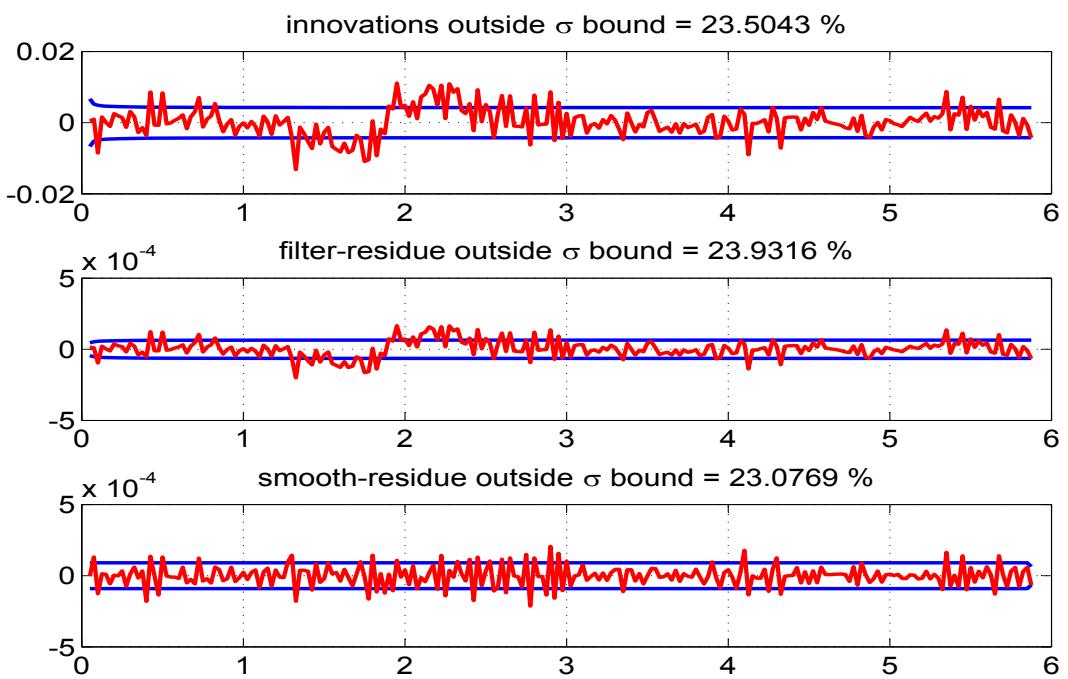
Figure 5.92: The innovations, filtered residue and smoothed residue of measurement 2
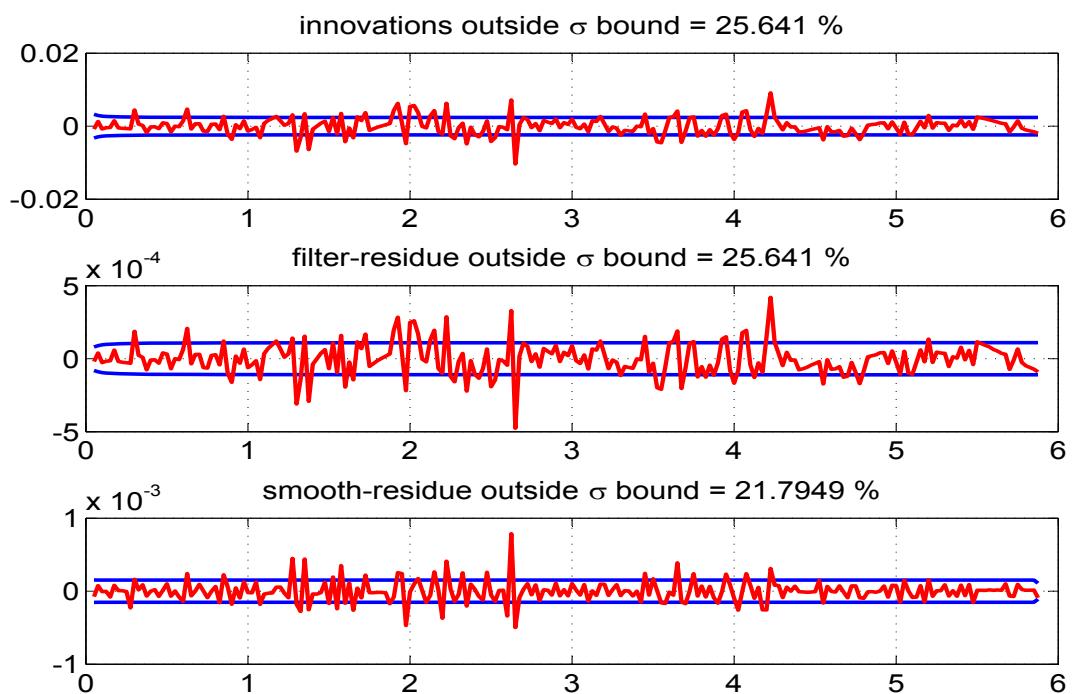
Figure 5.93: The innovations, filtered residue and smoothed residue of measurement 3
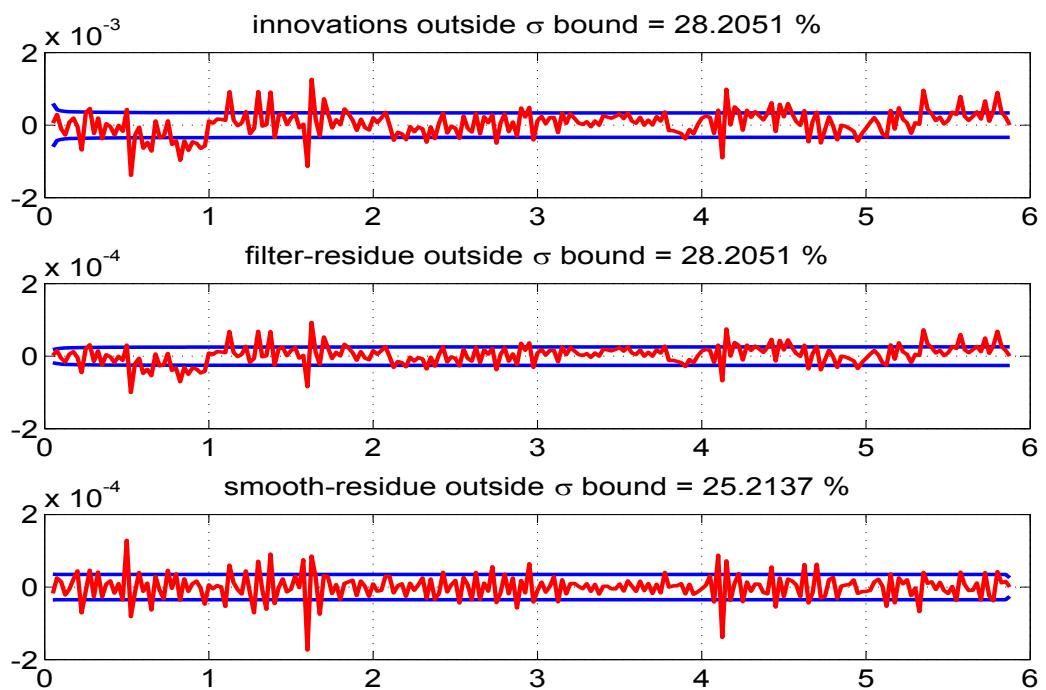
Figure 5.94: The innovations, filtered residue and smoothed residue of measurement 4
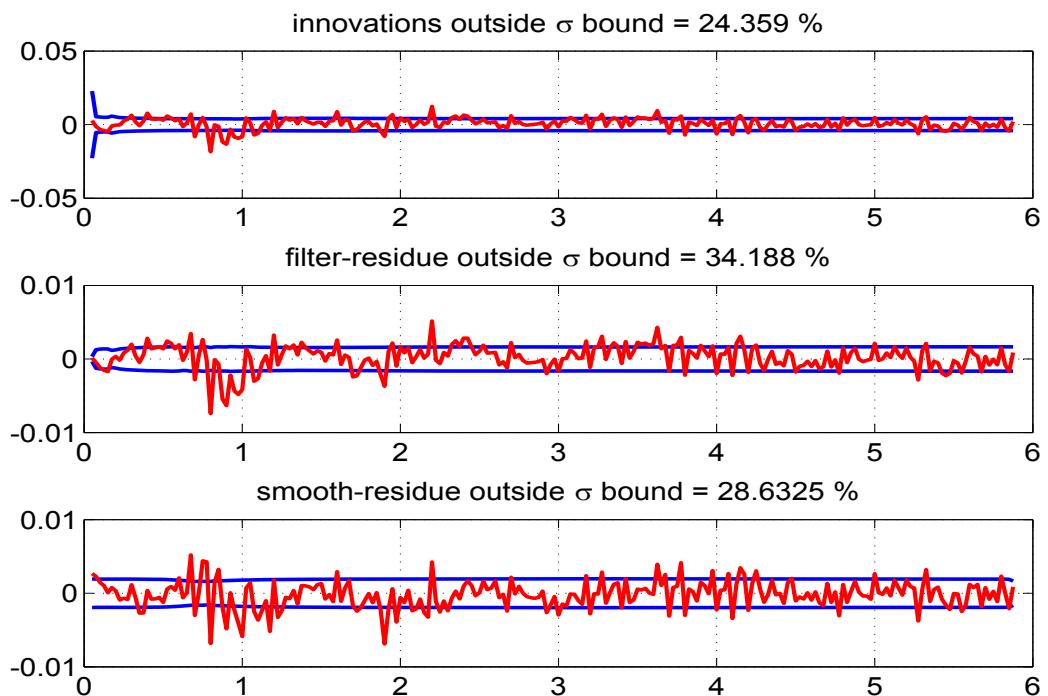
Figure 5.95: The innovations, filtered residue and smoothed residue of measurement 5
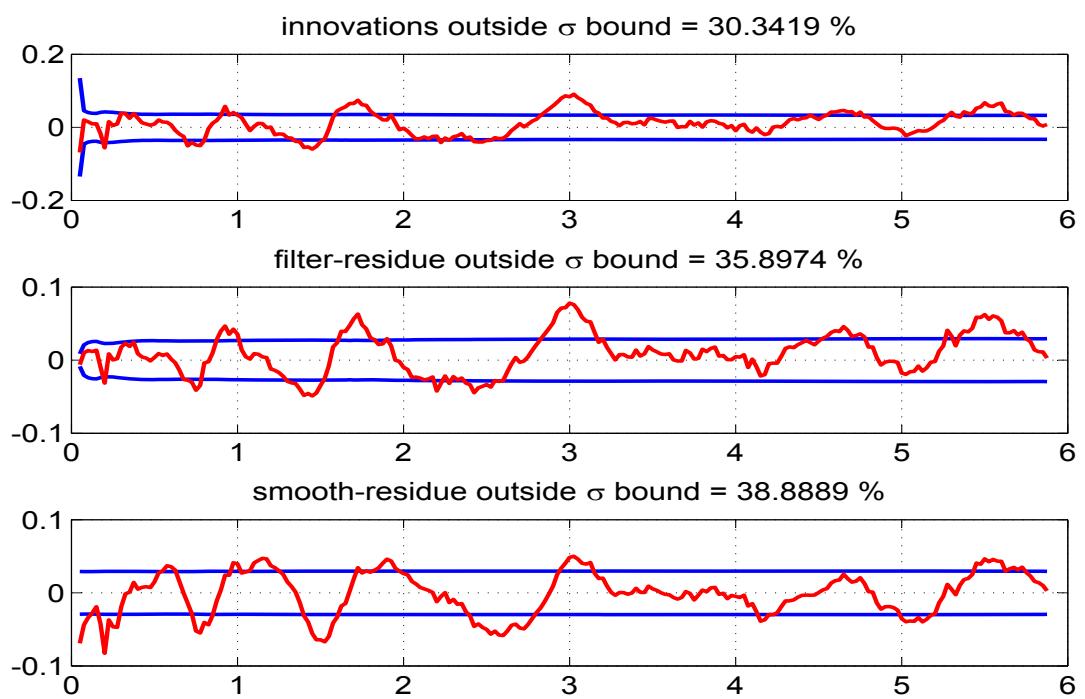
Figure 5.96: The innovations, filtered residue and smoothed residue of measurement 6
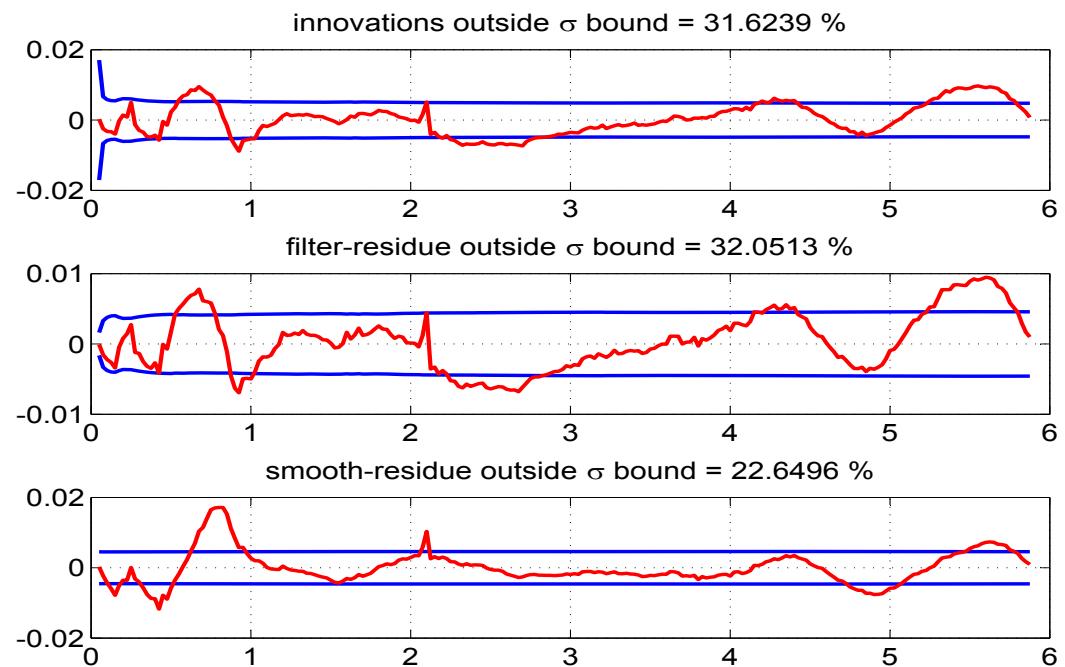
Figure 5.97: The innovations, filtered residue and smoothed residue of measurement 7
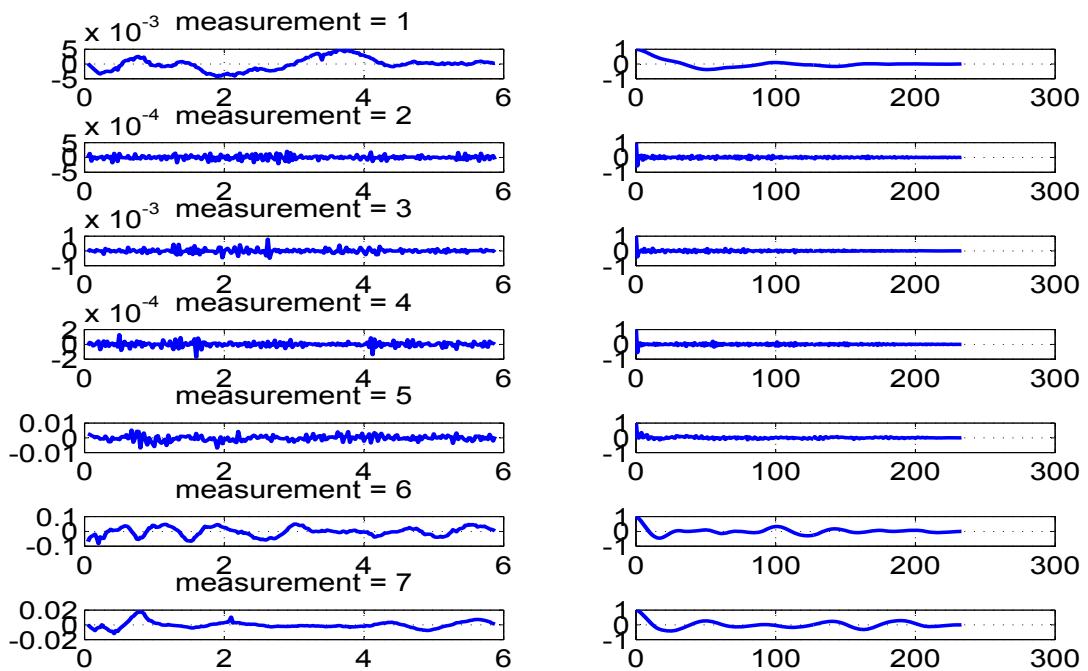
Figure 5.98: Time variation of estimated measurement noise (left) and

their autocorrelation (right)
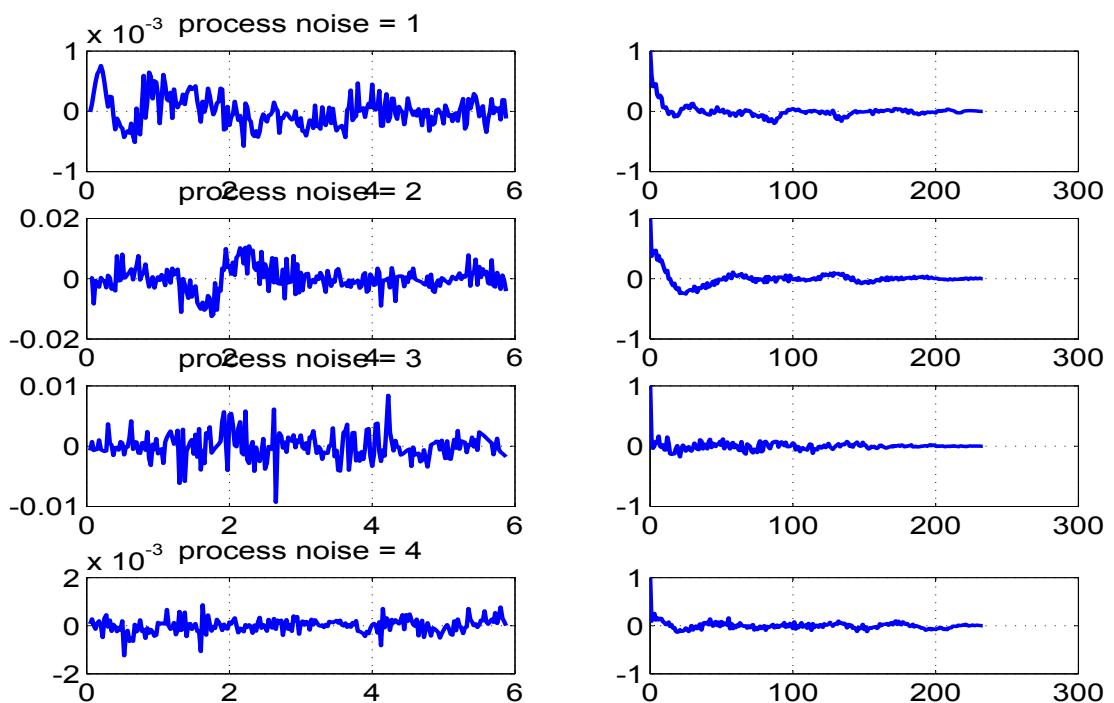
Figure 5.99: Time variation of estimated process noise (left) and

their autocorrelation (right)
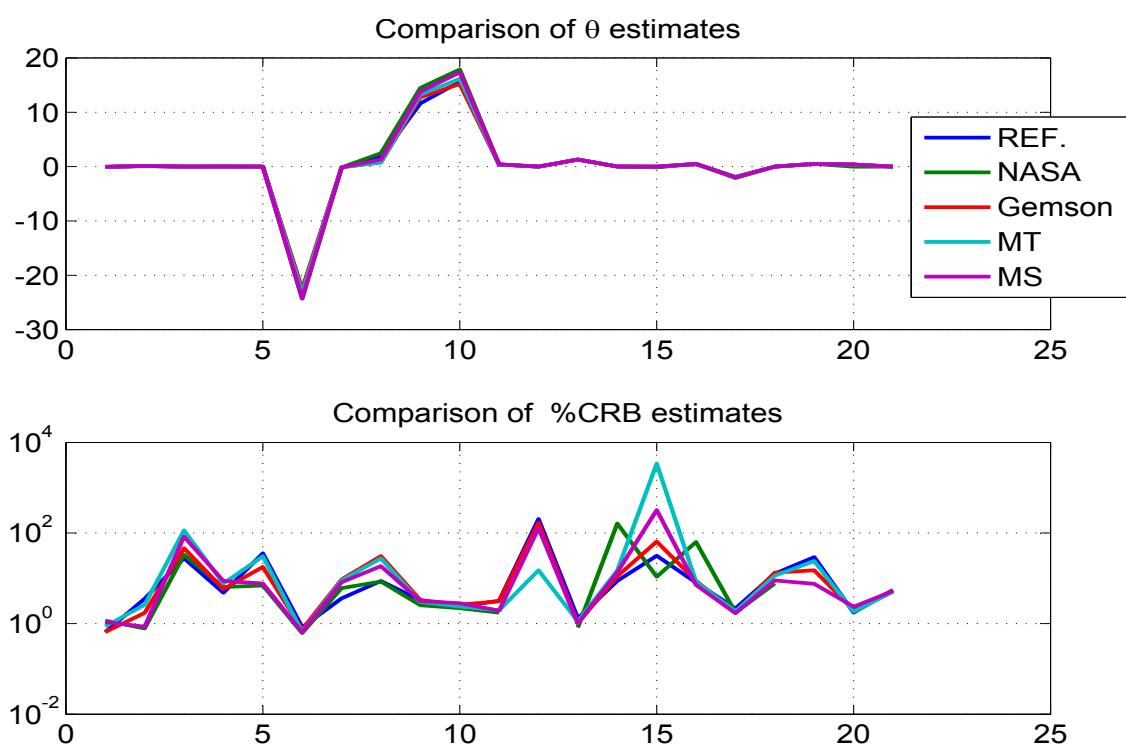
Figure 5.100: Comparison of the parameter estimates and %CRBs by different methods

5.5 Real Flight Test Case-5

The data set is obtained from NASA TP 1690 which describes the lateral motion of a oblique wing aircraft with zero wing skew excited by the control input (δ_a and δ_r in rad) as shown in Fig. 5.101. Some of the available measurements can also be used as inputs in the state equations which includes pitch angle (θ_m in rad), pitch rate (q_m in rad/s) and the angle of attack (α_m in rad) are shown in Fig. 5.102, Fig. 5.103 and Fig. 5.104 respectively. The state equations (n=4) for the angle of sideslip (β), roll rate (p), roll angle (ϕ) and yaw rate (r) are

$$\dot{\beta} = -\frac{\bar{q}S}{mV}(C_{Y_\beta}\beta + C_{Y_p}\frac{\bar{c}}{2V}p + C_{Y_r}\frac{\bar{c}}{2V}r + C_{Y_{\delta_a}}\delta_a + \beta_0) + \frac{g}{V}sin(\phi)cos(\theta_m) + psin(\alpha_m) - rcos(\alpha_m)$$

$$\dot{p} - \dot{r}\frac{Izx}{Ixx} = \frac{\bar{q}Sb}{Ixx}(C_{L_\beta}\beta + C_{L_p}\frac{\bar{c}}{2V}p + C_{L_r}\frac{\bar{c}}{2V}r + C_{L_{\delta_a}}\delta_a + C_{L_0}) + \frac{Iyy - Izz}{Ixx}rq_m + \frac{Izx}{Ixx}pq_m$$

$$\dot{\phi} = p + q\,tan(\theta_m)sin(\phi) + r\,tan(\theta_m)cos(\phi) + \phi_0$$

$$\dot{r} - \dot{p}\frac{Izx}{Izz} = \frac{\bar{q}Sb}{Izz}(C_{N_\beta}\beta + C_{N_p}\frac{\bar{c}}{2V}p + C_{N_r}\frac{\bar{c}}{2V}r + C_{N_{\delta_a}}\delta_a + C_{N_0}) + \frac{Ixx - Iyy}{Izz}pq_m - \frac{Izx}{Izz}rq_m$$

The angle of sideslip (β), roll rate (p), roll angle (ϕ), yaw rate ($\dot{r}$) and lateral acceleration (a_y) are measured (indicated with subscript 'm') in units of are rad, rad/s, rad, rad/s and ft/s^2 respectively. The measurement equations (m=5) are given by

$$\beta_m = \beta - K_\beta z_\beta \frac{p}{V} + K_\beta x_\beta \frac{r}{V}$$

$$p_m = p$$

$$\phi_m = \phi$$

$$r_m = r$$

$$a_{y_m} = \frac{\bar{q}S}{mg}(C_{Y_\beta}\beta + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r + C_{Y_0}) - \frac{z_{a_y}}{g}\dot{p} + \frac{x_{a_y}}{g}\dot{r}$$

The unknown parameter set ($p = 20$) is $\Theta = (C_{Y_\beta}, C_{Y_{\delta_r}}, \beta_0, C_{L_\beta}, C_{L_p}, C_{L_r}, C_{L_{\delta_a}}, C_{L_{\delta_r}}, C_{L_0},$ $\phi_0, C_{N_\beta}, C_{N_p}, C_{N_r}, C_{N_{\delta_a}}, C_{N_{\delta_r}}, C_{N_0}, C_{Y_0}, C_{Y_p}, C_{Y_r}, C_{Y_{\delta_a}})^T$. The ones with suffix '$\delta_a$' and '$\delta_r$' are the control derivatives, the ones with suffix zero are the biases and all others are aerodynamic derivatives. The initial states are taken as initial measurement and the initial parameter values are taken as $(-0.5, 0.1, -0.01, 0.01, -0.35, 0.01, 0.06, 0.01, -0.002,$ $0.002, 0.07, -0.055, -0.05, 0.003, -0.04, 0.0068, -0.025, 0.5, -1, 0.005)^T$.

Other constant values used for case-5

$\bar{q} = 865.3$	S=9.3	m=387.7	Ixx=314	Iyy=488	Izz=698	Izx=69
V=39.41	g=9.81	b=6.81	$K_{\beta}z_{\beta}$=0.305	$K_{\beta}x_{\beta}$=2.73	z_{a_y}=-0.098	x_{a_y}=0.651

Case-5 real data is run using the reference EKF ($\mathbf{Q} > 0$) with 100 iterations. Fig. 5.101-5.104 are the inputs used in state equations. The Fig. 5.105 shows the variation of parameter estimates and its initial covariance $\mathbf{P_0}$ with iterations and a similar Fig. 5.106 for $\mathbf{Q}$ and $\mathbf{R}$. The values of $\mathbf{J1}$-$\mathbf{J3}$ are close to the number of measurements ($m = 5$) with $\mathbf{J6}$-$\mathbf{J8}$ are close to the number of states ($n = 4$) as shown in Fig. 5.107 and Table-5.14. The $\mathbf{J5}$ is the negative log likelihood cost function. The later Fig. 5.108-5.112 compares (i) the state dynamics based on the estimated parameter after the filter pass through the data, (ii) the state after measurement update, (iii) the smoothed state and (iv) the measurement. The Fig. 5.113-5.117 shows the confidence in the innovations, filtered residue and smoothed residue. The estimated measurement and process noise do not appear to have constant statistical characteristics across time as seen in Fig. 5.118 and Fig. 5.119. Another experiment was carried out by generating a typical data set by using the estimated theta and injecting the estimated $\mathbf{Q}$ and $\mathbf{R}$ as additive white Gaussian noise. This is to determine the effect of non additive, non White and non Gaussian noise distribution in the real data on the CRBs. After each iteration in the reference recipe the Θ, $\mathbf{Q}$ and $\mathbf{R}$ were reset as from the real data. Similar experiment was also conducted by updating Θ as well. It turned out that there is not much of a difference in the final CRBs as can be seen from the Table-5.15.

Finally two other filter runs were carried out using the MT and MS statistics for the estimation of $\mathbf{Q}$ and $\mathbf{R}$. The behaviour of the various cost function and in particular $\mathbf{J6}$ and $\mathbf{J7}$ in Table-5.16 shows that the choice of the filter statistics for estimating $\mathbf{Q}$ and $\mathbf{R}$ in the proposed reference approach is the best possible when compared to other approaches presently considered.

5.5.1 Remarks on Case-5

The NASA results have been generated assuming $\mathbf{Q} = 0$ and are comparable with reference procedure for the parameter estimates and their CRBs. Further the MT and MS methods give quite different estimates for the $\mathbf{Q}$ and $\mathbf{R}$ values than in the reference case. We believe that the reference procedure provides the best possible parameter estimates and their uncertainties. From the plot of the parameter estimates and their %CRB in Fig 5.120, it can be seen that the parameters 1, 2, 4-9, 11-13, 15, 16 and 17 are strong and the parameters 3, 10, 14, 18, 19 and 20 are the weak ones and others can be considered as modestly controlling the dynamics of the system. The CRBs as estimated by different methods generally appear to vary widely. However what is interesting is that even the estimate of the strong parameter such as 5 varies widely among the methods. Such a behaviour of the filter across the parameter estimates shows how important is the tuning of the filter statistics namely $\mathbf{P_0}$, $\mathbf{Q}$ and $\mathbf{R}$ in parameter estimation and their uncertainties.

The rounded off 100×C matrix for case-5 is given by

$$
\begin{bmatrix}
100 & -48 & -2 & 13 & -2 & -1 & -2 & -6 & -7 & 0 & 27 & -3 & -2 & -4 & -13 & -14 & -53 & 0 & 0 & -14 \\
-48 & 100 & 4 & -6 & 0 & -3 & 2 & 13 & 13 & 0 & -13 & -1 & -6 & 4 & 27 & 26 & 96 & 0 & -1 & 16 \\
-2 & 4 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -8 & 0 & -1 & 3 & 14 & 14 & 0 & -1 & -5 & 1 \\
13 & -6 & 0 & 100 & -12 & -9 & -15 & -50 & -53 & -1 & 30 & -4 & -3 & -5 & -15 & -16 & -7 & 0 & 0 & -2 \\
-2 & 0 & 0 & -12 & 100 & 5 & 83 & -4 & -1 & 0 & -4 & 31 & 2 & 26 & -1 & 0 & 0 & 0 & 0 & 12 \\
-1 & -3 & 0 & -9 & 5 & 100 & 2 & -22 & -5 & 0 & -3 & 2 & 31 & 1 & -7 & -1 & -1 & 0 & 1 & 0 \\
-2 & 2 & 0 & -15 & 83 & 2 & 100 & 16 & 18 & 0 & -4 & 26 & 1 & 31 & 5 & 6 & 2 & 0 & 0 & 13 \\
-6 & 13 & 0 & -50 & -4 & -22 & 16 & 100 & 96 & 2 & -15 & -1 & -7 & 5 & 30 & 29 & 13 & 0 & 0 & 2 \\
-7 & 13 & 0 & -53 & -1 & -5 & 18 & 96 & 100 & 2 & -16 & 0 & -1 & 6 & 29 & 30 & 13 & 0 & 0 & 2 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 2 & 2 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
27 & -13 & -8 & 30 & -4 & -3 & -4 & -15 & -16 & 0 & 100 & -12 & -9 & -15 & -49 & -53 & -14 & -2 & -1 & -4 \\
-3 & -1 & 0 & -4 & 31 & 2 & 26 & -1 & 0 & 0 & -12 & 100 & 5 & 83 & -3 & -1 & 0 & 15 & 1 & 24 \\
-2 & -6 & -1 & -3 & 2 & 31 & 1 & -7 & -1 & 0 & -9 & 5 & 100 & 2 & -22 & -5 & -1 & 1 & 14 & 0 \\
-4 & 4 & 3 & -5 & 26 & 1 & 31 & 5 & 6 & 0 & -15 & 83 & 2 & 100 & 16 & 18 & 5 & 12 & 0 & 27 \\
-13 & 27 & 14 & -15 & -1 & -7 & 5 & 30 & 29 & 0 & -49 & -3 & -22 & 16 & 100 & 96 & 26 & -1 & -3 & 4 \\
-14 & 26 & 14 & -16 & 0 & -1 & 6 & 29 & 30 & 0 & -53 & -1 & -5 & 18 & 96 & 100 & 27 & 0 & -1 & 5 \\
-53 & 96 & 0 & -7 & 0 & -1 & 2 & 13 & 13 & 0 & -14 & 0 & -1 & 5 & 26 & 27 & 100 & 0 & 0 & 18 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 15 & 1 & 12 & -1 & 0 & 0 & 100 & 5 & 3 \\
0 & -1 & -5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 14 & 0 & -3 & -1 & 0 & 5 & 100 & 0 \\
-14 & 16 & 1 & -2 & 12 & 0 & 13 & 2 & 2 & 0 & -4 & 24 & 0 & 27 & 4 & 5 & 18 & 3 & 0 & 100
\end{bmatrix}
$$

5.5.2 Case-5 Tables

Table 5.14: Real flight test data case-5 results using the reference adaptive EKF
No of iterations=100

Study	Θ (Ref)	Θ (NASA)	Θ (Gemson)	σ_Θ (Ref)	σ_Θ (NASA)	σ_Θ (Gemson)	R $\times 10^{-6}$ (Ref)	Q $\times 10^{-6}$ (Ref)	J1-J8 (Ref)	Remarks
	-0.4529	-0.4792	-0.4761	0.0047	0.01711	0.0043				
	0.0739	0.0887	0.0981	0.0058	0.01955	0.0065				
	-0.0108	-0.10116	-0.0124	0.0048	0.00294	0.0021				
	-0.0168	-0.0205	-0.0182	0.0005	0.00107	0.0011				
	-0.3108	-0.36	-0.3585	0.0028	0.00713	0.0048				
	0.0749	0.0697	0.0731	0.0030	0.005884	0.0066				
	0.0558	0.0612	0.0622	0.0004	0.001050	0.0007			4.7460	
	0.0074	0.006	0.0089	0.0007	0.001252	0.0031			4.8152	
	-0.0020	-0.002	-0.0023	0.0001	0.0001467	0.0003	0.09		3.5033	
P_0 : Scaled up-[0,0;0,✓]	0.0018	0.1506	0.0023	0.0027	0.07034	0.0011	0.06	4.191	0.0004	
Q : EM-[✓,0;0,0]	0.0655	0.0705	0.0703	0.0005	0.000478	0.0009	0.23	5.171	-54.7239	Cost functions converge
R : EM-diag	-0.0474	-0.046	-0.0557	0.0031	0.004006	0.0039	0.02	4.943	3.9627	to the expected values
	-0.0818	-0.1062	-0.0576	0.0032	0.003562	0.0045	57.68	1.472	3.9620	
	0.0011	0.0006	0.0033	0.0005	0.0005924	0.0006			3.7135	
	-0.0470	-0.0513	-0.048	0.0008	0.0009139	0.0013				
	0.0066	0.0072	0.0068	0.0001	0.0001181	0.0002				
	-0.0221	-0.0242	-0.0251	0.0007	0.002307	0.0007				
	0.0650	–	–	0.4146	–	–				
	-1.2731	–	–	0.6334	–	–				
	0.0007	–	–	0.0023	–	–				

Table 5.15: Case-5 results using simulated Additive White Gaussian Noise

Study	σ_Θ (Simulated-without updating Θ)	σ_Θ (Simulated-with updating Θ)	σ_Θ (Ref)	Remarks
Case-5 data generated using simulated measurement and process noise (AWGN) of variance $\mathbf{Q}$ and $\mathbf{R}$ estimated by Reference EKF ($\mathbf{Q} > 0$)				
$\mathbf{P_0}$: Scaled up-$[0,0;0,\checkmark]$ $\mathbf{Q}$: $\mathbf{Q}$ (Ref) $\mathbf{R}$: $\mathbf{R}$ (Ref)	0.0052	0.0052	0.0047	No Significant change in σ_Θ
	0.0059	0.0059	0.0058	
	0.0048	0.0048	0.0048	
	0.0005	0.0005	0.0005	
	0.0030	0.0030	0.0028	
	0.0031	0.0031	0.0030	
	0.0004	0.0004	0.0004	
	0.0007	0.0007	0.0007	
	0.0001	0.0001	0.0001	
	0.0027	0.0027	0.0027	
	0.0006	0.0006	0.0005	
	0.0032	0.0032	0.0031	
	0.0034	0.0034	0.0032	
	0.0005	0.0005	0.0005	
	0.0007	0.0007	0.0008	
	0.0001	0.0001	0.0001	
	0.0007	0.0007	0.0007	
	0.4092	0.4092	0.4146	
	0.6996	0.6996	0.6334	
	0.0023	0.0023	0.0023	

Table 5.16: Real flight test data case-5 results using the MT and MS method.
No of iterations=100

Study	Θ (MT)	σ_Θ (MT)	R (MT) $\times10^{-6}$	Q (MT) $\times10^{-6}$	J1-J8 (MT)	Θ (MS)	σ_Θ (MS)	R (MS) $\times10^{-6}$	Q (MS) $\times10^{-6}$	J1-J8 (MS)	Remarks
	-0.4541	0.0053				-0.4642	0.0049				
	0.0741	0.0065				0.0797	0.0057				
	-0.0107	0.0034				-0.0109	0.0009				
	-0.0170	0.0004				-0.0177	0.0003				
	-0.3112	0.0027				-0.3080	0.0022				
	0.0733	0.0028				0.0757	0.0022				
	0.0557	0.0004			4.3450	0.0546	0.0003			4.8127	
	0.0073	0.0007			4.3888	0.0082	0.0005			4.8200	
	-0.0020	0.0001	0.0028		3.1039	-0.0021	0.0001	0.0130		4.5173	
$\mathbf{P_0}$: Scaled up-[0,0;0,✓]	0.0019	0.0012	0.0189	2.0481	0.0003	0.0019	0.0013	0.0881	0.0005	0.0003	Cost functions are
$\mathbf{Q}$: MT/MS-[✓,0;0,0]	0.0657	0.0004	0.0039	3.7876	-51.3490	0.0662	0.0004	0.0045	0.0007	-47.2441	not close to their
$\mathbf{R}$: MT/MS-diag	-0.0473	0.0023	0.0041	1.0057	9.3006	-0.0596	0.0021	0.0364	1.0975	7.5931	expected values in
	-0.0854	0.0023	0.0736	0.5502	9.3005	-0.1021	0.0010	0.0601	0.0016	7.5896	MT and MS method
	0.0010	0.0003			3.5105	0.0030	0.0003			3.9681	
	-0.0476	0.0006				-0.0502	0.0003				
	0.0067	0.0001				0.0069	0.00003				
	-0.0221	0.0008				-0.0228	0.0007				
	0.0821	0.2941				0.5223	0.0710				
	-1.2336	0.4470				-0.8452	0.1355				
	0.0011	0.0026				-0.0017	0.0024				

5.5.3 Case-5 Figures
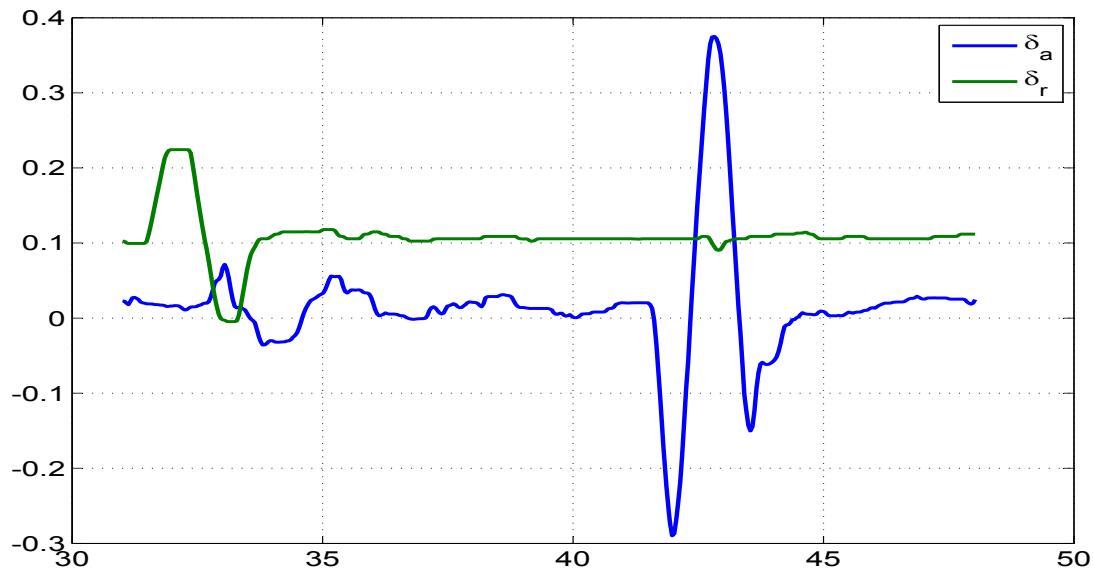
Figure 5.101: Control input (δ_a, δ_r) versus time (s)
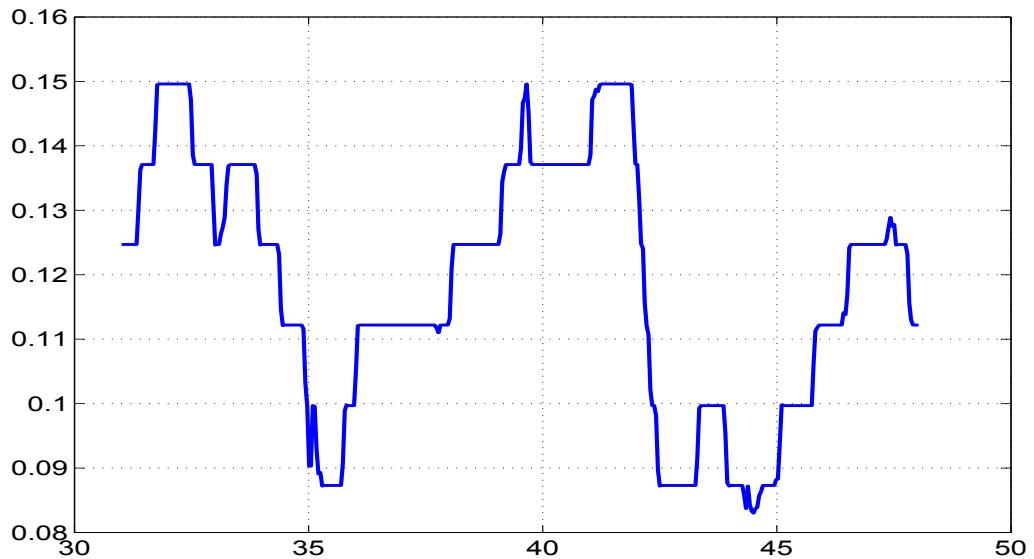
Figure 5.102: Measurement input versus (Θ_m) time (s)
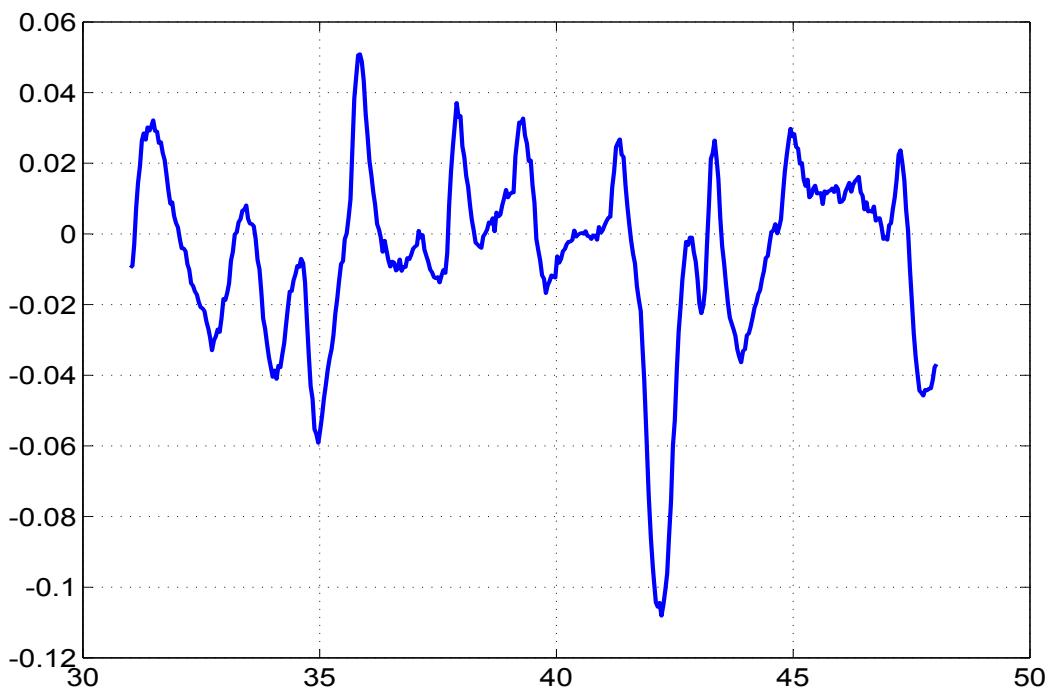
Figure 5.103: Measurement input versus (q_m) time (s)
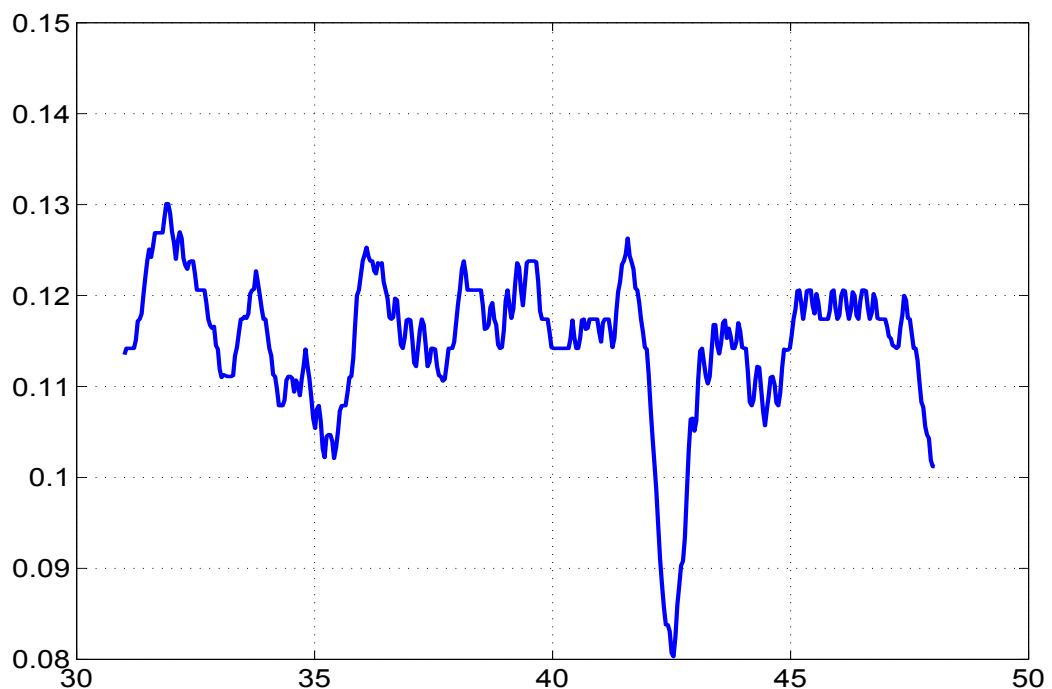
Figure 5.104: Measurement input versus (α_m) time (s)

302
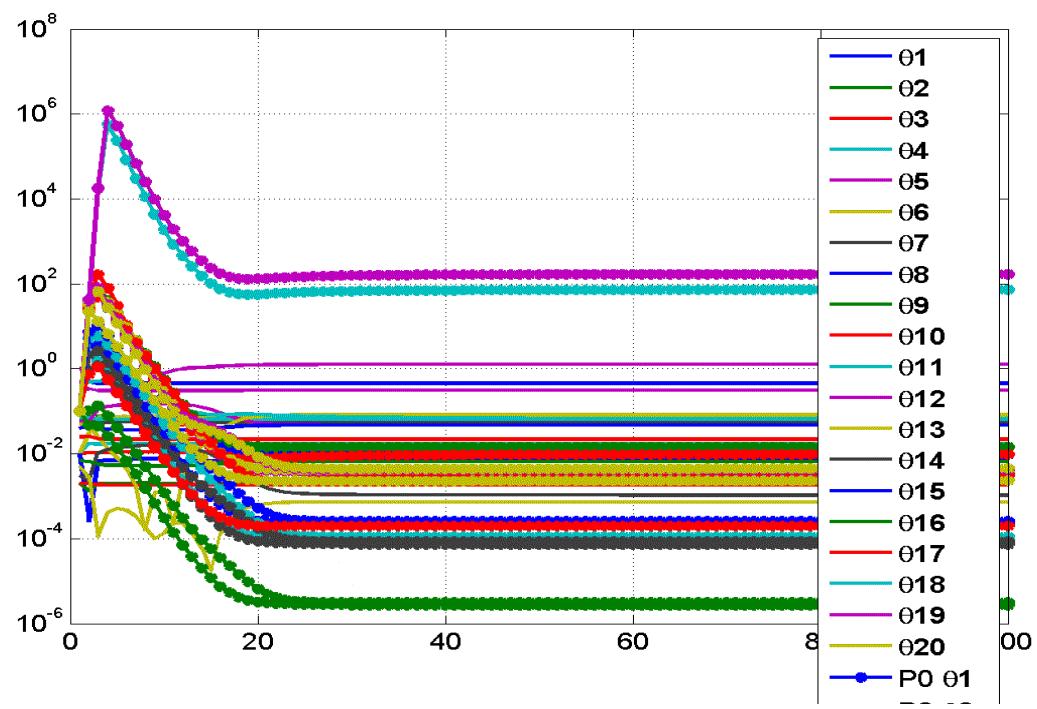
Figure 5.105: Variation of parameter and its initial covariance ($\mathbf{P_0}$) with iterations
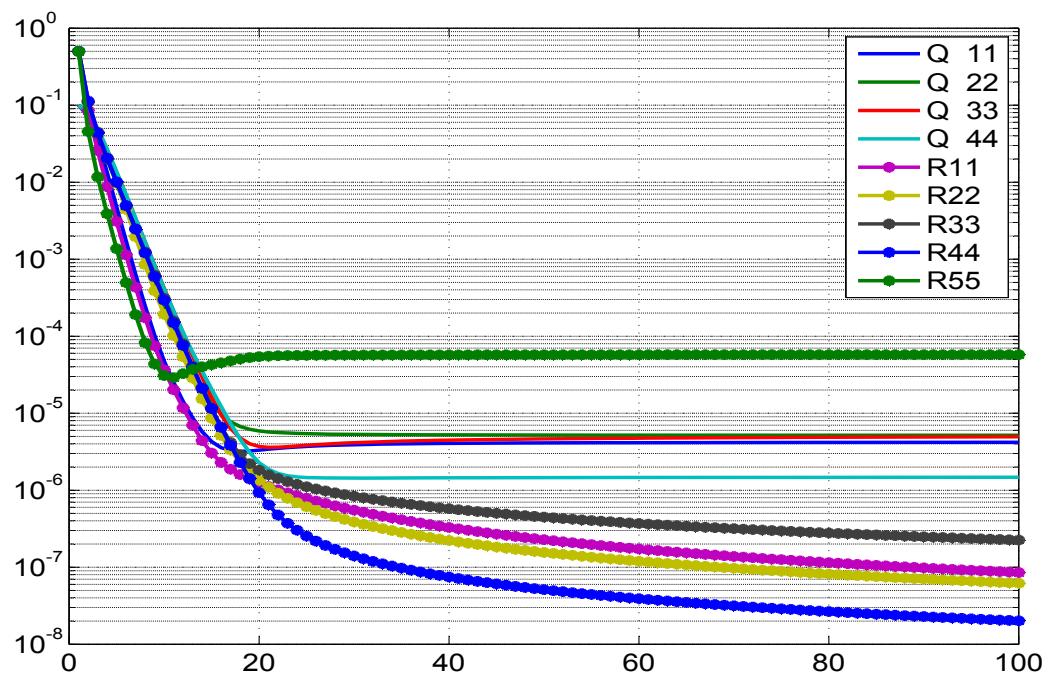
Figure 5.106: Variation of $\mathbf{R}$ with iterations
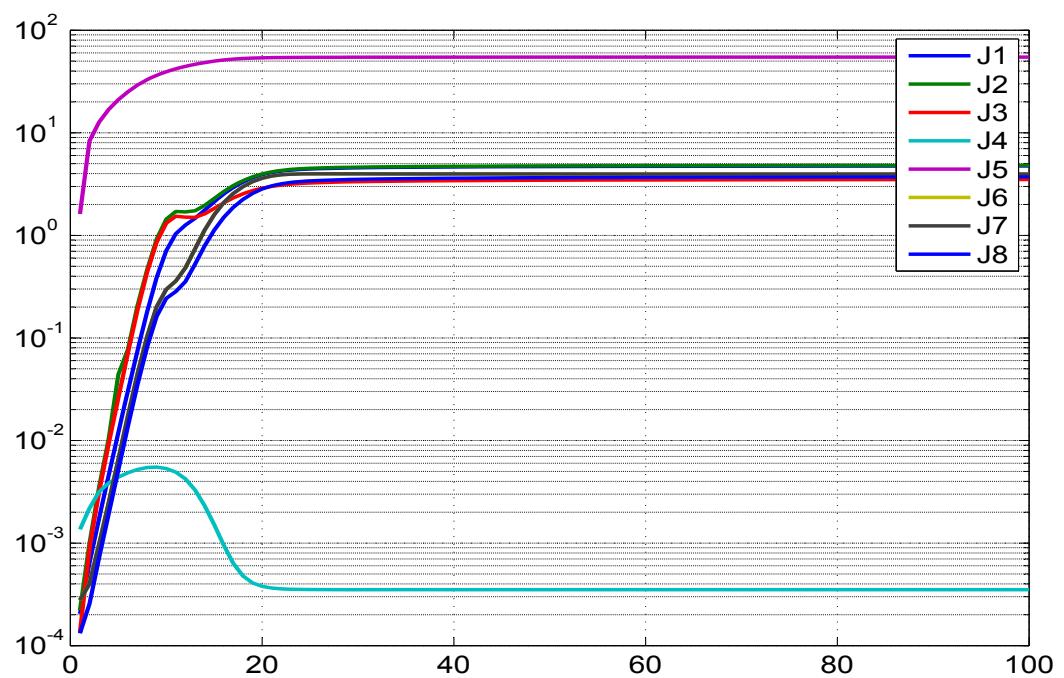
Figure 5.107: Variation of different costs (**J1-J8**) with iterations
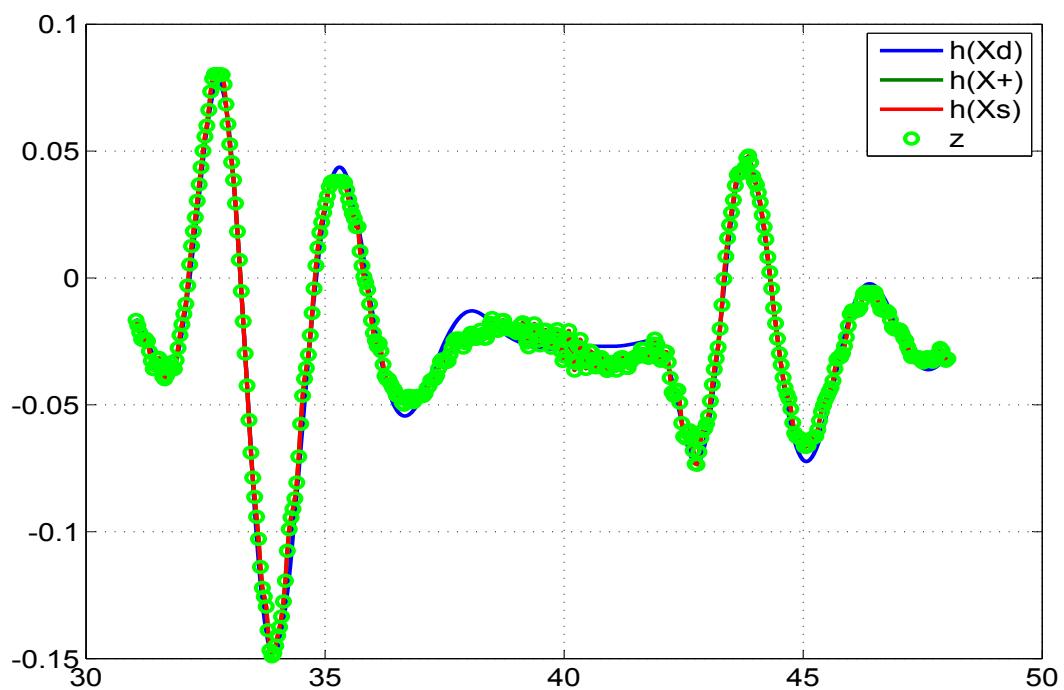
Figure 5.108: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 1
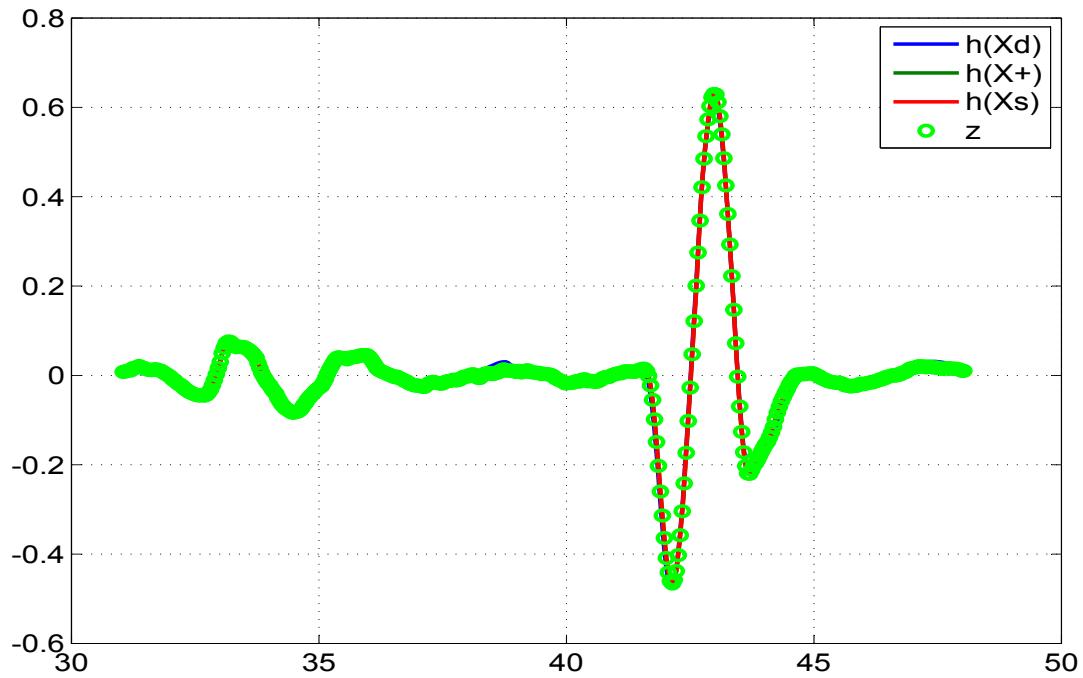
Figure 5.109: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 2
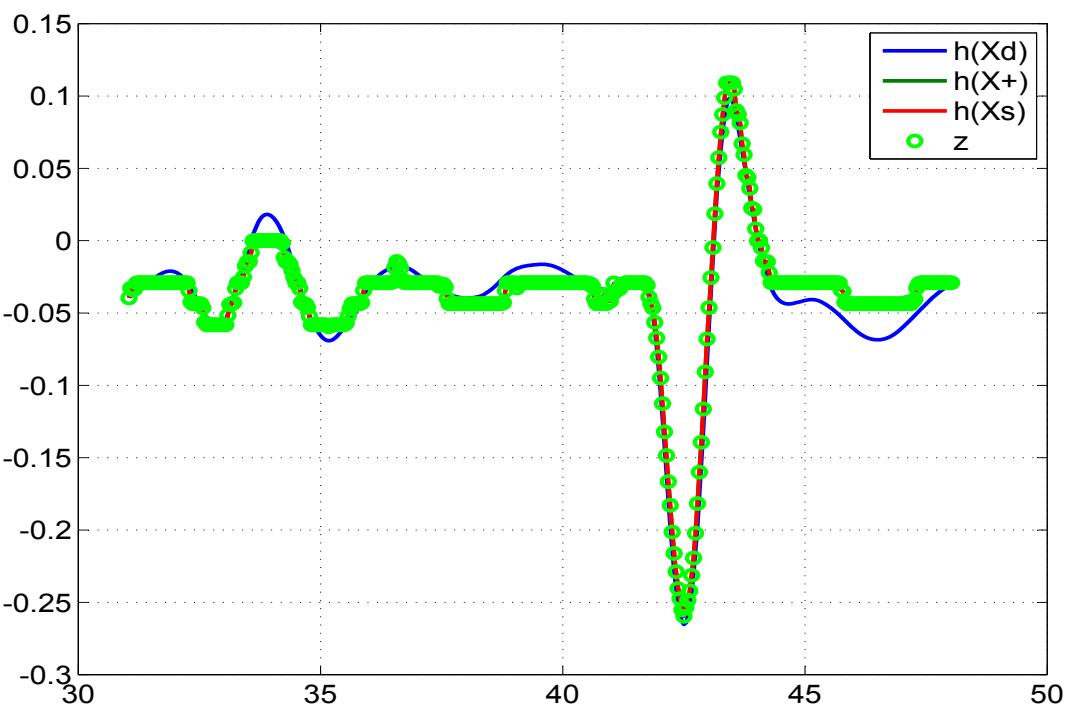
Figure 5.110: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 3
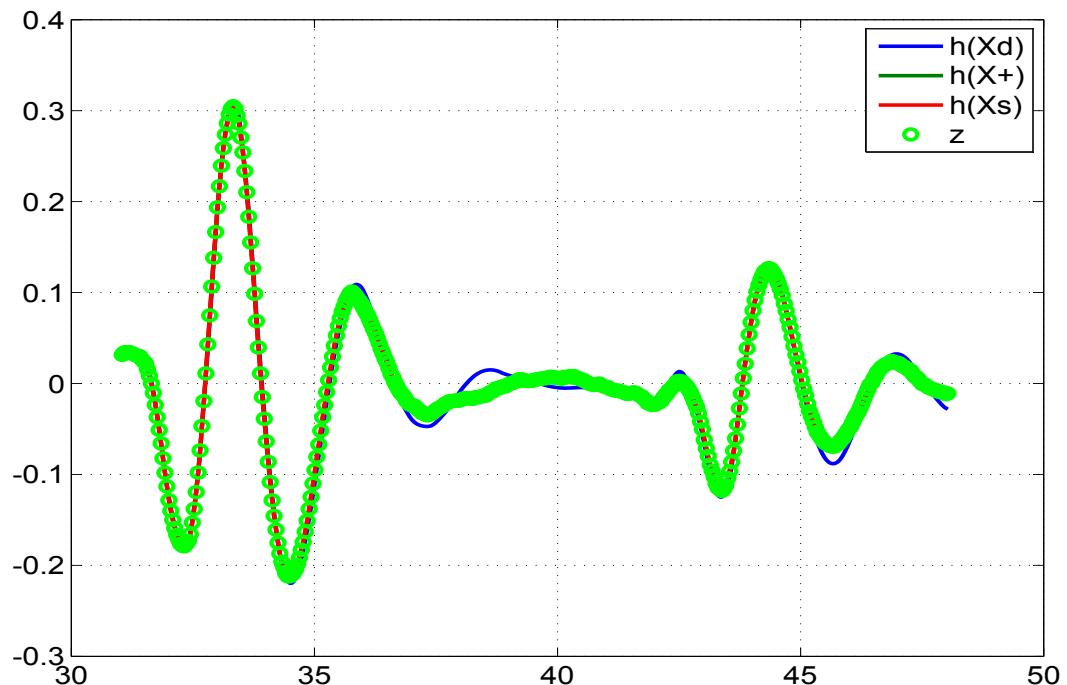
Figure 5.111: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 4
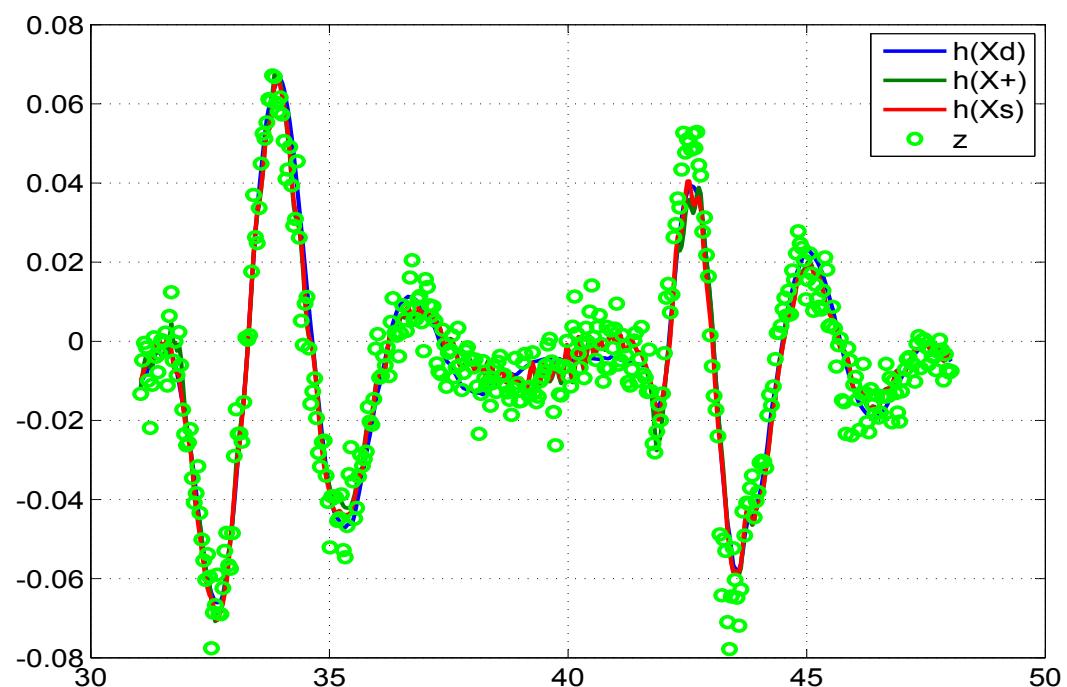
Figure 5.112: Comparison of the predicted dynamics, posterior, smoothed

and the measurement 5
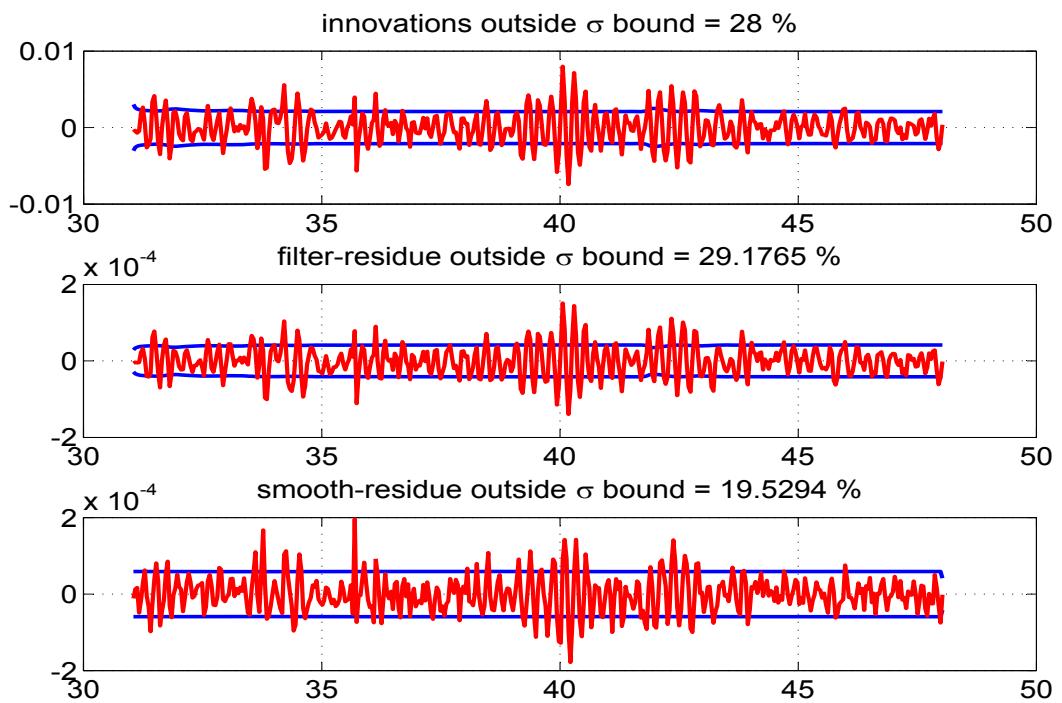
Figure 5.113: The innovations, filtered residue and smoothed residue of measurement 1
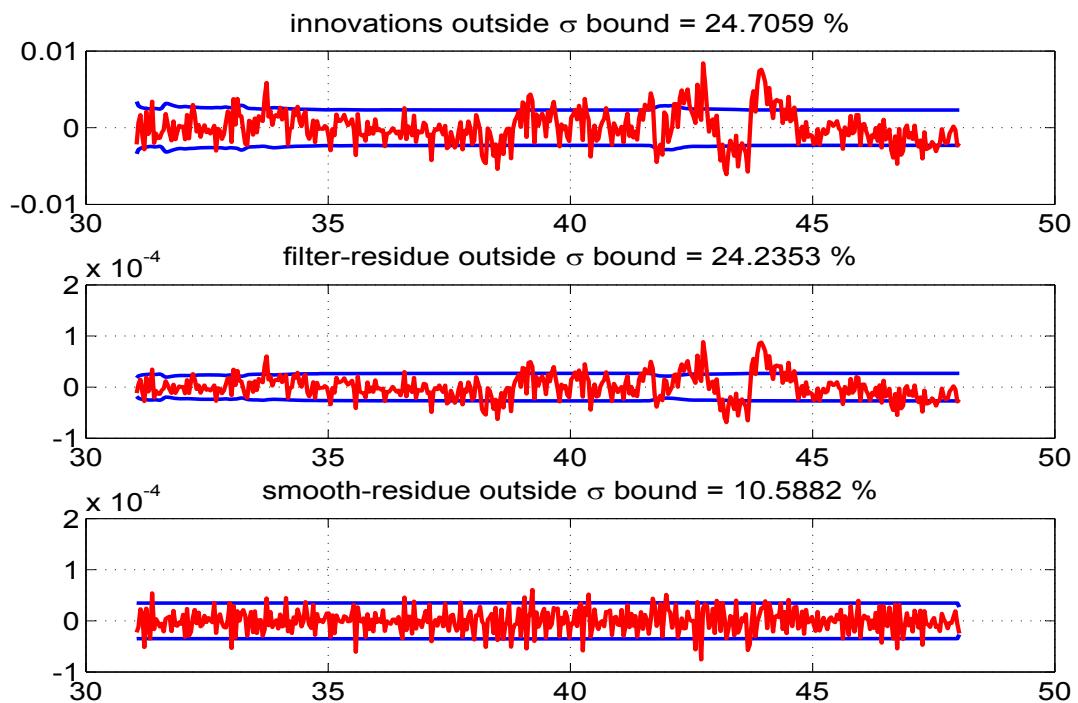
Figure 5.114: The innovations, filtered residue and smoothed residue of measurement 2
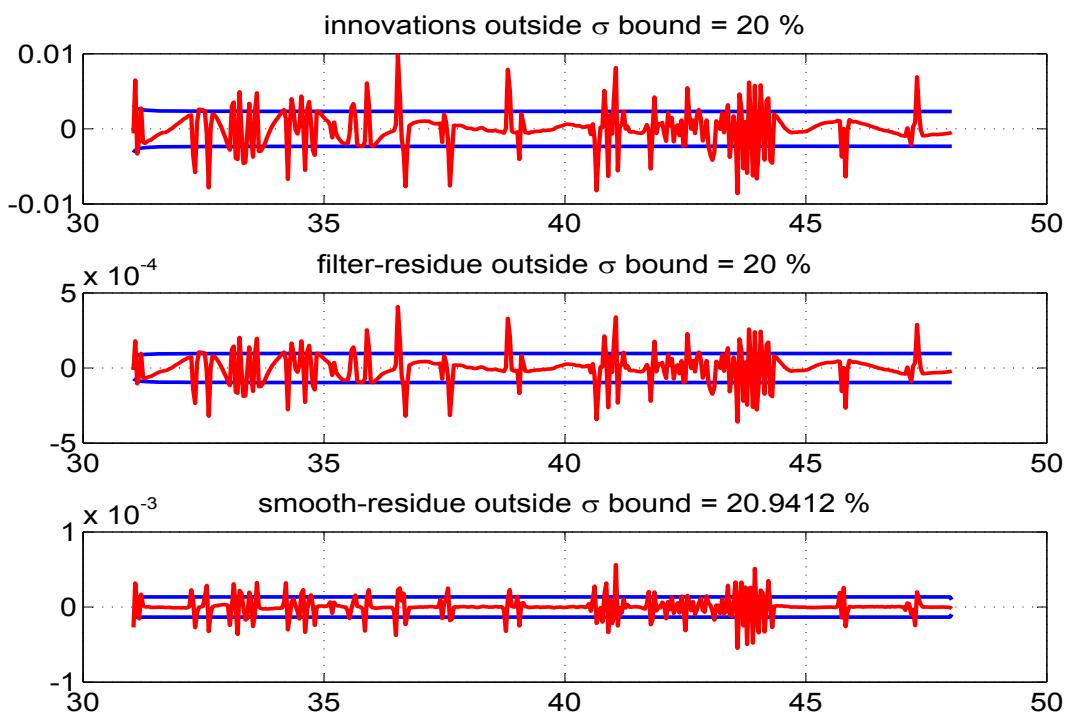
Figure 5.115: The innovations, filtered residue and smoothed residue of measurement 3
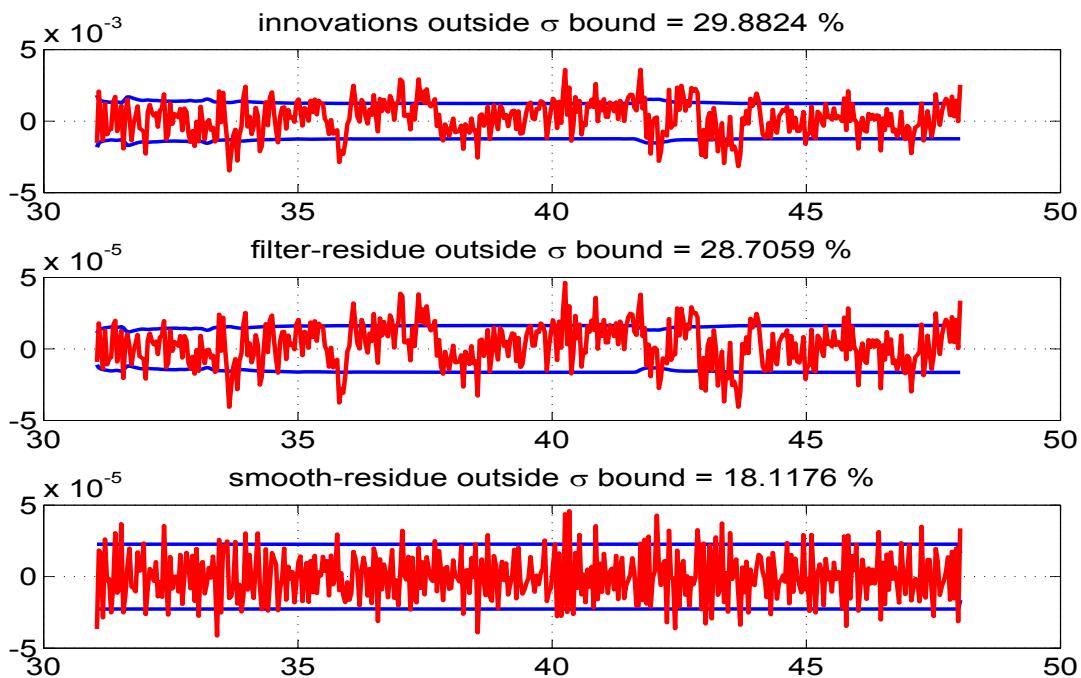
Figure 5.116: The innovations, filtered residue and smoothed residue of measurement 4
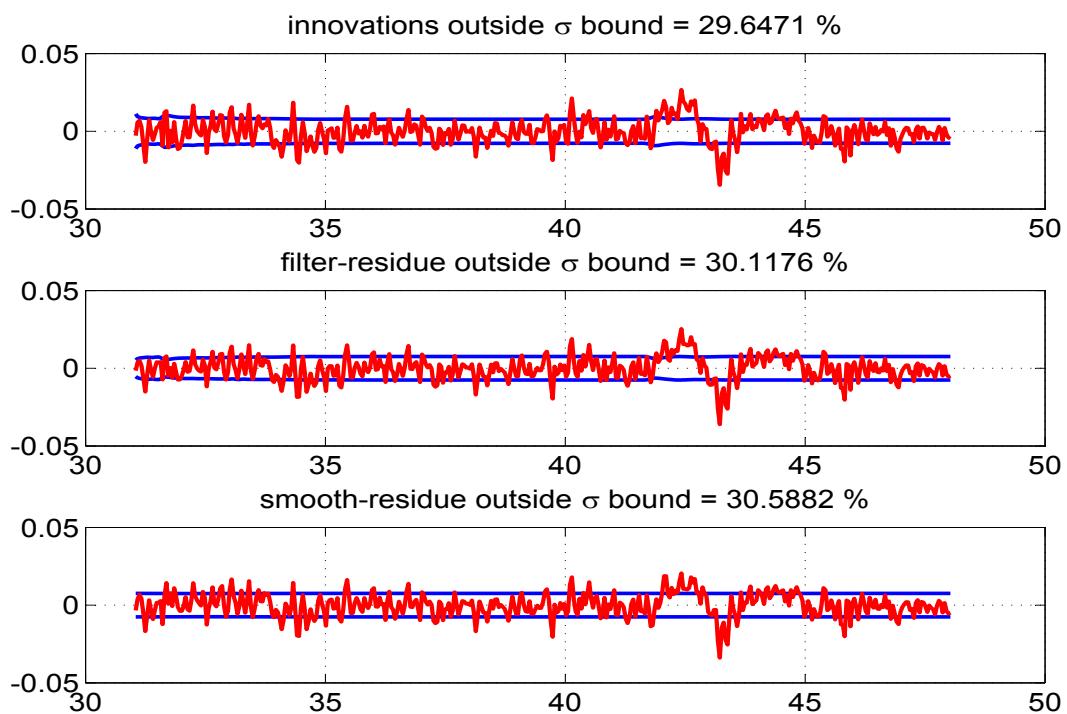
Figure 5.117: The innovations, filtered residue and smoothed residue of measurement 5
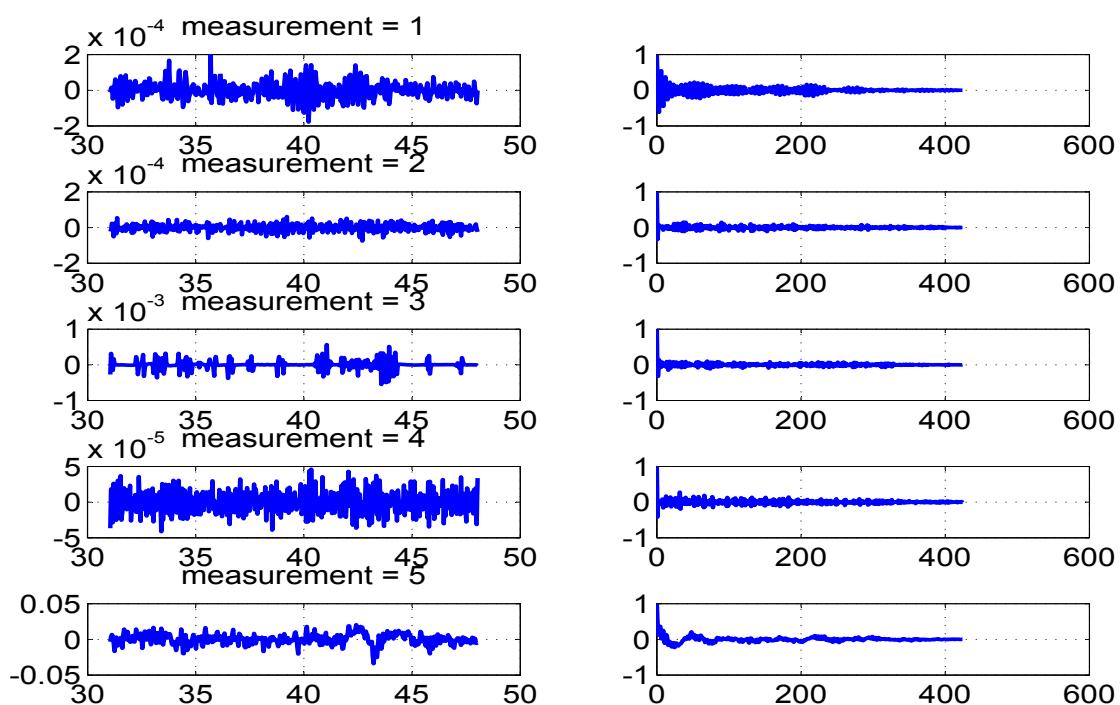
Figure 5.118: Time variation of estimated measurement noise (left) and

their autocorrelation (right)
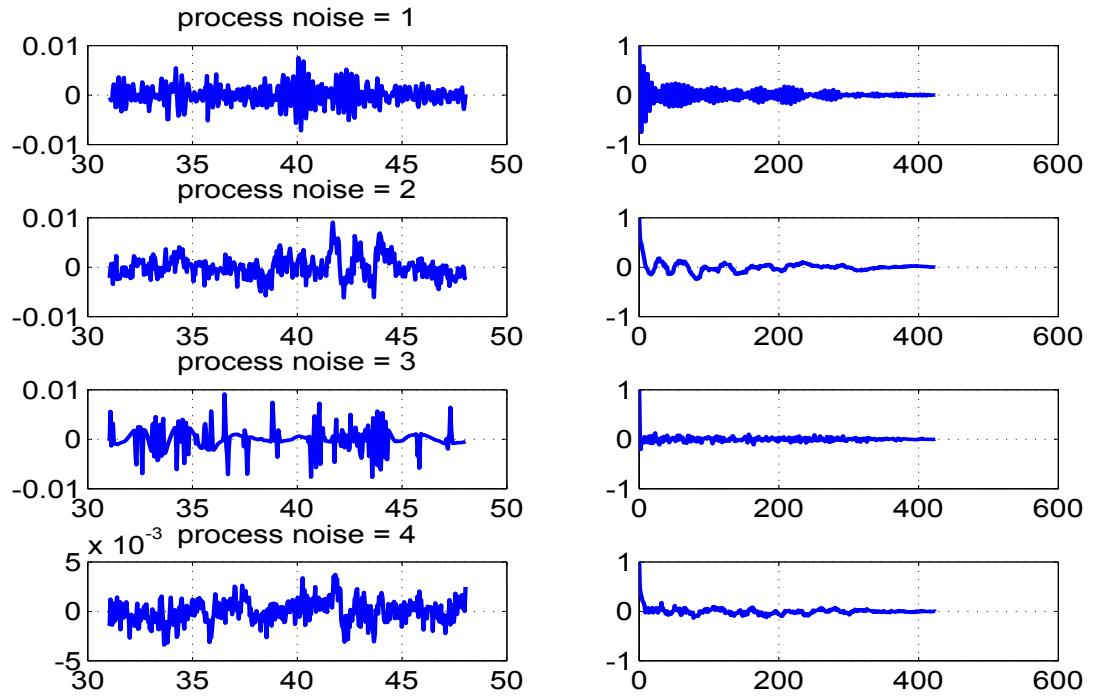
Figure 5.119: Time variation of estimated process noise (left) and

their autocorrelation (right)
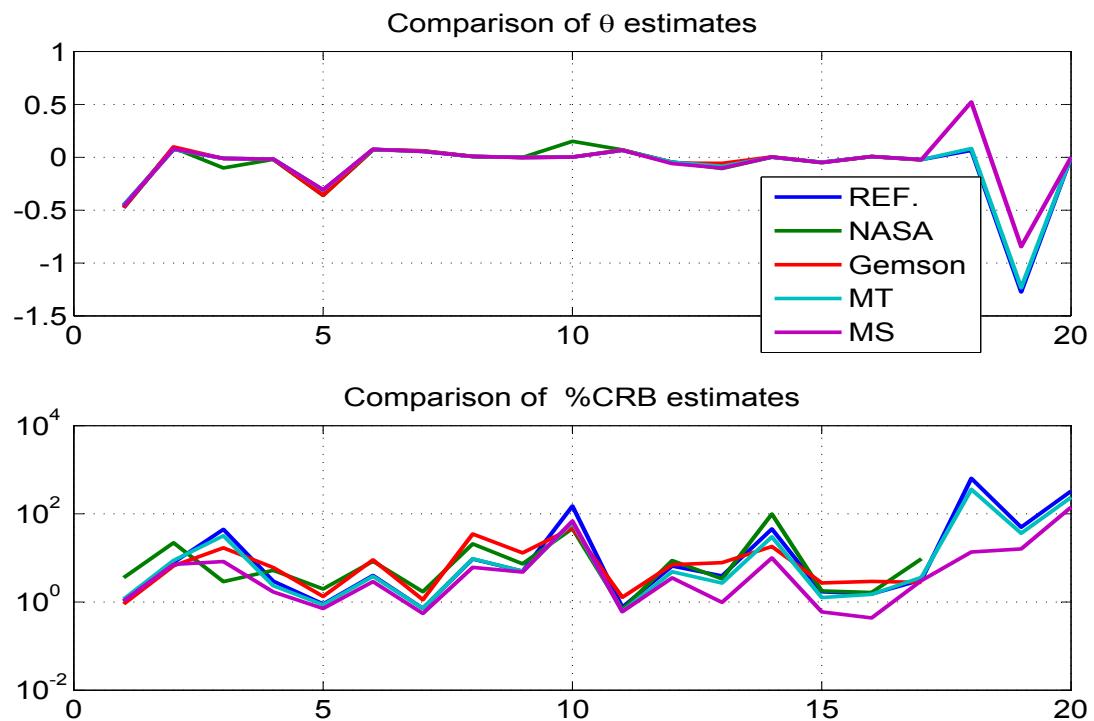
Figure 5.120: Comparison of the parameter estimates and %CRBs by different methods

Chapter 6

Conclusions and Suggestions for Further Work

One can now summarize the present study as follows,

1. A comparative study of the three different adaptive estimation technique is carried out, namely extended Expectation Maximisation (EM) suggested by Bavdekar et al.[104] (2011), Maximum Likelihood suggested by Mohamed and Schwarz[70] (1999), Covariance matching suggested by Myers and Tapley[34] (1976) and Gemson[55] (1991). A reference adaptive EKF procedure is proposed which is stable for a large range of initial guess values for the tuning parameters but sensitive to the measurement data.

2. In order to tune the filter and estimate $\mathbf{Q}, \mathbf{R}$ simultaneously, proper choice of their estimation 'statistics' is necessary.

3. The connection between the Newton Raphson (NR) results and the Kalman filter has been established for $\mathbf{Q} = 0$ case. The filter would give very close results to the NR optimization technique when the filter statistics ($\mathbf{P_0}$ and $\mathbf{R}$) are tuned properly.

4. The confidence in the filter results is best based on cost functions (**J1-J8**) and are summarised in Table-6.1 for $\mathbf{Q} > 0$ case.

5. A correct choice of $\mathbf{P_0}$ is necessary to achieve the proper CRB which can be estimated either by a simple heuristic Scaling up method, Inverse of the information matrix (IIM) or the smoothed $\mathbf{P_0}$ being scaled up.

6. A constant signal model with zero process noise is used for the unknown constant parameters rather than a random walk model. The augmented parameters learns recursively through different time and iterations with the right choice of $\mathbf{P_0}$.

7. The reference adaptive EKF achieves statistical equilibrium after few iterations and the mean value of the estimated unknown parameters and the noise statistics over many simulations tend to a stable value.

8. The reference adaptive EKF is robust to the change in initial conditions over a wide range. A thorough sensitivity study is conducted on simple and complex systems and the results are provided.

9. Real flight test data from NASA open source is processed using the proposed reference adaptive EKF and the results are compared with the existing techniques.

10. The estimated measurement and process noise may not be Additive White Gaussian Noise (AWGN) in a real case scenario which needs to be studied further.

11. It is useful to rework out the case studies, and real flight test data available in many of the text books, and references, and compare the results therein with the present method.

12. The estimated noise statistics refer to the complete data processed off line. In many applications when the system dynamics is very fast a rapid adjustment of **Q** becomes necessary. Based on some of the ideas in the present work it should be possible to tune **Q** rapidly on line.

13. In most variants of the Kalman filter implementation such as Unscented, Particle, and Ensemble filters generally researchers do not appear to be concerned about tuning the Kalman filter. The exploration of the tuning the statistics in such implementation in other fields of application would be interesting.

14. One should routinely tune off line using simulated data and then apply it for on line real and real time data analysis.

15. The present work has concentrated on tuning the filter statistics. The constant gain Kalman filter approach offers a much faster implementation for many real time applications to solve many involved problems using the generalised cost functions suggested in the present work.

Table 6.1: Cost convergence of different methods.

C-Converge, U/D-Under Perform/Diverge

Adaptive EKF	Constant	Ramp	SMD	Longitudinal	Lateral	Remarks
Reference	C	C	C	C	C	Cost functions converge to the expected value
MT	U/D	U/D	U/D	U/D	U/D	Converge only if $\mathbf{R}$ is known
MS	U/D	U/D	U/D	U/D	U/D	Converge only if $\mathbf{R}$ is known
Gemson	C	C	C	C	C	Initial $\mathbf{R}$ should be close to true value

Finally a few suggested term paper topics are

- The Least Squares (LS) history and Gauss procedure.

- Bayesian and Frequentist approaches.

- Prior probabilities.

- Rework the earlier test case studies in books and references.

- Try to automate the evaluation of the partial derivatives of functions needed in EKF.

- Use of simple Euler integration for propagating the differential equations and see how the estimates and CRBs are affected.

- Tuning the other variants, combinations, and implementations of the Kalman filter.

- Try to fine tune the scaled up full $\mathbf{P_0}$ of the parameters for statistical equilibrium.

- Attempt if a combination of some arbitrary combination $\mathbf{P_0}$ and $\mathbf{Q}$ can lead to the same CRB even in the zero process noise case?

Appendix A

MMLE and Cramer Rao Bound

The following section explain the parameter estimation technique using modified Newton Raphson (NR) optimisation for zero process noise ($\mathbf{Q} = 0$) case. It also gives a bound on the estimated parameters which is a check for efficiency of any estimator.

A.1 Newton Raphson Optimisation and CRB

Method of maximum likelihood estimation (MMLE) is one of the fundamental method in estimation theory (ET) developed by Fisher[1] (1922). The likelihood function is given by

$$L(\Theta|Z) = p(Z|\Theta)$$

where p(.) denotes the joint probability density function. The multivariate Gaussian analysis gives us

$$L(\Theta|Z) = \frac{(2\pi)^{-N/2}}{|\mathbf{R}|^N} exp\left[-\frac{1}{2} \sum_{k=1}^{N} (Z_k - h(x_k, \Theta))^T \mathbf{R}^{-1} (Z_k - h(x_k, \Theta)) \right]$$

Maximizing the log likelihood function leads to the minimisation of cost function $\mathbf{J}$ given by

$$\mathbf{J}(\Theta) = \sum_{k=1}^{N} (Z_k - h(x_k, \Theta))^T R^{-1} (Z_k - h(x_k, \Theta))$$

The above function is minimized for Θ using the well known modified Newton Raphson iterative procedure,

$$\Theta^{i+1} = \Theta^i - (D^2 J^i)^{-1} D J^i \tag{A.1}$$

where 'i' is the iteration number. The first and second order gradients of **J** are respectively,

$$DJ^i = \sum_{k=1}^{N} \frac{\partial h(x_k, \Theta^i)}{\partial \Theta}^T R^{-1}(Z_k - h(x_k, \Theta^i))$$

$$D^2J^i \approx \sum_{k=1}^{N} \frac{\partial h(x_k, \Theta^i)}{\partial \Theta}^T R^{-1} \frac{\partial h(x_k, \Theta^i)}{\partial \Theta}$$

where the partial derivative $(\partial h/\partial \Theta)$ of size $m \times p$ can be found using finite forward differences. The gradients are evaluated using the true initial state and the initial parameter, Θ^1 is chosen within $\pm 5\%$ error. If the measurement noise **R** is unknown then it can be estimated as

$$\mathbf{R}^{i+1} = \frac{1}{N} \sum_{k=1}^{N} (Z_k - h(x_k, \Theta^i))(Z_k - h(x_k, \Theta^i))^T \tag{A.2}$$

The Cramer Rao Bound or the uncertainty in the parameter estimate, $\Theta^{i_{max}}$ is given by

$$CRB = (D^2 J^{i_{max}})^{-1} \tag{A.3}$$

where 'i_{max}' is the maximum iteration number and 'diag' is the MATLAB$^{\circledR}$ notation for the diagonal values of a square matrix. The iterations are stopped when there is no significant change (say $\gamma = 10^{-5}$) in the cost function **J**,

$$\frac{|\mathbf{J}^i - \mathbf{J}^{i-1}|}{|\mathbf{J}^i|} < \gamma$$

As per Cramer Rao Bound criterion, for any estimator $P_{\Theta_{jj}} \geq CRB_{jj}$ for j^{th} parameter. If the estimator gives the CRB ratio $\frac{CRB_{jj}}{P_{\Theta_{jj}}} \approx 1$ then it is said to be an efficient estimator.

A.2 Posterior Cramer Rao Bound (PCRB)

A recursive equation for the lower bound on the discrete state with process noise was given by Tichavsky[68] (1998) based on Van Trees version of Cramer Rao inequality which is termed as Posterior Cramer Rao Bound (PCRB) and is given by

$$PCRB_k = D_k^{22} - D_k^{21}(PCRB_{k-1} + D_k^{11})^{-1}D_k^{12} \tag{A.4}$$

where

$$D_k^{11} = E[f'^T(x_{k-1},\Theta)\mathbf{Q}^{-1}f'(x_{k-1},\Theta)]$$

$$D_k^{22} = \mathbf{Q}^{-1} + E[h'^T(x_k,\Theta)\mathbf{R}^{-1}h'(x_k,\Theta)]$$

$$D_k^{12} = [D_k^{21}]^T = E[f'^T(x_{k-1},\Theta)]\mathbf{Q}^{-1}$$

where $f'(x_{k-1},\Theta) = \left[\frac{\partial f}{\partial x}\right]_{x=x_{k-1}}$ & $h'(x_k,\theta) = \left[\frac{\partial h}{\partial x}\right]_{x=x_k}$ are the Jacobians evaluated at true state and parameter values. The expectation operator used in calculating D_k's can be replaced by the average values taken over many simulations (say 50). The diagonal values of the PCRB evaluated at the last time instant ($PCRB_N$) is used as reference check for the state covariance (P_x) obtained from $P_{N|N} = [P_x, \checkmark; \checkmark, \checkmark]$ for $\mathbf{Q} > 0$ case.

Appendix B

Simple Modeling and Testing of the Estimates

Consider a second order polynomial with about 100 points added at each point by a white Gaussian noise. Now, instead of choosing a second order polynomial let us say one chooses a lower order 1, appropriate order 2, higher orders 5 and 25. Then using a cost function **J** based on a minimum mean square error it is possible to estimate all the unknown parameters in the appropriate order polynomial.

The Fig. B.1-B.8 shows a very simple example of a second order polynomial using a MATLAB$^{®}$ least squares program (shown below) illustrating the Output Error Method. It has no fancy numerical solution of differential equations and the use of sophisticated optimization techniques! If this is studied and further experiments are conducted then you have understood almost everything except for the CRB about the OEM.

We consider the following namely (i) A noisy second order polynomial with the fit based on the estimated parameters and the measurements, (ii) a typical normal probability distribution of the difference in the previous items, (iii) a typical autocorrelation of the above difference, and then based on 100 similar data sets the (iv) the normalized cost, (v) the distribution of the coefficients, and (vi) the histogram of the estimated fit coefficients. The first order fit fails the autocorrelation function (ACF) test, and the expected cost behaviour. The second order fit passes ACF test and all the other quantities follow their expected behaviour. The fifth order fit gives no clue about its behaviour except some small wiggles appear in the fit. The twenty fifth order fit is also similar to the fifth but the wiggles are more pronounced in its fit with the second order polynomial. The correlation coefficient matrix (C) of the estimated parameters for 100 simulations for different order fit is shown below. There is a very high positive and negative correlation among the coefficients for higher order fits. The reasonable correlation coefficient among the fit parameters show the

appropriateness of the fit of order 2 and the over fitting of the higher order ones. Ideally for the correct order the correlation coefficient matrix should be an identity matrix. It is thus clear just one or two test quantities are not enough but more will have to be formed to avoid deceptive and reach decisive conclusions.

The rounded off $100 \times C$ matrix for order = 1 is given by

$$\begin{bmatrix} 100 & -83 \\ -83 & 100 \end{bmatrix}$$

The rounded off $100 \times C$ matrix for order = 2 is given by

$$\begin{bmatrix} 100 & -96 & 72 \\ -96 & 100 & -86 \\ 72 & -86 & 100 \end{bmatrix}$$

The rounded off $100 \times C$ matrix for order = 5 is given by

$$\begin{bmatrix} 100 & -100 & 98 & -95 & 89 & -63 \\ -100 & 100 & -100 & 97 & -91 & 66 \\ 98 & -100 & 100 & -99 & 94 & -70 \\ -95 & 97 & -99 & 100 & -98 & 76 \\ 89 & -91 & 94 & -98 & 100 & -86 \\ -63 & 66 & -70 & 76 & -86 & 100 \end{bmatrix}$$

The rounded off 100×C matrix for order = 25 is given by

$$
\begin{bmatrix}
100 & -100 & 100 & -99 & 99 & -97 & 93 & -87 & 74 & -54 & 29 & -6 & -11 & 21 & -27 & 30 & -30 & 29 & -26 & 22 & -17 & 12 & -7 & 7 & -30 & 69 \\
-100 & 100 & -100 & 100 & -99 & 97 & -94 & 88 & -76 & 56 & -31 & 8 & 9 & -20 & 26 & -29 & 29 & -28 & 25 & -21 & 16 & -11 & 7 & -6 & 30 & -70 \\
100 & -100 & 100 & -100 & 99 & -98 & 95 & -89 & 78 & -58 & 34 & -11 & -6 & 17 & -24 & 27 & -27 & 26 & -23 & 19 & -14 & 9 & -5 & 5 & -29 & 72 \\
-99 & 100 & -100 & 100 & -100 & 99 & -97 & 91 & -80 & 62 & -38 & 15 & 2 & -14 & 20 & -23 & 24 & -23 & 20 & -17 & 12 & -7 & 3 & -3 & 28 & -75 \\
99 & -99 & 99 & -100 & 100 & -100 & 98 & -94 & 84 & -66 & 43 & -20 & 3 & 8 & -15 & 19 & -20 & 19 & -16 & 13 & -8 & 4 & 0 & 0 & -27 & 79 \\
-97 & 97 & -98 & 99 & -100 & 100 & -99 & 96 & -88 & 72 & -50 & 28 & -11 & -1 & 8 & -12 & 13 & -12 & 10 & -7 & 3 & 2 & -5 & 4 & 24 & -83 \\
93 & -94 & 95 & -97 & 98 & -99 & 100 & -99 & 93 & -79 & 59 & -39 & 22 & -10 & 2 & 2 & -3 & 3 & -1 & -2 & 5 & -9 & 13 & -11 & -20 & 88 \\
-87 & 88 & -89 & 91 & -94 & 96 & -99 & 100 & -98 & 88 & -72 & 53 & -37 & 25 & -18 & 13 & -11 & 11 & -12 & 14 & -18 & 21 & -23 & 21 & 13 & -93 \\
74 & -76 & 78 & -80 & 84 & -88 & 93 & -98 & 100 & -96 & 85 & -70 & 56 & -45 & 38 & -33 & 31 & -30 & 31 & -32 & 34 & -36 & 38 & -34 & -2 & 96 \\
-54 & 56 & -58 & 62 & -66 & 72 & -79 & 88 & -96 & 100 & -96 & 86 & -76 & 67 & -61 & 56 & -54 & 52 & -52 & 52 & -53 & 54 & -54 & 49 & -13 & -93 \\
29 & -31 & 34 & -38 & 43 & -50 & 59 & -72 & 85 & -96 & 100 & -97 & 91 & -85 & 80 & -76 & 74 & -72 & 71 & -70 & 70 & -69 & 68 & -62 & 28 & 83 \\
-6 & 8 & -11 & 15 & -20 & 28 & -39 & 53 & -70 & 86 & -97 & 100 & -98 & 95 & -92 & 89 & -87 & 85 & -84 & 82 & -81 & 79 & -76 & 70 & -40 & -69 \\
-11 & 9 & -6 & 2 & 3 & -11 & 22 & -37 & 56 & -76 & 91 & -98 & 100 & -99 & 97 & -96 & 94 & -93 & 91 & -89 & 87 & -84 & 81 & -75 & 49 & 56 \\
21 & -20 & 17 & -14 & 8 & -1 & -10 & 25 & -45 & 67 & -85 & 95 & -99 & 100 & -100 & 99 & -97 & 96 & -95 & 93 & -90 & 87 & -83 & 77 & -55 & -45 \\
-27 & 26 & -24 & 20 & -15 & 8 & 2 & -18 & 38 & -61 & 80 & -92 & 97 & -100 & 100 & -100 & 99 & -98 & 97 & -95 & 92 & -89 & 85 & -79 & 59 & 38 \\
30 & -29 & 27 & -23 & 19 & -12 & 2 & 13 & -33 & 56 & -76 & 89 & -96 & 99 & -100 & 100 & -100 & 99 & -98 & 96 & -94 & 91 & -87 & 82 & -63 & -34 \\
-30 & 29 & -27 & 24 & -20 & 13 & -3 & -11 & 31 & -54 & 74 & -87 & 94 & -97 & 99 & -100 & 100 & -100 & 99 & -98 & 96 & -93 & 89 & -84 & 66 & 32 \\
29 & -28 & 26 & -23 & 19 & -12 & 3 & 11 & -30 & 52 & -72 & 85 & -93 & 96 & -98 & 99 & -100 & 100 & -100 & 99 & -97 & 95 & -92 & 87 & -69 & -31 \\
-26 & 25 & -23 & 20 & -16 & 10 & -1 & -12 & 31 & -52 & 71 & -84 & 91 & -95 & 97 & -98 & 99 & -100 & 100 & -100 & 99 & -97 & 94 & -89 & 72 & 31 \\
22 & -21 & 19 & -17 & 13 & -7 & -2 & 14 & -32 & 52 & -70 & 82 & -89 & 93 & -95 & 96 & -98 & 99 & -100 & 100 & -100 & 98 & -96 & 92 & -75 & -32 \\
-17 & 16 & -14 & 12 & -8 & 3 & 5 & -18 & 34 & -53 & 70 & -81 & 87 & -90 & 92 & -94 & 96 & -97 & 99 & -100 & 100 & -100 & 98 & -95 & 78 & 34 \\
12 & -11 & 9 & -7 & 4 & 2 & -9 & 21 & -36 & 54 & -69 & 79 & -84 & 87 & -89 & 91 & -93 & 95 & -97 & 98 & -100 & 100 & -99 & 97 & -81 & -35 \\
-7 & 7 & -5 & 3 & 0 & -5 & 13 & -23 & 38 & -54 & 68 & -76 & 81 & -83 & 85 & -87 & 89 & -92 & 94 & -96 & 98 & -99 & 100 & -99 & 84 & 35 \\
7 & -6 & 5 & -3 & 0 & 4 & -11 & 21 & -34 & 49 & -62 & 70 & -75 & 77 & -79 & 82 & -84 & 87 & -89 & 92 & -95 & 97 & -99 & 100 & -90 & -30 \\
-30 & 30 & -29 & 28 & -27 & 24 & -20 & 13 & -2 & -13 & 28 & -40 & 49 & -55 & 59 & -63 & 66 & -69 & 72 & -75 & 78 & -81 & 84 & -90 & 100 & -10 \\
69 & -70 & 72 & -75 & 79 & -83 & 88 & -93 & 96 & -93 & 83 & -69 & 56 & -45 & 38 & -34 & 32 & -31 & 31 & -32 & 34 & -35 & 35 & -30 & -10 & 100 \\
\end{bmatrix}
$$

Source Code for Polynomial Fit :

```
clc;clear all; close all;

nstd = 0.05;
nens = 100;
i = 2;% fit order
x = 0:0.01:1;
N=length(x);
y = 4.*(x - (x.*x)); % second order polynomial

for nr = 1:nens;
n = nstd* randn(1,N);
z=y+n;
[p,s] = polyfit(x,z,i);
es = polyval(p,x);
sp(nr,:) = p;
cost(nr) = (sum((es-z).*(es-z)))/(N*nstd*nstd);
end;

[rk,k]=xcorr(es-z,'coeff');

figure(1);subplot(221);plot(x,y,'-b',x,y+n,'og',x,es,'-r','linewidth',2);grid;
xlabel('Variable','fontsize',14); ylabel('Function','fontsize',14);
set(gca,'fontsize',14),subplot(223);
plot(0:round(N/4),rk(intersect(find(k>=0),find (k<=round(N/4)))),'linewidth',2),
xlabel('Lag','fontsize',14); ylabel('ACF','fontsize',14);grid;
set(gca,'fontsize',14)

figure(1);subplot(222);normplot(es-z);grid on;
xlabel('Residue','fontsize',14);
 set(gca,'fontsize',14)
subplot(224); plot(cost,'-','linewidth',2);grid
xlabel('Simulation Number','fontsize',14); ylabel('COST','fontsize',14);
 set(gca,'fontsize',14)

figure(2);  subplot(211); semilogy(abs(sp),'linewidth',2);grid;
xlabel('Simulation Number','fontsize',14); ylabel('Coeffs','fontsize',14);
subplot(212);hist(sp,50);grid;
xlabel('Estimated Coefficients','fontsize',14); ylabel('Frequency','fontsize',14);

avgcost= mean(cost),stdvcost = std(cost)
avg = mean(sp),stdv = std(sp), round(100*corrcoef(sp))
```
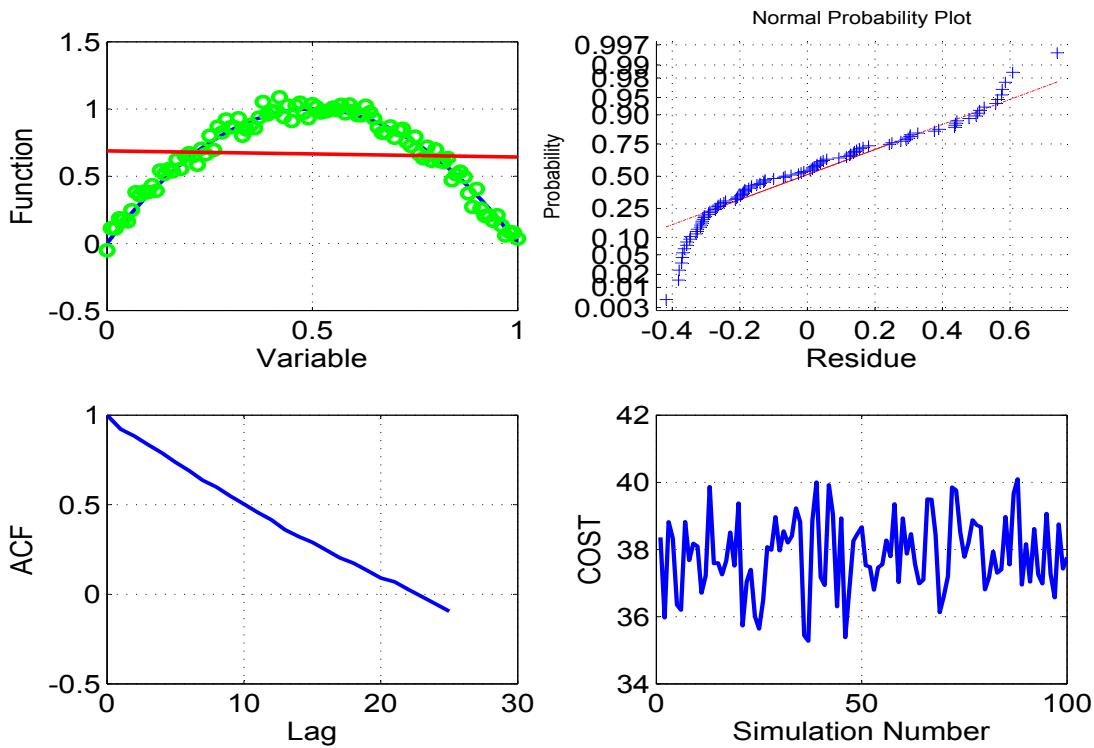
Figure B.1: Second order Polynomial : Fit Order = 1
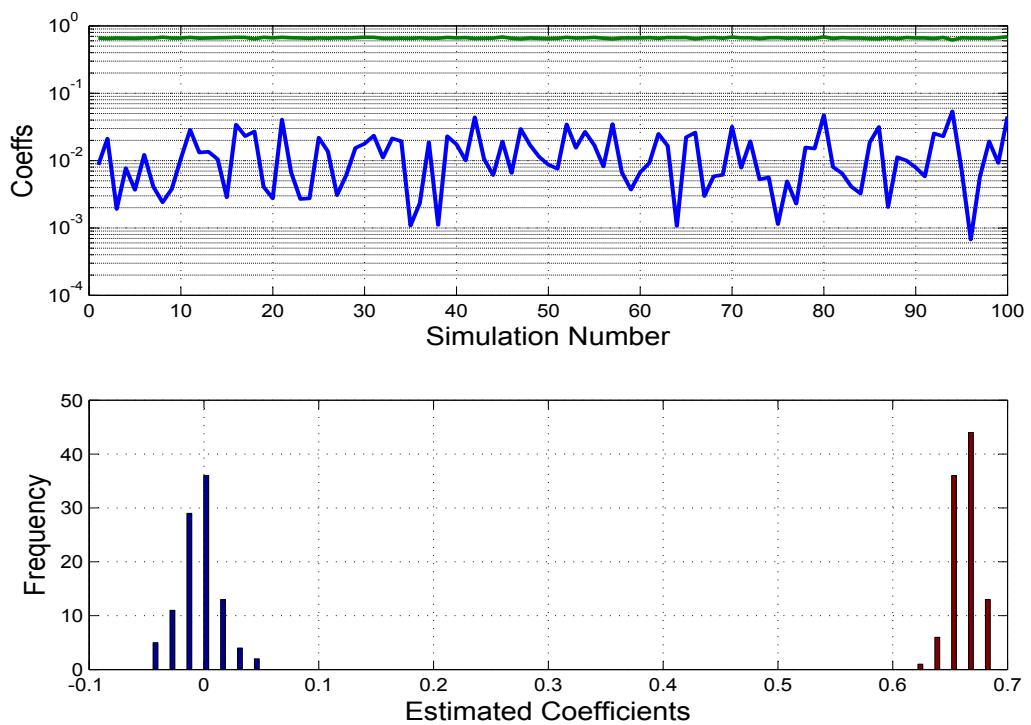
Figure B.2: Second order Polynomial : Fit Order = 1

Coefficient estimates and their Histogram over 100 ensembles
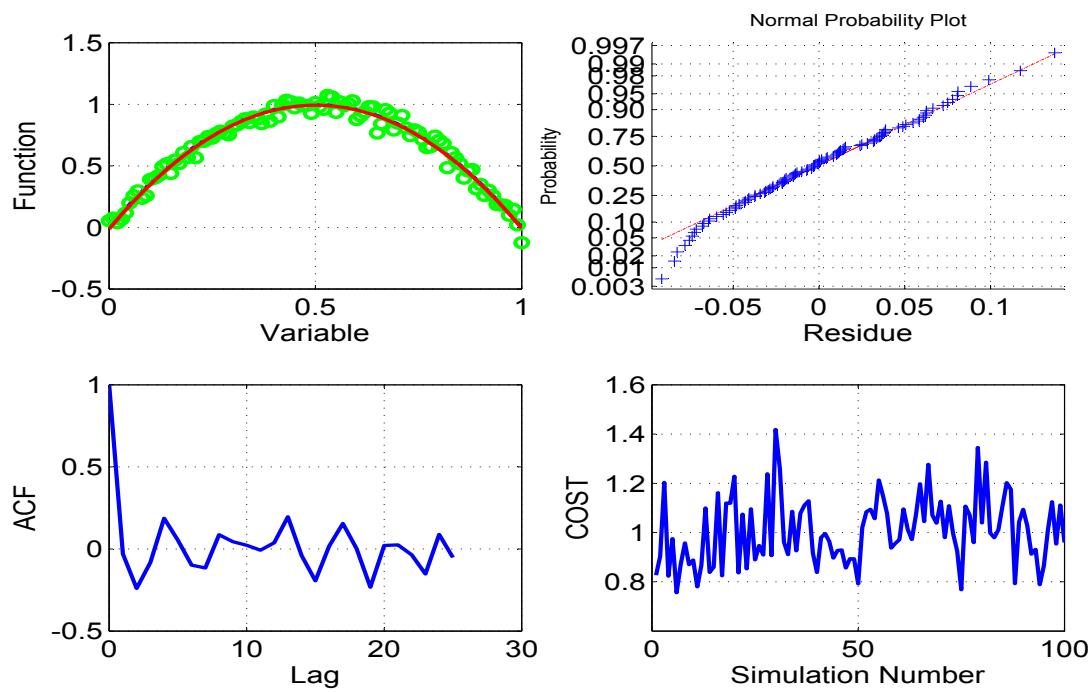
Figure B.3: Second order Polynomial : Fit Order = 2
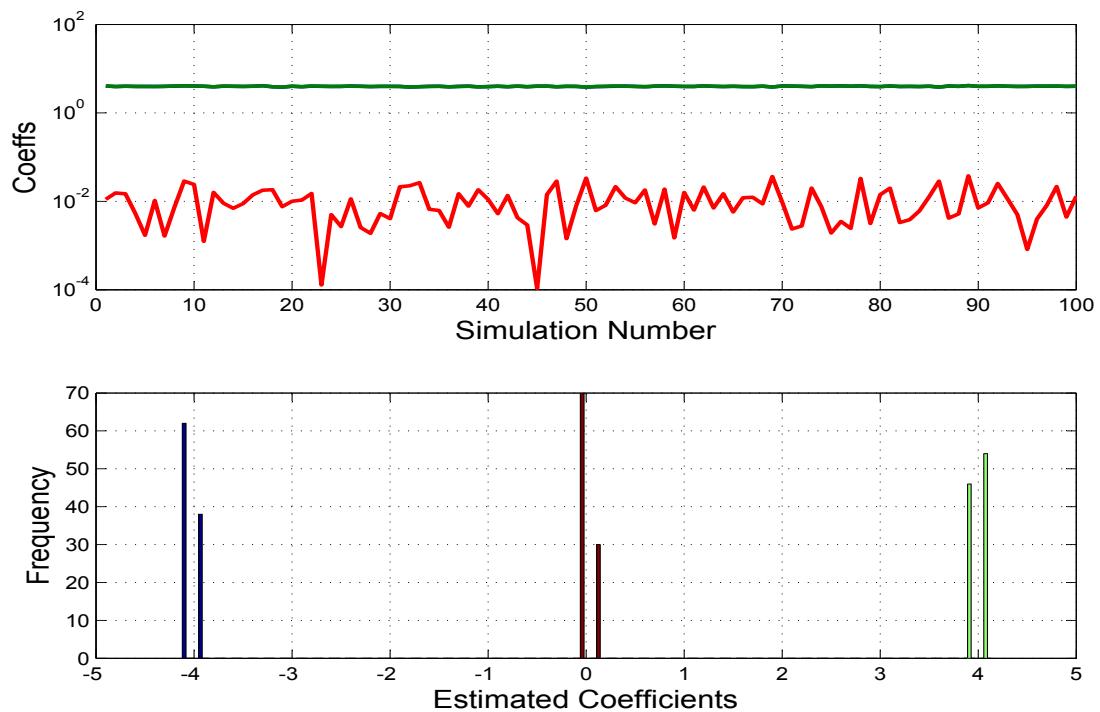
Figure B.4: Second order Polynomial : Fit Order = 2

Coefficient estimates and their Histogram over 100 ensembles
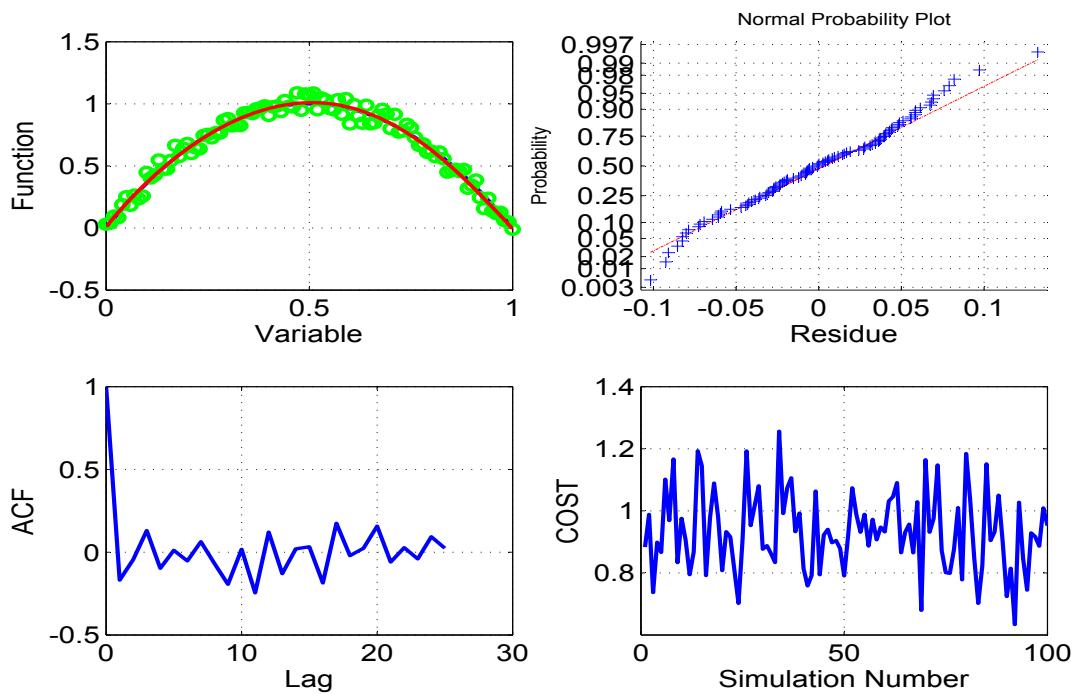
Figure B.5: Second order Polynomial : Fit Order = 5
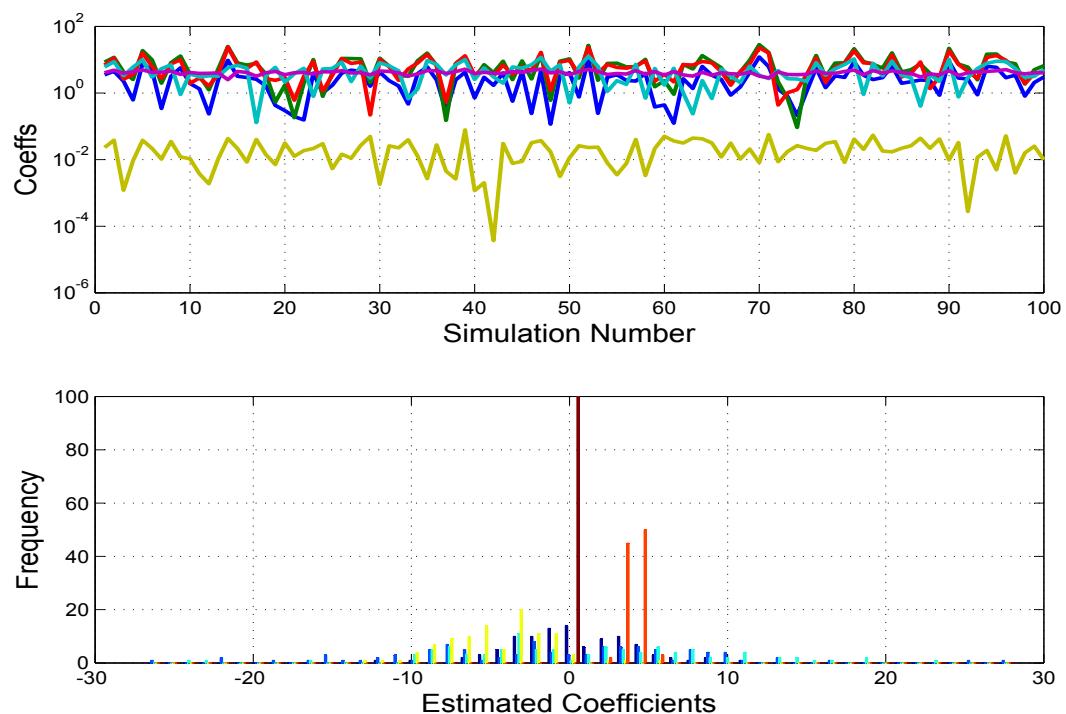
Figure B.6: Second order Polynomial : Fit Order = 5

Coefficient estimates and their Histogram over 100 ensembles
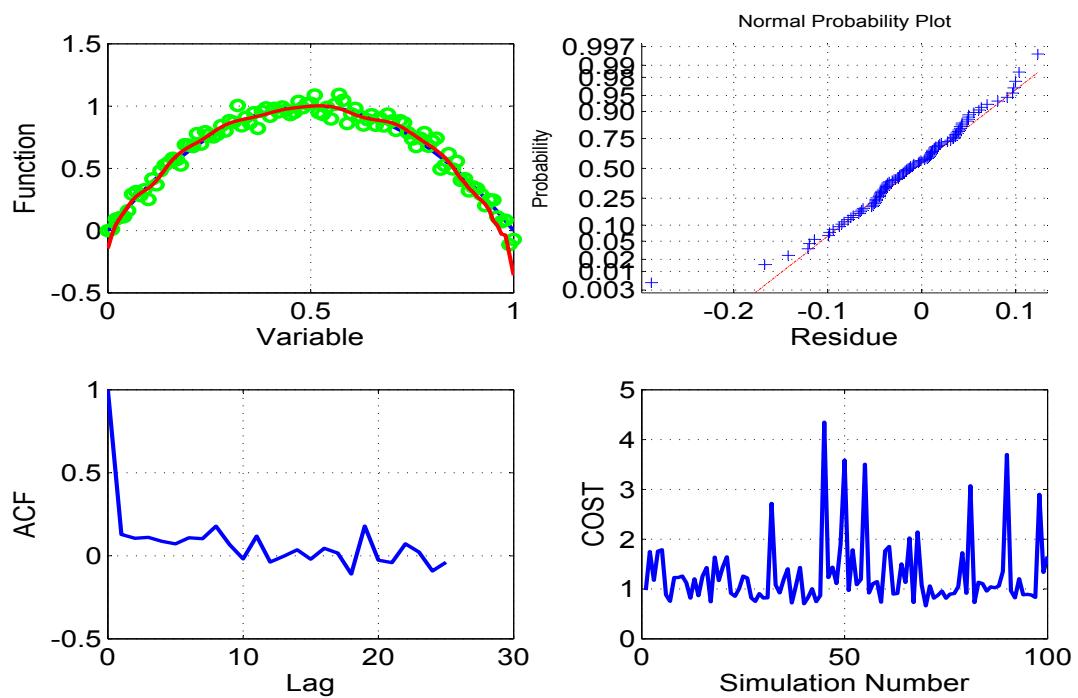
Figure B.7: Second order Polynomial : Fit Order = 25
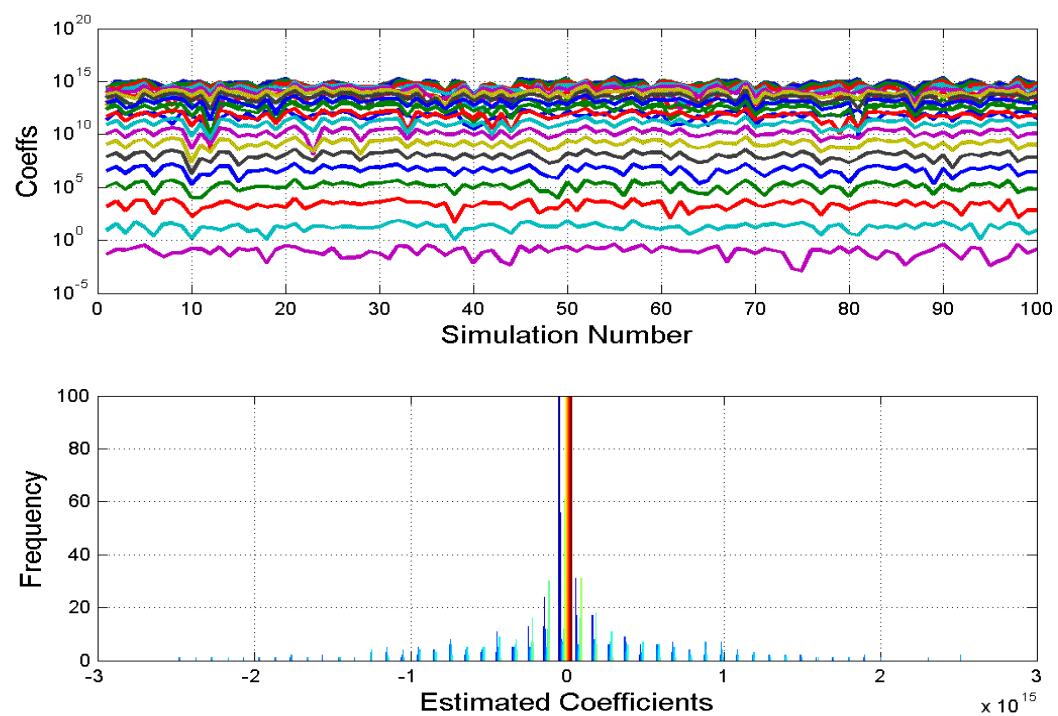
Figure B.8: Second order Polynomial : Fit Order = 25

Coefficient estimates and their Histogram over 100 ensembles

Appendix C

Filtering of a Noisy Constant Signal with and without P_0 Scaling

Consider a data of N = 100 data points obtained from a constant signal of value zero added with noise. This data is processed by known value of the measurement noise variance (**R**=0.25). The standard filter steps are used and the filter iterates over the data many times. The value of initial $\mathbf{X_0}$ and $\mathbf{P_0}$ is varied over a large range of respectively (-10:10:10) and (-10:5:10). The number of iterations was set at 10. The filter uses the estimated parameter value of X at the end of each pass for the next pass through the data. The final covariance at the end of each pass can be scaled up by the number of data points to be used as the initial covariance for the next pass or not. This makes the difference in the final estimates of the parameter value X and its uncertainty P (= CRB).

The Fig. C.1 shows if there is no scaling then starting from very low values of $\mathbf{P_0}$ the parameter is not updated much even if the initial X value is close to the true value of zero. For higher and very high initial $\mathbf{P_0}$ the parameter estimates get updated and reaches close to the true value indicated by the red circled point (showing the mean and variance/N of the data using the simple formulae in statistics) during the first few iterations. Subsequently though the parameter estimate X does not change but the P value keeps decreasing continuously which is not acceptable.

However if there is scaling then starting from even very low to very high values of P0 the initial parameter $\mathbf{X_0}$ gets updated as shown in Fig. C.2. Then commencing from any initial ($\mathbf{X_0}$, $\mathbf{P_0}$) the filter after some iterations reaches a statistically steady state trajectory (X, P) between the initial and the final states as it processes each one of the data point. This feature can be seen by expanding the figure around this region. The above simple

experiment shows the importance P_0 to be scaled after every filter pass. This feature is taken over even for more involved problems.

Effect of a Q pulse during filter pass in lieu of $P_0 = 0$

This experiment demonstrates that even if $P_0 = 0$ it can be substituted by injecting Q during the filter passes. The experiment was carried out from an initial (X_0, P_0) anywhere in the range say $(-10:10, 10^{-30} : 10^{15})$. Starting from $(-10, 10^{-30})$ a very small $Q = 10^{-4}$ was injected as a single impulse before processing the second data point in every pass and nowhere else. The filter estimates in Fig. C.3 shows no convergence towards the true value even after about 10 iterations but with 100 iterations the estimate was close but the covariance was lower than the true value. When Q was increased to 10^{-2} then in about 10 passes the filter reached an estimate close to the true value as shown in Fig. C.4. The reason being when a large Q is injected at any point the filter ignores the state estimate but believes the then measurement. Thus the filter estimate drops from a far off value to be within the measurement error band beyond which it improves the estimate by assimilating the subsequent measurement data.

An analogy from human life can be given though may not be quite perfect. The initial condition (X_0, P_0) represents the random conditions at birth. The sample measurements Z based on R are varied individual experiences in life. Based on the experience the goal is to realize the true state represented by the parameter value zero or sunya. The filter iterations represent rebirths. The injection of Q continuously or a few times denotes opportunities to learn. The state improves in each birth. If one is far away like X_0 (= -10) and confident that he knows the ultimate truth (little $P_0 = 10^{-30}$) with very low Q (= 10^{-4}) it would take far too many births to attain realization. The injection of a small $Q = 10^{-2}$ helps to reach the goal in fewer births. However if a very large Q is injected in one's life at some point then the final goal is reached in that birth itself. This is what happened to Buddha after which he meditated for many years and obtained realization. But how and from where does this large Q come about? It is God's Grace for one who has extraordinary compassion and humility. One can also read the analogy between the 'State Estimation' and 'The meaning of Life' in Dan Simon[92] (2006).

Source code showing the importance of P$_0$ Scaling :

```matlab
clc;close all;clear all;

rng('default');
N=100;%No of data samples
R=0.25;

X0_initial=-10:10:10;
P0_power=-10:5:10;

iter_max=10;
P0_iter_max=length(P0_power);
X0_iter_max=length(X0_initial);

Scale_up=N;
index =-2;
Q=0*10^index;

z=sqrt(R)*randn(N,1)+ 0;
X_red=mean(z);P_red=std(z)^2/N;

StoreX0=[];StoreX=[];StoreP0=[];StoreP=[];
x_prior=zeros(N,1);x_posterior=zeros(N,1);
p_prior=zeros(N,1);p_posterior=zeros(N,1);

for X0_iter=1:X0_iter_max % outer iteration loop for a varying P0_initial

    for P0_iter=1:P0_iter_max % outer iteration loop for a varying P0

        for iter=1:iter_max %inner iteration loop for using estimated x0 and P0

            if iter==1
            X0=X0_initial(X0_iter);
            P0=10^P0_power(P0_iter);
            else
            X0=X0_est;
            P0=P0_est;
            end

            display([X0_iter P0_iter iter],'x0_iter P0_iter iter');
            display([X0 P0],'(x0,P0)')
```

```matlab
for k=1:N %start of data processing loop

if k==1
xk_prev=X0;pk_prev=P0;
else
xk_prev=x_posterior(k-1);pk_prev=p_posterior(k-1);
end

%Kalman equations
x_prior(k)=xk_prev;
if(k==2 ), QA = Q; else QA=0.0;end;
p_prior(k) = pk_prev+QA;
innov_cov=p_prior(k)+R;
K(k)=p_prior(k)/(innov_cov);
r(k)= z(k)-x_prior(k);
p_posterior(k)=(1-K(k))*p_prior(k);
x_posterior(k)=x_prior(k)+K(k)*(r(k));

end  %end of data processing loop

P0_est=Scale_up*p_posterior(N);
X0_est=x_posterior(N);

StoreX0=[StoreX0;X0];StoreX=[StoreX;X0;x_posterior];
StoreP0=[StoreP0;P0];StoreP=[StoreP;P0;p_posterior];
end %inner loop iter

StoreX=[StoreX;NaN];StoreP=[StoreP;NaN];

figure(1),semilogy(StoreX0,StoreP0,'ko',StoreX,StoreP,'b--','linewidth',2),
hold on, grid on,
end
end

if Scale_up==1,String='Without Scaling up';else String='With Scaling up';end

plot(X_red,P_red,'ro','MarkerFaceColor','r','MarkerSize',10)
title(String, 'fontsize',14),xlabel('X','fontsize',14),ylabel('P','fontsize',14)
set(gca, 'fontsize',14)

format long ,VAL=[Q, X_red,P_red, x_posterior(N), p_posterior(N)]
```
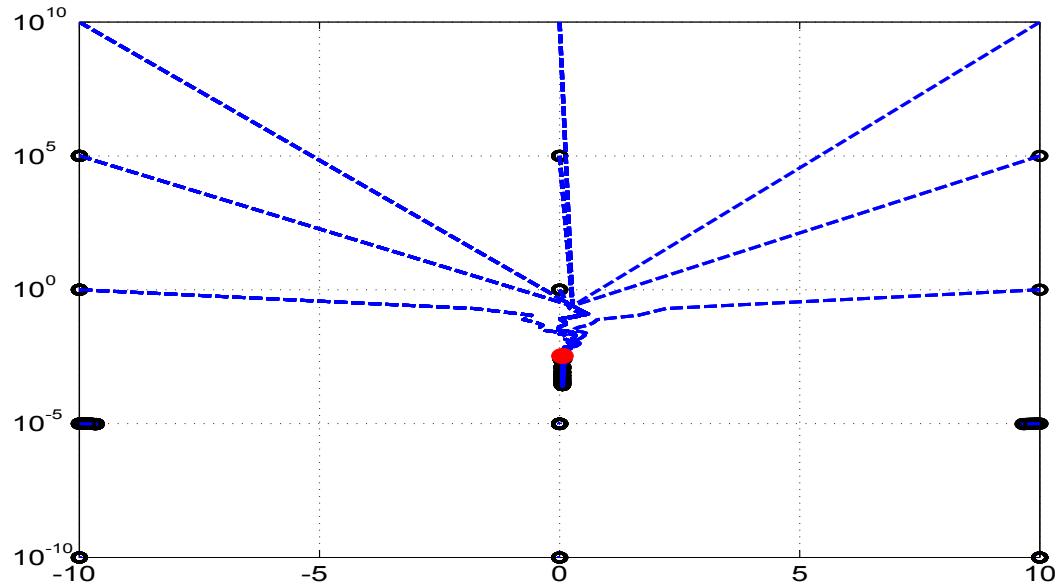
Figure C.1: Plot of P versus X without Scaling
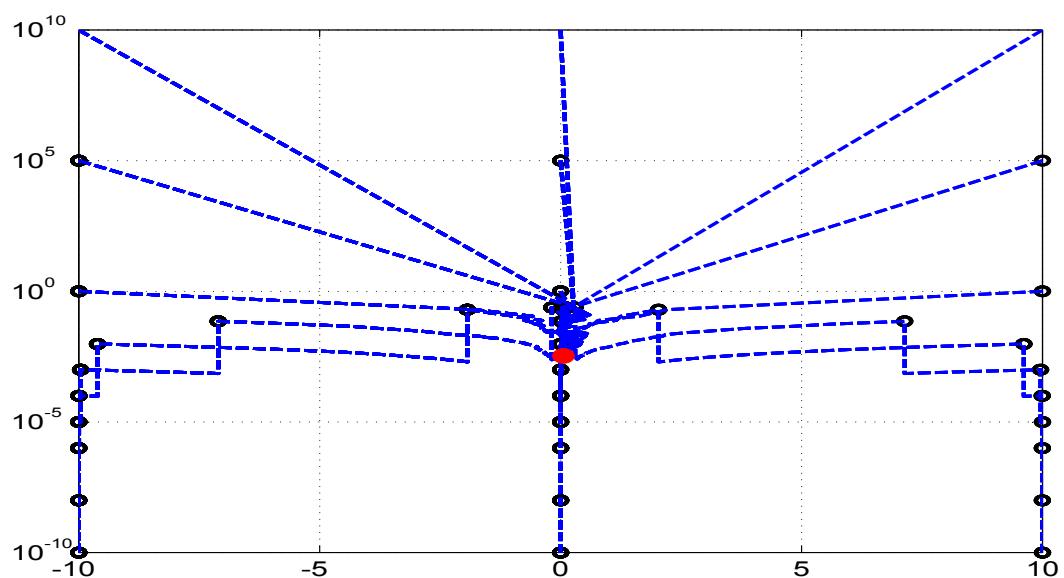
Figure C.2: Plot of P versus X with Scaling
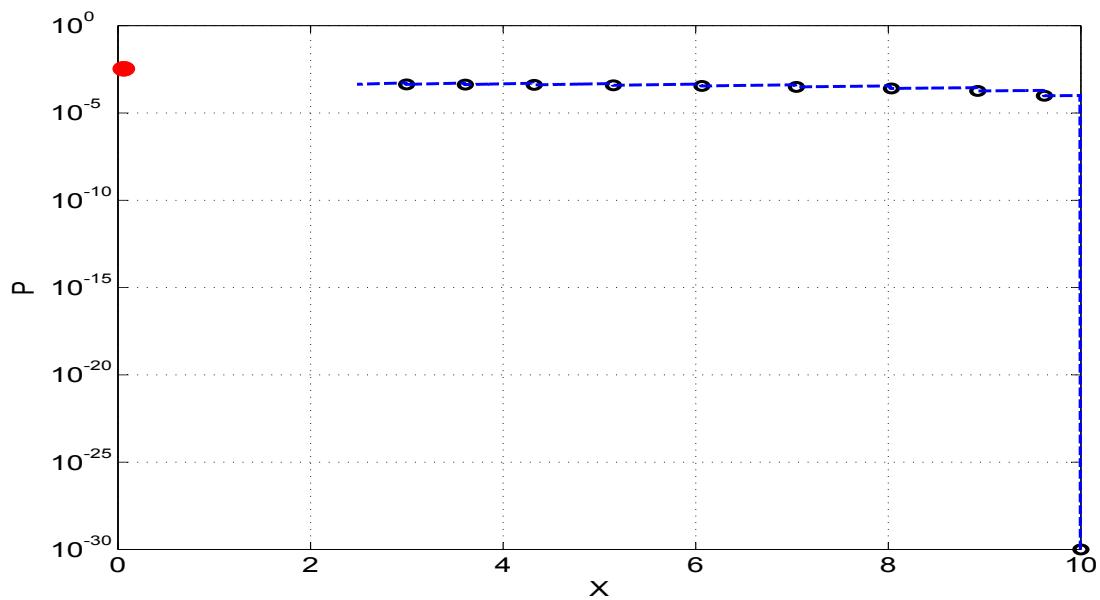
Figure C.3: Plot of P versus X with impulse $\mathbf{Q} = 10^{-4}$
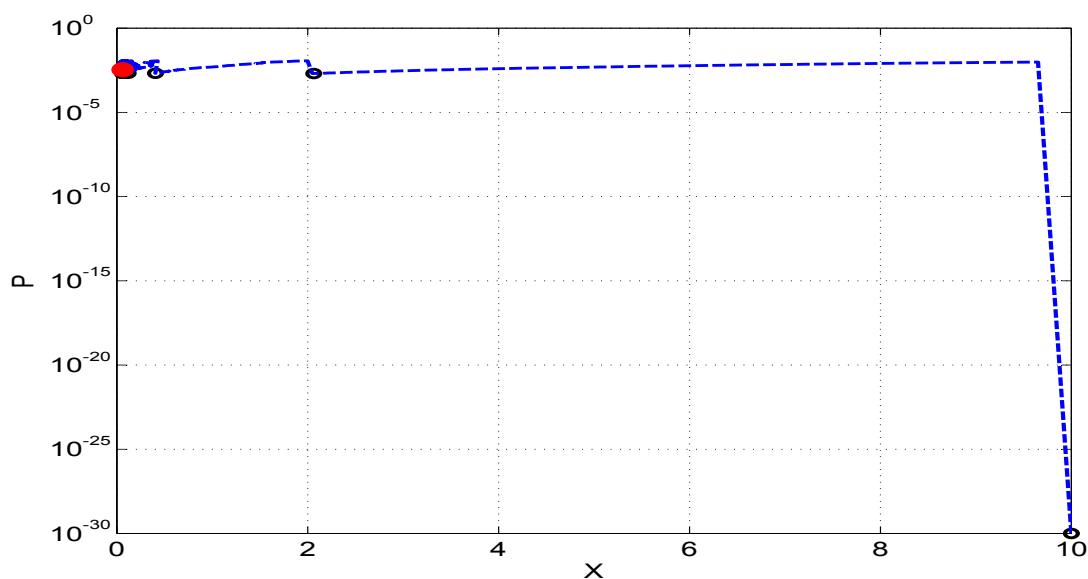
Figure C.4: Plot of P versus X with impulse $\mathbf{Q} = 10^{-2}$

Appendix D

Source Code used for the Simulated and Real Data Cases

In this section, the MATLAB$^{\circledR}$ source code (written in version 2012b) used in generating the results are printed for reference. There are two programs one corresponding to the simulated system case (D.1) and the other corresponding to the real data case (D.2). All the input options are provided in the first few lines of the program for the user to conduct sensitivity studies.

The soft copy of the source code and the data files are embedded below. One can right click on the below icons and select the option 'Save Embedded File to Disk' to save it in a desired folder.

D.1 Simulated System Program :

```
function EKF_REF_NEW(varargin)

tic;clc;clear all;close all;
global e_max total_sts R0_power m_noise p_noise theta N sys ns np nm dt t SOKF...
    X smooth_option q_true r_true Xd

%%1-Constant, 2-Ramp, 3-SMD, 4-SMD with weak theta,
%%5-Longitudinal, 6-Lateral.
sys=4;

%Initial values of P0=10^P0_power,R=2^R0_power,
%%Q= 10^Q0_power for the first iteration
P0_power=-1;
Q0_power=-1;
R0_power=-1;

iter_max=100;
e_max=50;

N=100;dt=0.1;
t = 0.0:dt:N*dt;
stop=0*10^-3;

%1-Scaleup,2-IIM,3-P0smoth,
P0_method=1;
%1-P0=cov(S P)=[0,0;0,NZ],2-diag,3-full,
P0_option=1;

%1-EM(Bavdekar=z-h(xs)+...),2-MS(z-h(x+)+...),3-MT(z-h(x-)-...),4-(z-h(xd))
R_method=1;
%1-EM(Bavdekar(xs(k)-f(xs(k-1)+...),2-MS(K(z-h(x-)(z-h(x-)'K'),
%%3-MT(x(+)-x(-)+...),4-DSDT(xs(k)-xd(k)+...),
Q_method=1;
%1-Q=cov(S,P)=[NZ,0;0,0],2-diag,3-full
Q_option=1;
%1-diag,2-full
R_option=1;

R2term=1;
Q2term=1;

%Use of last few samples
last_n=0.5*N;
Scale_up=1*N;
smooth_option=1;
```

```matlab
if sys==1;%1-Constant
m_noise=0.05;
p_noise=1*0.05;
x0_initial=[10;2];
end

if sys==2;%1-ramp
m_noise=0.05;
p_noise=1*0.05;
x0_initial=[10;3];
end

if sys==3;%SMD without weak theta
m_noise=1*[0.001,0.004];
p_noise=1*[0.001,0.002];
x0_initial=[1;0;5;0.5];
end

if sys==4;%SMD with weak theta
m_noise=1*[0.001,0.004];
p_noise=1*[0.001,0.002];
x0_initial=[1;0;5;0.5;0.5];
end

if sys==5%Longitudinal Dynamics
m_noise=[0.0001,0.0001,0.0001,1,0.1];
p_noise=1*[0.00005,0.0001,0.00005,0.5];
x0_initial=[0;0;0;0; -0.5040;-4.5532;-0.4358;-0.0078;-7.5368];
end

if sys==6;%Lateral Dynamics
m_noise=[0.00001,0.0001,0.0001,0.0001,0.005,0.004,0.001];
p_noise=1*[0.000005,0.0002,0.00005,0.0001];
x0_initial=[0;0;0;0;-0.216;-0.0033;0.168;-0.1164;-6.984;2.1384;0.0101; -0.7980;...
    -0.8544;-0.0054;0.0319;19.7208;0.5208;-0.5136;-3.38];
end

%used for sys=5, 6
IEKF=0;
SOKF=0;

[Z,R_inj,Q_inj,PCRB,hx]=get_data;

display(['No. of data points N =' num2str(N)]);
```

```matlab
display(['No. of states=' num2str(ns)]);
display(['No. of parameters=' num2str(np)]);
display(['No. of measurements=' num2str(nm)]);
display(['size(X0)=' num2str(total_sts) 'x' num2str(1)]);
display(['size(P0)=' num2str(total_sts) 'x' num2str(total_sts)]);
display(['size(Q)=' num2str(total_sts) 'x' num2str(total_sts)]);
display(['size(R)=' num2str(nm) 'x' num2str(nm)]);
display(['size(F)=' num2str(total_sts) 'x' num2str(total_sts)]);
display(['size(H)=' num2str(nm) 'x' num2str(total_sts)]);
display(['size(K)=' num2str(total_sts) 'x' num2str(nm)]);


%%Pre-allocating memory with zeros as values
StoreX0=zeros(e_max,total_sts);X_smooth=[];StoreP0=StoreX0;Save_Q=[];
P_EKF=zeros(e_max,total_sts);StoreR=zeros(e_max,nm);StoreQ=StoreX0;
Err=P_EKF;S1diag=zeros(N,nm);P_smooth=[];S2diag=S1diag;S3diag=S1diag;
Blob_X0=[];Blob_P0=[];J5=zeros(e_max,1);J1=zeros(e_max,nm);J2=J1;J3=J1;
J4=J1;Save_R=[];Save_P0=[];Save_X=[];J6=zeros(e_max,ns);J7=J6;J8=J6;
C1diag=zeros(N,ns);C2diag=C1diag;C3diag=C1diag;X_comb=[];P_comb=[];
K_all=[];Store_K=zeros(e_max,total_sts*nm);filename='workspace_ref';

xk_prior=zeros(total_sts,N);pk_prior=zeros(total_sts,total_sts,N);hx1=zeros(nm,N);
H1=zeros(nm,total_sts,N);hx2=hx1;hx3=hx1;hx4=hx1;invS=zeros(nm,nm,N);
xk_posterior=xk_prior;pk_minus=zeros(N,total_sts);pk_plus=pk_minus;fail=0;
K=zeros(total_sts,nm,N);dynamical_residue=hx1;S1=invS;S2=S1;S3=S1;
fxs=xk_prior;pk_posterior=pk_prior;F=pk_prior;Fs=F;Fd=F;innov=hx1;H2=H1;
filter_residue=hx1;smooth_residue=hx1;H3=H1;xd_est=xk_prior;w2=xk_prior;
X_NR=zeros(ns,N);tx=t(2:N+1);xx=repmat(tx,2,1);xx=xx(:);warning=zeros(e_max,1);
CRB_NR=NaN(e_max,np);Theta_NR=NaN(e_max,np);R_NR=NaN(e_max,nm);

for e=1:e_max

z=Z(:,:,e);flag=1;
Zcost=zeros(iter_max,nm,4);L=zeros(iter_max,1);Xcost=zeros(iter_max,ns,3);
if sum(p_noise)==0,[CRB_NR(e,:),Theta_NR(e,:),R_NR(e,:),X_NR]=Gauss_Newton(z);end

for iter=1:iter_max

if iter==1
x0=x0_initial(1:total_sts);
P0=10^P0_power*eye(total_sts);
Q=10^Q0_power*diag([ones(1,ns) 0*ones(1,np)]);
R=2^R0_power*eye(nm);
end
```

```matlab
if iter>1
x0=x0_est;
P0=P0_est;
R=R_est;
Q=Q_est;
end

%Q=1.0*diag([Q_inj(e,:) 0*ones(1,np)]);
%R=1.0*diag(R_inj(e,:));

%%For Q=0 case
if sum(p_noise)==0,Q=diag([ones(1,ns)*10^-10 zeros(1,np)]);end

%Forcing the diagonal values to be +ve
Q=Q-diag(diag(Q))+diag(abs(diag(Q)));
R=R-diag(diag(R))+diag(abs(diag(R)));
P0=P0-diag(diag(P0))+diag(abs(diag(P0)));

E_iter=[e,iter];
display(E_iter,'ensemble , iter');
display(x0','x0');display(diag(P0)','P0');
display(diag(R)','R');display(diag(Q)','Q');

for k = 1:N %Data processing loop Starts

if k==1 , x_prev=x0; p_prev=P0;end
if k>1 , x_prev=xk_posterior(:,k-1);p_prev=pk_posterior(:,:,k-1);end

[fx,F(:,:,k)]=state_transition(x_prev,k,k+1,p_prev);

%%%%Kalman time update equations
xk_prior(:,k)=fx;
pk_prior(:,:,k) =F(:,:,k)*p_prev*F(:,:,k)'+1*Q;

epsilon=1;i=0;%%Iterated EKF
while epsilon>10^-3

i=i+1;
if i==1,x_nominal=xk_prior(:,k);p_nominal=pk_prior(:,:,k);end
if i>1,x_nominal=x_updated;p_nominal=p_updated;end

[hx1(:,k),H1(:,:,k),l1]=measurement_transition(x_nominal,k,p_nominal);
```

```matlab
S1(:,:,k)=H1(:,:,k)*pk_prior(:,:,k)*H1(:,:,k)'+R+l1;invS(:,:,k)=inv(S1(:,:,k));
innov(:,k) =z(:,k)- hx1(:,k)-H1(:,:,k)*(xk_prior(:,k)-x_nominal);


%%%%Kalman measurement update equations
K(:,:,k) =pk_prior(:,:,k)*H1(:,:,k)'*invS(:,:,k);
p_updated=(eye(total_sts)-K(:,:,k)*H1(:,:,k))*pk_prior(:,:,k);
x_updated = xk_prior(:,k) + K(:,:,k)*innov(:,k);

if IEKF==1,epsilon=norm(x_updated-x_nominal);else epsilon=0;end
if i>100,fail=fail+1;break;end
end

xk_posterior(:,k) = x_updated;pk_posterior(:,:,k)=p_updated;


end %Data processing loop End

for k=1:N,L(iter,:)=L(iter,:)+innov(:,k,1)'*invS(:,:,k)*innov(:,k)+1*log(det(S1(:,:,k)));end
if iter>1&&flag==1,delta=abs((L(iter)-L(iter-1))/L(iter));if delta<stop,flag=0;end,end

if flag==1

[xk_smooth,Pk_smooth,pk_smooth,x0smooth,P0smooth,Plag]=...
smooth(xk_posterior,pk_posterior,xk_prior,pk_prior,F,H1,K,S1,invS,innov,x0,P0);
x0_est=x0smooth;

for k=1:N
if k==1,x_prev=x0_est;p_prev=P0smooth;xd_prev=x0_est;else
x_prev=xk_smooth(:,k-1);p_prev=pk_smooth(:,:,k-1);xd_prev=xd_est(:,k-1);end
[fxs(:,k),Fs(:,:,k)]=state_transition(x_prev,k,k+1,p_prev);

[xd_est(:,k),Fd(:,:,k)]=state_transition(xd_prev,k,k+1,0);
fxd(:,k)=1*xd_est(:,k)+Fd(:,:,k)*(x_prev-1*xd_prev);
w2(:,k)=xk_smooth(:,k)-1*xd_est(:,k)-Fd(:,:,k)*(x_prev-1*xd_prev);
end

for k=1:N,
[hx2(:,k),H2(:,:,k)]=measurement_transition(xk_posterior(:,k),k,pk_posterior(:,:,k));
[hx3(:,k),H3(:,:,k)]=measurement_transition(xk_smooth(:,k),k,pk_smooth(:,:,k));
[hx4(:,k)]=measurement_transition(xd_est(:,k),k,0);
dynamical_residue(:,k)= z(:,k)-hx4(:,k);
filter_residue(:,k) = z(:,k)-hx2(:,k);
smooth_residue(:,k) = z(:,k)-hx3(:,k);
```

```matlab
pk_minus(k,:)=diag(pk_prior(:,:,k)); pk_plus(k,:)=diag(pk_posterior(:,:,k));
S2(:,:,k)=-H2(:,:,k)*pk_posterior(:,:,k)*H2(:,:,k)'+R;S2diag(k,:)=abs(diag(S2(:,:,k)));
S3(:,:,k)=-H3(:,:,k)*pk_smooth(:,:,k)*H3(:,:,k)'+R;S3diag(k,:)=abs(diag(S3(:,:,k)));
S1diag(k,:)=diag(S1(:,:,k));
end

%%%Estimation of Q,R
w1=xk_smooth-fxs;
Qcap1=0;Qcap2=0;Rcap=0;
for k=1:N
if k==1,
    x_prev=x0_est;p_prev=P0smooth;else x_prev=xk_smooth(:,k-1);
    p_prev=pk_smooth(:,:,k-1);
end
Qcap1=Qcap1+w1(:,k)*w1(:,k)'+Q2term*(Fs(:,:,k)*p_prev*Fs(:,:,k)'+...
    pk_smooth(:,:,k)-Plag(:,:,k)*Fs(:,:,k)'-Fs(:,:,k)*Plag(:,:,k)');
Qcap2=Qcap2+w2(:,k)*w2(:,k)'+Q2term*(Fd(:,:,k)*p_prev*Fd(:,:,k)'+...
    pk_smooth(:,:,k)-Plag(:,:,k)*Fd(:,:,k)'-Fd(:,:,k)*Plag(:,:,k)');
Rcap=Rcap+smooth_residue(:,k)*smooth_residue(:,k)'+...
    R2term*(H3(:,:,k)*pk_smooth(:,:,k)*H3(:,:,k)');
end

Q_MT=0;w3=xk_posterior-xk_prior;
R_MT=0;R_MS=0;
for k=N-last_n+1:N
Q_MT=Q_MT+w3(:,k)*w3(:,k)'-Q2term*((pk_prior(:,:,k)-Q)-pk_posterior(:,:,k));
R_MT=R_MT+innov(:,k)*innov(:,k)'-R2term*(H1(:,:,k)*(pk_prior(:,:,k))*H1(:,:,k)');
R_MS=R_MS+filter_residue(:,k)*filter_residue(:,k)'+...
    R2term*(H2(:,:,k)*pk_posterior(:,:,k)*H2(:,:,k)');
end

if Q_method==1,Q_full=1/N*Qcap1;end
if Q_method==2,Q_full=K(:,:,N)*cov(innov(:,N-last_n+1:N)')*K(:,:,N)';end
if Q_method==3,Q_full=1/last_n*Q_MT;end
if Q_method==4,Q_full=1/N*Qcap2;end

if Q_option==1,Q_est=Q_full;Q_est(ns+1:total_sts,:)=0;
    Q_est(:,ns+1:total_sts)=0;Q_est=diag(abs(diag(Q_est)));end
if Q_option==2,Q_est=diag(abs(diag(Q_full)));end
if Q_option==3,Q_est=Q_full;end

if R_method==1,R_full=1/N*Rcap;end
if R_method==2,R_full=1/last_n*R_MS;end
if R_method==3,R_full=1/last_n*R_MT;end
if R_method==4,R_full=cov(dynamical_residue');end
```

```matlab
if R_option==1,R_est=diag(abs(diag(R_full)));end
if R_option==2,R_est=R_full;end

IM=0;%%Estimation of P0
for k=1:N,IM=IM+F(:,:,k)'*H1(:,:,k)'*inv(R)*H1(:,:,k)*F(:,:,k);end

if rcond(IM)<10^-10
IM=0;%%Estimation of P0
for k=1:N,
    IM=IM+F(1:ns,1:ns,k)'*H1(:,1:ns,k)'*inv(R)*H1(:,1:ns,k)*F(1:ns,1:ns,k);end
    IM=[diag(abs(diag(IM))),zeros(ns,np);zeros(np,ns),N*eye(np)];
end

if P0_method==1,P0_full=Scale_up*pk_posterior(:,:,N);end
if P0_method==2,P0_full=inv(1/N*IM);end
if P0_method==3,P0_full=P0smooth;end

if P0_option==1,P0_est=P0_full;P0_est(1:ns,:)=0;P0_est(:,1:ns)=0;
x0_est=[Xd(1,1:ns)';xk_posterior(ns+1:end,N)];end
if P0_option==2,P0_est=diag(diag(P0_full));end
if P0_option==3,P0_est=P0_full;end

for k=1:N
if k==1,p_prev=P0smooth;else p_prev=pk_smooth(:,:,k-1);end
Zcost(iter,:,1)=Zcost(iter,:,1)+innov(:,k)'.^2./S1diag(k,:);
Zcost(iter,:,2)=Zcost(iter,:,2)+filter_residue(:,k)'.^2./S2diag(k,:);
Zcost(iter,:,3)=Zcost(iter,:,3)+smooth_residue(:,k)'.^2./S3diag(k,:);
Zcost(iter,:,4)=Zcost(iter,:,4)+dynamical_residue(:,k)'.^2;

C1diag(k,:)=-diag(Fs(1:ns,1:ns,k)*p_prev(1:ns,1:ns)*Fs(1:ns,1:ns,k)'+pk_smooth(1:ns,1:ns,k)-...
    Plag(1:ns,1:ns,k)*Fs(1:ns,1:ns,k)'-Fs(1:ns,1:ns,k)*Plag(1:ns,1:ns,k)'-Q(1:ns,1:ns))';
C2diag(k,:)=-diag(Fs(1:ns,1:ns,k)*p_prev(1:ns,1:ns)*Fs(1:ns,1:ns,k)'+pk_smooth(1:ns,1:ns,k)-...
    Plag(1:ns,1:ns,k)*Fs(1:ns,1:ns,k)'-Fs(1:ns,1:ns,k)*Plag(1:ns,1:ns,k)'-Q(1:ns,1:ns))';
C3diag(k,:)=diag(pk_prior(1:ns,1:ns,k)-pk_posterior(1:ns,1:ns,k))';

if sum(p_noise)>0
Xcost(iter,:,1)=Xcost(iter,:,1)+w1(1:ns,k)'.^2./C1diag(k,:);
Xcost(iter,:,2)=Xcost(iter,:,2)+w2(1:ns,k)'.^2./C2diag(k,:);
Xcost(iter,:,3)=Xcost(iter,:,3)+w3(1:ns,k)'.^2./C3diag(k,:);
end
end
end

if flag==0;Zcost(iter,:,:)=Zcost(iter-1,:,:);Xcost(iter,:,:)=Xcost(iter-1,:,:);end
```

```matlab
if e==e_max

Save_X=[Save_X;x0'];Save_P0=[Save_P0;diag(P0)'];
Save_R=[Save_R;diag(R)'];Save_Q=[Save_Q;diag(Q(:,:,1))'];


for s=1:total_sts,
X_temp=[xk_prior(s,:);xk_posterior(s,:)];X_comb_iter(:,s)=X_temp(:);
P_temp=[pk_minus(:,s)';pk_plus(:,s)'];P_comb_iter(:,s)=P_temp(:);
end

X_comb=[X_comb;X_comb_iter;NaN(1,total_sts)];
P_comb=[P_comb;P_comb_iter;NaN(1,total_sts)];
X_smooth=[X_smooth;xk_smooth';NaN(1,total_sts)];
P_smooth=[P_smooth;Pk_smooth;NaN(1,total_sts)];
K_all=[K_all;reshape(K(:),total_sts*nm,[])';NaN(1,total_sts*nm)];
Blob_X0=[Blob_X0;x0'];Blob_P0=[Blob_P0;diag(P0)'];

end
end

%%saving ensemble values
StoreX0(e,:)=x0;StoreP0(e,:)=diag(P0);StoreQ(e,:)=diag(Q(:,:,1));StoreR(e,:)=diag(R);
P_EKF(e,:)=diag(pk_posterior(:,:,N));Err(e,:)=xk_posterior(:,N)'-1*X(N+1,:,e);
MSE_NR(e,:)=mean((X(2:N+1,1:ns,e)'-X_NR).^2,2);Kf=K(:,:,N);Store_K(e,:)=Kf(:);

J1(e,:)=Zcost(iter_max,:,1)/N;J2(e,:)=Zcost(iter_max,:,2)/N;J3(e,:)=Zcost(iter_max,:,3)/N;
J4(e,:)=Zcost(iter_max,:,4)/N;J5(e,:)=L(iter_max,:)/N;if flag==1,warning(e,1)=1;end
J6(e,:)=Xcost(iter_max,:,1)/N;J7(e,:)=Xcost(iter_max,:,2)/N;J8(e,:)=Xcost(iter_max,:,3)/N;

end

Q_mean_EKF=mean(StoreQ(:,1:ns),1);
R_mean_EKF=mean(StoreR,1);
R_injected=mean(R_inj,1);
Q_injected=mean(Q_inj,1);
Theta_EKF=StoreX0(:,ns+1:total_sts);
Theta_mean_EKF=mean(Theta_EKF,1);
Sigma_EKF=sqrt(P_EKF);
Sigma_mean_EKF=mean(Sigma_EKF,1);
Theta_mean_NR=mean(Theta_NR,1);
Sigma_NR=sqrt(CRB_NR);
Sigma_mean_NR=mean(Sigma_NR,1);
R_mean_NR=mean(R_NR,1);
```

```matlab
R_std_ekf=std(StoreR,1,1);
Q_std_ekf=std(StoreQ,1,1);
Theta_corrcoef_EKF=corrcoef(Theta_EKF);
Theta_corrcoef_NR=corrcoef(Theta_NR);
Err_NR=Theta_NR-repmat(theta',e_max,1);
STD_Theta=std(Theta_EKF,1,1);
SIGMA_avg=Sigma_mean_EKF(ns+1:end);

display('---------Displaying results------------');

theta_ratio_NR_true=abs(Theta_mean_NR./theta');
theta_ratio_EKF_NR=abs(Theta_mean_EKF./Theta_mean_NR);
theta_ratio_EKF_true=abs(Theta_mean_EKF./theta')

PCRB_ratio=sqrt(PCRB)./Sigma_mean_EKF(1:ns)
CRB_ratio=Sigma_mean_NR./Sigma_mean_EKF(ns+1:end)

Consistency_ratio_EKF=std(Theta_EKF,1,1)./Sigma_mean_EKF(ns+1:end)
Consistency_ratio_NR=std(Theta_NR,1,1)./Sigma_mean_NR

Spread_EKF=abs(100*mean(sqrt((P_EKF(:,ns+1:end)+ Err(:,ns+1:end).^2)),1)./theta')
Spread_NR=abs(100*mean(sqrt(CRB_NR+ Err_NR.^2),1)./theta')

R_ratio_EKF_true=R_mean_EKF./m_noise
R_ratio_EKF_NR=R_mean_EKF./R_mean_NR
Q_ratio_EKF_true=Q_mean_EKF./p_noise

Q_ratio_INJ_true=Q_injected./p_noise;
R_ratio_INJ_true=R_injected./m_noise;
R_ratio_NR_true=R_mean_NR./m_noise;

Q_ratio_EKF_INJ=Q_mean_EKF./Q_injected;
R_ratio_EKF_INJ=R_mean_EKF./R_injected;

J1_8_mean=[sum(mean(J1,1)) ,sum(mean(J2,1)),sum(mean(J3,1)),sum(mean(J4,1)),...
sum(mean(J5,1)),sum(mean(J6,1)),sum(mean(J7,1)),sum(mean(J8,1))]

J1_8_std=[sum(std(J1,1,1)) ,sum(std(J2,1,1)),sum(std(J3,1,1)),sum(std(J4,1,1)),...
sum(std(J5,1,1)),sum(std(J6,1,1)),sum(std(J7,1,1)),sum(std(J8,1,1))]

Mean_Kfinal=mean(Store_K,1),Sigma_Kfinal=std(Store_K,1,1)

%%--------Plotting figures----------------------------------

%generate x-axis values (time points) and legend entries for plotting
```

```matlab
t1=[];for l=1:iter_max, y=t(N+1)*(l-1):dt:t(N+1)*l;t1=[t1 y NaN];end

tp1=t(2):dt:t(N+1)*iter_max+dt*(iter_max-1);t2=[];
for l=1:iter_max,
tp02=repmat(tp1(N*(l-1)+1:N*l),2,1);
tp2=[tp02(:)' NaN tp1(~(N*(l-1)+1:N*l))];
t2=[t2 tp2];
end

y1=t(2):dt:t(N+1)*iter_max+dt*(iter_max-1);t3=[];
for l=1:iter_max,t3=[t3 y1(N*(l-1)+1:N*l) NaN y1(~(N*(l-1)+1:N*l))];end

t4=t(1):N*dt:t(1)+N*dt*iter_max;t4=t4(1:end-1);t5=1:iter_max;

string1=[];string2=[];string3=[];string4=[];string5=[];string6=[];

for p=1:np
string1=char(string1,['\bf\theta' num2str(p)]);
string3=char(string3,['P0 \bf\theta' num2str(p)]);
end

for s=1:ns+np
string4=char(string4,['Q  ' num2str(s) num2str(s)]);
string5=char(string5,['P0 ' num2str(s) num2str(s)]);
end

for m=1:nm
string2=char(string2,['R' num2str(m) num2str(m)]);
for s=1:total_sts
string6=char(string6,['K ' num2str(s) num2str(m)]);
end
end

r=smooth_residue;w=w1;
Err_Th=abs(Save_X(:,ns+1:end)-repmat(Save_X(iter_max,ns+1:end),size(Save_X,1),1));
Err_P0=abs(Save_P0(:,ns+1:end)-repmat(Save_P0(iter_max,ns+1:end),size(Save_P0,1),1));
Err_R=abs(Save_R-repmat(Save_R(iter_max,:),size(Save_R,1),1));
Err_Q=abs(Save_Q(:,1:ns)-repmat(Save_Q(iter_max,1:ns),size(Save_Q,1),1));

toc,save(filename),
evalin('base',sprintf('load("%s")',filename))%load workspace

%%-------------EKF Plots-----------------------------

figure,semilogy(t5,Save_Q(:,1:ns),'-',t5,Save_R,'*-','linewidth',2),grid on
```

```matlab
legend(char(string4(2:ns+1,:),string2(2:end,:))),
title('Variation of R and Q with iterations','fontsize',14)
set(gca,'fontsize',14)

figure,semilogy(t5,abs(sum(Zcost(:,:,1)/N,2)),t5,abs(sum(Zcost(:,:,2)/N,2)),t5,abs(sum(Zcost(:,:,3)/N,2)),...
    t5,abs(sum(Zcost(:,:,4)/N,2)),t5,abs(sum(L/N,2)),...
t5,abs(sum(Xcost(:,:,1)/N,2)),t5,abs(sum(Xcost(:,:,2),2)),t5,abs(sum(Xcost(:,:,3),2)),'linewidth',2),
legend('J1','J2','J3','J4','J5','J6','J7','J8'),grid on
title('Variation of different costs (J1-J8) with iterations','fontsize',14)
set(gca,'fontsize',14)

figure,semilogy(t5,abs(Save_X(:,ns+1:end)),'-','linewidth',2),hold on
semilogy(t5,Save_P0(:,ns+1:end),'*-','linewidth',2),grid on
legend(char(string1(2:end,:),string3(2:end,:))),
title('Variation of parameter and its initial covariance (P0) with iterations','fontsize',14)
set(gca,'fontsize',14)

for s=1:nm
figure,subplot(3,1,1),plot(t(2:end),sqrt(S1diag(:,s)),'b-',t(2:end),-sqrt(S1diag(:,s)),'b-',...
    t(2:end),innov(s,:)','r-','linewidth',2),grid on,
title(['innovations outside {\sigma} bound = ' ...
    num2str(100*length(find(sqrt(S1diag(:,s))<abs(innov(s,:)')))/N) ' %'],'fontsize',14),
set(gca,'fontsize',14)

subplot(3,1,2),plot(t(2:end),sqrt(S2diag(:,s)),'b-',t(2:end),-sqrt(S2diag(:,s)),'b-',...
    t(2:end),filter_residue(s,:)','r-','linewidth',2),grid on,
title(['filter-residue outside {\sigma} bound = ' ...
    num2str(100*length(find(sqrt(S2diag(:,s))<abs(filter_residue(s,:)')))/N) ' %'],'fontsize',14)
set(gca,'fontsize',14)

subplot(3,1,3),plot(t(2:end),sqrt(S3diag(:,s)),'b-',t(2:end),-sqrt(S3diag(:,s)),'b-',...
    t(2:end),smooth_residue(s,:)','r-','linewidth',2),grid on,
title(['smooth-residue outside {\sigma} bound = ' ...
    num2str(100*length(find(sqrt(S3diag(:,s))<abs(smooth_residue(s,:)')))/N) ' %'],'fontsize',14),
set(gca,'fontsize',14)
end

for s=1:nm
figure,plot(tx,hx4(s,:),'-',tx,hx2(s,:),'-',tx,hx3(s,:),'-',tx,z(s,:),'go','linewidth',2),
legend('h(Xd)','h(X+)','h(Xs)','Z'),grid on,
title(['Comparison of the predicted dynamics, posterior, smoothed and the measurement '...
    num2str(s) ],'fontsize',14),set(gca,'fontsize',14)
end

for s=1:nm
```

```matlab
n=(s-1)*2+1;
[acr1,lag_s1]=xcorr(r_true(:,m,e_max),'coeff'); [acr2,lag_s2]=xcorr(r(m,:)','coeff');
figure,subplot(2,1,1),plot(t(2:N+1),r_true(:,s,e_max),'b-',t(2:N+1),r(s,:)','r--','linewidth',2),grid on,
title(['Time variation of injected and estimated measurement noise ' num2str(s)],'fontsize',14)
set(gca,'fontsize',14)

subplot(2,1,2),plot(lag_s1(lag_s1>=0),acr1(lag_s1>=0,:),'b-',...
    lag_s2(lag_s2>=0),acr2(lag_s2>=0,:),'r--','linewidth',2),
title(['Time lag variation of  autocorrelation of injected and estimated measurement noise '...
    num2str(s)],'fontsize',14),legend('injected','estimated'),grid on
set(gca,'fontsize',14)
end

for s=1:ns
n=(s-1)*2+1;
[acr1,lag_s1]=xcorr(q_true(:,s,e_max),'coeff'); [acr2,lag_s2]=xcorr(w(s,:)','coeff');
figure,subplot(2,1,1),plot(t(2:N+1),q_true(:,s,e_max),'b-',t(2:N+1),w(s,:)','r--','linewidth',2),grid on,
title(['Time variation of injected and estimated process noise ' num2str(s)],'fontsize',14)
set(gca,'fontsize',14)

subplot(2,1,2),plot(lag_s1(lag_s1>=0),acr1(lag_s1>=0,:),'b-',...
    lag_s2(lag_s2>=0),acr2(lag_s2>=0,:),'r--','linewidth',2),
title(['Time lag variation of  autocorrelation of injected and estimated process noise '...
    num2str(s)],'fontsize',14)
legend('injected','estimated'),grid on
set(gca,'fontsize',14)
end

for p=1:np
figure,subplot(2,1,1),plot(t1,repmat([ones(N+1,1)*theta(p);NaN],iter_max,1),'k-',...
    t3,X_smooth(:,ns+p),'r-',t2,X_comb(:,ns+p),'b-',t4,Blob_X0(:,ns+p),'go','linewidth',2)
legend(char(['\bf\theta true' num2str(p)],'\bf\theta smooth','\bf\theta filtered','\bf\theta initial')),
title(['\bf{\theta} = ' num2str(p) ' versus iterated time instants'],'fontsize',14),grid on
set(gca,'fontsize',14)

subplot(2,1,2),semilogy(t3,P_smooth(:,ns+p),'r-',t2,P_comb(:,ns+p),'b-',...
    t4,Blob_P0(:,ns+p),'go','linewidth',2),hold on,legend(char('P smooth','P filtered','P0'))
title(['P\bf{\theta} = ' num2str(p) ' versus iterated time instants'],'fontsize',14),grid on
set(gca,'fontsize',14)
end

figure,semilogy(t5,Err_Th,'--',t5,Err_P0,'*-',t5,Err_R,'o-',t5,Err_Q,'-','linewidth',2),grid on
legend(char(string1(2:end,:),string3(2:end,:),string4(2:ns+1,:),string2(2:end,:))),
title('The absolute difference between the iterated and final values with iterations','fontsize',14)
set(gca,'fontsize',14)
```

```matlab
figure,plot(t3,K_all,'linewidth',2),grid on,legend(string6(2:end,:))
title('Gain variations through cumulative time with iterations ','fontsize',14)
set(gca,'fontsize',14)

%%%-------------Start of  simulated system function----------------------------

function [Z,varR,varQ,PCRB,hx] = get_data

global m_noise ns nm np sys p_noise Xd theta N e_max X total_sts...
    unit_f unit_h q_true r_true

rng('default');

if sys==1
theta=1;nm=1;ns=1;np=1;
x0_true=[10;theta];
end

if sys==2
theta=2;nm=1;ns=1;np=1;
x0_true=[10;theta];
end

if sys==3
nm=2;ns=2;np=2;
theta=[4;0.4];
x0_true=[1;0;theta];
end

if sys==4
nm=2;ns=2;np=3;
theta=[4;0.4;0.6];
x0_true=[1;0;theta];
end

if sys==5
nm=5;ns=4;np=5;
theta=[-0.42;-3.7943;-0.3632;-0.006489;-6.2807];
x0_true=[0;0;0;0;theta];
end
```

```matlab
if sys==6
nm=7;ns=4;np=15;
theta=[-0.18;-0.00278;0.14;-0.097;-5.82;1.782;0.0084;-0.665;-0.712;...
   -0.00447;0.02657;16.434;0.434;-0.428;-2.824];
x0_true=[0;0;0;0;theta];
end

total_sts=ns+np;
unit_f=eye(total_sts);
unit_h=eye(nm);

X=zeros(N+1,total_sts,e_max);Xd=zeros(N+1,total_sts);
Xd(1,:)=x0_true;Z=zeros(nm,N,e_max);
q_true=repmat(sqrt(p_noise),[N 1 e_max]).*randn(N,ns,e_max);
r_true=repmat(sqrt(m_noise),[N 1 e_max]).*randn(N,nm,e_max);
hx=zeros(nm,N,e_max);varR=zeros(e_max,nm);varQ=zeros(e_max,ns);
D11=zeros(ns,ns,N,e_max);D12=D11;D22=D11;

%%Adding process noise and measurement noise
for e=1:e_max

X(1,:,e)=x0_true;
varR(e,:)=var(r_true(:,:,e),1);
varQ(e,:)=var(q_true(:,:,e),1);

for k=2:N+1

[x_next_true,Khi]=state_transition(X(k-1,:,e)',k-1,k,0);
X(k,:,e)=x_next_true+[q_true(k-1,:,e)';zeros(np,1)];

if e==e_max
xd_next_true=state_transition(Xd(k-1,:)',k-1,k,0);
Xd(k,:)=xd_next_true;
end

[hx(:,k-1,e),H]=measurement_transition(X(k,:,e)',k-1,0);
Z(1:nm,k-1,e)=hx(:,k-1,e)+r_true(k-1,:,e)';

if sum(p_noise)>0
D11(:,:,k-1,e)=Khi(1:ns,1:ns)'*diag(1./varQ(e,:))*Khi(1:ns,1:ns);
D12(:,:,k-1,e)=Khi(1:ns,1:ns)'*diag(1./varQ(e,:));
D22(:,:,k-1,e)=diag(1./varQ(e,:))+H(:,1:ns)'*diag(1./varR(e,:))*H(:,1:ns);
end
end
end
```

```
if sum(p_noise)>0
D1=mean(D11,4);D2=mean(D12,4);D3=mean(D22,4);FIM=0;
for k=1:N
FIM=D3(:,:,k)-D2(:,:,k)'*inv(FIM+D1(:,:,k))*D2(:,:,k);
end
PCRB=diag(inv(FIM))';
else
PCRB=NaN(1,ns);
end

%%%%-------------------System Dynamics-------------------------

function dx=dynamics(t1,x,P)

global  ns np sys SOKF

if sys==3,dx=[x(2);-x(ns+1)*x(1)-x(ns+2)*x(2);zeros(np,1)];end
if sys==4,dx=[x(2);-x(ns+1)*x(1)-x(ns+3)*x(1)^3-x(ns+2)*x(2);zeros(np,1)];end
if sys==5,
dx=[x(ns+1)*x(1)+x(2)-0.0021*x(3)+x(ns+4)*get_u(t1);
x(ns+2)*x(1)+x(ns+3)*x(2)+x(ns+5)*get_u(t1);x(2);
15.67*x(1)-32.16*x(3)+8.354*get_u(t1); zeros(np,1)];
end
if sys==6,u=get_u(t1);
dx=[x(ns+1)*x(1)+x(ns+2)*x(2)+x(ns+3)*x(4)+x(ns+10)*u(1,1)+x(ns+11)*u(2,1);...
x(ns+4)*x(1)+x(ns+5)*x(2)+x(ns+6)*x(4)+x(ns+12)*u(1,1)+x(ns+13)*u(2,1);...
x(2);x(ns+7)*x(1)+x(ns+8)*x(2)+x(ns+9)*x(4)+x(ns+14)*u(1,1)+x(ns+15)*u(2,1);zeros(np,1)];
end

if norm(P)>0 && SOKF==1
f2term=get_f2term(x,P);
dx=dx+f2term;
end

%%%-------------------------------Get control input-------------------------------

function u=get_u(t1)

global sys

if sys==1 || sys==2 || sys==3 || sys==4, u=0; end

Uamp=0.1;
```

```matlab
if sys==5
if t1<=0.5,u=0;end
if t1<=1 && t1>0.5,u=Uamp;end
if t1<=1.5 && t1>1,u=-Uamp;end
if t1>1.5,u=0;end
end

if sys==6

if t1<=1,u(1,1)=0;end
if t1<=4 && t1>1,u(1,1)=Uamp;end
if t1<=6 && t1>4,u(1,1)=-Uamp;end
if t1<=7 && t1>6,u(1,1)=Uamp;end
if t1<=8 && t1>7,u(1,1)=-Uamp;end
if t1>8,u(1,1)=0;end

t1=t1+0.5;
if t1<=1,u(2,1)=0;end
if t1<=4 && t1>1,u(2,1)=Uamp;end
if t1<=6 && t1>4,u(2,1)=-Uamp;end
if t1<=7 && t1>6,u(2,1)=Uamp;end
if t1<=8 && t1>7,u(2,1)=-Uamp;end
if t1>8,u(2,1)=0;end

end
%%%--------------------------------State transition function-------------------------------

function [x_next,F] = state_transition(xp,k,k_next,P)

global  sys dt t np ns

delta=dt*(k_next-k);
if sys==1,F=[xp(2) xp(1);0 1];A=[xp(2) 0;0 1];x_next=A*xp;end
if sys==2,F=[1 delta;0 1];x_next=F*xp;end

if sys==3
A1 = [0 1 0 0; -xp(3) -xp(4) 0 0;zeros(np,ns+np)]; x_next=expm(A1*delta)*xp;
A= [0 1 0 0; -xp(3) -xp(4) -xp(1) -xp(2);zeros(np,ns+np)];F=expm(A*delta);
end

if sys==4
A1=[0 1 0 0 0;-xp(3)-(xp(5)*xp(1)^2) -xp(4) 0 0 0;zeros(np,ns+np)];x_next=expm(A1*delta)*xp;
A= [0 1 0 0 0; -xp(3)-(xp(5)*3*xp(1)^2) -xp(4) -xp(1) -xp(2) -xp(1)^3;zeros(np,ns+np)];
F=expm(A*delta);
end
```

```matlab
if sys==5
A=[xp(ns+1) 1 -0.0021 0 xp(1) 0 0 get_u(t(k)) 0;xp(ns+2) xp(ns+3) 0 0 0 xp(1) xp(2) 0  get_u(t(k));...
0 1 0 0 0 0 0 0 0;  15.67 0 -32.16 0 0 0 0 0 0;zeros(np,ns+np)];
F=expm(A*delta);
end

if sys==6
u=get_u(t(k));
A=[xp(ns+1) xp(ns+2) 0 xp(ns+3) xp(1) xp(2) xp(4) zeros(1,6) u(1,1) u(2,1) 0 0 0 0;...
xp(ns+4) xp(ns+5) 0 xp(ns+6) 0 0 0 xp(1) xp(2) xp(4) zeros(1,5) u(1,1) u(2,1) 0 0;...
0 1 zeros(1,17);xp(ns+7) xp(ns+8) 0 xp(ns+9) zeros(1,6) xp(1) xp(2) xp(4) 0 0 0 0 u(1,1) u(2,1);...
zeros(np,ns+np)];F=expm(A*delta);
end

if sys==5 || sys==6
[~,next_states]=ode45(@dynamics,t(k):delta:t(k+1),xp,[],P);
x_next=next_states(end,:)';
end

%%%--------Get second order terms of time update equation--------------------------

function f2term=get_f2term(x,P)

global sys unit_f ns

if sys==1,Fxx=[0 1;1 0]; f2term=0.5*unit_f(:,1)*trace(Fxx*P);end
if sys==2, f2term=0;end
if sys==3,Fxx=[0 0 -1 0; 0 0 0 -1; -1 0 0 0 ; 0 -1 0 0];f2term=0.5*unit_f(:,2)*trace(Fxx*P);end
if sys==4, Fxx=[-6*x(1)*x(5) 0 -1 0 -3*x(1)^2 ; 0 0 0 -1 0 ; -1 0 0 0 0 ; 0 -1 0 0 0;-3*x(1)^2  0 0 0 0];
   f2term=0.5*unit_f(:,2)*trace(Fxx*P) ;end

if sys==5,
Fxx(:,:,1)=[0 0 0 0 1 zeros(1,4);zeros(3,9);1 zeros(1,8);zeros(4,9)];
Fxx(:,:,2)=[0 0 0 0 0 1 zeros(1,3);0 0 0 0 0 0 1 0 0;zeros(3,9);1 zeros(1,8);0 1 zeros(1,7);...
   zeros(2,9)];
f2term=0;
for i=1:2
f2term=f2term+0.5*unit_f(:,i)*trace(Fxx(:,:,i)*P);
end
end

if sys==6,
Fxx(:,:,1)=[0 0 0 0 1 zeros(1,14);0 0 0 0 0 1 zeros(1,13);zeros(1,19);0 0 0 0 0 0 1 zeros(1,12);...
1 zeros(1,18);0 1 zeros(1,17);0 0 1 zeros(1,15);zeros(12,19)];
```

```matlab
Fxx(:,:,2)=[zeros(1,7) 1 zeros(1,11);zeros(1,8) 1 zeros(1,10);zeros(1,19);...
zeros(1,9) 1 zeros(1,9);zeros(3,19);1 zeros(1,18);0 1 zeros(1,17);0 0 0 1 zeros(1,15);zeros(9,19)];
Fxx(:,:,3)=zeros(19,19);
Fxx(:,:,4)=[zeros(1,10) 1 zeros(1,8);zeros(1,11) 1 zeros(1,7);zeros(1,19);zeros(1,12) 1 zeros(1,6);...
zeros(6,19);1 zeros(1,18);0 1 zeros(1,17);0 0 0 1 zeros(1,15);zeros(6,19)];
f2term=0;
for i=1:ns
f2term=f2term+0.5*unit_f(:,i)*trace(Fxx(:,:,i)*P);
end
end


%%%--------------------------Measurement linearisation function-----------------------------


function [zp,Hk,lambda]=measurement_transition(x,k,P)

global  sys ns t SOKF

lambda=0;
if sys==1,Hk=[1 0];zp=Hk*x;end
if sys==2,Hk=[1 0];zp=Hk*x;end
if sys==3,Hk=[1 0 0 0;0 1 0 0];zp=Hk*x;end
if sys==4,Hk=[1 0 0 0 0;0 1 0 0 0];zp=Hk*x;end
if sys==5,u0=100;
Hk=[1 0 0 0 0 0 0 0 0;0 1 0 0 0 0 0 0 0;0 0 1 0 0 0 0 0 0;0 0 0 1 0 0 0 0;...
   u0*[x(ns+1) 0 0 0 x(1) 0 0 get_u(t(k)) 0]];
h=@(x) [x(1);x(2);x(3);x(4);u0*(x(ns+1)*x(1)+x(ns+4)*get_u(t(k)))];zp=h(x);
end
if sys==6,u=get_u(t(k));u0=100;
Hk=[1 zeros(1,18);0 1 zeros(1,17);0 0 1 zeros(1,16);0 0 0 1 zeros(1,15);...
u0*[x(ns+1) 0 0 0 x(1) zeros(1,8) u(1,1) u(2,1) 0 0 0 0];...
x(ns+4) x(ns+5) 0 x(ns+6) 0 0 0 x(1) x(2) x(4) zeros(1,5) u(1,1) u(2,1) 0 0;...
x(ns+7) x(ns+8) 0 x(ns+9) zeros(1,6) x(1) x(2) x(4) 0 0 0 0 u(1,1) u(2,1)];
h=@(x) [x(1);x(2);x(3);x(4);u0*(x(ns+1)*x(1)+x(ns+10)*u(1,1)+x(ns+11)*u(2,1));...
x(ns+4)*x(1)+x(ns+5)*x(2)+x(ns+6)*x(4)+x(ns+12)*u(1,1)+x(ns+13)*u(2,1);...
x(ns+7)*x(1)+x(ns+8)*x(2)+x(ns+9)*x(4)+x(ns+14)*u(1,1)+x(ns+15)*u(2,1)];zp=h(x);
end

if norm(P)>0 && SOKF==1
[h2term,lambda]=get_h2term(P);
zp=zp+h2term;
end

%%%--------------------Get second order terms of measurement eqn-------------------------
```

```matlab
function [h2term,l]=get_h2term(P)

global  sys unit_h nm ns

if sys==1 || sys==2 || sys==3 || sys==4, h2term=0;l=0;end

if sys==5
Dxx=100*[0 0 0 0 1 0 0 0 0;zeros(3,9);1 zeros(1,8);zeros(4,9)];
h2term=0.5*unit_h(:,5)*trace(Dxx*P);
l=zeros(nm,nm);l(5,5)=0.5*trace(Dxx*P*Dxx*P);
end

if sys==6
u0=100;
Dxx(:,:,5)=u0*[0 0 0 0 1 zeros(1,14);zeros(3,19);1 zeros(1,18);zeros(14,19)];
Dxx(:,:,6)=[zeros(1,7) 1 zeros(1,11);zeros(1,8) 1 zeros(1,10);zeros(1,19);zeros(1,9) 1 zeros(1,9);...
zeros(3,19);1 zeros(1,18);0 1 zeros(1,17);0 0 0 1 zeros(1,15);zeros(9,19)];
Dxx(:,:,7)=[zeros(1,10) 1 zeros(1,8);zeros(1,11) 1 zeros(1,7);zeros(1,19);zeros(1,12) 1 zeros(1,6);...
zeros(6,19);1 zeros(1,18);0 1 zeros(1,17);0 0 0 1 zeros(1,15);zeros(6,19)];
h2term=0;
for i=ns+1:nm
h2term= h2term+0.5*unit_h(1:nm,i)*trace(Dxx(:,:,i)*P);
end
l=zeros(nm,nm);
for i=ns+1:nm
for j=ns+1:nm
l(i,j)=0.5*trace(Dxx(:,:,i)*P*Dxx(:,:,j)*P);
end
end
end

%%------------------Start of smooth function----------------------------------------------

function [xk_smooth,P_smooth,pk_smooth,x0s,P0s,P_lag] =...
   smooth(xk_plus,pk_plus,xk_minus,pk_minus,Khi,H,K,S,invS,innov,x0,P0)

global N total_sts smooth_option

xk_smooth(:,N)=xk_plus(:,N);P_smooth(N,:)=diag(pk_plus(:,:,N))';
pk_smooth(:,:,N)=pk_plus(:,:,N);


if smooth_option==1
```

```
%%-----------------RTS Smoothing--------------------------

for k = N-1:-1:1 %Smoothing loop Starts

%%Smoothing Equations
K_rts(:,:,k+1)=(pk_plus(:,:,k)*Khi(:,:,k+1)'/pk_minus(:,:,k+1));
pk_smooth(:,:,k)=pk_plus(:,:,k)+(K_rts(:,:,k+1)*(pk_smooth(:,:,k+1)-...
   pk_minus(:,:,k+1))*K_rts(:,:,k+1)');
xk_smooth(:,k)=xk_plus(:,k)+K_rts(:,:,k+1)*(xk_smooth(:,k+1)-xk_minus(:,k+1));

P_smooth(k,:)=(diag(pk_smooth(:,:,k)))';%save smoothed covariance

end %Smoothing loop Ends

K_rts(:,:,1)=(P0*Khi(:,:,1)'/pk_minus(:,:,1));
P0s=P0-(K_rts(:,:,1)*(pk_minus(:,:,1)-pk_smooth(:,:,1))*K_rts(:,:,1)');
x0s=x0+K_rts(:,:,1)*(xk_smooth(:,1)-xk_minus(:,1));
end

if smooth_option==2

for k = N-1:-1:1%Smoothing processing loop Starts

if k==N-1 ,l_fut=zeros(total_sts,1);L_fut=zeros(total_sts,total_sts); end
if k<N-1 ,l_fut=lambda_tilda;L_fut=Lambda_tilda; end

C=eye(total_sts)-(K(:,:,k)*H(:,:,k));
lambda=Khi(:,:,k+1)'*l_fut;Lambda=Khi(:,:,k+1)'*L_fut*Khi(:,:,k+1);

lambda_tilda=(lambda-(H(:,:,k)'*invS(:,:,k)*(innov(:,k)+S(:,:,k)*K(:,:,k)'*lambda)));
Lambda_tilda=C'*Lambda*C+H(:,:,k)'*invS(:,:,k)*H(:,:,k);

%pk_smooth(:,:,k)=pk_minus(:,:,k)-(pk_minus(:,:,k)*Lambda_tilda*pk_minus(:,:,k));
%xk_smooth(:,k)=xk_minus(:,k)-(pk_minus(:,:,k)*lambda_tilda);
%P_smooth(k,:)=(diag(pk_smooth(:,:,k)))';%save smoothed covariance

pk_smooth(:,:,k)=pk_plus(:,:,k)-pk_plus(:,:,k)*Lambda*pk_plus(:,:,k);
xk_smooth(:,k)=xk_plus(:,k)-(pk_plus(:,:,k)*lambda);
P_smooth(k,:)=(diag(pk_smooth(:,:,k)))';%save smoothed covariance
end %Smoothing processing loop End

P0s=P0-P0*Khi(:,:,1)'*Lambda_tilda*Khi(:,:,1)*P0;
x0s=x0-(P0*Khi(:,:,1)'* lambda_tilda);

end
```

```
for k=N:-1:1
if k==1,p_prev=P0;else p_prev=pk_plus(:,:,k-1); end
P_fow(:,:,k)=(eye(total_sts)-K(:,:,k)*H(:,:,k))*Khi(:,:,k)*p_prev;
P_lag(:,:,k)=(eye(total_sts)+(pk_smooth(:,:,k)-...
    pk_plus(:,:,k))*inv(pk_plus(:,:,k)))*P_fow(:,:,k);
end

%%---------------------------------Start of Newton method--------------------------------------------------

function [crb_est,theta_est,R_est_diag,X_est] = Gauss_Newton(z)

global m_noise theta N np nm sys

perturb=10^-5;e=eye(np);nr_iter_max=20;
J=NaN(nm,np,N);parameter_update=zeros(nr_iter_max,np);
theta_initial=theta-0.005*theta;

for nr_iter=1:nr_iter_max

if nr_iter==1
R=0.5*diag(m_noise);
parameter=theta_initial;
[~,r_perturb0]=J_function(parameter,R,z);
else
R=R_est;
parameter=parameter_update(nr_iter-1,:)';
end

for i=1:np
[~,r_perturb1]=J_function(parameter+perturb*e(:,i),R,z);
J(1:nm,i,1:N)=(r_perturb1-r_perturb0)/(perturb);
end

grad=0;hessian=0;
for k=1:N
grad=grad+(J(:,:,k)'*inv(R)*r_perturb0(:,k));
hessian=hessian+(J(:,:,k)'*inv(R)*J(:,:,k));
end

inv_hessian=inv(hessian);
inv_hessian=inv_hessian-diag(diag(inv_hessian))+diag(abs(diag(inv_hessian)));
parameter_update(nr_iter,:)=parameter-(inv_hessian*grad);
```

```matlab
if nr_iter<nr_iter_max
[~,r_perturb0,X_est]=J_function(parameter_update(nr_iter,:)',R,z);
R_est=cov(r_perturb0');
if sys==5 || sys==6
R_est=diag(diag(cov(r_perturb0')));
end
end


end


crb_est=(diag(inv_hessian));
theta_est=parameter_update(nr_iter_max,:);
R_est_diag=diag(R);
```

%%---------------J function (likelihood) NR optimisation----------------------------

```matlab
function [cost,innov,xd_k] = J_function(theta_est,R,z)

global Xd ns np nm N

innov=zeros(nm,N); xk_deterministic=NaN(ns+np,N);

for k = 1:N

if k==1 , x_prev=[Xd(1,1:ns)';theta_est];end
if k>1 , x_prev=xk_deterministic(:,k-1);end

xk_deterministic(:,k)=state_transition(x_prev,k,k+1,0);
hx=measurement_transition(xk_deterministic(:,k),k,0);
innov(:,k)=z(:,k)-hx;

end

cost=sum(var(innov,1,2)./diag(R));
xd_k=xk_deterministic(1:ns,:);
```

%%%------------------end--

D.2 Real Data Program :

```matlab
function EKF_REAL_NEW(varargin)

tic;clc;clear all;close all;
global  total_sts R0_power N case_study coupling x0_initial SOKF ns np nm dt t D

%%Real data case study no.
%1-Longitudinal dimensional
%2a-Longitudinal non dimensional with up and down elevator input
%2b-Longitudinal non dimensional with up and down elevator input with cubic term
%3-Longitudinal non dimensional with aileron input
%4-Lateral dimensional
%5-Lateral non dimensional
case_study='1';

%1-with coupling,0-without coupling for 2nd case
coupling=1;

%Initial values of P0=10^P0_power,R=2^R0_power,
%%%Q= 10^Q0_power for the first iteration
P0_power=-1;
R0_power=-1;
Q0_power=-1;

iter_max=100;
stop=0*10^-5;

%1-Scaleup,2-IIM,3-P0smoth,
P0_method=1;
%1-P0=cov(S P)=[0,0;0,NZ],2-diag,3-full,
P0_option=1;

%1-EM(Bavdekar=z-h(xs)+...),2-MS(z-h(x+)+...),3-MT(z-h(x-)-...),4-(z-h(xd))
R_method=1;
%1-EM(Bavdekar(xs(k)-f(xs(k-1)+...),2-DSDT(xs(k)-xd(k)+...),
%%3-MT(x(+)-x(-)+...),4-MS(K(z-h(x-)(z-h(x-)'K')
Q_method=1;
%1-Q=cov(S,P)=[NZ,0;0,0],2-diag,3-full
Q_option=1;
%1-diag,2-full
R_option=1;
```

```matlab
if strcmp(case_study,'1')
D=dlmread('case1.dat');
N=354;ns=3;nm=4;np=9;
Z=D(1:N+1,2:nm+1)';
x0_initial=[Z(1:ns,1);-0.46;-3.5;-0.47;-0.05;-5.74;0.08;0.16;-0.001;1;0];
end

if strcmp(case_study,'2a')
D=dlmread('case2.dat');
N=175;ns=3;nm=5;np=12;
Z=D(1:N+1,2:nm+1)';
x0_initial=[Z(1:ns,1);4;0.24;0.17;-0.48;-17;-0.9;-0.05;-0.02;0.175;-0.3;-0.083;-0.015];
end

if strcmp(case_study,'2b')
D=dlmread('case2.dat');
N=175;ns=3;nm=5;np=13;
Z=D(1:N+1,2:nm+1)';
x0_initial=[Z(1:ns,1);4;0.24;0.17;-0.48;-17;-0.9;-0.05;-0.02;0.175;-0.3;0.03;-0.083;-0.015];
end

if strcmp(case_study,'3')
D=dlmread('case3.dat');
N=375;ns=3;nm=4;np=10;
Z=D(1:N+1,2:nm+1)';
x0_initial=1*[Z(1:ns,1);4;0.15;0.2;-0.5;-11.5;-5;-1.38;-0.06;-0.01;0.2];
end

if strcmp(case_study,'4')
D=dlmread('case4.dat');
N=234;ns=4;nm=7;np=21;
Z=D(1:N+1,2:nm+1)';
x0_initial=[Z(1:ns,1);-0.04;0.01;0.0001;0.01;0.0001;-16.8;-0.24;0.4;12;20; ...
0.0001;0.0001;1.5;-0.003;-0.04;0.35;-2.5;0.0001;0.2;0.01;0.001];
end

if strcmp(case_study,'5')
D=dlmread('case5.dat');
N=425;ns=4;nm=5;np=20;
Z=D(1:N+1,2:nm+1)';
x0_initial=[Z(1:ns,1);-0.5;0.1;-0.01;0.01;-0.35;0.01;0.06;0.01;-0.002;0.002; ...
0.07;-0.055;-0.05;0.003;-0.04;0.0068;-0.025;0.5;-1;0.005];
end

%x0_initial=[Z(1:ns,1);zeros(np,1)];
```

```matlab
%Not Used
IEKF=0;
SOKF=0;

e_max=1;
R2term=1;
Q2term=1;
Scale_up=N;
last_n=round(0.5*N);
total_sts=ns+np;
t=D(:,1)';dt=t(2)-t(1);

display(['No. of data points N =' num2str(N)]);
display(['No. of states=' num2str(ns)]);
display(['No. of parameters=' num2str(np)]);
display(['No. of measurements=' num2str(nm)]);
display(['size(X0)=' num2str(total_sts) 'x' num2str(1)]);
display(['size(P0)=' num2str(total_sts) 'x' num2str(total_sts)]);
display(['size(Q)=' num2str(total_sts) 'x' num2str(total_sts)]);
display(['size(R)=' num2str(nm) 'x' num2str(nm)]);
display(['size(F)=' num2str(total_sts) 'x' num2str(total_sts)]);
display(['size(H)=' num2str(nm) 'x' num2str(total_sts)]);
display(['size(K)=' num2str(total_sts) 'x' num2str(nm)]);

%%Pre-allocating memory with zeros as values
StoreX0=zeros(e_max,total_sts);X_smooth=[];StoreP0=StoreX0;K_all=[];
P_EKF=zeros(e_max,total_sts);StoreR=zeros(e_max,nm);StoreQ=StoreX0;
S1diag=zeros(N,nm);P_smooth=[];S2diag=S1diag;S3diag=S1diag;Blob_X0=[];
Blob_P0=[];J5=zeros(e_max,1);J1=zeros(e_max,nm);J2=J1;J3=J1;J4=J1;
Save_P0=[];Save_X=[];Save_Q=[];J6=zeros(e_max,ns);J7=J6;J8=J6;
C1diag=zeros(N,ns);C2diag=C1diag;C3diag=C1diag;Save_R=[];
X_comb=[];P_comb=[];filename='workspace_real';


xk_prior=zeros(total_sts,N);pk_prior=zeros(total_sts,total_sts,N);
hx1=zeros(nm,N);H1=zeros(nm,total_sts,N);hx2=hx1;hx3=hx1;hx4=hx1;
invS=zeros(nm,nm,N);xk_posterior=xk_prior;pk_minus=zeros(N,total_sts);
pk_plus=pk_minus;K=zeros(total_sts,nm,N);S1=invS;S2=S1;S3=S1;fxs=xk_prior;
pk_posterior=pk_prior;F=pk_prior;Fs=F;Fd=F;innov=hx1;filter_residue=hx1;
smooth_residue=hx1;xd_est=xk_prior;w2=xk_prior;tx=t(2:N+1);xx=repmat(tx,2,1);
xx=xx(:);fail=0;H3=H1;H2=H1;dynamical_residue=hx1;

for e=1:e_max
```

```matlab
z=Z(:,2:end);flag=1;
Zcost=zeros(iter_max,nm,4);L=zeros(iter_max,1);Xcost=zeros(iter_max,ns,3);

for iter=1:iter_max

if iter==1
x0=x0_initial(1:total_sts);
P0=10^P0_power*eye(total_sts);
Q=10^Q0_power*diag([ones(1,ns) 0*ones(1,np)]);
R=2^R0_power*eye(nm);
end

if iter>1
x0=x0_est;
P0=P0_est;
R=R_est;
Q=Q_est;
end

% Q=1.0*diag([Q_inj(e,:) 0*ones(1,np)]);
% R=1.0*diag(R_inj(e,:));
% Q=diag([ones(1,ns)*10^-10 zeros(1,np)]);

%Forcing the diagonal values to be +ve
Q=Q-diag(diag(Q))+diag(abs(diag(Q)));
R=R-diag(diag(R))+diag(abs(diag(R)));
P0=P0-diag(diag(P0))+diag(abs(diag(P0)));

E_iter=[e,iter];
display(E_iter,'ensemble , iter');
display(x0','x0');display(diag(P0)','P0');
display(diag(R)','R');display(diag(Q)','Q');

for k = 1:N %Data processing loop Starts

if k==1 , x_prev=x0; p_prev=P0;end
if k>1 , x_prev=xk_posterior(:,k-1);p_prev=pk_posterior(:,:,k-1);end

[fx,F(:,:,k)]=state_transition(x_prev,k,k+1,p_prev);

%%%%Kalman time update equations
xk_prior(:,k)=fx;
pk_prior(:,:,k) =F(:,:,k)*p_prev*F(:,:,k)'+Q;

epsilon=1;i=0;%%Iterated EKF
```

```
while epsilon>10^-3

i=i+1;
if i==1,x_nominal=xk_prior(:,k);p_nominal=pk_prior(:,:,k);end
if i>1,x_nominal=x_updated;p_nominal=p_updated;end

[hx1(:,k),H1(:,:,k),l1]=measurement_transition(x_nominal,k,p_nominal);
S1(:,:,k)=H1(:,:,k)*pk_prior(:,:,k)*H1(:,:,k)'+R+l1;invS(:,:,k)=inv(S1(:,:,k));
innov(:,k) =z(:,k)- hx1(:,k)-H1(:,:,k)*(xk_prior(:,k)-x_nominal);

%%%%Kalman measurement update equations
K(:,:,k) =pk_prior(:,:,k)*H1(:,:,k)'*invS(:,:,k);
p_updated=(eye(total_sts)-K(:,:,k)*H1(:,:,k))*pk_prior(:,:,k);
x_updated = xk_prior(:,k) + K(:,:,k)*innov(:,k);

if IEKF==1,epsilon=norm(x_updated-x_nominal);else epsilon=0;end
if i>100,fail=fail+1;break;end
end

xk_posterior(:,k) = x_updated;pk_posterior(:,:,k)=p_updated;

end %Data processing loop End

for k=1:N,L(iter,:)=L(iter,:)+innov(:,k,1)'*invS(:,:,k)*innov(:,k)+1*log(det(S1(:,:,k)));end
if iter>1&&flag==1,delta=abs((L(iter)-L(iter-1))/L(iter));if delta<stop,flag=0;end,end

if flag==1

[xk_smooth,Pk_smooth,pk_smooth,x0smooth,P0smooth,Plag]=...
smooth(xk_posterior,pk_posterior,xk_prior,pk_prior,F,H1,K,S1,invS,innov,x0,P0);
x0_est=[z(1:ns,1);xk_posterior(ns+1:end,N)];

for k=1:N
if k==1,x_prev=x0_est;p_prev=P0smooth;xd_prev=x0_est;else
x_prev=xk_smooth(:,k-1);p_prev=pk_smooth(:,:,k-1);xd_prev=xd_est(:,k-1);end
[fxs(:,k),Fs(:,:,k)]=state_transition(x_prev,k,k+1,p_prev);

[xd_est(:,k),Fd(:,:,k)]=state_transition(xd_prev,k,k+1,0);
fxd(:,k)=1*xd_est(:,k)+Fd(:,:,k)*(x_prev-1*xd_prev);
w2(:,k)=xk_smooth(:,k)-1*xd_est(:,k)-Fd(:,:,k)*(x_prev-1*xd_prev);
end

for k=1:N,
[hx2(:,k),H2(:,:,k)]=measurement_transition(xk_posterior(:,k),k,pk_posterior(:,:,k));
[hx3(:,k),H3(:,:,k)]=measurement_transition(xk_smooth(:,k),k,pk_smooth(:,:,k));
```

```matlab
[hx4(:,k)]=measurement_transition(xd_est(:,k),k,0);
dynamical_residue(:,k)= z(:,k)-hx4(:,k);
filter_residue(:,k) = z(:,k)-hx2(:,k);
smooth_residue(:,k) = z(:,k)-hx3(:,k);

pk_minus(k,:)=diag(pk_prior(:,:,k)); pk_plus(k,:)=diag(pk_posterior(:,:,k));
S2(:,:,k)=-H2(:,:,k)*pk_posterior(:,:,k)*H2(:,:,k)'+R;S2diag(k,:)=abs(diag(S2(:,:,k)));
S3(:,:,k)=-H3(:,:,k)*pk_smooth(:,:,k)*H3(:,:,k)'+R;S3diag(k,:)=abs(diag(S3(:,:,k)));
S1diag(k,:)=diag(S1(:,:,k));
end

%%Estimation of Q,R
w1=xk_smooth-fxs;
Qcap1=0;Qcap2=0;Rcap=0;
for k=1:N
if k==1,
    x_prev=x0_est;p_prev=P0smooth;else x_prev=xk_smooth(:,k-1);
    p_prev=pk_smooth(:,:,k-1);
end
Qcap1=Qcap1+w1(:,k)*w1(:,k)'+Q2term*(Fs(:,:,k)*p_prev*Fs(:,:,k)'+...
    pk_smooth(:,:,k)-Plag(:,:,k)*Fs(:,:,k)'-Fs(:,:,k)*Plag(:,:,k)');
Qcap2=Qcap2+w2(:,k)*w2(:,k)'+Q2term*(Fd(:,:,k)*p_prev*Fd(:,:,k)'+...
    pk_smooth(:,:,k)-Plag(:,:,k)*Fd(:,:,k)'-Fd(:,:,k)*Plag(:,:,k)');
Rcap=Rcap+smooth_residue(:,k)*smooth_residue(:,k)'+...
    R2term*(H3(:,:,k)*pk_smooth(:,:,k)*H3(:,:,k)');
end

Q_MT=0;w3=xk_posterior-xk_prior;
R_MT=0;R_MS=0;
for k=N-last_n+1:N
Q_MT=Q_MT+w3(:,k)*w3(:,k)'-Q2term*((pk_prior(:,:,k)-Q)-pk_posterior(:,:,k));
R_MT=R_MT+innov(:,k)*innov(:,k)'-R2term*(H1(:,:,k)*(pk_prior(:,:,k))*H1(:,:,k)');
R_MS=R_MS+filter_residue(:,k)*filter_residue(:,k)'+...
    R2term*(H2(:,:,k)*pk_posterior(:,:,k)*H2(:,:,k)');
end

if Q_method==1,Q_full=1/N*Qcap1;end
if Q_method==2,Q_full=K(:,:,N)*cov(innov(:,N-last_n+1:N)')*K(:,:,N)';end
if Q_method==3,Q_full=1/last_n*Q_MT;end
if Q_method==4,Q_full=1/N*Qcap2;end

if Q_option==1,Q_est=Q_full;Q_est(ns+1:total_sts,:)=0;
    Q_est(:,ns+1:total_sts)=0;Q_est=diag(abs(diag(Q_est)));end
if Q_option==2,Q_est=diag(abs(diag(Q_full)));end
if Q_option==3,Q_est=Q_full;end
```

```
if R_method==1,R_full=1/N*Rcap;end
if R_method==2,R_full=1/last_n*R_MS;end
if R_method==3,R_full=1/last_n*R_MT;end
if R_method==4,R_full=cov(dynamical_residue');end

if R_option==1,R_est=diag(abs(diag(R_full)));end
if R_option==2,R_est=R_full;end

IM=0;%%Estimation of P0
for k=1:N,IM=IM+F(:,:,k)'*H1(:,:,k)'*inv(R)*H1(:,:,k)*F(:,:,k);end

if rcond(IM)<10^-10
IM=0;%%Estimation of P0
for k=1:N,
    IM=IM+F(1:ns,1:ns,k)'*H1(:,1:ns,k)'*inv(R)*H1(:,1:ns,k)*F(1:ns,1:ns,k);end
    IM=[diag(abs(diag(IM))),zeros(ns,np);zeros(np,ns),N*eye(np)];
end

if P0_method==1,P0_full=Scale_up*pk_posterior(:,:,N);end
if P0_method==2,P0_full=inv(1/N*IM);end
if P0_method==3,P0_full=P0smooth;end

if P0_option==1,P0_est=P0_full;P0_est(1:ns,:)=0;P0_est(:,1:ns)=0;end
if P0_option==2,P0_est=diag(diag(P0_full));end
if P0_option==3,P0_est=P0_full;end

for k=1:N
if k==1,p_prev=P0smooth;else p_prev=pk_smooth(:,:,k-1);end
Zcost(iter,:,1)=Zcost(iter,:,1)+innov(:,k)'.^2./S1diag(k,:);
Zcost(iter,:,2)=Zcost(iter,:,2)+filter_residue(:,k)'.^2./S2diag(k,:);
Zcost(iter,:,3)=Zcost(iter,:,3)+smooth_residue(:,k)'.^2./S3diag(k,:);
Zcost(iter,:,4)=Zcost(iter,:,4)+dynamical_residue(:,k)'.^2;

C1diag(k,:)=-diag(Fs(1:ns,1:ns,k)*p_prev(1:ns,1:ns)*Fs(1:ns,1:ns,k)'+pk_smooth(1:ns,1:ns,k)-...
    Plag(1:ns,1:ns,k)*Fs(1:ns,1:ns,k)'-Fs(1:ns,1:ns,k)*Plag(1:ns,1:ns,k)'-Q(1:ns,1:ns))';
C2diag(k,:)=-diag(Fs(1:ns,1:ns,k)*p_prev(1:ns,1:ns)*Fs(1:ns,1:ns,k)'+pk_smooth(1:ns,1:ns,k)-...
    Plag(1:ns,1:ns,k)*Fs(1:ns,1:ns,k)'-Fs(1:ns,1:ns,k)*Plag(1:ns,1:ns,k)'-Q(1:ns,1:ns))';
C3diag(k,:)=diag(pk_prior(1:ns,1:ns,k)-pk_posterior(1:ns,1:ns,k))';

Xcost(iter,:,1)=Xcost(iter,:,1)+w1(1:ns,k)'.^2./C1diag(k,:);
Xcost(iter,:,2)=Xcost(iter,:,2)+w2(1:ns,k)'.^2./C2diag(k,:);
Xcost(iter,:,3)=Xcost(iter,:,3)+w3(1:ns,k)'.^2./C3diag(k,:);

end
```

```matlab
end

if flag==0;Zcost(iter,:,:)=Zcost(iter-1,:,:);Xcost(iter,:,:)=Xcost(iter-1,:,:);end

Save_X=[Save_X;x0'];Save_P0=[Save_P0;diag(P0)'];
Save_R=[Save_R;diag(R)'];Save_Q=[Save_Q;diag(Q(:,:,1))'];

for s=1:total_sts,
X_temp=[xk_prior(s,:);xk_posterior(s,:)];X_comb_iter(:,s)=X_temp(:);
P_temp=[pk_minus(:,s)';pk_plus(:,s)'];P_comb_iter(:,s)=P_temp(:);
end

X_comb=[X_comb;X_comb_iter;NaN(1,total_sts)];
P_comb=[P_comb;P_comb_iter;NaN(1,total_sts)];
X_smooth=[X_smooth;xk_smooth';NaN(1,total_sts)];
P_smooth=[P_smooth;Pk_smooth;NaN(1,total_sts)];
K_all=[K_all;reshape(K(:),total_sts*nm,[])';NaN(1,total_sts*nm)];
Blob_X0=[Blob_X0;x0'];Blob_P0=[Blob_P0;diag(P0)'];

end

%%saving ensemble values
StoreX0(e,:)=x0;StoreP0(e,:)=diag(P0);StoreQ(e,:)=diag(Q(:,:,1));
StoreR(e,:)=diag(R);P_EKF(e,:)=diag(pk_posterior(:,:,N));

J1(e,:)=Zcost(iter_max,:,1)/N;J2(e,:)=Zcost(iter_max,:,2)/N;J3(e,:)=Zcost(iter_max,:,3)/N;
J4(e,:)=Zcost(iter_max,:,4)/N;J5(e,:)=L(iter_max,:)/N;
J6(e,:)=Xcost(iter_max,:,1)/N;J7(e,:)=Xcost(iter_max,:,2)/N;J8(e,:)=Xcost(iter_max,:,3)/N;
end

Q_mean_EKF=mean(StoreQ(:,1:ns),1);
R_mean_EKF=mean(StoreR,1);
Theta_EKF=StoreX0(:,ns+1:total_sts);
Theta_mean_EKF=mean(Theta_EKF,1);
Sigma_EKF=sqrt(P_EKF);
Sigma_mean_EKF=mean(Sigma_EKF(:,ns+1:end),1);
J1_8_mean=[sum(mean(J1,1)) ,sum(mean(J2,1)),sum(mean(J3,1)),sum(mean(J4,1)),...
sum(mean(J5,1)),sum(mean(J6,1)),sum(mean(J7,1)),sum(mean(J8,1))];

I=inv(pk_posterior(ns+1:end,ns+1:end,N));
for i=1:np
for j=1:np
C(i,j)=I(i,j)/sqrt(I(i,i)*I(j,j));
end
end
```

```matlab
display('---------Displaying results------------');

Theta_correlation=C

Theta_mean_EKF
Sigma_mean_EKF
R_mean_EKF
Q_mean_EKF
J1_8_mean


%%-------Plotting figures-------------------------------

%generate x-axis values (time points) and legend entries for plotting
t1=[];for l=1:iter_max, y=t(N+1)*(l-1):dt:t(N+1)*l;t1=[t1 y NaN];end

tp1=t(2):dt:t(N+1)*iter_max+dt*(iter_max-1);t2=[];
for l=1:iter_max,tp02=repmat(tp1(N*(l-1)+1:N*l),2,1);
    tp2=[tp02(:)' NaN tp1(~(N*(l-1)+1:N*l))];t2=[t2 tp2];end

y1=t(2):dt:t(N+1)*iter_max+dt*(iter_max-1);t3=[];
for l=1:iter_max,t3=[t3 y1(N*(l-1)+1:N*l) NaN y1(~(N*(l-1)+1:N*l))];end

t4=t(1):N*dt:t(1)+N*dt*iter_max;t4=t4(1:end-1);t5=1:iter_max;

string1=[];string2=[];string3=[];string4=[];string5=[];string6=[];

for p=1:np
string1=char(string1,['\bf\theta' num2str(p)]);
string3=char(string3,['P0 \bf\theta' num2str(p)]);
end

for s=1:ns+np
string4=char(string4,['Q ' num2str(s) num2str(s)]);
string5=char(string5,['P0 ' num2str(s) num2str(s)]);
end

for m=1:nm
string2=char(string2,['R' num2str(m) num2str(m)]);
for s=1:total_sts
string6=char(string6,['K ' num2str(s) num2str(m)]);
end
end
```

```matlab
r=smooth_residue;w=w1;
Err_Th=abs(Save_X(:,ns+1:end)-repmat(Save_X(end-1,ns+1:end),size(Save_X,1),1));
Err_R=abs(Save_R-repmat(Save_R(end-1,:),size(Save_R,1),1));
Err_Q=abs(Save_Q(:,1:ns)-repmat(Save_Q(end-1,1:ns),size(Save_Q,1),1));
Err_P0=abs(Save_P0(:,ns+1:end)-repmat(Save_P0(end-1,ns+1:end),size(Save_P0,1),1));


toc,save(filename),
evalin('base',sprintf('load("%s")',filename))%save and load workspace

%%---------EKF Plots------------------------


for s=1:ns
n=(s-1)*2+1;
[acr2,lag_s2]=xcorr(w(s,:)','coeff');
figure(1),subplot(ns,2,n),plot(t(2:N+1),w(s,:)','-','linewidth',2),grid on,
title(['process noise = ' num2str(s)],'fontsize',14)
set(gca,'fontsize',14)

subplot(ns,2,n+1),plot(lag_s2(lag_s2>=0),acr2(lag_s2>=0,:),'-','linewidth',2),grid on
title('autocorrelation versus time lag','fontsize',14),grid on
set(gca,'fontsize',14)
end

for m=1:nm
n=(m-1)*2+1;
[acr2,lag_s2]=xcorr(r(m,:)','coeff');
figure(2),subplot(nm,2,n),plot(t(2:N+1),r(m,:)','-','linewidth',2),grid on,
title(['measurement = ' num2str(m)],'fontsize',14)
set(gca,'fontsize',14)

subplot(nm,2,n+1),plot(lag_s2(lag_s2>=0),acr2(lag_s2>=0,:),'-','linewidth',2),grid on
title('autocorrelation versus time lag','fontsize',14),grid on
set(gca,'fontsize',14)
end


figure,semilogy(t5,Save_Q(:,1:ns),'-',t5,Save_R,'*-','linewidth',2),grid on
legend(char(string4(2:ns+1,:),string2(2:end,:))),
title('Variation of R and Q with iterations','fontsize',14),
set(gca,'fontsize',14)


figure,semilogy(t5,abs(sum(Zcost(:,:,1)/N,2)),t5,abs(sum(Zcost(:,:,2)/N,2)),t5,abs(sum(Zcost(:,:,3)/N,2)),...
```

```
     t5,abs(sum(Zcost(:,:,4)/N,2)),t5,abs(sum(L/N,2)),...
t5,abs(sum(Xcost(:,:,1)/N,2)),t5,abs(sum(Xcost(:,:,2),2)),t5,abs(sum(Xcost(:,:,3),2)),'linewidth',2),
legend('J1','J2','J3','J4','J5','J6','J7','J8'),grid on
title('Variation of different costs (J1-J8) with iterations','fontsize',14)
set(gca,'fontsize',14)

figure,semilogy(t5,abs(Save_X(:,ns+1:end)),'-','linewidth',2),hold on
semilogy(t5,Save_P0(:,ns+1:end),'*-','linewidth',2),grid on
legend(char(string1(2:end,:),string3(2:end,:))),
title('Variation of parameter and its initial covariance (P0) with iterations','fontsize',14)
set(gca,'fontsize',14)

for s=1:nm
figure,subplot(3,1,1),plot(t(2:end),sqrt(S1diag(:,s)),'b-',t(2:end),...
   -sqrt(S1diag(:,s)),'b-',t(2:end),innov(s,:)','r-','linewidth',2),grid on,
title(['innovations outside {\sigma} bound = ' ...
   num2str(100*length(find(sqrt(S1diag(:,s))<abs(innov(s,:)'))/N) ' %'],...
'fontsize',14),set(gca,'fontsize',14)

subplot(3,1,2),plot(t(2:end),sqrt(S2diag(:,s)),'b-',t(2:end),...
   -sqrt(S2diag(:,s)),'b-',t(2:end),filter_residue(s,:)','r-','linewidth',2),grid on,
title(['filter-residue outside {\sigma} bound = '...
   num2str(100*length(find(sqrt(S2diag(:,s))<abs(filter_residue(s,:)'))/N) ' %'],'fontsize',14)
set(gca,'fontsize',14)

subplot(3,1,3),plot(t(2:end),sqrt(S3diag(:,s)),'b-',t(2:end),-sqrt(S3diag(:,s)),'b-',...
   t(2:end),smooth_residue(s,:)','r-','linewidth',2),grid on,
title(['smooth-residue outside {\sigma} bound = ' ...
   num2str(100*length(find(sqrt(S3diag(:,s))<abs(smooth_residue(s,:)'))/N) ' %'],'fontsize',14),
set(gca,'fontsize',14)
end

for s=1:nm
figure,plot(tx,hx4(s,:),'-',tx,hx2(s,:),'-',tx,hx3(s,:),'-',tx,z(s,:),'go','linewidth',2),grid on,
legend('h(Xd)','h(X+)','h(Xs)','z'),
title(['Comparison of the predicted dynamics, posterior, smoothed and the measurement '...
     num2str(s) ],'fontsize',14),
set(gca,'fontsize',14)
end

for p=1:np
figure,subplot(2,1,1),plot(t3,X_smooth(:,ns+p),'r-', ...
t2,X_comb(:,ns+p),'b-',t4,Blob_X0(:,ns+p),'go','linewidth',2),grid on
legend(char('\bf\theta smooth','\bf\theta filtered','\bf\theta initial')),
title(['\bf{\theta} = ' num2str(p) ' versus iterated time instants'],'fontsize',14),
```

```matlab
set(gca,'fontsize',14)

subplot(2,1,2),semilogy(t3,P_smooth(:,ns+p),'r-',t2,P_comb(:,ns+p),'b-',...
    t4,Blob_P0(:,ns+p),'go','linewidth',2),hold on,
legend(char('P smooth','P filtered','P0')),grid on
title(['P\bf{\theta} = ' num2str(p) ' versus iterated time instants'],'fontsize',14),
set(gca,'fontsize',14)
end

figure,plot(t3,K_all,'linewidth',2),grid on,legend(string6(2:end,:))
title('Gain variations through cumulative time with iterations ','fontsize',14)
set(gca,'fontsize',14)

%%%%--------------------System Dynamics--------------------------

function dx=dynamics(t1,x,P)

global  ns np case_study SOKF coupling

if strcmp(case_study,'1')
d=get_u(t1);u=d(9);
dx=[x(ns+1)*x(1)+x(2)-0.00221*x(3)+x(ns+4)*u+x(ns+6);x(ns+2)*x(1)+...
    x(ns+3)*x(2)+x(ns+5)*u+x(ns+7);0.9916*x(2)+x(ns+8); zeros(np,1)];
end

if strcmp(case_study,'2a')

data=get_u(t1);

u=data(9);
am=data(2);
beta=data(10);
pm=data(11);
rm=data(12);
phi=data(13);
Ixx=4142.9;Iyy=3922.4;Izz=7642.5;

CNalpha=x(ns+1);
CNDelta=x(ns+2);
CLzero=x(ns+3);
CMalpha=x(ns+4);
CMq=x(ns+5);
CMDelta=x(ns+6);
CMzero=x(ns+7);
Thetazero_dot=x(ns+8);
```

```matlab
CNzero=x(ns+9);
CAalpha=x(ns+10);
CAdelta=x(ns+11);
CAzero=x(ns+12);

CN=CNalpha*x(1)+CNDelta*u+CNzero;
CA=CAalpha*x(1)+CAdelta*u+CAzero;
CL=CN*cos(x(1))-CA*sin(x(1))+CLzero;

qsmv=83.08*184/(172.667*403.1);
qsci=83.08*184*5.58/3922.4;
c2v=5.58/(2*403.1);

if coupling==0
dx=[-qsmv*CL+x(2)+32.2/403.1;qsci*(CMalpha*x(1)+CMq*c2v*x(2)+CMDelta*u+CMzero);
x(2)+Thetazero_dot; zeros(np,1)];
end

if coupling==1
dx=[-qsmv*CL+x(2)+32.2/403.1*(cos(phi)*cos(x(3))*cos(am)+...
   sin(x(3)))-beta*(pm*cos(am)+rm*sin(am));
qsci*(CMalpha*x(1)+CMq*c2v*x(2)+CMDelta*u+CMzero+(Izz-Ixx)/Iyy*rm*pm);
x(2)*cos(phi)-rm*sin(phi)+Thetazero_dot; zeros(np,1)];
end
end


if strcmp(case_study,'2b')

data=get_u(t1);

u=data(9);
am=data(2);
beta=data(10);
pm=data(11);
rm=data(12);
phi=data(13);
Ixx=4142.9;Iyy=3922.4;Izz=7642.5;

CNalpha=x(ns+1);
CNDelta=x(ns+2);
CLzero=x(ns+3);
CMalpha=x(ns+4);
CMq=x(ns+5);
CMDelta=x(ns+6);
```

```matlab
CMzero=x(ns+7);
Thetazero_dot=x(ns+8);
CNzero=x(ns+9);
CAalpha=x(ns+10);
CAalpha2=x(ns+11);
CAdelta=x(ns+12);
CAzero=x(ns+13);

CN=CNalpha*x(1)+CNDelta*u+CNzero;
CA=CAalpha*x(1)+CAalpha2*x(1)^2+CAdelta*u+CAzero;
CL=CN*cos(x(1))-CA*sin(x(1))+CLzero;

qsmv=83.08*184/(172.667*403.1);
qsci=83.08*184*5.58/3922.4;
c2v=5.58/(2*403.1);

if coupling==0
dx=[-qsmv*CL+x(2)+32.2/403.1;qsci*(CMalpha*x(1)+CMq*c2v*x(2)+CMDelta*u+CMzero);
x(2)+Thetazero_dot; zeros(np,1)];
end

if coupling==1
dx=[-qsmv*CL+x(2)+32.2/403.1*(cos(phi)*cos(x(3))*cos(am)+sin(x(3)))-...
   beta*(pm*cos(am)+rm*sin(am));
qsci*(CMalpha*x(1)+CMq*c2v*x(2)+CMDelta*u+CMzero+(Izz-Ixx)/Iyy*rm*pm);
x(2)*cos(phi)-rm*sin(phi)+Thetazero_dot; zeros(np,1)];
end
end

if strcmp(case_study,'3')

u=get_u(t1);
vm=u(8);
dele=u(9);
beta=u(10);
pm=u(11);
rm=u(12);
phi=u(13);
qbar=u(16);
am=u(2)-(0.0279/vm)*u(3);

S=184;m=196;g=32.2;cbar=5.58;
Iyy=3953.2;Ixx=6892.7;Izz=10416.4;

CLalpha=x(ns+1);
```

```matlab
CLdelta=x(ns+2);
CLzero=x(ns+3);
CMalpha=x(ns+4);
CMq=x(ns+5);
CMalpha_dot=x(ns+6);
CMdelta=x(ns+7);
CMzero=x(ns+8);
thetazero_dot=x(ns+9);
CNzero=x(ns+10);

qsmv=qbar*S/(m*vm*cos(beta));
qsmg=qbar*S/(m*g);c2v=cbar/(2*vm);
CL=CLzero+CLalpha*x(1)+CLdelta*dele;
CN=CLalpha*x(1)+CLdelta*dele+CNzero;

qsci=qbar*S*cbar/Iyy;
adot=-qsmv*CL+x(2)-tan(beta)*(pm*cos(am)+rm*sin(am))+...
   g/(vm*cos(beta))*(cos(phi)*cos(x(3))*cos(am)+sin(x(3))*sin(am));
qdot=qsci*(CMzero+CMalpha*x(1)+c2v*(CMq*x(2)+CMalpha_dot*adot)+...
   CMdelta*dele)+((Izz-Ixx)/Iyy)*rm*pm;

dx=[adot; qdot;x(2)*cos(phi)-rm*sin(phi)+thetazero_dot; zeros(np,1)];

end

if strcmp(case_study,'4')
data=get_u(t1);u=data(9:10);
dx=[x(ns+1)*x(1)+x(ns+2)*x(2)+0.0069*x(3)-x(4)+x(ns+3)*u(1)+x(ns+4)*u(2)+x(ns+5);...
x(ns+6)*x(1)+x(ns+7)*x(2)+x(ns+8)*x(4)+x(ns+9)*u(1)+x(ns+10)*u(2)+x(ns+11);x(2)+x(ns+12);...
x(ns+13)*x(1)+x(ns+14)*x(2)+x(ns+15)*x(4)+x(ns+16)*u(1)+x(ns+17)*u(2)+x(ns+18);zeros(np,1)];
end

if strcmp(case_study,'5')

u=get_u(t1);

delta_a=u(9);
delta_r=u(10);
theta_m=u(13);
qm=u(12);
am=u(14);

qbar=865.3;S=9.3;m=387.7;V=39.41;b=6.81;g=9.81;
Ixx=314;Iyy=488;Izz=698;Ixz=69;
Kz=0.305;Kx=2.73;
```

```
CYbeta=x(ns+1);
CYdelta_r=x(ns+2);
Beta_zero=x(ns+3);
CLbeta=x(ns+4);
CLp=x(ns+5);
CLr=x(ns+6);
CLdelta_a=x(ns+7);
CLdelta_r=x(ns+8);
CL_zero=x(ns+9);
Phi_zero=x(ns+10);
CNbeta=x(ns+11);
CNp=x(ns+12);
CNr=x(ns+13);
CNdelta_a=x(ns+14);
CNdelta_r=x(ns+15);
CN_zero=x(ns+16);
CY_zero=x(ns+17);
CYp=x(ns+18);
CYr=x(ns+19);
CYdelta_a=x(ns+20);

qsmv=qbar*S/(m*V);
b2v=b/(2*V);
qsbix=qbar*S*b/Ixx;
qsbiz=qbar*S*b/Izz;
qsmg=qbar*S/(m*g);

beta_dot=qsmv*(CYbeta*x(1)+b2v*CYp*x(2)+b2v*CYr*x(4)+...
   CYdelta_a*delta_a+CYdelta_r*delta_r+Beta_zero)+ ...
x(2)*sin(am)-x(4)*cos(am)+(g/V)*sin(x(3))*cos(theta_m);
dot1=qsbix*(CLbeta*x(1)+b2v*CLp*x(2)+b2v*CLr*x(4)+...
   CLdelta_a*delta_a+CLdelta_r*delta_r+CL_zero)+ ...
((Iyy-Izz)/Ixx)*x(4)*qm+Ixz/Ixx*x(2)*qm;
phi_dot=x(2)+qm*tan(theta_m)*sin(x(3))+...
   x(4)*tan(theta_m)*cos(x(3))+Phi_zero;
dot2=qsbiz*(CNbeta*x(1)+b2v*CNp*x(2)+b2v*CNr*x(4)+...
   CNdelta_a*delta_a+CNdelta_r*delta_r+CN_zero)+ ...
((Ixx-Iyy)/Izz)*x(2)*qm-Ixz/Izz*x(4)*qm;
prdot=inv([1 -Ixz/Ixx;-Ixz/Izz 1])*[dot1;dot2];

dx=[beta_dot;prdot(1);phi_dot;prdot(2);zeros(np,1)];

end
```

```matlab
if norm(P)>0 && SOKF==1
f2term=get_f2term(x,P);
dx=dx+f2term;
end


%%%--------------------------Get control input-----------------------------


function u=get_u(t1)


global t D


flag=1;k=1;
while flag==1
if t1>t(k),k=k+1;else
flag=0;end
end
u=D(k,:);




%%%----------------------State transition function---------------------------


function [x_next,F] = state_transition(x,k,k_next,P)


global  case_study coupling dt t np ns


if strcmp(case_study,'1')
data=get_u(t(k));u=data(9);
A=[x(ns+1) 1 -0.00221 x(1) 0 0 u 0 1 0 0 0;x(ns+2) x(ns+3) 0 0 x(1) x(2) 0 u 0 1 0 0;...
0 0.9916 zeros(1,8) 1 0;zeros(np,ns+np) ];
end


if strcmp(case_study,'2a')


data=get_u(t(k));


u=data(9);
am=data(2);
beta=data(10);
pm=data(11);
rm=data(12);
phi=data(13);
Ixx=4142.9;Iyy=3922.4;Izz=7642.5;


CNalpha=x(ns+1);
```

```
CNDelta=x(ns+2);
CLzero=x(ns+3);
CMalpha=x(ns+4);
CMq=x(ns+5);
CMDelta=x(ns+6);
CMzero=x(ns+7);
Thetazero_dot=x(ns+8);
CNzero=x(ns+9);
CAalpha=x(ns+10);
CAdelta=x(ns+11);
CAzero=x(ns+12);

CN=CNalpha*x(1)+CNDelta*u+CNzero;
CA=CAalpha*x(1)+CAdelta*u+CAzero;
CL=CN*cos(x(1))-CA*sin(x(1))+CLzero;

qsmv=83.08*184/(172.667*403.1);
qsci=83.08*184*5.58/3922.4;
c2v=5.58/(2*403.1);


if coupling==0
A=[-qsmv*(-CN*sin(x(1))+cos(x(1))*CNalpha-CA*cos(x(1))-...
    CAalpha*sin(x(1)) ) 1 0 -qsmv*cos(x(1))*[x(1) u] -qsmv ...
0 0 0 0 0 -qsmv*cos(x(1)) qsmv*sin(x(1))*[x(1) u 1]; ...
qsci*[CMalpha CMq*c2v 0 0 0 0 x(1) x(2)*c2v u 1 zeros(1,5)]; ...
0 1 zeros(1,8) 1 0 0 0 0 ;zeros(np,ns+np)];
end

if coupling==1
A=[-qsmv*(-CN*sin(x(1))+cos(x(1))*CNalpha-CA*cos(x(1))-...
    CAalpha*sin(x(1)) ) 1 32.2/403.1*(-cos(phi)*cos(am)*sin(x(3))+...
    sin(am)*cos(x(3))) -qsmv*cos(x(1))*[x(1) u] -qsmv ...
0 0 0 0 0 -qsmv*cos(x(1)) qsmv*sin(x(1))*[x(1) u 1]; ...
qsci*[CMalpha CMq*c2v 0 0 0 0 x(1) x(2)*c2v u 1 zeros(1,5)]; ...
0 1 zeros(1,8) 1 0 0 0 0 ;zeros(np,ns+np)];
end
end

if strcmp(case_study,'2b')
data=get_u(t(k));
u=data(9);
am=data(2);
beta=data(10);
pm=data(11);
```

```
rm=data(12);
phi=data(13);
Ixx=4142.9;Iyy=3922.4;Izz=7642.5;

CNalpha=x(ns+1);
CNDelta=x(ns+2);
CLzero=x(ns+3);
CMalpha=x(ns+4);
CMq=x(ns+5);
CMDelta=x(ns+6);
CMzero=x(ns+7);
Thetazero_dot=x(ns+8);
CNzero=x(ns+9);
CAalpha=x(ns+10);
CAalpha2=x(ns+11);
CAdelta=x(ns+12);
CAzero=x(ns+13);

CN=CNalpha*x(1)+CNDelta*u+CNzero;
CA=CAalpha*x(1)+CAalpha2*x(1)^2+CAdelta*u+CAzero;
CL=CN*cos(x(1))-CA*sin(x(1))+CLzero;

qsmv=83.08*184/(172.667*403.1);
qsci=83.08*184*5.58/3922.4;
c2v=5.58/(2*403.1);

if coupling==0
A=[-qsmv*(-CN*sin(x(1))+cos(x(1))*CNalpha-CA*cos(x(1))-...
   (CAalpha+2*x(1)*CAalpha2)*sin(x(1)) ) 1 0 -qsmv*cos(x(1))*[x(1) u] -qsmv ...
0 0 0 0 0 -qsmv*cos(x(1)) qsmv*sin(x(1))*[x(1) x(1)^2 u 1]; ...
qsci*[CMalpha CMq*c2v 0 0 0 0 x(1) x(2)*c2v u 1 zeros(1,6)]; ...
0 1 zeros(1,8) 1 0 0 0 0 0 ;zeros(np,ns+np)];
end

if coupling==1
A=[-qsmv*(-CN*sin(x(1))+cos(x(1))*CNalpha-CA*cos(x(1))-...
   (CAalpha+2*x(1)*CAalpha2)*sin(x(1)) ) 1 32.2/403.1*(-
cos(phi)*cos(am)*sin(x(3))+sin(am)*cos(x(3))) -qsmv*cos(x(1))*[x(1) u] -qsmv ...
0 0 0 0 0 -qsmv*cos(x(1)) qsmv*sin(x(1))*[x(1) x(1)^2 u 1]; ...
qsci*[CMalpha CMq*c2v 0 0 0 0 x(1) x(2)*c2v u 1 zeros(1,6)]; ...
0 1 zeros(1,8) 1 0 0 0 0 0 ;zeros(np,ns+np)];
end

end
```

```
if strcmp(case_study,'3')
u=get_u(t(k));
vm=u(8);
dele=u(9);
beta=u(10);
pm=u(11);
rm=u(12);
phi=u(13);
qbar=u(16);
am=u(2)-(0.0279/vm)*u(3);
S=184;m=196;g=32.2;cbar=5.58;
Iyy=3953.2;Ixx=6892.7;Izz=10416.4;

CLalpha=x(ns+1);
CLdelta=x(ns+2);
CLzero=x(ns+3);
CMalpha=x(ns+4);
CMq=x(ns+5);
CMalpha_dot=x(ns+6);
CMdelta=x(ns+7);
CMzero=x(ns+8);
thetazero_dot=x(ns+9);
CNzero=x(ns+10);
qsmv=qbar*S/(m*vm*cos(beta));
qsmg=qbar*S/(m*g);c2v=cbar/(2*vm);
CL=CLzero+CLalpha*x(1)+CLdelta*dele;
CN=CLalpha*x(1)+CLdelta*dele+CNzero;

qsci=qbar*S*cbar/Iyy;
adot=-qsmv*CL+x(2)-tan(beta)*(pm*cos(am)+rm*sin(am))+...
   g/(vm*cos(beta))*(cos(phi)*cos(x(3))*cos(am)+sin(x(3))*sin(am));
qdot=qsci*(CMzero+CMalpha*x(1)+c2v*(CMq*x(2)+...
   CMalpha_dot*adot)+CMdelta*dele)+((Izz-Ixx)/Iyy)*rm*pm;

f1change_a=-qsmv*CLalpha;
f1change_theta=g/(vm*cos(beta))*(-cos(phi)*sin(x(3))*cos(am)+cos(x(3))*sin(am));
f2change_a=qsci*(CMalpha+CMalpha_dot*c2v*f1change_a);
f2change_theta=qsci*c2v*CMalpha_dot*f1change_theta;

A=[ f1change_a 1 f1change_theta -qsmv*[x(1) dele 1] zeros(1,7) ; ...
f2change_a qsci*c2v*[(CMq+CMalpha_dot) f2change_theta -...
CMalpha_dot*qsmv*[x(1) dele 1] ] qsci*[x(1) c2v*[x(2) adot] dele 1] 0 0; ...
0 cos(phi) zeros(1,9) 1 0 ; zeros(np,ns+np)];
end
```

```matlab
if strcmp(case_study,'4')
data=get_u(t(k));u=data(9:10);
A=[x(ns+1) x(ns+2) 0.0069 -1 x(1) x(2) u(1) u(2) 1 zeros(1,16);...
x(ns+6) x(ns+7) 0 x(ns+8) 0 0 0 0 0 x(1) x(2) x(4)  u(1) u(2) 1 zeros(1,10);...
0 1 zeros(1,13) 1 zeros(1,9);x(ns+13) x(ns+14) 0 x(ns+15) zeros(1,12) ...
x(1) x(2) x(4) u(1) u(2) 1 0 0 0 ;zeros(np,ns+np)];
end

if strcmp(case_study,'5')

u=get_u(t(k));

delta_a=u(9);
delta_r=u(10);
theta_m=u(13);
qm=u(12);
am=u(14);

qbar=865.3;S=9.3;m=387.7;V=39.41;b=6.81;g=9.81;
Ixx=314;Iyy=488;Izz=698;Ixz=69;
Kz=0.305;Kx=2.73;

CYbeta=x(ns+1);
CYdelta_r=x(ns+2);
Beta_zero=x(ns+3);
CLbeta=x(ns+4);
CLp=x(ns+5);
CLr=x(ns+6);
CLdelta_a=x(ns+7);
CLdelta_r=x(ns+8);
CL_zero=x(ns+9);
Phi_zero=x(ns+10);
CNbeta=x(ns+11);
CNp=x(ns+12);
CNr=x(ns+13);
CNdelta_a=x(ns+14);
CNdelta_r=x(ns+15);
CN_zero=x(ns+16);
CY_zero=x(ns+17);
CYp=x(ns+18);
CYr=x(ns+19);
CYdelta_a=x(ns+20);

qsmv=qbar*S/(m*V);
b2v=b/(2*V);
```

```matlab
qsbix=qbar*S*b/Ixx;
qsbiz=qbar*S*b/Izz;
qsmg=qbar*S/(m*g);

beta_dot=qsmv*(CYbeta*x(1)+b2v*CYp*x(2)+b2v*CYr*x(4)+...
    CYdelta_a*delta_a+CYdelta_r*delta_r+Beta_zero)+ ...
x(2)*sin(am)-x(4)*cos(am)+(g/V)*sin(x(3))*cos(theta_m);
dot1=qsbix*(CLbeta*x(1)+b2v*CLp*x(2)+b2v*CLr*x(4)+...
    CLdelta_a*delta_a+CLdelta_r*delta_r+CL_zero)+ ...
((Iyy-Izz)/Ixx)*x(4)*qm+Ixz/Ixx*x(2)*qm;
phi_dot=x(2)+qm*tan(theta_m)*sin(x(3))+x(4)*tan(theta_m)*cos(x(3))+Phi_zero;
dot2=qsbiz*(CNbeta*x(1)+b2v*CNp*x(2)+b2v*CNr*x(4)+...
    CNdelta_a*delta_a+CNdelta_r*delta_r+CN_zero)+ ...
((Ixx-Iyy)/Izz)*x(2)*qm-Ixz/Izz*x(4)*qm;
prdot=inv([1 -Ixz/Ixx;-Ixz/Izz 1])*[dot1;dot2];

i1=Ixx*Izz/(Ixx*Izz-Ixz^2);i2=Ixz*Izz/(Ixx*Izz-Ixz^2);i3=Ixz*Ixx/(Ixx*Izz-Ixz^2);
d1b=qsbix*CLbeta;
d2b=qsbiz*CNbeta;
d1p=qsbix*CLp*b2v+(Ixz/Ixx)*qm;
d2p=qsbiz*CNp*b2v+((Ixx-Iyy)/Izz)*qm;
d1r=qsbix*CLr*b2v+((Iyy-Izz)/Ixx)*qm;
d2r=qsbiz*CNr*b2v-(Ixz/Izz)*qm;

A=[qsmv*CYbeta qsmv*b2v*CYp+sin(am) (g/V)*cos(x(3))*cos(theta_m) ...
    qsmv*b2v*CYr-cos(am)  qsmv*[x(1) delta_r 1] zeros(1,14) ...
qsmv*[b2v*x(2) b2v*x(4) delta_a]; ...
(i1*d1b+i2*d2b) (i1*d1p+i2*d2p) 0 (i1*d1r+i2*d2r) 0 0 0 (i1*qsbix*x(1)) (i1*qsbix*x(2)*b2v) ...
(i1*qsbix*x(4)*b2v) (i1*qsbix*delta_a) (i1*qsbix*delta_r) (i1*qsbix) 0 (i2*qsbiz*x(1)) ...
(i2*qsbiz*x(2)*b2v) (i2*qsbiz*x(4)*b2v) (i2*qsbiz*delta_a) (i2*qsbiz*delta_r) (i2*qsbiz) zeros(1,4); ...
0 1 qm*tan(theta_m)*cos(x(3))-x(4)*tan(theta_m)*sin(x(3)) tan(theta_m)*cos(x(3)) zeros(1,9) 1
zeros(1,10); ...
(i3*d1b+i1*d2b) (i3*d1p+i1*d2p) 0 (i3*d1r+i1*d2r) 0 0 0 (i3*qsbix*x(1)) (i3*qsbix*x(2)*b2v) ...
(i3*qsbix*x(4)*b2v) (i3*qsbix*delta_a) (i3*qsbix*delta_r) (i3*qsbix) 0 (i1*qsbiz*x(1)) ...
(i1*qsbiz*x(2)*b2v) (i1*qsbiz*x(4)*b2v) (i1*qsbiz*delta_a) (i1*qsbiz*delta_r) (i1*qsbiz) ...
zeros(1,4); zeros(np,ns+np)] ;
end


[~,next_states]=ode45(@dynamics,t(k):dt/2:t(k+1),x,[],P);
x_next=next_states(end,:)'; F=expm(A*dt);


%%%------------------Measurement linearisation function-------------------------------
```

```matlab
function [zp,Hk,lambda]=measurement_transition(x,k,P)

global  case_study ns t SOKF

lambda=0;

if strcmp(case_study,'1')
data=get_u(t(k));u=data(9);
u0=415.2;
h=@(x) [x(1);x(2);x(3);-u0/32.2*(x(ns+1)*x(1)+x(ns+4)*u+x(ns+6))+...
(-0.01/32.2)*(x(ns+2)*x(1)+x(ns+3)*x(2)+x(ns+5)*u+x(ns+7))+x(ns+9)];zp=h(x);
Hk=[1 zeros(1,11);0 1 zeros(1,10);0 0 1 zeros(1,9);...
   -(u0/32.2)*x(ns+1)-(0.01/32.2)*x(ns+2) -0.01/32.2*x(ns+3) 0 ...
-u0/32.2*x(1) -0.01/32.2*x(1) -0.01/32.2*x(2)  -u*u0/32.2  -0.01/32.2*u -u0/32.2 -0.01/32.2 0 1];
end


if strcmp(case_study,'2a')

data=get_u(t(k));

u=data(9);
am=data(2);
beta=data(10);
pm=data(11);
rm=data(12);
phi=data(13);
Ixx=4142.9;Iyy=3922.4;Izz=7642.5;

CNalpha=x(ns+1);
CNDelta=x(ns+2);
CLzero=x(ns+3);
CMalpha=x(ns+4);
CMq=x(ns+5);
CMDelta=x(ns+6);
CMzero=x(ns+7);
Thetazero_dot=x(ns+8);
CNzero=x(ns+9);
CAalpha=x(ns+10);
CAdelta=x(ns+11);
CAzero=x(ns+12);

CN=CNalpha*x(1)+CNDelta*u+CNzero;
CA=CAalpha*x(1)+CAdelta*u+CAzero;
```

```
CL=CN*cos(x(1))-CA*sin(x(1))+CLzero;

qsmv=83.08*184/(172.667*403.1);
qsci=83.08*184*5.58/3922.4;
c2v=5.58/(2*403.1);

qsmg=83.08*184/(172.667*32.2);
qdot=qsci*(CMalpha*x(1)+CMq*c2v*x(2)+CMDelta*u+CMzero);

h=@(x) [x(1)+0.0279*x(2)/403.1;x(2);x(3);qsmg*CN+...
    (0.101/32.2)*qdot;-qsmg*CA-(1.17/32.2)*qdot ];zp=h(x);

Hk=[1 0.0279/403.1 zeros(1,13);0 1 zeros(1,13);0 0 1 zeros(1,12);...
qsmg*CNalpha+0.101/32.2*qsci*CMalpha 0.101/32.2*qsci*c2v*CMq 0 qsmg*[x(1) u] ...
0 0.101/32.2*qsci*[x(1) x(2)*c2v u 1] 0 qsmg 0 0 0;...
-qsmg*CAalpha-1.17/32.2*qsci*CMalpha -1.17/32.2*qsci*[c2v*CMq 0 0 0 0 x(1) x(2)*c2v u 1] ...
0 0 -qsmg*[x(1) u 1] ];
end

if strcmp(case_study,'2b')

data=get_u(t(k));

u=data(9);
am=data(2);
beta=data(10);
pm=data(11);
rm=data(12);
phi=data(13);
Ixx=4142.9;Iyy=3922.4;Izz=7642.5;

CNalpha=x(ns+1);
CNDelta=x(ns+2);
CLzero=x(ns+3);
CMalpha=x(ns+4);
CMq=x(ns+5);
CMDelta=x(ns+6);
CMzero=x(ns+7);
Thetazero_dot=x(ns+8);
CNzero=x(ns+9);
CAalpha=x(ns+10);
CAalpha2=x(ns+11);
CAdelta=x(ns+12);
CAzero=x(ns+13);
```

```matlab
CN=CNalpha*x(1)+CNDelta*u+CNzero;
CA=CAalpha*x(1)+CAalpha2*x(1)^2+CAdelta*u+CAzero;
CL=CN*cos(x(1))-CA*sin(x(1))+CLzero;

qsmv=83.08*184/(172.667*403.1);
qsci=83.08*184*5.58/3922.4;
c2v=5.58/(2*403.1);

qsmg=83.08*184/(172.667*32.2);
qdot=qsci*(CMalpha*x(1)+CMq*c2v*x(2)+CMDelta*u+CMzero);

h=@(x) [x(1)+0.0279*x(2)/403.1;x(2);x(3);qsmg*CN+(0.101/32.2)*qdot;-qsmg*CA-(1.17/32.2)*qdot ];
zp=h(x);

Hk=[1 0.0279/403.1 zeros(1,14);0 1 zeros(1,14);0 0 1 zeros(1,13);...
qsmg*CNalpha+0.101/32.2*qsci*CMalpha 0.101/32.2*qsci*c2v*CMq ...
0 qsmg*[x(1) u] 0 0 0.101/32.2*qsci*[x(1) x(2)*c2v u 1] 0 qsmg 0 0 0;...
-qsmg*(CAalpha+2*x(1)*CAalpha2)-1.17/32.2*qsci*CMalpha ...
-1.17/32.2*qsci*[c2v*CMq 0 0 0 0 x(1) x(2)*c2v u 1] 0 0 -qsmg*[x(1) x(1)^2 u 1] ];

end

if strcmp(case_study,'3')

u=get_u(t(k));
vm=u(8);
dele=u(9);
beta=u(10);
pm=u(11);
rm=u(12);
phi=u(13);
qbar=u(16);
am=u(2)-(0.0279/vm)*u(3);

S=184;m=196;g=32.2;cbar=5.58;
Iyy=3953.2;Ixx=6892.7;Izz=10416.4;

CLalpha=x(ns+1);
CLdelta=x(ns+2);
CLzero=x(ns+3);
CMalpha=x(ns+4);
CMq=x(ns+5);
CMalpha_dot=x(ns+6);
CMdelta=x(ns+7);
CMzero=x(ns+8);
```

```matlab
thetazero_dot=x(ns+9);
CNzero=x(ns+10);

qsmv=qbar*S/(m*vm*cos(beta));
qsmg=qbar*S/(m*g);c2v=cbar/(2*vm);
CL=CLzero+CLalpha*x(1)+CLdelta*dele;
CN=CLalpha*x(1)+CLdelta*dele+CNzero;

qsci=qbar*S*cbar/Iyy;
adot=-qsmv*CL+x(2)-tan(beta)*(pm*cos(am)+rm*sin(am))+...
   g/(vm*cos(beta))*(cos(phi)*cos(x(3))*cos(am)+sin(x(3))*sin(am));
qdot=qsci*(CMzero+CMalpha*x(1)+c2v*(CMq*x(2)+...
   CMalpha_dot*adot)+CMdelta*dele)+((Izz-Ixx)/Iyy)*rm*pm;

f1change_a=-qsmv*CLalpha;
f2change_a=qsci*(CMalpha+CMalpha_dot*c2v*f1change_a);
f1change_theta=g/(vm*cos(beta))*(-cos(phi)*sin(x(3))*cos(am)+cos(x(3))*sin(am));
f2change_theta=qsci*c2v*CMalpha_dot*f1change_theta;

h=@(x) [x(1)+0.0279*x(2)/vm;x(2);x(3);qsmg*CN+(0.101/g)*qdot ];zp=h(x);

Hk=[1 0.0279/vm zeros(1,11);0 1 zeros(1,11);0 0 1 zeros(1,10); ...
qsmg*CLalpha+0.101/g*f2change_a 0.101/g*qsci*c2v*[(CMq+CMalpha_dot) ...
f2change_theta] ...
qsmg*x(1)-0.101/g*qsci*c2v*CMalpha_dot*qsmv*x(1) ...
qsmg*dele-0.101/g*qsci*c2v*CMalpha_dot*qsmv*dele ...
-0.101/g*qsci*c2v*CMalpha_dot*qsmv 0.101/g*qsci*[x(1) c2v*[x(2) adot] dele 1] 0 qsmg];
end

if strcmp(case_study,'4')
u0=4665;g=32.2; data=get_u(t(k));u=data(9:10);
h=@(x) [x(1);x(2);x(3);x(4);u0/g*(x(ns+1)*x(1)+x(ns+3)*u(1)+x(ns+4)*u(2)+x(ns+5))+x(ns+19);...
x(ns+6)*x(1)+x(ns+7)*x(2)+x(ns+8)*x(4)+x(ns+9)*u(1)+x(ns+10)*u(2)+x(ns+20);...
x(ns+13)*x(1)+x(ns+14)*x(2)+x(ns+15)*x(4)+x(ns+16)*u(1)+x(ns+17)*u(2)+x(ns+21)];zp=h(x);
Hk=[1 zeros(1,24);0 1 zeros(1,23);0 0 1 zeros(1,22);0 0 0 1 zeros(1,21);...
   u0/g*[x(ns+1) 0 0 0 x(1) 0 u(1) u(2) 1]  zeros(1,13) 1 0 0;...
x(ns+6) x(ns+7) 0 x(ns+8) 0 0 0 0 0 x(1) x(2) x(4) u(1) u(2) zeros(1,9)  1 0;
x(ns+13) x(ns+14) 0 x(ns+15) zeros(1,12) x(1) x(2) x(4) u(1) u(2) 0 0 0 1];
end

if strcmp(case_study,'5')

u=get_u(t(k));

delta_a=u(9);
```

```
delta_r=u(10);
theta_m=u(13);
qm=u(12);
am=u(14);
pdot=u(6);
rdot=u(7);

qbar=865.3;S=9.3;m=387.7;V=39.41;b=6.81;g=9.81;
Ixx=314;Iyy=488;Izz=698;Ixz=69;
Kz=0.305;Kx=2.73;

CYbeta=x(ns+1);
CYdelta_r=x(ns+2);
Beta_zero=x(ns+3);
CLbeta=x(ns+4);
CLp=x(ns+5);
CLr=x(ns+6);
CLdelta_a=x(ns+7);
CLdelta_r=x(ns+8);
CL_zero=x(ns+9);
Phi_zero=x(ns+10);
CNbeta=x(ns+11);
CNp=x(ns+12);
CNr=x(ns+13);
CNdelta_a=x(ns+14);
CNdelta_r=x(ns+15);
CN_zero=x(ns+16);
CY_zero=x(ns+17);
CYp=x(ns+18);
CYr=x(ns+19);
CYdelta_a=x(ns+20);

qsmv=qbar*S/(m*V);
b2v=b/(2*V);
qsbix=qbar*S*b/Ixx;
qsbiz=qbar*S*b/Izz;
qsmg=qbar*S/(m*g);

beta_dot=qsmv*(CYbeta*x(1)+b2v*CYp*x(2)+b2v*CYr*x(4)+...
    CYdelta_a*delta_a+CYdelta_r*delta_r+Beta_zero)+ ...
x(2)*sin(am)-x(4)*cos(am)+(g/V)*sin(x(3))*cos(theta_m);
dot1=qsbix*(CLbeta*x(1)+b2v*CLp*x(2)+b2v*CLr*x(4)+...
    CLdelta_a*delta_a+CLdelta_r*delta_r+CL_zero)+ ...
((Iyy-Izz)/Ixx)*x(4)*qm+Ixz/Ixx*x(2)*qm;
phi_dot=x(2)+qm*tan(theta_m)*sin(x(3))+x(4)*tan(theta_m)*cos(x(3))+Phi_zero;
```

```matlab
dot2=qsbiz*(CNbeta*x(1)+b2v*CNp*x(2)+b2v*CNr*x(4)+CNdelta_a*delta_a+...
    CNdelta_r*delta_r+CN_zero)+ ...
((Ixx-Iyy)/Izz)*x(2)*qm-Ixz/Izz*x(4)*qm;
prdot=inv([1 -Ixz/Ixx;-Ixz/Izz 1])*[dot1;dot2];

i1=Ixx*Izz/(Ixx*Izz-Ixz^2);i2=Ixz*Izz/(Ixx*Izz-Ixz^2);i3=Ixz*Ixx/(Ixx*Izz-Ixz^2);
d1b=qsbix*CLbeta;
d2b=qsbiz*CNbeta;
d1p=qsbix*CLp*b2v+(Ixz/Ixx)*qm;
d2p=qsbiz*CNp*b2v+((Ixx-Iyy)/Izz)*qm;
d1r=qsbix*CLr*b2v+((Iyy-Izz)/Ixx)*qm;
d2r=qsbiz*CNr*b2v-(Ixz/Izz)*qm;

h=@(x) [x(1)-Kz*x(2)/V+Kx*x(4)/V;x(2);x(3);x(4);qsmg*(CYbeta*x(1)+...
    CYdelta_a*delta_a+CYdelta_r*delta_r+CY_zero)+...
    (0.098/9.81)*prdot(1)+0.651/9.81*prdot(2) ];zp=h(x);

Hk=[1 -Kz/V 0 Kx/V zeros(1,20);0 1 zeros(1,22);0 0 1 zeros(1,21); 0 0 0 1 zeros(1,20); ...
qsmg*CYbeta+(0.098/9.81)*(i1*d1b+i2*d2b)+0.651/9.81*(i3*d1b+i1*d2b) ...
(0.098/9.81)*(i1*d1p+i2*d2p)+0.651/9.81*(i3*d1p+i1*d2p) 0 ...
(0.098/9.81)*(i1*d1r+i2*d2r)+0.651/9.81*(i3*d1r+i1*d2r) ...
qsmg*[x(1) delta_r] 0  (0.098/9.81)*i1*qsbix*x(1)+0.651/9.81*i3*qsbix*x(1) ...
(0.098/9.81)*i1*qsbix*x(2)*b2v+0.651/9.81*i3*qsbix*x(2)*b2v ...
(0.098/9.81)*i1*qsbix*x(4)*b2v+0.651/9.81*i3*qsbix*x(4)*b2v ...
(0.098/9.81)*i1*qsbix*delta_a+0.651/9.81*i3*qsbix*delta_a ...
(0.098/9.81)*i1*qsbix*delta_r+0.651/9.81*i3*qsbix*delta_r ...
(0.098/9.81)*i1*qsbix+0.651/9.81*i3*qsbix 0 ...
(0.098/9.81)*i2*qsbiz*x(1)+0.651/9.81*i1*qsbiz*x(1) ...
(0.098/9.81)*i2*qsbiz*x(2)*b2v+0.651/9.81*i1*qsbiz*x(2)*b2v ...
(0.098/9.81)*i2*qsbiz*x(4)*b2v+0.651/9.81*i1*qsbiz*x(4)*b2v ...
(0.098/9.81)*i2*qsbiz*delta_a+0.651/9.81*i1*qsbiz*delta_a ...
(0.098/9.81)*i2*qsbiz*delta_r+0.651/9.81*i1*qsbiz*delta_r ...
(0.098/9.81)*i2*qsbiz+0.651/9.81*i1*qsbiz qsmg 0 0 qsmg*delta_a];


end

if norm(P)>0 && SOKF==1
[h2term,lambda]=get_h2term(P);
zp=zp+h2term;
end

%%%--------------------Get second order terms--------------------------

function [h2term,l]=get_h2term(P)
```

```matlab
h2term=0;l=0;
function f2term=get_f2term(x,P)
f2term=0;
```

```matlab
%%------------------Start of smooth function----------------------------

function [xk_smooth,P_smooth,pk_smooth,x0s,P0s,P_lag] =...
   smooth(xk_plus,pk_plus,xk_minus,pk_minus,Khi,H,K,S,invS,innov,x0,P0)

global N total_sts

xk_smooth(:,N)=xk_plus(:,N);P_smooth(N,:)=diag(pk_plus(:,:,N))';
pk_smooth(:,:,N)=pk_plus(:,:,N);


%%-----------------RTS Smoothing--------------------------

for k = N-1:-1:1 %Smoothing loop Starts

%%Smoothing Equations
K_rts(:,:,k+1)=(pk_plus(:,:,k)*Khi(:,:,k+1)'/pk_minus(:,:,k+1));
pk_smooth(:,:,k)=pk_plus(:,:,k)+(K_rts(:,:,k+1)*(pk_smooth(:,:,k+1)-...
   pk_minus(:,:,k+1))*K_rts(:,:,k+1)');
xk_smooth(:,k)=xk_plus(:,k)+K_rts(:,:,k+1)*(xk_smooth(:,k+1)-xk_minus(:,k+1));

P_smooth(k,:)=(diag(pk_smooth(:,:,k)))';%save smoothed covariance

end %Smoothing loop Ends

K_rts(:,:,1)=(P0*Khi(:,:,1)'/pk_minus(:,:,1));
P0s=P0-(K_rts(:,:,1)*(pk_minus(:,:,1)-pk_smooth(:,:,1))*K_rts(:,:,1)');
x0s=x0+K_rts(:,:,1)*(xk_smooth(:,1)-xk_minus(:,1));


for k=N:-1:1
if k==1,p_prev=P0;else p_prev=pk_plus(:,:,k-1); end
P_fow(:,:,k)=(eye(total_sts)-K(:,:,k)*H(:,:,k))*Khi(:,:,k)*p_prev;
P_lag(:,:,k)=(eye(total_sts)+...
   (pk_smooth(:,:,k)-pk_plus(:,:,k))*inv(pk_plus(:,:,k)))*P_fow(:,:,k);
end

%%%------------------------end----------------------------------
```

References

[1] Fisher, R. A., " On The Mathematical Foundations Of Theoretical Statistics, " *Phil. Trans. Roy. Soc. London Ser. A 222 401415, 1922.*

[2] Burgess, Ebenezer., " Translation of the Surya Siddhanta, " *Reprint from the Edition of 1860, University of Calcutta, 1935.*

[3] Plackett, R. L., " Some theorems in least squares, " *Biometrika 37, 149157, 1950.*

[4] N. Wiener, " The Extrapolation, Interpolation and Smoothing of Stationary Time Series: with Engineering Applications, " *Cambridge, Massachusetts: MIT Press; New York: Wiley and Sons; London: Chapman & Hall, 1949.*

[5] P. Swerling, " A Proposed Stagewise Differential Correction Procedure for Satellite Tracking and Prediction, " *Tech. Rep. P-1292, Rand Corporation, 1958.*

[6] R. L. Stratonovich, " Application of the Theory of Markov Process in Optimal Signal Discrimination, " *Radio Engineering and Electronic Physics, Vol. 1 : pp. 1-19, 1960.*

[7] R. E. Kalman, " A New Approach to linear Filtering and Prediction Problems, " *Transactions of the ASME-Journal of Basic Engineering, 82 (Series D) : pp. 35-45, 1960.*

[8] R. E. Kalman, R. S.Bucy " New Results in Linear Filtering and Prediction Theory, " *Journal of Basic Engineering, 1961.*

[9] G.L. Smith, S.F. Schmidt and L.A. McGee, " Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle, " *National Aeronautics and Space Administration, 1962.*

[10] L. A. Zadeh., " From Circuit Theory to Systems Theory, " *Proc. IRE., Vol. 50 : pp. 856 865, 1962.*

[11] Gauss, K. F. " Teoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium, " (1809). Reprinted translation : " Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Sections, " *Dover, New York, 1963.*

[12] H. E. Rauch, F. Tung and C. T. Striebel, " Maximum Likelihood Estimates of Linear Dynamic Systems, " *AIAA Journal, Vol.. 3, No.-8, 1965.*

[13] C. Hilborn and D. Lainiotis, " Optimal estimation in the presence of unknown parameters, " *IEEE Trans. Systems, Science, and Cybernetics, 5(1):3843, 1969.*

[14] Kailath. T., " An Innovation Approach To Detection and Estimation Theory, " *Proceedings of the lEEE, Vol. 58, 1970.*

[15] R. K. Mehra, " On the identification of variances and adaptive Kalman filtering, " *IEEE Trans. Automat. Control, Vol. 15, N.2 : pp. 175-184, 1970.*

[16] R. Kashyap, " Maximum likelihood identification of stochastic linear systems, " *IEEE Trans. Auto. Cont., 15(1):25-34, 1970.*

[17] Sorenson. H. W., " Least Squares estimation from Gauss to Kalman, " *IEEE Spectrum, Vol. 7, pp63-68, 1970.*

[18] A. H. Jazwinski, " Stochastic Process and Filtering Theory, " *Academic Press, New York, 1970.*

[19] R. Billard, " Aryabhata and Indian Astronomy, " *Indian Journal of History of Science, Vol. 12, No. 2 : pp. 207-224, " 1977.* Also R. Billard, " L'astronomie Indienne, " *Publications De Ecole Francaise D'Extreme-Orient, Paris, 1971.*

[20] R. Mehra., " Approaches to adaptive filtering, " *IEEE Trans. Auto. Cont., 17:903908, 1972.*

[21] B. Carew and P. Belanger, " Identification of optimum filter steady-state gain for systems with unknown noise covariances, " *IEEE Trans. Auto. Cont., 18(6): 582587, 1973.*

[22] Roskam, J " Methods for estimating stability and control derivarives of conventional subsonic airplanes, " *University of Kansas, USA, 1973.*

[23] D. Alspach. " A parallel filtering algorithm for linear systems with unknown time varying noise statistics, " *IEEE Trans. Auto. Cont., 19(5): 552556, 1974.*

[24] C. Neethling and P. Young. Comments on " Identification of optimum filter steady state gain for systems with unknown noise covariances, " *IEEE Trans. Auto. Cont., 19(5):623625, 1974.*

[25] Sorenson, H. W. " On the development of practical non linear filters in Estimation Theory, " *Edited by D. G. Lainiotis, American Elsevier Publishing Co Inc, New York, 1974.*

[26] P. R. Belanger, " Estimation of noise covariances for a linear time-varying stochastic process, " *Automatica 10, 1974.*

[27] A. Gelb, Editor, " Applied Optimal Estimation, " *Cambridge, Massachusetts: MIT Press, 1974.*

[28] P. Eykhoff., " System Identification: Parameter and State Estimation, " *Wiley, London, 1974.*

[29] Narasimha, R., " Performance Reliability of High Maintenance Systems, " *Journal of Franklin Institute, Vol. 303 : pp. 15-29, 1975.*

[30] Mary F. Shafer, " Stability and Control Derivatives of the T-37B Airplane, " *NASA TM X-56036, 1975.*

[31] Maine R. E. and Iliff K. W, "A Fortran program for determining aircraft stability and control derivatives from flight data, " *NASA TN-D7831, 1975.*

[32] T. Bohlin, " Four cases of identification of changing systems, " *System Identification: Advances and Case Studies, Academic Press, 1st edition, 1976.*

[33] Schultz, G . " Maximum Likelihood Identification Using Kalman Filtering Least Square Estimation, " *ESA TT-258, 1976.*

[34] K. A. Myers and B. D. Tapley, " Adaptive Sequential Estimation with Unknown Noise Statistics, " *IEEE Transactions on Automatic Control, Vol. AC 21 : pp. 520-525, 1976.*

[35] K. V. Sarma," Tradition of Aryabhatiya in Kerala: Revision of Planetary Parameters, " *Indian Journal of History of Science, Vol. 12, No. 2 : pp. 194-199, 1977.*

[36] Bierman, G.J., " Factorization Methods for Discrete Sequential Estimation, " *Dover Publications : pp. 223-224, 1977.*

[37] Beck, J. V. and Arnold, K. J.," Parameter Estimation in Engineering Science, " *John Wiley and Sons, New York, 1977.*

[38] Goodwin, G. C and Payne R. L," Dynamic System Identification : Experiment Design and Data Analysis, " *Academic press, New York, 1977.*

[39] C. F. Van Loan, "Computing Integrals Involving the Matrix Exponential", *IEEE Trans. Automatic Control, AC-23 (3): 395-404, 1978.*

[40] Sprott, D. A., " Gauss's Contributions to Statistics, " *Historia Mathematica 5 : pp. 183-203, 1978.*

[41] P. S. Maybeck, " Stochastic Models, Estimation, and Control: Volume 1, " *New York: Academic Press, 1979.*

[42] Klein, V., " Identification Evaluation Methods, Parameter Identification, " *AGARD-LS- 104 : pp. 2-21, 1979.*

[43] Ljung, L., " Asymptotic behaviour of the EKF as a parameter estimator for linear systems, " *IEEE trans. Automatic control, Vol. AC 24, pp. 36-50, 1979.*

[44] K. J. Astrom, " Maximum Likelihood and Prediction Error Methods, " *Automatica. Vol. 16 : pp. 551 574 Pergamon Press Ltd., 1980.*

[45] Sorenson. H. W., " Parameter Estimation Principles And Problems, "*Mercel Defier Inc. New York, 1980.*

[46] Maine R. E. and Iliff K. W., "Programmer's manual for MMLE3, a general Fortran program for Maximum Likelihood parameter estimation", *NASA TP-1690, 1981.*

[47] Eykhoff, P., " Trends And Progress In System Identification, " *Pergarnon Press New York. 1981.*

[48] Robert H. Shumway and David S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm, " *J. Time Series Anal., 3 : pp. 253-264, 1982.*

[49] Candy, J.V., " Signal Processing The Model based approach, "*McGraw Hill Book Company, 1986.*

[50] C. R. Rao, " Statistics and Truth: Putting Chance to Work - Ramanujan Memorial Lectures, " *Council of Scientific and Industrial Research, New Delhi, India, 1987.*

[51] M. Minsky, " The Society of the Mind, " *Picador Edition, 1988.*

[52] H. Visser and J. Molenaar., " Kalman Filter Analysis in Dendroclimatology, " *Biometrics, 44., pp 929-940, 1988.*

[53] Henk A. P. Blom, Bar-Shalom, " The Interacting Multiple Model algorithm for systems with Markovian Switching Coefficients, " *IEEE Transactions on Automatic Control, Vol. 33, No. 8, 1988.*

[54] Takashi Isobe and Eric D. Feigelson, Michael G. Akritas and Gutti Jogesh Babu, " Linear Regression in Astronomy. I., " *The Astrophysical Journal, 364 : 104-113, 1990.*

[55] Gemson, R. M. 0., " Estimation Of Aircraft Aerodynamic Derivatives Accounting For Measurement And Process Noise By EKF Through Adaptive Filter Tuning, " *PhD Thesis, Dept. of Aerospace Engineering IISc, Bangalore, 1991.*

[56] Eric D. Feigelson and Gutti Jogesh Babu, " Linear Regression in Astronomy. II., " *The Astrophysical Journal, 397 : 55-67, 1992.*

[57] Y. Bar-Shalom and X. Li, " Estimation and Tracking : Principles, Techniques and Software, " *Boston: Artech House, 1993.*

[58] N. J. Gordon, D. J. Salmond, A. F. M. Smith., " Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation, " *IEEE Proceedings, Vol. 140, No. 2 : pp. 107-113, 1993.*

[59] G. Evensen, " Sequential Data Assimilation with a Nonlinear Quasigeostrophic Model Using Monte Carlo Methods to Forecast Error Statistics, " *Journal of Geophysical Research, Vol. 99 (C5) : pp. 143-162, 1994.*

[60] Dag Ljungquist and Jens G. Balchen, " Recursive prediction error methods for online estimation in nonlinear state space models, " *Modeling, Identification and Control, Vol. 15 No. 2 : pp. 109-121, 1994.*

[61] Simon Julier, Jeffrey Uhlmann, and Hugh Durrant-Whyte, " A New Approach for Filtering Nonlinear Systems, " *Proceedings of the 1995 American Control Conference, IEEE Press, 1995.*

[62] Alfred Kleusberg and Peter J.G. Teunissen (Eds), " GPS for Geodesy, " *Springer Verlag 1996.*

[63] C. Wells, " The Kalman Filter in Finance, " *Springer-Science+Business Media, BV., 1996.*

[64] Ishimoto, S., " New Algorithm Of Maximum Likelihood Parameter Estimation For Flight Vehicles, " *AIAA -97- 3784, 1997.*

[65] Simon Julier, Jeffrey Uhlmann, " A New Approach for Filtering Nonlinear Systems, " *Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI, 182, 1997.*

[66] W. T. Federer and B. R. Murthy., " Kalman Filter Bibliography : Agriculture, Biology, and Medicine, " *Technical Report BU-1436-M Department of Biometrics, Cornell University, Ithaca, 1998.*

[67] E. Mazor, A. Averbuch, Y. Bar-Shalom, J. Dayan, " Interacting Multiple Model methods in Target Tracking: A Survey, " *IEEE Transactions on Aerospace and Electronic Systems, Vol. 34 : pp. 103 - 123, 1998.*

[68] P. Tichavsky, C. H. Muravchik and A. Nehorai, " Posterior Cramer Rao bounds for discrete time nonlinear filtering, " *Signal Processing, IEEE Transactions, Vol. 46, No. 5, 1998.*

[69] R. M. O. Gemson and R. Ananthasayanam, " Importance of Initial State Covariance Matrix for the Parameter Estimation Using Adaptive Extended Kalman Filter, " *AIAA-98-4153, 1998.*

[70] A. H. Mohamed and K. P. Schwarz, " Adaptive Kalman Filtering for INS/GPS, " *Journal of Geodesy, 1999.*

[71] Anilkumar, A. K., " Application of Controlled Random Search Optimisation Technique in MMLE with Process Noise, " *MSc Thesis, Dept. of Aerospace Engineering IISc, Bangalore, 2000.*

[72] Valappil J and Georgakis C, " Systematic Estimation of State Noise Statistics for Extended Kalman Filters, " *AIChe Journal Vol. 46, No. 2, 2000.*

[73] Y. Oshman and I. Shaviv., " Optimal Tuning of a Kalman Filter Using Genetic Algorithm, " *AIAA Paper 2000-4558, 2000.*

[74] R.H. Shumway, D.S. Stoffer, " Time Series Analysis and its Applications, " *Springer, Verlag, New York Inc., 2000.*

[75] R. Fruhwirth, M. Regier, R. K. Bock, H. Grote and D. Notz, " Data Analysis Techniques for High-Energy Physics, " *Cambridge Monographs on Particle Physics, Nuclear physics and Cosmology, 2000.*

[76] Oussalah M., De Schutter J., " Adaptive Kalman Filter for Noise Identification, " *International Conference on Noise and Vibration Engineering, ISMA25 : pp. 1225-1232, 2000.*

[77] Christian Bohn, " Recursive Parameter Estimation for Non Linear Continuous time systems through Sensitivity-Model based Adaptive Filters, " *PhD Thesis, Dept. of Electrical Engineering and Information Science, 2000.*

[78] Georgios, B. G., Erchin, S., " A bibliography on nonlinear system identification, " *Signal Processing 81, 533-580, 2001.*

[79] Yaakov Bar-Shalom, X. Rong Li and Thiagalingam Kirubarajan., " Estimation with Applications To Tracking and Navigation, Theory, Algorithm and Software, " *John Wiley and Sons. Inc, 2001.*

[80] M. R. Ananthasayanam, H. S. Suresh and M. R. Muralidharan, " GUI based software for teaching parameter estimation technique using MMLE, " *Report 2001 FM 1,Project-JATP/MRA/053,IISc-Bangalore, 2001.*

[81] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson and P. J. Nordlund, " Particle Filters for Positioning, Navigation, and Tracking, " *IEEE Transactions on Signal Processing, Vol. 50, No. 2 : pp. 425-437, 2002.*

[82] T. D. Powell, " Automayed tuning of an Extended Kalman Filter Using the Downhill Simplex Algorithm, " *J. Guidance, Control,and Dynamics, Vol. 25, 2002.*

[83] N. K. Philip and M. R. Ananthasayanam, " Relative Position and Attitude Estimation and Control Schemes for the Final Phase of an Autonomous Docking Mission, " *Acta Astronautica, Vol. 52, No. 7 : pp. 511-522, 2003.*

[84] G. Evensen, " The Ensemble Kalman Filter: Theoretical Formulation and Practical Implementation, " *Ocean Dynamics, Vol. 53 : pp. 343-367, 2003.*

[85] Z. Chen " Bayesian Filtering: From Kalman filters to particle filters, and beyond, " *Available online : http : //soma.crl.mcmaster.ca/zhechen/homepage.htm, 2003.*

[86] Simon J. Julier and Jeffrey K. Uhlmann, " Unscented Filtering and Nonlinear Estimation, " *Proceedings of the IEEE, Vol. 92, No. 3 : pp. 401-422, March 2004.*

[87] Helen Basil, M. R. Ananthasayanam, and S. N. Puri, " An Application of Constant Kalman Gain Approach for the Problem of Integrating the MEMS-INS and the GPS, " *KSAS-JSASS Joint Symposium on Aerospace Engineering, 2004.*

[88] B. Ristic, S. Arulampalam and N. Gordon, " Beyond the Kalman Filter: Particle Filters for Tracking Applications, *Artech House Radar Library, 2004.*

[89] Fred Daum, " Nonlinear Filters: Beyond the Kalman Filter, " *IEEE A&E systems magazine Vol. 20, No. 8, Part 2: Tutorials-Daum, 2005.*

[90] R. Anandraj, M. R. Ananthasayanam, P. R. Mahapatra and P. Soma, " Constant Gain Kalman Filter Approach Using Genetic Algorithm for Estimating the Ionospheric Total Electron Content Over Low Latitude Indian and Other Stations During Magnetic Storms, " *28th General Assembly, URSI, 2005.*

[91] M. R. Ananthasayanam, A. K. Anilkumar and P. V. Subba Rao, " New Approach for the Evolution and Expansion of Space Debris Scenario, " *Journal of Spacecraft and Rockets, Vol. 43, No. 6 : pp. 1271-1282, 2006.*

[92] Dan Simon, "Optimal State Estimation, Kalman, H_∞, and Nonlinear Approaches," *John Wiley & Sons : Appendix C, pp. 493-499, 2006.*

[93] Jonathan Tennenbaum and Bruce Director, " How Gauss Determined The Orbit of Ceres, " *Reprinted from the Summer 1998 issue of Fidelio Magazine, 2006.*

[94] B. J. Odelson, A. Lutz, J. B. Rawlings, " The autocovariance-least squares method for estimating covariances: application to model-based control of chemical reactors, " *IEEE Trans. Control Syst. Technol. 14, 2006.*

[95] Ravindra V. Jategaonkar., " Flight Vehicle System Identification : A Time Domain Methodology, " *Published by AIAA, Volume 216., 2006.*

[96] V. Klein and E.A. Morelli., " Aircraft System Identification. Theory and Practice, " *AIAA Education Series, 2006.*

391

[97] M. Costagli, and E. E. Kuruoglu., " Image separation using particle filters, " *Digital Signal Processing 17 : pp. 935946, 2007.*

[98] A. K. Anilkumar, M. R. Ananthasayanam and P. V. Subba Rao, " A Constant Gain Kalman filter Approach for the Prediction of Re-entry of Risk Objects, " *Acta Astronautica, Vol. 61, No. 10 : pp. 831-839, 2007.*

[99] Grewal, M. S., Lawrence, R.W., Andrews, A. P., "Global Positioning Systems, Inertial Navigation, and Integration, Second Edition, " *John Wiley & Sons, Inc., Publication, 2007.*

[100] Grewal, M. S. and Andrews, A. P., " Kalman Filtering : Theory and Practice Using MATLAB, 3rd Edition, " *John Wiley & Sons, Inc., Publication, 2008.*

[101] M. R. Ananthasayanam and A. K. Sarkar, " The Philosophy of Kalman Filter in its Applications to Physical and Other Sciences, " *International Conference for Avionics, ICAS, Hyderabad, 2008.*

[102] G. Evensen, " Data Assimilation: The Ensemble Kalman Filter, Second Edition, " *Springer Verlag, 2009.*

[103] Jitendra R. Raol, " Multi-Sensor Data Fusion with MATLAB$^{®}$, " *CRC Press, January 2010*

[104] Vinay A Bavdekar, Anjali P Deshpande, Sachin C Patwardhan, " Identification of process and measurement noise covariance for state and parameter estimation using extended Kalman filter, " *Journal of Process control, 21 : 585-601, 2011.*

[105] M. R. Ananthasayanam and K. M. Bharadwaj, " The Philosophy, Principles, and Practice of Kalman Filter since Ancient Times to the Present, " *45th IAA History of Astronautics Symposium, 2011.*

[106] Tak Kit Lau and Kai-wun Lin, " Evolutionary tuning of sigma-point Kalman filters, " *IEEE International Conference on Robotics and Automation (ICRA), 2011 .*

[107] K. Ramasubramanian and M. S. Sriram, " Tantrasangraha of Nilakantha Somayaji, " *Sources and Studies in the History of Mathematics and Physical Sciences, Springer, 2011 .*

[108] R. Brown and P. Hwang, " Introduction to Random Signals and Applied Kalman Filtering, With MATLAB Exercises, 4th Edition" *John Wiley & Sons, Inc., 2012.*

[109] Shyam, M. M, " An Iterative Tuning Strategy For Achieving Cramer Rao Bound Using Extended Kalman Filter For A Parameter Estimation Problem, " *MTech Thesis, Dept. of Electrical Engineering, IIT Kanpur, 2014*.

[110] Manika S, Bhaswati G and Ratna G, " Two novel metrics for determining the tuning parameters of the Kalman Filter, " *IEEE Transactions on Instrumentation and Measurement, Vol. 63, No. 4, 2014*.

[111] Megan A. Zagrobelny and James B. Rawlings, " Identification of Disturbance Covariances Using Maximum Likelihood Estimation, " *TR No. 2014-02, Department of Chemical and Biological Engineering, University of Wisconsin-Madison, 2014*.