Hierarchical MPC control for Plug-and-Play resource distribution

Jakob Stoustrup

Automation and Control
Department of Electronic Systems
Faculties of Engineering, Science & Medicine
Aalborg University
`jakob@es.aau.dk`
`http://es.aau.dk/staff/jakob`

May 20, 2010

Outline

1. Problem formulation

Outline

1. Problem formulation
2. Proposed solution

Outline

1. Problem formulation
2. Proposed solution
3. Smart Grid Example

Outline

1. Problem formulation
2. Proposed solution
3. Smart Grid Example
4. Conclusions

Outline

1. Problem formulation
2. Proposed solution
3. Smart Grid Example
4. Conclusions

Distributed resource control

By a distributed resource control system we shall understand a system for which:

Distributed resource control

By a distributed resource control system we shall understand a system for which:

- The system has a number of decentral storages that can each store a certain amount of some resource

Distributed resource control

By a distributed resource control system we shall understand a system for which:

- The system has a number of decentral storages that can each store a certain amount of some resource
- Each storage can be filled or emptied at some maximal rate(s)

Distributed resource control

By a distributed resource control system we shall understand a system for which:

- The system has a number of decentral storages that can each store a certain amount of some resource
- Each storage can be filled or emptied at some maximal rate(s)
- A central controller has the responsibility of balancing supply and demand by use of the storages

Problem formulation

Problem

Find an architecture and appropriate algorithms for a distributed resource control system for which:

Problem formulation

Problem

Find an architecture and appropriate algorithms for a distributed resource control system for which:

- *Stability is obtained*

Problem formulation

Problem

Find an architecture and appropriate algorithms for a distributed resource control system for which:

- *Stability is obtained*
- *Quadratic performance is guaranteed*

Problem formulation

Problem

Find an architecture and appropriate algorithms for a distributed resource control system for which:

- *Stability is obtained*
- *Quadratic performance is guaranteed*
- *The solution is scalable*

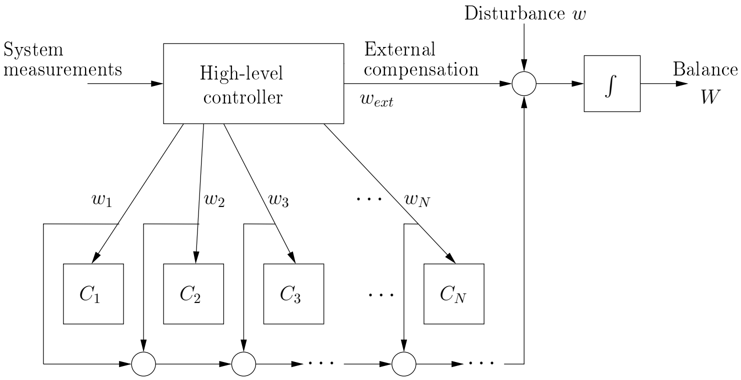
Problem formulation

Problem

Find an architecture and appropriate algorithms for a distributed resource control system for which:

- *Stability is obtained*
- *Quadratic performance is guaranteed*
- *The solution is scalable*
- *The solution supports plug-and-play of subsystems*

Problem setup



High-level controller

The high-level controller must solve the following (discrete-time) optimisation problem at any given time t:

$$
\begin{aligned}
\min_{w_i, w_{\text{ext}}} \quad & \sum_{k=1}^{N_h} \rho W_k^2 + \phi(w_{\text{ext},k}, w_{\text{ext},k-1}) \\
\text{s.t.} \quad & \underline{W} \leq W_k \leq \bar{W} \\
& \underline{w}_i \leq w_{i,k} \leq \bar{w}_i, 1 \leq i \leq N
\end{aligned}
$$

$\underline{W}$, $\bar{W}$: Constraints on the balance,

ρ: Scalar cost on balance

$\phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$: A cost function of the absolute value of w_{ext} as well as changes in w_{ext}.

N_h: is the prediction horizon of the controller.

Storage model

Consumption by the storages are for simplicity assumed cost-free. Furthermore, it is assumed that each storage can be modelled as a marginally stable linear systems.
Each storage is characterized by its own state equation

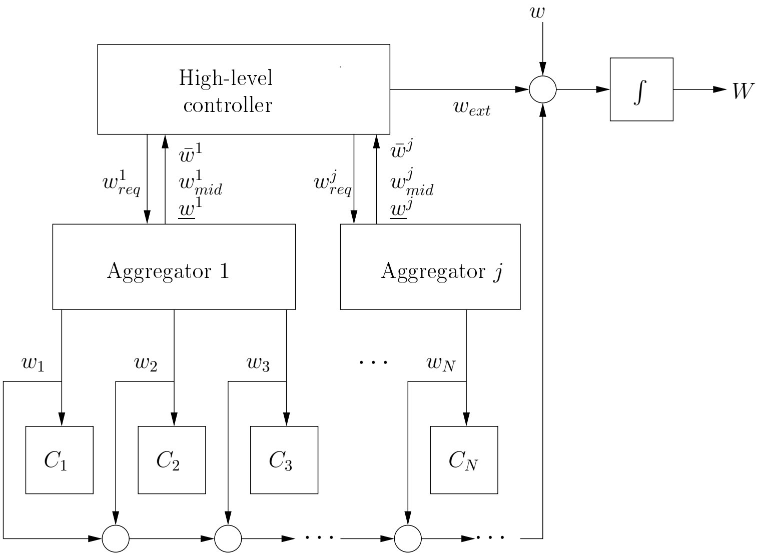
$$\frac{dW_i(t)}{dt} = w_i(t)$$

which must satisfy $0 \leq W_i(t) \leq \bar{W}_i$ at all times.

Furthermore, each storage has limits on how much of the resource it will accept at any given time, $\underline{w}_i \leq w_i(t) \leq \bar{w}_i$.

Outline

Modified architecture



Resource balance

The main objective of the high-level control is to keep the resource balance governed by

$$\frac{dW(t)}{dt} = w_{\text{ext}}(t) - w(t) - w_a(t)$$

at zero.

$w(t)$: An external disturbance

$w_a(t) = \sum_i^N w_i(t)$: The resource absorbed by the intelligent storages $(C_i, 1 \leq i \leq N)$.

It is assumed that the top level controller can control $w_{\text{ext}}(t)$ directly and constrained only by a rate limit, but we would also like to keep the time derivative small.

The aggregator

At each sample, the aggregator solves the simple optimisation problem of

$$\min_{w_i} \sum (W_i(t+T_s) - W_{i,\mathrm{ref}})^2,$$

s.t.

$$\begin{array}{l} \sum_i w_i = w_{\mathrm{req}}, \\ \underline{w}_i \le w_i(t) \le \bar{w}_i \\ 0 \le W_i(t+T_s) \le \bar{W}_i \end{array}$$

with $W_i(t+T_s) = W_i(t) + T_s w_i$, where T_s is the sampling time.

The aggregator

In short the objective of each aggregator is to make sure that:

- The maximum capacity is available for the upper level at any time instance

The aggregator

In short the objective of each aggregator is to make sure that:

- The maximum capacity is available for the upper level at any time instance
- The load for storages is equalized over the number of storages

The aggregator

In short the objective of each aggregator is to make sure that:

- The maximum capacity is available for the upper level at any time instance
- The load for storages is equalized over the number of storages
- The deviation from the nominal consumption is minimized for each storage

The aggregator

In short the objective of each aggregator is to make sure that:

- The maximum capacity is available for the upper level at any time instance
- The load for storages is equalized over the number of storages
- The deviation from the nominal consumption is minimized for each storage
- The capacity constraint for each storage is not violated

The aggregator

In short the objective of each aggregator is to make sure that:

- The maximum capacity is available for the upper level at any time instance
- The load for storages is equalized over the number of storages
- The deviation from the nominal consumption is minimized for each storage
- The capacity constraint for each storage is not violated
- The rate constraint for each storage is not violated

Stability

First, in order to assess stability, it is of course necessary to establish a sensible definition, especially as the system has constraints with the usual implications for stability.
Intuitively, stability for a system of the type described above will comprise the following features:

1. For constant inputs, all trajectories will tend to constant values

Stability

First, in order to assess stability, it is of course necessary to establish a sensible definition, especially as the system has constraints with the usual implications for stability. Intuitively, stability for a system of the type described above will comprise the following features:

1. For constant inputs, all trajectories will tend to constant values
2. In steady state, a minimal number of constraints will be invoked

Stability

First, in order to assess stability, it is of course necessary to establish a sensible definition, especially as the system has constraints with the usual implications for stability.
Intuitively, stability for a system of the type described above will comprise the following features:

1. For constant inputs, all trajectories will tend to constant values
2. In steady state, a minimal number of constraints will be invoked
3. Wind-up behavior of system states is avoided for any bounded input set

Complexity

In terms of complexity, the proposed design methodology scales in the same way as quadratic optimization, which is $O(N^2)$, where N is the number of storages.

It should be noted, however, that:

Complexity

In terms of complexity, the proposed design methodology scales in the same way as quadratic optimization, which is $O(N^2)$, where N is the number of storages.

It should be noted, however, that:

- The optimization problem has a high degree of sparsity.

Complexity

In terms of complexity, the proposed design methodology scales in the same way as quadratic optimization, which is $O(N^2)$, where N is the number of storages.

It should be noted, however, that:

- The optimization problem has a high degree of sparsity.
- This has not been exploited in the implementation applied in the simulations below, but . . .

Complexity

In terms of complexity, the proposed design methodology scales in the same way as quadratic optimization, which is $O(N^2)$, where N is the number of storages.

It should be noted, however, that:

- The optimization problem has a high degree of sparsity.
- This has not been exploited in the implementation applied in the simulations below, but . . .
- . . . it should be expected that the complexity could be further reduced by implementing a dedicated quadratic programming solver, which exploits the sparsity.

Complexity

In terms of complexity, the proposed design methodology scales in the same way as quadratic optimization, which is $O(N^2)$, where N is the number of storages.

It should be noted, however, that:

- The optimization problem has a high degree of sparsity.
- This has not been exploited in the implementation applied in the simulations below, but . . .
- . . . it should be expected that the complexity could be further reduced by implementing a dedicated quadratic programming solver, which exploits the sparsity.
- If the quadratic cost function is not central, a linear cost can be applied at the aggregator level, making the complexity in no. of storages linear.

Performance

In terms of performance, the following five parameters are decisive:

Performance

In terms of performance, the following five parameters are decisive:

- The prediction horizon

Performance

In terms of performance, the following five parameters are decisive:

- The prediction horizon
- The total installed flexible capacity

Performance

In terms of performance, the following five parameters are decisive:

- The prediction horizon
- The total installed flexible capacity
- The instantaneous flexible capacity

Performance

In terms of performance, the following five parameters are decisive:

- The prediction horizon
- The total installed flexible capacity
- The instantaneous flexible capacity
- The total cumulative rate limitation of flexible units

Performance

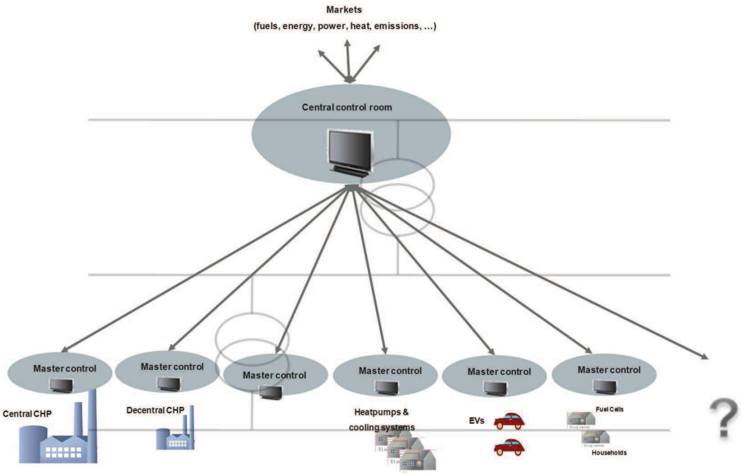
In terms of performance, the following five parameters are decisive:

- The prediction horizon
- The total installed flexible capacity
- The instantaneous flexible capacity
- The total cumulative rate limitation of flexible units
- The instantaneous cumulative rate limitation of flexible units

Outline

1. Problem formulation
2. Proposed solution
3. Smart Grid Example
4. Conclusions

Virtual Power Plants



Markets
(fuels, energy, power, heat, emissions, ...)

Central control room

Master control

Master control

Master control

Master control

Master control

Master control

Central CHP

Decentral CHP

Heatpumps & cooling systems

EVs

Fuel Cells

Households

?

Example: Smart Grid application

The main objective of the top level control is to keep the energy balance governed by

$$\frac{dE(t)}{dt} = P_{\text{ext}}(t) - P_{\text{load}}(t) - P_{\text{a}}(t) \tag{1}$$

at zero.

$P_a = \sum_i P_i$: Power absorbed by the intelligent consumers (ICs).

P_{load}: Power absorbed by other consumers (considered as a disturbance here).

P_{ext}: Power produced by a number of suppliers such as power plants etc.

It is assumed that the top level controller can control P_{ext} directly and restrained only by a rate limit, but we would also like to keep the time derivative small.

Example: Smart Grid application

Each intelligent consumer is characterized by its own energy balance

$$\frac{dE_i(t)}{dt} = P_i(t)$$

which must satisfy $0 \leq E_i(t) \leq \bar{E}_i$ at all times. Furthermore, each intelligent consumer can only consume a limited amount of power $\underline{P}_i \leq P_i(t) \leq \bar{P}_i$.

Optimization problem

At each sample, at time t, the aggregator solves the simple optimisation problem of

$$\begin{array}{l}\min_{P_i} \sum (E_i(t+T_s) - E_{i,\mathrm{ref}})^2, \\ \text{s.t.} \\ \sum P_i = P_{\mathrm{req}}, \\ \underline{P}_i \leq P_i(t) \leq \bar{P}_i, \\ 0 \leq E_i(t+T_s) \leq \bar{E}_i \end{array}$$

with $E_i(t+T_s) = E_i(t) + T_s P_i$, thereby distributing the power in a way that brings the energy levels as close to the reference as possible in a quadratic sense.

High level optimization

The top level control optimizes over a prediction horizon N_p. It minimizes the performance function

$$
\begin{aligned}
J_t = & \sum_{i=1}^{N_p} E(t + T_s i)^2 \\
& + \beta_p \sum_{i=1}^{N_c} (P_{\text{ext}}(t + T_s i) - P_{\text{ext}}(t + T_s(i-1)))^2 \\
& + \beta_r \sum_{i=1}^{N_c} (P_{\text{req}}(t + T_s i) - P_{\text{mid}}(t))^2
\end{aligned}
$$

with N_c samples of P_{ext} and P_{req} as decision variables.

Constraints

The optimisation is subject to constraints on the decision variables. There is a rate limit on the power from the power plants:

$$\underline{P}_{\mathrm{ext}} \leq P_{\mathrm{ext}}(t + T_s i) - P_{\mathrm{ext}}(t + T_s(i-1)) \leq \bar{P}_{\mathrm{ext}}$$

The aggregator provides limits on P_a that can be sustained over a horizon N_l. These limits are conservative in the sense that if P_{req} is for instance negative for the first part of the horizon, then a positive P_{req} higher than $\bar{P}$ may be feasible for the rest. However, in order to simplify the top level computations, the constraint

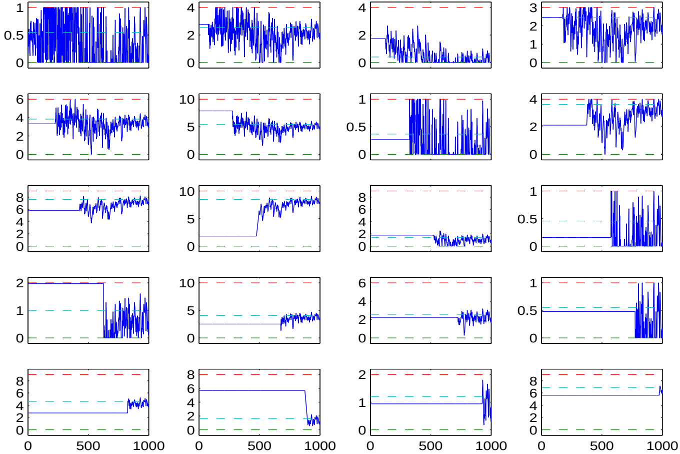
$$\underline{P}(t) \leq P_{\mathrm{req}}(t + T_s i) \leq \bar{P}(t)$$

is imposed over the whole horizon.

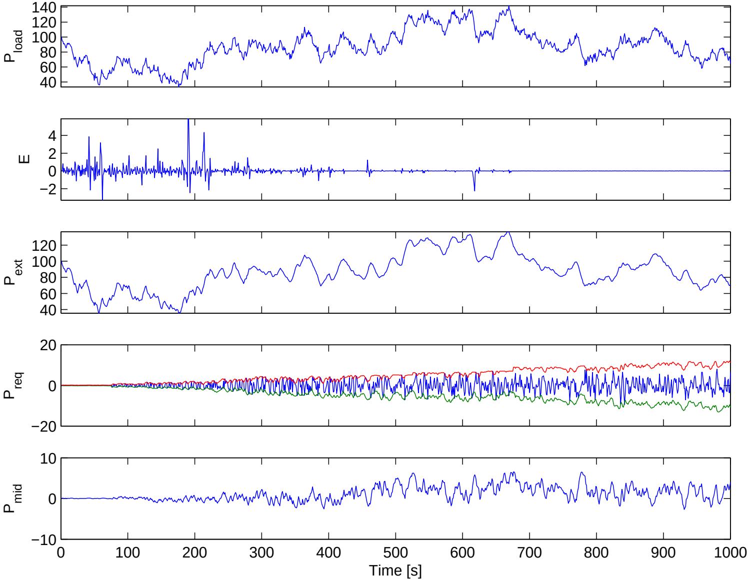
Parameters for 20 consumers in simulation

i	$\bar{E}_i$	$\underline{P}_i$	$\bar{P}_i$	i	$\bar{E}_i$	$\underline{P}_i$	$\bar{P}_i$
1	1.0	-1.7	1.4	11	9.0	-0.2	1.1
2	4.0	-1.4	0.8	12	1.0	-1.0	1.2
3	4.0	-0.2	1.8	13	2.0	-1.6	1.6
4	3.0	-1.3	0.3	14	10.0	-1.3	1.9
5	6.0	-1.6	0.9	15	6.0	-0.3	0.4
6	10.0	-1.3	1.1	16	1.0	-1.1	0.9
7	1.0	-0.7	1.2	17	9.0	-1.9	1.8
8	4.0	-1.9	0.2	18	8.0	-0.2	0.8
9	9.0	-1.1	0.2	19	2.0	-0.9	0.6
10	10.0	-1.1	0.2	20	9.0	-1.6	0.5

Simulation with aggregator



Simulation example



Outline

1. Problem formulation
2. Proposed solution
3. Smart Grid Example
4. Conclusions

Conclusions

Conclusions

- An MPC architecture for a distributed resource control system has been proposed.

Conclusions

- An MPC architecture for a distributed resource control system has been proposed.
- The proposed solution is scalable and can be implemented with a linear growth in complexity.

Conclusions

- An MPC architecture for a distributed resource control system has been proposed.
- The proposed solution is scalable and can be implemented with a linear growth in complexity.
- The proposed solution admits plug-and-play of new units simply by adding them in the aggregator tables.

Conclusions

- An MPC architecture for a distributed resource control system has been proposed.
- The proposed solution is scalable and can be implemented with a linear growth in complexity.
- The proposed solution admits plug-and-play of new units simply by adding them in the aggregator tables.
- The approach is sensitive to prediction horizon. This is not surprising by emphasizes the need for good load predictions.