Occupant-Centric Grid-Interactive Buildings

9. Modeling & Control

CE 397

Spring 2024

Prof Dr Zoltan Nagy

Tentative Course Outline / Schedule

Week	Class	Topic	Guest Lecture
1	01/17	Introduction / Overview / Python	
2	01/24	Machine Learning I	
3	01/31	Machine Learning II	
4	02/07	Machine Learning III	Justin Hill (Southern)
5	02/14	Occupant Behavior Modeling	
6	02/21	Occupant Behavior Modeling	Tanya Barham (CEL)
7	02/28	Occupant Behavior Modeling	Jessica Granderson (LBNL)
8	03/06	Occupant Behavior Modeling	Hussain Kazmi (KU Leuven)
9	03/13	Spring Break	
10	03/20	Advanced Control & Calibration	Ankush Chakrabarty (MERL)
11	04/27	Calibration	Donghun Kim (LBNL)
12	04/03	Introduction to CityLearn	
13	04/10	Project Work	Siva Sankaranarayanan (EPRI)
14	04/17	Project work	
15	04/24	Project work	

Plan for today

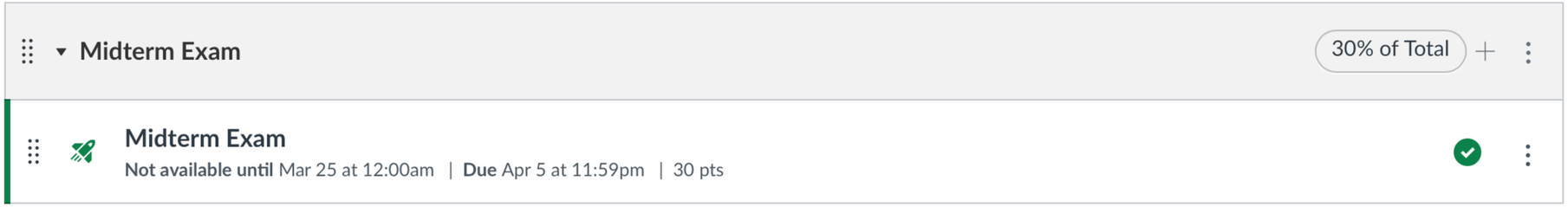
- Midterm info
- Modeling & Control of Buildings

- Continue on HW 4/5

- Guest Lecture

Midterm Info

- Quiz on Canvas
- Open March 25
- Due April 5
- No extensions will be given



- There are 30 questions. Each question is 1 point.
- You have 60min once you started the exam
- The questions are either on Machine Learning, Building Performance Simulation or Python Programming
- Questions are based on the course material until spring break
- Each question has exactly one correct answer
- You can go back and forth between the questions
- You have only one submission (no resubmissions)
- Questions and answers are shuffled randomly

Modeling and Control of Buildings

References for today

- Data-Driven modeling of building thermal dynamics

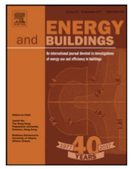
- All you need to know about model predictive control for buildings

Energy & Buildings 203 (2019) 109405

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enbuild



Data-driven modeling of building thermal dynamics: Methodology and state of the art

Zequn Wang*, Yuxiang Chen

Department of Civil and Environmental Engineering, University of Alberta, 9211 116 Street NW, Edmonton, Alberta T6G 1H9, Canada

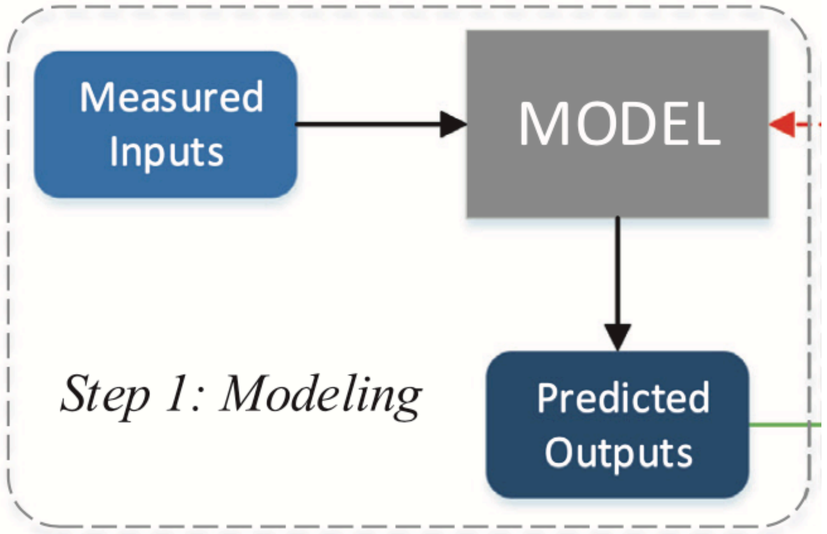
Annual Reviews in Control

Volume 50, 2020, Pages 190-232

All you need to know about model predictive control for buildings

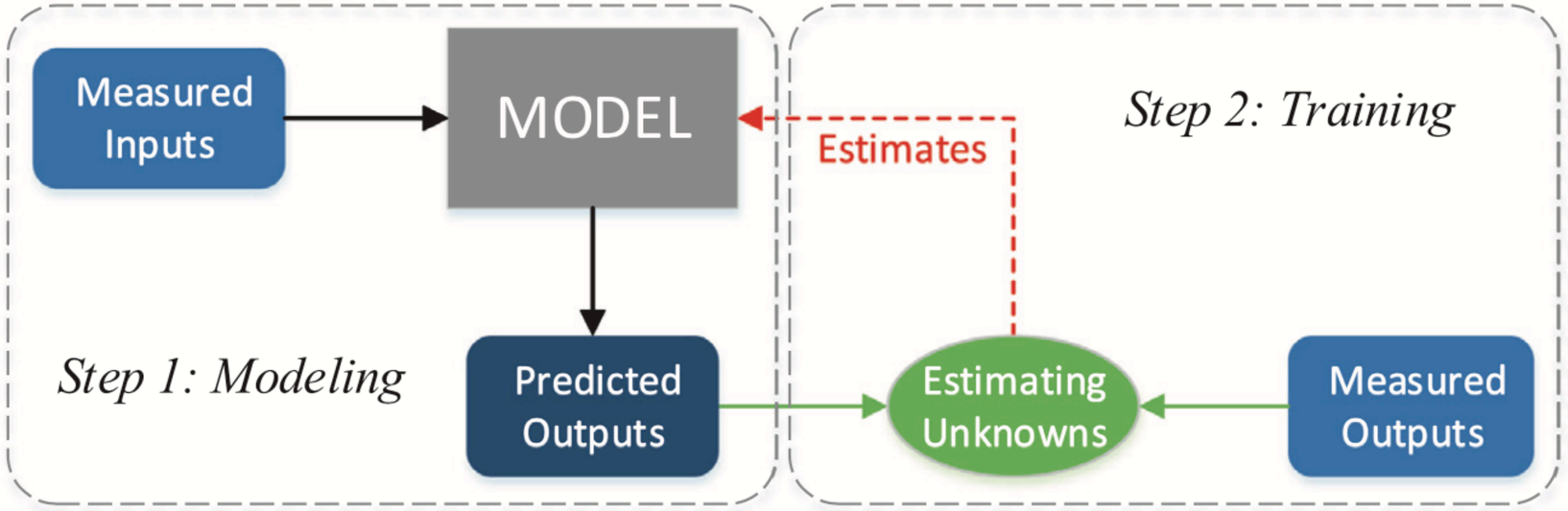
Ján Drgoňa [a, b], Javier Arroyo [b, c, d], Iago Cupeiro Figueroa [b, c], David Blum [e], Krzysztof Arendt [f], Donghun Kim [e, g], Enric Perarnau Ollé [b], Juraj Oravec [h], Michael Wetter [e], Draguna L. Vrabie [a], Lieve Helsen [b, c]

Data-Driven Modeling of building thermal dynamics



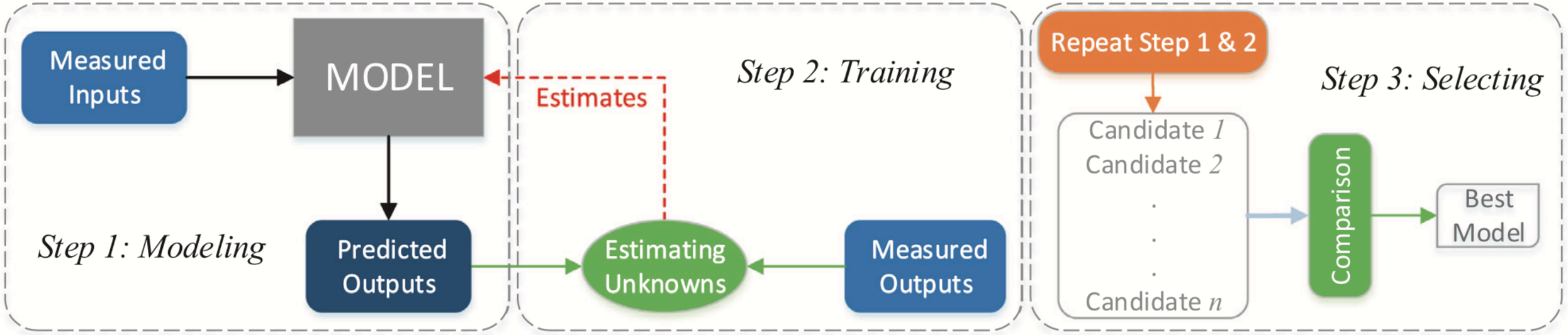
1. **Modeling**: formulate mathematical model(s) with unknown parameters to predict system outputs using measured inputs

Data-Driven Modeling of building thermal dynamics



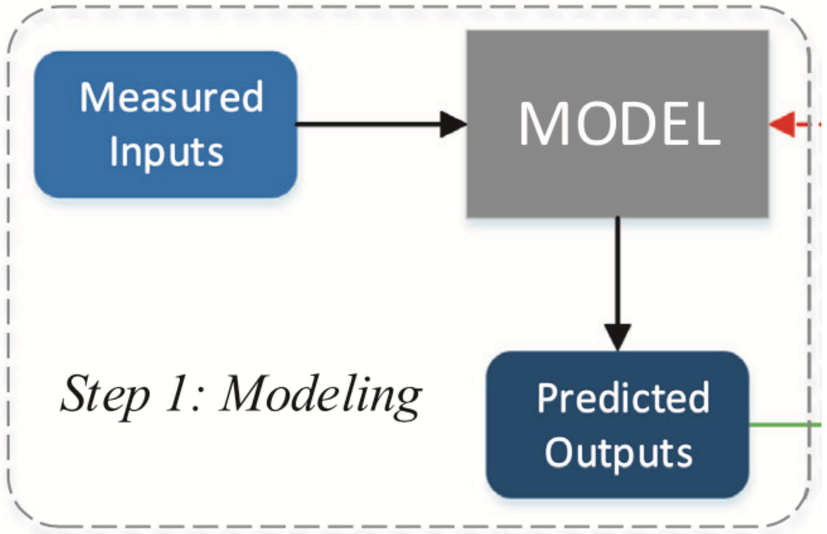
1. **Modeling**: formulate mathematical model(s) with unknown parameters to predict system outputs using measured inputs
2. **Training**: estimate unknown parameters by matching measured outputs with predicted outputs

Data-Driven Modeling of building thermal dynamics



1. **Modeling**: formulate mathematical model(s) with unknown parameters to predict system outputs using measured inputs
2. **Training**: estimate unknown parameters by matching measured outputs with predicted outputs
3. **Selecting**: choose best model

Data-Driven Modeling of building thermal dynamics



1. **Modeling**: formulate mathematical model(s) with unknown parameters to predict system outputs using measured inputs

Categories of data-driven models

- Resistor-Capacitor (RC) models
- Time-invariant transfer function (TF) models
- Artificial intelligence (AI) based models

- most building physics/thermal processes are non-linear; however most models are linear and time-invariant (except AI models)

Categories of data-driven models

- **Resistor-Capacitor (RC) models**
- Time-invariant transfer function (TF) models
- Artificial intelligence (AI) based models

- most building physics/thermal processes are non-linear; however most models are linear and time-invariant (except AI models)

Resistor-Capacitor (RC) Models

- Electrical-thermal equivalency
- Thermal dynamics are captured by a network of temperatures nodes, thermal resistors and thermal capacitors
- ‘equivalent’ thermal parameters

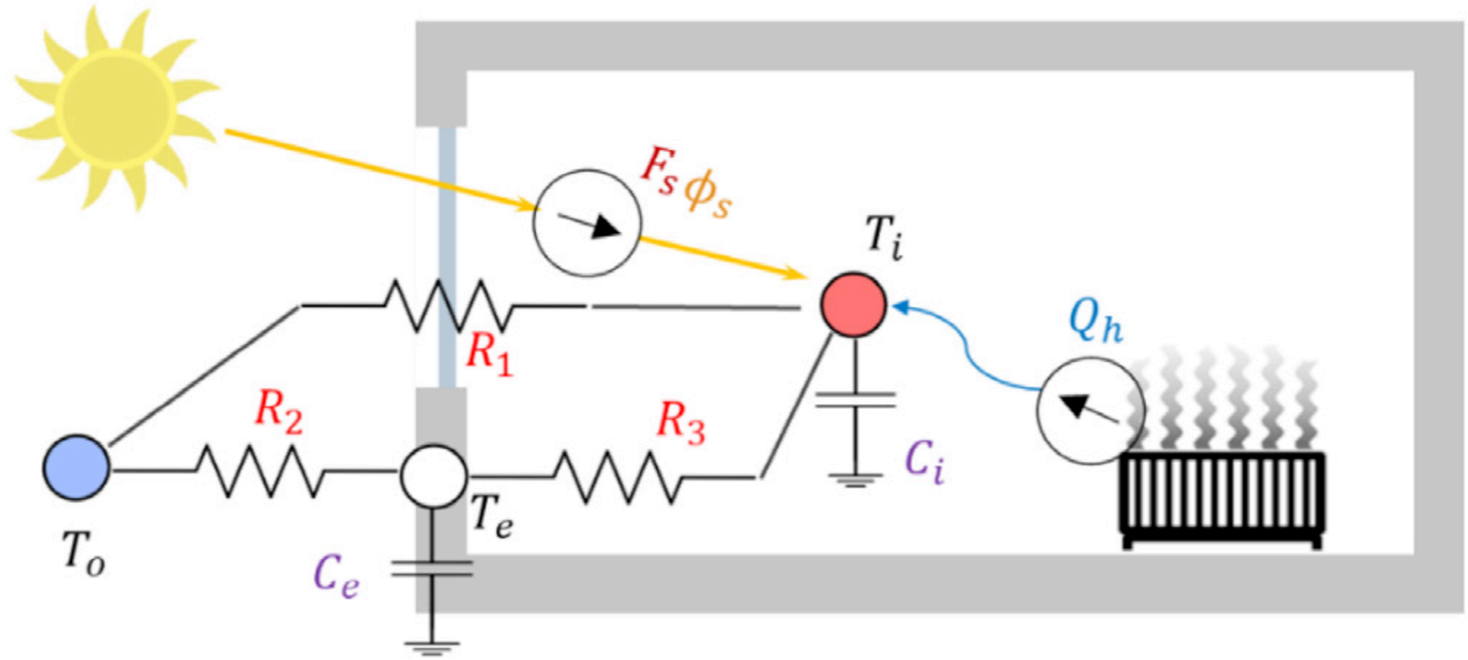


Fig. 2. An example representation of a room with a RC network (T_i: indoor air temperature; T_e: the average temperature of building envelop; T_o: outdoor air temperature; Q_h: heating power; ϕ_s: solar radiation on south façade; F_s: a ratio factor).

Resistor-Capacitor (RC) Models

- known/measured values of variables are called *system inputs/outputs*
- temperatures linked to thermal nodes are *system states*
- number of states determines the model order

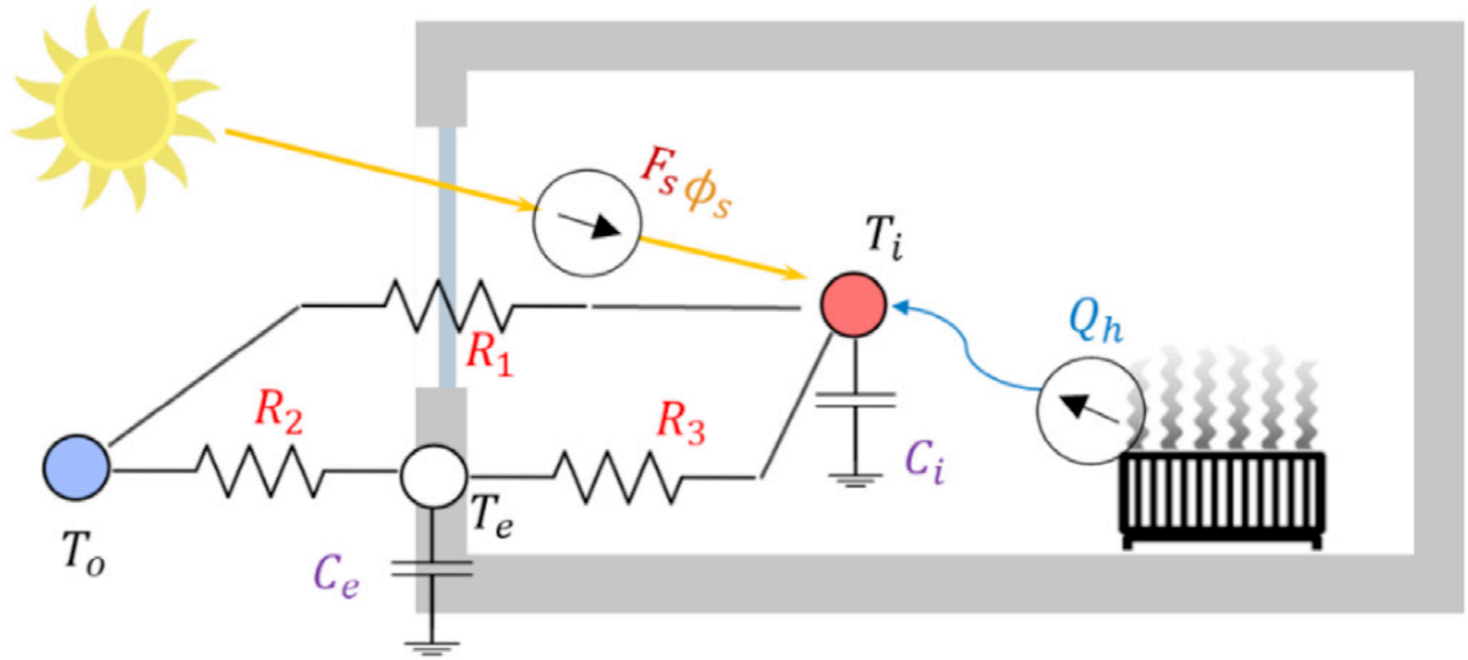


Fig. 2. An example representation of a room with a RC network (T_i: indoor air temperature; T_e: the average temperature of building envelop; T_o: outdoor air temperature; Q_h: heating power; ϕ_s: solar radiation on south façade; F_s: a ratio factor).

Resistor-Capacitor (RC) Models

- Each node temperature is governed by ordinary differential equation:

$$C_k \frac{dT_k}{dt} = \sum_k \frac{T_k - T_k}{R_{k,k}} + \sum_j F_j Q_j \qquad (1)$$

where,

T_k represents the temperature of the k^{th} node;
C_k is the thermal capacitance of the k^{th} node;
T_k represents the temperature of a neighbor of the k^{th} node.
$R_{k,\,k}$ is the thermal resistance between the k^{th}node and its neighbor;
Q_j denotes the j^{th} heat input to the k^{th} node (e.g., solar radiation); and
F_j is a factor that evaluates effective heat gains to the k^{th} node due to Q_j.

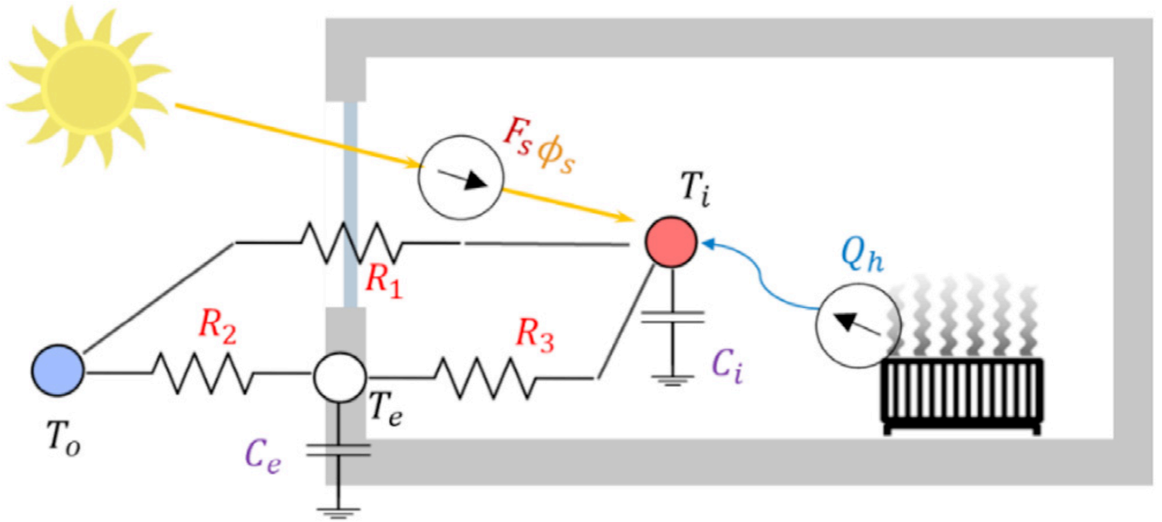


Fig. 2. An example representation of a room with a RC network (T_i: indoor air temperature; T_e: the average temperature of building envelop; T_o: outdoor air temperature; Q_h: heating power; ϕ_s: solar radiation on south façade; F_s: a ratio factor).

Resistor-Capacitor (RC) Models

- ODEs are then first rearranged into state-space representation in continuous time

$$\frac{dx}{dt} = \tilde{A}x + \tilde{B}u$$

$$y = \tilde{C}x + \tilde{D}u$$

$\tilde{t}$ represents time;
x, u, and y are vectors of model states, inputs, and outputs, respectively; and
$\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$ are matrices determined from model parameters e.g., (R_1; R_2; R_3; C_e; C_i; F_s).

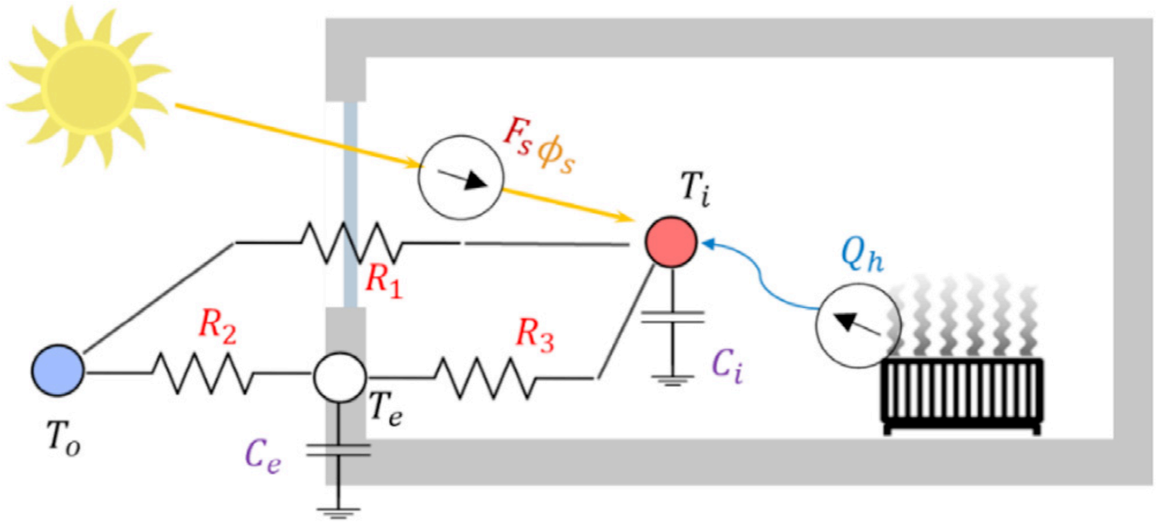


Fig. 2. An example representation of a room with a RC network (T_i: indoor air temperature; T_e: the average temperature of building envelop; T_o: outdoor air temperature; Q_h: heating power; ϕ_s: solar radiation on south façade; F_s: a ratio factor).

Resistor-Capacitor (RC) Models

- and then into finite difference form for training and control (control oriented model)

$$x_{t+1} = Ax_t + Bu_t + Ke_t$$

$$y_t = Cx_t + Du_t + e_t$$

t represents the t^{th} time step;
x_t is a vector of states converted from x_t;
A, B, C, and D are matrices calculated from $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$;
K is the optimal Kalman gain matrix; and
e_t is a vector of stochastic processes, often assumed as normally distributed white noises [15,18,20,40,41].

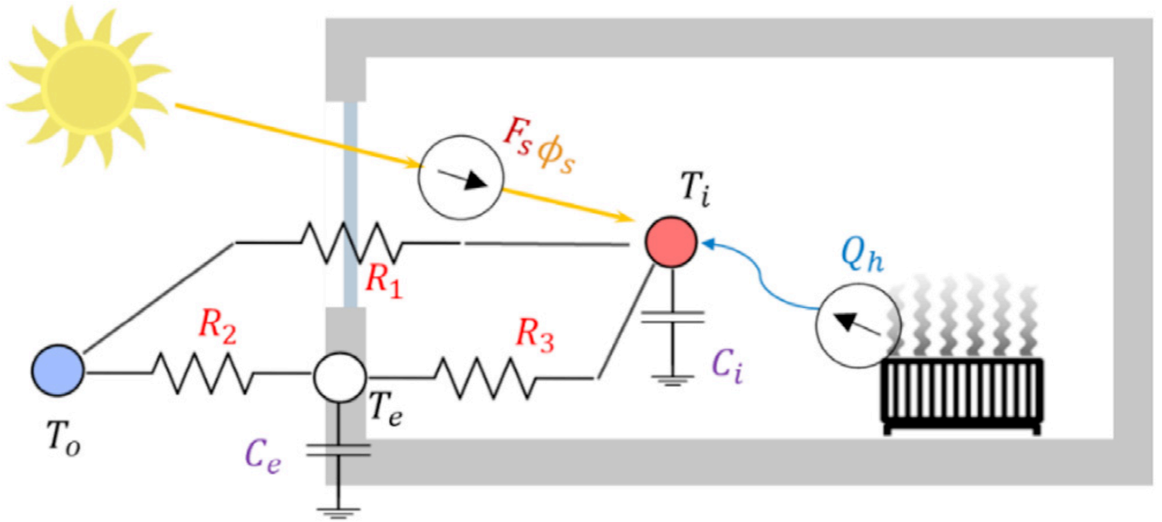
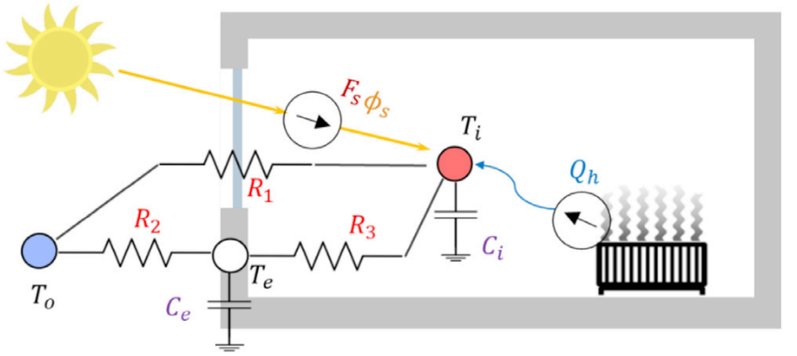


Fig. 2. An example representation of a room with a RC network (T_i: indoor air temperature; T_e: the average temperature of building envelop; T_o: outdoor air temperature; Q_h: heating power; ϕ_s: solar radiation on south façade; F_s: a ratio factor).

Resistor-Capacitor (RC) Models



esentation of a room with a RC network (T_i: indoor air tem-
e temperature of building envelop; T_o: outdoor air temper-
r; ϕ_s: solar radiation on south façade; F_s: a ratio factor).

Table 1
An example of RC model formulation.

Model structure	RC network in Fig. 2
Variables	Input: $u = [T_o \quad \phi_s \quad Q_h]'$; Output: $y = T_i$; State: $x = [T_e \quad T_i]'$; Parameter: $\theta = [R_1 \quad R_2 \quad R_3 \quad C_e \quad C_i \quad F_s \quad F_h]'$
Ordinary differential equations	$C_e \frac{dT_e}{d\tilde{t}} = \frac{T_o - T_e}{R_2} + \frac{T_i - T_e}{R_3}$
	$C_i \frac{dT_i}{d\tilde{t}} = \frac{T_o - T_i}{R_1} + \frac{T_e - T_i}{R_3} + F_s\phi_s + F_hQ_h \ (F_h = 1)$
State-space representation in continuous-time	$\frac{d}{d\tilde{t}}\begin{bmatrix} T_e \\ T_i \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{1}{C_eR_2} - \frac{1}{C_eR_3} & \frac{1}{C_eR_3} \\ \frac{1}{C_iR_3} & -\frac{1}{C_iR_1} - \frac{1}{C_iR_3} \end{bmatrix}}_{\tilde{A}}\begin{bmatrix} T_e \\ T_i \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{1}{C_eR_2} & 0 & 0 \\ \frac{1}{C_iR_1} & \frac{F_s}{C_i} & \frac{1}{C_i} \end{bmatrix}}_{\tilde{B}}\begin{bmatrix} T_o \\ \phi_s \\ Q_h \end{bmatrix}$
	$T_i = \underbrace{[0 \quad 1]}_{\tilde{C}}\begin{bmatrix} T_e \\ T_i \end{bmatrix} + \underbrace{0 \quad 0 \quad 0}_{\tilde{D}}\begin{bmatrix} T_o \\ \phi_s \\ Q_h \end{bmatrix}$

Categories of data-driven models

- Resistor-Capacitor (RC) models
- **Time-invariant transfer function (TF) models**
- Artificial intelligence (AI) based models

- most building physics/thermal processes are non-linear; however most models are linear and time-invariant (except AI models)

TF Models

- Principle: Output(s) of a system depend on current and past inputs.
- For linear system current output can be related to the history of inputs
- Linear time-invariant TF model: $y_t = G(\theta, z)u_t + H(\theta, z)e_t$

z is a forward-shift operator: $u_{t+1} = zu_t$ or $u_t = z^{-1}u_{t+1}$;
$G(\theta, z)$ is the transfer function for inputs;
$H(\theta, z)$ is the transfer function for system noise; and
e_t is the noise term accounting for system disturbances.

$G(\theta, z)$ and $H(\theta, z)$ are often expressed in rational function forms, e.g., $G(\theta, z) = \mathcal{B}(z)/\mathcal{A}(z)$ and $H(\theta, z) = \mathcal{C}(z)/\mathcal{D}(z)$, where $\mathcal{A}(z)$, $\mathcal{B}(z)$, $\mathcal{C}(z)$, and $\mathcal{D}(z)$ are polynomial functions:

$$\mathcal{A}(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{na} z^{-na}$$
$$\mathcal{B}(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{nb} z^{-nb}$$
$$\mathcal{C}(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_{nc} z^{-nc}$$
$$\mathcal{D}(z) = d_0 + d_1 z^{-1} + d_2 z^{-2} + \ldots + d_{nd} z^{-nd}$$

TF Models

- Depending on the polynomial functions TF models can be categorized as

$$y_t = G(\theta, z)u_t + H(\theta, z)e_t$$

BJ	$y_t = \frac{\mathcal{B}(z)}{\mathcal{A}(z)}u_t + \frac{\mathcal{C}(z)}{\mathcal{D}(z)}e_t$
ARMAX	$y_t = \frac{\mathcal{B}(z)}{\mathcal{A}(z)}u_t + \frac{\mathcal{C}(z)}{\mathcal{A}(z)}e_t$
ARX	$y_t = \frac{\mathcal{B}(z)}{\mathcal{A}(z)}u_t + \frac{1}{\mathcal{A}(z)}e_t$
OE	$y_t = \frac{\mathcal{B}(z)}{\mathcal{A}(z)}u_t + e_t$

- Model structures are different by the noise model H.
- ARMAX & ARX most suitable for building applications

TF Models

Table 2
An example of TF model (ARX) formulation.

Model structure	ARX (na, nb) $nb = [nb_1 \quad nb_2 \quad nb_3]$
Variables	Input: $u = [T_o \quad \phi_s \quad Q_h]'$; Output: $y = T_i$; Parameter: $\theta = [a_1 \quad b_{1,0} \quad b_{1,1} \quad b_{2,0} \quad b_{2,1} \quad b_{3,0}]'$
Transfer function form	$(1 + a_1 z^{-1} + \ldots + a_{na} z^{-na}) T_{i,t} = \begin{bmatrix} b_{1,0} + b_{1,1} z^{-1} + \ldots + b_{1,nb_1} z^{-nb_1} \\ b_{2,0} + b_{2,1} z^{-1} + \ldots + b_{2,nb_2} z^{-nb_2} \\ b_{3,0} + b_{3,1} z^{-1} + \ldots + b_{3,nb_3} z^{-nb_3} \end{bmatrix} [T_{o,t} \quad \phi_{s,t} \quad Q_{h,t}] + e_t$
Polynomial form	e.g., $na = 1$, $nb = [2 \quad 2 \quad 1]$ $T_{i,t} = -a_1 T_{1,t-1} + b_{1,0} T_{o,t} + b_{1,1} T_{o,t-1} + b_{2,0} \phi_{s,t} + b_{2,1} \phi_{s,t-1} + b_{3,0} Q_{h,t} + e_t$

- Parameters in TF models are not as easily interpretable as in RC models
- TF models are more straightforward to develop
- Mostly used for prediction (room temperatures, humidity, heating/ cooling loads)
- TF models are purely statistical. May violate conservation laws, or be unstable and unphysical

Categories of data-driven models

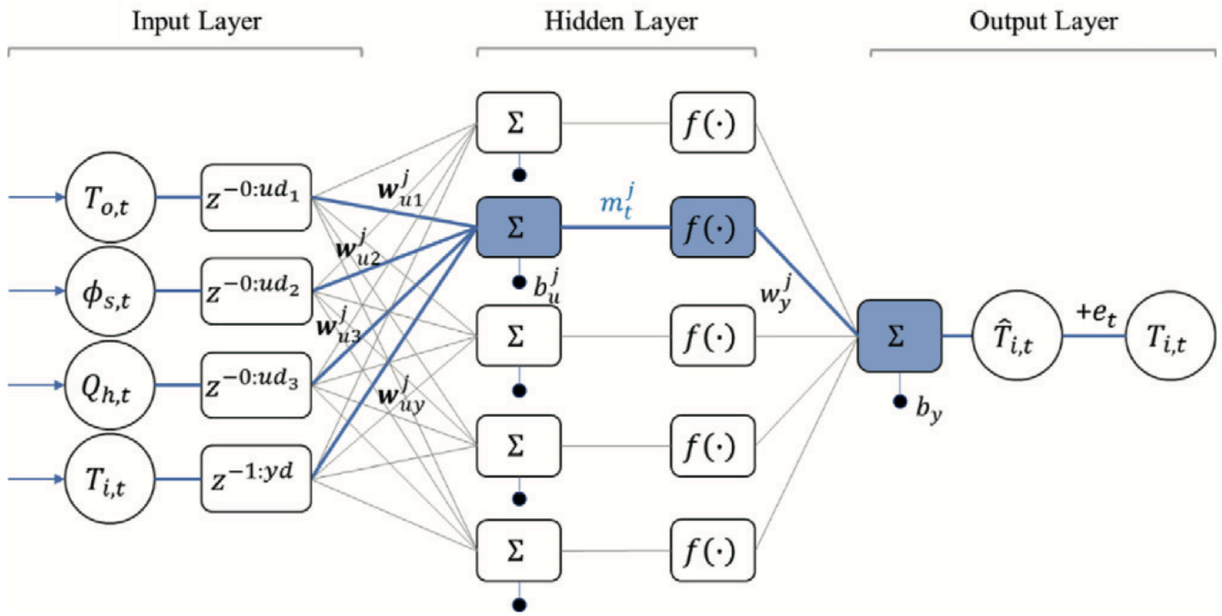
- Resistor-Capacitor (RC) models
- Time-invariant transfer function (TF) models
- **Artificial intelligence (AI) based models**

- most building physics/thermal processes are non-linear; however most models are linear and time-invariant (except AI models)

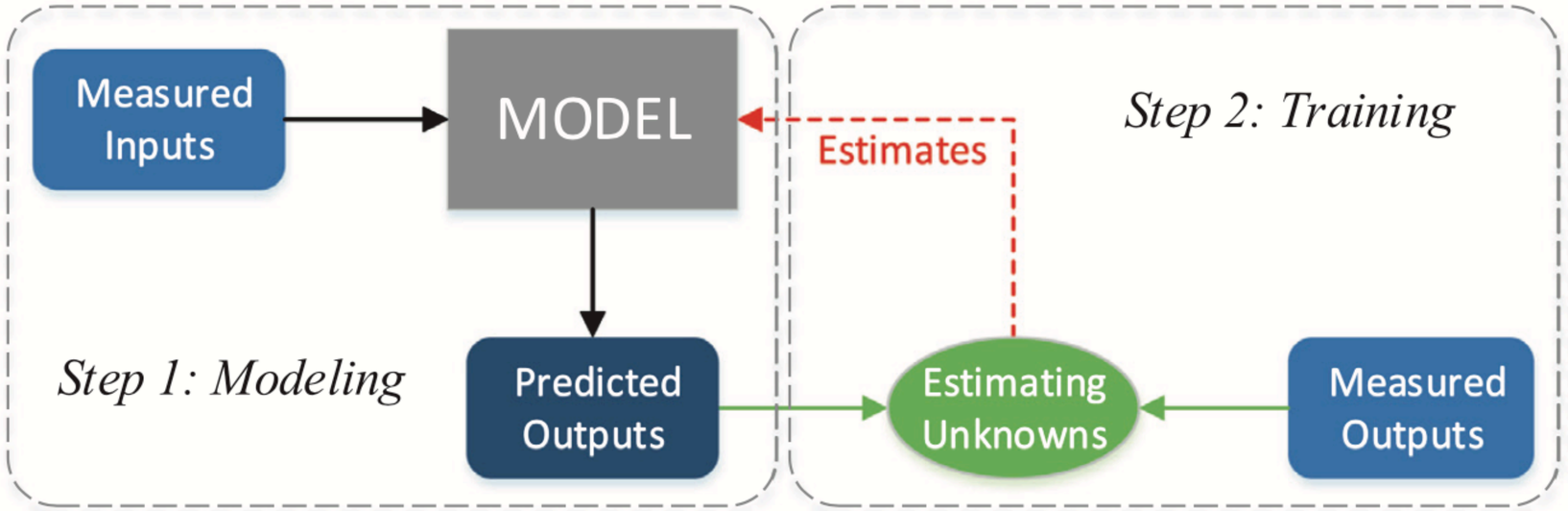
AI Based models

- Supervised learning models relating inputs to outputs
- Feedforward ANN can be turned into a recurrent NN (RNN) by connecting its output with its input
- Most AI methods are non-linear and not explainable, but have higher prediction accuracy

Table 3
An example of AI model (ANN) formulation.

Model Structure	
Variables	Input: $\boldsymbol{u} = [T_o \quad \phi_s \quad Q_h]'$; Output: $\boldsymbol{y} = T_i$; Parameter: $\boldsymbol{\theta} = \{\boldsymbol{w}_{u1}^{1:nl} \quad \boldsymbol{w}_{u2}^{1:nl} \quad \boldsymbol{w}_{u3}^{1:nl} \quad \boldsymbol{w}_{uy}^{1:nl} \quad b_u^{1:nl} \quad w_y^{1:nl} \quad b_y\}$; e.g., $w_y^{1:nl} = [w_y^1, w_y^2, \dots, w_y^{nl}]$

Data-Driven Modeling of building thermal dynamics



1. **Modeling**: formulate mathematical model(s) with unknown parameters to predict system outputs using measured inputs
2. **Training**: estimate unknown parameters by matching measured outputs with predicted outputs

Model Training

- Optimization process to estimate the unknown parameters by minimizing the value of an objective function

$$\hat{\theta} = \min_{\theta} V(\theta)$$

- V(t) is a parameter dependent objective function
- objective function is also called *loss function*
- Two examples: Prediction error method (PEM) or Maximum likelihood estimation (MLE)

Model Training: PEM

- The loss function is a function of the prediction errors:

$$V(\theta) = \frac{1}{N}\sum_{t=1}^{N} e_t(\theta)' e_t(\theta)$$

N is the number of samples;
$e_t(\theta)$ is the prediction error at time t: $e_t(\theta) = y_t - \hat{y}_{t|t-h}(\theta)$;
$e_t(\theta)'$ is the transpose of $e_t(\theta)$;
h is the prediction horizon;
$\hat{y}_{t|t-h}(\theta)$ is the predicted output at t given θ and Y_{t-h}; and
Y_{t-h} contains the measured outputs up to $t-h$: $Y_{t-h} = [y_{t-h}, y_{t-h-1}, \ldots, y_1]$.

- h is hyperparameter and should be chosen on purpose (short term vs long term prediction)
- h=1: one step ahead prediction
- h>1: multi-step ahead prediction (suitable for model predictive controllers)
- h=infinity : simulation error

Model Training: MLE

- MLE finds parameter estimates that make the observations (measurements) most likely to be within the predicted outputs.

$$V(\theta) = -\log L(\theta;\ Y_N)$$

- The joint probability density of all the observations - *the likelihood function* - should be maximized.

$L(\cdot)$ is the likelihood function: $L(\theta;\ Y_N) = \prod_{t=1}^{N} p(y_t|Y_{t-1}, \theta)$;
Y_t contains the observations up to time t: $Y_t = [y_t,\ y_{t-1},\ \ldots,\ y_1]$; and
$p(\cdot|\cdot)$ is the conditional density function, often assumed to be Gaussian.

The Gaussian densities are determined by the conditional mean $\hat{y}_{t|t-1}$ and the conditional covariance $\mathbf{\Sigma}_{t|t-1}$, i.e.,

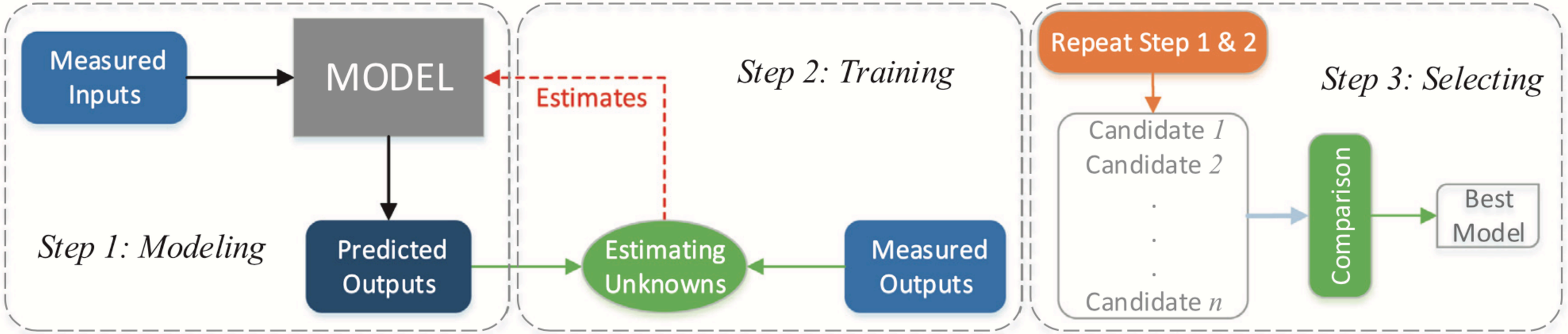
$$p(y_t|Y_{t-1}, \theta) = \exp\left[-\frac{1}{2}\left(y_t - \hat{y}_{t|t-1}\right)'\mathbf{\Sigma}_{t|t-1}\left(y_t - \hat{y}_{t|t-1}\right)\right] / \sqrt{(2\pi)^{ny}\left|\mathbf{\Sigma}_{t|t-1}\right|}$$

Model Training: Offline vs Online

- Offline Training: size of the dataset is fixed before the algorithms start to optimize the parameter values

- Online Training: Update parameter estimates continuously as new data become available. Typically older data are “forgotten”.

Data-Driven Modeling of building thermal dynamics

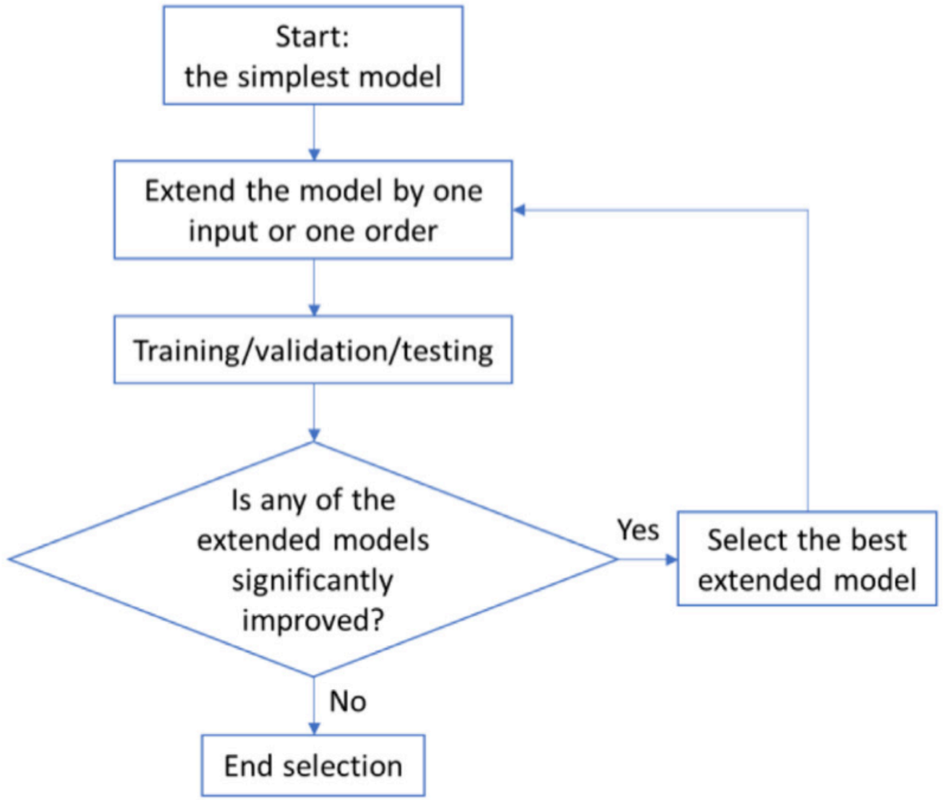


1. **Modeling**: formulate mathematical model(s) with unknown parameters to predict system outputs using measured inputs
2. **Training**: estimate unknown parameters by matching measured outputs with predicted outputs
3. **Selecting**: choose best model

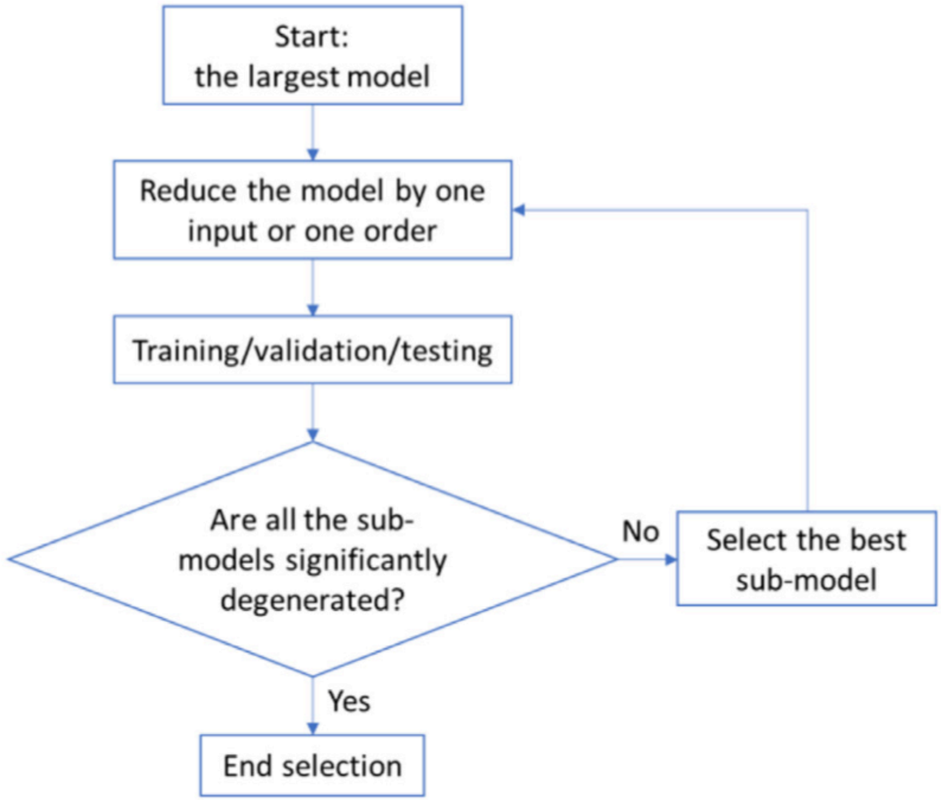
Model Selection

- Systematic routine through which candidate models are validated, tested and compared
- Purpose: find the most suitable model structure that gives favorable prediction accuracy with the least possible complexity.

Forward vs Backward selection

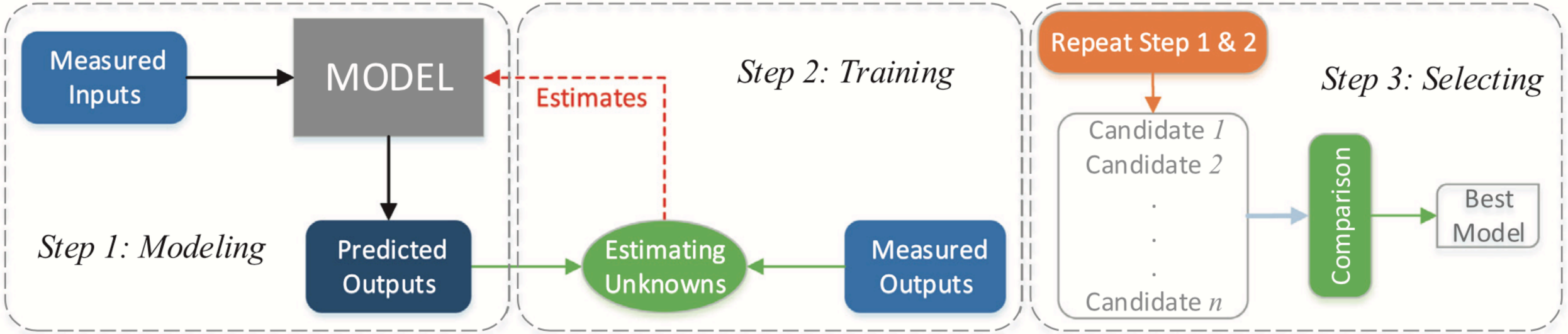


(a) forward selection



(b) backward selection

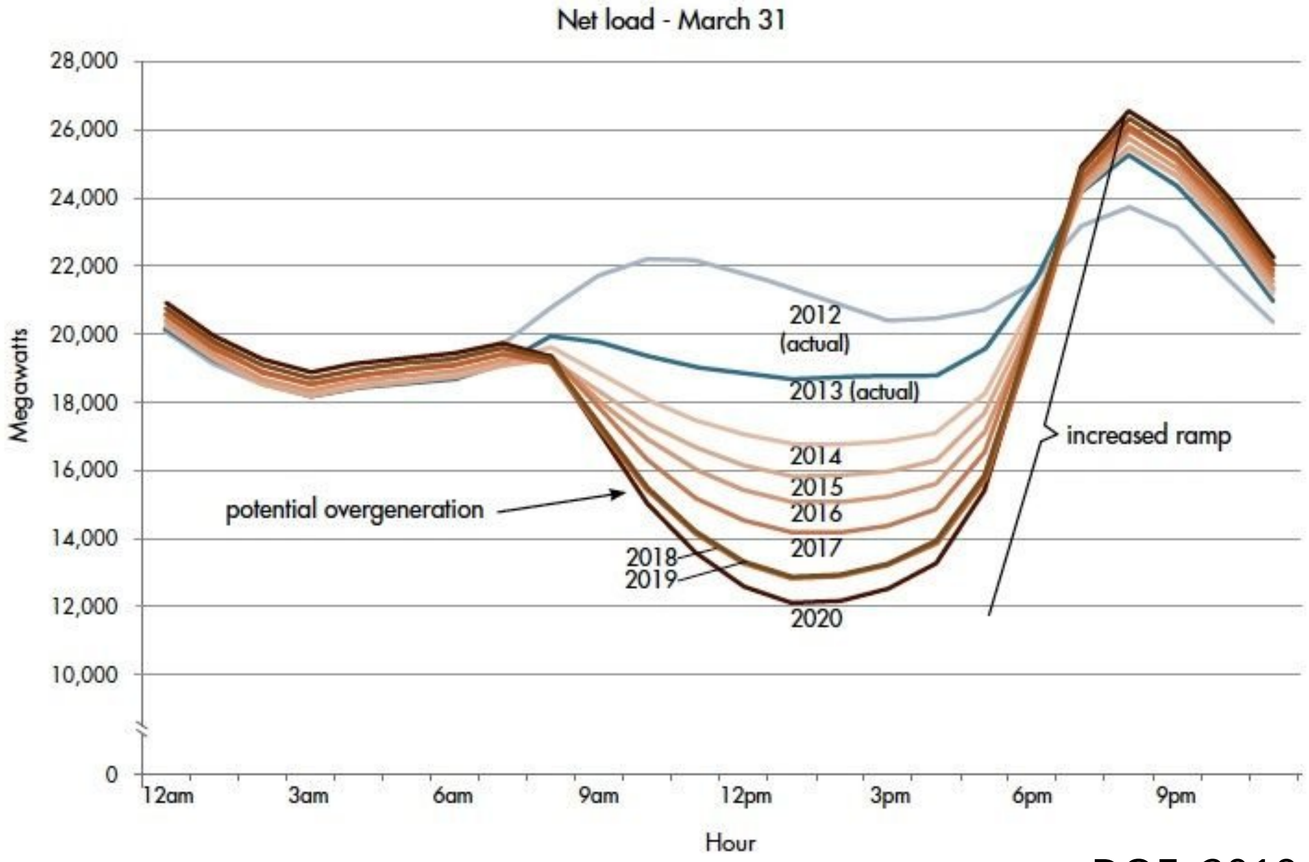
Data-Driven Modeling of building thermal dynamics



1. **Modeling**: formulate mathematical model(s) with unknown parameters to predict system outputs using measured inputs
2. **Training**: estimate unknown parameters by matching measured outputs with predicted outputs
3. **Selecting**: choose best model

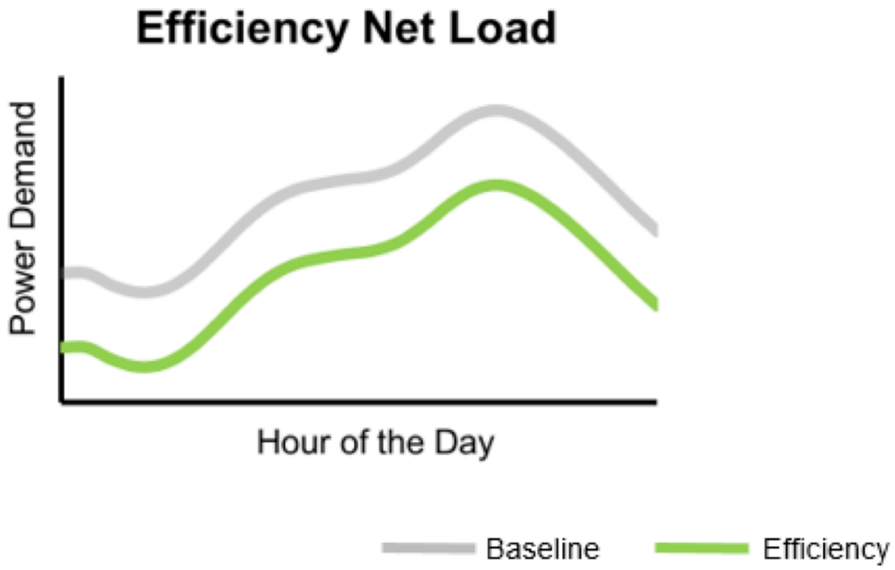
Break

Energy flexibility needed to align supply with demand



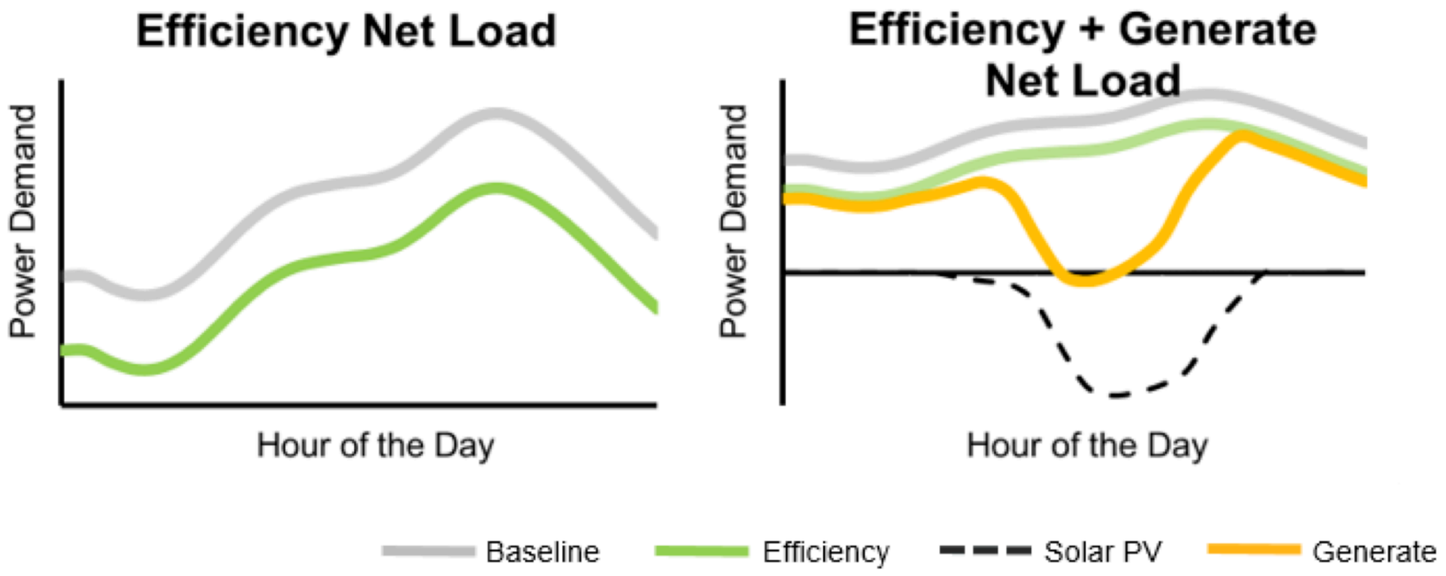
DOE, 2019

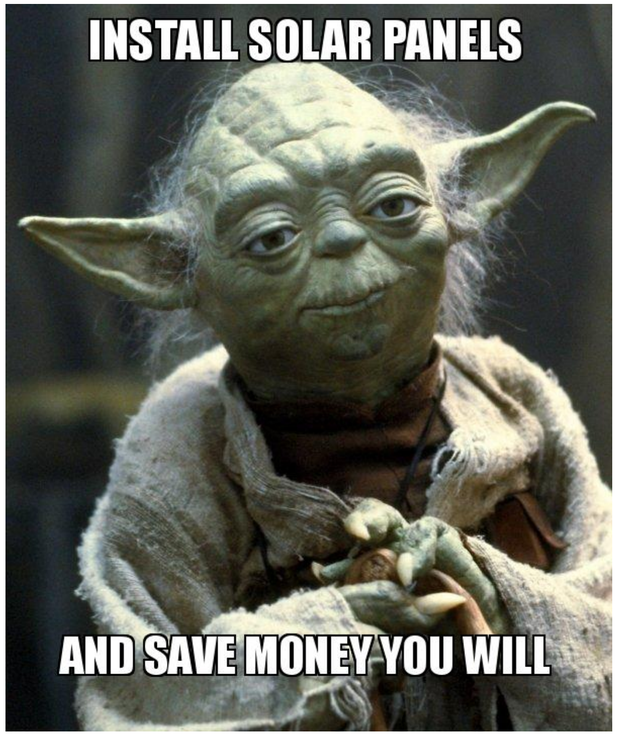
Efficiency alone does not shift loads



US Dept of Energy, 2019

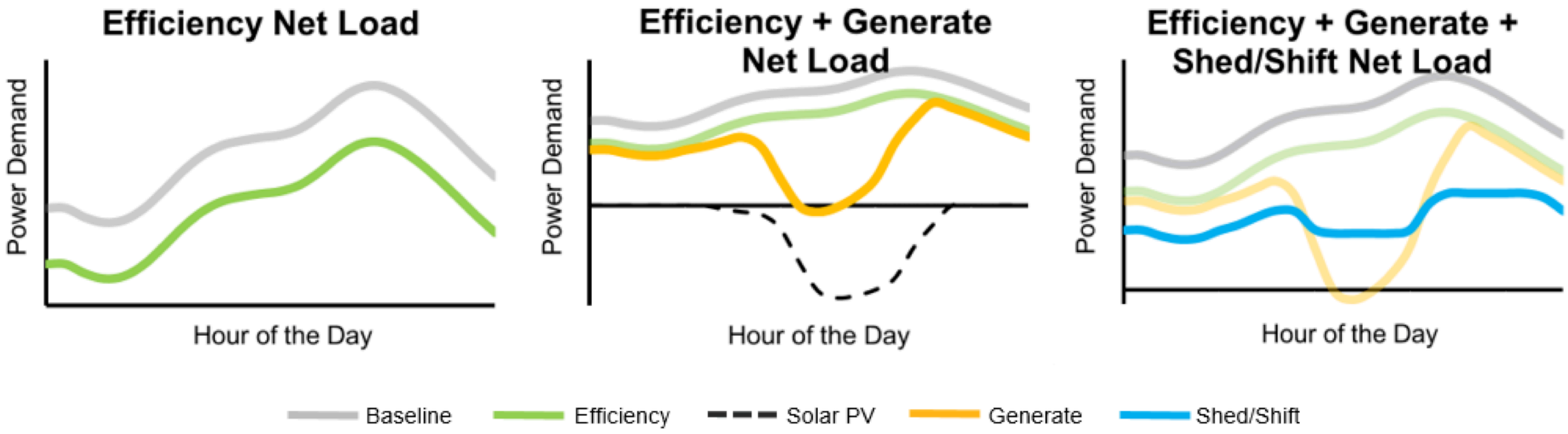
Solar power creates the duck/canyon curve





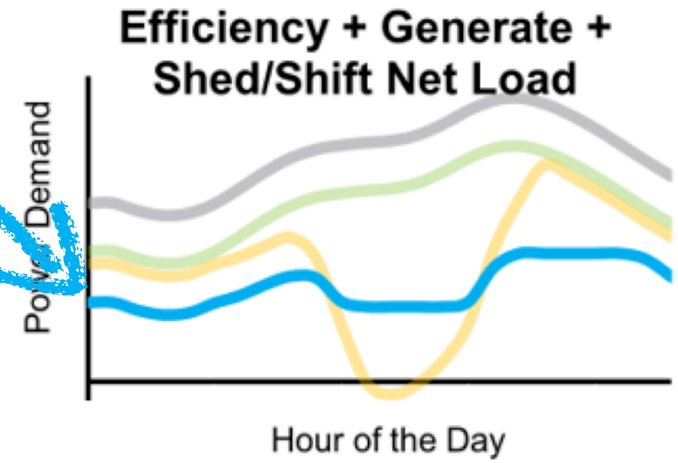
US Dept of Energy, 2019

Grid-interactive communities can shift loads



US Dept of Energy, 2019

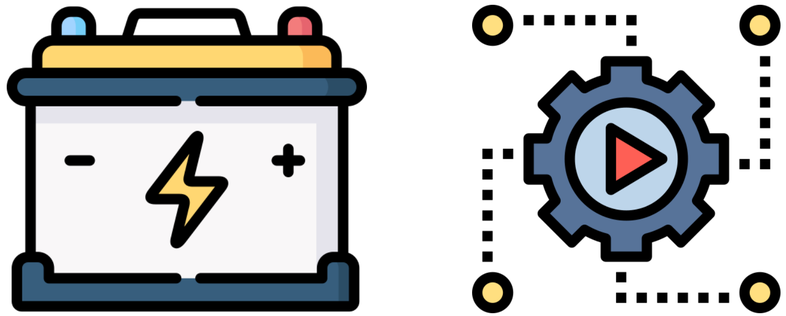
How do we get to here?



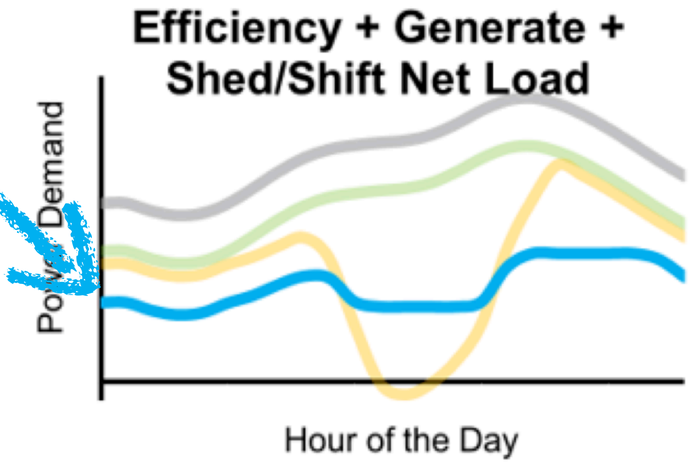
Baseline Efficiency Solar PV Generate Shed/Shift

US Dept of Energy, 2019

How do we get to here?



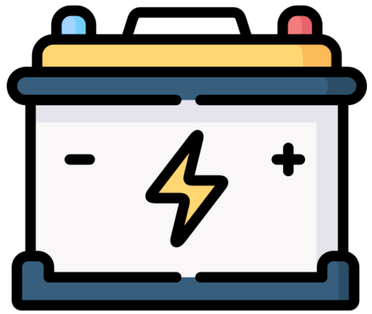
Storage + Control



US Dept of Energy, 2019

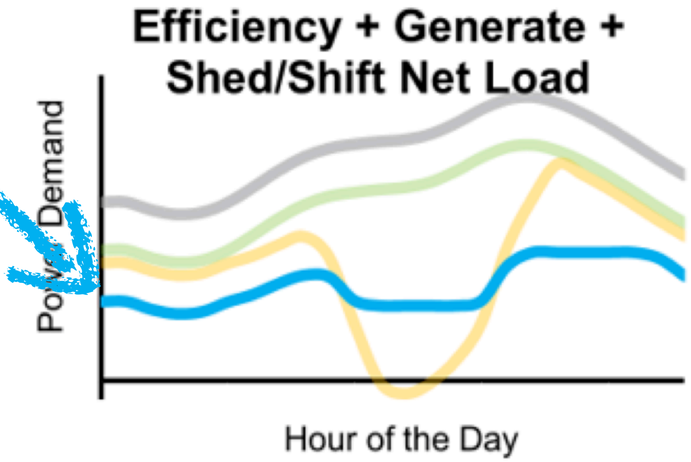
icons created by noomtah and Freepik

How do we get to here?



Storage

Electric Battery
Domestic Hot Water
Thermal Storage
(Passive & active)

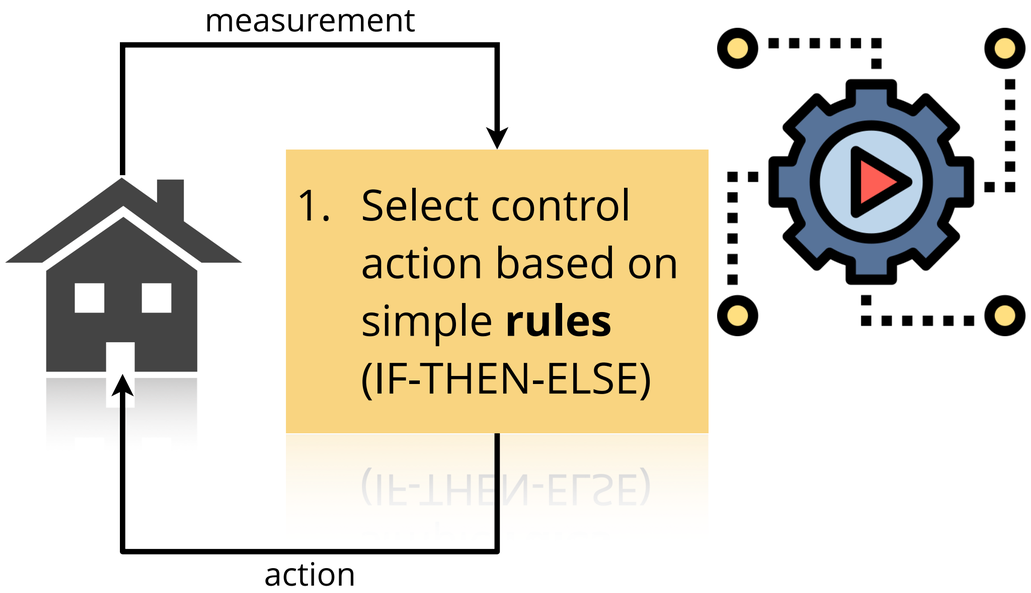


US Dept of Energy, 2019

icons created by noomtah and Freepik

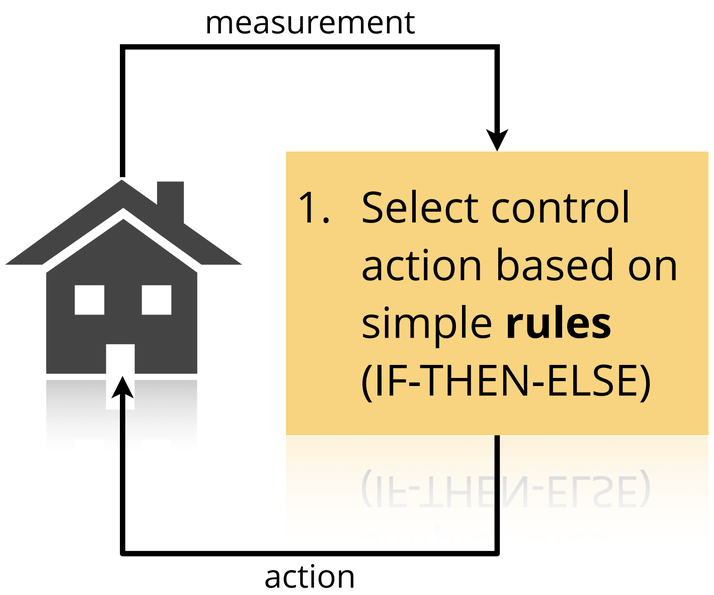
Control strategies for building energy management

Rule based control (RBC)

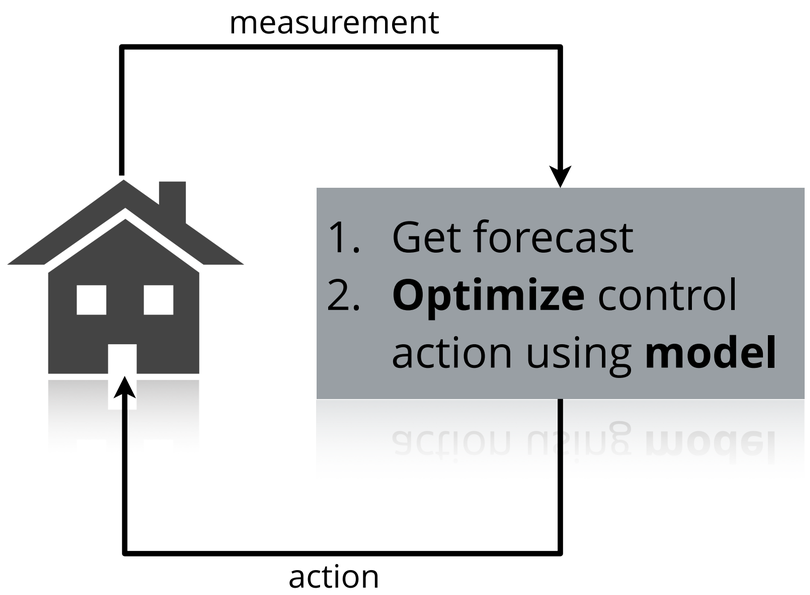


Control strategies for building energy management

Rule based control (RBC)



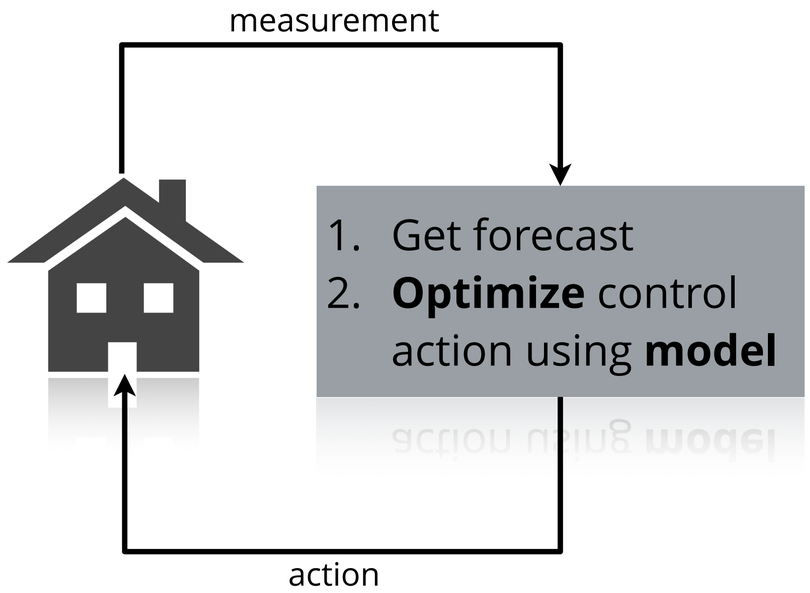
Model Predictive Control (MPC)



Control strategies for building energy management

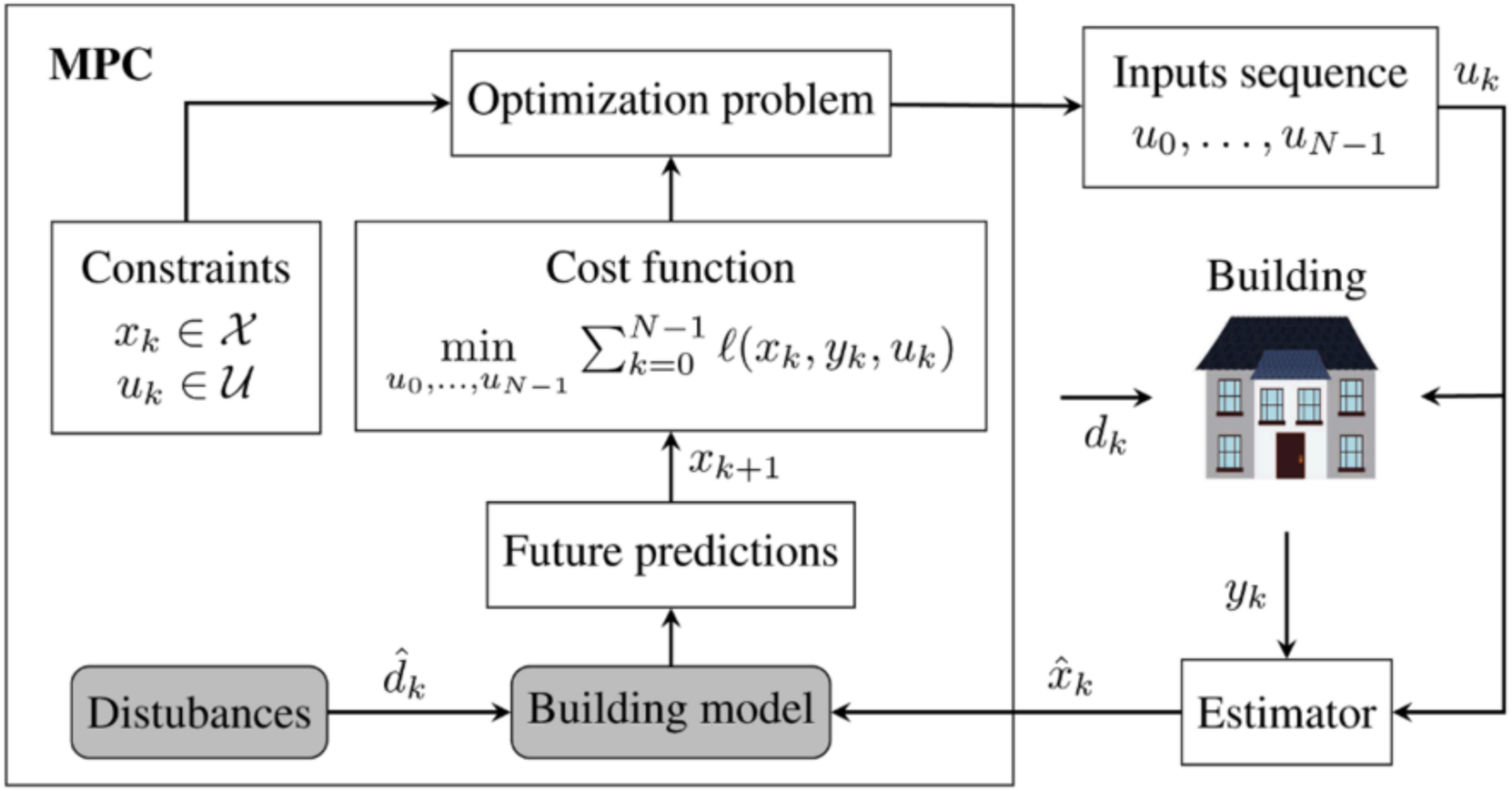
Rule based control (RBC)

Model Predictive Control (MPC)



Reinforcement Learning Control (RLC) / AI

All you Need to Know About Model Predictive Control



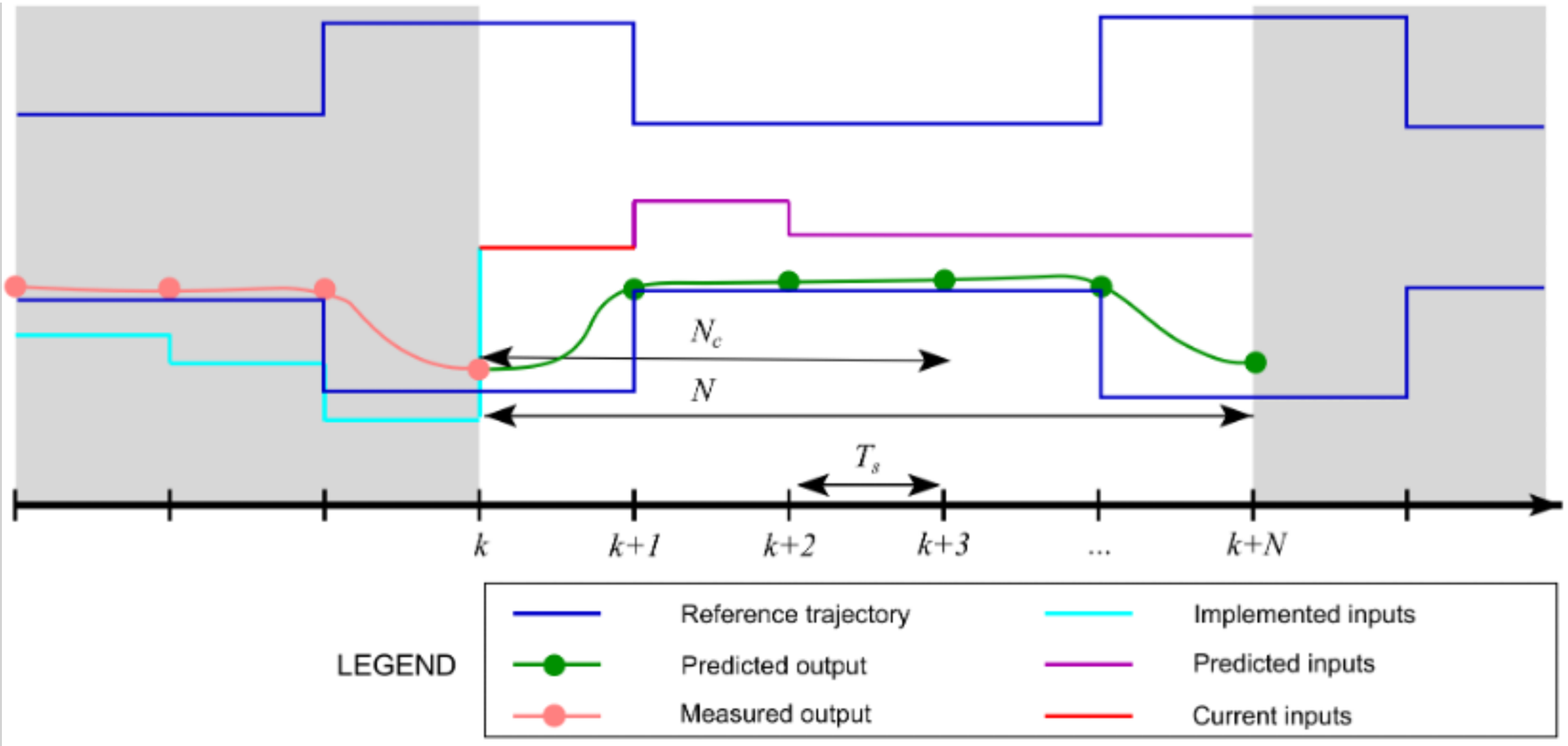
$$\min_{u_0,\ldots,u_{N-1}} \ell_N(x_N) + \sum_{k=0}^{N-1} \ell_k(x_k, y_k, r_k, u_k, s_k)$$

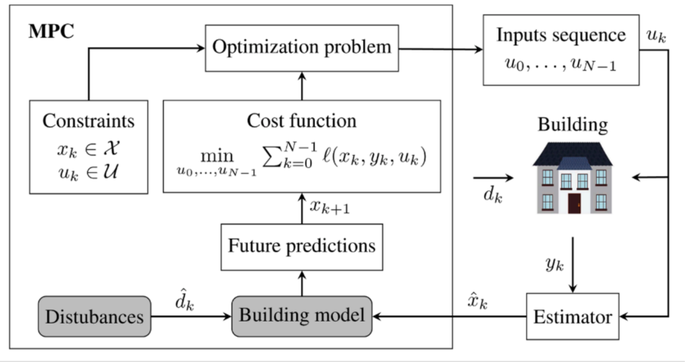
$$\text{s.t. } x_{k+1} = f(x_k, u_k, d_k), k \in \mathbb{N}_0^{N-1}$$

$$y_k = g(x_k, u_k, d_k), k \in \mathbb{N}_0^{N-1}$$

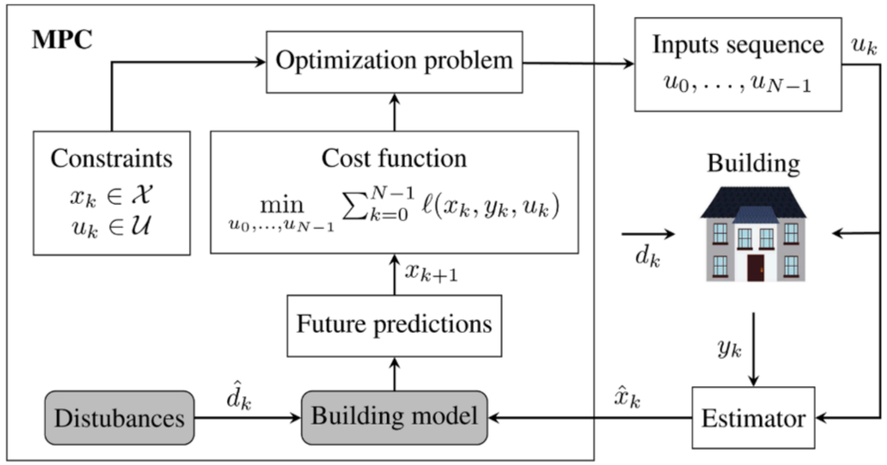
$$u_k = f_{\text{HVAC}}(x_k, a_k, m_k), k \in \mathbb{N}_0^{N-1}$$

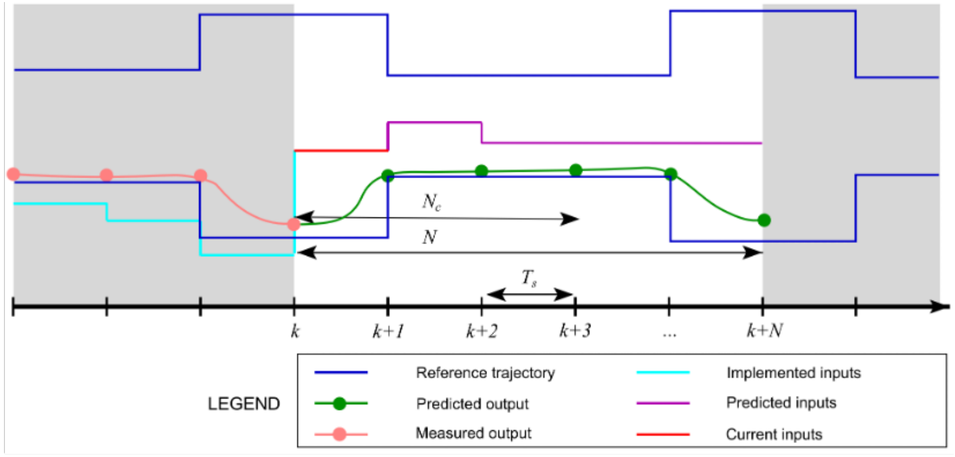
All you Need to Know About Model Predictive Control

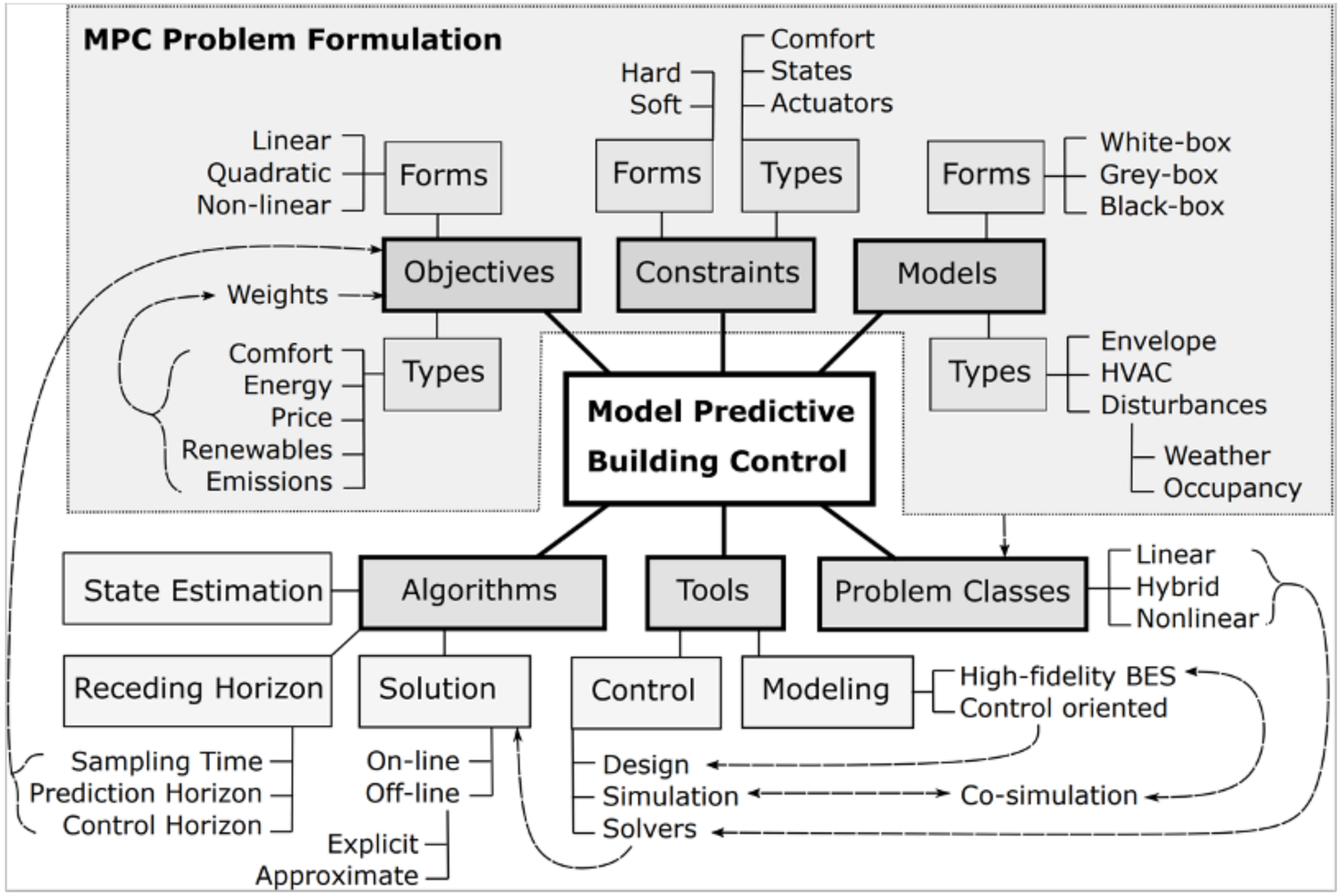




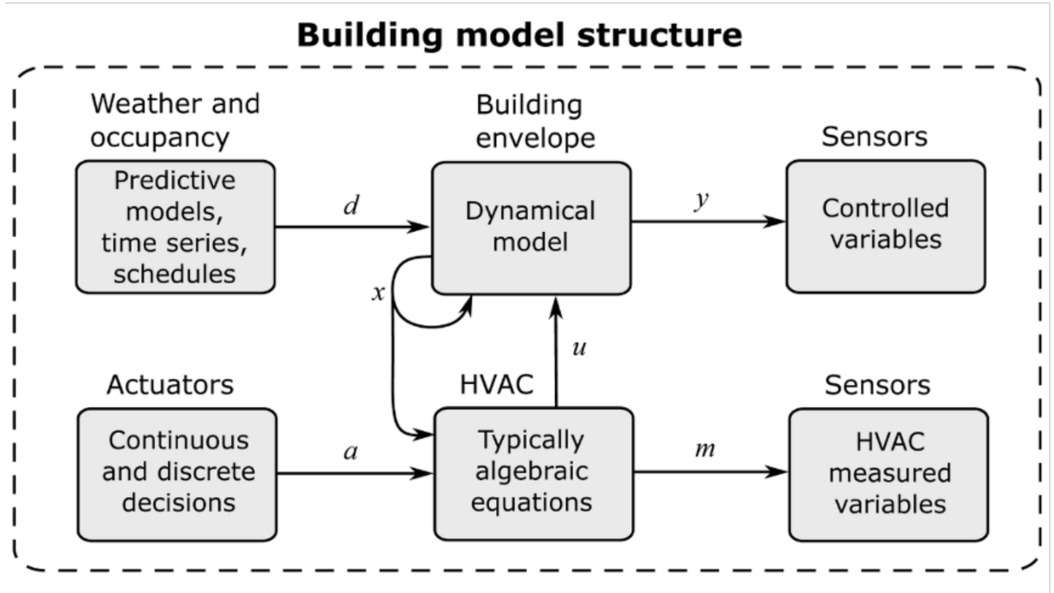
All you Need to Know About Model Predictive Control

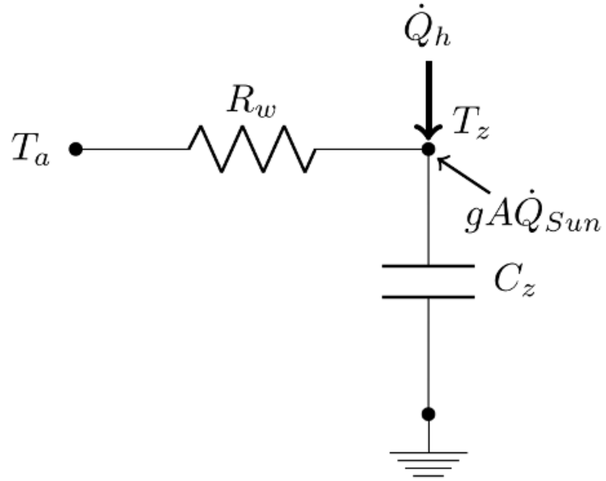






All you Need to Know About Model Predictive Control

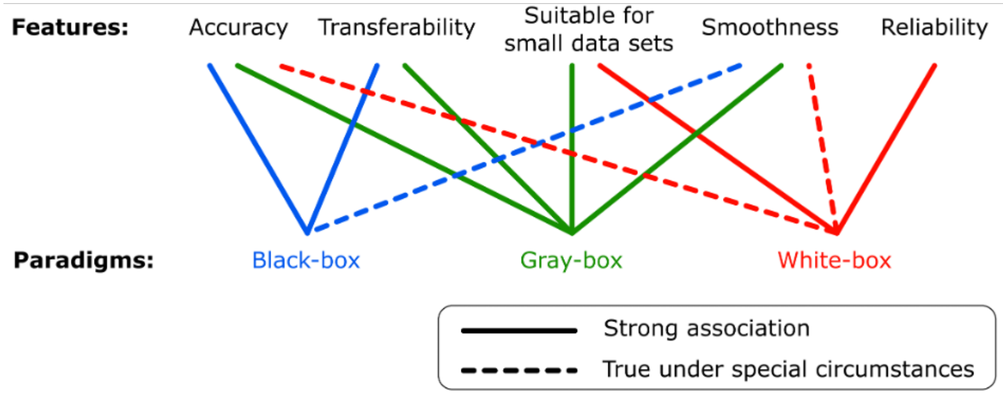




$$C_z \frac{dT_z}{dt} = \frac{T_a - T_z}{R_w} + \dot{Q}_h + gA\dot{Q}_{Sun}$$

$$x_{k+1} = Ax_k + Bu_k + Ed_k + w_k,$$

$$y_k = Cx_k + Du_k + v_k,$$



Break