Control of Distributed Energy Resources: PDEs and Hopfield Methods

Scott Moura

Assistant Professor | eCAL Director
University of California, Berkeley

caesars2018
Advances in Modelling and Control for Power Systems of the Future

eCAL
energy, controls, & applications lab


Dr. Hector PEREZ


Dr. Hongcai ZHANG


Dr. Milad MEMARZADEH


Laurel DUNN


Zach GIMA


Saehong PARK


Dong ZHANG


Victoria CHENG


Sangjae BAE


Bertrand TRAVACCA


Mathilde BADOUAL


Zhe ZHOU


Tianyu HU


Tianyu YANG


Ramon CRESPO


Armando DOMINGOS


Soomin WOO


Yiqi ZHAO


Dylan KATO


Teng ZENG


Patrick KEYANTUO

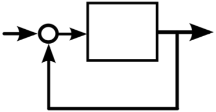

Aaron KANDEL


Pierre-François MASSIANI

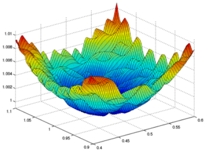

Emily YOU

eCAL
energy, controls, & applications lab

FUNDAMENTAL RESEARCH



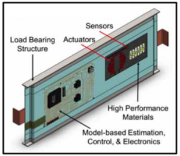
Dynamic Systems & Control



Optimization

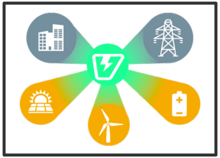
Data Science

APPLICATIONS



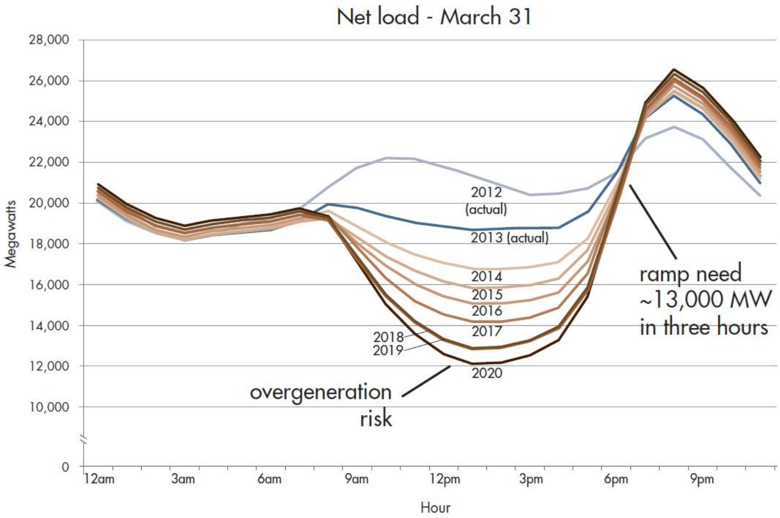
Battery Management Systems

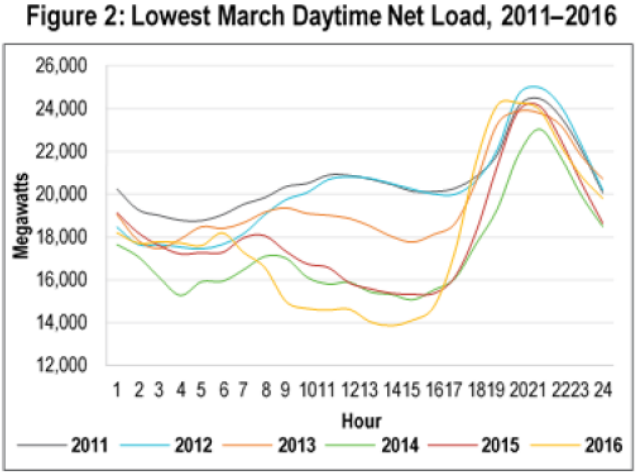
Automated, Connected, & Electric Vehicles



Distributed Energy Resources

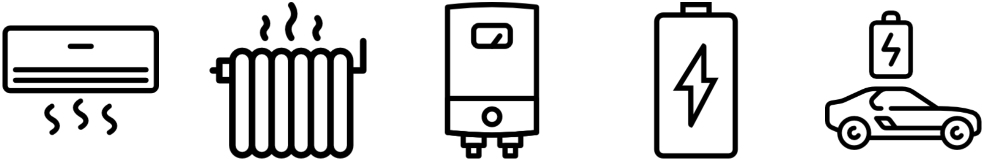
The duck curve shows steep ramping needs and overgeneration risk





Source: C. Vlahoplus, G. Litra, P. Quinlan, C. Becker, "Revising the California Duck Curve: An Exploration of Its Existence, Impact, and Migration Potential," *Scott Madden, Inc.*, Oct 2016.

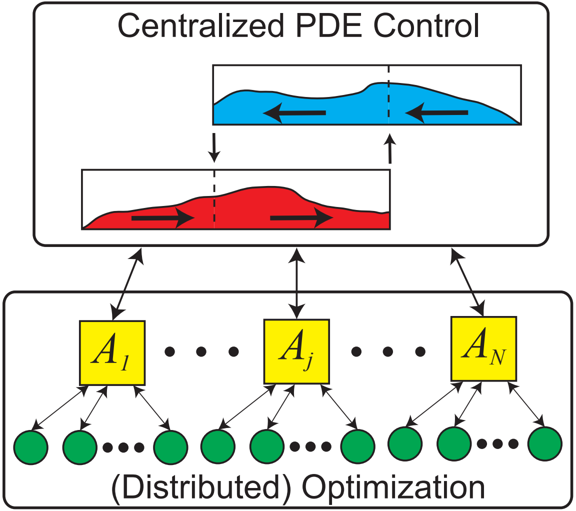
Aggregate Flexible Loads into Virtual Power Plant



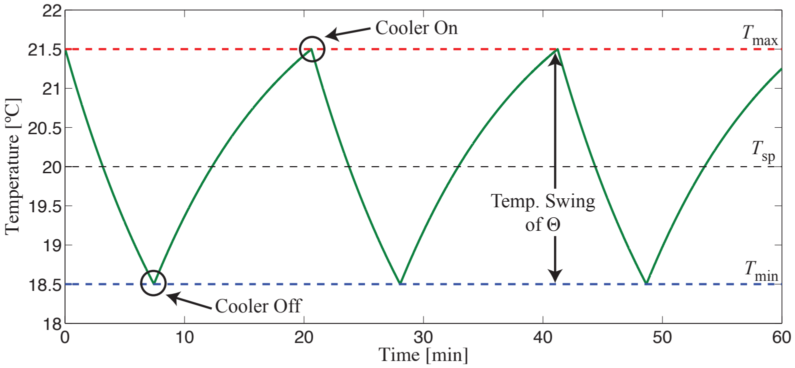
Air Conditioning | **Heater** | **Water Heater** | **Air Conditioning** | **Plug-in Electric Vehicles**

- Chen, Hashmi, Mathias, Busic, Meyn (2018)

Hierarchical Control



Modeling Thermostatically Controlled Loads (TCLs)



$$\dot{T}_i(t) = \frac{1}{R_i C_i}\left[T_\infty - T_i(t) - s_i(t) R_i P_i\right], \quad i = 1, 2, \ldots, N$$

$$s_i \in \{0, 1\}$$

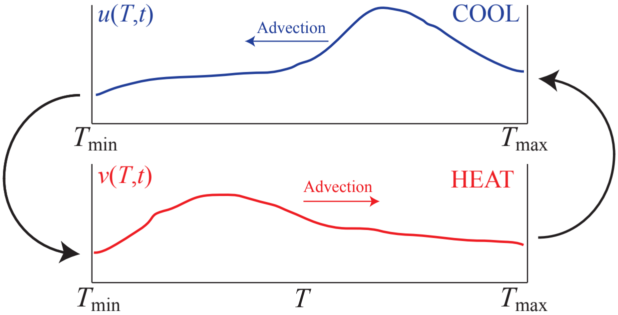
Modeling Aggregated TCLs

Main Idea: Convert $> 10^3$ ODEs into two coupled linear PDEs (Malhamé and Chong, 1985)

Modeling Aggregated TCLs

Main Idea: Convert $> 10^3$ ODEs into two coupled linear PDEs (Malhamé and Chong, 1985)

$u(T,t)$	# TCLs / °C, in COOL state, @ temp T, time t
$v(T,t)$	# TCLs / °C, in HEAT state, @ temp T, time t



Modeling Aggregated TCLs

Main Idea: Convert $> 10^3$ ODEs into two coupled linear PDEs (Malhamé and Chong, 1985)

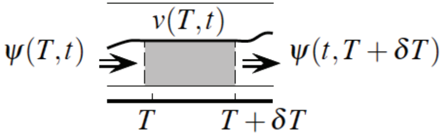
$u(T,t)$ | # TCLs / °C, in COOL state, @ temp T, time t

$v(T,t)$ | # TCLs / °C, in HEAT state, @ temp T, time t

Flux of TCLs in HEAT state:
#TCLs / sec

$$\psi(T,t) = v(T,t)\frac{dT}{dt}(t) = \frac{1}{RC}\left[T_\infty - T(t)\right] v(T,t)$$

Control volume:



Modeling Aggregated TCLs

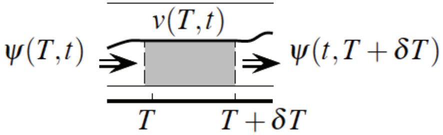
Main Idea: Convert $> 10^3$ ODEs into two coupled linear PDEs (Malhamé and Chong, 1985)

$u(T,t)$ | # TCLs / °C, in COOL state, @ temp T, time t
$v(T,t)$ | # TCLs / °C, in HEAT state, @ temp T, time t

Flux of TCLs in HEAT state:
#TCLs / sec

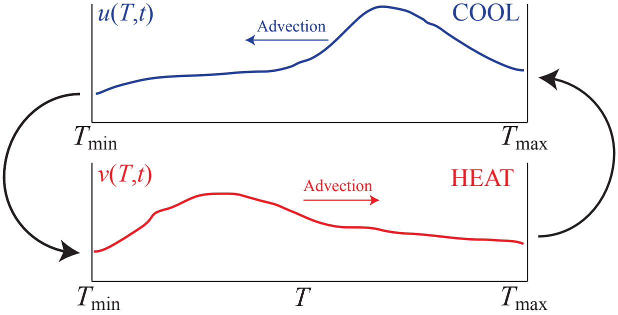
$$\psi(T,t) = v(T,t)\frac{dT}{dt}(t) = \frac{1}{RC}\left[T_\infty - T(t)\right] v(T,t)$$

Control volume:

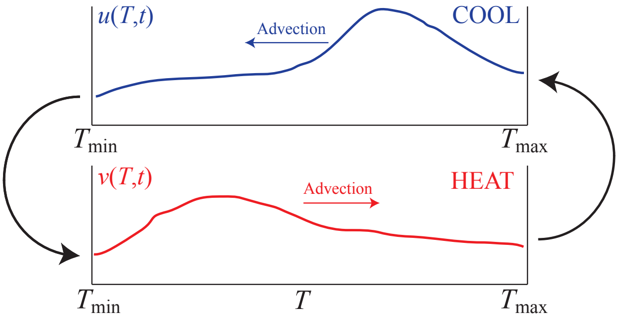


$$\begin{aligned}\frac{\partial v}{\partial t}(T,t) &= \lim_{\delta T \to 0}\left[\frac{\psi(T+\delta T,t) - \psi(T,t)}{\delta T}\right] \\ &= \frac{\partial \psi}{\partial T}(T,t) \\ &= -\frac{1}{RC}\left[T_\infty - T(t)\right]\frac{\partial v}{\partial T}(T,t) + \frac{1}{RC} v(T,t)\end{aligned}$$

PDE Model of Aggregated TCLs

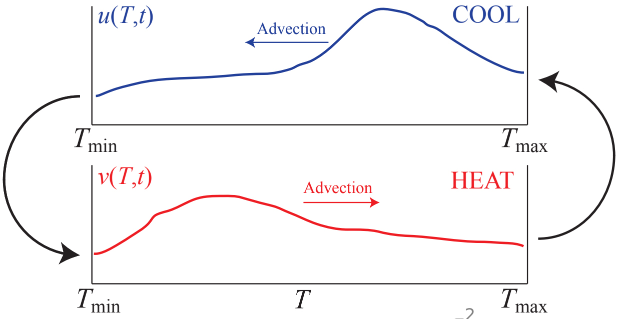


PDE Model of Aggregated TCLs



$$u_t(T,t) = \alpha\lambda(T)u_T(T,t) + \alpha u(T,t)$$
$$v_t(T,t) = -\alpha\mu(T)v_T(T,t) + \alpha v(T,t)$$
$$u(T_{\max},t) = q_1 v(T_{\max},t)$$
$$v(T_{\min},t) = q_2 u(T_{\min},t)$$

PDE Model of Aggregated TCLs



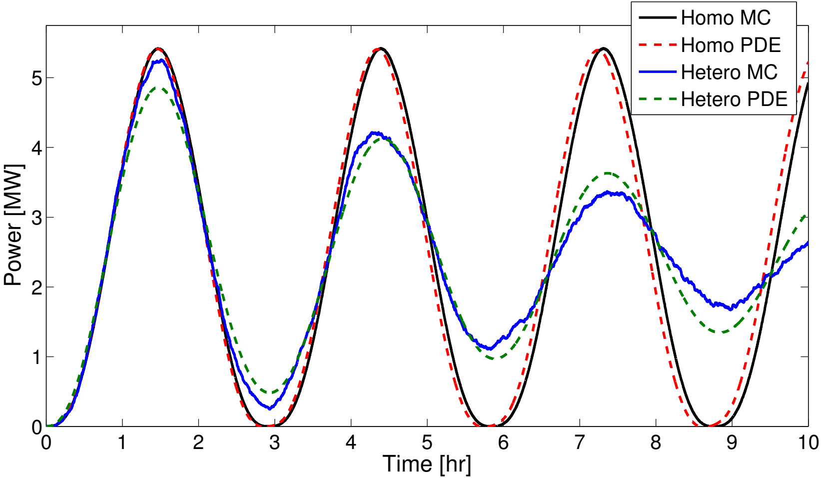
$$u_t(T,t) = \alpha\lambda(T) u_T(T,t) + \alpha u(T,t) + \frac{\sigma^2}{2} u_{TT}(T,t)$$

$$v_t(T,t) = -\alpha\mu(T) v_T(T,t) + \alpha v(T,t) + \frac{\sigma^2}{2} v_{TT}(T,t)$$

$$u(T_{\max},t) = q_1 v(T_{\max},t) \qquad u_T(T_{\min},t) = -v_T(T_{\min},t)$$

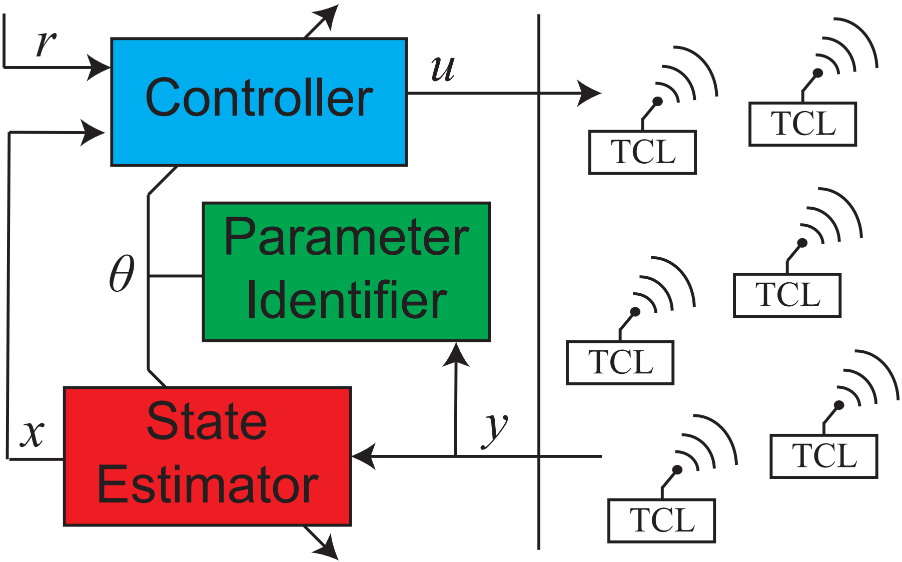
$$v(T_{\min},t) = q_2 u(T_{\min},t) \qquad v_T(T_{\max},t) = -u_T(T_{\max},t)$$

Model Comparison



S. J. Moura, J. Bendsten, V. Ruiz, "Parameter Identification of Aggregated Thermostatically Controlled Loads for Smart Grids using PDE Techniques," International Journal of Control, 2014 (Invited Paper) DOI: 10.1080/00207179.2014.915083.

Feedback Control System



Question:

Given low-bandwidth communication, can we estimate the states of the TCL population in real-time?

PDE State Estimator

Heterogeneous PDE Model: (u, v)

$$\begin{aligned} u_t(x,t) &= \alpha\lambda(x)u_x + \alpha u + \beta u_{xx} \\ u_x(0,t) &= -v_x(0,t) \\ u(1,t) &= q_1 v(1,t) \end{aligned}$$

$$\begin{aligned} v_t(x,t) &= -\alpha\mu(x)v_x + \alpha v + \beta v_{xx} \\ v(0,t) &= q_2 u(0,t) \\ v_x(1,t) &= -u_x(1,t) \end{aligned}$$

Measurements? Send signal only when switching ON/OFF

- $u(0,t), v(1,t)$
- $u_x(1,t), v_x(0,t)$

PDE State Estimator

Estimator: $(\hat{u}, \hat{v})$

$$
\begin{aligned}
\hat{u}_t(x,t) &= \alpha\lambda(x)\hat{u}_x + \alpha\hat{u} + \beta\hat{u}_{xx} + p_1(x)\left[u(0,t) - \hat{u}(0,t)\right] \\
\hat{u}_x(0,t) &= -v_x(0,t) + p_{10}\left[u(0,t) - \hat{u}(0,t)\right] \\
\hat{u}(1,t) &= q_1 v(1,t)
\end{aligned}
$$

$$
\begin{aligned}
\hat{v}_t(x,t) &= -\alpha\mu(x)\hat{v}_x + \alpha\hat{v} + \beta\hat{v}_{xx} + p_2(x)\left[v(1,t) - \hat{v}(1,t)\right] \\
\hat{v}(0,t) &= q_2 u(0,t) \\
\hat{v}_x(1,t) &= -u_x(1,t) + p_{20}\left[v(1,t) - \hat{v}(1,t)\right]
\end{aligned}
$$

PDE State Estimator

Estimation Error Dynamics: $(\tilde{u}, \tilde{v}) = (u - \hat{u}, v - \hat{v})$

$$
\begin{aligned}
\tilde{u}_t(x,t) &= \alpha\lambda(x)\tilde{u}_x + \alpha\tilde{u} + \beta\tilde{u}_{xx} - p_1(x)\tilde{u}(0,t) \\
\tilde{u}_x(0,t) &= -p_{10}\tilde{u}(0,t) \\
\tilde{u}(1,t) &= 0
\end{aligned}
$$

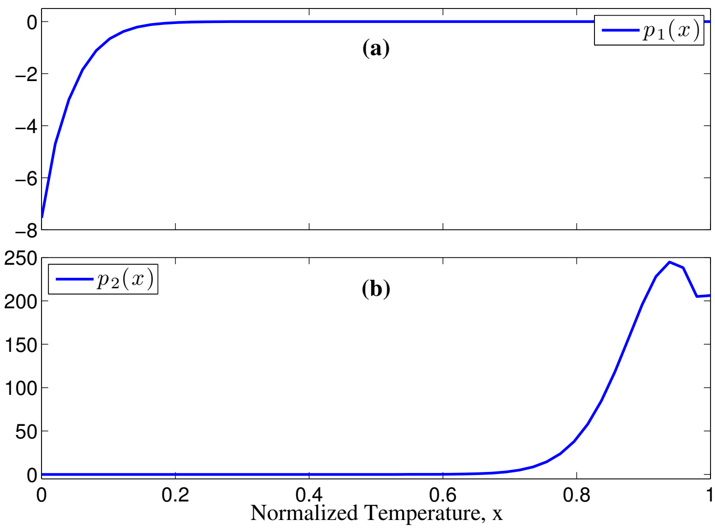
$$
\begin{aligned}
\tilde{v}_t(x,t) &= -\alpha\mu(x)\tilde{v}_x + \alpha\tilde{v} + \beta\tilde{v}_{xx} - p_2(x)\tilde{v}(1,t) \\
\tilde{v}(0,t) &= 0 \\
\tilde{v}_x(1,t) &= -p_{20}\tilde{v}(1,t)
\end{aligned}
$$

Goal: Design estimation gains:

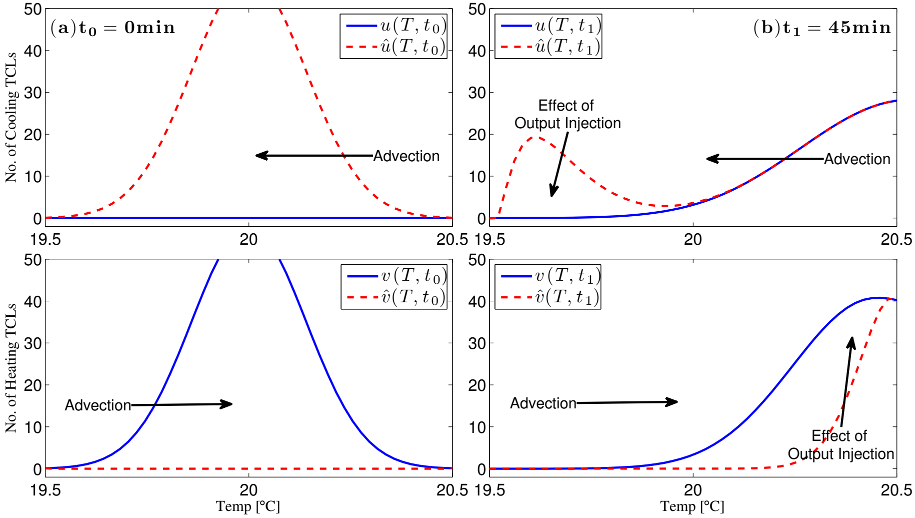
- $p_1(x), p_2(x) : (0,1) \to \mathbb{R}$
- $p_{10}, p_{20} \in \mathbb{R}$

such that $(\tilde{u}, \tilde{v}) = (0, 0)$ is exponentially stable

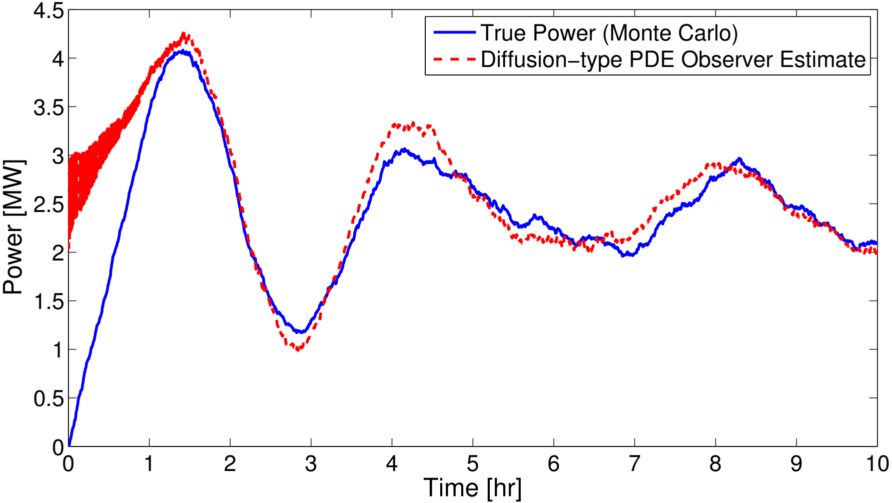
Observer Gains



Simulations

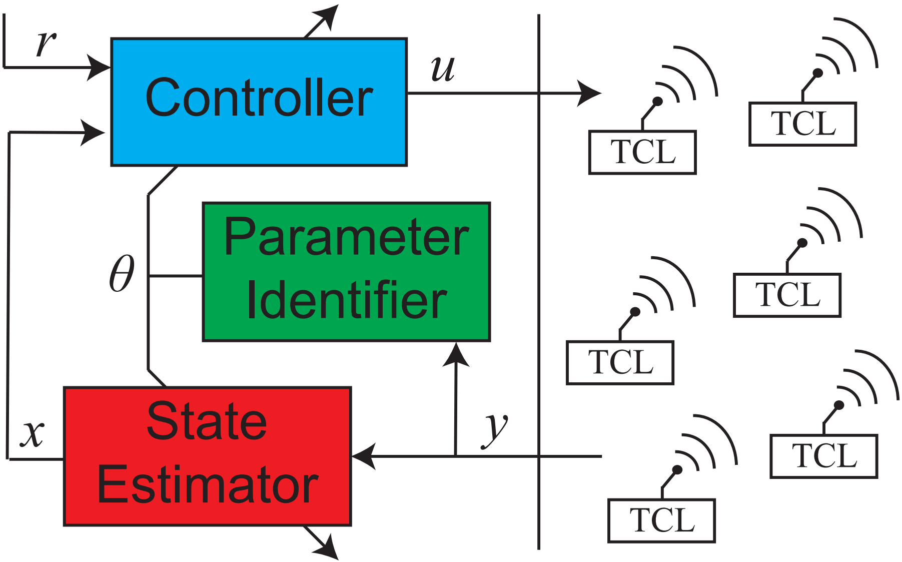


Simulations



S. J. Moura, J. Bendsten, V. Ruiz, "Observer Design for Boundary Coupled PDEs: Application to Thermostatically Controlled Loads in Smart Grids," IEEE Conf. on Decision and Control, Florence, Italy, 2013.

Feedback Control System



Question:

Can we learn uncertain model parameters online?

Parameter Identification

Uncertain parameters

$$u_t(x,t) = \alpha\lambda(x)u_x + \alpha u + \beta u_{xx} \qquad v_t(x,t) = -\alpha\mu(x)v_x + \alpha v + \beta v_{xx}$$

$$u_x(0,t) = -v_x(0,t) \qquad v(0,t) = q_2 u(0,t)$$

$$u(1,t) = q_1 v(1,t) \qquad v_x(1,t) = -u_x(1,t)$$

$$P(t) = \frac{\overline{P}}{\eta}\int_0^1 u(x,t)dx$$

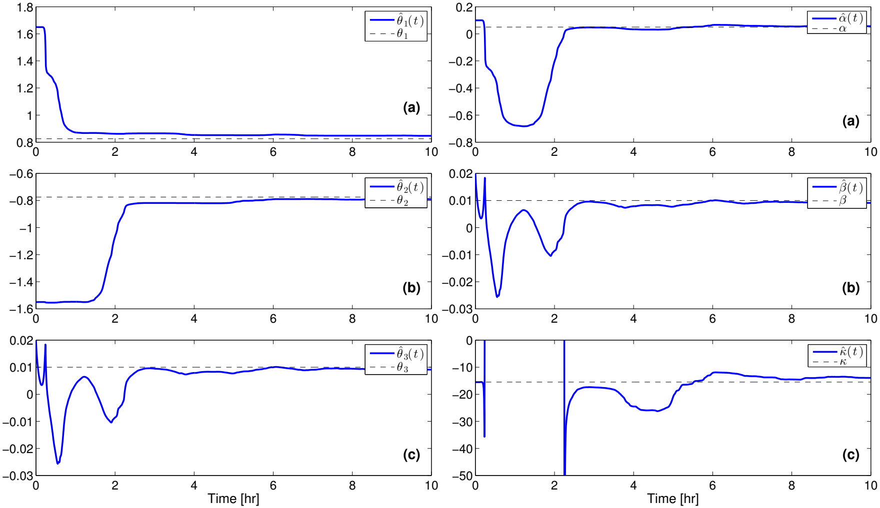
Assumptions:

1. Aggregate Power $P(t)$ is measured
2. No. of TCLs switching $u(0,t), u(1,t), u_x(0,t), u_x(1,t)$ is measured

Approach:

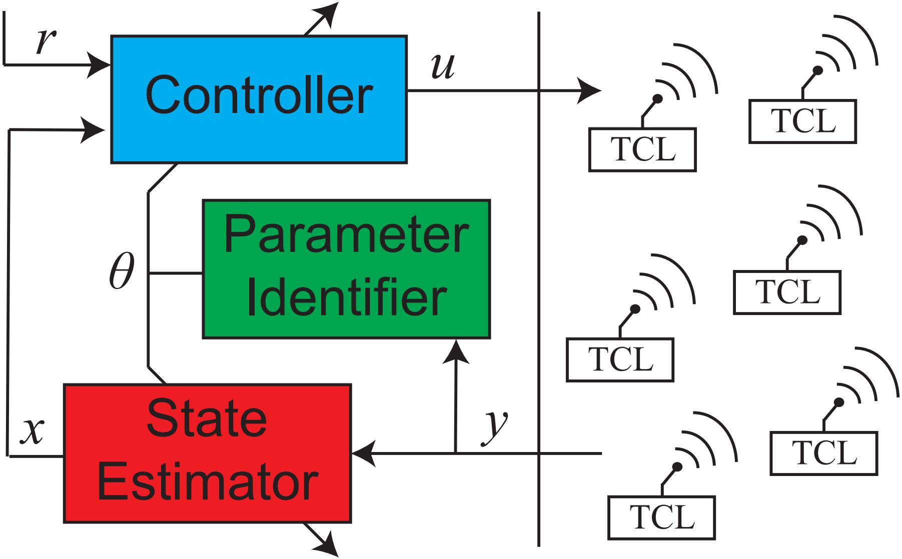
1. Form linear-in-the-parameters regression model
2. Use filters to avoid differentiation
3. Apply recursive least squares

Simulations: Params ID'ed from 1,000 heterogeneous TCLs



S. J. Moura, V. Ruiz, J. Bendsten, "Modeling Heterogeneous Populations of Thermostatically Controlled Loads using Diffusion-Advection PDEs," ASME Dynamic Systems and Control Conference, Stanford, CA, 2013.

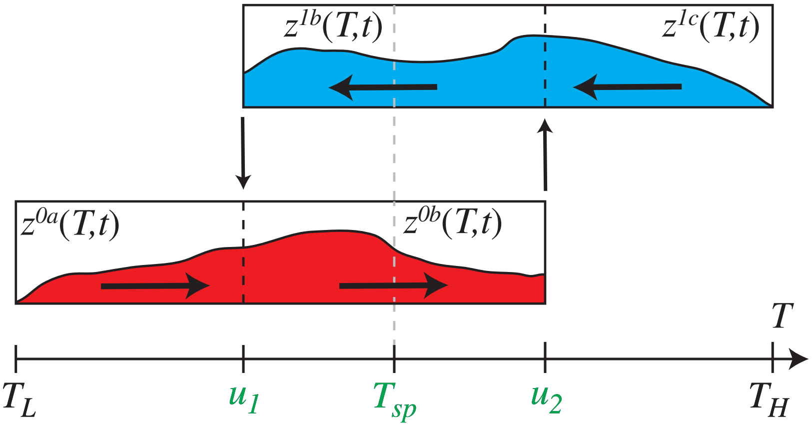
Feedback Control System



Question:

Great! We can monitor DER populations well. What about control?

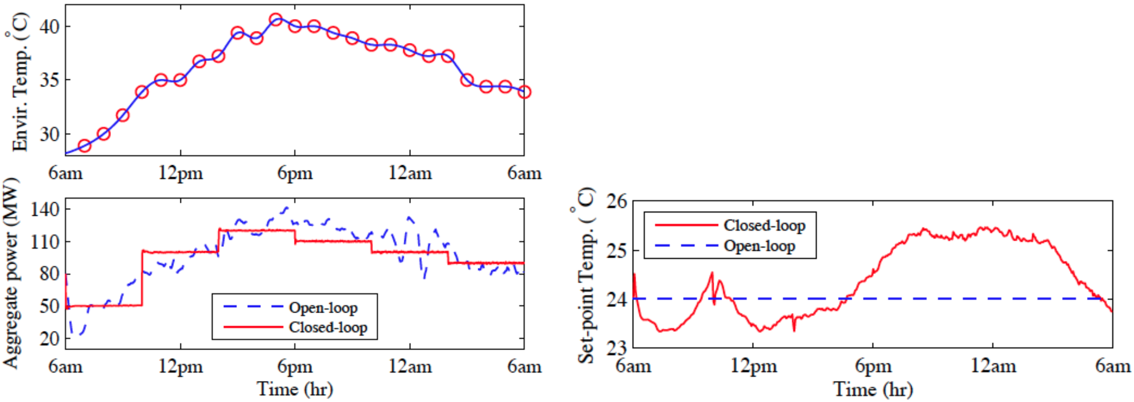
Set-point / Deadband Control



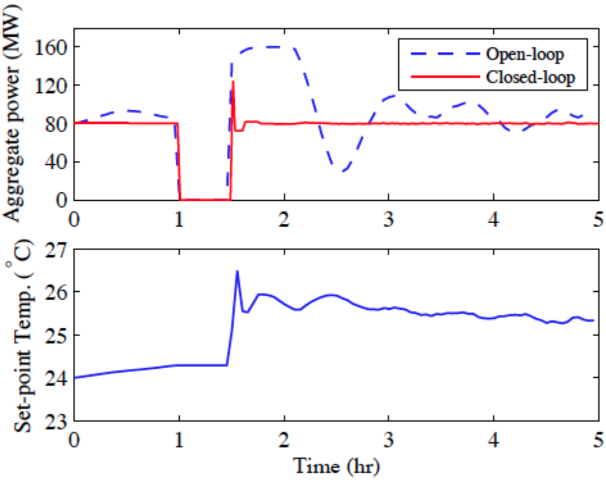
$$z_t^{1j}(T,t) = \alpha\lambda(T) z_T^{1j}(T,t) + \alpha z^{1j}(T,t), \qquad j \in \{b,c\}$$
$$z_t^{0j}(T,t) = -\alpha\mu(T) z_T^{0j}(T,t) + \alpha z^{0j}(T,t), \qquad j \in \{a,b\}$$

with boundary conditions

Aggregate Power Control



Aggregate Power Control



A. Ghaffari, S. J. Moura, M. Krstic, "Analytic Modeling and Integral Control of Heterogeneous Thermostatically Controlled Load Populations," ASME Dynamic Systems and Control Conference, San Antonio, TX, 2014.

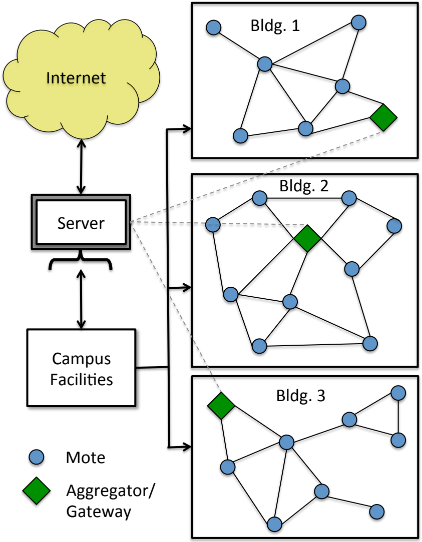
UC San Diego Campus: A Living Laboratory

UC San Diego Campus: A Living Laboratory

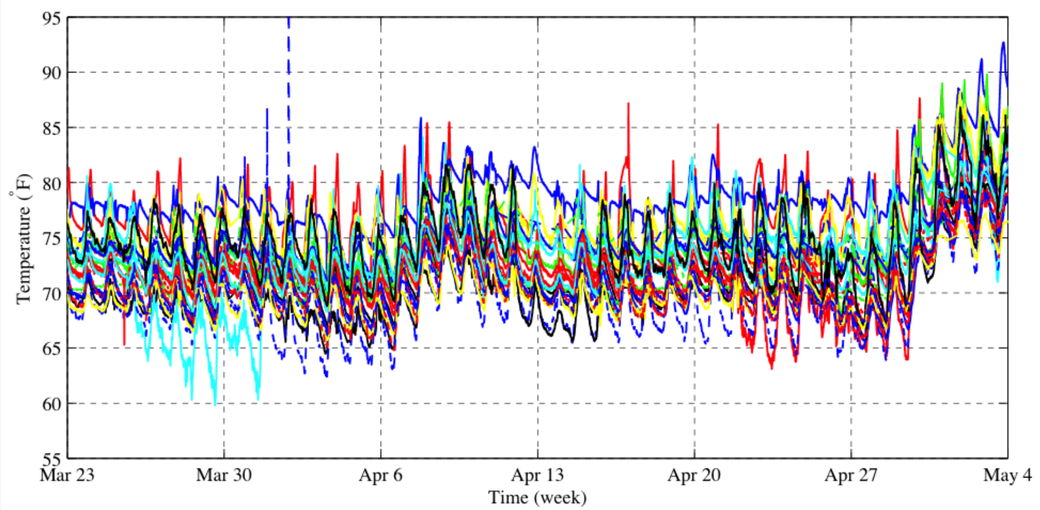
Goal: DR for Bldg Energy Mgmt

1. Deploy wireless sensor network
2. Model/estimator verification
3. Control design
4. Campus implementation

Sensor Nodes (Temp & Humidity)

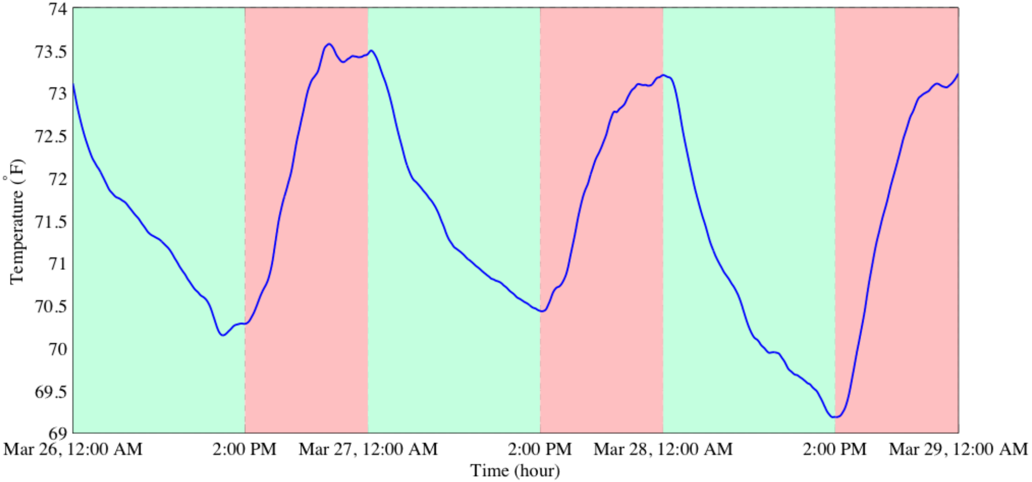


UCSD Office Temperature Data



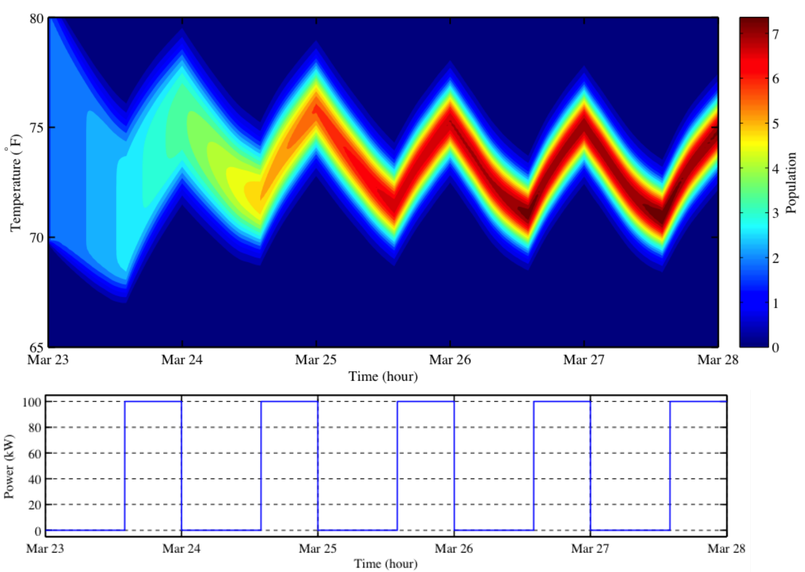
Temperature variation of over six weeks. Peak temperature happens at midnight!?

UCSD Office Temperature Data

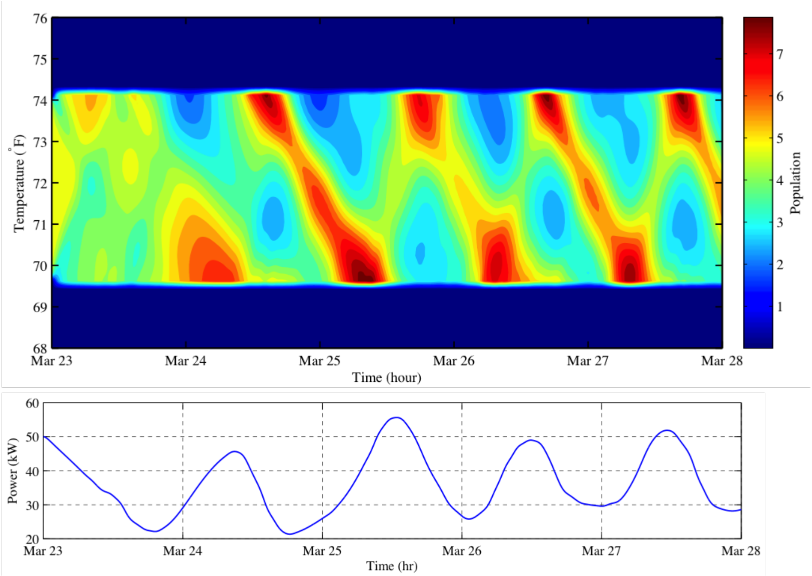


HVAC system is time scheduled. ON | OFF

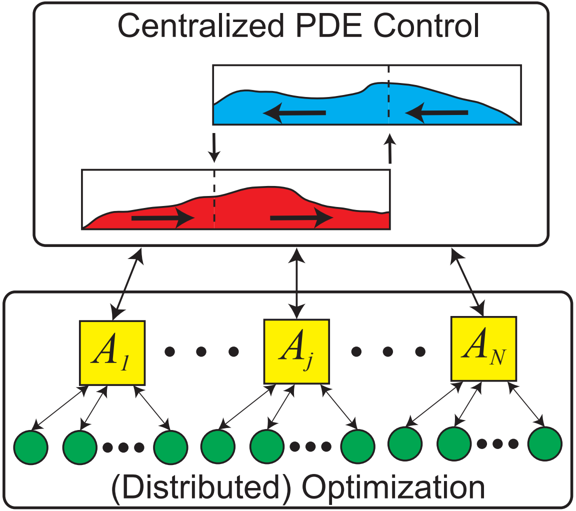
UCSD Office TCL Simulation - Existing Open Loop Control



Closed loop Control : 13% energy reduction



Hierarchical Control



PEV Charge Schedule Optimization is a MIP!

eMotorWerks' Juicebox

Control ∈ *on* (40 *A*)
or *off* (0 *A*)

UC Berkeley Smart EV Charger

Richmond Field Station

Control ∈ {0 *A*} ∪ [12 *A*, 30 *A*]

Problem Statement

Consider a mixed integer nonlinear program (MINLP):

$$\begin{aligned} \text{minimize} \quad & f(x) & (1)\\ \text{subject to:} \quad & g_j(x) \le 0, \quad j = 1, \cdots, m & (2)\\ & x_i \in \{0, 1\}, \quad i = 1, \cdots, p < n & (3)\\ & 0 \le x_i \le 1, \quad i = p+1, \cdots, n & (4) \end{aligned}$$

$x \in \mathbb{R}^n$ is the optimization variable

the first $p < n$ variables must be binary

$f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is quadratic and L_f – smooth

$g_j(\cdot) : \mathbb{R}^n \to \mathbb{R}$ are quadratic and L_j – smooth

Problem Statement

Consider a mixed integer nonlinear program (MINLP):

$$\begin{aligned} \text{minimize} \quad & f(x) && (1) \\ \text{subject to:} \quad & g_j(x) \leq 0, \quad j = 1, \cdots, m && (2) \\ & x_i \in \{0, 1\}, \quad i = 1, \cdots, p < n && (3) \\ & 0 \leq x_i \leq 1, \quad i = p+1, \cdots, n && (4) \end{aligned}$$

$x \in \mathbb{R}^n$ is the optimization variable
the first $p < n$ variables must be binary
$f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is quadratic and L_f – smooth
$g_j(\cdot) : \mathbb{R}^n \to \mathbb{R}$ are quadratic and L_j – smooth

Challenge

Solve **LARGE-SCALE** MINLPs, e.g. $n = 10^3, 10^4, 10^5, \cdots$

P vs NP – Millenium Prize Problem

CLAY MATHEMATICS INSTITUTE

Existing Convex Relaxation Methods

1. Binary relaxation
2. Lagrangian relaxation
3. Semi-definite relaxation

Existing Convex Relaxation Methods

1. Binary relaxation
2. Lagrangian relaxation
3. Semi-definite relaxation

Stochastic approach to recover integer constraint:

Let x^r be solution to binary relaxation. Feasible x can be drawn randomly from $\{0,1\}$ following Bernoulli distribution $\mathcal{B}(x^r)$.

This can be sub-optimal.

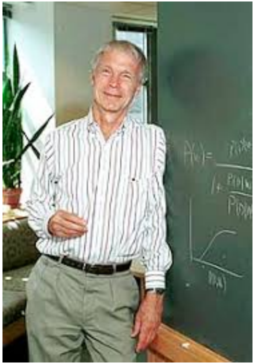
Example

$$\text{minimize}_{x\in\{0,1\}} \left(x - \frac{1}{4}\right)^2 = \frac{1}{16} \quad (x^\star = 0 \ \text{ is the optimal solution})$$

If we apply binary relaxation, we get $x^r = \frac{1}{4}$ and $\mathbb{E}_{x\sim\mathcal{B}(x^r)} \left(x - \frac{1}{4}\right)^2 = \frac{3}{16} > \frac{1}{16}$!

A short history of Hopfield Networks

- (1982) J. J. Hopfield used neural nets to model collaborative computations
- (1985) J. J. Hopfield showed that neural nets can be used to solve optimization problems
- (1990's) Hopfield methods became very popular for solving MIQPs in power systems optimization
- In literature, power system engineers admit they didn't fully understand *why* Hopfield methods work well.



The Hopfield Method

Consider MINLP

$$\begin{aligned} \text{minimize} \quad & f(x) && (5) \\ \text{subject to:} \quad & x_i \in \{0,1\}, \quad i = 1, \cdots, p < n && (6) \\ & 0 \le x_i \le 1, \quad i = p+1, \cdots, n && (7) \end{aligned}$$

The Hopfield Method

Consider MINLP

$$\begin{aligned} \text{minimize} \quad & f(x) && (5) \\ \text{subject to:} \quad & x_i \in \{0,1\}, \quad i = 1, \cdots, p < n && (6) \\ & 0 \le x_i \le 1, \quad i = p+1, \cdots, n && (7) \end{aligned}$$

Hopfield method follows dynamics:

$$\frac{d}{dt} x_H(t) = -\nabla f(x(t)); \qquad x_H(0) = x(0) \in (0,1)^n \quad (8)$$

$$x(t) = \sigma(x_H(t)) \quad (9)$$

where $\sigma(\cdot) : \mathbb{R}^n \to [0,1]^n$ is an "activiation function" defined element-wise as:

$$\sigma(x) : x \mapsto [\sigma_1(x_1), \cdots, \sigma_n(x_n)]$$

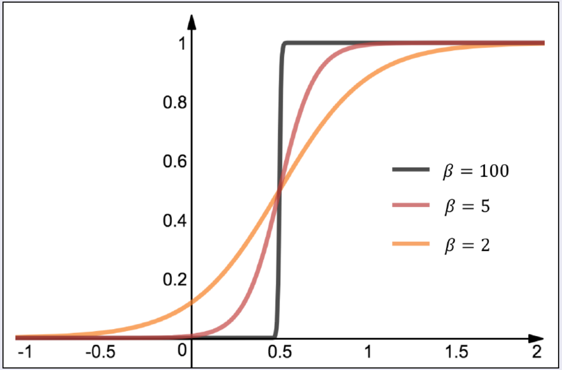
What is activation function $\sigma(x)$?

- strictly increasing
- $\sigma(\cdot) \in \mathbb{C}^1$ with Lipschitz constant L_{σ_i}

Example: tanh

$$\sigma_i(x) = \frac{1}{2}\tanh(\beta_i(x - \frac{1}{2})) + \frac{1}{2}; \qquad \beta_i > 0$$

"soft projection operator" from $\mathbb{R}$ to $\{0, 1\}$



Nonlinear Gradient Descent

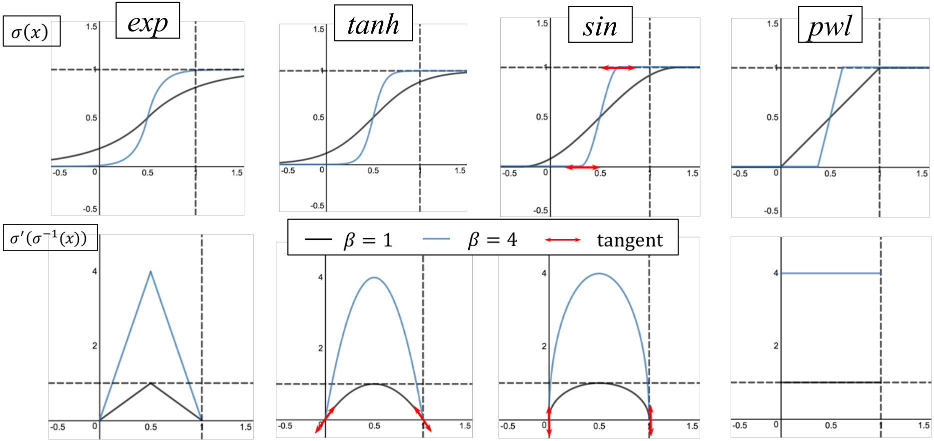
If $\sigma(\cdot)$ is a homeomorphism, then we see a nonlinear gradient descent:

$$\frac{d}{dt}x(t) = -\sigma'(\sigma^{-1}(x(t))) \odot \nabla f(x(t)) \tag{10}$$

Nonlinear Gradient Descent

If $\sigma(\cdot)$ is a homeomorphism, then we see a nonlinear gradient descent:

$$\frac{d}{dt}x(t) = -\sigma'(\sigma^{-1}(x(t))) \odot \nabla f(x(t)) \tag{10}$$



Discretize time dynamics

Forward Euler time discretization of Hopfield dynamics:

$$x_H^{k+1} = x_H^k - \alpha^k \nabla f(x^k); \qquad x_H^0 = x^0 \in (0,1)^n \tag{11}$$

$$x^k = \sigma(x_H^k) \tag{12}$$

Discretize time dynamics

Forward Euler time discretization of Hopfield dynamics:

$$x_H^{k+1} = x_H^k - \alpha^k \nabla f(x^k); \qquad x_H^0 = x^0 \in (0,1)^n \tag{11}$$

$$x^k = \sigma(x_H^k) \tag{12}$$

For quadratic $f(x) = \frac{1}{2}x^T Q x$

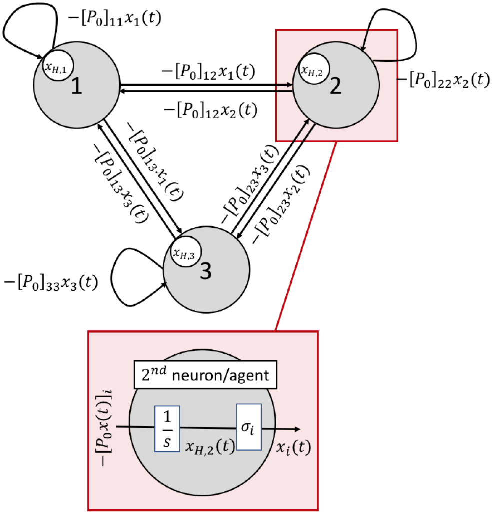
$$x_H^{k+1} = x_H^k - \alpha^k Q x^k; \qquad x_H^0 = x^0 \in (0,1)^n \tag{13}$$

$$x^k = \sigma(x_H^k) \tag{14}$$

Graphical Interpretation of Hopfield Method

Forward Simulation of Hopfield Neural Net!

- Undirected weighted graph
- n nodes, one for each x_i
- Each node has internal ($x_{H,i} \in \mathbb{R}$) and external ($x_i \in R$) states
- Weights $[P_0]_{ij}$ are elements of gradients of obj fcn



Hopfield vs Projected Gradient Descent

Hopfield

$$x_H^{k+1} = x_H^k - \alpha^k \nabla f(x^k) \tag{15}$$
$$x^k = \sigma(x_H^k) \tag{16}$$

Projected Gradient Descent

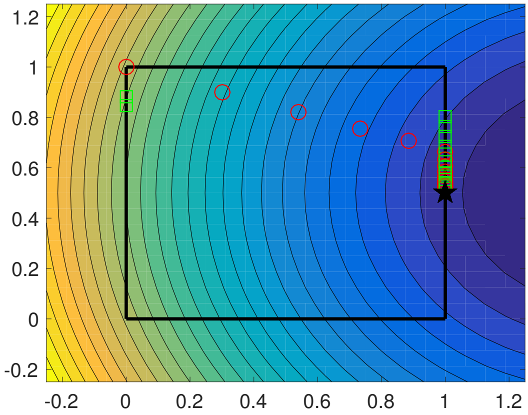
$$x_H^{k+1} = x^k - \alpha^k \nabla f(x^k) \tag{17}$$
$$x^k = \text{Proj}_{[0,1]}(x_H^k) \tag{18}$$

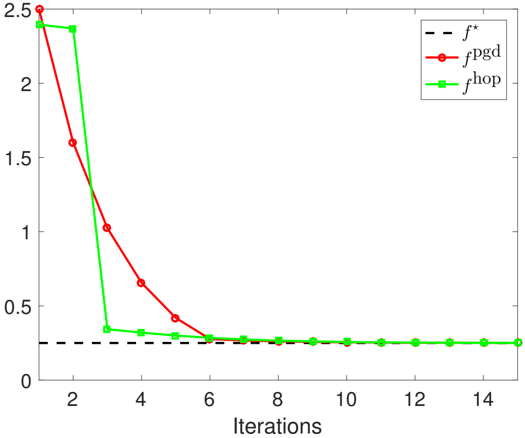
Simple Comparison

$$\text{minimize}_{x_1, x_2} \quad (x_1 - 1.5)^2 + (x_2 - 0.5)^2 \quad (19)$$

$$\text{subject to:} \quad x_1 \in \{0, 1\} \quad (20)$$

$$0 \leq x_2 \leq 1 \quad (21)$$





Continuous Improvement to a Fixed Point

Theorem: Continuous Improvement

The Hopfield method yields $f(x^{k+1}) \leq f(x^k), \forall\, k$ for an appropriate step size α^k. Specifically, the incremental improvement is bounded by:

$$0 \leq f(x^k) - f(x^{k+1}) \leq 0.5\alpha^k \cdot \nabla f(x^k)^T \Sigma^k \nabla f(x^k)$$

Corollary: Convergence to a fixed point (may not be minimizer)

There exists a $f^\dagger$ such that $f(x^k) \rightarrow f^\dagger$ as $k \rightarrow \infty$, and x^k converges to the (non-empty) set

$$\mathcal{X} = \left\{ x \in [0,1]^n \;| x_i \in \{0,1\}, i = 1, \cdots, p \quad \vee \quad \frac{\partial}{\partial x_i} f(x^k) = 0 \right\}$$

Convergence Rates

Two convergence rate results, depending on structure of obj. fcn. $f(x)$

Theorem: Sub-linear convergence in general

If $f(x)$ is convex and $\sigma(\cdot)$ is smooth, then

- $f(x^k) - f_0^\dagger = \mathcal{O}\left(\frac{1}{k^t}\right)$, with $0 < r < 1$
- To achieve precision ε, the worst case number of iterations is $2Mn/(\beta^2\varepsilon)$

Remark: Slower than gradient descent, for which convergence is guaranteed at a rate $\mathcal{O}\left(\frac{1}{k}\right)$

Dual Hopfield Method

So far, we have considered Hopfield methods to approximately solve

$$\begin{aligned} \text{minimize} \quad & f(x) && (22)\\ \text{subject to:} \quad & 0 \leq x_i \leq 1 \quad i = 1, \cdots, n && (23)\\ & x_i \in \{0, 1\} \quad i = 1, \cdots, p < n && (24) \end{aligned}$$

Dual Hopfield Method

So far, we have considered Hopfield methods to approximately solve

$$\begin{aligned} \text{minimize} \quad & f(x) && (22) \\ \text{subject to:} \quad & 0 \leq x_i \leq 1 \quad i = 1, \cdots, n && (23) \\ & x_i \in \{0, 1\} \quad i = 1, \cdots, p < n && (24) \end{aligned}$$

We now consider inequality constraints:

$$\begin{aligned} \text{minimize} \quad & f(x) && (25) \\ \text{subject to:} \quad & g_j(x) \leq 0, \quad j = 1, \cdots, m && (26) \\ & 0 \leq x_i \leq 1 \quad i = 1, \cdots, n && (27) \\ & x_i \in \{0, 1\} \quad i = 1, \cdots, p < n && (28) \end{aligned}$$

Idea: Instead of considering the “full” Lagrangian relaxation, consider

$$L(x, \mu) = f(x) + \sum_{j=1}^{m} \mu_j g_j(x) \tag{29}$$

Dual Hopfield Method

Apply Lagrangian relaxation

Idea: Instead of considering the “full” Lagrangian relaxation, consider

$$L(x, \mu) = f(x) + \sum_{j=1}^{m} \mu_j g_j(x) \tag{29}$$

Then the *dual function* is

$$\begin{aligned} D(\mu) = \min_{x} \quad & L(x, \mu) = f(x) + \sum_{j=1}^{m} \mu_j g_j(x) && (30) \\ \text{subject to:} \quad & 0 \leq x_i \leq 1 \quad i = 1, \cdots, n && (31) \\ & x_i \in \{0, 1\} \quad i = 1, \cdots, p < n && (32) \end{aligned}$$

which is amenable to Hopfield method, given μ.

Dual Ascent via Hopfield

Then solve the Dual Problem:

$$\max_{\mu \geq 0} D(\mu) \tag{33}$$

$$D(\mu) = \min_{x} L(x, \mu) = \min_{x} f(x) + \sum_{j=1}^{m} \mu_j g_j(x) \tag{34}$$

Dual Ascent via Hopfield

Then solve the Dual Problem:

$$\max_{\mu \geq 0} D(\mu) \tag{33}$$

$$D(\mu) = \min_{x} L(x, \mu) = \min_{x} f(x) + \sum_{j=1}^{m} \mu_j g_j(x) \tag{34}$$

Run Hopfield method to approximately solve $D(\mu) = \min_x L(x, \mu)$.

Suppose $x^\star(\mu) = \arg\min_x L(x, \mu)$.

The subgradient of $D(\mu)$ along dimension j: $g_j(x^\star(\mu)) \in \partial_j D(\mu)$

Algorithm 1 Dual (sub)-gradient Ascent via Hopfield Method

Initialize $\lambda^0 \geq 0$; Choose $\beta > 0$

for $k = 0, 1, \cdots, k_{\max}$

(1) use Hopfield method to approximately compute dual function

for $\ell = 0, \cdots, \ell_{\max}$

$x_H^{\ell+1} = x_H^{\ell} - \alpha^{\ell} \nabla_x L(x^{\ell}, \mu^k)$

$x^{\ell} = \sigma(x_H^{\ell+1})$

$x_{\text{hop}}^k \leftarrow x^{\ell}$

until stopping criterion is met

(2) update dual variable μ via (sub)-gradient ascent

$\mu^{k+1} = \mu^k + \beta^k \sum_{j=1}^{m} g_j(x_{\text{hop}}^k(\mu^k))$

end for

Examples: Random MIQPs

Consider solving MIQP w.r.t. $x \in \mathbb{R}^n$

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T Q x + R^T x && (35) \\ \text{subject to:} \quad & Ax \leq b && (36) \\ & A_{eq} x = b_{eq} && (37) \\ & lb \leq x \leq ub && (38) \\ & x_i \in \{0, 1\}, \quad i = 1, \cdots, p && (39) \end{aligned}$$

- Randomly generated parameters $Q, R, A, b, A_{eq}, b_{eq}, lb, ub$ for each n
- Number of constraints also randomized

Comparative Analysis

All problems solved on Matlab:

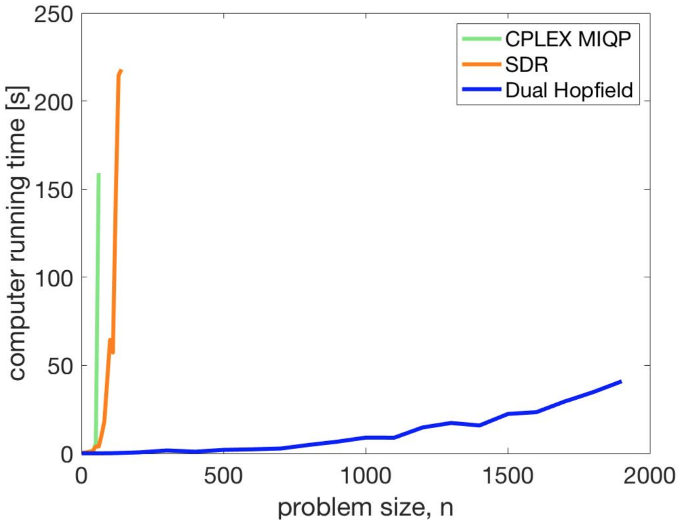
- CPLEX MIQP: using function *cplexmiqp* developed by IBM
- Binary Relaxation via CPLEX QP : using function *cplexqp*
- Semi-definite relaxation (SDR): corresponding SDP solved using *CVX*
- Hopfield: Dual Ascent Hopfield Method uses dual variables from *cplexqp*

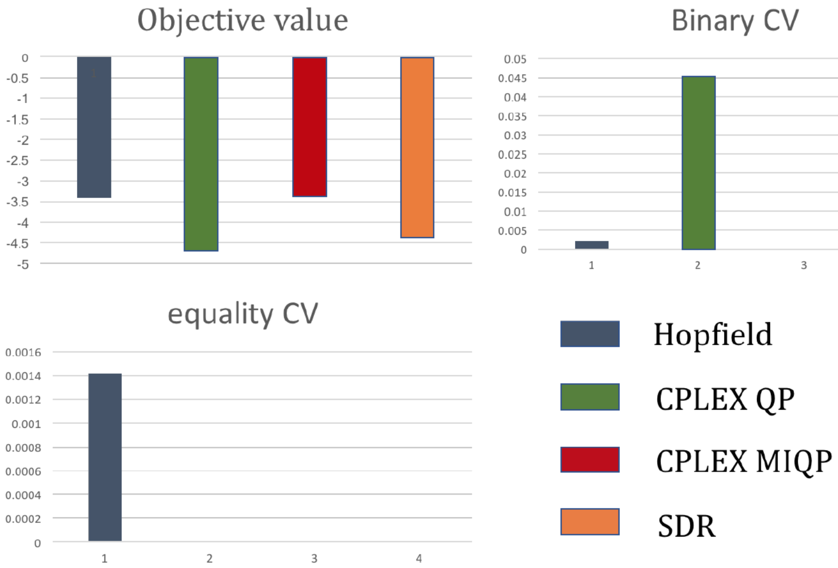
For each method, we compute:

- computer running time [sec]
- constraint violations (CV):
  - binary CV: $\frac{1}{p}\sum_{i=1}^{p} d(x_i, \{0,1\})$
  - inequality CV: $\frac{1}{m}\sum_{j=1}^{m} |[Ax - b]_j|$
  - equality CV: $\frac{1}{\ell}\sum_{k=1}^{\ell} |[A_{eq}x - b_{eq}]_k|$
- objective function value

Comparative Analysis

Computer running time




Objective value
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-5
Binary CV
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
1
2
3
4
equality CV
0.0016
0.0014
0.0012
0.001
0.0008
0.0006
0.0004
0.0002
0
1
2
3
4
Hopfield
CPLEX QP
CPLEX MIQP
SDR

SUMMARY:

- Aggregate Modeling, Estimation, Identification, and Control with PDEs
- Hopfield Methods for MINLPs – *An efficient heuristic with provable convergence*

VISIT US!

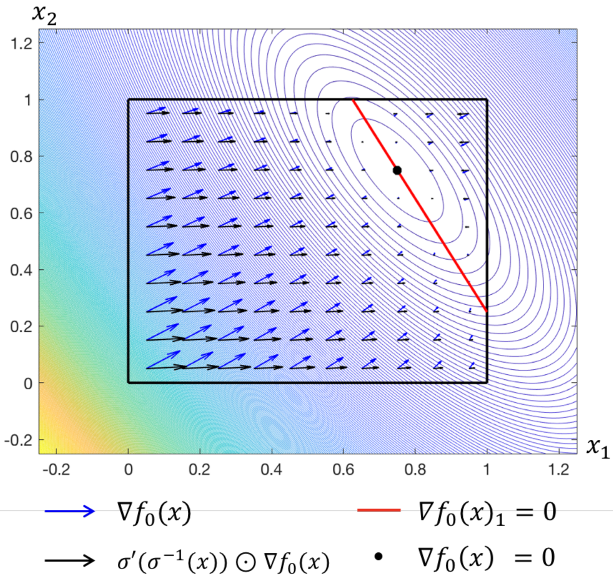
Energy, Controls, and Applications Lab (eCAL)

ecal.berkeley.edu

smoura@berkeley.edu

eCAL
energy, controls, & applications lab

APPENDIX SLIDES


x_2
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
x_1
$\nabla f_0(x)$
$\sigma'(\sigma^{-1}(x)) \odot \nabla f_0(x)$
$\nabla f_0(x)_1 = 0$
$\nabla f_0(x) = 0$

Stochastic approach to recover integer constraint:

Let x^r be solution to binary relaxation. Feasible x can be drawn randomly from $\{0, 1\}$ following Bernoulli distribution $\mathcal{B}(x^r)$.

This can be sub-optimal.

Stochastic approach to recover integer constraint:

Let x^r be solution to binary relaxation. Feasible x can be drawn randomly from $\{0, 1\}$ following Bernoulli distribution $\mathcal{B}(x^r)$.

This can be sub-optimal.

Example

$$\text{minimize}_{x \in \{0,1\}} \left(x - \frac{1}{4}\right)^2 = \frac{1}{16} \quad (x^\star = 0 \ \text{ is the optimal solution})$$

If we apply binary relaxation, we get $x^r = \frac{1}{4}$ and $\mathbb{E}_{x \sim \mathcal{B}(x^r)} \left(x - \frac{1}{4}\right)^2 = \frac{3}{16} > \frac{1}{16}$!

Other ideas:

- Branch & Bound, Branch & Cut

Existing Methods

Convex Relaxation #2: Lagrangian Relaxation

Notice that $x_i \in \{0, 1\}$ is equivalent to satisfying $x_i(1 - x_i) = 0$

$$\begin{aligned} \text{minimize} \quad & f(x) && (40) \\ \text{subject to:} \quad & g_j(x) \leq 0, \quad j = 1, \cdots, m && (41) \\ & 0 \leq x \leq 1 && (42) \\ & x_i(1 - x_i) = 0, \quad i = 1, \cdots, p < n && (43) \end{aligned}$$

Existing Methods

Convex Relaxation #2: Lagrangian Relaxation

Notice that $x_i \in \{0, 1\}$ is equivalent to satisfying $x_i(1 - x_i) = 0$

$$\begin{aligned} \text{minimize} \quad & f(x) && (40) \\ \text{subject to:} \quad & g_j(x) \leq 0, \quad j = 1, \cdots, m && (41) \\ & 0 \leq x \leq 1 && (42) \\ & x_i(1 - x_i) = 0, \quad i = 1, \cdots, p < n && (43) \end{aligned}$$

Form the *Lagrangian*:

$$L(x, \mu, \underline{\mu}, \overline{\mu}, \lambda) = f(x) + \sum_{j=1}^{m} \left[\mu_j g_j(x) + \underline{\mu}_j x_i + \overline{\mu}_j (1 - x_i) \right] + \sum_{i=1}^{p} \lambda_i x_i (1 - x_i) \tag{44}$$

Existing Methods

Convex Relaxation #2: Lagrangian Relaxation

Notice that $x_i \in \{0, 1\}$ is equivalent to satisfying $x_i(1 - x_i) = 0$

$$\begin{aligned} \text{minimize} \quad & f(x) && (40) \\ \text{subject to:} \quad & g_j(x) \leq 0, \quad j = 1, \cdots, m && (41) \\ & 0 \leq x \leq 1 && (42) \\ & x_i(1 - x_i) = 0, \quad i = 1, \cdots, p < n && (43) \end{aligned}$$

Form the *Lagrangian*:

$$L(x, \mu, \underline{\mu}, \overline{\mu}, \lambda) = f(x) + \sum_{j=1}^{m} \left[\mu_j g_j(x) + \underline{\mu}_j x_i + \overline{\mu}_j (1 - x_i) \right] + \sum_{i=1}^{p} \lambda_i x_i (1 - x_i) \tag{44}$$

Define the (concave) *dual function* of $\Lambda = [\mu, \underline{\mu}, \overline{\mu}, \lambda]$

$$D(\Lambda) = \min_{x \in \mathbb{R}^n} L(x, \mu, \underline{\mu}, \overline{\mu}, \lambda) \tag{45}$$

Weak duality approach: Solve convex program $\max_\Lambda D(\Lambda)$

Existing Methods

Convex Relaxation #3: Semi-definite Relaxation

Introduce new variable $X = xx^T$. This is called "lifting". Can re-write MIQCQP

Existing Methods

Convex Relaxation #3: Semi-definite Relaxation

Introduce new variable $X = xx^T$. This is called “lifting”. Can re-write MIQCQP

$$\text{minimize} \quad \frac{1}{2}\text{Tr}(QX) + R^T x + S \tag{46}$$

$$\text{subject to:} \quad \frac{1}{2}\text{Tr}(Q_j X) + R_j^T x + S_j \leq 0, \quad j = 1, \cdots, m \tag{47}$$

$$0 \leq x \leq 1 \tag{48}$$

$$X_{ii} = x_i, \quad i = 1, \cdots, p < n \tag{49}$$

$$X = xx^T \tag{50}$$

If Q, Q_i are positive semi-definite, then only $X = xx^T$ makes this non-convex.

Existing Methods

Convex Relaxation #3: Semi-definite Relaxation

Introduce new variable $X = xx^T$. This is called "lifting". Can re-write MIQCQP

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\text{Tr}(QX) + R^T x + S && (46) \\ \text{subject to:} \quad & \frac{1}{2}\text{Tr}(Q_j X) + R_j^T x + S_j \leq 0, \quad j = 1, \cdots, m && (47) \\ & 0 \leq x \leq 1 && (48) \\ & X_{ii} = x_i, \quad i = 1, \cdots, p < n && (49) \\ & X = xx^T && (50) \end{aligned}$$

If Q, Q_i are positive semi-definite, then only $X = xx^T$ makes this non-convex. Relax into convex inequality $X \succeq xx^T$. Using Schur complement:

$$X \succeq xx^T \Leftrightarrow \begin{bmatrix} X & x \\ x & 1 \end{bmatrix} \succeq 0 \qquad (51)$$

This can be cast as a semi-definite program (SDP).