<http://www.diva-portal.org>

Preprint

This is the submitted version of a paper presented at *15th IEEE International Conference on Environment and Electrical Engineering (EEEIC)*.

Citation for the original published paper:

Mahmood, F., Hooshyar, H., Vanfretti, L. (2015)

A method for extracting steady state components from synchrophasor data using Kalman filters.

In: *2015 IEEE 15th International Conference on Environment and Electrical Engineering, EEEIC 2015 - Conference Proceedings* (pp. 1498-1503). IEEE

<http://dx.doi.org/10.1109/EEEIC.2015.7165393>

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-175857>

application specific, i.e. in order to feed the signal shown in Fig. 1 to steady state and dynamic applications, the processing should be different in both cases. The focus of this paper is steady state applications, so the raw PMU data is processed in an appropriate way in order to extract the steady state component from the signal.

Fig. 1. Signal features at different time scales in a PMU signal

B. Problems in PMU Measurements

Fig. 2 shows a voltage magnitude as generated by a PMU using a Hardware-In-the-Loop (HIL) lab setup. Discrete events, such as, tap changer operation results in outliers in PMU data, as can be seen on the left hand side of Fig. 2. These outliers can affect the performance of target applications. The other problem shown in Fig. 2, is the missing data. Missing data is not acceptable for many applications as calculations may be affected. Also shown in Fig. 2 is the problem of measurement noise. Feeding noisy data to steady state applications can lead to wrong results.

Therefore, all of these practical problems need to be addressed by using appropriate data processing methods. A data processing method using KF is adopted in this paper to overcome these issues, as discussed next.

Fig. 2. Problems in PMU Voltage Measurement

III. IMPLEMENTATION METHOD

This section presents first the theoretical background of traditional KF. For details the readers are referred to [10] [11]. Second, our contribution on the application of the KF method is described. In the last subsection, the experimental setup to generate PMU data used in section IV is explained.

A. The Kalman Filter (KF)

A linear discrete time controlled process is assumed to have linear stochastic process equations and measurement equations

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_k + w_{k-1} \\ z_k &= Hx_k + v_k \end{aligned} \quad (1)$$

where x is the state vector, z is the measurement vector, A is the $n \times n$ matrix that relates the state at previous time step $k-1$ to the state at current step k , which is assumed to be constant in each iteration, B is the control input which relates input u to the state x and H is the $m \times n$ matrix which relates state x_k to the measurements z_k . The process noise w_k and measurement noise v_k are assumed to be two mutually independent random variables with normal probability distributions

$$\begin{aligned} p(w) &\sim \mathcal{N}(0, Q) \\ p(v) &\sim \mathcal{N}(0, R) \end{aligned} \quad (2)$$

where Q is process noise covariance matrix and R is measurement noise covariance matrix, these two matrices are usually constant but can be updated at each time step.

The a priori state estimate $\hat{x}_k^-$ is defined as the estimate at step k , given knowledge of the process prior to step k . An a posteriori state estimate $\hat{x}_k$ is the estimate at time step k given the measurement z_k . In this respect the priori estimate error and a priori estimate covariance are given respectively as $e_k^- \equiv x_k - \hat{x}_k^-$ and $P_k^- \equiv E[e_k^- e_k^{-T}]$. Similarly, the a posteriori estimate error and a posteriori estimate covariance are given respectively as $e_k \equiv x_k - \hat{x}_k$ and $P_k \equiv E[e_k e_k^T]$.

The KF estimates the state by using feedback control: the filter estimates the process state at some time and then obtains feedback in the form of noisy measurements. KF can be divided into two parts:

Time update equations (Prediction): The time update equations are responsible of projecting forward (in time) the previous state $\hat{x}_{k-1}$ and error covariance estimate P_{k-1} to obtain the a priori estimates for the next step k . The time update equations are considered to be the predictor equations.

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \quad (3)$$

Measurement update equations (Correction): Measurement update equations are responsible for feedback by incorporating a new measurement z_k into the a priori estimate to obtain an improved a posteriori estimate. The measurement update equations can be thought of as corrector equations.

$$\begin{aligned} K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \\ P_k &= (I - K_k H)P_k^- \end{aligned} \quad (4)$$

where K is an $n \times m$ matrix known as Kalman gain matrix, z_k is the actual measurement at step k , $H\hat{x}_k^-$ is the predicted measurement, $\hat{x}_k$ is the a posteriori estimate which is a linear combination of an a priori estimate $\hat{x}_k^-$ and a weighted

difference between an actual measurement and predicted measurement. From the above expressions it can be concluded that as R approaches zero, an actual measurement z_k is trusted more and predicted measurement $H\hat{x}_k^-$ is trusted less. On the other hand, as the a priori estimate error covariance P_k^- approaches zero, the predicted measurement is trusted more than the actual measurement.

To summarize, the KF is a predictor-corrector algorithm. After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project a new priori estimate. In addition, the KF does not require all the previous data at each estimate, instead, it just recursively conditions the current estimate on all the past measurements. This makes the KF a suitable method for real time applications.

B. Kalman Filter Application and Implementation

KF is implemented in this paper to extract the steady state components from PMU signals. This is carried out by assuming that dynamic components in the PMU data are measurement noise. According to [8], accuracy of state estimators using KF are largely influenced by measurement and process noise covariance matrices, i.e. R and Q . Therefore, KF is applied to reduce the noise and increase the accuracy of PMU data by updating the value of R in each KF iteration. The KF algorithm was implemented in MATLAB. PMUs have 4 state variables to be estimated, i.e. V , θ , I , δ , which corresponds to the variables that a PMU measures [6]. Assumed process model given in (1), A is an identity matrix because the time step between KF iterations are small enough to assume that the current state is equal to the previous state¹, B is zero as there is no input to the process and H is also an identity matrix because the states are directly measured.

The overview of the KF implementation is shown in Fig. 3. The initial estimate of $\hat{x}_{k-1}$ and $\hat{P}_{k-1}$ are the input to the time update (prediction step). The predictor step is the same as given in equation (3), projecting forward (in time) the previous state and error covariance estimate to produce the state and error covariance for the next step. The corrector step is, however, not the same as used in conventional KF. Instead, in addition to equation (4), a Residue is calculated as the difference between state estimation $\hat{x}_k$ and actual measurement z_k

$$Residue = \hat{x}_k - z_k \quad (5)$$

The variance of the calculated residue in equation (5) is computed in each step in equation (6) using rolling windows. This gives the measurement covariance matrix R .

$$\begin{aligned} R_{new} &= Var(Residue) \\ R_{new} &= E[(Residue - E[Residue])^2] \end{aligned} \quad (6)$$

where E represents the mathematical expectation of the value. R is therefore updated in each time step, and is used in the start of next correction step.

$$R = R_{new} \quad (7)$$

¹This assumption is valid for PMU reporting rates of above 30 samples per seconds.

Fig. 3. Kalman Filter Overview

It was observed that instead of using a whole parcel of data at a time, using rolling windows in time gives better results. For example, small signal oscillations and noise in the PMU data can be effectively filtered out by updating the value of R in rolling windows. During the period when oscillations are large, a large difference between actual measurement z_k and state estimation $\hat{x}_k^-$ occurred, causing the residue to be higher. As R is calculated on the basis of the residue, R becomes larger as well. In this case the actual measurement z_k is trusted less, while a priori state estimation $\hat{x}_k^-$ is trusted more when calculating the new estimate $\hat{x}_k$. Therefore oscillations can be easily filtered out and a smoother response is obtained for steady state applications. The size of rolling window is also an important factor which will be discussed in section IV-C

C. Experimental setup for generating PMU data

The PMU data is generated using a Hardware-In-the-Loop (HIL) lab setup as shown in Fig. 4. A SimPowerSystems (SPS) model is simulated in real time (hardware synchronized mode) using an OPAL-RT simulator [12]. The voltage and current are available at the analog output channel of the simulator. These quantities are acquired by a National Instrument Compact Rio (NI cRIO) 9074 PMU[13], with software configured to perform PMU functions [14] through voltage and current amplifiers by Megger [15]. The network connection of the PMU streams the data to a Phasor Data Concentrator (PDC) server. The raw PMU data was acquired from the PDC and archived in Comma Separated Values (CSV) format to be fed to the KF execution environment.

IV. CASE STUDIES AND RESULTS

A. Difference between simulated data and PMU data

WAMS application development cannot rely on simulated data. There are many practical issues which can be observed from real PMUs which are not present in simulations. For example, Fig. 5 shows a very clear illustration for this argument. A power system is disturbed with random load variations (RLV) $t = 45sec$ and deactivated at $t = 115sec$. The voltage response from the simulation and the HIL PMU setup is shown in Fig. 5. It can be clearly seen that the voltage from the HIL

Fig. 4. Hardware in the loop (HIL) Lab Setup

PMU setup has missing data, which is due to the fact that the PMU data is delayed for a longer time than the PDC maximum waiting time. Also with the discrete event of connection and disconnection of the RLV, PMU generates outliers. Hence, in the development of the method in section III-B and the example below, only data from the RT HIL setup is used.

B. Kalman Filter Performance

The developed KF method is applied on HIL PMU data. The disturbance here is the same as in Section IV-A. The results for the PMU voltage magnitude and PMU voltage angle are shown. The top of Fig. 6 shows the KF algorithm performance for the PMU voltage magnitude signal. It can be seen that missing data is compensated for (can be seen in zoomed in view). As soon as the KF detects a missing data point, it holds the value of the previous state. When the disturbance is applied at $t = 45\text{sec}$ and removed at $t = 115\text{sec}$, outliers are generated in the PMU voltage signal. The RLV excites the small signal oscillations, which should be neglected in a steady state application. Hence by updating the R as explained in Section III-B, the dynamics and outliers are filtered, giving a much smoother response for use in steady state applications. The resulting response of R is shown in the bottom of Fig. 6.

Fig. 6. Kalman Filter output for the PMU's voltage magnitude signal

Fig. 7. Kalman Filter output for the PMU's voltage angle signal

The top of Fig. 7 shows the KF algorithm performance for the PMU voltage angle signal. Similar to the voltage magnitude, the KF compensates for missing data (can be seen in zoomed in view). The RLV also excites the small signal oscillations in PMU voltage angle signal. These dynamic components are removed by updating the value of R, in each iteration in the correction step of KF algorithm (response of R shown in bottom of Fig. 7). Hence smoother estimation for the voltage angle is obtained as shown in red in the top of Fig. 7.

Fig. 5. Difference between voltage responses from simulation and an actual PMU for the same disturbance

C. Impact of varying rolling window length on smoothing

As mentioned in Section III-B, the size of the Rolling Window (RW) affects the smoothing of the states in the presence of outliers and oscillations. By varying the RW length, the updating of R varies, resulting in different responses for signal smoothing. Fig. 8 shows different responses for the PMU voltage magnitude with different lengths of RW. For length of $RW = 0.5$ seconds, a fast KF response is obtained together with a smoothed output which captures the exponential decay of the small signal oscillations. However, as soon as the RW length increases, the response of the KF becomes slower. For example the response for $RW = 5$ seconds (middle part of Fig. 8, zoomed in view of top part) is slow and it could not produce accurate smoothing. It can be thought of an over filtered response. The bottom of Fig. 8 shows different responses when updating R with variable RW length.

Fig. 8. Impact of variation in rolling window size on smoothing

D. Performance Analysis

Table I shows the performance analysis of the KF algorithm for different RW lengths. The value of standard deviation (SD) is minimum for RW = 0.5 sec, which means that certainty is maximum for this case. Secondly, the areas between the steady state value and the filtered signals are calculated using (8):

$$A_i = \int_{t_1}^{t_2} f_0(t)dt - \int_{t_1}^{t_2} f_i(t)dt \quad (8)$$

where f_0 is a straight line at (t_1, y_1) and (t_2, y_2) and $i = 1, 2$ and 3 correspond to the filtered signals with RW lengths of 0.5, 2 and 5 sec, respectively (see Fig. 8). The integral is computed in MATLAB using the trapezoidal numerical integration `trapz` command.

The evaluation metric A/K , i.e. per unit area, has the highest value for the case RW = 0.5 sec. Here, K is the base value for calculating per unit area. Table II shows the computational time for different RW lengths. As the calculated results varied stochastically, the computations were repeated 50 times to obtain averaged values, as shown in Table II. It is evident that RW = 0.5 sec has lowest computational effort. Hence, the performance analysis of this KF algorithm shows that a faster response is obtained with a small RW length and more adequate signals to be used in steady state applications are produced. This is because a small RW length gives better smoothing with less computational effort.

TABLE I
KF PERFORMANCE METRICS FOR DIFFERENT RWS

$RW(sec)$	R_{min}	R_{max}	R_m	$SD(\sigma)$	$Area$	A/K
0.5	0.027	72.9	6.83	8.31	1401973	0.88
2	0.023	146	35.6	21.07	1168762	0.77
5	0.0152	456	106.81	68.50	810802	0.54

V. CONCLUSIONS AND FUTURE WORK

The results in this paper show that the proposed method using Kalman Filters is suitable for processing PMU data to be fed to steady state applications. It has been shown that by

TABLE II
AVERAGE COMPUTATIONAL EFFORT FOR DIFFERENT RWS

$RW(sec)$	$CompTime(sec)$
0.5	4.3272
2	4.3543
5	4.4320

updating the value of measurement noise covariance matrix R , appropriately, dynamics can be filtered out from PMU data to obtain a smoother response to be used in steady state applications. The proposed KF method has been developed for real time implementation, as it does not require all previous data points to predict the next state, instead it recursively conditions the current state on all the past measurements. Currently, the authors are working on a real time implementation of this method using LABVIEW, results from this work will be presented in our future publications.

ACKNOWLEDGMENTS

This work was supported in part by EU-funded FP7 IDE4L Project, the STandUP for Energy Collaboration Initiative and by Statnett SF, the Norwegian Transmission System Operator.

REFERENCES

- [1] IDE4L: Ideal grid for all. [Online]. Available: <http://www.ide4l.eu/>
- [2] IEEE Std C37.118, "IEEE standard for synchrophasor measurements for power systems," no. 2, pp. 1–2011, 2006.
- [3] M. Almas, J. Kilter, and L. Vanfretti, "Experiences with steady-state PMU compliance testing using standard relay testing equipment," in *Electric Power Quality and Supply Reliability Conference (PQ)*, 2014, June 2014, pp. 103–110.
- [4] Q. Zhang, X. Luo, D. Bertagnolli, S. Maslennikov, and B. Nubile, "PMU data validation at ISO new england," in *Power and Energy Society General Meeting (PES)*, 2013 IEEE, July 2013, pp. 1–5.
- [5] D. Shi, D. Tylavsky, and N. Logic, "An adaptive method for detection and correction of errors in PMU measurements," in *Power and Energy Society General Meeting (PES)*, 2013 IEEE, July 2013, pp. 1–1.
- [6] L. Vanfretti, J. Chow, S. Sarawgi, and B. Fardanesh, "A phasor-data-based state estimator incorporating phase bias correction," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 111–119, Feb 2011.
- [7] J. Tate and T. Overbye, "Extracting steady state values from phasor measurement unit data using fir and median filters," in *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, March 2009, pp. 1–8.
- [8] L. Zanni, S. Sarri, M. Pignati, R. Cherkaoui, and M. Paolone, "Probabilistic assessment of the process-noise covariance matrix of discrete kalman filter state estimation of active distribution networks," in *International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 2014.
- [9] J. Giri, D. Sun, and R. Avila-Rosales, "Wanted: A more intelligent grid," *Power and Energy Magazine, IEEE*, vol. 7, no. 9, pp. 34–40, March 2009.
- [10] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*, 1995.
- [11] P. S. Maybeck, *Stochastic models, estimation and control. Volume I*, A. Press, Ed., 1979.
- [12] The Opal-RT Technologies, Inc. OP5600 Hardware-in-the-Loop (HIL) simulator description. [Online]. Available: http://www.opal-rt.com/product/op5600-hil-hardware-in-the-loop-computer-and-io-system?quicktabs_3=7#quicktabs-3
- [13] National Instruments. NI cRIO-9074 description. [Online]. Available: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/203964>
- [14] M. Paolone, A. Borghetti, and C. A. Nucci, "A synchrophasor estimation algorithm for the monitoring of active distribution networks in steady state and transient conditions," in *Proc. of the 17th Power Systems Computation Conference (PSCC 2011)*, Stockholm, Sweden, 2011.
- [15] Megger. Single Phase Relay Tester (SMRT1) description. [Online]. Available: <http://www.megger.com/us/products/ProductDetails.php?ID=1529>