Current Research Topics in Distributed Optimization of Smart Grids

Philipp Sauerteig
Technische Universität Ilmenau

joint work with
Karl Worthmann (TU Ilmenau)

funded by
the Federal Ministry of Education and Research

Federal Ministry
of Education
and Research

SIST Seminar 18170
ShanghaiTech University, 10 June 2019

Technische Universität Ilmenau

www.ima-ev.eu

Technische Universität Ilmenau

www.ima-ev.eu

www.nuernbergluftbild.de

Technische Universität Ilmenau

www.nuernbergluftbild.de

www.ima-ev.eu

Technische Universität Ilmenau

www.ima-ev.eu

www.nuernbergluftbild.de

Technische Universität Ilmenau

www.ima-ev.eu

www.nuernbergluftbild.de

	Shanghai
size [km^2]	6 340
population	26 000 000
students	35 000

Technische Universität Ilmenau

www.ima-ev.eu

www.nuernbergluftbild.de

	Shanghai	Ilmenau
size [km^2]	6 340	200
population	26 000 000	26 000
students	35 000	6 000

Technische Universität Ilmenau

www.ima-ev.eu

www.nuernbergluftbild.de

	Shanghai	Ilmenau	Thuringia
size [km^2]	6 340	200	16 000
population	26 000 000	26 000	2 150 000
students	35 000	6 000	33 000

KONSENS: Konsistente Optimierung uNd Stabilisierung Elektrischer NetzwerkSysteme

Federal Ministry of Education and Research

Mathematics for Innovations as Contribution to Energy Transition

KONSENS: Konsistente Optimierung uNd Stabilisierung Elektrischer NetzwerkSysteme

Federal Ministry of Education and Research

Mathematics for Innovations as Contribution to Energy Transition

MAX PLANCK INSTITUTE FOR DYNAMICS OF COMPLEX TECHNICAL SYSTEMS MAGDEBURG

- Model Order Reduction and Flexibility Information

TECHNISCHE UNIVERSITÄT CHEMNITZ

- Robust Model Analysis and Control
- Mixed-Integer and Semi-Definite Power Flow Optimization

KON SENS

TECHNISCHE UNIVERSITÄT ILMENAU

- Distributed Optimization and Control of Microgrids

KONSENS: Konsistente Optimierung uNd Stabilisierung Elektrischer NetzwerkSysteme

Federal Ministry of Education and Research

Mathematics for Innovations as Contribution to Energy Transition

MAX PLANCK INSTITUTE FOR DYNAMICS OF COMPLEX TECHNICAL SYSTEMS MAGDEBURG

- Model Order Reduction and Flexibility Information

TECHNISCHE UNIVERSITÄT CHEMNITZ

- Robust Model Analysis and Control
- Mixed-Integer and Semi-Definite Power Flow Optimization

TenneT Taking power further

KON SENS

energy saxony

TECHNISCHE UNIVERSITÄT ILMENAU

- Distributed Optimization and Control of Microgrids

enso NETZ

VENIOS® Uniting IT and Energy

Outline

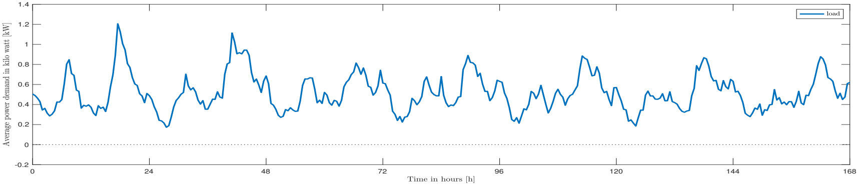
- Optimal Control of Distributed Energy Storage Devices
  - Motivation
  - Modelling Residential Energy Systems
  - Distributed Optimization via ADMM

Outline

- Optimal Control of Distributed Energy Storage Devices
    - Motivation
    - Modelling Residential Energy Systems
    - Distributed Optimization via ADMM
- Current Research
    - Distributed Optimization via ALADIN
    - Multiobjective Optimization
    - Coupled Microgrids
    - Surrogates

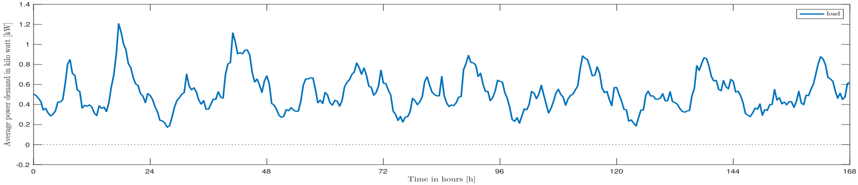
Motivation

Aggregated power demand profile (1 week)



Motivation

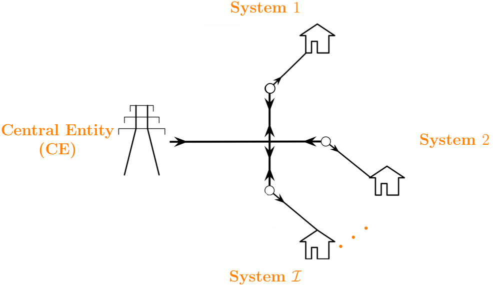
Aggregated power demand profile (1 week)



Problem

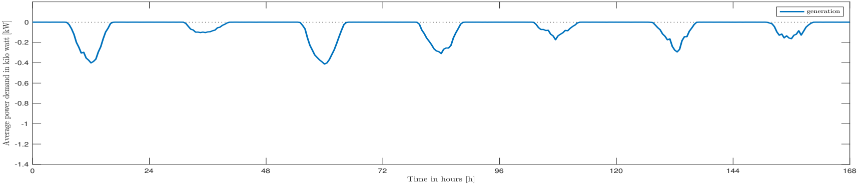
- volatile load

Smart grid



Motivation

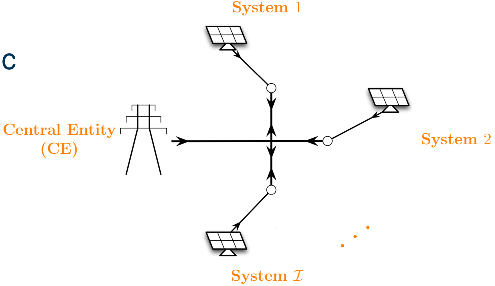
Aggregated power demand profile (1 week)



Problem

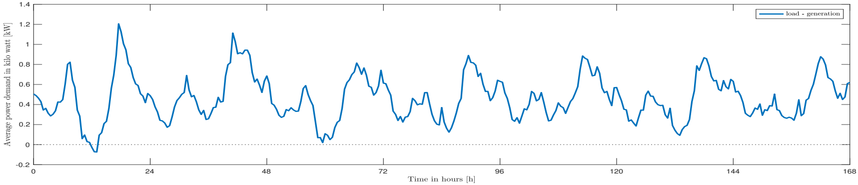
- volatile load
- generation via photovoltaic

Smart grid



Motivation

Aggregated power demand profile (1 week)

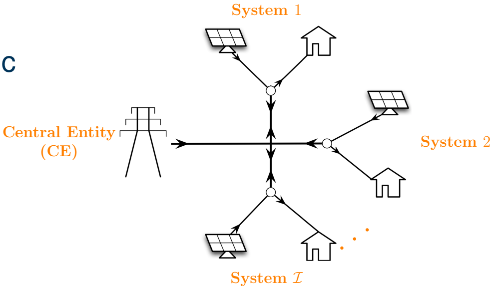


Problem

- volatile load
- generation via photovoltaic

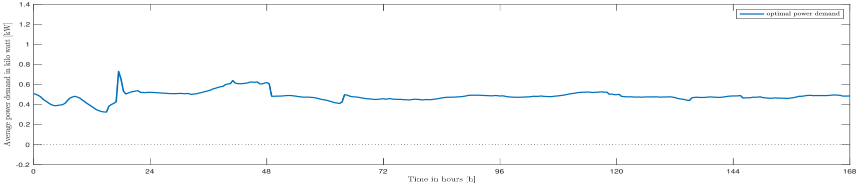
⇝ volatile power demand

Smart grid



Motivation

Aggregated power demand profile (1 week)



Problem

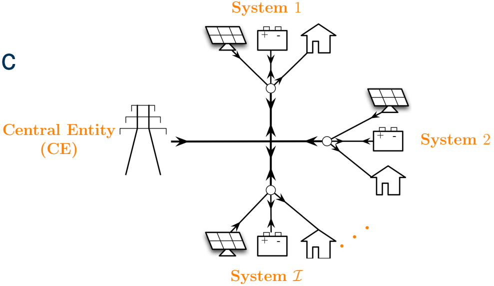
- volatile load
- generation via photovoltaic

⇝ volatile power demand

Remedy: exploit flexibilities

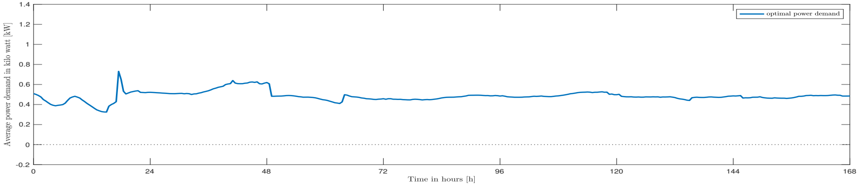
- storage devices

Smart grid



Motivation

Aggregated power demand profile (1 week)



Problem

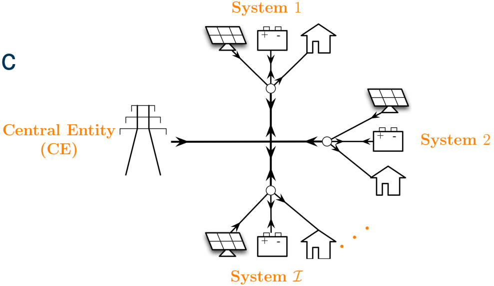
- volatile load
- generation via photovoltaic

⇝ volatile power demand

Remedy: exploit flexibilities

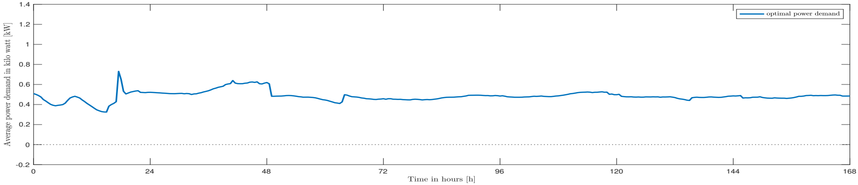
- storage devices
- energy exchange

Smart grid



Motivation

Aggregated power demand profile (1 week)



Problem

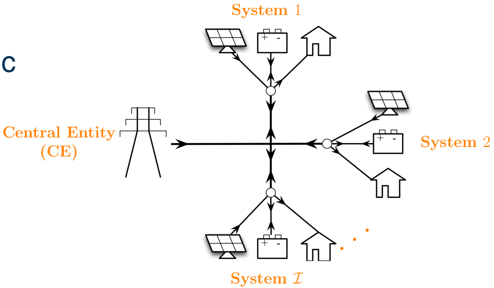
- volatile load
- generation via photovoltaic

⇝ volatile power demand

Remedy: exploit flexibilities

- storage devices
- energy exchange
- controllable loads, ...

Smart grid



Dynamics & Constraints

Given: $\mathcal{I} \in \mathbb{N}$ subsystems (smart homes)

System equation of subsystem $i \in [1 : \mathcal{I}] := \{1, 2, \ldots, \mathcal{I}\}$ at time instants $n \in [k : k + N - 1]$, $k \in \mathbb{N}_0$, $x_i(k) = \hat{x}_i$:

$$x_i(n+1)$$
$$z_i(n)$$

Notation

- State of charge $x_i(n)$
- Power demand $z_i(n) \in \mathbb{R}$

Constraints: For all

$i \in [1 : \mathcal{I}]$ and all $n \in \mathbb{N}_0$

$$0 \leq \qquad x_i(n) \qquad \leq C_i$$

Dynamics & Constraints

Given: $\mathcal{I} \in \mathbb{N}$ subsystems (smart homes)

System equation of subsystem $i \in [1:\mathcal{I}] := \{1, 2, \ldots, \mathcal{I}\}$ at time instants $n \in [k : k+N-1]$, $k \in \mathbb{N}_0$, $x_i(k) = \hat{x}_i$:

$$x_i(n+1) = x_i(n) + T(\quad u_i^+(n) + u_i^-(n))$$
$$z_i(n)$$

Notation

- State of charge $x_i(n)$
- Power demand $z_i(n) \in \mathbb{R}$
- (Dis-)Charging rate $u_i(n)$
- Time interval length $T > 0$

Constraints: For all $i \in [1:\mathcal{I}]$ and all $n \in \mathbb{N}_0$

$$\begin{aligned} 0 \leq & \quad x_i(n) & \leq C_i \\ \underline{u}_i \leq & \quad u_i^-(n) & \leq 0 \\ 0 \leq & \quad u_i^+(n) & \leq \bar{u}_i \\ 0 \leq & \quad \frac{u_i^-(n)}{\underline{u}_i} + \frac{u_i^+(n)}{\bar{u}_i} & \leq 1 \end{aligned}$$

Dynamics & Constraints

Given: $\mathcal{I} \in \mathbb{N}$ subsystems (smart homes)

System equation of subsystem $i \in [1 : \mathcal{I}] := \{1, 2, \dots, \mathcal{I}\}$ at time instants $n \in [k : k + N - 1]$, $k \in \mathbb{N}_0$, $x_i(k) = \hat{x}_i$:

$$
\begin{aligned}
x_i(n+1) &= x_i(n) + T\left(u_i^+(n) + u_i^-(n)\right) \\
z_i(n) &= w_i(n) + u_i^+(n) + u_i^-(n)
\end{aligned}
$$

Notation

- State of charge $x_i(n)$
- Power demand $z_i(n) \in \mathbb{R}$
- Net consumption $w_i(n) = \ell_i(n) - g_i(n) \in \mathbb{R}$
- (Dis-)Charging rate $u_i(n)$
- Time interval length $T > 0$

Constraints: For all $i \in [1 : \mathcal{I}]$ and all $n \in \mathbb{N}_0$

$$
\begin{aligned}
0 \leq\ & x_i(n) && \leq C_i \\
\underline{u}_i \leq\ & u_i^-(n) && \leq 0 \\
0 \leq\ & u_i^+(n) && \leq \bar{u}_i \\
0 \leq\ & \frac{u_i^-(n)}{\underline{u}_i} + \frac{u_i^+(n)}{\bar{u}_i} && \leq 1
\end{aligned}
$$

Philipp Sauerteig
Institute for Mathematics

Dynamics & Constraints

Given: $\mathcal{I} \in \mathbb{N}$ subsystems (smart homes)

System equation of subsystem $i \in [1:\mathcal{I}] := \{1, 2, \ldots, \mathcal{I}\}$ at time instants $n \in [k : k+N-1]$, $k \in \mathbb{N}_0$, $x_i(k) = \hat{x}_i$:

$$
\begin{aligned}
x_i(n+1) &= \alpha_i x_i(n) + T(\beta_i u_i^+(n) + u_i^-(n)) \\
z_i(n) &= w_i(n) + u_i^+(n) + \gamma_i u_i^-(n)
\end{aligned}
$$

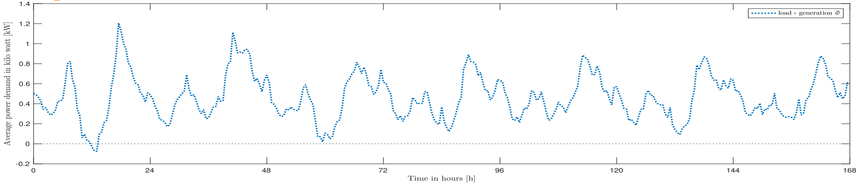
Notation

- State of charge $x_i(n)$
- Power demand $z_i(n) \in \mathbb{R}$
- Net consumption $w_i(n) = \ell_i(n) - g_i(n) \in \mathbb{R}$
- (Dis-)Charging rate $u_i(n)$
- Time interval length $T > 0$
- Efficiencies $\alpha_i, \beta_i, \gamma_i \in (0, 1]$

Constraints: For all $i \in [1:\mathcal{I}]$ and all $n \in \mathbb{N}_0$

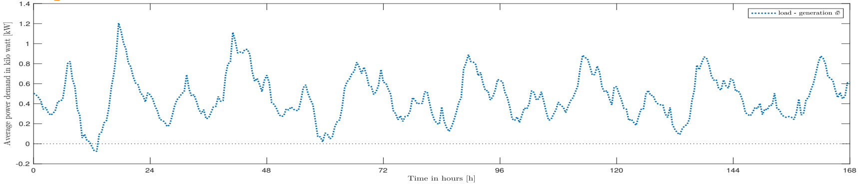
$$
\begin{aligned}
0 \leq &\quad x_i(n) &\leq C_i \\
\underline{u}_i \leq &\quad u_i^-(n) &\leq 0 \\
0 \leq &\quad u_i^+(n) &\leq \bar{u}_i \\
0 \leq &\quad \frac{u_i^-(n)}{\underline{u}_i} + \frac{u_i^+(n)}{\bar{u}_i} &\leq 1
\end{aligned}
$$

Objective



Goal: Flatten aggregated power demand profile

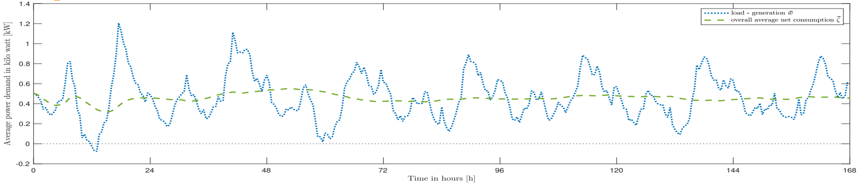
Objective



Goal: Flatten aggregated power demand profile

Idea: Trace some steady reference trajectory

Objective



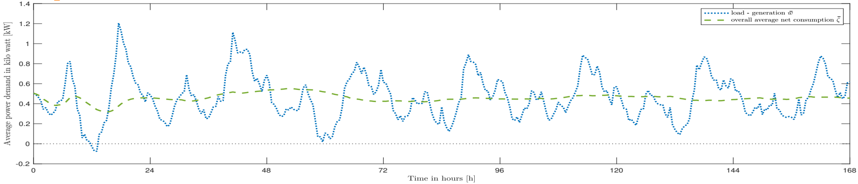
Goal: Flatten aggregated power demand profile

Idea: Trace some steady reference trajectory, e.g. overall average net consumption

$$\bar{\zeta}(n) = \frac{1}{\mathcal{I} \cdot \min\{N_1, n+1\}} \sum_{j=n-\min\{n, N_1-1\}}^{n} \sum_{i=1}^{\mathcal{I}} w_i(j)$$

for some $N_1 \in \mathbb{N}_{\geq 2}$

Objective



Goal: Flatten aggregated power demand profile

Idea: Trace some steady reference trajectory, e.g. overall average net consumption

$$\bar{\zeta}(n) = \frac{1}{\mathcal{I} \cdot \min\{N_1, n+1\}} \sum_{j=n-\min\{n, N_1-1\}}^{n} \sum_{i=1}^{\mathcal{I}} w_i(j)$$

for some $N_1 \in \mathbb{N}_{\geq 2}$

Problem: Future net consumption w_i unknown

$\rightsquigarrow$ prediction of $w_i(j)$, $j \in [k : k + N_2 - 1]$
($N_1 = N_2 = N$ prediction horizon)

Model Predictive Control

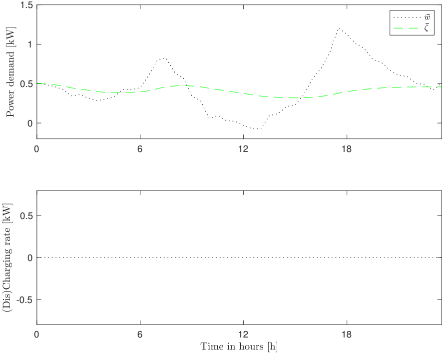
Open loop

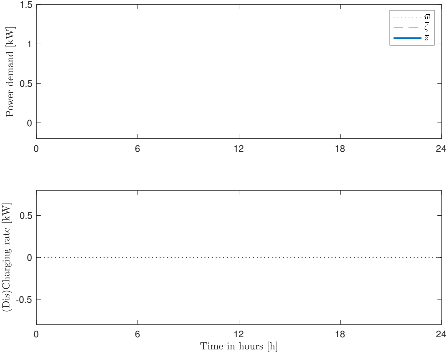
Closed loop

Model Predictive Control

Open loop

Closed loop

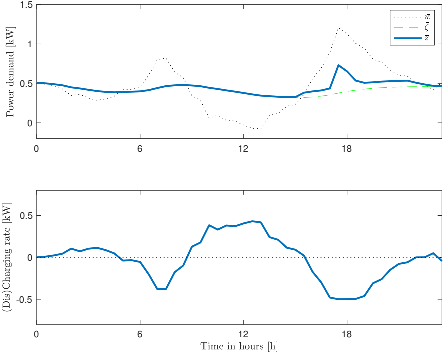




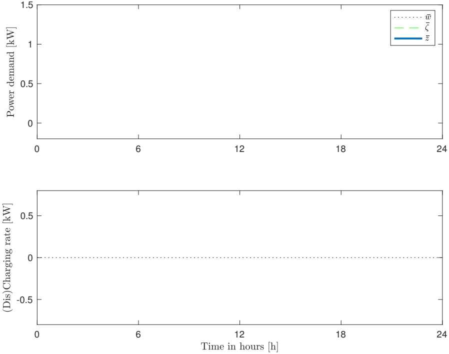
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.

Model Predictive Control

Open loop



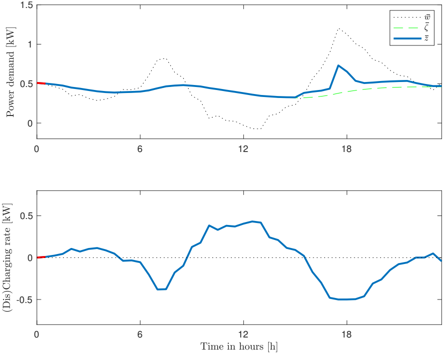
Closed loop



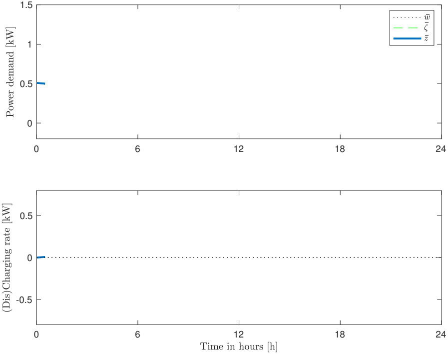
2. Compute optimal input $u^* = (u^*(k), \dots, u^*(k+N-1))^\top$.

Model Predictive Control

Open loop



Closed loop

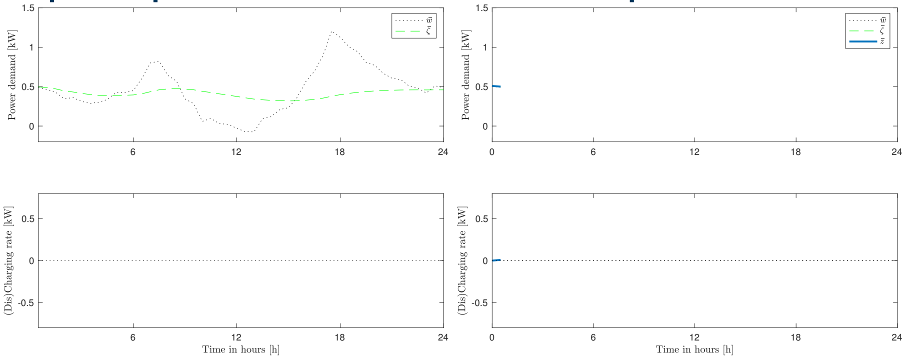


3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop

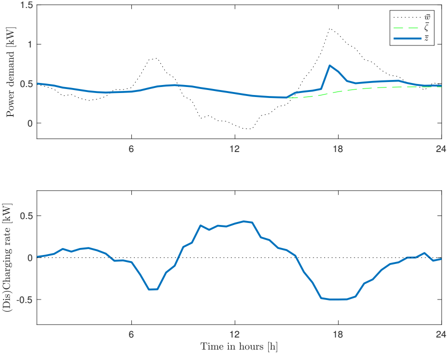
Closed loop



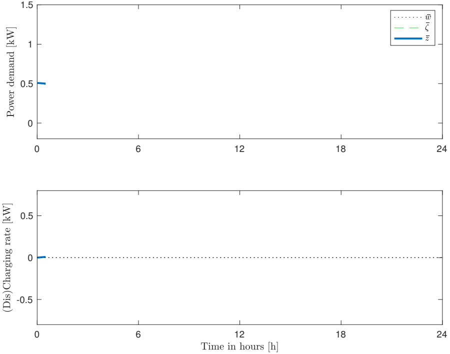
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.

Model Predictive Control

Open loop



Closed loop

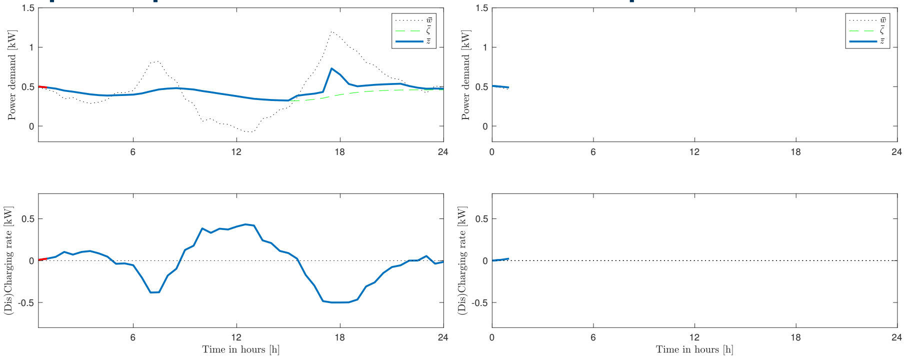


1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.

Model Predictive Control

Open loop

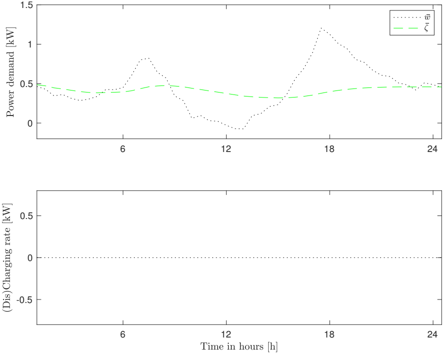
Closed loop



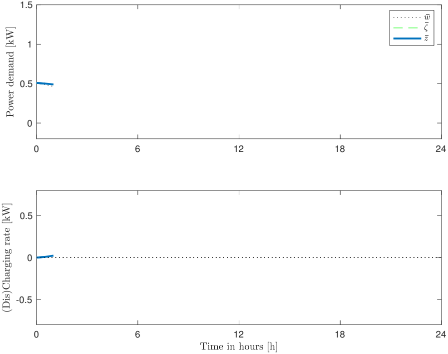
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



Closed loop

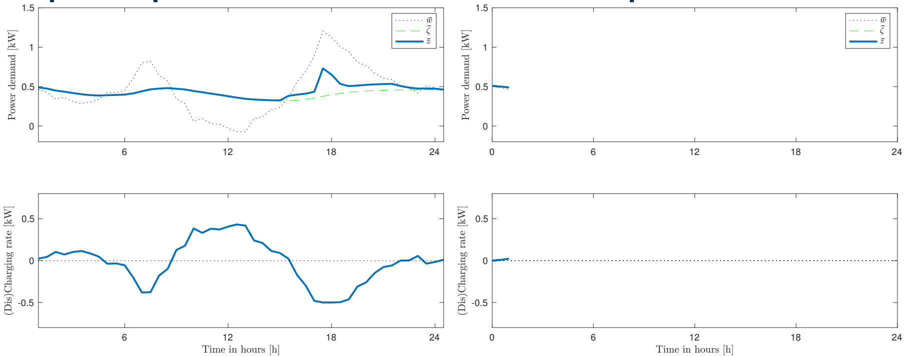


1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop

Closed loop

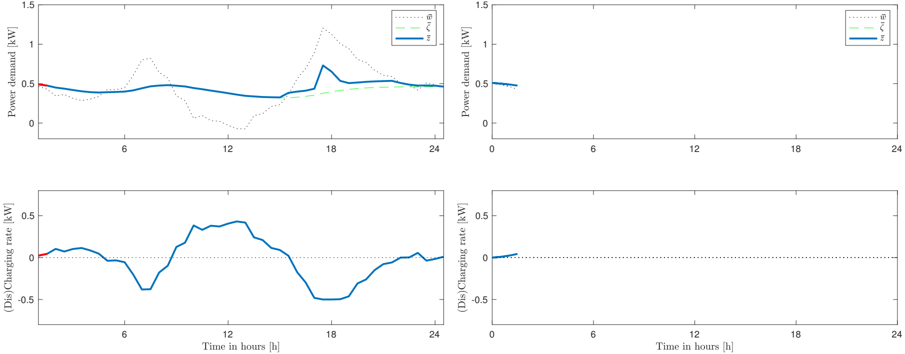


1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop

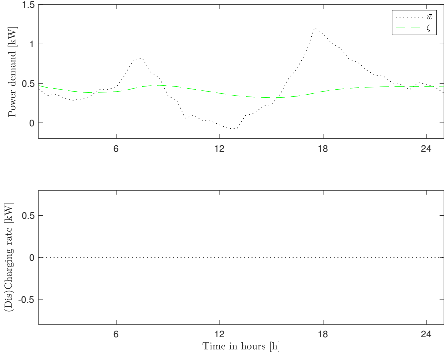
Closed loop



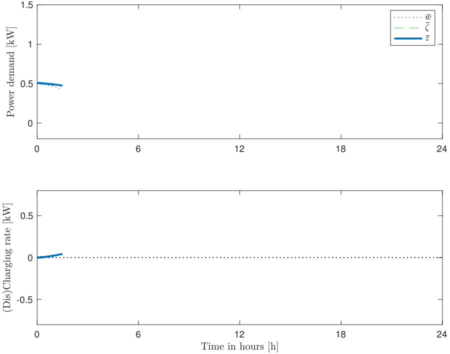
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



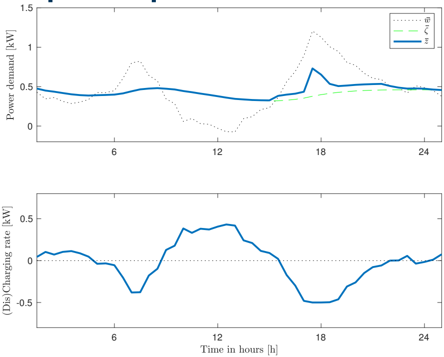
Closed loop



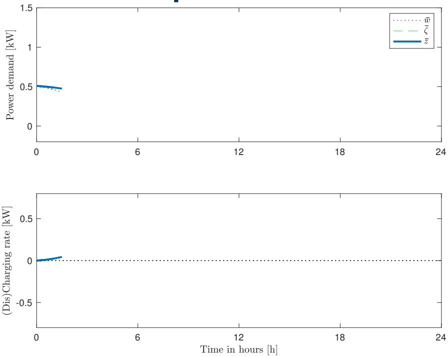
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



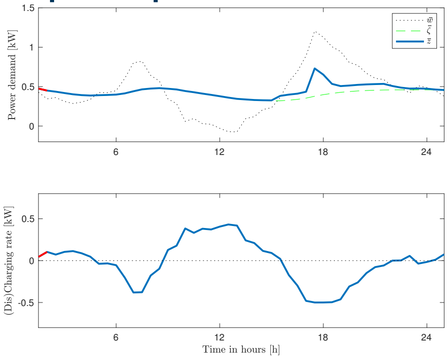
Closed loop



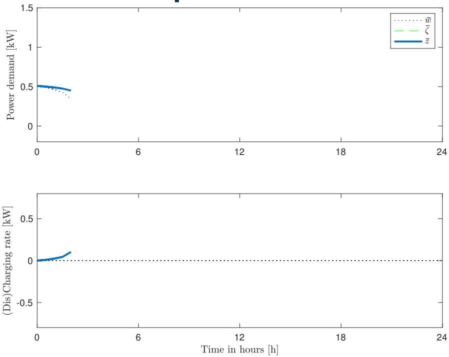
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



Closed loop

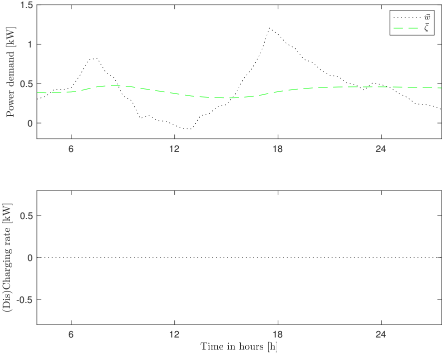


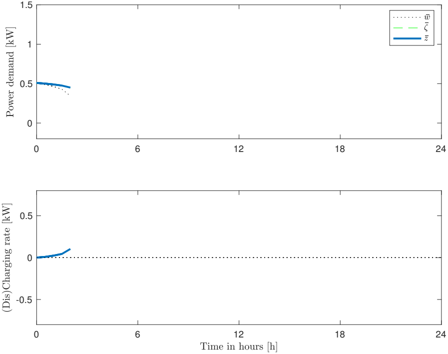
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop

Closed loop

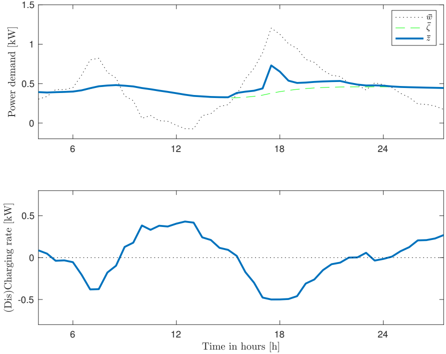




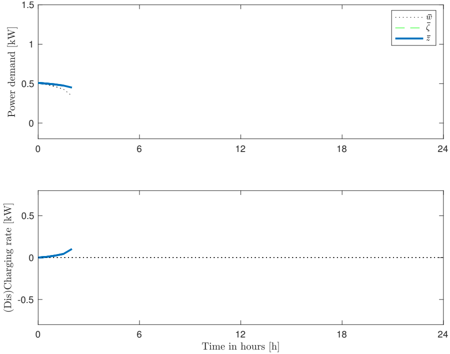
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



Closed loop

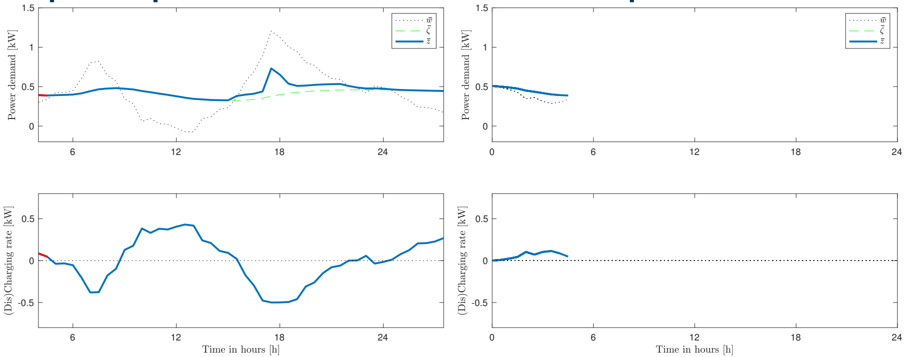


1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop

Closed loop

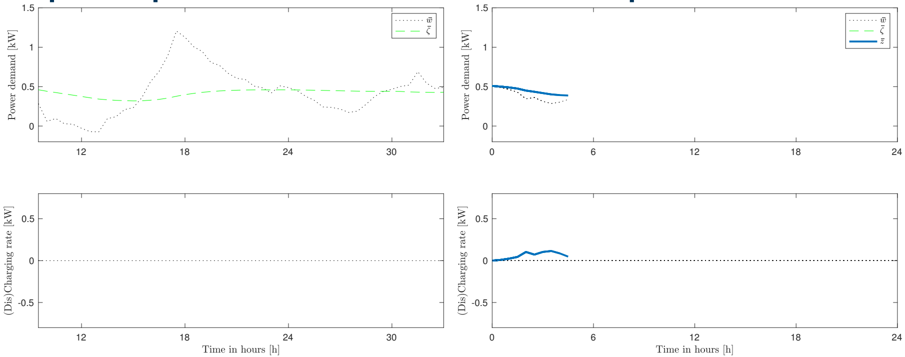


1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop

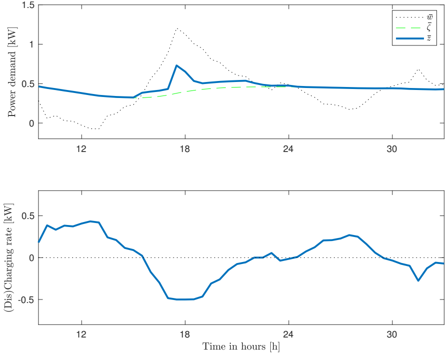
Closed loop



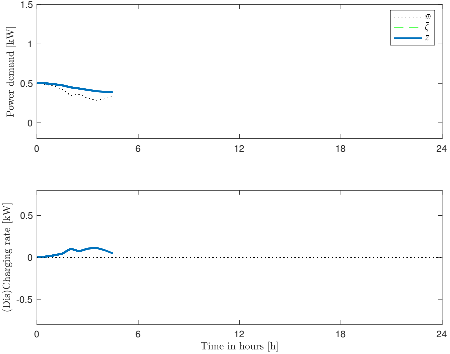
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



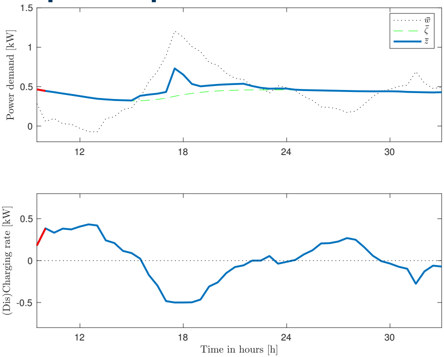
Closed loop



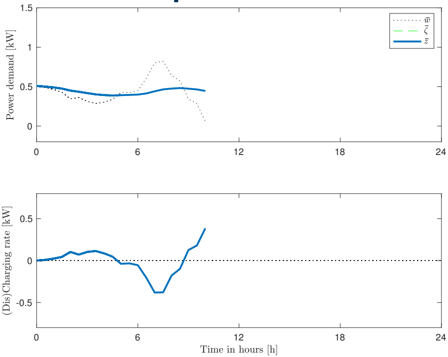
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



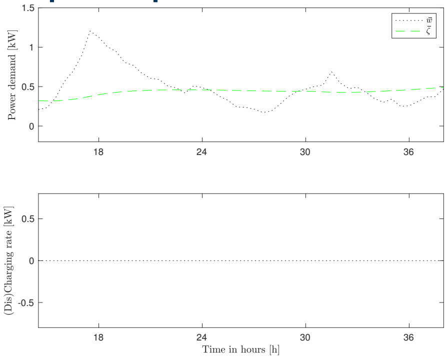
Closed loop



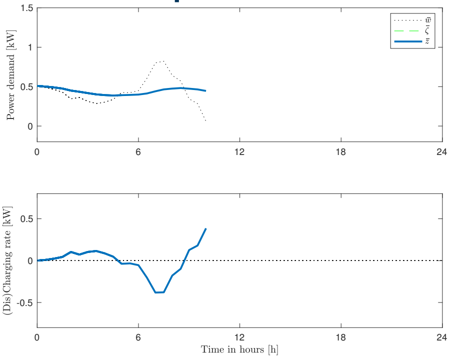
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



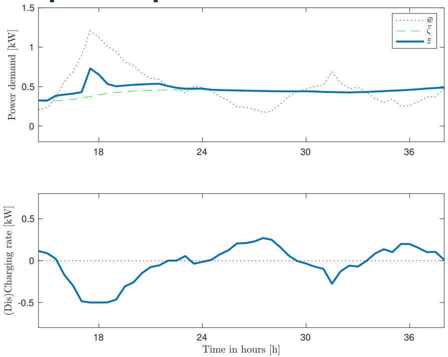
Closed loop



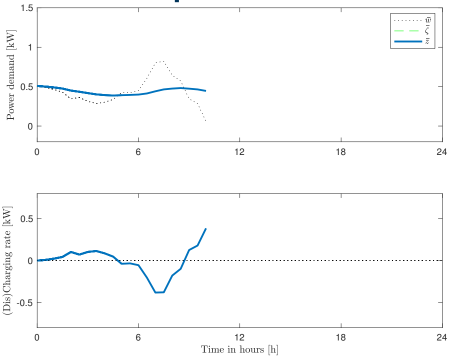
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



Closed loop

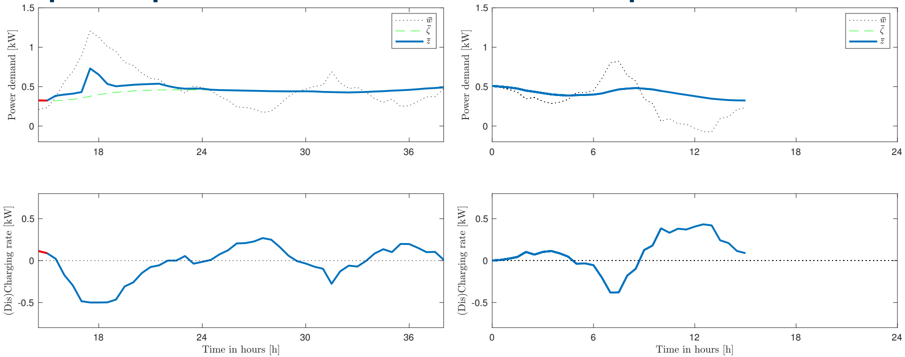


1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop

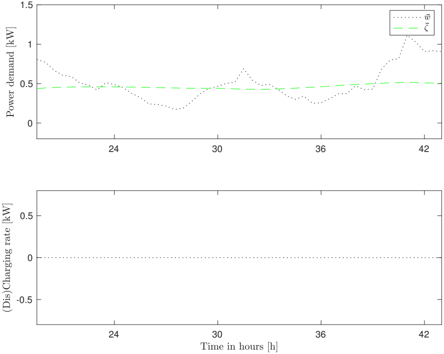
Closed loop



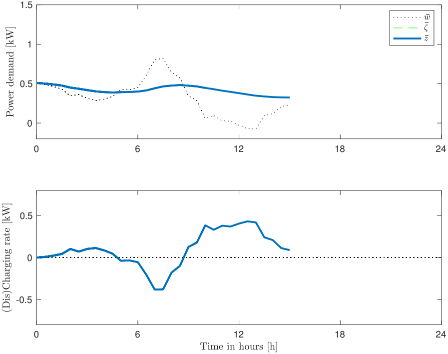
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



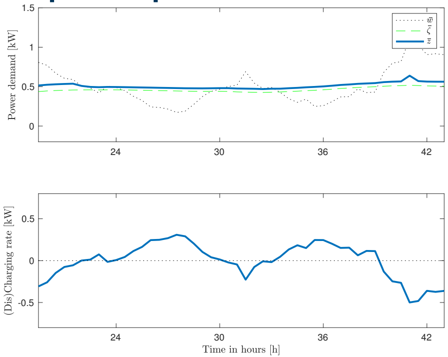
Closed loop



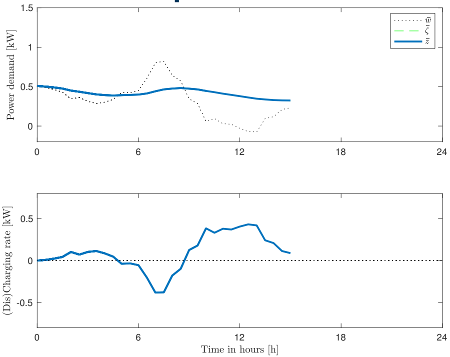
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



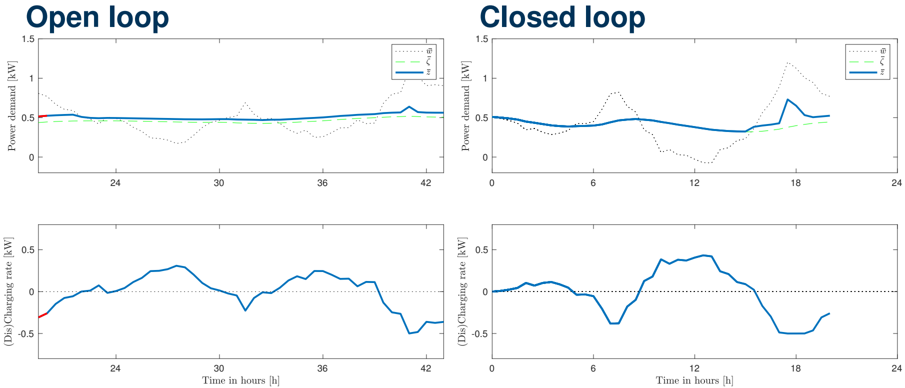
Closed loop



1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

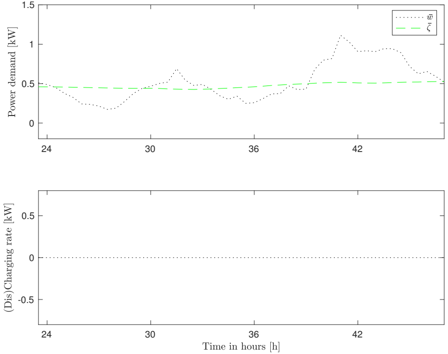
Open loop / Closed loop



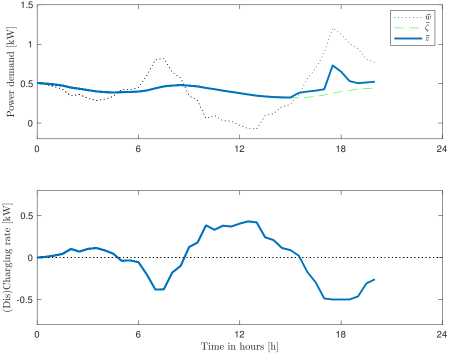
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



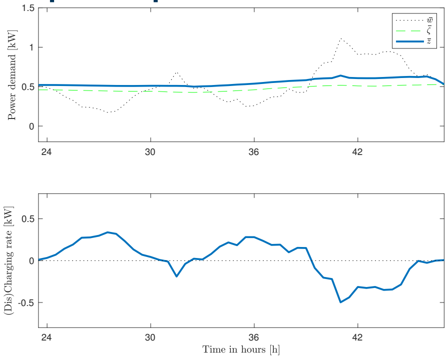
Closed loop



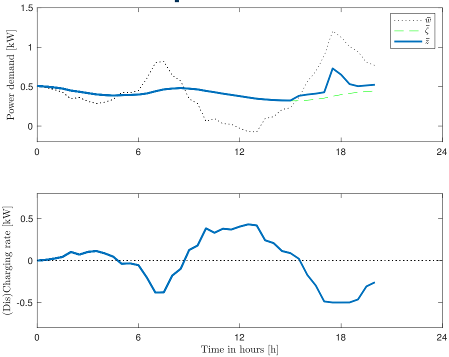
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



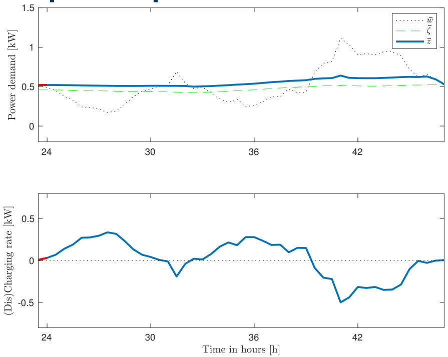
Closed loop



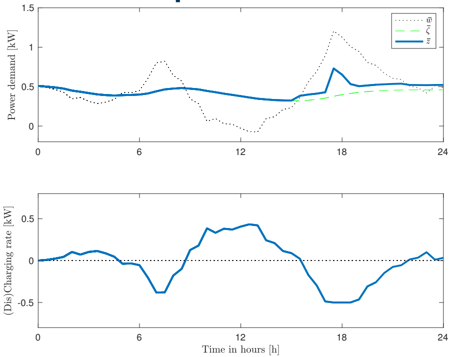
1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Model Predictive Control

Open loop



Closed loop



1. Measure current SoC $\hat{x} = x(k)$ and predict (exogenous) input $w = (w(k), \ldots, w(k+N-1))^\top$.
2. Compute optimal input $u^* = (u^*(k), \ldots, u^*(k+N-1))^\top$.
3. Implement $u^*(k)$ and increment k.

Optimization Problem

Centralized formulation

$$\min_{u=(u^+,u^-)} \quad \sum_{n=k}^{k+N-1} \left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \left[w_i(n) + u_i^+(n) + \gamma_i u_i^-(n) \right] - \bar{\zeta}(n) \right)^2$$

$$\text{s.t.} \quad \text{system dynamics and constraints}$$

Optimization Problem

Centralized formulation

$$\min_{u=(u^+,u^-)} \quad \sum_{n=k}^{k+N-1} \left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \overbrace{\left[w_i(n) + u_i^+(n) + \gamma_i u_i^-(n) \right]}^{=z_i(n)} - \bar{\zeta}(n) \right)^2$$

$$\text{s.t.} \quad \text{system dynamics and constraints}$$

Optimization Problem

Centralized formulation

$$\min_{u=(u^+,u^-)} \sum_{n=k}^{k+N-1} \left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \overbrace{\left[w_i(n) + u_i^+(n) + \gamma_i u_i^-(n) \right]}^{=z_i(n)} - \bar{\zeta}(n) \right)^2$$

$$\text{s.t.} \quad \text{system dynamics and constraints}$$

Distributed formulation

$$\min_{(z,\bar{a})} \sum_{n=k}^{k+N-1} \left(\bar{a}(n) - \bar{\zeta}(n) \right)^2$$

$$\text{s.t.} \quad \text{system dynamics and constraints}$$

$$z_i - a_i = 0, i \in [1:\mathcal{I}], \quad \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} a_i - \bar{a} = 0$$

Optimization Problem

Centralized formulation

$$\min_{u=(u^+,u^-)} \sum_{n=k}^{k+N-1} \left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \overbrace{\left[w_i(n) + u_i^+(n) + \gamma_i u_i^-(n) \right]}^{=z_i(n)} - \bar{\zeta}(n) \right)^2$$

s.t. system dynamics and constraints

Distributed formulation

$$\min_{(z,\bar{a})} \sum_{n=k}^{k+N-1} \left(\bar{a}(n) - \bar{\zeta}(n)\right)^2 = \|\bar{a} - \bar{\zeta}\|_2^2 =: g(\bar{a})$$

s.t. system dynamics and constraints

$$z_i - a_i = 0, i \in [1 : \mathcal{I}], \quad \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} a_i - \bar{a} = 0$$

Alternating Direction Method of Multipliers

[Boyd et al., 2011]

Augmented Lagrangian: dual λ and $\rho > 0$

$$\mathcal{L}_\rho(z, a, \lambda) = g(\bar{a}) + \sum_{i=1}^{\mathcal{I}} \left(\lambda_i^\top (z_i - a_i) + \frac{\rho}{2} \|z_i - a_i\|_2^2 \right)$$

Alternating Direction Method of Multipliers

[Boyd et al., 2011]

Augmented Lagrangian: dual λ and $\rho > 0$

$$\mathcal{L}_\rho(z, a, \lambda) = g(\bar{a}) + \sum_{i=1}^{\mathcal{I}} \left(\lambda_i^\top (z_i - a_i) + \frac{\rho}{2} \|z_i - a_i\|_2^2 \right)$$

Basic ADMM Scheme

1. Parallel step
   $z_i^{\ell+1} \;=\; \arg\min_{z_i \in \mathbb{D}_i} \; z_i^\top \lambda_i^\ell + \frac{\rho}{2} \left\| z_i - a_i^\ell \right\|_2^2$

Alternating Direction Method of Multipliers

[Boyd et al., 2011]

Augmented Lagrangian: dual λ and $\rho > 0$

$$\mathcal{L}_\rho(z, a, \lambda) = g(\bar{a}) + \sum_{i=1}^{\mathcal{I}} \left(\lambda_i^\top (z_i - a_i) + \frac{\rho}{2} \|z_i - a_i\|_2^2 \right)$$

Basic ADMM Scheme

1. Parallel step
   $z_i^{\ell+1} \;=\; \arg\min_{z_i \in \mathbb{D}_i} \; z_i^\top \lambda_i^\ell + \frac{\rho}{2} \left\| z_i - a_i^\ell \right\|_2^2$
2. Central step
   $a^{\ell+1} \;=\; \arg\min_a \; g(\bar{a}) - \sum_{i=1}^{\mathcal{I}} a_i^\top \lambda_i^\ell + \frac{\rho}{2} \left\| z_i^{\ell+1} - a_i \right\|_2^2$

Alternating Direction Method of Multipliers

[Boyd et al., 2011]

Augmented Lagrangian: dual λ and $\rho > 0$

$$\mathcal{L}_\rho(z, a, \lambda) = g(\bar{a}) + \sum_{i=1}^{\mathcal{I}} \left(\lambda_i^\top (z_i - a_i) + \frac{\rho}{2} \|z_i - a_i\|_2^2 \right)$$

Basic ADMM Scheme

1. Parallel step
   $z_i^{\ell+1} \;=\; \arg\min_{z_i \in \mathbb{D}_i} \; z_i^\top \lambda_i^\ell + \frac{\rho}{2} \left\| z_i - a_i^\ell \right\|_2^2$
2. Central step
   $a^{\ell+1} \;=\; \arg\min_a \; g(\bar{a}) - \sum_{i=1}^{\mathcal{I}} a_i^\top \lambda_i^\ell + \frac{\rho}{2} \left\| z_i^{\ell+1} - a_i \right\|_2^2$
3. Dual update
   $\lambda_i^{\ell+1} \;=\; \lambda_i^\ell + \rho(z_i^{\ell+1} - a_i^{\ell+1})$

Alternating Direction Method of Multipliers

[Boyd et al., 2011]

Augmented Lagrangian: dual λ and $\rho > 0$

$$\mathcal{L}_\rho(z, a, \lambda) = g(\bar{a}) + \sum_{i=1}^{\mathcal{I}} \left(\lambda_i^\top (z_i - a_i) + \frac{\rho}{2} \|z_i - a_i\|_2^2 \right)$$

Modified ADMM Scheme [Braun et al., 2018]

1. Parallel step
   $$z_i^{\ell+1} \;=\; \arg\min_{z_i \in \mathbb{D}_i} \; \tfrac{\rho}{2} \left\| z_i - z_i^\ell + \Pi^\ell \right\|_2^2$$

Alternating Direction Method of Multipliers

[Boyd et al., 2011]

Augmented Lagrangian: dual λ and $\rho > 0$

$$\mathcal{L}_\rho(z, a, \lambda) = g(\bar{a}) + \sum_{i=1}^{\mathcal{I}} \left(\lambda_i^\top (z_i - a_i) + \frac{\rho}{2} \|z_i - a_i\|_2^2 \right)$$

Modified ADMM Scheme [Braun et al., 2018]

1. Parallel step

   $$z_i^{\ell+1} \;=\; \arg\min_{z_i \in \mathbb{D}_i} \; \tfrac{\rho}{2} \left\| z_i - z_i^\ell + \Pi^\ell \right\|_2^2$$

2. Central step: Compute average $\bar{z}^{\ell+1}$

   $$\bar{a}^{\ell+1} \;=\; \arg\min_{\bar{a}} \; g(\bar{a}) + \tfrac{\rho \mathcal{I}}{2} \left\| \bar{z}^{\ell+1} - \bar{a} + \tfrac{\bar{\lambda}^\ell}{\rho} \right\|_2^2$$

Alternating Direction Method of Multipliers

[Boyd et al., 2011]

Augmented Lagrangian: dual λ and $\rho > 0$

$$\mathcal{L}_\rho(z, a, \lambda) = g(\bar{a}) + \sum_{i=1}^{\mathcal{I}} \left(\lambda_i^\top (z_i - a_i) + \frac{\rho}{2} \| z_i - a_i \|_2^2 \right)$$

Modified ADMM Scheme [Braun et al., 2018]

1. Parallel step
   $z_i^{\ell+1} \;=\; \arg\min_{z_i \in \mathbb{D}_i} \; \frac{\rho}{2} \left\| z_i - z_i^\ell + \Pi^\ell \right\|_2^2$
2. Central step: Compute average $\bar{z}^{\ell+1}$
   $\bar{a}^{\ell+1} \;=\; \arg\min_{\bar{a}} \; g(\bar{a}) + \frac{\rho\mathcal{I}}{2} \left\| \bar{z}^{\ell+1} - \bar{a} + \frac{\bar{\lambda}^\ell}{\rho} \right\|_2^2$
3. Dual update
   $\bar{\lambda}^{\ell+1} \;=\; \bar{\lambda}^\ell + \rho(\bar{z}^{\ell+1} - \bar{a}^{\ell+1})$

Alternating Direction Method of Multipliers

[Boyd et al., 2011]

Augmented Lagrangian: dual λ and $\rho > 0$

$$\mathcal{L}_\rho(z, a, \lambda) = g(\bar{a}) + \sum_{i=1}^{\mathcal{I}} \left(\lambda_i^\top (z_i - a_i) + \frac{\rho}{2} \|z_i - a_i\|_2^2 \right)$$

Modified ADMM Scheme [Braun et al., 2018]

1. Parallel step
   $z_i^{\ell+1} \; = \; \arg\min_{z_i \in \mathbb{D}_i} \; \frac{\rho}{2} \left\| z_i - z_i^\ell + \Pi^\ell \right\|_2^2$
2. Central step: Compute average $\bar{z}^{\ell+1}$
   $\bar{a}^{\ell+1} \; = \; \arg\min_{\bar{a}} \; g(\bar{a}) + \frac{\rho \mathcal{I}}{2} \left\| \bar{z}^{\ell+1} - \bar{a} + \frac{\bar{\lambda}^\ell}{\rho} \right\|_2^2$
3. Dual update
   $\bar{\lambda}^{\ell+1} \; = \; \bar{\lambda}^\ell + \rho(\bar{z}^{\ell+1} - \bar{a}^{\ell+1})$
4. Broadcast variable
   $\Pi^{\ell+1} \; = \; \bar{z}^{\ell+1} - \bar{a}^{\ell+1} + \frac{\bar{\lambda}^{\ell+1}}{\rho}$

Current Research

Distributed Optimization using ALADIN

(joint work with Yuning Jiang, ShanghaiTech University)

Distributed Optimization using ALADIN

(joint work with Yuning Jiang, ShanghaiTech University)

Additional Local Costs

Penalize the (dis-)charging of the batteries by introducing $f_i : \mathbb{R}^{2N} \to \mathbb{R}$

$$f_i(u_i) = \|u_i\|_{Q_i}^2$$

with some scaling matrix $Q_i \in \mathbb{R}^{2N}$.

Distributed Optimization using ALADIN

(joint work with Yuning Jiang, ShanghaiTech University)

Additional Local Costs

Penalize the (dis-)charging of the batteries by introducing $f_i : \mathbb{R}^{2N} \to \mathbb{R}$

$$f_i(u_i) = \|u_i\|_{Q_i}^2$$

with some scaling matrix $Q_i \in \mathbb{R}^{2N}$.

Optimization Problem

$$\begin{aligned}
\min_{\bar{z},u} \quad & g(\bar{z}) + \sum_{i=1}^{\mathcal{I}} f_i(u_i) \\
\text{s.t.} \quad & \bar{z} = \bar{w} + \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} I_N \otimes \begin{pmatrix} 1 & \gamma_i \end{pmatrix} u_i \\
& u_i \in \mathbb{D}_i = \left\{ u_i \in \mathbb{R}^{2N} \;\middle|\; D_i u_i \leq d_i \right\}, \quad i \in [1 : \mathcal{I}]
\end{aligned}$$

Distributed Optimization using ALADIN

[Houska et al., 2016]

1. Parallel step: Solve

$$\min_{v_i \in \mathbb{D}_i} \quad f_i(v_i) - \lambda^\top A_i v_i + \frac{1}{2} \|v_i - u_i\|_{H_i}^2$$

and compute $g_i = A_i^\top \lambda + H_i(u_i - v_i)$ in parallel.

Distributed Optimization using ALADIN

[Houska et al., 2016]

1. Parallel step: Solve

$$\min_{v_i \in \mathbb{D}_i} \quad f_i(v_i) - \lambda^\top A_i v_i + \frac{1}{2} \|v_i - u_i\|_{H_i}^2$$

   and compute $g_i = A_i^\top \lambda + H_i(u_i - v_i)$ in parallel.

2. Terminal condition: Terminate if $\max_{i \in [1:\mathcal{I}]} \|v_i - u_i\|_\infty \leq \varepsilon$.

Distributed Optimization using ALADIN

[Houska et al., 2016]

1. Parallel step: Solve

$$\min_{v_i \in \mathbb{D}_i} \quad f_i(v_i) - \lambda^\top A_i v_i + \frac{1}{2} \left\| v_i - u_i \right\|_{H_i}^2$$

   and compute $g_i = A_i^\top \lambda + H_i(u_i - v_i)$ in parallel.

2. Terminal condition: Terminate if $\max_{i \in [1:\mathcal{I}]} \left\| v_i - u_i \right\|_\infty \leq \varepsilon$.
3. Consensus step: Update μ and $H_i = 2Q_i + \mu D_i^{\mathrm{act}^\top} D_i^{\mathrm{act}}$ optionally, solve

$$\min_{\bar{z}^+, u^+} \quad g(\bar{z}^+) + \sum_{i=1}^{\mathcal{I}} \frac{1}{2} \left\| u_i^+ - v_i \right\|_{H_i}^2 + g_i^\top u_i^+$$

$$\text{s.t.} \quad \bar{z}^+ = \bar{w} + \sum_{i=1}^{\mathcal{I}} A_i u_i^+ \quad | \lambda^+$$

   and update $\bar{z} = \bar{z}^+$, $u = u^+$, $\lambda = \lambda^+$.

Comparison of Both Algorithms

	ADMM	ALADIN
requirements		
convergence rate		
communication ↑		
communication ↓		

Comparison of Both Algorithms

	ADMM	ALADIN
requirements	convex objective functions	
convergence rate		
communication ↑		
communication ↓		

Comparison of Both Algorithms

	ADMM	ALADIN
requirements	convex objective functions	$\mathcal{C}^2$ objective functions
convergence rate		
communication $\uparrow$		
communication $\downarrow$		

Comparison of Both Algorithms

	ADMM	ALADIN
requirements	convex objective functions	$\mathcal{C}^2$ objective functions
convergence rate	linear	
communication $\uparrow$		
communication $\downarrow$		

Comparison of Both Algorithms

	ADMM	ALADIN
requirements	convex objective functions	$\mathcal{C}^2$ objective functions
convergence rate	linear	globally linear locally quadratic
communication $\uparrow$		
communication $\downarrow$		

Philipp Sauerteig
Institute for Mathematics

TECHNISCHE UNIVERSITÄT ILMENAU

Comparison of Both Algorithms

	ADMM	ALADIN
requirements	convex objective functions	$\mathcal{C}^2$ objective functions
convergence rate	linear	globally linear locally quadratic
communication $\uparrow$	N	
communication $\downarrow$		

Comparison of Both Algorithms

	ADMM	ALADIN
requirements	convex objective functions	$\mathcal{C}^2$ objective functions
convergence rate	linear	globally linear locally quadratic
communication $\uparrow$	N	
communication $\downarrow$	N	

Comparison of Both Algorithms

	ADMM	ALADIN
requirements	convex objective functions	$\mathcal{C}^2$ objective functions
convergence rate	linear	globally linear locally quadratic
communication $\uparrow$	N	$4N$ or $(4+2n_i^{\text{act}})N$
communication $\downarrow$	N	

Comparison of Both Algorithms

	ADMM	ALADIN
requirements	convex objective functions	$\mathcal{C}^2$ objective functions
convergence rate	linear	globally linear locally quadratic
communication $\uparrow$	N	$4N$ or $(4+2n_i^{\mathrm{act}})N$
communication $\downarrow$	N	$2N$ or $2N+1$

Comparison of Both Algorithms

	ADMM	ALADIN
requirements	convex objective functions	$\mathcal{C}^2$ objective functions
convergence rate	linear	globally linear locally quadratic
communication $\uparrow$	N	$4N$ or $(4 + 2n_i^{\mathrm{act}})N$
communication $\downarrow$	N	$2N$ or $2N + 1$

Outlook: ALADIN could be applied to

- more complex (in particular non-convex) objective functions
- nonlinear battery dynamics

Multiobjective Optimization Problem

[S., Worthmann, 2019]

Multiobjective Optimization Problem

[S., Worthmann, 2019]

Objective 1 Flatten aggregated power demand profile:

$$\min_{\bar{z}\in\mathbb{R}^N} \quad g(\bar{z}) = \left\|\bar{z} - \bar{\zeta}\right\|_2^2 .$$

Multiobjective Optimization Problem

[S., Worthmann, 2019]

Objective 1 Flatten aggregated power demand profile:

$$\min_{\bar{z} \in \mathbb{R}^N} \quad g(\bar{z}) = \left\| \bar{z} - \bar{\zeta} \right\|_2^2 .$$

Objective 2 [Braun et al., 2018]: Stay within time-varying tube constraints, i.e.

$$\underline{c} \leq \bar{z} \leq \bar{c} \tag{1}$$

for some $\underline{c}, \bar{c} \in \mathbb{R}^N$

Multiobjective Optimization Problem

[S., Worthmann, 2019]

Objective 1 Flatten aggregated power demand profile:

$$\min_{\bar{z} \in \mathbb{R}^N} \quad g(\bar{z}) = \left\| \bar{z} - \bar{\zeta} \right\|_2^2 .$$

Objective 2 [Braun et al., 2018]: Stay within time-varying tube constraints, i.e.

$$\underline{c} - \underline{s} \quad \leq \quad \bar{z} \quad \leq \quad \bar{c} + \bar{s} \tag{1}$$

for some $\underline{c}, \bar{c} \in \mathbb{R}^N$ and auxiliary variables $\underline{s}, \bar{s} \in \mathbb{R}^N_{\geq 0}$:

$$\min_{s \in \mathbb{R}^{2N}} \quad h(s) = \|s\|^2 .$$

Multiobjective Optimization Problem

[S., Worthmann, 2019]

Objective 1 Flatten aggregated power demand profile:

$$\min_{\bar{z} \in \mathbb{R}^N} \quad g(\bar{z}) = \left\| \bar{z} - \bar{\zeta} \right\|_2^2 .$$

Objective 2 [Braun et al., 2018]: Stay within time-varying tube constraints, i.e.

$$\underline{c} - \underline{s} \quad \leq \quad \bar{z} \quad \leq \quad \bar{c} + \bar{s} \tag{1}$$

for some $\underline{c}, \bar{c} \in \mathbb{R}^N$ and auxiliary variables $\underline{s}, \bar{s} \in \mathbb{R}^N_{\geq 0}$:

$$\min_{s \in \mathbb{R}^{2N}} \quad h(s) = \|s\|^2 .$$

Multiobjective Optimization Problem Formulation

$$\min_{\bar{z}, s} \quad \begin{pmatrix} g(\bar{z}) \\ h(s) \end{pmatrix} \quad \text{s.t.} \quad \text{system dynamics and constraints and (1)}$$

Efficiency, Pareto Frontier

Definition

Consider

$$\min_{(\bar{z},s)\in\mathbb{S}} \begin{pmatrix} g(\bar{z}) \\ h(s) \end{pmatrix}, \tag{2}$$

where $\mathbb{S}$ denotes the feasible set.

Efficiency, Pareto Frontier

Definition

Consider

$$\min_{(\bar{z},s)\in\mathbb{S}} \begin{pmatrix} g(\bar{z}) \\ h(s) \end{pmatrix}, \tag{2}$$

where $\mathbb{S}$ denotes the feasible set. A point $(\bar{z}^*, s^*) \in \mathbb{S}$ is called efficient (or Pareto optimal) for (2) if for all $(\bar{z}, s) \in \mathbb{S}$:

$$\begin{aligned} & g(\bar{z}) < g(\bar{z}^*) && \Rightarrow && h(s^*) < h(s) \\ \textit{and} \quad & h(s) < h(s^*) && \Rightarrow && g(\bar{z}^*) < g(\bar{z}). \end{aligned}$$

Efficiency, Pareto Frontier

Definition

Consider

$$\min_{(\bar{z},s)\in\mathbb{S}} \begin{pmatrix} g(\bar{z}) \\ h(s) \end{pmatrix}, \tag{2}$$

where $\mathbb{S}$ *denotes the feasible set. A point* $(\bar{z}^*, s^*) \in \mathbb{S}$ *is called* *efficient (or Pareto optimal)* *for* (2) *if for all* $(\bar{z}, s) \in \mathbb{S}$*:*

$$\begin{aligned} & g(\bar{z}) < g(\bar{z}^*) \quad \Rightarrow \quad h(s^*) < h(s) \\ \textit{and} \quad & h(s) < h(s^*) \quad \Rightarrow \quad g(\bar{z}^*) < g(\bar{z}). \end{aligned}$$

The set

$$\left\{ (g(\bar{z}^*), h(s^*)) \in \mathbb{R}^2 \;\middle|\; (\bar{z}^*, s^*) \textit{ is efficient for } (2) \right\}$$

is called *Pareto frontier*.

Characterization of Efficiency

Proposition (S., Worthmann, 2019, Ehrgott, 2005)

Consider

$$\min_{(\bar{z},s)\in\mathbb{S}} \begin{pmatrix} g(\bar{z}) \\ h(s) \end{pmatrix} \tag{2}$$

Characterization of Efficiency

Proposition (S., Worthmann, 2019, Ehrgott, 2005)

Consider

$$\min_{(\bar{z},s)\in\mathbb{S}} \begin{pmatrix} g(\bar{z}) \\ h(s) \end{pmatrix} \tag{2}$$

and for $\kappa \in [0,1]$

$$\min_{(\bar{z},s)\in\mathbb{S}} \quad f_\kappa(\bar{z},s) \quad = \quad \kappa g(\bar{z}) + (1-\kappa)h(s). \tag{3}$$

Characterization of Efficiency

Proposition (S., Worthmann, 2019, Ehrgott, 2005)

Consider

$$\min_{(\bar{z},s)\in\mathbb{S}} \begin{pmatrix} g(\bar{z}) \\ h(s) \end{pmatrix} \tag{2}$$

and for $\kappa \in [0,1]$

$$\min_{(\bar{z},s)\in\mathbb{S}} f_\kappa(\bar{z},s) = \kappa g(\bar{z}) + (1-\kappa)h(s). \tag{3}$$

The following statements hold true:

1. $(\bar{z}^*, s^*)$ *efficient for* (2) $\Rightarrow \exists\, \kappa \in [0,1] : (\bar{z}^*, s^*)$ *optimal for* (3)

Characterization of Efficiency

Proposition (S., Worthmann, 2019, Ehrgott, 2005)

Consider

$$\min_{(\bar{z},s)\in\mathbb{S}} \begin{pmatrix} g(\bar{z}) \\ h(s) \end{pmatrix} \tag{2}$$

and for $\kappa \in [0,1]$

$$\min_{(\bar{z},s)\in\mathbb{S}} \quad f_\kappa(\bar{z},s) \quad = \quad \kappa g(\bar{z}) + (1-\kappa)h(s). \tag{3}$$

The following statements hold true:

1. $(\bar{z}^*, s^*)$ *efficient for* (2) $\Rightarrow \exists\, \kappa \in [0,1] : (\bar{z}^*, s^*)$ *optimal for* (3)
2. $\kappa \in (0,1)$, $(\bar{z}^*, s^*)$ *optimal for* (3) $\Rightarrow (\bar{z}^*, s^*)$ *efficient for* (2)

Characterization of Efficiency

Proposition (S., Worthmann, 2019, Ehrgott, 2005)

Consider

$$\min_{(\bar{z},s)\in\mathbb{S}} \quad \begin{pmatrix} g(\bar{z}) \\ h(s) \end{pmatrix} \tag{2}$$

and for $\kappa \in [0,1]$

$$\min_{(\bar{z},s)\in\mathbb{S}} \quad f_\kappa(\bar{z},s) \quad = \quad \kappa g(\bar{z}) + (1-\kappa)h(s). \tag{3}$$

The following statements hold true:

1. $(\bar{z}^*, s^*)$ *efficient for* (2) $\Rightarrow \exists\, \kappa \in [0,1] : (\bar{z}^*, s^*)$ *optimal for* (3)
2. $\kappa \in (0,1)$, $(\bar{z}^*, s^*)$ *optimal for* (3) $\Rightarrow (\bar{z}^*, s^*)$ *efficient for* (2)
3. $\kappa = 0$, $(\bar{z}, s^*)$ *optimal for* (3) $\Rightarrow \exists!\, \bar{z}^* :\ (\bar{z}^*, s^*)$ *efficient for* (2)

Characterization of Efficiency

Proposition (S., Worthmann, 2019, Ehrgott, 2005)

Consider

$$\min_{(\bar{z},s)\in\mathbb{S}} \quad \begin{pmatrix} g(\bar{z}) \\ h(s) \end{pmatrix} \tag{2}$$

and for $\kappa \in [0,1]$

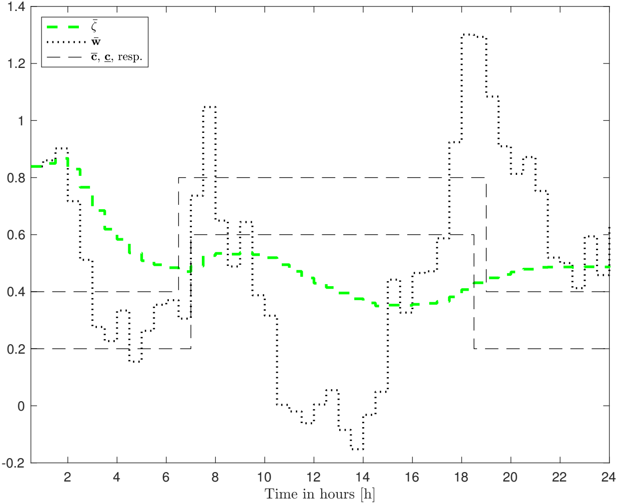
$$\min_{(\bar{z},s)\in\mathbb{S}} \quad f_\kappa(\bar{z},s) \quad = \quad \kappa g(\bar{z}) + (1-\kappa)h(s). \tag{3}$$

The following statements hold true:

1. $(\bar{z}^*, s^*)$ *efficient for* (2) $\Rightarrow \exists\, \kappa \in [0,1] : (\bar{z}^*, s^*)$ *optimal for* (3)
2. $\kappa \in (0,1)$, $(\bar{z}^*, s^*)$ *optimal for* (3) $\Rightarrow (\bar{z}^*, s^*)$ *efficient for* (2)
3. $\kappa = 0$, $(\bar{z}, s^*)$ *optimal for* (3) $\Rightarrow \exists!\, \bar{z}^* : (\bar{z}^*, s^*)$ *efficient for* (2)
4. $\kappa = 1$, $(\bar{z}^*, s)$ *optimal for* (3) $\Rightarrow \exists!\, s^* : (\bar{z}^*, s^*)$ *efficient for* (2)

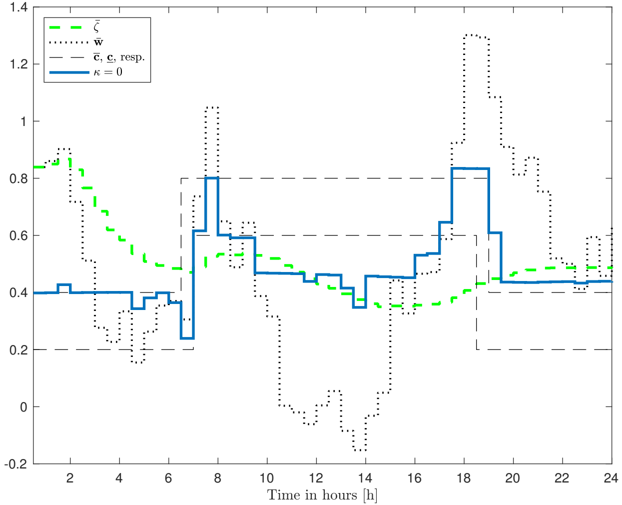
Efficient Control

Setting



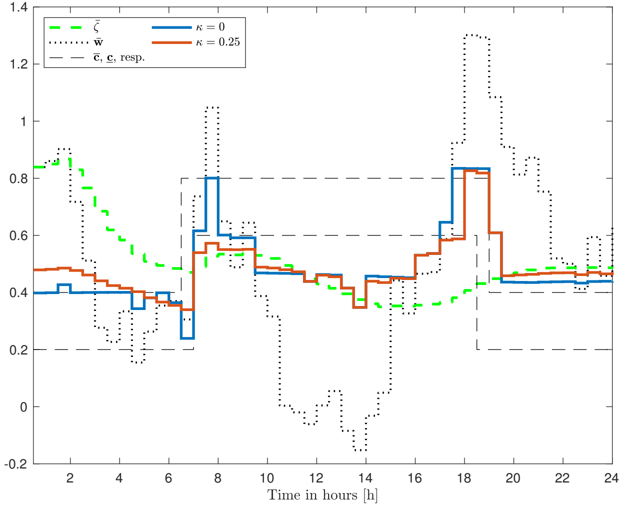
Efficient Control

Open Loop Performance



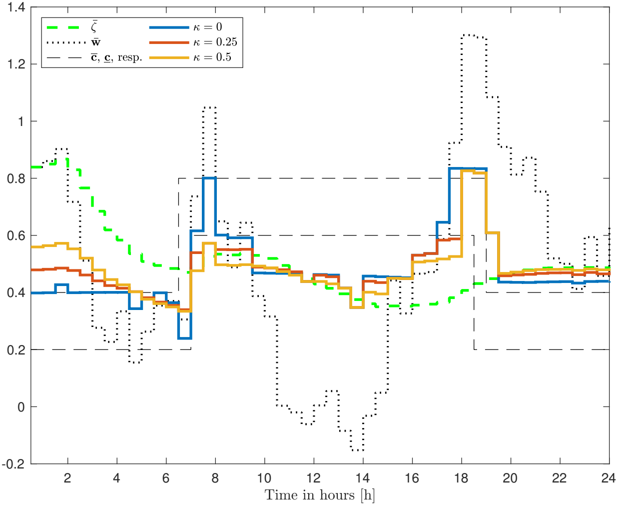
Efficient Control

Open Loop Performance



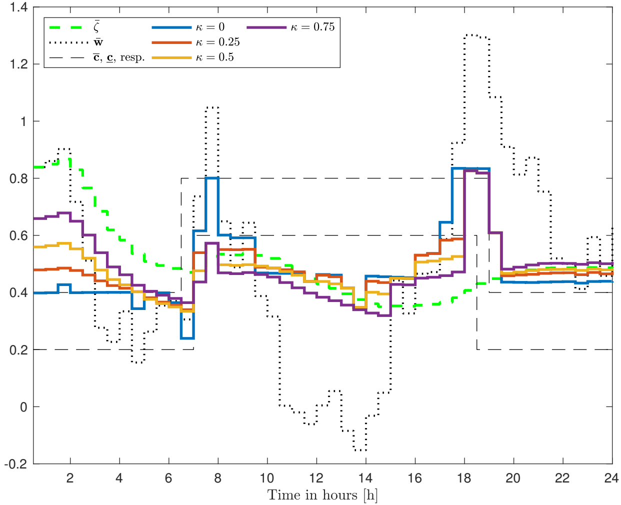
Efficient Control

Open Loop Performance



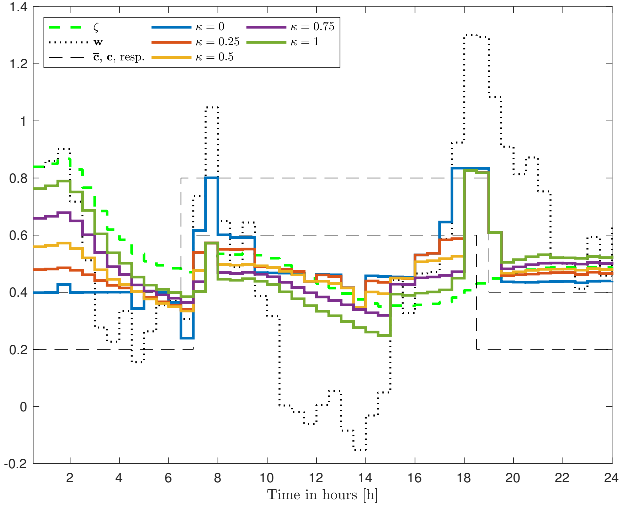
Efficient Control

Open Loop Performance

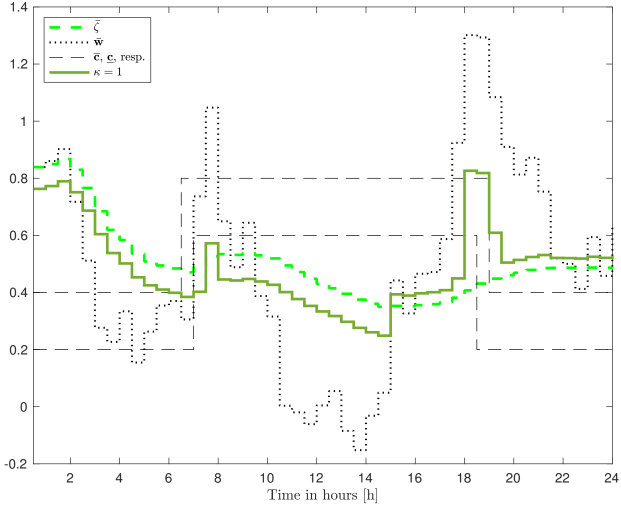


Efficient Control

Open Loop Performance

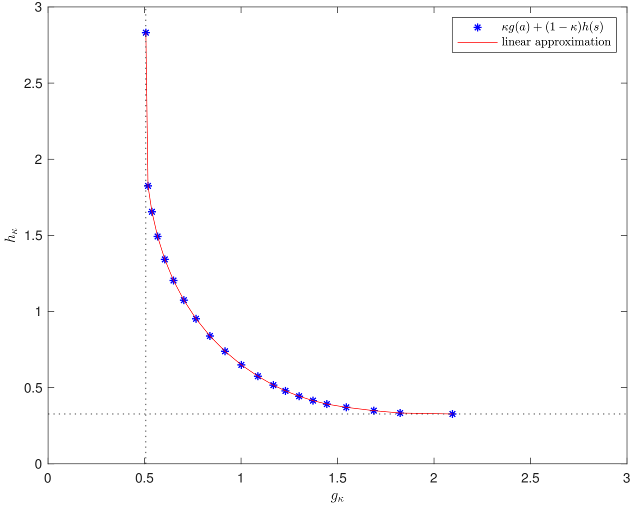


Efficient Control
Open Loop Performance



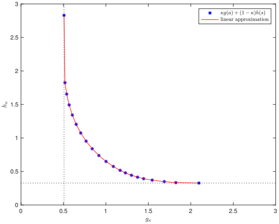
Efficient Control

Pareto Frontier



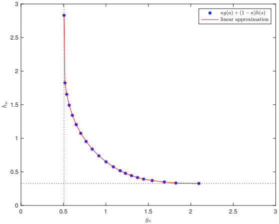
Proper Efficiency

Pareto Frontier



Proper Efficiency

Pareto Frontier

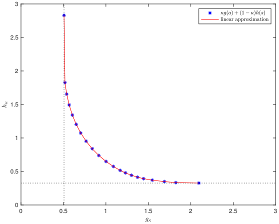


Interpretation

- efficient points = non-dominated points

Proper Efficiency

Pareto Frontier



Interpretation

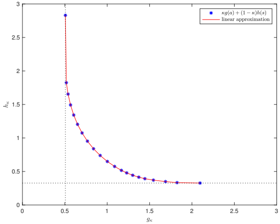
- efficient points = non-dominated points

Definition

A point $(\bar{z}^*, s^*) \in \mathbb{S}$ *is called properly efficient for* (2)

Proper Efficiency

Pareto Frontier



Interpretation

- efficient points = non-dominated points

Definition

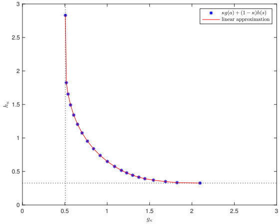
A point $(\bar{z}^*, s^*) \in \mathbb{S}$ *is called* *properly efficient* *for* (2) *if it is efficient and there exists* $L > 0$ *such that for all* $(\bar{z}, s) \in \mathbb{S}$*:*

$$g(\bar{z}) < g(\bar{z}^*) \quad \Rightarrow \quad \frac{g(\bar{z}^*) - g(\bar{z})}{h(s) - h(s^*)} \leq L$$

$$h(s) < h(s^*) \quad \Rightarrow \quad \frac{h(s^*) - h(s)}{g(\bar{z}) - g(\bar{z}^*)} \leq L$$

Proper Efficiency

Pareto Frontier



Interpretation

- efficient points = non-dominated points
- trade-off is bounded for $\kappa \in (0, 1)$

Definition

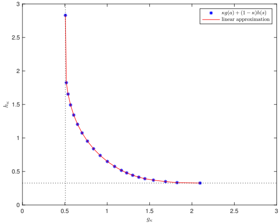
A point $(\bar{z}^*, s^*) \in \mathbb{S}$ is called properly efficient *for* (2) *if it is efficient and there exists $L > 0$ such that for all $(\bar{z}, s) \in \mathbb{S}$:*

$$g(\bar{z}) < g(\bar{z}^*) \quad \Rightarrow \quad \frac{g(\bar{z}^*) - g(\bar{z})}{h(s) - h(s^*)} \leq L$$

$$h(s) < h(s^*) \quad \Rightarrow \quad \frac{h(s^*) - h(s)}{g(\bar{z}) - g(\bar{z}^*)} \leq L$$

Proper Efficiency

Pareto Frontier



Interpretation

- efficient points = non-dominated points
- trade-off is bounded for $\kappa \in (0, 1)$

$\rightsquigarrow$ information on costs of improvement w.r.t. one objective

Definition

A point $(\bar{z}^*, s^*) \in \mathbb{S}$ is called properly efficient for (2) if it is efficient and there exists $L > 0$ such that for all $(\bar{z}, s) \in \mathbb{S}$:

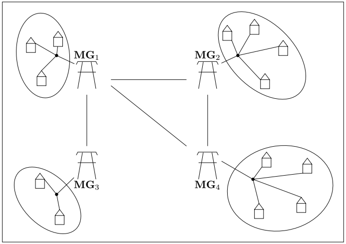
$$
\begin{aligned}
g(\bar{z}) < g(\bar{z}^*) \quad &\Rightarrow \quad \frac{g(\bar{z}^*) - g(\bar{z})}{h(s) - h(s^*)} \leq L \\
h(s) < h(s^*) \quad &\Rightarrow \quad \frac{h(s^*) - h(s)}{g(\bar{z}) - g(\bar{z}^*)} \leq L
\end{aligned}
$$

Coupled Microgrids

(joint work with Philipp Braun (University of Newcastle),
[Braun, S., Worthmann, 2019])

Coupled Microgrids

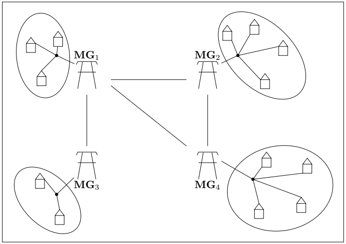
(joint work with Philipp Braun (University of Newcastle), [Braun, S., Worthmann, 2019])



- $\Xi \in \mathbb{N}$ number of MGs
- $\delta_{\kappa,\nu} \in [0, 1]$ exchange between MG_κ and MG_ν
- $\eta_{\kappa,\nu} \in [0, 1]$ efficiency of the exchange

Coupled Microgrids

(joint work with Philipp Braun (University of Newcastle), [Braun, S., Worthmann, 2019])



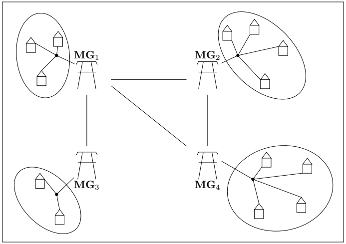
- $\Xi \in \mathbb{N}$ number of MGs
- $\delta_{\kappa,\nu} \in [0,1]$ exchange between MG_κ and MG_ν
- $\eta_{\kappa,\nu} \in [0,1]$ efficiency of the exchange

Optimization Problem

$$\min_{\delta} \quad J(\delta) = \sum_{n=k}^{k+N-1} \sum_{\kappa=1}^{\Xi} \left(\mathcal{I}_\kappa \bar{\zeta}_\kappa(n) - \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \mathcal{I}_\nu \bar{z}_\nu(n) \right)^2$$

Coupled Microgrids

(joint work with Philipp Braun (University of Newcastle), [Braun, S., Worthmann, 2019])



- $\Xi \in \mathbb{N}$ number of MGs
- $\delta_{\kappa,\nu} \in [0,1]$ exchange between MG_κ and MG_ν
- $\eta_{\kappa,\nu} \in [0,1]$ efficiency of the exchange

Optimization Problem

$$\min_{\delta} \quad J(\delta) = \sum_{n=k}^{k+N-1} \sum_{\kappa=1}^{\Xi} \left(\mathcal{I}_\kappa \bar{\zeta}_\kappa(n) - \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \mathcal{I}_\nu \bar{z}_\nu(n) \right)^2$$

$$\text{s.t.} \quad \delta \in \Delta = \left\{ \delta \;\middle|\; \sum_{\nu=1}^{\Xi} \delta_{\kappa,\nu}(n) = 1 \text{ and } \delta_{\kappa,\nu}(n) \cdot \delta_{\nu,\kappa}(n) \leq 0 \; (\kappa \neq \nu) \right\}$$

Bilevel Optimization Scheme

1. MG_κ: Solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa} g_\kappa(\bar{z}_\kappa)$ and send $\bar{z}_\kappa$ to CE.

Bilevel Optimization Scheme

1. MG_κ: Solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa} g_\kappa(\bar{z}_\kappa)$ and send $\bar{z}_\kappa$ to CE.
2. CE: Collect $\bar{z}_\kappa$, $\kappa \in [1 : \Xi]$, solve $\min_{\delta \in \Delta} J(\delta)$ and compute

$$\bar{z}_\kappa^+(n) = \frac{1}{\mathcal{I}_\kappa} \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \bar{z}_\nu(n) \mathcal{I}_\nu, \quad \kappa \in [1 : \Xi].$$

Bilevel Optimization Scheme

1. MG_κ: Solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa} g_\kappa(\bar{z}_\kappa)$ and send $\bar{z}_\kappa$ to CE.
2. CE: Collect $\bar{z}_\kappa$, $\kappa \in [1:\Xi]$, solve $\min_{\delta \in \Delta} J(\delta)$ and compute

$$\bar{z}_\kappa^+(n) = \frac{1}{\mathcal{I}_\kappa} \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \bar{z}_\nu(n) \mathcal{I}_\nu, \quad \kappa \in [1:\Xi].$$

3. CE: Communicate $\Delta \bar{z}_\kappa = \bar{z}_\kappa^+ - \bar{z}$ to MG_κ, $\kappa \in [1:\Xi]$.

Bilevel Optimization Scheme

1. MG_κ: Solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa} g_\kappa(\bar{z}_\kappa)$ and send $\bar{z}_\kappa$ to CE.
2. CE: Collect $\bar{z}_\kappa$, $\kappa \in [1 : \Xi]$, solve $\min_{\delta \in \Delta} J(\delta)$ and compute

$$\bar{z}_\kappa^+(n) = \frac{1}{\mathcal{I}_\kappa} \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \bar{z}_\nu(n) \mathcal{I}_\nu, \quad \kappa \in [1 : \Xi].$$

3. CE: Communicate $\Delta \bar{z}_\kappa = \bar{z}_\kappa^+ - \bar{z}$ to MG_κ, $\kappa \in [1 : \Xi]$.
4. MG_κ: Update $\bar{w}_\kappa^+ = \bar{w}_\kappa + \Delta \bar{z}_\kappa$ and solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa^+} g_\kappa(\bar{z}_\kappa)$.

Bilevel Optimization Scheme

1. MG_κ: Solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa} g_\kappa(\bar{z}_\kappa)$ and send $\bar{z}_\kappa$ to CE.
2. CE: Collect $\bar{z}_\kappa$, $\kappa \in [1:\Xi]$, solve $\min_{\delta \in \Delta} J(\delta)$ and compute
$$\bar{z}_\kappa^+(n) = \frac{1}{\mathcal{I}_\kappa} \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \bar{z}_\nu(n) \mathcal{I}_\nu, \quad \kappa \in [1:\Xi].$$
3. CE: Communicate $\Delta \bar{z}_\kappa = \bar{z}_\kappa^+ - \bar{z}$ to MG_κ, $\kappa \in [1:\Xi]$.
4. MG_κ: Update $\bar{w}_\kappa^+ = \bar{w}_\kappa + \Delta \bar{z}_\kappa$ and solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa^+} g_\kappa(\bar{z}_\kappa)$.

Results

MG	$\lVert\bar{z} - \bar{\zeta}\rVert_2^2$	$\lVert\bar{z}^+ - \bar{\zeta}\rVert_2^2$
1	920.57	451.44
2	124.85	85.07
3	32.25	79.82
4	34.03	85.07
overall	1111.70	701.39

Bilevel Optimization Scheme

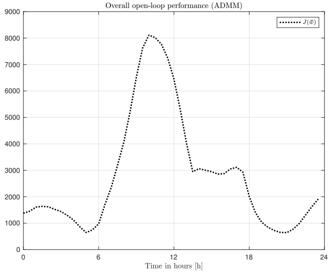
1. MG_κ: Solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa} g_\kappa(\bar{z}_\kappa)$ and send $\bar{z}_\kappa$ to CE.
2. CE: Collect $\bar{z}_\kappa$, $\kappa \in [1:\Xi]$, solve $\min_{\delta \in \Delta} J(\delta)$ and compute

$$\bar{z}_\kappa^+(n) = \frac{1}{\mathcal{I}_\kappa} \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \bar{z}_\nu(n) \mathcal{I}_\nu, \quad \kappa \in [1:\Xi].$$

3. CE: Communicate $\Delta \bar{z}_\kappa = \bar{z}_\kappa^+ - \bar{z}$ to MG_κ, $\kappa \in [1:\Xi]$.
4. MG_κ: Update $\bar{w}_\kappa^+ = \bar{w}_\kappa + \Delta \bar{z}_\kappa$ and solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa^+} g_\kappa(\bar{z}_\kappa)$.

Results

MG	$\lVert \bar{z} - \bar{\zeta} \rVert_2^2$	$\lVert \bar{z}^+ - \bar{\zeta} \rVert_2^2$
1	920.57	451.44
2	124.85	85.07
3	32.25	79.82
4	34.03	85.07
overall	1111.70	701.39



Bilevel Optimization Scheme

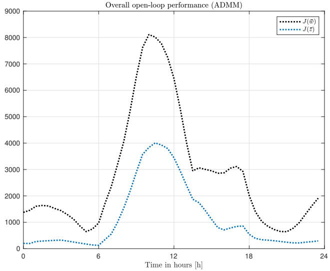
1. MG_κ: Solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa} g_\kappa(\bar{z}_\kappa)$ and send $\bar{z}_\kappa$ to CE.
2. CE: Collect $\bar{z}_\kappa$, $\kappa \in [1 : \Xi]$, solve $\min_{\delta \in \Delta} J(\delta)$ and compute

$$\bar{z}_\kappa^+(n) = \frac{1}{\mathcal{I}_\kappa} \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \bar{z}_\nu(n) \mathcal{I}_\nu, \quad \kappa \in [1 : \Xi].$$

3. CE: Communicate $\Delta \bar{z}_\kappa = \bar{z}_\kappa^+ - \bar{z}$ to MG_κ, $\kappa \in [1 : \Xi]$.
4. MG_κ: Update $\bar{w}_\kappa^+ = \bar{w}_\kappa + \Delta \bar{z}_\kappa$ and solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa^+} g_\kappa(\bar{z}_\kappa)$.

Results

MG	$\lVert \bar{z} - \bar{\zeta} \rVert_2^2$	$\lVert \bar{z}^+ - \bar{\zeta} \rVert_2^2$
1	920.57	451.44
2	124.85	85.07
3	32.25	79.82
4	34.03	85.07
overall	1111.70	701.39



Bilevel Optimization Scheme

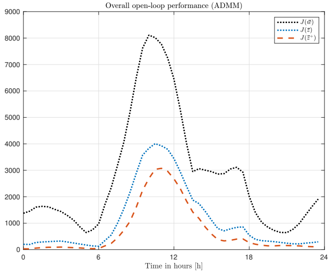
1. MG_κ: Solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa} g_\kappa(\bar{z}_\kappa)$ and send $\bar{z}_\kappa$ to CE.
2. CE: Collect $\bar{z}_\kappa$, $\kappa \in [1:\Xi]$, solve $\min_{\delta \in \Delta} J(\delta)$ and compute

$$\bar{z}_\kappa^+(n) = \frac{1}{\mathcal{I}_\kappa} \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \bar{z}_\nu(n) \mathcal{I}_\nu, \quad \kappa \in [1:\Xi].$$

3. CE: Communicate $\Delta \bar{z}_\kappa = \bar{z}_\kappa^+ - \bar{z}$ to MG_κ, $\kappa \in [1:\Xi]$.
4. MG_κ: Update $\bar{w}_\kappa^+ = \bar{w}_\kappa + \Delta \bar{z}_\kappa$ and solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa^+} g_\kappa(\bar{z}_\kappa)$.

Results

MG	$\|\bar{z} - \bar{\zeta}\|_2^2$	$\|\bar{z}^+ - \bar{\zeta}\|_2^2$
1	920.57	451.44
2	124.85	85.07
3	32.25	79.82
4	34.03	85.07
overall	1111.70	701.39



Philipp Sauerteig
Institute for Mathematics

TECHNISCHE UNIVERSITÄT ILMENAU

Bilevel Optimization Scheme

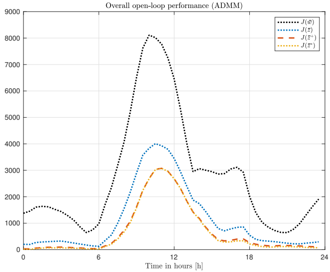
1. MG_κ: Solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa} g_\kappa(\bar{z}_\kappa)$ and send $\bar{z}_\kappa$ to CE.
2. CE: Collect $\bar{z}_\kappa$, $\kappa \in [1:\Xi]$, solve $\min_{\delta \in \Delta} J(\delta)$ and compute

$$\bar{z}_\kappa^+(n) = \frac{1}{\mathcal{I}_\kappa} \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \bar{z}_\nu(n) \mathcal{I}_\nu, \quad \kappa \in [1:\Xi].$$

3. CE: Communicate $\Delta \bar{z}_\kappa = \bar{z}_\kappa^+ - \bar{z}$ to MG_κ, $\kappa \in [1:\Xi]$.
4. MG_κ: Update $\bar{w}_\kappa^+ = \bar{w}_\kappa + \Delta \bar{z}_\kappa$ and solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa^+} g_\kappa(\bar{z}_\kappa)$.

Results

MG	$\lVert \bar{z} - \bar{\zeta} \rVert_2^2$	$\lVert \bar{z}^+ - \bar{\zeta} \rVert_2^2$
1	920.57	451.44
2	124.85	85.07
3	32.25	79.82
4	34.03	85.07
overall	1111.70	701.39



Bilevel Optimization Scheme

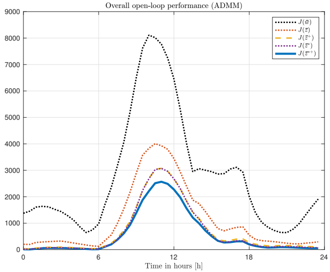
1. MG_κ: Solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa} g_\kappa(\bar{z}_\kappa)$ and send $\bar{z}_\kappa$ to CE.
2. CE: Collect $\bar{z}_\kappa$, $\kappa \in [1:\Xi]$, solve $\min_{\delta \in \Delta} J(\delta)$ and compute

$$\bar{z}_\kappa^+(n) = \frac{1}{\mathcal{I}_\kappa} \sum_{\nu=1}^{\Xi} \delta_{\nu,\kappa}(n) \eta_{\nu,\kappa} \bar{z}_\nu(n) \mathcal{I}_\nu, \quad \kappa \in [1:\Xi].$$

3. CE: Communicate $\Delta \bar{z}_\kappa = \bar{z}_\kappa^+ - \bar{z}$ to MG_κ, $\kappa \in [1:\Xi]$.
4. MG_κ: Update $\bar{w}_\kappa^+ = \bar{w}_\kappa + \Delta \bar{z}_\kappa$ and solve $\min_{\bar{z}_\kappa \in \mathbb{D}_\kappa^+} g_\kappa(\bar{z}_\kappa)$.

Results

MG	$\lVert \bar{z} - \bar{\zeta} \rVert_2^2$	$\lVert \bar{z}^+ - \bar{\zeta} \rVert_2^2$
1	920.57	451.44
2	124.85	85.07
3	32.25	79.82
4	34.03	85.07
overall	1111.70	701.39



Surrogate Models for Microgrids

(joint work with Sara Grundel and Manuel Baumann, MPI Magdeburg, [Grundel, S., Worthmann, 2019])

Surrogate Models for Microgrids

(joint work with Sara Grundel and Manuel Baumann, MPI Magdeburg, [Grundel, S., Worthmann, 2019])

Approach for single MGs

- Model Predictive Control (MPC)

$\rightsquigarrow$ solve optimization problem in each MPC iteration

Surrogate Models for Microgrids

(joint work with Sara Grundel and Manuel Baumann, MPI Magdeburg, [Grundel, S., Worthmann, 2019])

Approach for single MGs

- Model Predictive Control (MPC)

$\rightsquigarrow$ solve optimization problem in each MPC iteration

- Alternating Direction Method of Multipliers (ADMM)

$\rightsquigarrow$ time consuming + communication effort

Surrogate Models for Microgrids

(joint work with Sara Grundel and Manuel Baumann, MPI Magdeburg, [Grundel, S., Worthmann, 2019])

Approach for single MGs

- Model Predictive Control (MPC)

$\rightsquigarrow$ solve optimization problem in each MPC iteration

- Alternating Direction Method of Multipliers (ADMM)

$\rightsquigarrow$ time consuming + communication effort

Surrogate Models

$$\begin{pmatrix} x_1(k) \ldots x_{\mathcal{I}}(k) \\ \bar{w}(k) \ldots \bar{w}(k+N-1) \\ \bar{\zeta}(k) \ldots \bar{\zeta}(k+N-1) \end{pmatrix} \overset{\text{ADMM}}{\rightsquigarrow} \quad (\bar{z}(k) \ldots \bar{z}(k+N-1))$$

Surrogate Models for Microgrids

(joint work with Sara Grundel and Manuel Baumann, MPI Magdeburg, [Grundel, S., Worthmann, 2019])

Approach for single MGs

- Model Predictive Control (MPC)

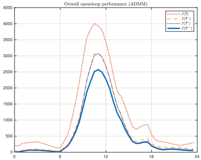
$\rightsquigarrow$ solve optimization problem in each MPC iteration

- Alternating Direction Method of Multipliers (ADMM)

$\rightsquigarrow$ time consuming + communication effort

Surrogate Models

$$\begin{pmatrix} x_1(k) \dots x_{\mathcal{I}}(k) \\ \bar{w}(k) \dots \bar{w}(k+N-1) \\ \bar{\zeta}(k) \dots \bar{\zeta}(k+N-1) \end{pmatrix} \overset{\text{ADMM}}{\rightsquigarrow} (\bar{z}(k) \dots \bar{z}(k+N-1))$$



Surrogate Models for Microgrids

(joint work with Sara Grundel and Manuel Baumann, MPI Magdeburg, [Grundel, S., Worthmann, 2019])

Approach for single MGs

- Model Predictive Control (MPC)

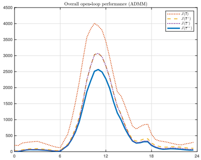
$\rightsquigarrow$ solve optimization problem in each MPC iteration

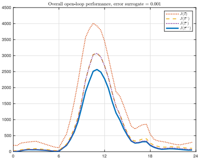
- Alternating Direction Method of Multipliers (ADMM)

$\rightsquigarrow$ time consuming + communication effort

Surrogate Models

$$\begin{pmatrix} x_1(k) \dots x_{\mathcal{I}}(k) \\ \bar{w}(k) \dots \bar{w}(k+N-1) \\ \bar{\zeta}(k) \dots \bar{\zeta}(k+N-1) \end{pmatrix} \overset{\text{ADMM}}{\rightsquigarrow} (\bar{z}(k) \dots \bar{z}(k+N-1))$$





Surrogate Models for Microgrids

(joint work with Sara Grundel and Manuel Baumann, MPI Magdeburg, [Grundel, S., Worthmann, 2019])

Approach for single MGs

- Model Predictive Control (MPC)

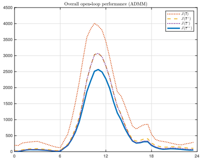
$\rightsquigarrow$ solve optimization problem in each MPC iteration

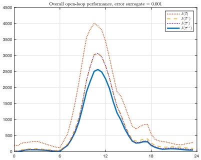
- Alternating Direction Method of Multipliers (ADMM)

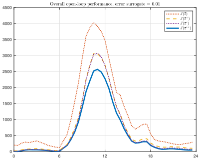
$\rightsquigarrow$ time consuming + communication effort

Surrogate Models

$$\begin{pmatrix} x_1(k)\dots x_{\mathcal{I}}(k) \\ \bar{w}(k)\dots\bar{w}(k+N-1) \\ \bar{\zeta}(k)\dots\bar{\zeta}(k+N-1) \end{pmatrix} \overset{\text{ADMM}}{\rightsquigarrow} \quad (\bar{z}(k)\dots\bar{z}(k+N-1))$$







Surrogate Models for Microgrids

(joint work with Sara Grundel and Manuel Baumann, MPI Magdeburg, [Grundel, S., Worthmann, 2019])

Approach for single MGs

- Model Predictive Control (MPC)

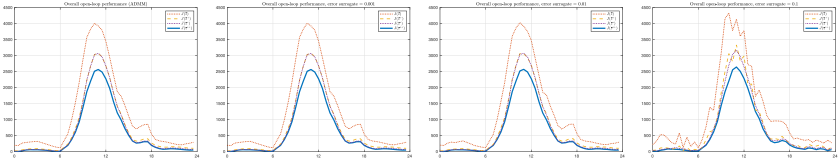
$\rightsquigarrow$ solve optimization problem in each MPC iteration

- Alternating Direction Method of Multipliers (ADMM)

$\rightsquigarrow$ time consuming + communication effort

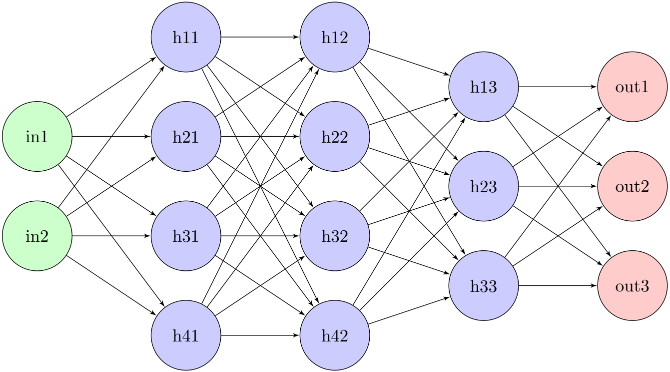
Surrogate Models

$$\begin{pmatrix} x_1(k) \dots x_{\mathcal{I}}(k) \\ \bar{w}(k) \dots \bar{w}(k+N-1) \\ \bar{\zeta}(k) \dots \bar{\zeta}(k+N-1) \end{pmatrix} \overset{\text{ADMM}}{\rightsquigarrow} (\bar{z}(k) \dots \bar{z}(k+N-1))$$



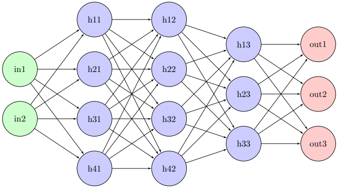
Introduction to Neural Networks

Basic Scheme incorporating 5 layers



- 1 input layer with 2 neurons
- 1 output layer with 3 neurons
- 3 hidden layers with 3 or 4 neurons, resp.

Introduction to Neural Networks



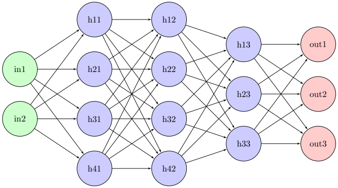
- sigmoid function $\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}$
- weights $W^{[\ell+1]} \in \mathbb{R}^{m_{\ell+1} \times m_\ell}$
- biases $b^{[\ell+1]} \in \mathbb{R}^{m_{\ell+1}}$

with number m_ℓ of neutrons in layer ℓ.

Output of layer $\ell + 1$ given output y^ℓ of previous layer

$$y^{\ell+1} = \sigma(W^{[\ell+1]} y^\ell + b^{[\ell+1]})$$

Introduction to Neural Networks



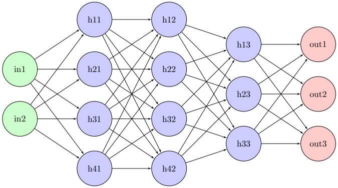
- sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$
- weights $W^{[\ell+1]} \in \mathbb{R}^{m_{\ell+1} \times m_\ell}$
- biases $b^{[\ell+1]} \in \mathbb{R}^{m_{\ell+1}}$

with number m_ℓ of neutrons in layer ℓ.

Output of neural network given input x

$$z = F(x) = \sigma(W^{[4]}\sigma(W^{[3]}\sigma(W^{[2]}x + b^{[2]}) + b^{[3]}) + b^{[4]})$$

Introduction to Neural Networks



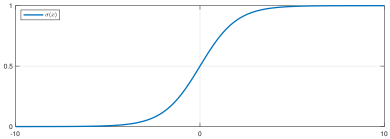
- sigmoid function $\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}$
- weights $W^{[\ell+1]} \in \mathbb{R}^{m_{\ell+1} \times m_\ell}$
- biases $b^{[\ell+1]} \in \mathbb{R}^{m_{\ell+1}}$

with number m_ℓ of neutrons in layer ℓ.

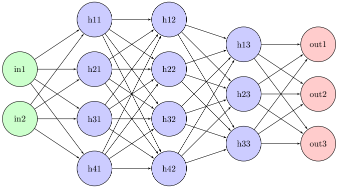
Output of neural network given input x

$$z = F(x) = \sigma(W^{[4]}\sigma(W^{[3]}\sigma(W^{[2]}x + b^{[2]}) + b^{[3]}) + b^{[4]})$$

Sigmoid function



Introduction to Neural Networks



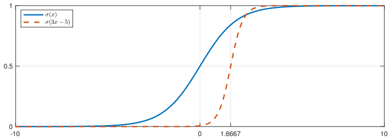
- sigmoid function $\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}$
- weights $W^{[\ell+1]} \in \mathbb{R}^{m_{\ell+1} \times m_\ell}$
- biases $b^{[\ell+1]} \in \mathbb{R}^{m_{\ell+1}}$

with number m_ℓ of neutrons in layer ℓ.

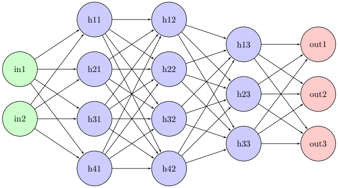
Output of neural network given input x

$$z = F(x) = \sigma(W^{[4]}\sigma(W^{[3]}\sigma(W^{[2]}x + b^{[2]}) + b^{[3]}) + b^{[4]})$$

Sigmoid function



Introduction to Neural Networks



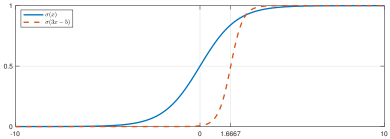
- sigmoid function $\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}$
- weights $W^{[\ell+1]} \in \mathbb{R}^{m_{\ell+1} \times m_\ell}$
- biases $b^{[\ell+1]} \in \mathbb{R}^{m_{\ell+1}}$

with number m_ℓ of neutrons in layer ℓ.

Output of neural network given input x

$$z = F(x) = \sigma(W^{[4]}\sigma(W^{[3]}\sigma(W^{[2]}x + b^{[2]}) + b^{[3]}) + b^{[4]})$$

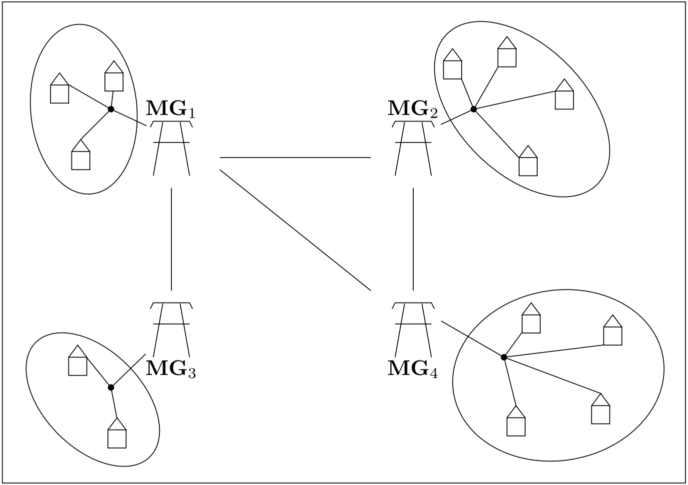
Sigmoid function



- "smoothed" step function
- imitating neurons in brain:
  - $\sigma = 1 \Leftrightarrow$ neuron firing
  - $\sigma = 0 \Leftrightarrow$ neuron inactive
- [Higham, Higham, 2018]

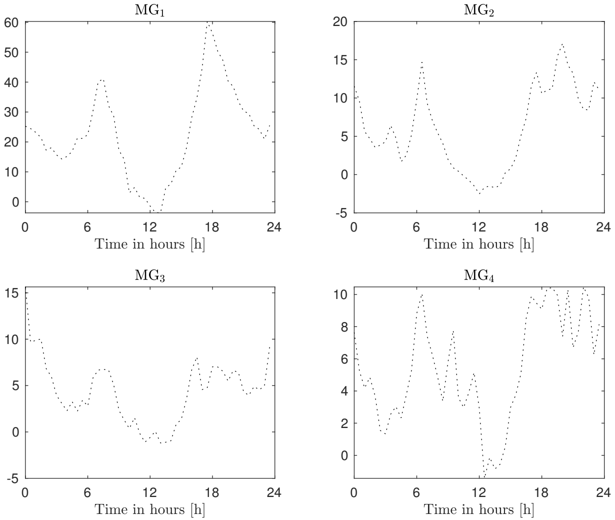
Results

Coupled MGs



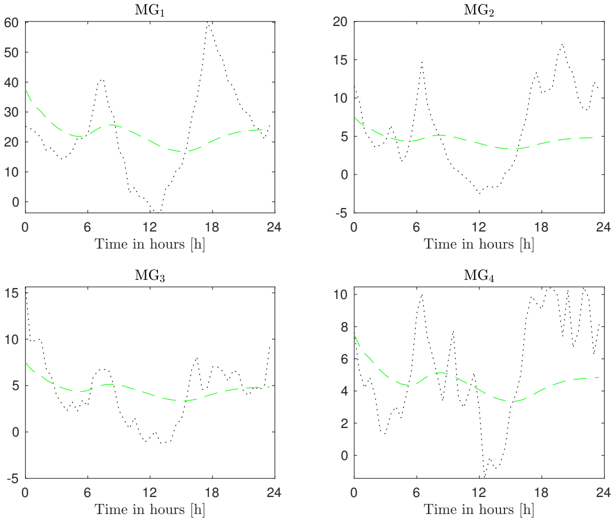
Results

Net consumption



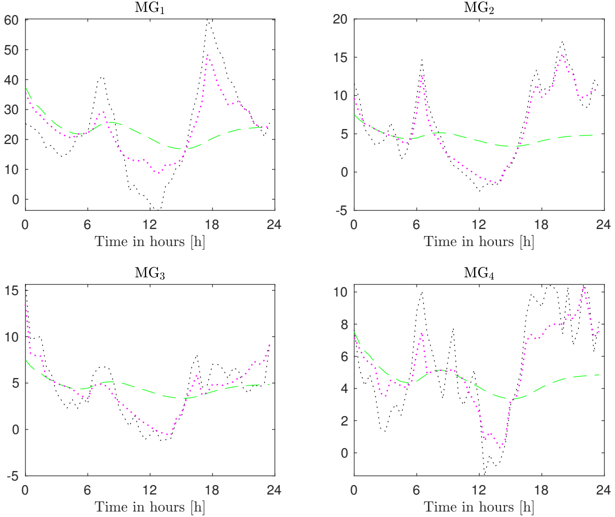
Results

Reference trajectory



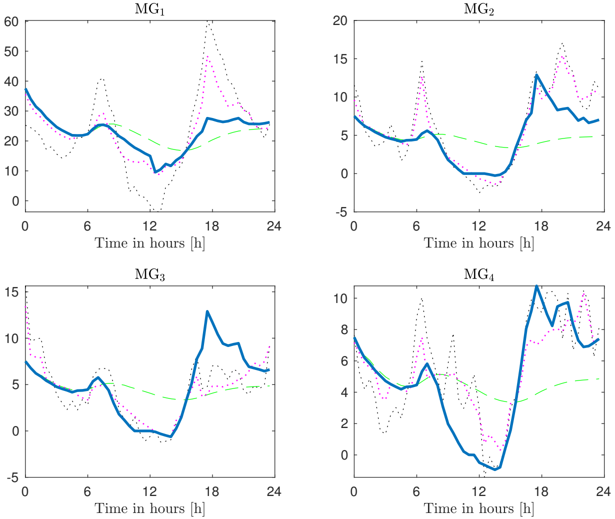
Results

Solution without additional exchange



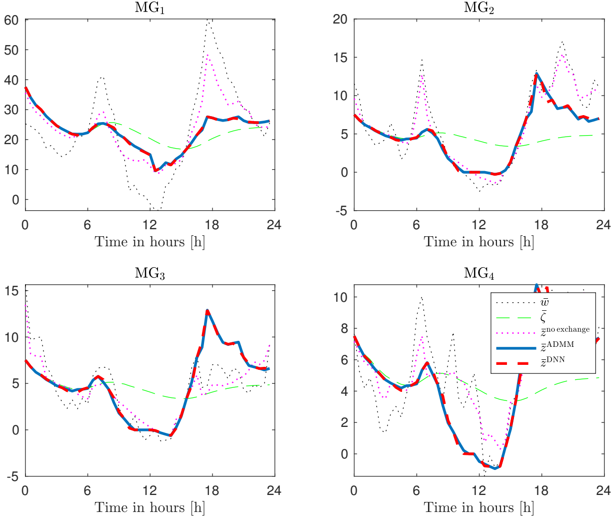
Results

Solution incorporating exchange within MGs



Results

Approximation (using MatLab's Neural Network)



Thank you for your attention!

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, *Foundations and Trends in Machine Learning*, 3(**1**), 1-122, 2011

[2] P. Braun, T. Faulwasser, L. Grüne, C. M. Kellett, S. R. Weller and K. Worthmann. Hierarchical Distributed ADMM for Predictive Control with Applications in Power Networks, *IFAC Journal of Systems and Control*, **3**, 10-22, 2018

[3] P. Braun, P. Sauerteig, K. Worthmann. Distributed optimization based control on the example of microgrids, *Computational Intelligence and Optimization Methods for Control Engineering*, 2019 (accepted)

[4] M. Ehrgott. Multicriteria Optimization, Springer, 2005

[5] S. Grundel, P. Sauerteig, K. Worthmann. Surrogate Models for Coupled Microgrids, *Progress in Industrial Mathematics at ECMI 2018*, 2019 (accepted)

[6] C. F. Higham, D. J. Higham. Deep Learning: An Introduction for Applied Mathematicians, 2018

[7] B. Houska, J. Frasch, M. Diehl. An Augmented Lagrangian Based Algorithm for Distributed Non-Convex Optimization, *SIAM Journal on Optimization*, 26(**2**), 1101-1127, 2016

[8] P. Sauerteig, K. Worthmann. Towards Multiobjective Optimization and Control of Smart Grids, *Optimal Control: Applications and Methods*, 2019 (submitted)

Appendix

Extension: Controllable Loads

Assumptions

- Net consumption is split

$$w_i(n) = w_i^s(n) + w_i^c(n)$$

  with static part $w_i^s(n)$ and controllable part $w_i^c(n)$.
- $w_i^c(n)$ is scheduled within time window $[n : n + \bar{N}]$ with $\bar{N} \in \mathbb{N}$
- $w_i^c(n) \in [0, \bar{w}_i^c]$ with $\bar{w}_i^c \geq 0$

Extension: Controllable Loads

Assumptions

- Net consumption is split

$$w_i(n) = w_i^s(n) + w_i^c(n)$$

  with static part $w_i^s(n)$ and controllable part $w_i^c(n)$.
- $w_i^c(n)$ is scheduled within time window $[n : n + \bar{N}]$ with $\bar{N} \in \mathbb{N}$
- $w_i^c(n) \in [0, \bar{w}_i^c]$ with $\bar{w}_i^c \geq 0$

System Dynamics

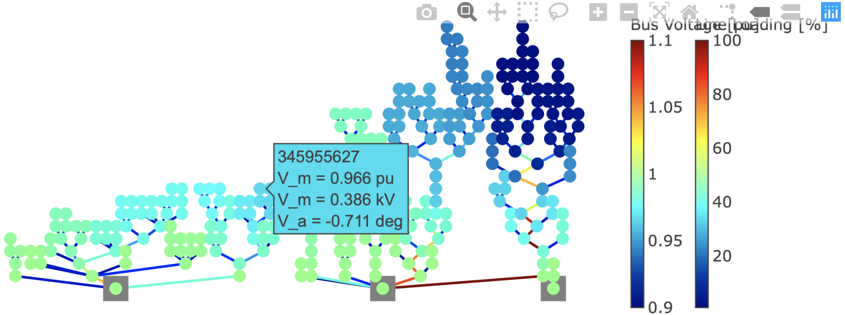
$$\begin{aligned} x_i(n+1) &= \alpha_i x_i(n) + T(\beta_i u_i^+(n) + u_i^-(n)) \\ z_i(n) &= w_i^s(n) + u_i^+(n) + \gamma_i u_i^-(n) + u_i^c(n) \end{aligned}$$

Towards Optimal Line Line Loading

Line Loading of a Test Grid

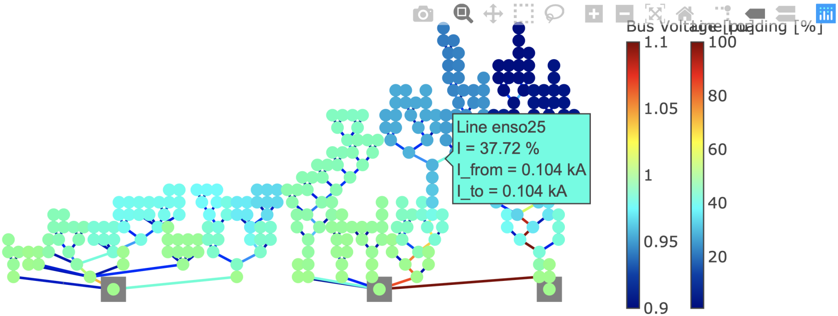
Towards Optimal Line Line Loading

Line Loading of a Test Grid



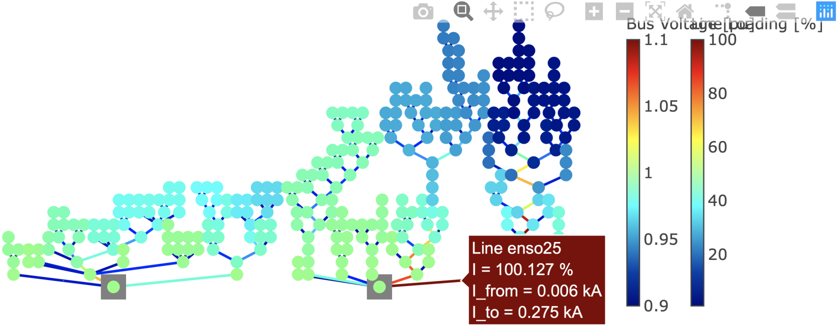
Towards Optimal Line Line Loading

Line Loading of a Test Grid



Towards Optimal Line Line Loading

Line Loading of a Test Grid



created with `python` toolbox `PandaPower`

Towards Optimal Line Loading

Optimization Problem

$$
\begin{aligned}
\min_{u,Q} \quad & \|s\|_2^2 \\
\text{s.t.} \quad & P_i - |V_i| \sum_{j \neq i} |V_j| (G_{ij} \cos(\delta_{ij}) + B_{ij} \sin(\delta_{ij})) = 0 \\
& Q_i - |V_i| \sum_{j \neq i} |V_j| (G_{ij} \sin(\delta_{ij}) - B_{ij} \cos(\delta_{ij})) = 0 \\
& x_i(k+1) = \alpha_i x_i(k) + T(\beta_i u_i^+(k) + u_i^-(k)) \\
& P_i(k) = z_i(k) = w_i(k) + u_i^+(k) + \gamma_i u_i^-(k) \\
& \underline{c} - \underline{s} \leq \mathcal{J}(|\mathbf{V}|, \delta) \leq \bar{c} + \bar{s} \\
& \text{constraints of each subsystem } i
\end{aligned}
$$

- line loading $\mathcal{J}$: function of voltage $\mathbf{V} = \left(|V_1|\mathrm{e}^{\mathrm{i}\delta_1}, \dots, |V_{\mathcal{I}}|\mathrm{e}^{\mathrm{i}\delta_{\mathcal{I}}}\right)$
- admittance matrix $Y_{ij} = G_{ij} + \mathrm{i}B_{ij}$ encodes grid topology
- tube constraints $\underline{c} < \bar{c}$