Scalable Distributed Model Predictive Control for Building and Renewable Energy Systems

Christopher Bay, Rohit Chintala, Jennifer King, and Venkatesh Chinde
06.20.2020
Overview

Buildings

AES Framework

P,Q Limits  P,Q Setpoints  P,Q
Overview

Building Controllers

- Building Info
- Setpoints
- P,Q Limits
- P,Q Setpoints
- Grid Info

Grid Controller

Building Models

- P,Q

Power System Model

NREL | 3
Modeling and Control of Buildings

• Building control traditionally focused on energy reduction and occupant comfort

• Need for buildings to provide ancillary services (while keeping people happy)

• Need for coordinating large numbers of building energy systems

• MPC lends itself to a lot of the challenges found in buildings

• Technique presented today has also been applied to wind farm control
Modeling and Control of Buildings

- Distributed control with the Limited-Communication Distributed MPC method (LC-DMPC)

- IEEE 13 Node Test Feeder consisting of building nodes
Modeling and Control of Buildings

- Distributed control with the Limited-Communication Distributed MPC method (LC-DMPC)
- IEEE 13 Node Test Feeder consisting of building nodes

A few key notes:
- Distributed algorithm can scale computationally beyond centralized methods
- Subsystems don’t require model knowledge of other subsystems (robust to changes, modular)
- Can be done in hierarchical manner, with local and supervisory setups
Limited-Communication Distributed MPC

PAST  FUTURE/PREDICTION

MEASURED OUTPUT

CONTROL

REFERENCE

PREDICTED OUTPUT

OPTIMAL CONTROL TRAJECTORY

CONTROL HORIZON $c$

PREDICTION HORIZON $p$

$k$  $k + c$  $k + p$
LC-DMPC: The method

- Divide system into subsystems with local models
- Establish prediction horizons
- Identify connections between subsystems

\[
x_i(k+1) = A_i x_i(k) + B_{u,i}(k) + B_{v,i}(k)
\]

\[
y_i(k) = C_{y,i} x_i(k) + D_{y,i} u_i(k)
\]

\[
z_i(k) = C_{z,i} x_i(k) + D_{z,i} u_i(k)
\]

\[
Y_i = \begin{bmatrix} y_i^T(k+1) & y_i^T(k+2) & \cdots & y_i^T(k+N_{p,i}) \end{bmatrix}^T
\]

\[
Z_i = \begin{bmatrix} z_i^T(k+1) & z_i^T(k+2) & \cdots & z_i^T(k+N_{p,i}) \end{bmatrix}^T
\]

\[
V_i = \begin{bmatrix} v_i^T(k+1) & v_i^T(k+2) & \cdots & v_i^T(k+N_{p,i}) \end{bmatrix}^T
\]

\[
N_{p,i} \text{ is the prediction horizon.}
\]

\[
V = \Gamma Z
\]

\[
\Gamma \text{ is the interconnection matrix.}
\]
LC-DMPC: The method

• By repeated application of the local model along $Np$, the future dynamics for $Y$ and $Z$ can be found.

• These prediction matrices are built for each subsystem.

• $N_y / N_z$ and $P_y / P_z$ are the same as $M_y / M_z$ with $B_u$ replaced by $B_v / B_d$.

$$Y_i = F_{y,i} x_{0,i} (k) + M_{y,i} U_i + N_{y,i} V_i + P_{y,i} D_i$$

$$Z_i = F_{z,i} x_{0,i} (k) + M_{z,i} U_i + N_{z,i} V_i + P_{z,i} D_i$$

$$F_{y,i} = \begin{bmatrix} (C_{y,i} A_i)^T & (C_{y,i} A_i^2)^T & \cdots & (C_{y,i} A_i^{N_p})^T \end{bmatrix}^T$$

$$F_{z,i} = \begin{bmatrix} (C_{z,i} A_i)^T & (C_{z,i} A_i^2)^T & \cdots & (C_{z,i} A_i^{N_p})^T \end{bmatrix}^T$$

$$M_{y,i} = \begin{bmatrix} D_{y,i} & 0 & \cdots & 0 \\ C_{y,i} B_{u,i} & D_{y,i} & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 \\ C_{y,i} A_i^{N_p-2} B_{u,i} & C_{y,i} A_i^{N_p-3} B_{u,i} & \cdots & D_{y,i} \end{bmatrix}$$

$$M_{z,i} = \begin{bmatrix} D_{z,i} & 0 & \cdots & 0 \\ C_{z,i} B_{u,i} & D_{z,i} & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 \\ C_{z,i} A_i^{N_p-2} B_{u,i} & C_{z,i} A_i^{N_p-3} B_{u,i} & \cdots & D_{z,i} \end{bmatrix}$$
LC-DMPC: The optimization

\[
\begin{align*}
\min_{U_i} J_i &= e_i^T Q_i e_i + U_i^T S_i U_i + \Psi_i^T Z_i \\
\text{s.t.} \quad Y_i &= F_{y,i} x_{0,i} (k) + M_{y,i} U_i + N_{y,i} \dot{V}_i \\
Z_i &= F_{z,i} x_{0,i} (k) + M_{z,i} U_i + N_{z,i} \dot{V}_i \\
U_{i_{\min}} \leq U_i \leq U_{i_{\max}}.
\end{align*}
\]

- Objective function has quadratic error and control terms with a linear penalty term
- Local dynamics can be moved into objective function from constraints
- Sensitivities are calculated based on upstream system disturbances

\[
\begin{align*}
\min_{U_i} J_i &= U_i^T H_i U_i + 2U_i^T F_i + V_i^T E_i V_i + 2V_i^T T_i \\
\text{s.t.} \quad A_i U_i &\leq B_i \\
H_i &= M_{y,i}^T Q_{i} M_{y,i} + S_i, \quad E_i = N_{y,i}^T Q_{i} N_{y,i} \\
F_i &= M_{y,i}^T Q_{i} \left[ F_{y,i} x_{0,i} (k) + N_{y,i} \dot{V}_i + P_{y,i} D_i - r_i (k) \right] + 0.5 M_{z,i}^T \Psi_i \\
T_i &= N_{y,i}^T Q_{i} \left[ F_{y,i} x_{0,i} (k) - r_i (k) \right] + 0.5 N_{z,i}^T \Psi_i \\
A_i &= \text{diag} \left( \begin{bmatrix} I_{i \times N_p} \\ -I_{i \times N_p} \end{bmatrix} \right), \quad B_i = \begin{bmatrix} U_{i_{\max}}^T \\ U_{i_{\min}}^T \end{bmatrix}^T \\
\gamma_{i+1} &= \frac{\partial J_{i+1}}{\partial V_{i+1}} = 2 \left[ E_i V_{i+1} + T_{i+1} + N^T_{y,i} Q_{i+1} M_{y,i} U_{i+1} \right] \\
\Psi &= \left[ \Psi_1^T, \Psi_1^T, \ldots, \Psi_p^T \right]^T = \Gamma^T \left[ \left[ \gamma_1^T, \gamma_1^T, \ldots, \gamma_p^T \right]^T \right] = \Gamma^T \gamma
\end{align*}
\]

(Jalal, et al., 2016)
LC-DMPC: The optimization

- Objective function has quadratic error and control terms with a linear penalty term
- Local dynamics can be moved into objective function from constraints
- Sensitivities are calculated based on upstream system disturbances

Algorithm 1 LC-DMPC Algorithm

Initialization: Given $x_{0,i}(k)$ & $N_a$, $V_i(0), U_i(0), \Psi_i(0) = 0$.

Step 1: Exchange current information with local agents:
$$V(j + 1) = \Gamma Z(j), \quad \Psi(j + 1) = \Gamma^T \Psi(j)$$

Step 2: Solve problem (13) and assign the result as $U_i^{QP}$.

Step 3: Compute the convex summation for $\beta \in [0,1)$:
$$U_i(j + 1) = \beta U_i(j) + (1 - \beta) U_i^{QP}(j)$$

Step 4: Use the result from step 3 to compute:
$$Z_i(j + 1) = F_{z,i} x_{0,i}(k) + M_{z,i} U_i(j + 1) + N_{z,i} V_i(j)$$

Step 5: Use the result from step 3 to compute:
$$\gamma_i(j + 1) = -2N_{y,i}^T Q_i r_i(k) + 2N_{y,i}^T Q_i M_{y,i} U_i(j + 1) + 2N_{y,i}^T Q_i N_{y,i} V_i(j) + N_{z,i} \Psi_i(j) + 2N_{y,i}^T Q_i F_{y,i} x_{0,i}(k)$$

Step 6: If $j \neq N_a$ go to step 1, otherwise go to step 7.

Step 7: Apply the first value of $U_i$.

Step 8: Get new measurements for $x_{0,i}$ and go to step 1.

(Jalal, et al., 2016)
The Model

- IEEE 13 Node Test Feeder consisting of building nodes
- Added grid aggregator to distribute reference signal from the grid
- Grid aggregator is at same level as buildings, not hierarchical
Interconnections

- Grid aggregator is both upstream and downstream to each building

Upstream

Downstream

- Electrical Connections

![Diagram showing electrical connections and grid aggregator locations]
Interconnections

- Grid aggregator is both upstream and downstream to each building

Upstream  Downstream

$P_{ref,1}$  $P_1$
Interconnections

- Grid aggregator is both upstream and downstream to each building

Communication Connections

Upstream  Downstream

\[ \gamma_{bldg, i} \]

\[ \gamma_{grid, i} \]
A bulk reference signal is sent from the grid to the feeder.

The grid aggregator determines power references for the buildings:

- Model includes summation of individual building powers.
- Optimization chooses reference signals such that the reference tracking error is minimized.
- Through iterative communication, aggregator and buildings come to consensus on control actions.
Building Model

- Used DOE Large Office Building Model
- For first implementation, used the ground floor
  - Lumped the 5 zones into 1 zone
  - Equipment consists of 1 AHU
  - Only considered cooling
- Used to generate truth model
- Control model was then identified from the truth model
EKF-based prediction model

EKF-based approach adopted to make RC models feasible for real world implementation

- 3R-2C model used to describe building thermodynamics

\[
T_{in}(k + 1) = T_{in}(k) + \frac{t_s}{R_{in,e} \cdot C_{in}} (T_e - T_{in}) + \frac{t_s}{R_{in,a} \cdot C_{in}} (T_a - T_{in}) + \frac{t_s}{C_{in}} (Q_{solar} + Q_{internal} + Q_{hvac} + Q_{inf})
\]

\[
T_{e}(k + 1) = T_{e}(k) + \frac{t_s}{R_{in,e} \cdot C_{e}} (T_{in} - T_{e}) + \frac{t_s}{R_{e,a} \cdot C_{e}} (T_a - T_{in}) + \frac{t_s}{C_{e}} (Q_{solar} + Q_{internal} + Q_{hvac} + Q_{inf})
\]

- \(T_{in}\) - Indoor air temperature
- \(T_e\) - Exterior wall temperature
- \(T_{a}\) - Outdoor air temperature
- \(t_s\) - Duration of simulation time step
- \(k\) – Current time step
- \(R_{in,a}, R_{in,e}, R_{e,a}\) - Equivalent resistances
- \(C_{in}, C_{e}\) - Equivalent capacitance values
- \(Q_{solar}\) - Solar heat gain through windows
- \(Q_{internal}\) - Internal heat gain
- \(Q_{inf}\) - Infiltration heat load
- \(Q_{hvac}\) - Cooling or heating energy delivered by the HVAC system
EKF-based prediction model

Initial modeling assumptions

• $Q_{solar}$ is assumed to bear a simple relationship with $Q_{ghi}$
  \[ Q_{solar} = \alpha \cdot Q_{ghi} \]

• Effect of wind on $Q_{inf}$ is not captured by the model
  \[ Q_{inf} \propto (T_a - T_{in}) \]

• $Q_{internal}$ is known to us.
EKF-based prediction model

EKF algorithm

• State-space representation of the 3R-2C model

\[
\begin{align*}
x(k+1) &= Ax(k) + Bu(k) \\
y(k) &= Cx(k)
\end{align*}
\]

• Model parameters are represented as states of the equation

\[x = [T_{in}, T_e, C_{in}, R_{in,e}, R_{in,a}, C_e, R_{e,a}, \alpha]\]

• One-month historical data of indoor air temperature and weather information is used to train the data.

• Discrepancy between measured and predicted values of \(T_{in}\) are used to update the initial estimates of the states
EKF Algorithm

EKF Pseudocode

\[
\text{for } k = 1:n_{\text{train}} \\
\text{if } k = 1 : \\
\quad x_k := \text{x}_{\text{init}} \quad \text{(Initial state estimates)} \\
\quad e_{\text{mse-old}} = 100 \quad \text{(initial state mean squared error)} \\
\text{else:} \\
\quad e_{\text{mse}} = \left( \frac{1}{n_{\text{val}}} \right) \sum_{i=1}^{n_{\text{val}}} \left( T_{\text{in}}(i + n_{\text{pred}}) - Hx_{i+n_{\text{pred}}i} \right)^2 \\
\quad \text{if } e_{\text{mse}} < e_{\text{mse-old}}: \\
\quad \quad e_{\text{mse-old}} = e_{\text{mse}} \\
\quad \quad x_k := x_{k|k} \quad \text{(measurement update)} \\
\quad \text{else:} \\
\quad \quad x_k(1:2) = x_{k|k}(1:2) \quad \text{(measurement update only for temperature states)} \\
\quad x_k := x_{k+1|k} \quad \text{(time update)}
\]

EKF Matrices and Equations

\[
x = [T_{\text{in}}, T_e, C_{\text{in}}, R_{\text{in,a}}, R_{\text{in,e}}, C_e, R_{e,a}, \alpha]
\]

\[
h = T_{\text{in}}, u = [T_{\text{oa}}, \dot{Q}_{\text{gh}}, \dot{Q}_{\text{heat}}]
\]

\[
\dot{Q}_{\text{heat}} = \dot{Q}_{\text{internal}} + \dot{Q}_{\text{inf}} + \dot{Q}_{\text{hvac}}
\]

\[
f(x_k, u_k) \equiv \text{derived from 3R2C Model}
\]

Measurement Update

\[
H = \left. \frac{\partial h(x_k, u_k)}{\partial x} \right|_{x_k,u_k}
\]

\[
y_k = T_{\text{in}}(k) - Hx_{k|k-1}
\]

\[
K = P_{k|k-1} H^T (HP_{k|k-1}H^T + R)^{-1}
\]

\[
x_{k|k} = x_{k|k-1} + Ky_k
\]

\[
P_{k|k} = P_{k|k-1} + Ky_k
\]

Time Update

\[
\dot{x}_{k+1|k} = f(x_k, u_k)
\]

\[
F = \left. \frac{\partial f(x_k, u_k)}{\partial x} \right|_{x_k,u_k}
\]

\[
y = \left. \frac{\partial f(x_k, u_k)}{\partial u} \right|_{x_k,u_k}
\]

\[
P_{k+1|k} = FP_{k|k}F^T + VMV^T
\]
• 5 Zone building modeled as a single zone.
Linear Parametric Model for MPC

- Linear parametric equation for building envelope modeling
  - ARX model structure to predict room temperature dynamics
    \[ y(k) = a_1 y(k-1) + a_2 y(k-2) + \ldots + a_{n_a} y(k-n_a) + \\
    b_1 u(k-1) + b_2 u(k-2) + \ldots + b_{n_b} (k-n_b) + e(t) \]
  - System identification to find the parameters \( \theta \) of the ARX model.
    \[ \theta = [a_1, a_2, \ldots, a_{n_a}, b_1, b_2, \ldots, b_{n_b}] \]

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_{oa} ) (Outside air temperature)</td>
<td></td>
</tr>
<tr>
<td>( Q_{int} ) (Internal convective heat gain)</td>
<td>( T_{in} )</td>
</tr>
<tr>
<td>( Q_{ghi} ) (Solar heat gain)</td>
<td></td>
</tr>
<tr>
<td>( Q_{hvac} ) (Sensible heat from HVAC system)</td>
<td></td>
</tr>
</tbody>
</table>
Building model has 2 power consuming components:
  • AHU Fan
  • Chiller

Truth model uses non-linear equations shown on the right

Controller model uses linearized version of the equations around the current operating point
Preliminary Results

- Tested first with one building node tracking a power reference.
- Additionally, buildings able to maintain temperature within constraints even when power reference exceeds capabilities.
Preliminary Results Cont.

**Building Temperatures**

- Temperature (°C)
- Time
- Bldg 0
- Bldg 1
- Bldg 2
- Bldg 3
- Bldg 4
- Bldg 5
- Bldg 6
- Bldg 7
- Bldg 8
- Bldg 9
- Bldg 10
- Bldg 11
- Outdoor Air Temp

**Total Power**

- Power [kW]
- Time
- Bldg Power
- Power Ref Stpts
- Grid Power Ref

**Building Powers**

- Power [kW]
- Time
- Bldg 0
- Bldg 1
- Bldg 2
- Bldg 3
- Bldg 4
- Bldg 5
- Bldg 6
- Bldg 7
- Bldg 8
- Bldg 9
- Bldg 10
- Bldg 11

**Power Error**

- Error [%]
- Time
- Error Curves

NREL | 27
Conclusions

• Grid aggregator allows for buildings to “pushback” with own objectives
• Used novel EKF approach for building model
• LC-DMPC allows for systems to be both upstream and downstream agents (mesh networks)
• Method can be used at different levels of systems
• Implement voltage constraints/reactive power
• Convergence studies for communication iterations/beta
• Reduce power model time-step and aggregate control actions to reflect thermodynamics at a larger time-step
• Examine higher fidelity truth modelling; use machine learning/data-driven techniques for control models
References


Questions?

www.nrel.gov

christopher.bay@nrel.gov
rohit.chintala@nrel.gov