Control of Prosumer Networks

Nicolas Gensollen, Monique Becker, Vincent Gauthier and Michel Marot
CNRS UMR 5157 SAMOVAR,
Telecom SudParis/Institut Mines Telecom
Email: {nicolas.gensollen, vincent.gauthier, michel.marot, monique.becker}@telecom-sudparis.eu

***Abstract*—The ability to control a given network by dynamically injecting inputs offers ways to make sure that it stays in stable regions. In the case of smart grids, where end users exhibit complex behaviors and renewable production is quite unstable, control of the system is paramount. In this paper, we study the case of a network composed of entities called prosumers. These agents have the ability to both consume and produce according to external conditions. We first model and simplify the underlying dynamic of such network using a second order coupled oscillators network model. Under some conditions, the system synchronize to a common frequency. However, in case of perturbations in the power distribution, the system might lose synchrony and require control to bring it back to the stable state. In this situation, control can be seen as energy absorbed or injected in the system at specific locations. Moreover, the power outputs of the prosumers are susceptible to change such that loads and generators are not fixed. In this context, it is important to select the right subset of nodes that yields, on average, the cheapest control in terms of energy.**

I. Introduction

Modernizing the power grid and increasing the share of renewables in the production are substantial objectives of the energetic transition. Because of progresses in communication, data management, and storage, the upcoming emergence of a power grid "2.0", often called smart grid, appears as a cross-disciplinary challenge of the 21st century [1].

The today centralized top-down architecture is set to evolve to more distributed and bi-directional systems. Furthermore, the emergence of renewable and stochastic distributed energy resources (DER) in the distribution networks will require more flexibility. All these evolutions might change deeply the ways end users perceive and consume electricity. The consumer of the future, while seeking maximum utility/profit tradeoffs, is expected to be more involved in the system operation. Using dynamic signals such as real time electricity prices for instance, load curves could be shaped to some extend as to agree with the production conditions of the grid. Within these complex systems with numerous agents, it is assumed that multiple aggregation levels will be required in order to organize and optimize communication. Agents responsible for generation and load portfolio managment, i.e prosumers [2], will provide services (generation, load shedding, frequency regulation) against remuneration. Insure grid stability within this uncertain context appears as a complex task.

Several studies highlighted the need for increasing the storage capacity of the system. Some of them even consider electric vehicles as moving capacities that could be used for frequency regulation. In the case of fixed storage devices, the locations where they should be installed is an important question since it impacts the performances of the system. In a very static and top-down architecture, where loads and generators are fixed, this question is far from trivial. But in a smart grid scenario with bi-directional flows and agents that produce and consume depending on weather conditions, this becomes even more challenging. In this paper, we explore the use of storage in a network of prosumers whose power outputs are susceptible to change due to various external causes.

More precisely, we model the power grid as a coupled oscillator network, where each node is represented as an oscillator with a phase angle and a natural frequency that are impacted by the ones of their neighbors. Since the work of Kuramoto [], networks of coupled oscillators have been used to model various phenomenons such as fireflies, ex2, or power grids. Indeed, it is well-known that synchronization of all rotating elements to a common frequency is necessary within a power grid for stable operation. Inspired by [3], we use a second order Kuramoto model in order to obtain a simplified dynamics for the power grid (see Section II).

We will see that under some conditions, the system synchronizes to a common frequency Ω. However, this synchronized state is suceptible to break if perturbations occur. If, for example, the production of some generator decreases while the load remains the same, the imbalance between production and consumption might steer the frequency away from Ω and drive the system to unstable conditions. These kinds of perturbations are very likely in a scenario with high penetration of DER. Multiple control techniques, with different time scales, exits in order to cope with such events. Among others, it has been argued that storage and vehicle to grid (V2G) technology could be used for frequency regulation. If a sudden drop or raise of the frequency due to an imbalance is detected, some devices are charged or discharged as to restore synchrony. This approach poses a few questions such as the location of the storage devices, or the quantities of power that should be injected or absorbed. Since charging and discharging a battery is costly, we are particularly interested in the energy necessary to restaure synchrony. Moreover, in a prosumer scenario where generators and loads tend not to be fixed, we would like to find a subset of locations for installing storage such that the energy is low on average.

We will see that these storage actions can be interpreted as an additional term in the oscillators dynamics for a subset of nodes. We propose here to use complex network control theory [4] in order to model the impact of the storage equipments

on the network dynamics (see Section III). Placing storage at some node can thus be seen as selecting this node in the driver set, such that finding the locations for the storage devices could be related to finding the driver set of the network. Nevertheless, selecting the drivers that, on average, will control the network with low energy is a hard problem. Recent studies showed though the existence of relationships between controllability and submodular set functions [5]. These functions, that we introduce in more depth in section X, have the interesting property of having diminishing returns as we increase the size of the set considered. Although finding the optimal set is NP-hard, [6] proposed a simple greedy algorithm with a worst case guaranty.

Since the system under study is a power grid which lines and batteries have finite capacities, we have further constraints on the control signals. Obviously, no line can be overloaded while trying to restaure synchrony. Likewise, no battery can inject power when empty, or charge/discharge quicker than some maximum rate. We propose here to incorporate these constraints in the search of the driver set (see section IV).

This paper is organized as follows, section II presents the oscillator model used for simplifying the grid's dynamics. Section 3 introduces submodular functions and their maximization. Section 4 gives basic notions of control theory and links controllability to submodular set functions. In section IV we develop the constraints relative to the system and the batteries. Finally, in section V, we show some results.

II. The Model

In this section, we introduce the coupled oscillators network model used to simplify the power grid dynamic. The detailed derivation of the second order dynamics can be found in [3], and the key points are reproduced here for the convenience of the reader.

The objective is to achieve synchronization of the grid at the main frequency $\Omega = 50Hz$. Each oscillator i has a phase angle δ_i and a frequency $\dot{\delta}_i$. Therefore, we seek an equilibrium of the form : $\forall i,\ \dot{\delta}_i = \Omega$. For convenience, we express the dynamic of the oscillators in terms of the deviations from the main frequency : $\delta_i(t) = \Omega t + \theta_i(t)$. Let $\omega_i = \dot{\theta}_i$, such that $\dot{\delta}_i = \Omega + \omega_i$. The equilibrium, in terms of the deviations dynamics, is : $\forall i,\ \omega_i = 0$

The next step consists in translating the dynamics of the generators and machines into equations involving the phase angles θ_i and the frequencies ω_i. Generators and machines are composed of a turbine that dissipates energy at a rate proportional to the square of the angular velocity ($P_{diss} = FV \sim V^2$) :

$$P_{diss,i}(t) = K_{Di}(\dot{\delta}_i(t))^2 \tag{1}$$

where K_{Di} is the dissipation constant of entity i.

Furthermore, it also accumulates kinetic energy at a rate :

$$P_{acc,i}(t) = \frac{1}{2} I_i \frac{d}{dt}\left(\dot{\delta}_i(t)^2\right) \tag{2}$$

where I_i is the moment of inertia of entity i. For simplicity, we consider that all entities have the same dissipation constants(K_D) and moment of inertia (I).

The condition for the power transmission between i and j is that the two devices do not operate in phase : The phase difference between i and j is : $\delta_j(t) - \delta_i(t) = \Omega t + \theta_j(t) - \Omega t - \theta_i(t) = \theta_j(t) - \theta_i(t)$. The transmitted power along the line can be written as :

$$P_{transmitted} = -P_{ij}^{MAX} sin(\theta_j - \theta_i) \tag{3}$$

with P_{ij}^{MAX} being the maximum capacity of the line (i, j). If i is connected to more than one other entities, this equation becomes ($\mathcal{N}_i$ being the neighborhood of i) :

$$P_{transmitted} = -\sum_{j \in \mathcal{N}_i} P_{ij}^{MAX} sin(\theta_j - \theta_i) \tag{4}$$

Each entity i is then described by a power balance equation of the type :

$$P_{S,i} = P_{diss,i} + P_{acc,i} + P_{transmitted,i} \tag{5}$$

By substituting and re-arranging the terms, we obtain the following non-linear coupled system of equations :

$$\begin{aligned} &P_{S,i} = I\Omega\ddot{\theta}_i + \left[I\ddot{\theta}_i + 2K_D\Omega\right]\dot{\theta}_i + K_D\Omega^2 \\ &+K_D{\dot{\theta}_i}^2 - \textstyle\sum_{j \in \mathcal{N}_i} P_{ij}^{MAX} sin\left[\theta_j - \theta_i\right] \end{aligned} \tag{6}$$

We now use simplifications based on the fact that we consider small deviations from the main frequency : $\dot{\delta}_i \sim \Omega$ which means that $\omega_i = \dot{\theta}_i << \Omega$, such that the squared term $K_D\dot{\theta}_i^2$ can be neglected. Moreover, we assume that the rate at which energy is stored in the kinetic term is much less of the rate at which energy is dissipated by friction : $\ddot{\theta}_i << \frac{2K_D}{I}$ (see [3] for more details). Equation 6 becomes :

$$\ddot{\theta}_i \sim \psi_i - \alpha\dot{\theta}_i + \sum_{j \neq i} K_{ij} g_{ji} sin\left[\theta_j - \theta_i\right] \tag{7}$$

Where $\alpha = \frac{2K_D}{I}$ is the dissipation term, $K_{ij} = \frac{P_{ij}^{MAX}}{I\Omega}$ are the coupling strengths, $\psi_i = \left[\frac{P_{S,i}}{I\Omega} - \frac{K_D\Omega}{I}\right]$, and g_{ij} is the coefficient of the adjacency matrix G. By working in a rotating frame, we can simplify $\psi_i = \frac{P_{S,i}}{I\Omega}$.

The dynamic is still non linear because of the sine coupling. Therefore, we also assume that the phase angle differences are small such that $sin\left[\theta_j - \theta_i\right] \sim \theta_j - \theta_i$. By using vector notations, the dynamic can be written in the following form :

$$\ddot{\theta} = \Psi - \alpha\dot{\theta} - KL\theta \tag{8}$$

Where L is the Laplacian matrix of the underlying topology (k_i is the degree of node i):

$$L_{ij} = \begin{cases} k_i & if\ i = j \\ -g_{ij} & if\ i \neq j \end{cases} \tag{9}$$

Equation 8 is a continuous time second order linear system of N equations. We first transform this into a continuous time

first order linear system of $2N$ equations by introducing the following vector : $X = \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}$:

$$\dot{X} = \begin{pmatrix} 0 & I \\ -KL & -\alpha I \end{pmatrix} X + \begin{pmatrix} 0 \\ \Psi \end{pmatrix} \tag{10}$$

Which can be written in discrete time:

$$X(t + \Delta t) = MX(t) + \begin{pmatrix} 0 \\ \Psi \Delta t \end{pmatrix} \tag{11}$$

With $M = \begin{pmatrix} I & I\Delta t \\ -KL\Delta t & (1 - \alpha\Delta t)I \end{pmatrix}$.

By setting $Y(t) = \begin{pmatrix} X(t) \\ 1 \end{pmatrix}$, let the dynamic be :

$$Y(t + \Delta t) = AY(t) \tag{12}$$

With transition matrix A :

$$A = \begin{pmatrix} I & I\Delta t & 0 \\ -KL\Delta t & (1 - \alpha\Delta t)I & \Psi\Delta t \\ 0 & 0 & 1 \end{pmatrix} \tag{13}$$

Equation 12 is a linear discrete time system of $2N + 1$ equations. Note that the transition matrix A encodes all the system parameters, topology, and power distribution.

III. Control and Submodularity

A. Submodularity

In this section we provide a very brief introduction to what submodular functions are, and how their maximization could be achieved in reasonable time. More information can be found in [7].

A set function $F : 2^V \longrightarrow \Re$ defined over a finite set V is said to be submodular if it satisfies :

$$\forall S, T \in V,\ F(S) + F(T) \geq F(S \cup T) + F(S \cap T) \tag{14}$$

Another equivalent definition is that for all sets $X, Y \in V$, such that $X \subseteq Y$ and for all element $x \in V \setminus Y$,

$$F(X \cup \{x\}) - F(X) \geq F(Y \cup \{x\}) - F(Y) \tag{15}$$

This basically means that submodular functions exhibit a diminishing return property which makes them particularly interesting for optimization. A very intuitive example is the optimum placement of sensors in an area (see figure **??**). Sensors can be placed on a grid of locations and function F computes the surface of the space that is being sensed. Note first that adding a new sensor i to the current set S cannot decrease the value of F : $F(S + \{i\}) >= F(S)$. Furthermore, if we add sensor i to a small set S_1 we tend to get larger improvements than if we add i to a larger set $S_2 \supset S_1$.

Finding the set S_k of size k that maximizes a set function F is a difficult problem because the number of sets grows exponentially with the number of nodes. Therefore complete enumeration and evaluation is only a feasible solution on very small examples. Nevertheless, if the set function is

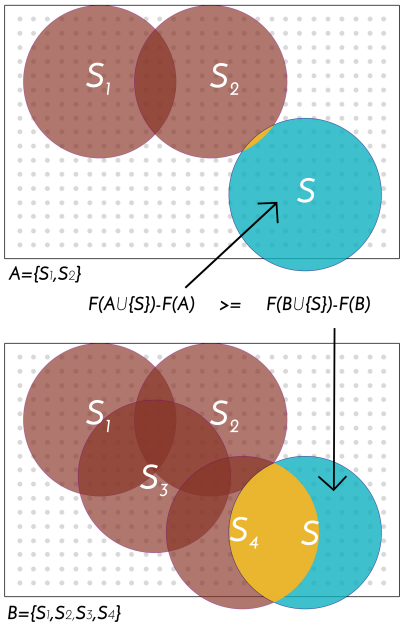

Figure 1: Submodularity example with sensor placements.

submodular, a simple greedy heuristic returns a solution $S_k^\star$ such that, in the worst case, $\frac{F(S_k^\star)}{F(S_k^{OPT})} \sim 63\%$ (where S_k^{OPT} is the optimum set of size k) [7]. This heuristic starts with a set S (possibly empty) and add the element i that exhibits the highest marginal gain : $F(S \cup \{i\}) \geq F(S \cup \{j\})\ \forall j$.

For a ground set of N elements, this heuristic computes F $\frac{k(2N-k+1)}{2}$ times. Since the evaluation of F can be costly, a well-known lazy-greedy variation has been proposed by [6]. This smart implementation uses the submodular structure of the marginal gains in order to reduce the number of calls to F. This requires to maintain a sorted table of marginal gains for all elements. When looking for new element i to add to set S, the top one is selected and the new marginal gain $F(S \cup \{i\}) - F(S)$ is computed. If this gain is larger than the gain of the second element in the table, then i is added to S. Otherwise i is inserted back in the table with its updated gain and the same treatment is applied to the element that is now on the top of the table. Because of the submodularity of F, this method performs as well as the original one, but can result in speedups of several order of magnitudes.

B. Control

In this section, we introduce basic tools from control theory and relate them to the submodular functions introduced above.

We assume the control of a subset of the nodes that we call driver nodes. In our particular power grid situation, a driver node can be seen as a prosumer where some additional storage equipment is used to inject or absorb energy from the system. The dynamics of equation 12 becomes :

$$Y(t + \Delta t) = AY(t) + Bu(t) \tag{16}$$

Where the control matrix B is an indicator of the driver nodes, and $u(t)$ represents the control vector input at time t. The system is said to be controllable in T steps if it can be steered from any initial state Y_0 to any final state Y_f through a sequence of control inputs $u(t)$. If the system is controllable, there might be more than one sequence of control inputs that could achieve such a thing. Among all these possibilities, we are interested in the one that requires the minimum amount of energy [8]. Let $\mathcal{E} = \int_{t_0}^{t_0+T} \|u(t)\|^2 dt$ be the energy used for the control of the system. It can be shown [4] that the control input sequence that minimizes the control energy can be written as :

$$u^\star(t) = B^T (A^T)^{T-t-1} W^{-1}(T) \nu(T) \tag{17}$$

where $\nu(T) = Y_f - A^T Y_0$ is the difference between the desired final state and the final free state, and $W(T) = \sum_{k=0}^{T} A^k B B^T (A^T)^k$ is called the Gramian matrix of the system. The Gramian is a popular tool in control theory since it is deeply linked to the controllability of the system. It can indeed be shown that the system is controllable if and only if the Gramian is not singular, and that its rank indicates the dimension of the controllable subspace. Besides, the minimum control energy associated with the inputs $u^\star(t)$ can be written as :

$$\mathcal{E}_{min} = \nu(T)^T W^{-1}(T) \nu(T) \tag{18}$$

Not surprisingly, the control energy depends on the initial and final states as well as on the inverse of the Gramian. In our prosumer case, we want to find the best battery placement, whose quality should not depend on the initial and final states. Indeed, these states as well as the power distribution Ψ are susceptible to change, and we would like to perform well, on average, whatever the states and distributions. Therefore, we concentrate on the Gramian as a way to quantify the average controllability.

Since the control energy is related to W^{-1}, small energies are obtained from "large" W. In [], the authors introduce several Gramian-based metrics that have concrete meanings in terms of the system controllability. For example, the trace of the inverse Gramian ($Tr[W^{-1}(T)]$) quantifies the average control energy necessary to move the system around the state space. These metrics measures some notions of the control allowed by a given driver node set. But they can be understood the other way around : what drivers should one select as to optimize a given control-based metric ?

Using the Gramian-based metric $\mathcal{M}$, let $F(S) = \mathcal{M}(W_S(T))$, where $W_S(T)$ is the Gramian matrix based on the driver node set S. [] showed that for some metrics, F is a submodular set function. This means that, for these metrics, we can use the greedy heuristic discussed above in order to maximize F.

IV. Control of the Prosumer Network

A. Controllability Constraints

Considering the dynamic of equation (), we notice that the Gramian based on the transition matrix A is never invertible, meaning that the system should not be controllable. It is easy to see that, indeed, we do not have full control because of the last row of the matrix. Nevertheless, as this row was only added in order to include the power distribution inside A, we do not need to control the system along this "fake" dimension. That is, we only seek the control of the system in the first $2N$ dimensions of the space (N phase angles θ_i and N frequencies ω_i). Therefore, whenever $rank[W(T)] = 2N$, we will use the Moore-Penrose pseudo inverse of the gramian $W^\dagger(T)$.

B. Flow Constraints

Recall that the simplified dynamic (see section II) implies that, the power transmitted on line (i, j) is $P_{i \longrightarrow j} = -P_{ij}^{MAX} (\theta_j(t) - \theta_i(t))$. Therefore, we write the flow constraints at time t as :

$$\forall (i,j) \in V^2, \quad g_{ij} \mid \theta_j(t) - \theta_i(t) \mid \leq 1 \tag{19}$$

If this constraint is verified for all instant t in the control time range, then the control inputs satisfy the flow constraints.

C. Power Distribution Constraints

As explained above, we wish to obtain results that do not depend on the power distribution Ψ of the prosumers. However, in order to achieve synchronization along with feasible power flows, we need to define some constraints on this distribution. As shown in [], a condition for achieving synchronisation in a coupled oscillators network is $\|L^\dagger \omega\|_\infty \leq sin(\gamma)$, where L is the Laplacian matrix of the network, ω is the vector of the natural frequencies of the oscillators, and $\|x\|_\infty = max\,|x_i - x_j|$. If this condition is satisfied, the oscillators will synchronize at the common frequency ω_{SYNC} with the phase lock $g_{ij}\,|\theta_i - \theta_j| \leq \gamma \in [0, \frac{\pi}{2}],\ \forall (i,j) \in V^2$.

Therefore, we can write the following synchronization constraint :

$$\| \left(L \circ P^{MAX}\right)^\dagger \Psi \|_\infty \leq sin(1) \tag{20}$$

Where $A \circ B$ represents the Hadamard product between matrices A and B. If constraint 20 is satisfied, the oscillators will synchronize to a common frequency $\omega_{SYNC} = \frac{\sum_{k=1}^{N} \Psi_k}{\sum_{k=1}^{N} \alpha_k} = \frac{\sum_{k=1}^{N} P_{S,k}}{I \Omega N \alpha}$, and the flows at steady state will be feasible. Note that ω_{SYNC} should be zero in order to synchronize to the main frequency Ω (recall that the phase angles and frequencies represent deviations from Ω). This gives the following intuitive zero sum constraint :

$$\sum_{k=1}^{N} P_{S,k} = 0 \tag{21}$$

Constraint 21 simply states that production should match demand in order for the system to remain stable.

D. Battery Constraints

Let i be a battery. In the simple model we adopt here, it is characterised by its maximum charge/discharge rate r_i, the amount of energy stored at time t $\Lambda_i(t)$, and its maximum capacity $\Lambda_{i,MAX}$. For simplicity, we will consider all maximum charge/discharge rates to be the same (r), and all maximum capacities to be the same Λ_{Max}. Furthermore, we denote by $\Lambda(t)$ the vector of energy level at time t.

Since the control input vector $u^\star(t)$ specifies the control inputs, the energy level dynamic of the batteries can be written as :

$$\Lambda(t+\Delta t) = \Lambda(t) - u^\star(t) I\Omega \tag{22}$$

Which can also be written as :

$$\Lambda(t) = \Lambda(0) - I\Omega \sum_{k=0}^{t} u^\star(k) \tag{23}$$

Where $\Lambda(0)$ is the vector of initial levels in the batteries.

Obviously, the energy level in a battery is bounded. This leads to the following set of constraints :

$$\forall t, \quad 0 \leq \Lambda(t) \leq \Lambda_{MAX} \tag{24}$$

Which can be written in terms of the control inputs :

$$\forall t, -\frac{\Lambda(0)}{I\Omega} \leq -\sum_{k=0}^{t} u^\star(k) \leq \frac{\Lambda_{MAX} - \Lambda(0)}{I\Omega} \tag{25}$$

Finally, during each time slot, the battery cannot charge or discharge at a rate higher than r :

$$\forall t, \quad \mid u^\star(t) I\Omega \mid \leq r \tag{26}$$

If, for a given $u^\star$, the constraints of eq. 25 and 26 are satisfied, then the control inputs satisfy the storage constraints.

E. Algorithm

We propose the following greedy algorithm in order to find the smallest set of locations that minimizes the average control energy and satisfy the constraints. We assume here that the system is in a given initial state Y_0 and we want to reach a desired final state Y_f in T time steps. The algorithm **??** starts with an empty set and, as long as that constraints are not satisfied, it increases the size of the set. For a given size k, the set S_k is found using lazy greedy submodular optimization of a Gramian-based set function F. In practice we use $F(S) = -Tr\left[W_S^\dagger(T)\right]$ which quantifies the average energy required to move the system around the controllable subspace. The algorithm **??** returns then the tuple $(k, S_k, u^\star)$ such that S_k is the smallest set that minimizes the average control energy and, in this case, satisfies the constraints with control $u^\star$.

while $Not \begin{cases} rank\left[W_{S_k}(T)\right] < 2N \ and \\ \forall t, \forall (i,j) \in V^2, g_{ij} \left|\theta_j(t) - \theta_i(t)\right| \leq 1 \ and \\ \forall t, -\frac{\Lambda(0)}{I\Omega} \leq -\sum_{k=0}^{t} u^\star(k) \leq \frac{\Lambda_{MAX}-\Lambda(0)}{I\Omega} and \\ \forall t, \ |u^\star(t)| \leq \frac{r}{I\Omega} \end{cases}$

do

$k \leftarrow k+1$

$S_k \leftarrow Subopt(F,k)$

$u^\star(t) \leftarrow OptControl(S_k)$

end while

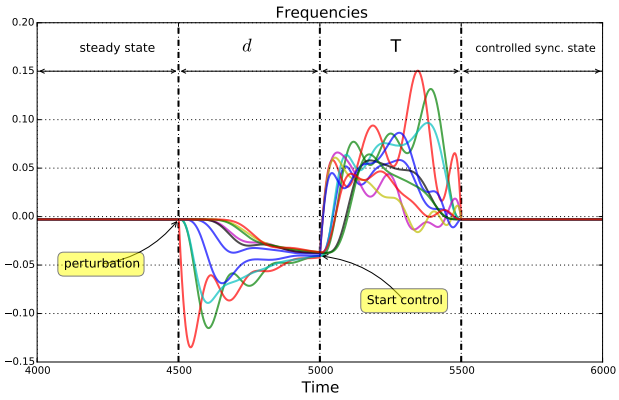

Figure 2: Time series of the N frequencies ω_i (phase angles are not shown for clarity). d is the delay between the perturbation and the begining of the control phase that bring the system bak in the synchronized state in T time steps. Here, d was chosen quite large for the clarity of the figure. Note that, if the perturbation is still active at this point, the system still needs control in order to stay synchronized (batteries should still compensate the power imbalance)

V. Results

For a direct illustrative application, we chose a small system of 10 nodes connected with an erdos-renyi topology. The power distribution and the capacities of the lines are selected randomly such that the system is at equilibrium. Therefore, the production matches the consumption, no line is overloaded and all nodes are synchronized to the main frequency. We use the greedy algorithm (X) based on the trace of the inverse Gramian as to select the driver set S. At a given time t_k, we introduce a small perturbation in the form of a mismatch between the production and consumption : $P_i(t_k) = P_i(t_{k-1}) + \delta P$ for some random node i. Consequently, the frequencies $\dot{\theta}_i \forall i$ of the oscillators start to deviate from the synchronized state. At time $t_k + d$, we apply to the drivers the optimal control of equation (X) such that the system is brought to the synchronized state at time $t_k + d + T$, where T is the control time. At this time, if the mismatch between production and consumption is still active, we maintain synchrony by using optimal control with $T = 1$. Figure 1 shows the evolution of a subset (for clarity) of the nodes' frequencies during the control phase.

The purpose of selecting the drivers according to a Gramian related metric is to minimize, on average, the amount of

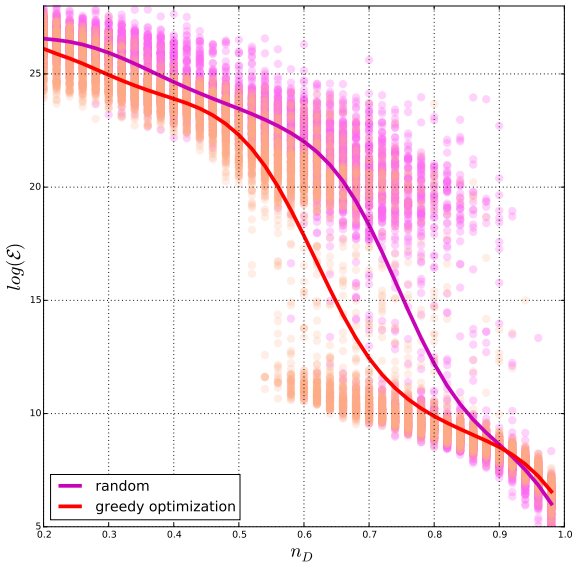

Figure 3: Control energy $log(\mathcal{E})$ against size of control set n_D for random and optimized driver sets. Topologies are drawn from a Barabasi-Albert model with $N_{nodes} = 50$, power and line capacity distributions are random. For both case, 10^4 instances are considered and the results are displayed as dots. The lines shows the average for both case.

control energy needed. In other words, if we select the drivers randomly, we expect to need, on average, more energy to control the system. In figure 2, we draw 10^4 example systems of 50 nodes with random scale-free topologies (Barabasi-Albert model is used), and random power distributions and line capacities. For each system we select a random number of drivers $N_D \sim \mathcal{U}(1, N_{nodes})$ and we find two driver sets of size N_D. One is chosen randomly and the other is found by greedy optimization of a gramian based metric. For both sets, we draw a random initial state Y_i and a random final state Y_f, and we compute the control energy required for driving the system from Y_i to Y_f. On figure **??** we plot each system as a dot with coordinates $(n_D = N_D/N_{nodes}, log(\mathcal{E}))$. The color indicates what method has been used, and the lines show the average accross n_D. As visible, both curve decreases as the number of drivers increases, meaning that, as the number of drivers grows, the average control energy tends to decrease. Furthermore, we tend to use, as exected, less energy when the drivers are carefully selected. Note that this difference tends to zero when n_D tend to one, because almost all nodes are then selected, yielding little flexibility for optimization.

We investigate next how the topology of the grid affects the size of the control set. We consider first the case of an erdos renyi topology with probability of connection p. We show on Figure 3 how the size of the driver set evolves with p for different Gramian based metrics and for two levels of

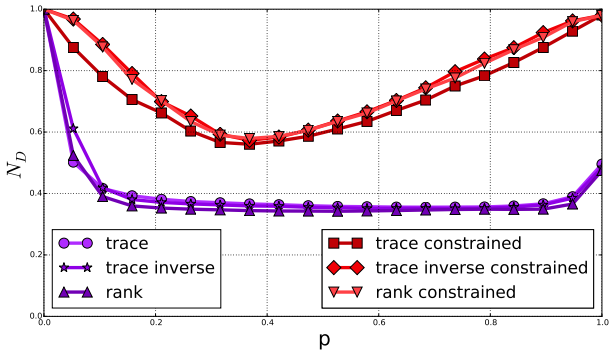

Figure 4: n_D against link probability p for erdos-renyi topologies ($N = 100$). Curves are averaged over 100 realizations. The top three curves show the results for the three metrics taken into consideration when all constraints are considered (see Level 2 in the main text). The bottom three curves exhibit the results for the same metrics when only the full controllability constraint is considered (see Level 1 in the main text).

constraints :

- Level 1 : full control : the system can be moved from any point to any other point of the state space. We do not consider constraints on the energy which can tend to infinity.
- Level 2 : full control + line capacities constraints + battery constraints (see algorithm X). This level is much more demanding since we impose the ability to move the system from any point to any other point, without overloading any line or breaking any battery limits in the process.

When p is close to zero, nodes tend to be very poorly connected such that we need to control almost all nodes in the grid. As p increases, the connectivity of the graph augments and the number of drivers decreases. At some point, the connectivity of the graph starts to harm its controllability, and more drivers are needed. This effect is in accordance with the litterature although it is less marked here because the topology has less influence in our model (recall that matrix A in the dynamics is not the adjacency matrix of the graph but a block matrix where one of the block is the Laplacian matrix of the graph). As expected, the driver sets for the level 2 of constraints are larger than for level 1 (for all metrics) because we impose far more constraints.

Because transmission power grids are spatially and politically constrained (each contry has its own transmission grid which is interconnected with neighboring grids at some specific locations), we study the control in clustered graphs. More precisely, we use a block model $(N, N_{clusters}, p_{in}, p_{out})$ to generate random topologies where N is the number of nodes, $N_{clusters}$ is the number of clusters, p_{in} is the probability that two nodes within the same cluster are linked, and p_{out} is the probability that two nodes in two distinct clusters are

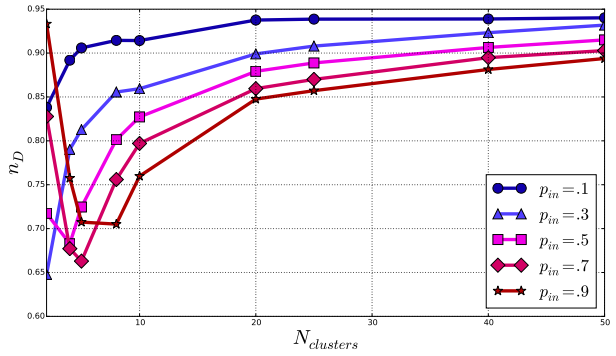

Figure 5: n_D against $N_{clusters}$ for different values of p_{in} and $p_{out} = 0.1$. $N = 200$ and curves are averaged over 100 realizations.

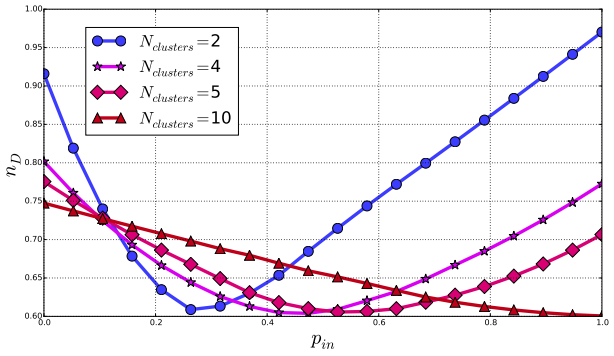

Figure 6: n_D against p_{in} for systems of $N = 200$, $p_{out} = 0.1$, and for different values of $N_{clusters}$. The curves are averaged over 100 realizations.

connected.

Figure 4 displays how the number of drivers evolves with the number of clusters for different values of p_{in} when p_{out} is fixed ($p_{out} = 0.1$). For small values of $p_{in} \sim p_{out}$ clusters are poorly marked, and the connectivity is low. The number of drivers in these conditions is large and increases slowly when the number of clusters augments. As p_{in} increases, the clusters becomes more densily connected such that within a cluster less drivers are required in order to control it. As the number of clusters grows, more drivers are required. For large values of p_{in}, the behavious is more complex. We see indeed that for $p_{in} = 0.9$ and $N_{clusters} = 2$, almost 94% of the nodes are needed for control, but this quantity first decreases with the addition of a few more clusters (71% for $N_{clusters} = 5$), before increasing. This behavior can be explained by the fact that controlling a very dense network requires, in our settings, a very large portion of nodes to be drivers. Controlling one cluster alone thus requires to control almost all nodes within this cluster. Nevertheless, when a relatively small (compared to the number of nodes in the graph) number of clusters are interconnected with a few links, nodes in one cluster are able to control nodes in other clusters to which they are connected. As the number of clusters grows, the global connectivity increases rapidely, such that the number of drivers also rises. We show this behavior in more details in figure 5, where the number of drivers is plotted against p_{in} for $N = 200$ and $N_{clusters} \in \{2, 4, 5\}$. For $N_{clusters} = 2$ and $p_{in} = 0.1$ the number of drivers is large since the graph is poorly connected. When p_{in} increases, n_D decreases until a minimum value $n_D \sim 0.64$ at $p_{in} \sim 0.6$. After this point, n_D augments with p_{in}. For $N_{clusters} = 5$, we do not see this behavior : n_D decreases continuously as p_{in} increases as expected from the curves of figure 4.

Until now, we have considered all batteries to be equal, i.e they have the same capacity and charge/discharge rate. Although simple, this assumption might not be verified in practice where different types of batteries exist. Optimizing the locations of different types of storage as to optimize

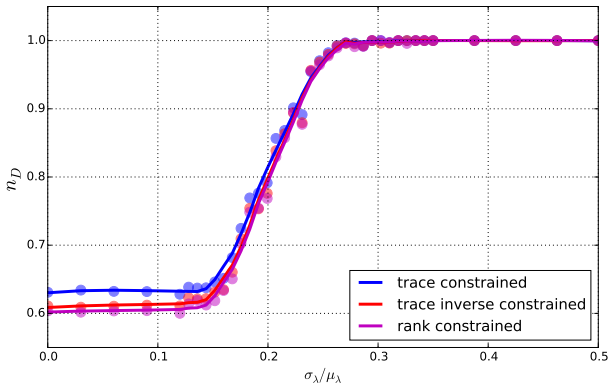

Figure 7: n_D against the relative standard deviation of the capacity distribution of the batteries $\frac{\sigma_\lambda}{\mu_\lambda}$. The dots are averages over 100 realizations and the curves are obtained using a Savitzky-Golay filter, each color shows the results for a given metric. The graphs are erdos-renyi with $N = 100$ and $p = 0.3$, and $\mu_\lambda = 100$ units.

the cost / reliability tradeoff is out of the scope of this paper. Nevertheless, we studied how the number of drivers behaves when the batteries have different capacities. More precisely, we draw the capacities of the batteries from a normal distribution $\mathcal{N}(\mu_\lambda, \sigma_\lambda)$ and keep a simple erdos-renyi topology for the power grid. Note that each node is assigned a capacity regardless of its caracteristics (degree, betweeness, and so on...). In figure 6, we plot n_D as a function of the relative standard deviation of the battery capacity distribution $\sigma_\lambda/\mu_\lambda$. We see that n_D increases with the variance of the capacity distribution for all three gramian based metrics considered. Moreover, this rise is abrupt since n_D goes from 60% for $\frac{\sigma_\lambda}{\mu_\lambda} \sim 0.15$ to 100% for $\frac{\sigma_\lambda}{\mu_\lambda} \sim 0.27$.

Real power grids topologies are known to be far from random such that using the erdos-renyi model probably do not give a realistic insight into the control of these systems. Depending on the voltage level

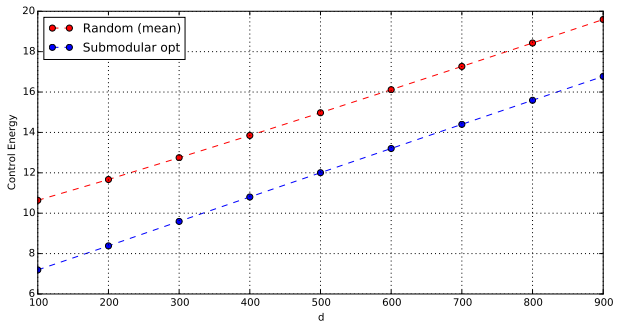

Figure 8: Control energy used in order to bring the system back to the synchronized state when the delay between perturbation and control increases, while the driver node set is fixed. The blue dotted curve shows the results for $S_k^\star$ our algorithm **??**. The red line shows (average and standard deviation) the results when the driver set is sampled uniformly from all possible sets of the size k.

In this section, we provide some results in order to illustrate the work described above. We use a small 10 prosumers example, connected with a random topology. The line maximum capacities are selected according to a normal distribution $\mathcal{N}(\mu_l, \sigma_l)$. The power distribution is chosen as a zero mean normal distribution $\mathcal{N}(0, \sigma_p)$, and the samples should satisfy the constraints 20 and 21. Obviously, μ_l, σ_l, and σ_p should be chosen such that, without any perturbation, the lines are not already overloaded.

Once the synchronized state is reached, we apply a perturbation to the power distribution such that there is a mismatch between production and demand (constraint 21 is not satisfied). We denote by d the delay between the perturbation and the start of the control phase (see figure **??**). d depends on how fast the perturbation can be identified and located, the control inputs calculated and communicated without error through the communication network.

If the driver set is fixed, we expect that the higher d, the higher the control energy required to drive the system back to the synchronized state. Eventually, if d is too large, the minimum energy control inputs will result in some constraints being broken (battery levels, line flows, and so on...).

As visible on figure **??**, when d increases, the system deviates more from the equilibrium and this results in more energy being used to steer it back to synchrony. We also investigate here how the driver node set selected with algorithm **??** performs against other driver sets of the same size. we use a Monte-Carlo sampling of the drivers sets of the same size as our solution, and filter out the ones that do not meet all the constraints. Figure **??** shows that our solution (blue dotted curve) uses less energy than other driver sets of the same size (red curve).

VI. Conclusion

C'est fini !

References

[1] S. D. Ramchurn *et al.*, "Putting the 'smarts' into the smart grid: A grand challenge for artificial intelligence," *Commun. ACM*, vol. 55, no. 4, pp. 86–97, Apr. 2012.

[2] a. J. D. Rathnayaka, V. Potdar, and M. H. Ou, "Prosumer management in socio-technical smart grid," *Proceedings of the CUBE International Information Technology Conference on - CUBE '12*, p. 483, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2381716.2381808

[3] G. Filatrella, a. H. Nielsen, and N. F. Pedersen, "Analysis of a power grid using a Kuramoto-like model," *The European Physical Journal B*, vol. 61, no. 4, pp. 485–491, mar 2008. [Online]. Available: http://www.springerlink.com/index/10.1140/epjb/e2008-00098-8

[4] Y.-Y. Liu and A.-L. Barabási, "Control Principles of Complex Networks," pp. 1–55, 2015. [Online]. Available: http://arxiv.org/abs/1508.05384

[5] T. Summers, F. Cortesi, and J. Lygeros, "On Submodularity and Controllability in Complex Dynamical Networks," *arXiv preprint arXiv:1404.7665*, pp. 1–10, 2014. [Online]. Available: http://arxiv.org/abs/1404.7665

[6] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*, ser. Lecture Notes in Control and Information Sciences, J. Stoer, Ed. Springer Berlin Heidelberg, 1978, vol. 7, pp. 234–243. [Online]. Available: http://dx.doi.org/10.1007/BFb0006528

[7] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, pp. 71–104, 2014.

[8] G. Yan, J. Ren, Y. C. Lai, C. H. Lai, and B. Li, "Controlling complex networks: How much energy is needed?" *Physical Review Letters*, vol. 108, no. 21, pp. 1–5, 2012.