**Referee: 1**

Recommendation:

Needs Major Revision (revised Paper Will Be Sent Out For Re-review)

Comments:

The modeling ideas proposed in this work are mathematically sound and interesting. The main concern I have is with the evaluation of the practical benefits of the proposed approach. The authors outline several goals informed by a need for real-time data processing. These include operating with incomplete information, uncertainty and doing so with relatively low complexity algorithms. To further strengthen the case for the proposed techniques I would suggest the following:

- Real datasets are used in the empirical evaluation. While the results are convincing (and realistic) they are also somewhat limited to just two deployments, with small numbers of sensors and specific characteristics in terms of correlation. Do these results generalize to larger deployments? to cases where there is less correlation between sensor measurements? based on the results provided it is hard to determine whether similarly good results can be achieved in general. As an example, even for a relatively modest size deployment (16 sensors) in the second real dataset, the authors already need to make simplifications in their model (p. 20). Potentially a broader experimental evaluation with simulated data might help to make this case. A more general discussion of the kinds of networks sizes for which these approaches are suited would also be helpful (can they be used for networks of 100s of nodes?).

**Experiments included on MIDAS dataset, comprising 670 sensors.**

- The evaluation of complexity is also unconvincing. First, there are no comparisons with other methods, so we don't know how much savings the proposed method leads to. Also, the table that is provided gives actual computation values in seconds in an off the shelf computer using Matlab. The authors should explain whether this is a realistic setting. From the paper it is not clear what is required in order to operate in "real-time". Is it possible to describe the kinds of compute resources, delay constraints, etc. that would apply in a desirable "low cost" deployment?

**Updated computational cost tables, including a table describing the numbers of hyperparameter samples and stored data for the tested examples.**

**Referee: 2**

Recommendation:

Needs Major Revision (revised Paper Will Be Sent Out For Re-review)

Comments:

The implementation and evaluation sections could be expanded, for completeness and also to make the technique more accessible to an estimation/learning non-specialist. Some specific suggestions are as follows:

1. Consider providing a 'roadmap' for going from the theory and equations to an implementation i.e. starting with observations, outline the sequence of computations leading to the final results.

**Included pseudocode for the algorithm.**

2. Since the limitations of and improvement over state-space models provides a key motivation for the new technique, this aspect could be brought out more strongly. In addition to the discussion in Section 7, highlighting the same in a specific manner for the implementation and evaluation in Section 5 will be useful. Provide more detail on the Kalman filter implementation to convince the reader that a fair comparison is being made. Consider including a table with results for the comparative evaluation for several different instances/variables/experiments. Also, use SNR for performance evaluation, in addition to RMSE.

**Included Normed Mean Square Error as a measure of algorithm comparison.**

3. Add detail to empirical evaluation. To help connect with the theory in Section 3, for each environmental variable in Section 5,

- identify the hyperparameters

- show choice of covariance functions

**Covariance functions described for all datasets.**

Typos:

Pg 4: "large quantities of data timestamped data."

**corrected**

Pg 13: "parses and stores is using Jena"

**corrected**

Pg 15: "Air temperature was chosen since they exhibit"

**corrected**