Michigan Power & Energy Laboratory

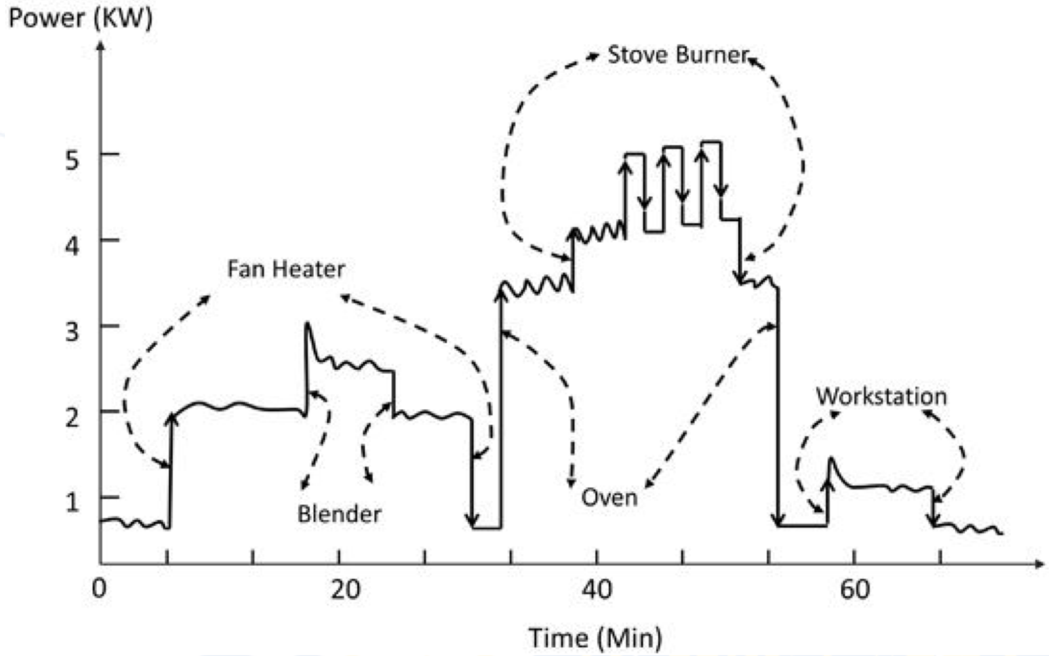
Real-Time Energy Disaggregation of a Distribution Feeder's Demand Using Online Learning

Johanna Mathieu, Gregory Ledva, Laura Balzano, Zhe Du
Department of Electrical Engineering & Computer Science
University of Michigan

Supported by NSF Grant ECCS-1508943

Energy Disaggregation

a.k.a Non-Intrusive Load Monitoring (NILM)



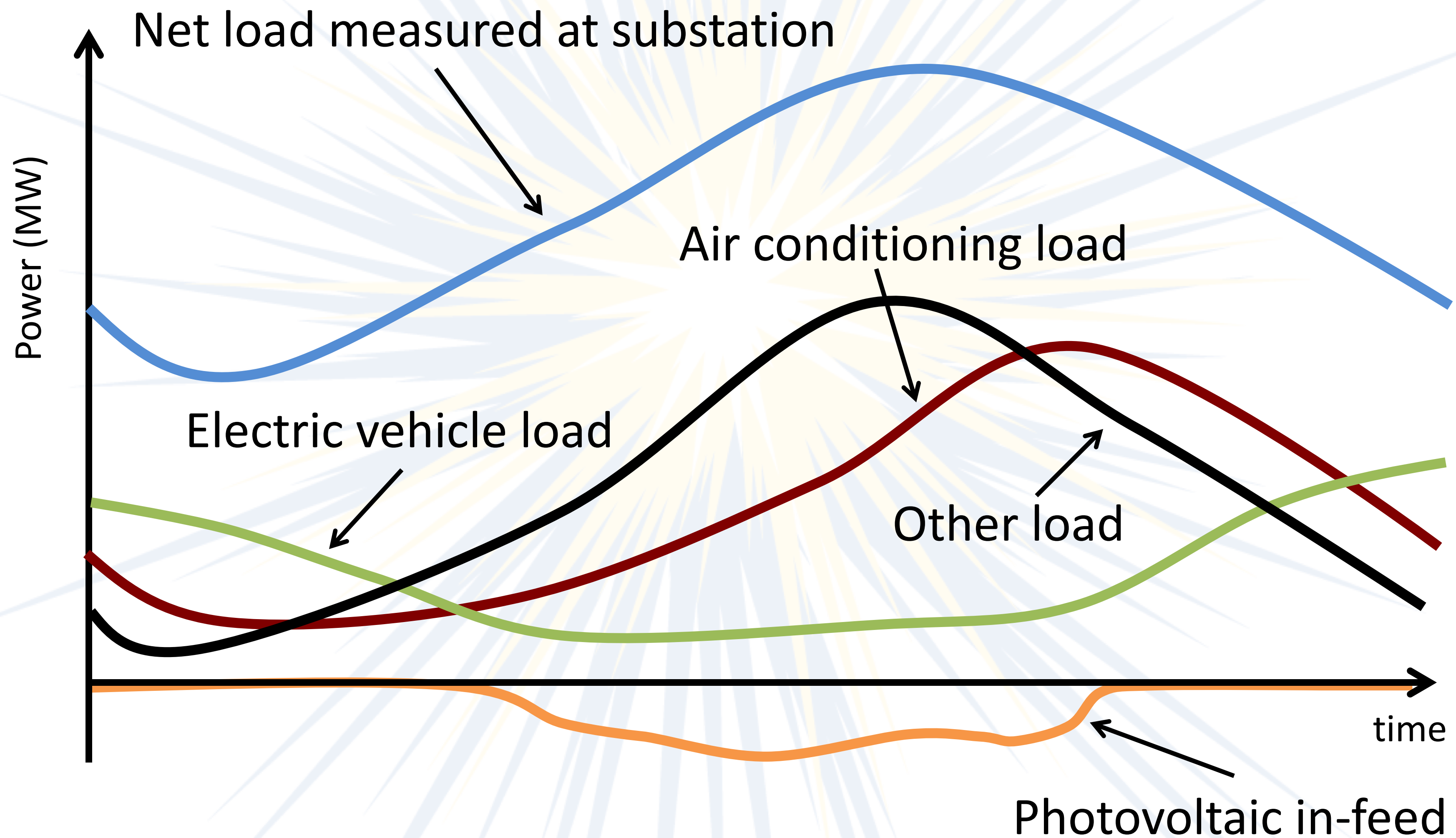
[Hart 2010; Ziefman and Roth 2011; Berges et al. 2009; Dong, Sastry, et al. 2013, 2014; Wytock & Kolter 2013; Kolter and Jaakkola 2012; Kim et al. 2010; …]

Fig from Zoha et al. 2012

Problem: Estimate individual load from a single power measurement (usually) sampled at high frequency (10kHz-1MHz) from the household main

Solution approaches: offline algorithms including change detection, supervised learning, unsupervised learning

Feeder Energy Disaggregation



Michigan Power & Energy Laboratory

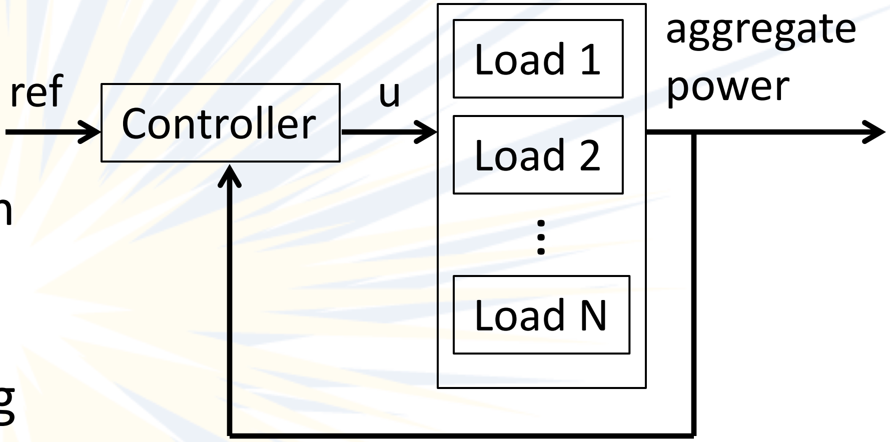
Key differences

- We assume measurements at the substation, not the household
- We estimate the power consumption of all loads of a specific type, not individual loads
- We solve the problem online, not offline
- We use lower frequency measurements (e.g., taken every second to minute)
- In some cases, we may get to be "intrusive," but not in this talk!

Why disaggregate feeder load?

Uses in demand response…

- Load control feedback [noisy aggregate power measurements are assumed in Mathieu et al. 2013; Can Kara et al. 2013; Bušić and Meyn 2016; Callaway 2009; …]
- Load aggregator bidding
- Demand response event signaling (when/how much)



Beyond demand response…

- Energy efficiency via conservation voltage reduction (disaggregate by ZIP load type)
- Contingency planning (disaggregate motor loads)
- Reserve planning (disaggregate PV production)

Possible Methods

- Short-term load (component) forecasting
  - Doesn't incorporate real-time feedback
- State estimation
  - Linear techniques require linear system models
  - Nonlinear techniques can be computationally demanding
- Online learning
  - (Typically) data-driven, model-free
- Hybrid approach: Dynamic Fixed Share & Dynamic Mirror Descent [Hall & Willet 2015]
  - Admits dynamic models of arbitrary forms
  - Optimization-based method to choose a weighted combination of the estimates of a collection of models

Michigan Power & Energy Laboratory

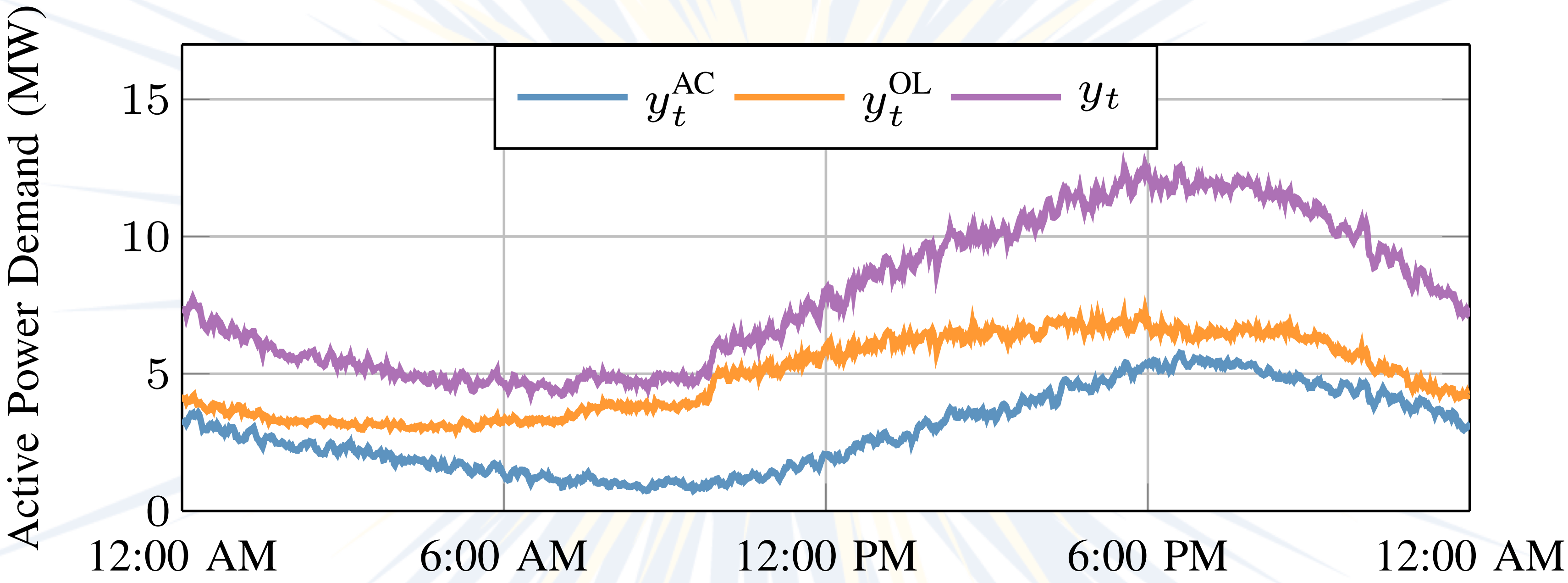
Outline

- Problem Framework
- Algorithms: Dynamic Mirror Descent (DMD) & Dynamic Fixed Share (DFS)
- Models
- Algorithm Modifications
- Case studies
- Connections with Kalman Filtering

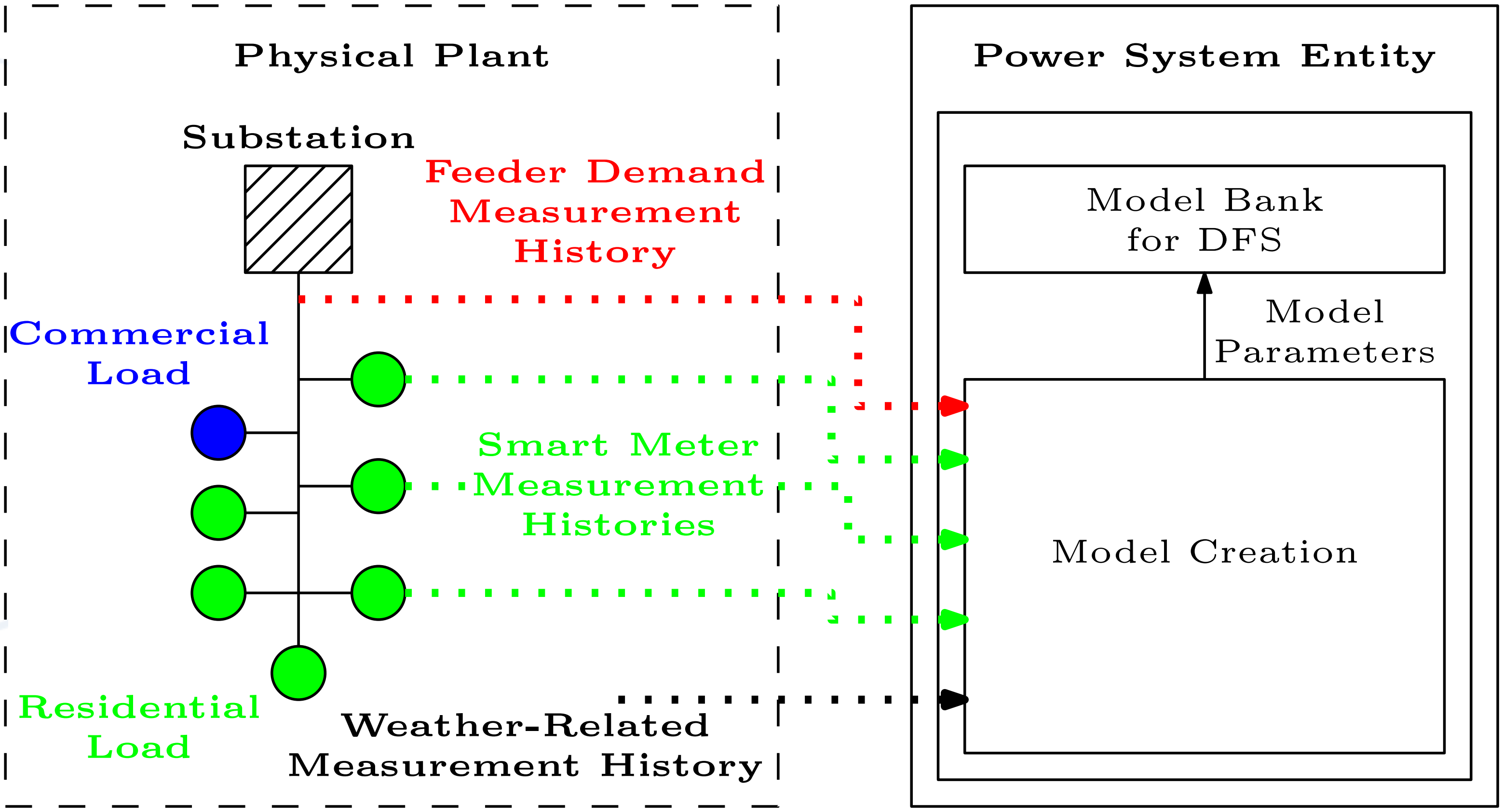
Ledva, Balzano, and Mathieu, "Inferring the Behavior of Distributed Energy Resources with Online Learning," Allerton 2015.

Ledva, Balzano, and Mathieu, "Real-time Energy Disaggregation of a Distribution Feeder's Demand using Online Learning," IEEE Transactions on Power Systems, 2018.

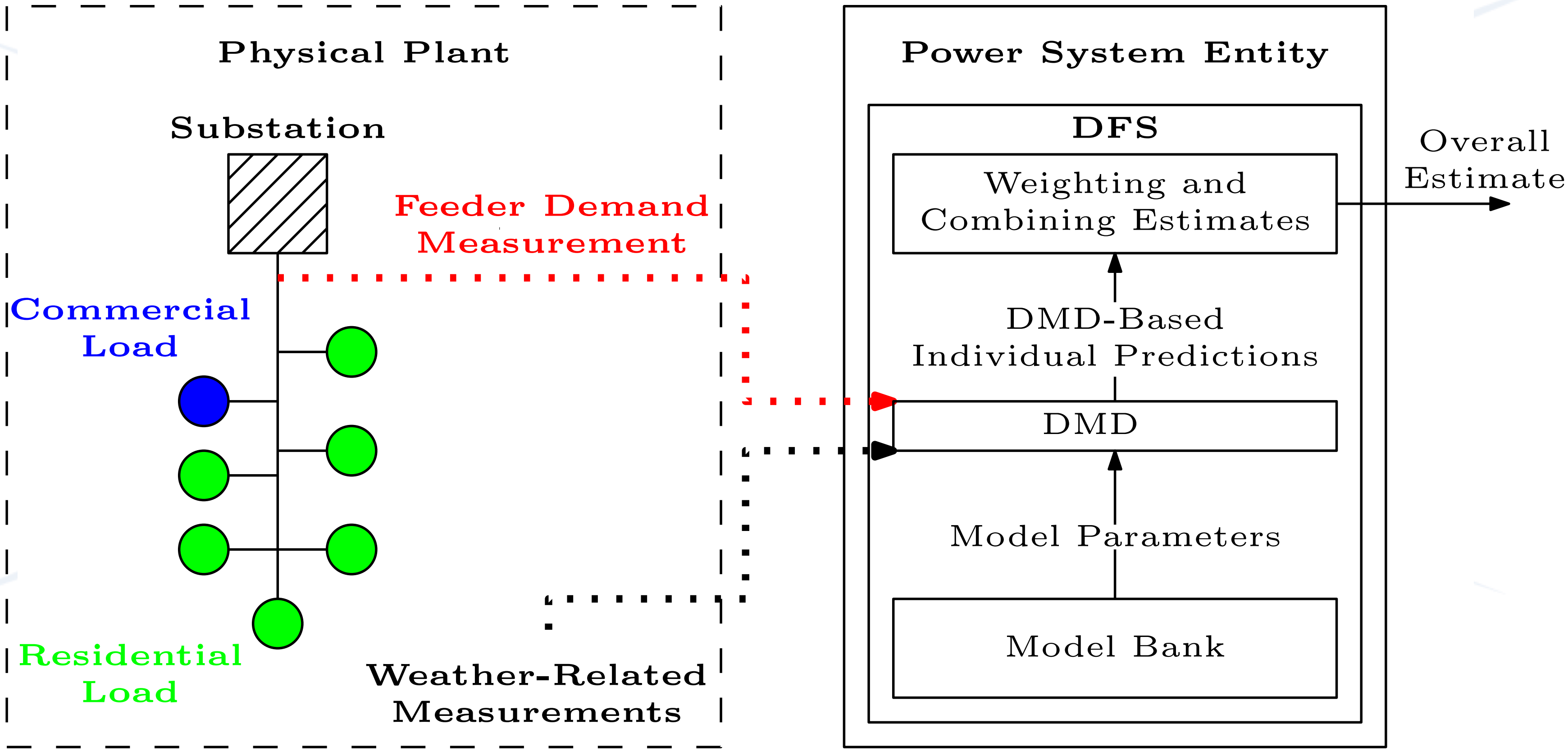
Problem Framework



Problem Framework: Offline Model Generation



Problem Framework: Real-time Estimation



Dynamic Mirror Descent

[Hall & Willet 2015]

For each model m we compute

1. an observation-based update

$$\widetilde{\theta}_t^m = \arg\min_{\theta\in\Theta} \eta^{\mathrm{s}} \left\langle \nabla \ell_t(\widehat{\theta}_t^m),\ \theta \right\rangle + D\left(\theta \| \widehat{\theta}_t^m\right)$$

where $\ell_t(\widehat{\theta}_t^m)$ is a convex loss function and D is a Bregman divergence function

2. a model-based update

$$\widehat{\theta}_{t+1}^m = \Phi^m(\widetilde{\theta}_t^m)$$

Dynamic Fixed Share

[Hall & Willet 2015]

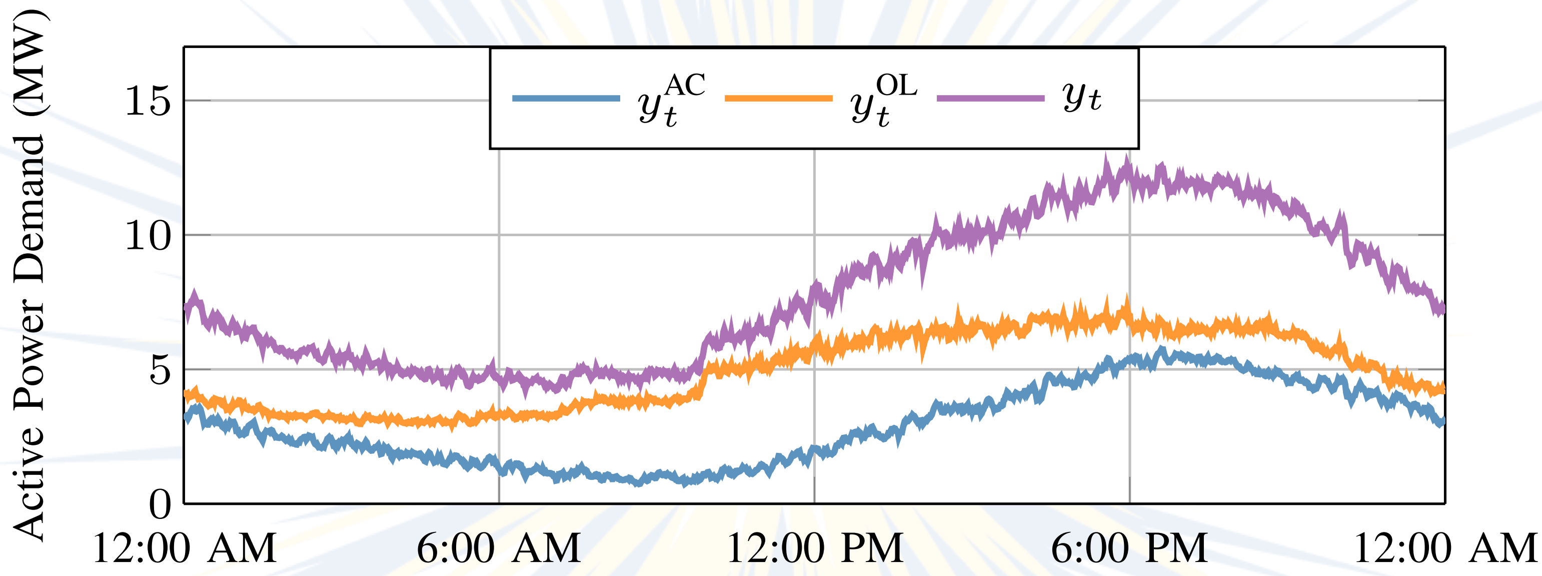
3. Next, we update the weight of each model

$$w_{t+1}^{m} = \frac{\lambda}{N^{\mathrm{mdl}}} + (1-\lambda)\frac{w_t^m \exp\left(-\eta^r \,\ell_t\left(\widehat{\theta}_t^m\right)\right)}{\sum_{j=1}^{N^{\mathrm{mdl}}} w_t^j \exp\left(-\eta^r \,\ell_t\left(\widehat{\theta}_t^j\right)\right)}$$

4. and compute the overall estimate.

$$\widehat{\theta}_{t+1} = \sum_{m\in\mathcal{M}^{\mathrm{mdl}}} w_{t+1}^{m}\, \widehat{\theta}_{t+1}^{m}$$

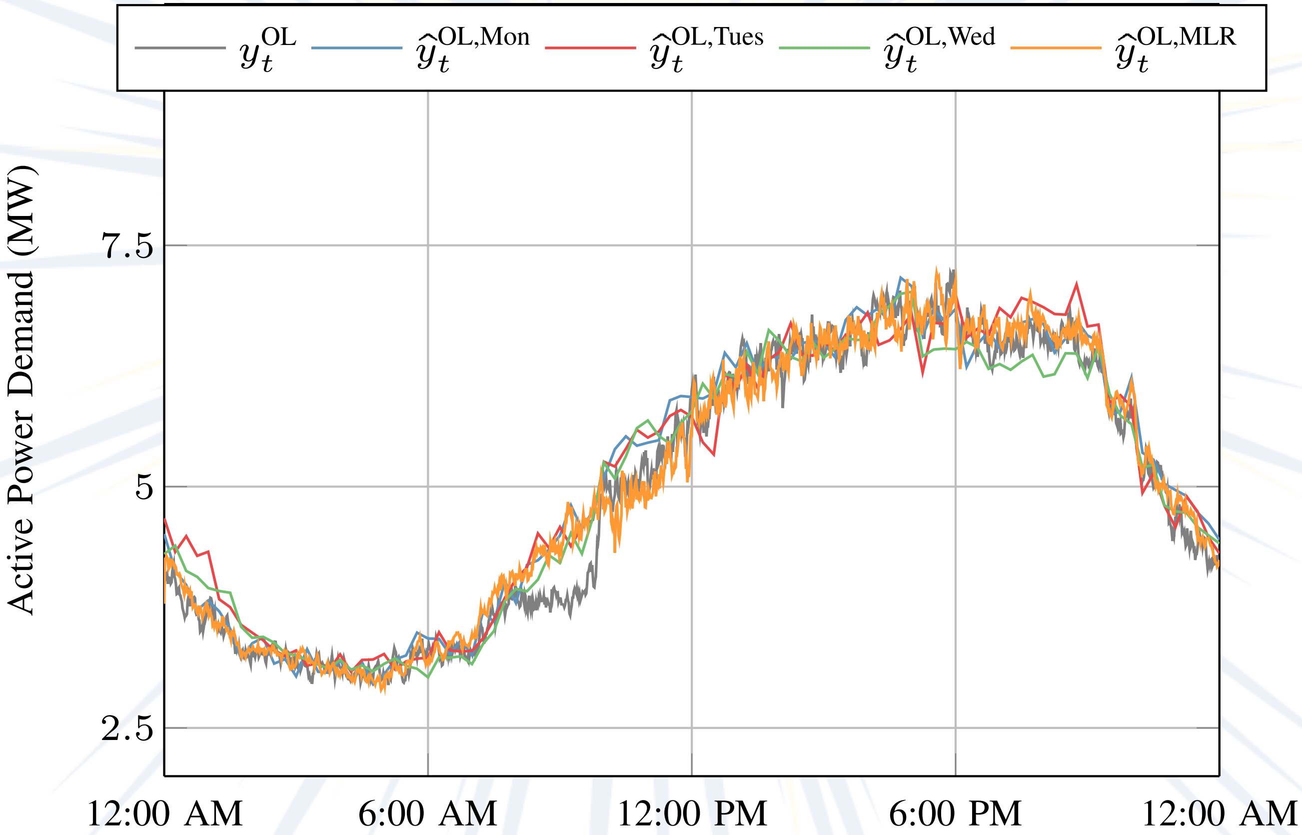
Models



- 29 aggregate air conditioning load (AC) models
- 6 "other load" (OL) model
- 1 AC model + 1 OL model = 1 total load model
  → 174 total load models

"Other load" Models

- (Smoothed) load on previous days (Mon-Fri)
- Multiple linear regression (MLR) model using time-of-week, outdoor temperature, and previous total demand measurement



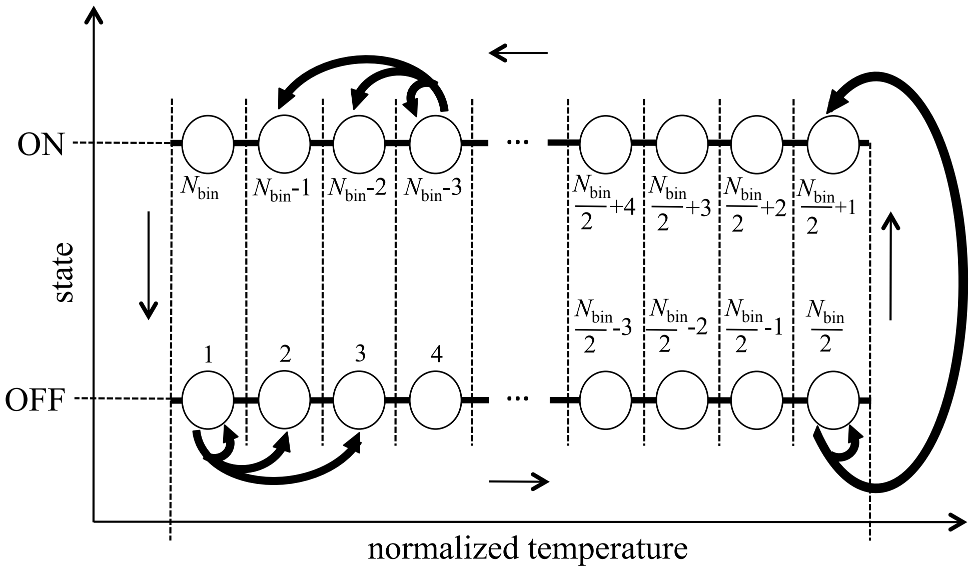
AC Models

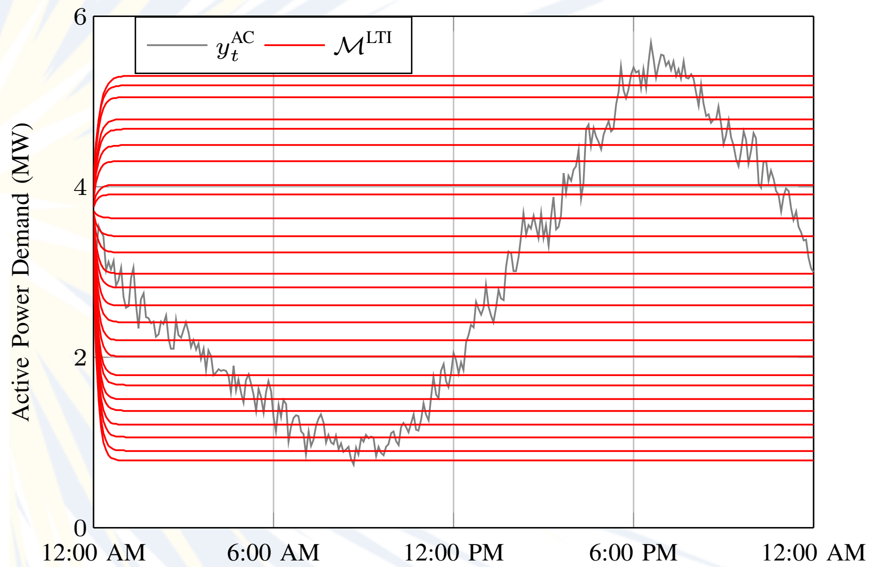
- Multiple linear regression (MLR) model using time-of-week and current/past outdoor temperatures
- Linear time-invariant (LTI) system models corresponding to different outdoor temperatures
- Linear time-varying (LTV) system models

LTI AC Model

[Mathieu et al. 2013]

$$\widehat{x}_{t+1}^{\mathrm{LTI},m} = A^{\mathrm{LTI},m}\ \widehat{x}_t^{\mathrm{LTI},m}$$
$$\widehat{y}_t^{\mathrm{AC,LTI},m} = C^{\mathrm{LTI},m}\ \widehat{x}_t^{\mathrm{LTI},m}$$





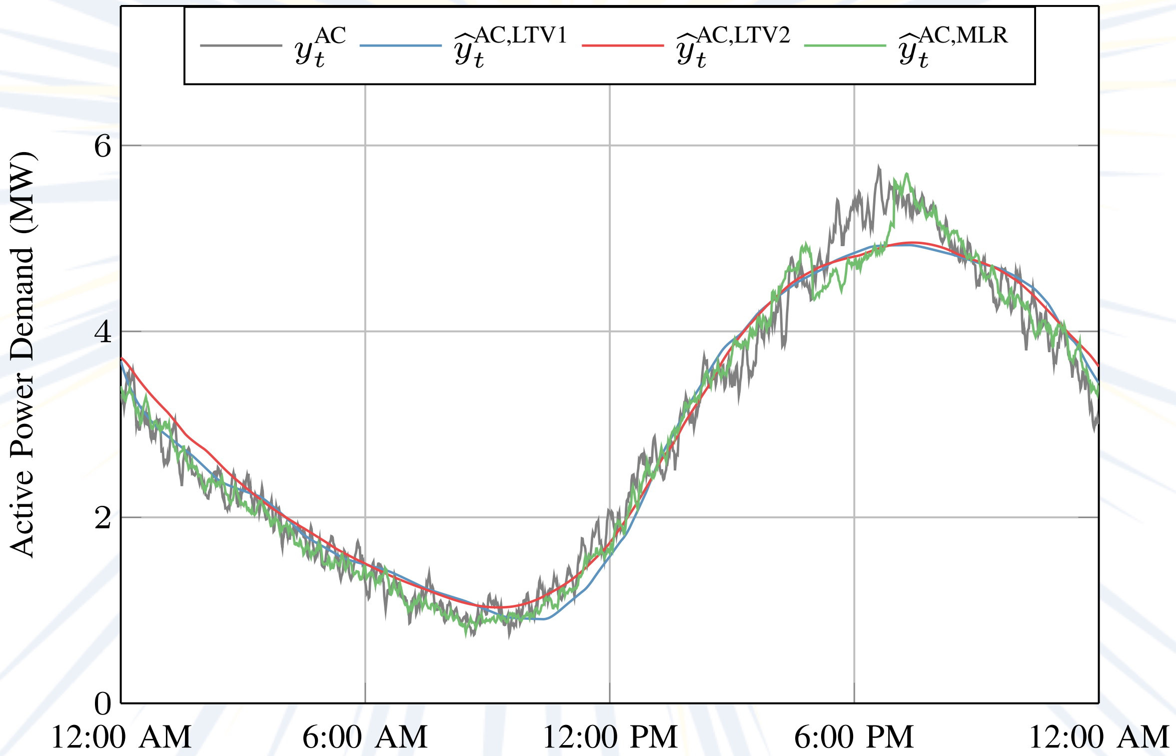
LTV AC Models

[Mathieu et al. 2015]

$$\widehat{x}_{t+1}^{\text{LTV1}} = A_t^{\text{LTV1}} \; \widehat{x}_t^{\text{LTV1}}$$

$$\widehat{y}_t^{\text{AC,LTV1}} = C_t^{\text{LTV1}} \; \widehat{x}_t^{\text{LTV1}}$$

Two variants identify A_t using delayed temperature measurements vs. a moving average of past temperature measurements



Algorithm Modifications

- The models have different structures, dynamic states, and/or parameters. It is difficult to define a common "state" θ_t.

- Two versions
  - Update Method 1: updates the output
  - Update Method 2: updates the state (i.e., for the LTI/LTV models, update the state, i.e., $\theta_t = x_t$)

Update Method 1

Adjust the output (demand estimates), rather than the state.

$$\widetilde{\theta}_t^m = \arg\min_{\theta \in \Theta} \eta^s \left\langle \nabla \ell_t(\widehat{\theta}_t^m),\ \theta \right\rangle + D\left(\theta \| \widehat{\theta}_t^m\right)$$
$$\widehat{\theta}_{t+1}^m = \Phi^m(\widetilde{\theta}_t^m)$$

$$\widehat{\kappa}_{t+1} = \arg\min_{\theta \in \Theta} \eta^s \left\langle \nabla \ell_t(\widehat{\theta}_t),\ \theta \right\rangle + D\left(\theta \| \widehat{\kappa}_t\right)$$
$$\check{\theta}_{t+1} = \Phi(\check{\theta}_t)$$
$$\widehat{\theta}_{t+1} = \check{\theta}_{t+1} + \widehat{\kappa}_{t+1}.$$

Then, with $\ell_t(\widehat{\theta}_t) = \frac{1}{2}\|C_t\widehat{\theta}_t - y_t\|_2^2$ and $D(\theta\|\widehat{\kappa}_t) = \frac{1}{2}\|\theta - \widehat{\kappa}_t\|_2^2$ we can derive a closed form update:

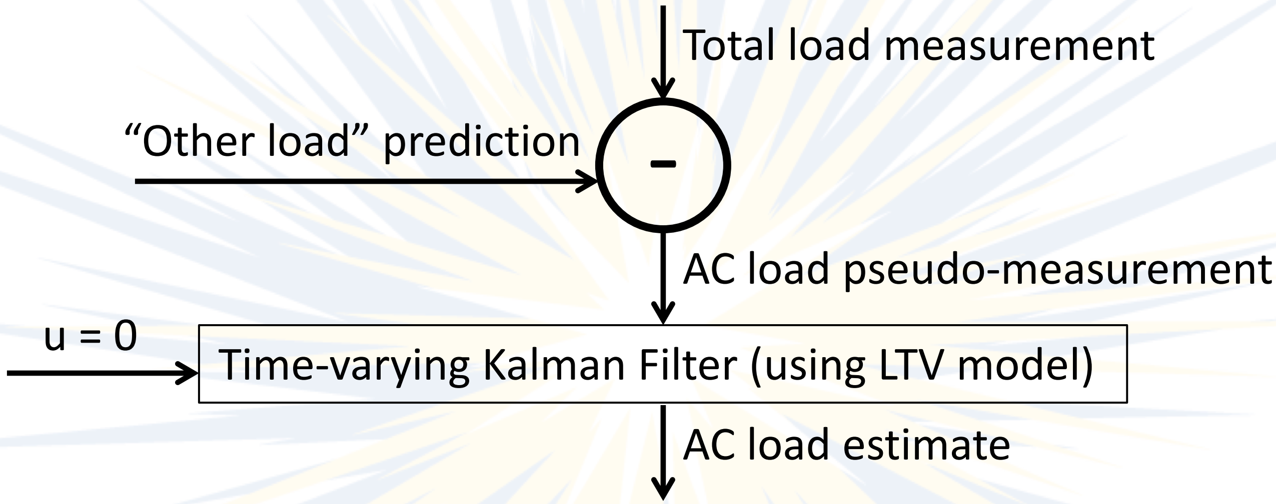
$$\widehat{\kappa}_{t+1} = \widehat{\kappa}_t + \eta^s C_t^T \left(y_t - C_t\widehat{\theta}_t\right)$$

Michigan Power & Energy Laboratory

Case studies: Plant

- Residential load and weather data from Pecan Street Dataport (Austin, TX)
- Commercial load data from Pacific Gas & Electric Company; weather data from NOAA (Bay Area, CA)
- GridLab-D feeder used to size the load

Case studies: Benchmark

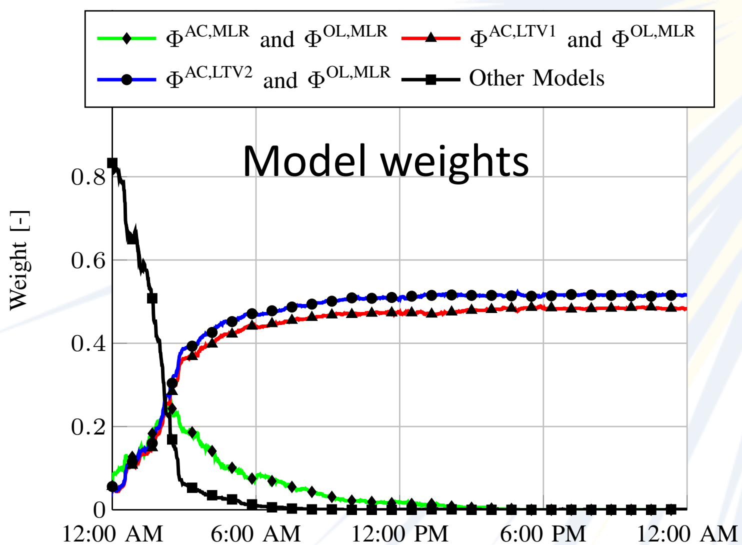


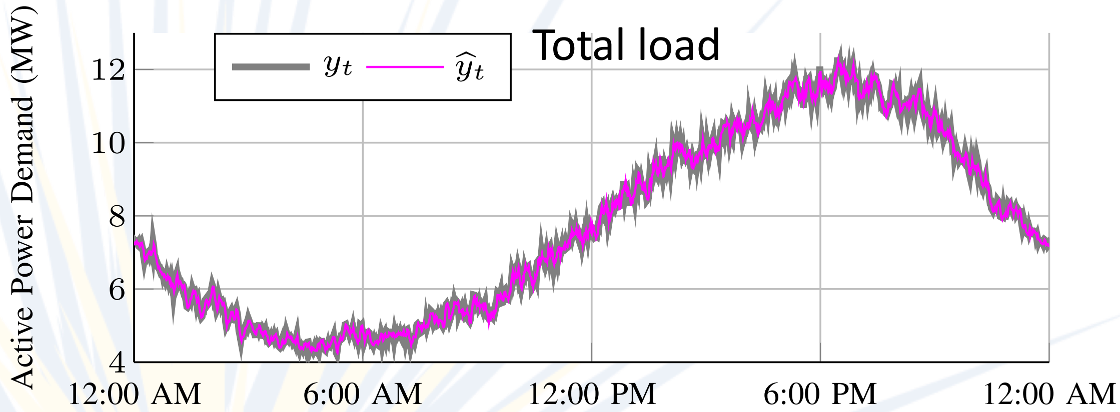
- Each "other load" model + LTV AC model combination is used to compute one AC load estimate.
- We obtain the estimates from all Kalman filters and compute the a posteriori best (BKF) and average (AKF) results.

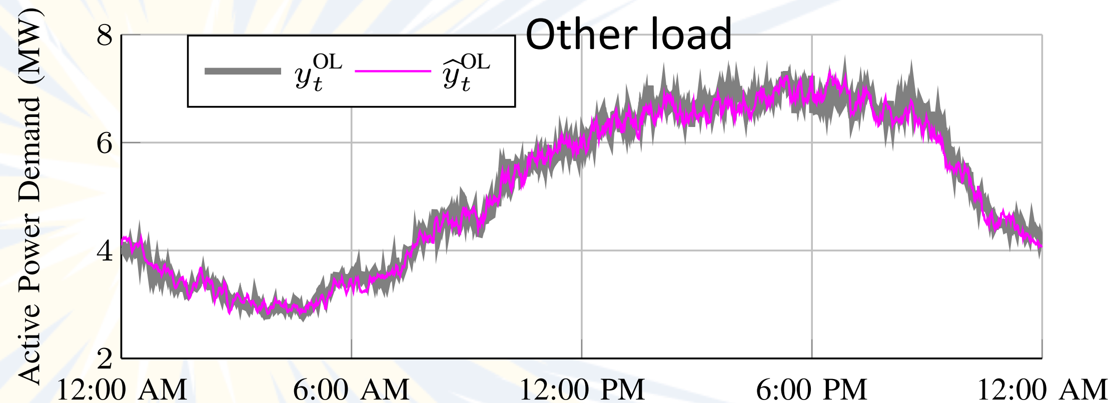
Example results

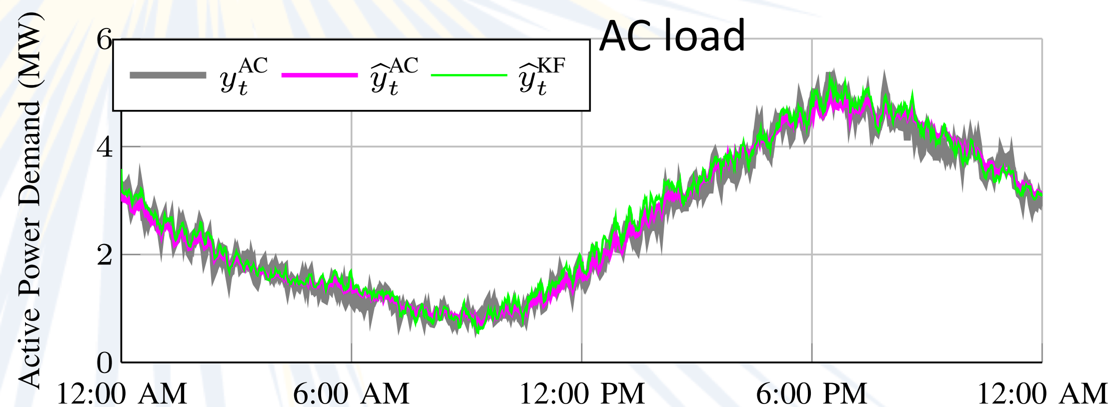
AC load RMSE

- DFS/DMD 151 kW
- BKF 177 kW
- AKF 214 kW

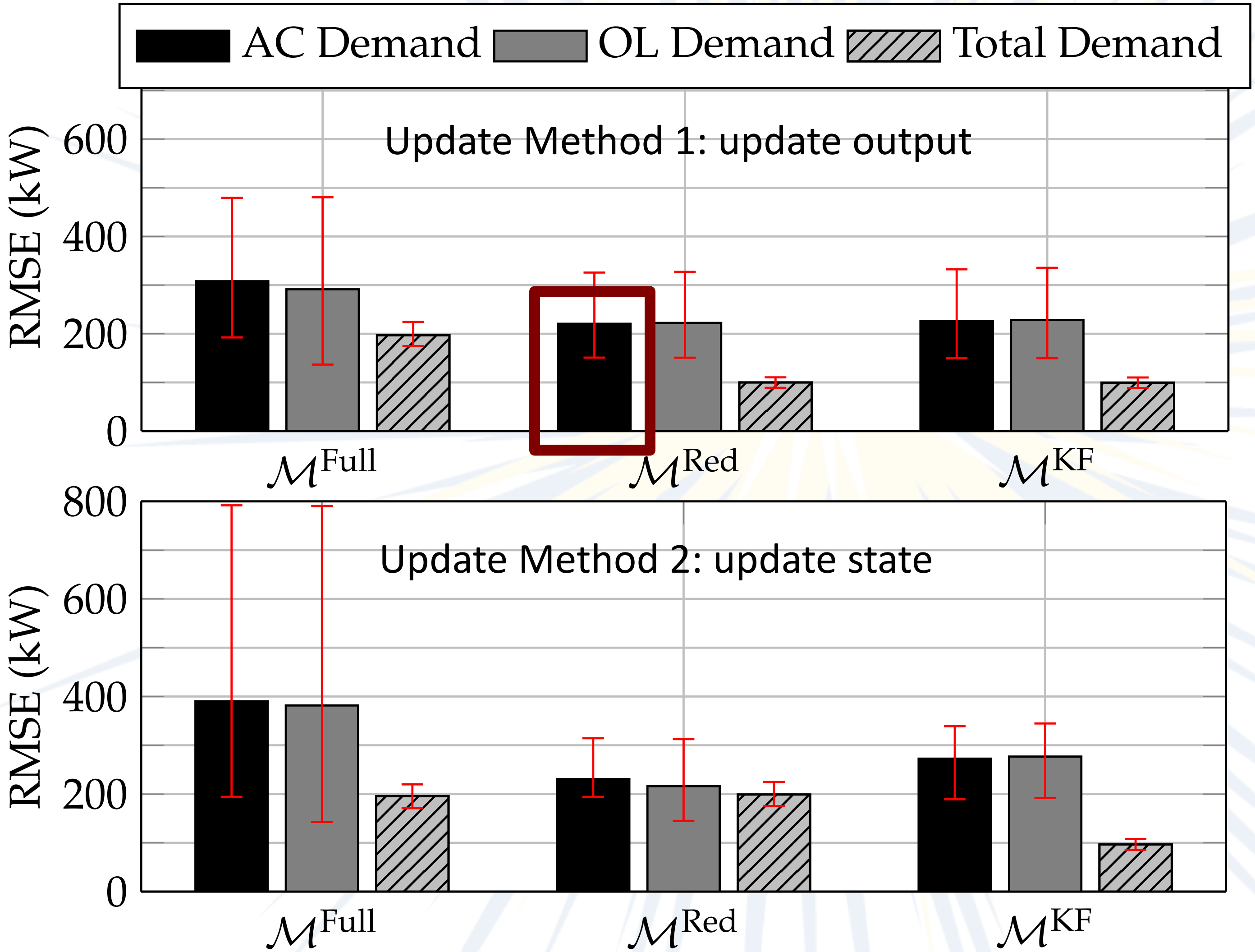








Comparison across model sets & update methods

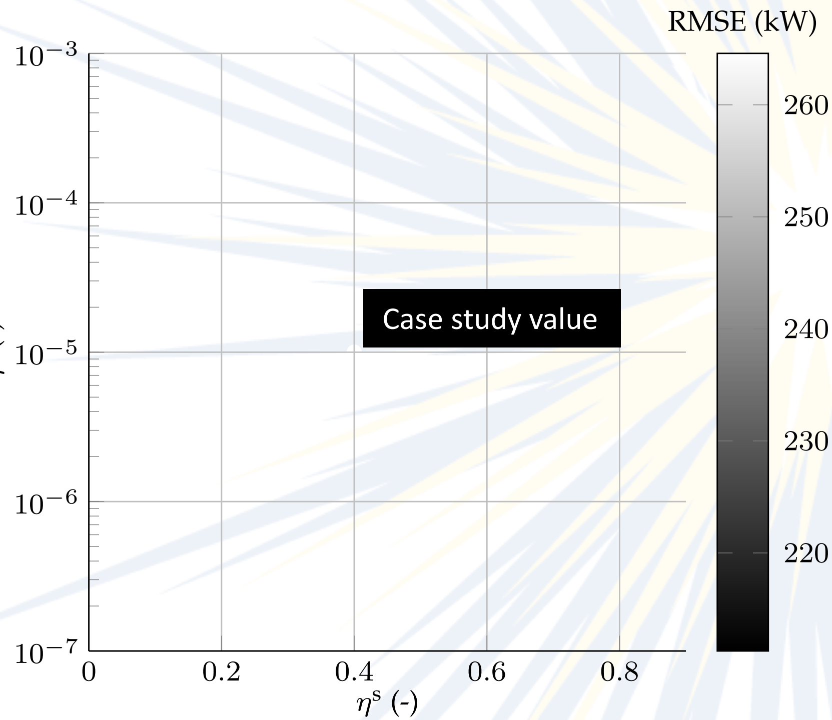


a posteriori BKF RMSE
- Mean 195 kW
- Min 148 kW
- Max 319 kW

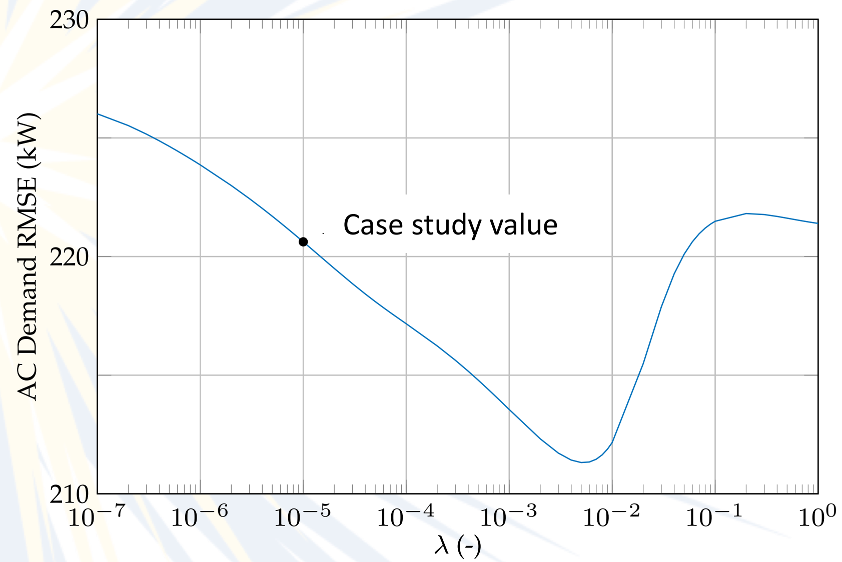
AKF RMSE
- Mean 259 kW
- Min 173 kW
- Max 358 kW

Sensitivity to parameter selections

faster transitions in weights →



larger adjustments to estimates based on prediction error →



estimate closer to the average of all models →

← a single model can dominate

Michigan Power & Energy Laboratory

Connections with Kalman Filtering

- In a Kalman Filter we assume the model form and use/choose the process and measurement noise covariance
- In DMD the model can take any form and we choose the loss and divergence functions. What should we pick?

Ledva, Du, Balzano, and Mathieu, "Disaggregating Load by Type from Distribution System Measurements in Real Time," In: Energy Markets and Responsive Grids, 2018.

Ledva, Balzano, and Mathieu, "Exploring Connections Between a Multiple Model Kalman Filter and Dynamic Fixed Share with Applications to Demand Response," IEEE CCTA 2018 (to appear).

Including Error Statistics in DMD

If we make the same assumption as we make for a Kalman filter (linear model, normally distributed errors, etc.), and we choose the following loss and divergence functions

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

$$D\left(x\|\widehat{x}_k\right) = \tfrac{1}{2}\left(x - \widehat{x}_k\right)^T \widehat{P}_k^{-1}\left(x - \widehat{x}_k\right)$$

where and are symmetric positive-definite covariance matrices corresponding to the model and measurement prediction error, then the DMD updates are identical to those of the Kalman filter.

Including error statistics

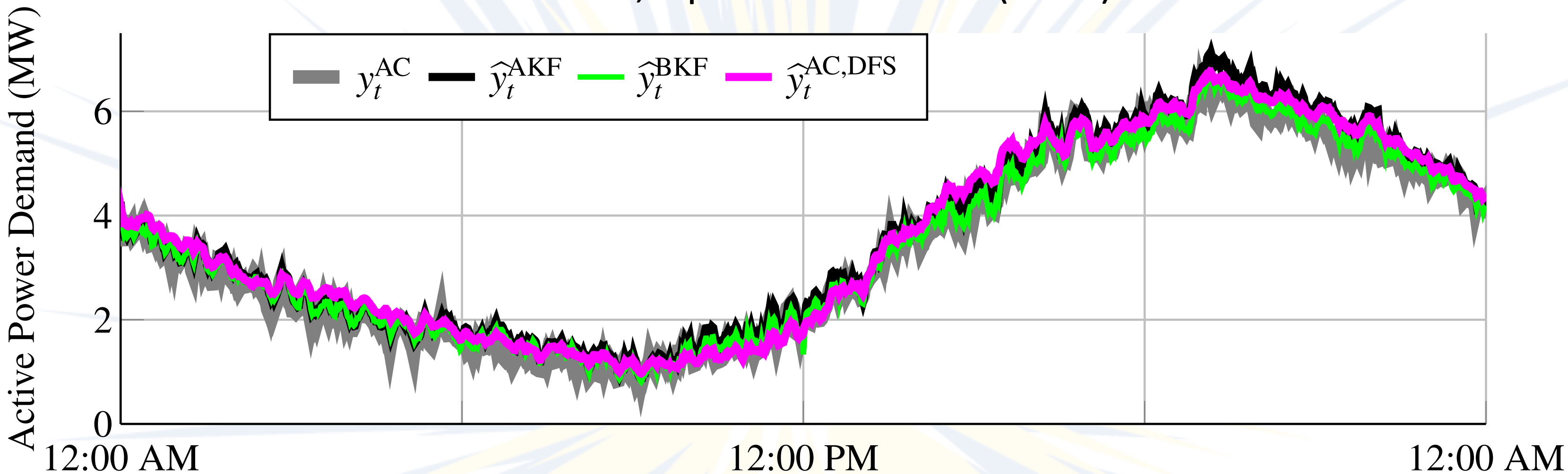
Use the Kalman Filter covariance update equations and compare three options for obtaining the process and measurement noise covariance

Method	Covariance	Total Demand			AC Demand			OL Demand		
		Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
UM 1	Identity	88.9	100.0	110.5	151.0	220.6	325.8	150.8	222.3	327.2
UM 1	Historical	98.4	114.8	123.2	155.0	252.2	371.5	150.2	250.1	372.5
UM 1	Real-Time	146.6	154.3	168.4	120.2	125.3	131.8	104.8	114.5	130.5
UM 2	Identity	175.4	199.1	224.8	194.2	230.9	314.5	145.0	216.2	312.7
UM 2	Historical	100.5	119.5	126.1	192.0	259.8	311.5	190.6	265.5	320.2
UM 2	Real-Time	120.8	125.2	129.1	104.0	116.5	140.1	96.6	109.4	131.9
BKF	Historical	-	-	-	148.4	195.3	318.9	-	-	-
AKF	Historical	-	-	-	173.1	259.4	357.5	-	-	-

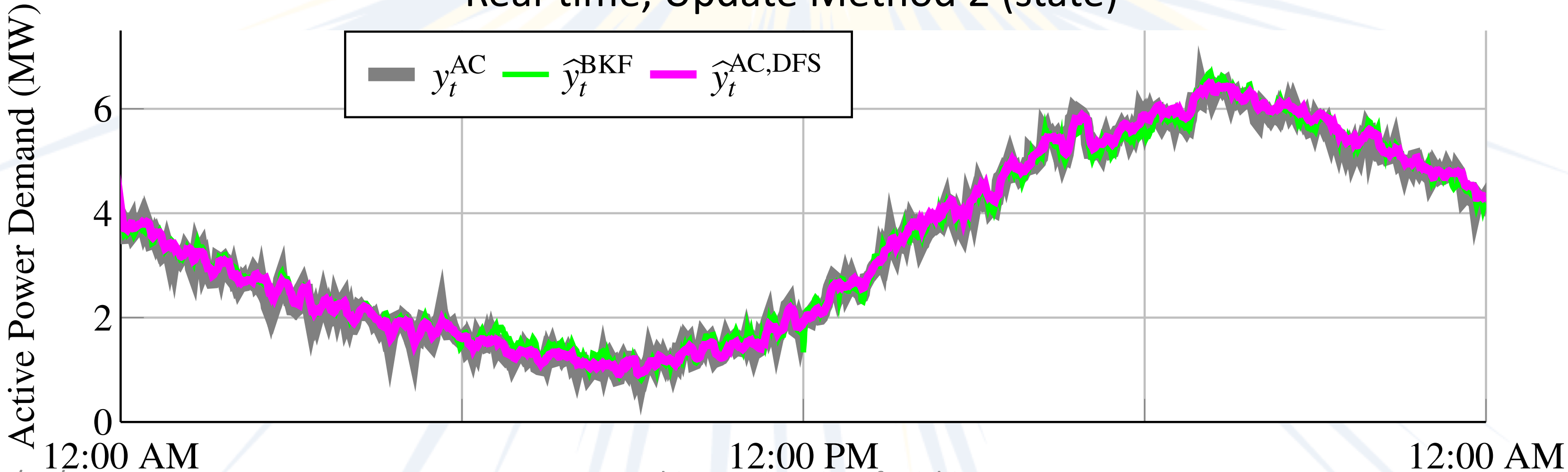
UM 1: Update Method 1 (output), UM 2: Update Method 2 (state)

Historical vs. Real-time

Historical, Update Method 2 (state)

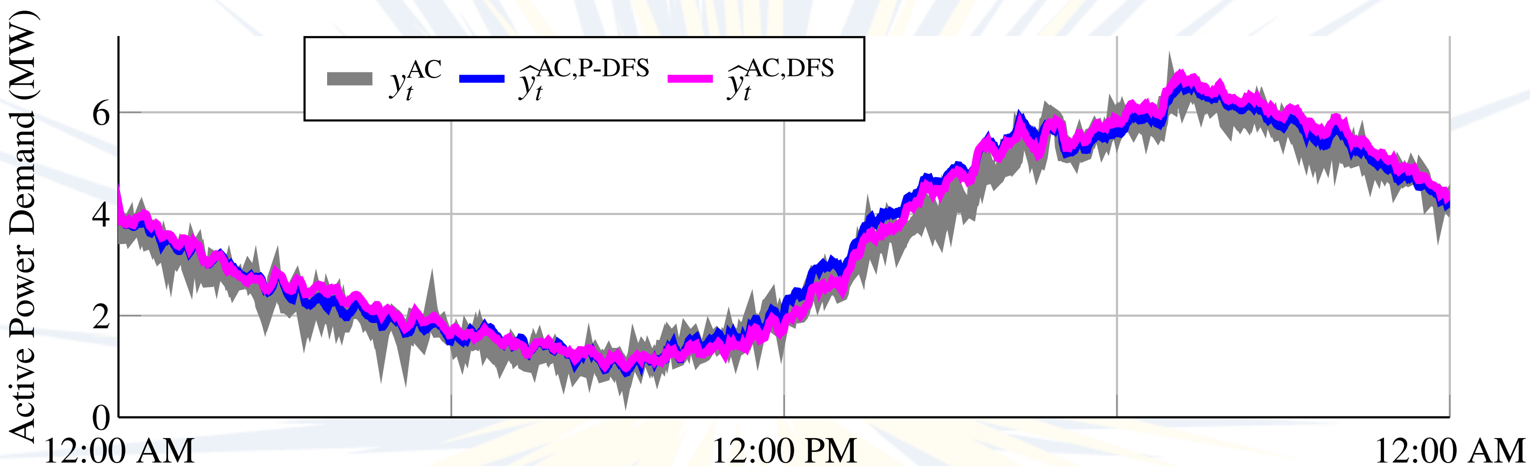


Real-time, Update Method 2 (state)

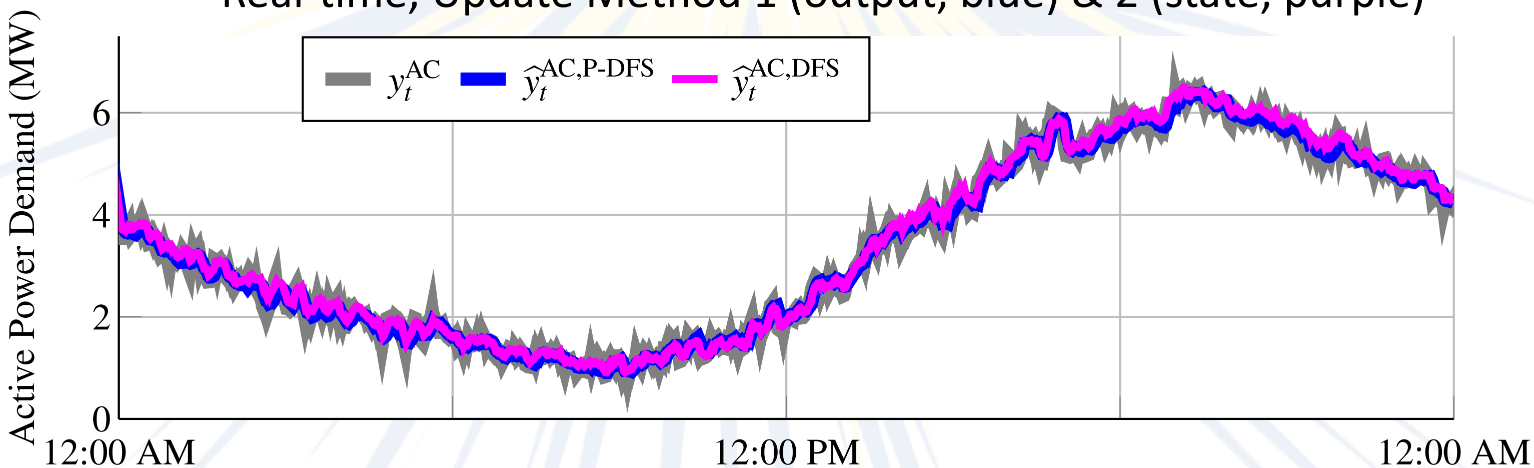


Historical vs. Real-time

Historical, Update Method 1 (output, blue) & 2 (state, purple)



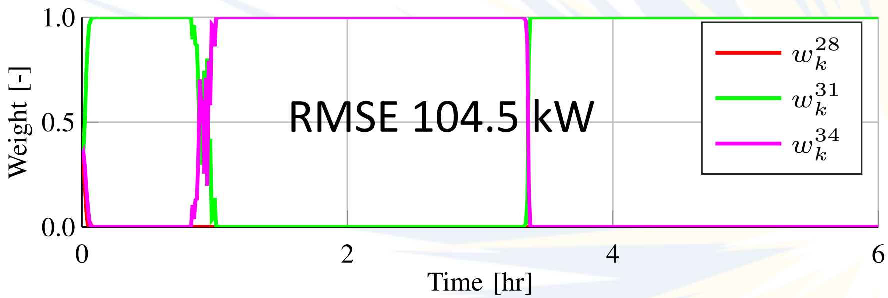
Real-time, Update Method 1 (output, blue) & 2 (state, purple)



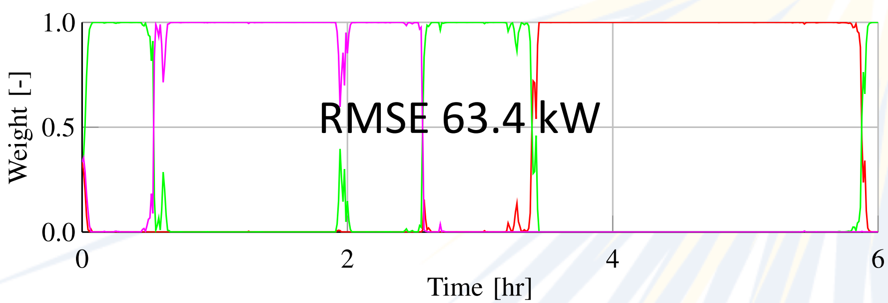
DFS and Multiple Model Kalman Filtering

- We can also construct DFS to produce identical updates to a Multiple Model Kalman Filter (MMKF).
- A number of heuristics have been developed for MMKFs; these can be adapted for DFS.
  - Setting a minimum weight
  - Exponential decay used to update weights
  - Sliding window used to update weights

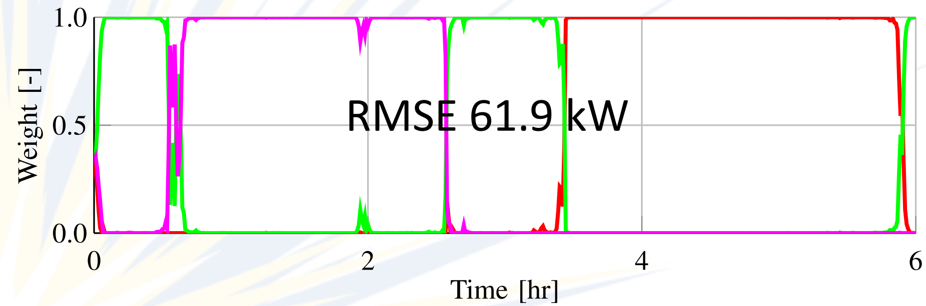
Example results



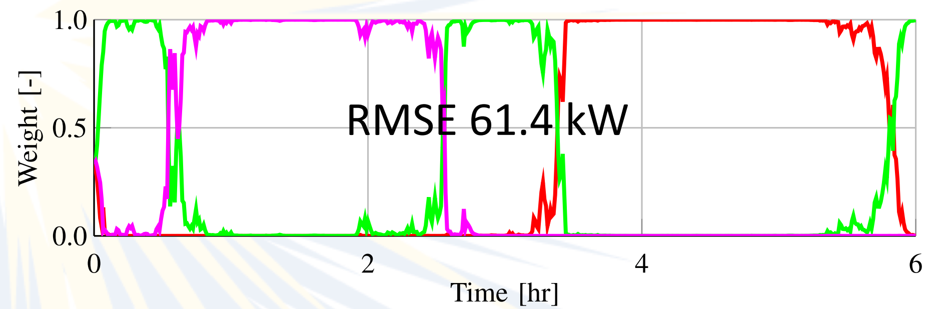
(a) MMKF Weights



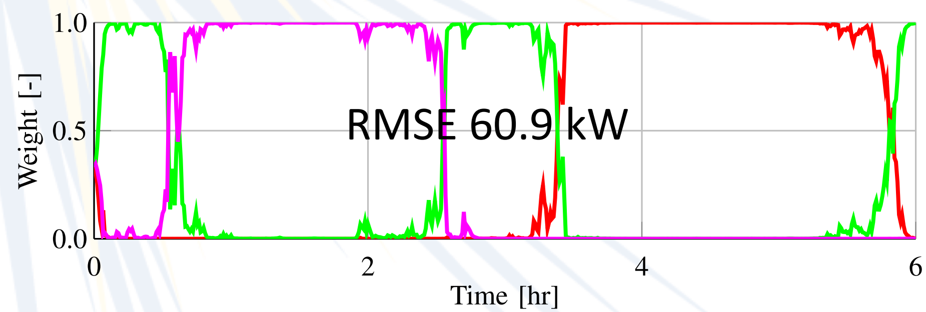
(b) MMKF-M Weights



(c) DFS-M Weights



(d) DFS-W Weights



(e) DFS-E Weights

Conclusions

- Dynamic Mirror Descent (DMD) and Dynamic Fixed Share (DFS) enables us to solve the feeder energy disaggregation problem leveraging dynamical models of arbitrary form
- Empirical results are often comparable to the a posteriori best Kalman filter (obtained from the same models)
- We can leverage ideas from Kalman filtering to inform our choice of DFS functions/parameters and heuristics

Backup

Algorithmic guarantees

- Regret: performance with respect to a comparator $\boldsymbol{\theta}_T$

$$R_T(\boldsymbol{\theta}_T) \triangleq \sum_{t=1}^{T} \ell_t(\widehat{\theta}_t) - \sum_{t=1}^{T} \ell_t(\theta_t).$$

- Often the comparator is the performance of a batch algorithm

- Hall and Willet derive bounds on the regret and show that for many classes of comparators regret scales sublinearly in T