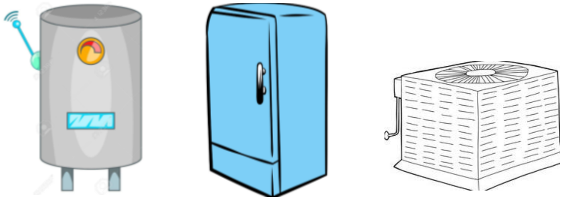
Exploring Connections Between a Multiple Model Kalman Filter and Dynamic Fixed Share with Applications to Demand Response

Gregory S. Ledva, Laura Balzano, and Johanna L. Mathieu
The University of Michigan

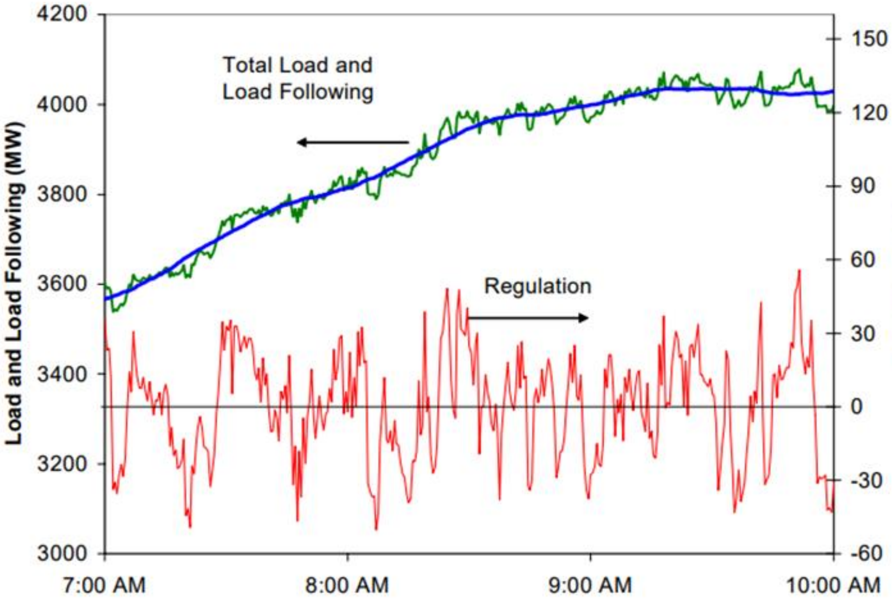
This research was funded under NSF Grant #ECCS-1508943

The power consumption of TCLs can be coordinated to help the electric power grid balance supply and demand of electricity

- Large populations of TCLs



- Often residential air conditioners (ACs)
- Can provide frequency regulation
- Common to assume an aggregator receives a desired power signal and controls an aggregation of loads to match that signal
- Rely on control and estimation algorithms
- Estimates of the aggregate AC demand can be used as a feedback signal for control and estimation algorithms



In this work, we apply Kalman filter and online learning algorithms to estimate the aggregate AC demand.

- **Contribution 1:** We show that Dynamic Fixed Share (DFS) can be constructed to produce identical estimates to those produced by a multiple model Kalman filter (MMKF)

- **Contribution 2:** We incorporate three heuristics used within MMKFs into DFS

- **Contribution 3:** We compare the performance of DFS and a MMKF within a demand response simulation

Contents

A Kalman filter relies on a linear model of the system and normally-distributed random variables to estimate a state.

- **System model:** $x_{k+1} = A_k \, x_k + \omega_k$

$$y_k = C_k \, x_k + v_k$$

- **Model-based update:** $\widehat{x}_{k+1} = A_k \, \widetilde{x}_k$

- **Measurement-based update:** $\widetilde{x}_k = \widehat{x}_k + \widehat{P}_k C_k^T \left[C_k \widehat{P}_k C_k^T + R_k \right]^{-1} (y_k - C_k \widehat{x}_k)$

- **Convex optimization formulation [Mattingley 2010]:**

$$\begin{array}{ll} \min\limits_{x, v_k} & v_k^{\mathrm{T}} R_k^{-1} v_k \; + \; (x - \widehat{x}_k)^{\mathrm{T}} \, \widehat{P}_k^{-1} \, (x - \widehat{x}_k) \\ \text{s.t.} & y_k = C_k x + v_k . \end{array}$$

[Simon 2006]

MPEL Michigan Power & Energy Laboratory

Similar to a Kalman filter, DMD uses a system model and a convex optimization formulation to update the predictions

- **System model:** Assumed below

- **Model-based update:** $\widehat{x}_{k+1} = \Phi(\widetilde{x}_k)$

- **Measurement-based update:** Depends on the user construction of the convex optimization function below

- **Convex optimization formulation:** $\widetilde{x}_k = \underset{x \in \mathcal{X}}{\operatorname{argmin}} \ \eta^s \left(\nabla \ell_k(\widehat{x}_k)\right)^T x + D\left(x \| \widehat{x}_k\right)$

[Hall 2015]

DMD can produce identical updates to a Kalman filter if the user-defined functions and parameters are chosen appropriately.

- **Desired update:** $\widetilde{x}_k = \widehat{x}_k + \widehat{P}_k C_k^T \left[C_k \widehat{P}_k C_k^T + R_k \right]^{-1} (y_k - C_k \widehat{x}_k)$

1. **Choose the Bregman divergence:** $D\left(x \| \widehat{x}_k\right) = \frac{1}{2}\left(x - \widehat{x}_k\right)^T \widehat{P}_k^{-1} \left(x - \widehat{x}_k\right)$
2. **Set step-size parameter:** $\eta^{\mathrm{s}} = 1$
3. **Solve the convex optimization formulation:** $\widetilde{x}_k = \widehat{x}_k + \widehat{P}_k \left(-\nabla \ell_k(\widehat{x}_k)\right)$
4. **Select the loss function to match the desired update:**

$$\ell_k(\widehat{x}_k) = \frac{1}{2} \left(C_k \widehat{x}_k - y_k\right)^T (\widehat{P}_k^{\mathrm{y}})^{-1} \left(C_k \widehat{x}_k - y_k\right)$$

$$-\nabla \ell_k(\widehat{x}_k) = C_k^T (\widehat{P}_k^{\mathrm{y}})^{-1} \left(y_k - C_k \widehat{x}_k\right)$$

[Ledva 2018]

An MMKF combines estimates from separate underlying Kalman filters using a Gaussian likelihood function.

- **Likelihood function:**

$$h(y_k|m^i) = \left[(2\pi)^{q/2}\sqrt{|\widehat{P}_k^{\mathrm{y},i}|}\right]^{-1} \exp\left(-\frac{1}{2}d_k^{\mathrm{y}}(\widehat{y}_k^i)\right)$$

$$d_k^{\mathrm{y}}(\widehat{y}_k^i) = (\widehat{y}_k^i - y_k)^T(\widehat{P}_k^{\mathrm{y},i})^{-1}(\widehat{y}_k^i - y_k)$$

- **Weighting function:**

$$w_{k+1}^i = \frac{h(y_k|m^i)\ w_k^i}{\sum_{j\in\mathcal{M}^{\mathrm{mdl}}} h(y_k|m^j)\ w_k^j}$$

- **Overall prediction:**

$$\widehat{x}_{k+1} = \sum_{i\in\mathcal{M}^{\mathrm{mdl}}} w_{k+1}^i\ \widehat{x}_{k+1}^i$$

[Welch 2001]

DFS combines estimates for separate underlying DMD algorithms, which each incorporate different models.

- **Weighting Function:** $$w_{k+1}^{i} = \frac{\lambda}{N^{\mathrm{mdl}}} + (1-\lambda)\,\frac{w_k^i\,\exp\left(-\eta^r\,\ell_k\left(\widehat{x}_k^i\right)\right)}{\sum_{j=1}^{N^{\mathrm{mdl}}} w_k^j\,\exp\left(-\eta^r\,\ell_k\left(\widehat{x}_k^j\right)\right)}$$

- **Overall estimate:** $$\widehat{x}_{k+1} = \sum_{i\in\mathcal{M}^{\mathrm{mdl}}} w_{k+1}^i\,\widehat{x}_{k+1}^i$$

[Hall 2015]

We can make the weighting functions equivalent between the MMKF and DFS by scaling parameters within the functions.

- **Set parameters in DFS:** $w_{k+1}^{i} = \frac{\lambda}{N^{\text{mdl}}} + (1-\lambda)\frac{w_k^i \exp\left(-\eta^r \ell_k\left(\widehat{x}_k^i\right)\right)}{\sum_{j=1}^{N^{\text{mdl}}} w_k^j \exp\left(-\eta^r \ell_k\left(\widehat{x}_k^j\right)\right)}$

- **Both weight updates now have the form:** $w_{k+1}^{i} = \frac{h(y_k|m^i)\, w_k^i}{\sum_{j\in\mathcal{M}^{\text{mdl}}} h(y_k|m^j)\, w_k^j}$

- **Scale output covariance:** $h(y_k|m^i) = \left[(2\pi)^{q/2}\sqrt{|\widehat{P}_k^{y,i}|}\right]^{-1}\exp\left(-\frac{1}{2}d_k^y(\widehat{y}_k^i)\right)$

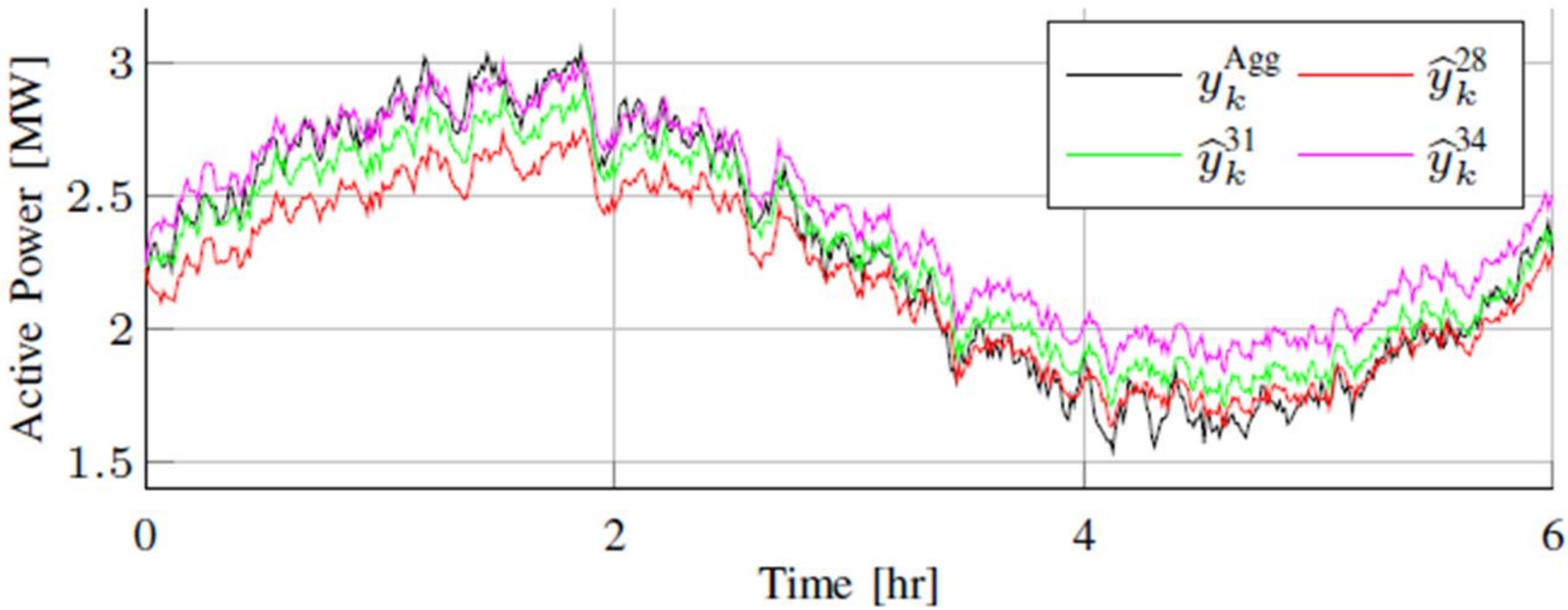
We show that several heuristic adjustments that are used within MMKFs can be incorporated into DFS by modifying the weight update.

- **A minimum weight:** $$w_{k+1}^{i} = \frac{\lambda}{N^{\text{mdl}}} + (1-\lambda)\,\frac{w_k^i \exp\left(-\eta^r\,\ell_k\left(\widehat{x}_k^i\right)\right)}{\sum_{j=1}^{N^{\text{mdl}}} w_k^j \exp\left(-\eta^r\,\ell_k\left(\widehat{x}_k^j\right)\right)}$$

- **Exponential decay:** $$w_{k+1}^{i} = \frac{\lambda}{N^{\text{mdl}}} + (1-\lambda)\,\frac{(w_k^i)^{\gamma} \exp\left(-\eta^r\,\ell_k(\widehat{x}_k^i)\right)}{\sum_{j=1}^{N^{\text{mdl}}} (w_k^j)^{\gamma} \exp\left(-\eta^r\,\ell_k(\widehat{x}_k^j)\right)}$$

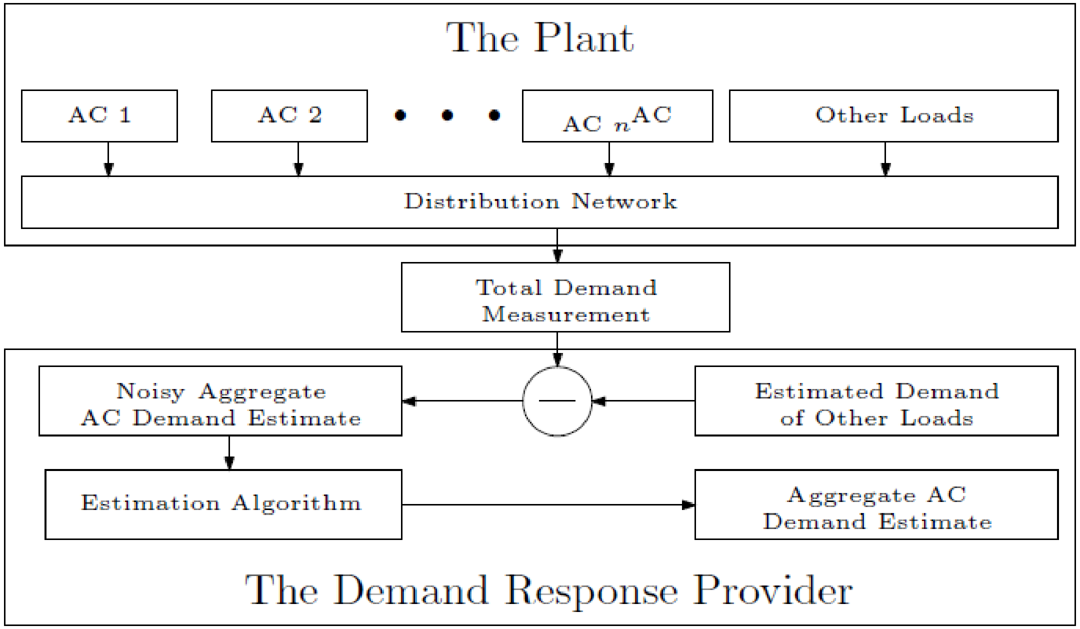
- **Sliding Window:** $$w_{k+1}^{i} = \frac{\lambda}{N^{\text{mdl}}} + (1-\lambda)\,\frac{\prod_{t=k-N^{\ell}}^{k} \exp\left(-\eta^r\,\ell_t\left(\widehat{x}_t^i\right)\right)}{\sum_{j=1}^{N^{\text{mdl}}} \prod_{t=k-N^{\ell}}^{k} \exp\left(-\eta^r\,\ell_t\left(\widehat{x}_t^j\right)\right)}$$

We simulate a demand response scenario to evaluate the ability of different algorithm implementations to estimate the aggregate demand.



- **Each algorithm uses three underlying models to form its overall prediction**
- **Identical individual predictions across the multiple model algorithm implementations**

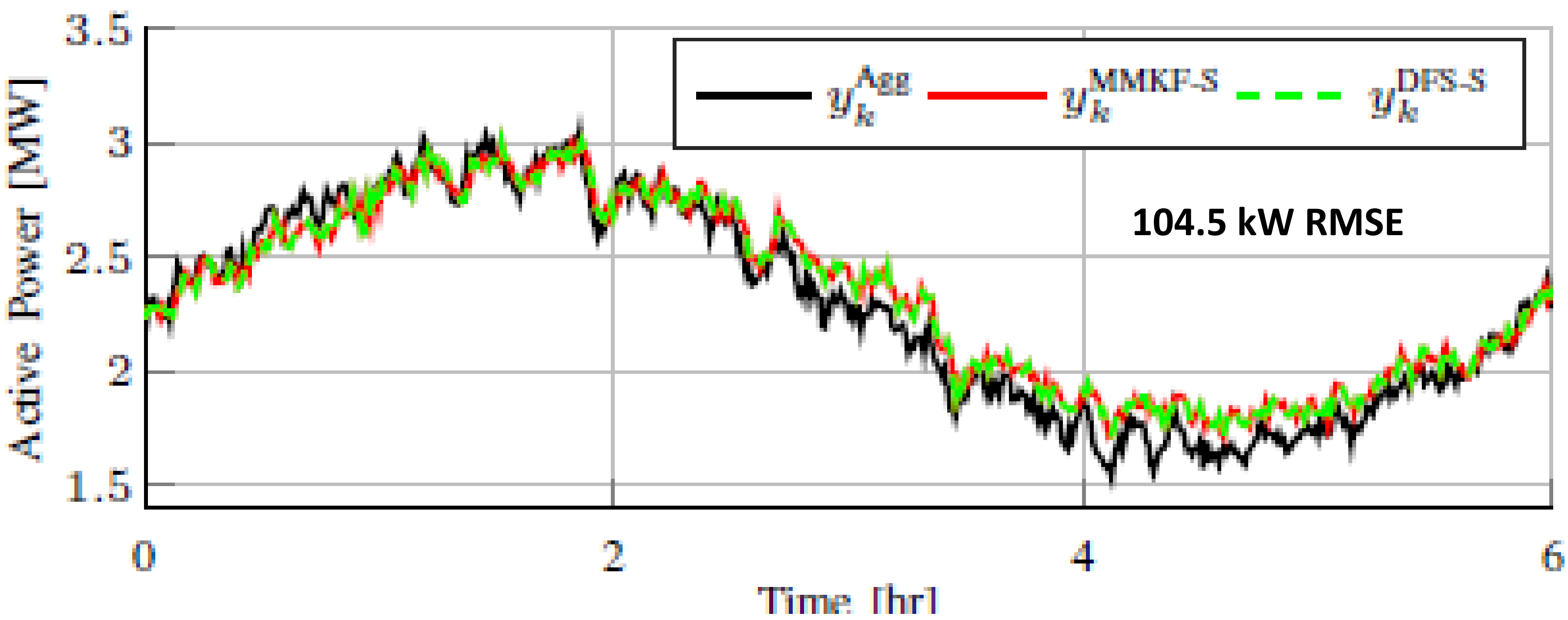
We simulate a population of 1,000 ACs over six hours using 4 second time-steps and a sinusoidal outdoor temperature.



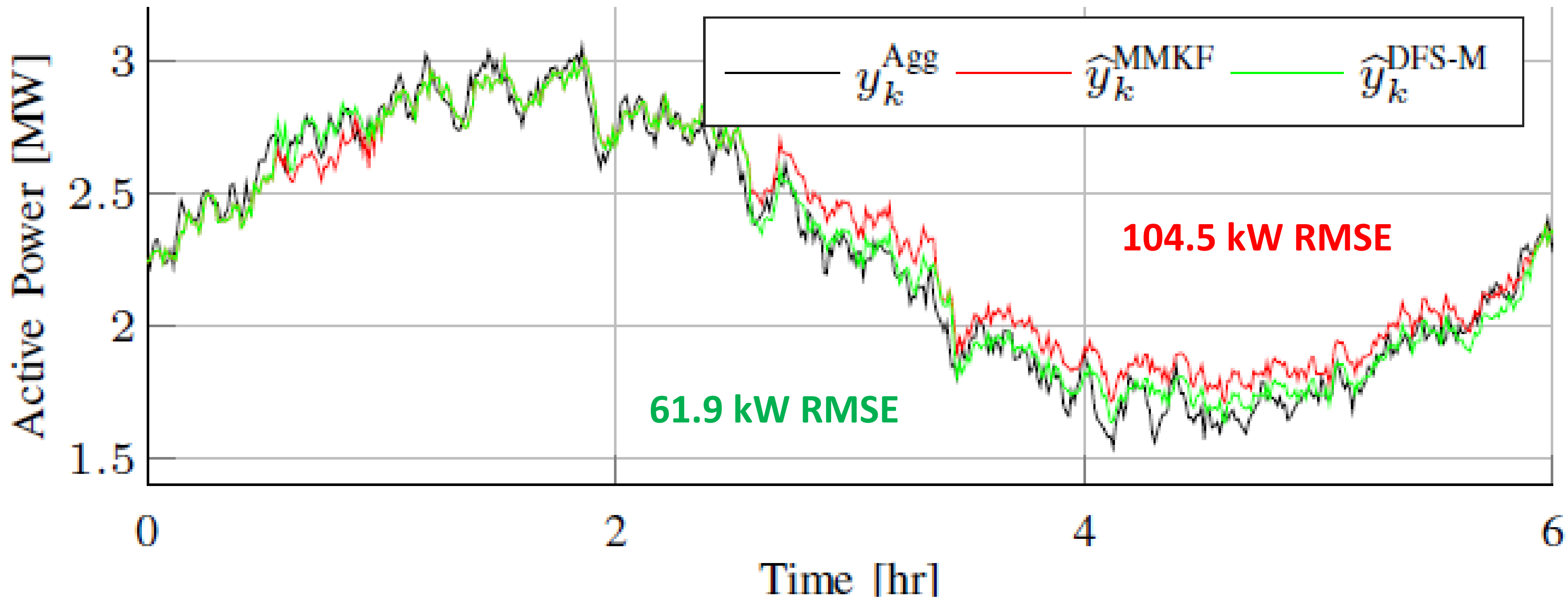
$$\theta_{k+1} = a\,\theta_k + (1-a)\,(\theta_k^{o} - m_k \Lambda P)$$

$$m_{k+1} = \begin{cases} 0 & \text{if } \theta_{k+1} < \theta^{\text{set}} - \frac{\theta^{\text{db}}}{2} \\ 1 & \text{if } \theta_{k+1} > \theta^{\text{set}} + \frac{\theta^{\text{db}}}{2} \\ m_k & \text{otherwise,} \end{cases}$$

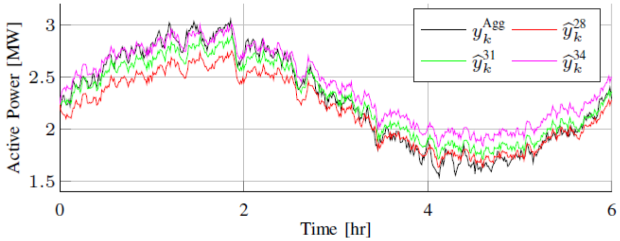
Performing the scaling of the MMKF weight update produces the same estimates for both the MMKF and DFS

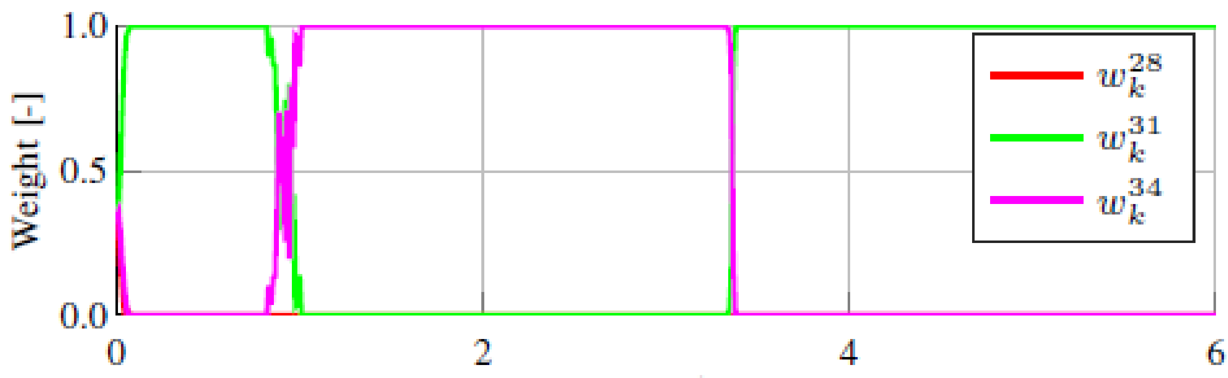


Including a minimum weight into DFS and a MMKF significantly improves the estimation accuracy



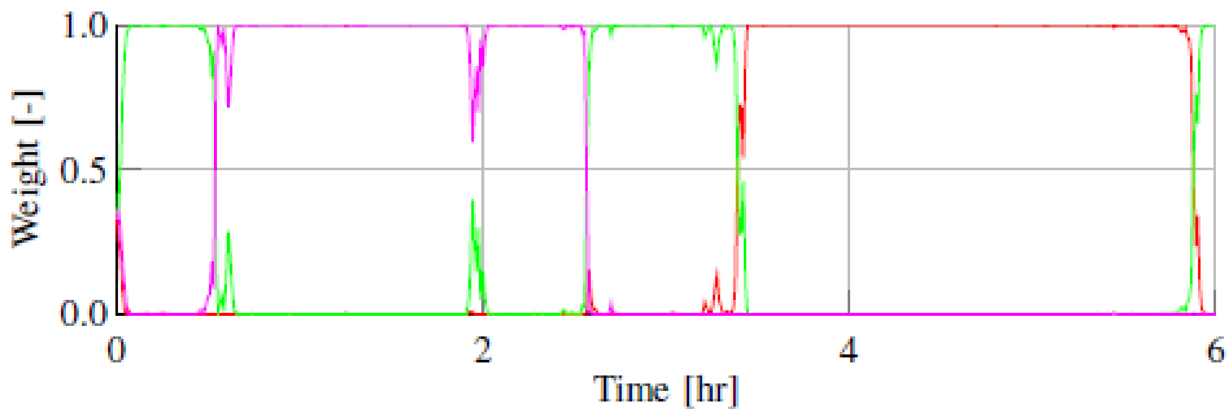
A minimum weight in MMKF allows the algorithm to shift weight to the most accurate model near the end of the simulation.





(a) MMKF Weights

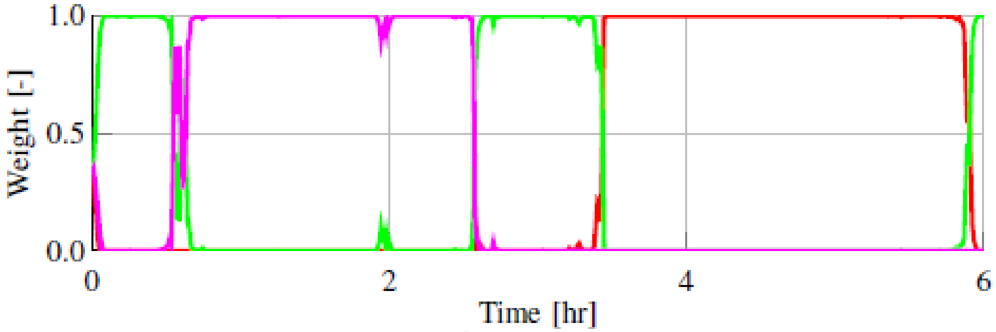
104.5 kW RMSE



(b) MMKF-M Weights

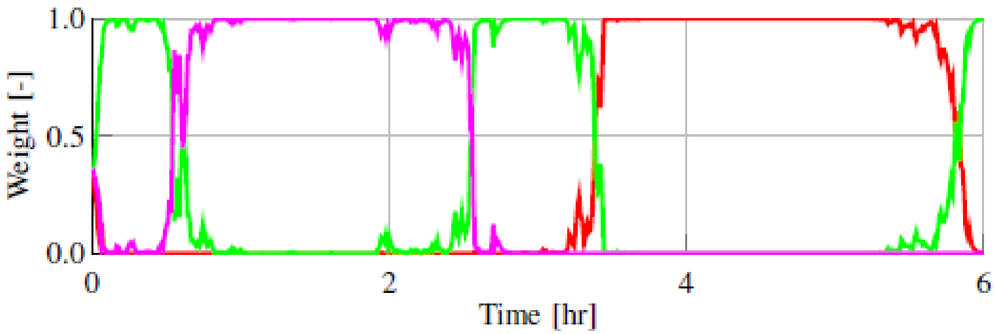
63.4 kW RMSE

Incorporating a sliding window and exponential decay allow the weights to remain dynamic as time progresses.



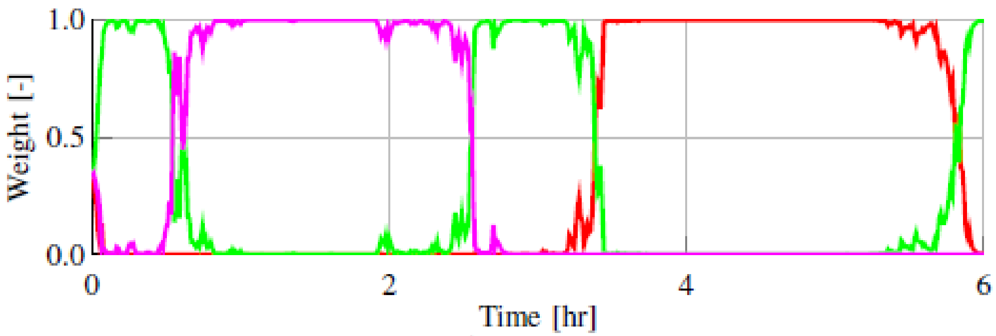
(c) DFS-M Weights

61.9 kW RMSE



(d) DFS-W Weights

61.4 kW RMSE



(e) DFS-E Weights

60.9 kW RMSE

There are close similarities between online learning methods and Kalman filter approaches

- Manipulating the weighting equations in the MMKF results in an identical weight update as in DFS
- DFS can be modified to incorporate heuristics that are commonly used within a MMKF
- Including a minimum weight threshold improves the performance of DFS and a MMKF
- Including exponential decay or a sliding window in the DFS weight update allows more consistent, responsive behavior in the weights.
- The DFS and MMKF algorithms effectively estimate the aggregate AC demand within the demand response simulation.

References

- J. Mattingley and S. Boyd, “Real-time convex optimization in signal processing,” IEEE Signal processing magazine, vol. 27, no. 3, pp. 50–61, 2010.
- D. Simon, “Optimal state estimation,” in Kalman, H Infinity, and Nonlinear Approaches, Wiley & Sons, 2006.
- E. C. Hall and R. M. Willett, “Online convex optimization in dynamic environments,” IEEE Journal of Selected Topics in Signal Processing, vol. 9, no. 4, pp. 647–662, 2015.
- G. Ledva, Z. Du, L. Balzano, and J. Mathieu, “Disaggregating load by type from distribution system measurements in real-time,” in Energy Markets and Responsive Grids, I. Hiskens, S. Meyn, T. Samad, and J. Stoustrup, Eds. New York: Springer, 2018.
- G. Welch and G. Bishop, “An introduction to the Kalman filter,” http://www.cs.unc.edu/tracker/media/pdf/SIGGRAPH2001CoursePack 08.pdf, August 2001, SIGGRAPH 2001 Course 8.