AGREFLEX: notes on the mathematics of communication between the central and the local controllers

I. INTRODUCTION

Part of AGREFLEX is to try and implement a communication line between one central controller and a set of local controllers. Nowadays, standard grid operation relies on a central operator issuing informations/requests on next-day operation. Given weather forecast, historical records of production and consumption under similar circumstances (including weather forecast but also day of the week; time of the year; holidays; status of hydroelectric and other storage aso.), the grid operator gives predictions on consumption one day in advance and issues recommendation to producers. In this framework, there is no demand-side management. We want to incorporate load shifting into the game.

In the scheme discussed here, a central controller processes relevant informations – those mentioned above, augmented with a set of weights giving information on the expected discrepancy between production and demand, *if the latter were not adapted*, i.e. if it were acting independently of production. The hope is that this set of weights gives sufficient information to local load controllers so that the latter adapt the consumption profile to better suit the production curve. One goal is to figure out if providing minimal information to the local controllers is sufficient to have them react efficiently and adapt the loads without direct, centralized control. Below, we show mathematically that this is feasible. How well this will work in practice is still to be determined, however the results to be presented here are quite encouraging.

II. PRELIMINARY COMMENTS ON THE ACTION OF LOCAL CONTROLLERS

Generally speaking, one may think of forwarding some time-dependent "pricing function" to the local controllers, which the latter then try and minimize by adapting their loads, in particular the time at which they will be turned on and off, given general constraints such as deadlines for tasks to be finished, temperature brackets within which a building, a fridge aso. should remain, and so forth. To do so, we first discretize time, (for instance in $N = 96$ 15 mins. periods). The function to be minimized is expressed as $f(\{E_i\}) = \sum_{i=1}^{96} c_i E_i^a$, where $\{c_i\}$ is the set of "weight" and $\{E_i\}$ gives the loads to be adapted. Given a general constraint that the total daily consumption is predetermined, $\sum_{i=1}^{96} E_i = E_{\rm tot}$, the method of Lagrange multipliers (see en.wikipedia.org/wiki/Lagrange_multiplier) can be applied to give $E_i = E_{\rm tot}/\sum_j (c_i/c_j)^{1/(a-1)}$.

Our first observation is that the method fails if $a \leq 1$. For $a = 1$, the minimization gives $E_j = E_{\rm tot}$ if $c_j = \min_i\{c_i\}$ and $E_i = 0$ otherwise. For $a < 1$, it is even more advantageous to put all possible loads on the spot j with smallest c_j. This obviously does not reflect reality, as the electricity price depends on the consumption - piling up all loads on a single slot will result in a strong electricity price increase. Moreover, and perhaps more importantly for our purpose here, putting all loads on a single slot will not bring the consumption curve closer to the production curve.

Additionally, for a given set $\{c_i\}$, the method is more sensitive (gives more contrasted $\{E_i\}$) for smaller values of a (close to but above one) – the tendency of the E_i's to congregate persists. The optimal value of a to be used remains to be determined, not however that different choices of a by the local controller can be counterbalanced by a different choice $c_i \rightarrow c_i^{a-1}$ from the central controller. The choice of the minimization function will be left to the local controllers – for our purpose here, it is sufficient to note that, as just seen, the central controller can in principle adapt to any choice of the local controllers. From now on, we focus on the central controller, whose task will be to adapt its method for generating $\{c_i\}$ to the observed reaction of the local controllers, in particular the success (or lack thereof) of the resulting load shift.

III. CENTRAL CONTROLLER AND ITS ACTION

The main task of the central controller is thus to generate a set of weights $\{c_i\}$, $i = 1, 2, ...N$ being a discrete time variable. We first give a general sketch of how this can be done. At this stage, we restrict ourselves to day-ahead operation.

A. Inputs

The central controller gets the following information:
(i) next day weather forecast, 15 mins per 15 mins. These include temperature, wind and insolation data and so forth;
(ii) history of consumption, given the current and forecasted weather as well as other relevant factors (day of the week; holidays aso.);
(iii) current status of storages;
(iv) history of $\{c_i\}$ and, possibly, some measure (grading) on how successful they were.
Note: at this point (i.e. 01.04.2014) it is unclear that we will need (iv).

B. Processing of inputs

The forecasted next day production curve $\{P_i\}$ is determined from (i). Wind and insolation data, together with the geographic distributions of solar panels and wind turbines allows to forecast the pv and wind electricity production curve. To this, ribbon-like productions are added (geothermal, biomass, run-of-the-river hydroelectricity, as well as thermal, including nuclear electricity. Finally, the production of more flexible sources (mostly dammed hydro) has to be forecasted. This will be made based on day-before production, current and expected evolution of the filling of dams [point (iii) above] and other factors.

The forecasted next day consumption curve $\{E_i^{(0)}\}$ is determined from (ii). (I put a superindex $^{(0)}$ to indicate that this is the forecasted consumption, not the adapted consumptions to be optimized).

C. Generating the main output: passive case

The task is then to try and influence loads to operate in such a way as to modify the final consumption curve so that it gets closer to the production curve: one would like to increase consumption when the latter is predicted to be below production and decrease consumption when it is above production. This will be done by generating a set of weights $\{c_i\}$ which define a cost function $f(\{E_i\}) = \sum_i c_i E_i^a$ for the consumption E_i in each period $i = 1, ...N$. This function will be minimized by the local controllers. The goal of the central controller is to generate sets of $\{c_i\}$ giving optimal reaction from the local controllers.

In the initial operation stage, stored historical consumption curves [input (ii) above] are "passive ones", i.e. they were not influenced by demand-side management/load-shifts. At this level, one generates $\{c_i\}$ without information on previous attempts to do so, if these attempts/the generated sets were successful aso. The method we develop now for this case does not rely anyway on any knowledge of consumption, only on the forecasted production $\{P_i\}$; right now we do not need/include the history of success/failure of the approach.

1. *Ideally flexible loads*

The case of ideal loads is easily solved analytically. Ideal loads are fully flexible, i.e. they have to be turned on at some point during the day, but it does no matter at all when. Each E_i is allowed to vary as $E_i \in [0, E_{\rm tot}]$ with no further constraint than fixed total consumption, $\sum_i E_i = E_{\rm tot}$. We first consider that the total consumption is equal to the total production – this the real-life situation, where imports and/or exports are included in the production/consumption. The Lagrange multiplier method gives us the weights as

$$c_i = \left(\frac{E_{\rm tot}}{P_i}\right)^{(a-1)} . \tag{1}$$

This choice of weights ensures that minimizing $\sum_i c_i E_i^a$ with $\sum_i E_i = E_{\rm tot}$ gives $E_i = P_i$, i.e. that the consumption profile matches the production profile.

For later use, we next waive the condition of equal total consumption and production, $E_{\rm tot} = \sum_i E_i \neq \sum_i P_i = P_{\rm tot}$. One then obtains, using also the Lagrange multiplier approach [with the additional minimization of the sum of square displacements $\sum_i (P_i - E_i)^2$],

$$c_i = \left(\frac{E_{\rm tot}}{P_i - (P_{\rm tot} - E_{\rm tot})/N}\right)^{(a-1)} , \tag{2}$$

with, as before, the number N of time slices. This corresponds to a consumption curve with constant distance to the production curve, i.e. $P_i - E_i = (P_{\rm tot} - E_{\rm tot})/N$, $\forall i$.

2. Nonideal loads

The above results, obtained in a simple situation suggest that an efficient strategy would be to consider three different classes of loads:
(i) fully flexible loads, i.e. those just defined and discussed,
(ii) partially controllable loads, which have a finite time window $[i_i, i_f]$ in which they act, and
(iii) fully uncontrollable loads.
There are two situations to be differentiated. In the first one, we know nothing about the loads, and only want to generate a set $\{c_i\}$, and hope for the best. In the second one, we know the forecasted consumption curve for each class above, with in particular total consumptions $E_{\rm tot}^{ff}$ for class (i), $E_{\rm tot}^{pf}$ for class (ii) and $E_{\rm tot}^{ul}$ for class (iii) (such that $P_{\rm tot} = E_{\rm tot}^{ff} + E_{\rm tot}^{pf} + E_{\rm tot}^{ul}$). We consider these two situations sequentially.
First, assume that we know nothing about the loads. The set $\{c_i\}$ is generated as in Eq. (1). Class (i) will act as planned - the process is optimized for that class of ideally flexible loads. Class (iii) will not react at all – it wouldn't anyway, regardless of our choice of $\{c_i\}$. Thus we only need to discuss the reaction of class (ii). Each load in class (ii) is restricted to a time window - these windows may or may not be different. Applying the algorithm we developed for fully flexible loads, but restricted to one time window optimizes the set $\{c_i\}$ for the load corresponding to that window. We rewrite Eq. (1) in the form,

$$c_i^{-1/(a-1)} = \frac{P_i}{E_{\rm tot}} \cdot \sum_{j=1}^{N} c_j^{-1/(a-1)} \,, \tag{3}$$

where we stress that the sum over indices j on the right-hand-side runs over all time slices, $j = 1, 2, ...N$. Eq. (1) is recovered under the condition $\sum_i P_i = P_{\rm tot} = E_{\rm tot}$, in which case one sees that $\sum_j c_j^{-1/(a-1)} = 1$. This is no longer the case once the procedure is restricted to a partial time window, say $[i_i, i_f]$. In this case, Eq. (3) has to be modified with $E_{\rm tot} \rightarrow \sum_{j\in[i_i,i_f]} E_i$, and $\sum_{j=1}^{N} c_j^{-1/(a-1)} \rightarrow \sum_{j\in[i_i,i_f]} c_j^{-1/(a-1)}$. Additionally, it is not anymore the case that in that time window, $E_{\rm tot} = P_{\rm tot}$, thus we need to substitute $P_i \rightarrow P_i - (P_{\rm win} - E_{\rm win})/N_{\rm win}$, with $\sum_{i\in[i_i,i_f]} P_i = P_{\rm win}$, $\sum_{i\in[i_i,i_f]} E_i = E_{\rm win}$ and $N_{\rm win} = i_f - i_i$, the number of time slices in that window. We obtain that, for the partially flexible loads in class (ii), the weights should be, instead of Eq. (1)

$$\tilde{c}_i = \left(\frac{E_{\rm tot}}{P_i - (P_{\rm win} - E_{\rm win})/N_{\rm win}} \frac{1}{\sum_{j\in[i_i,i_f]} \tilde{c}_j^{-1/(a-1)}} \right)^{(a-1)} \,, \quad i = i_i, i_i + 1, ...i_f \,. \tag{4}$$

It is easily checked that the so defined $\{\tilde{c}_i\}$ satisfy $\sum_{j\in[i_i,i_f]} \tilde{c}_j^{-1/(a-1)} = 1$, so that this term drops out of Eq. (4). Obviously, the operation $c_i \rightarrow \tilde{c}_i$ can be performed locally and individually by each partially controllable load: provided they central controller sends them not only $\{c_i\}$ but also the $\{E_i^{(0)}\}$, they can reconstruct the set $\{P_i\}$ and calculate $E_{\rm win}$ and $P_{\rm win}$. It seems promising to have the central controller send $\{c_i\}$ as defined in Eq. (1) and the forecasted consumption $\{E_i^{(0)}$ and let the local controller be programmed as fully flexible or partially flexible and treat the weights accordingly.
We are left with discussing the second situation mentioned above, where we can forecast consumption for each class of loads. The strategy is then to start with $\{P_i\}$, subtract the forecasted consumption of uncontrollable loads, $\{E_i^{(ul)}\}$ to obtain $\{\tilde{P}_i\}$, optimize the consumption curve of the partially flexible loads and subtract that curve from $\{\tilde{P}_i\}$ to obtain $\{\overline{P}_i\}$, to finally optimize the fully flexible loads with respect to $\{\overline{P}_i\}$. As a matter of fact, the rule of thumb seems to be to optimize less flexible loads first and more flexible loads last (less/more flexible means "dispatchable inside a shorter/longer time window"). The question is whether such procedures, which require cross-talk between the loads (i.e. the more flexible loads need to know the reaction of the less flexible loads before they plan theirs), is compatible with the philosophy of agreflex. Even if it is, it may be taken into account at a later stage, once the first approach has been implemented and experience with it accumulated.