La Programmation Dynamique (Stochastique)
en Génie Électrique

Pierre Haessig

EDF R&D LME, ENS Cachan SATIE
contact : pierre.haessig@ens-cachan.fr

Journées SEEDS/JCGE'13
Workshop Modélisation & Optimisation, St-Nazaire 6 juin 2013

Plan de la présentation

1. Qu'est-ce que la (S)DP ?

2. Application à un système éolien-stockage
   - Contexte éolien-stockage
   - Mise en oeuvre de la SDP
   - Premiers résultats : convergence
   - Effet du choix de la fonction coût

3. Conclusion
   - Avantages-inconvénients de la méthode
   - Références

Plan de la présentation

1. Qu'est-ce que la (S)DP ?
2. Application à un système éolien-stockage
3. Conclusion

Qu'est-ce que la (S)DP ?

La Programmation Dynamique Stochastique (SDP) est

Definition

une méthode pour résoudre des problème d'*optimisation dynamique* (Bellman, $\sim$ 1950)

Qu'est-ce que la (S)DP ?

La Programmation Dynamique Stochastique (SDP) est

Definition

une méthode pour résoudre des problème d'*optimisation dynamique* (Bellman, $\sim$ 1950)

l'optimisation dynamique (stochastique) est aussi appelée *contrôle optimal (stochastique)*.
La Programmation Dynamique est une méthode non-variationnelle (à la différence du principe de maximum de Pontryagin).

Exemples de problèmes

La Programmation Dynamique Stochastique (SDP) peut servir à gérer optimalement des *stocks* (entre autres).

Exemples de stocks à gérer :

- **barrage** hydro-électrique (incertitude sur les apports d'eau)
- **batterie** couplée à des énergies renouvelables (incertitudes de production, de consommation)
- batterie de voiture, en particulier les cas hybrides (incertitude sur le profil de mission)
- ressources naturelles : minerai, poissons, ...

Le principe de la méthode

Les problèmes d'optimisation dynamique sont caractérisés par la présence d'un *couplage entre les instants* :

la dynamique/mémoire/inertie du système

$$\text{“}x_{k+1} = f(x_k, u_k, w_k)\text{”}$$

Ce couplage temporel implique que les choix des variables de contrôle u_k ne peut *pas être fait indépendamment*.

Le principe de la méthode

Les problèmes d'optimisation dynamique sont caractérisés par la présence d'un *couplage entre les instants* :

la dynamique/mémoire/inertie du système

$$\text{“}x_{k+1} = f(x_k, u_k, w_k)\text{”}$$

Ce couplage temporel implique que les choix des variables de contrôle u_k ne peut *pas être fait indépendamment*.

La Programmation Dynamique casse ce couplage temporel en faveur d'une résolution *itérative, un instant après l'autre*.

Pour cela, la méthode introduit la notion de *cost-to-go* (valeur/fonction de Bellman), qui mesure *le coût du futur*.

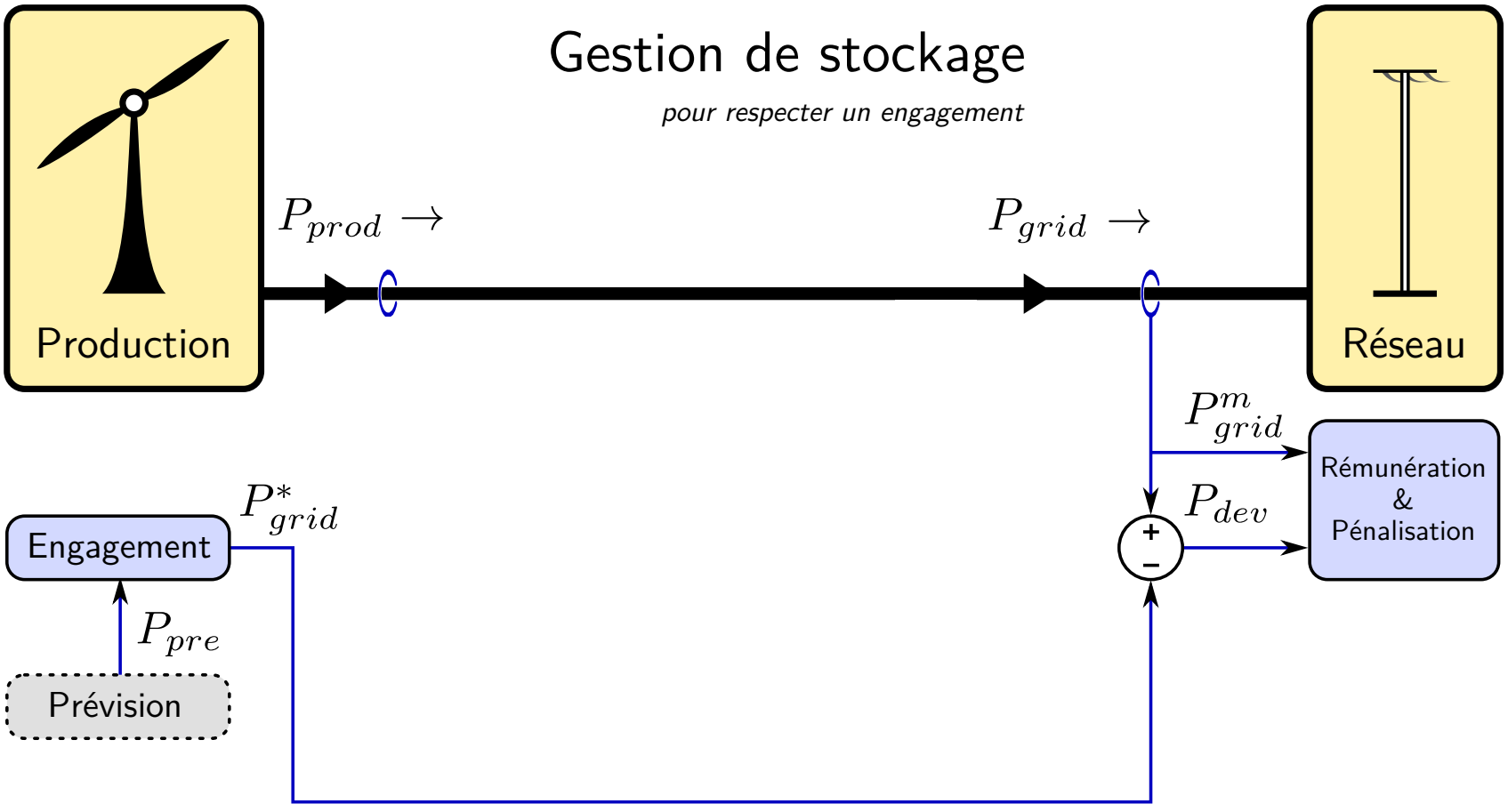
Plan de la présentation

1 Qu'est-ce que la (S)DP ?

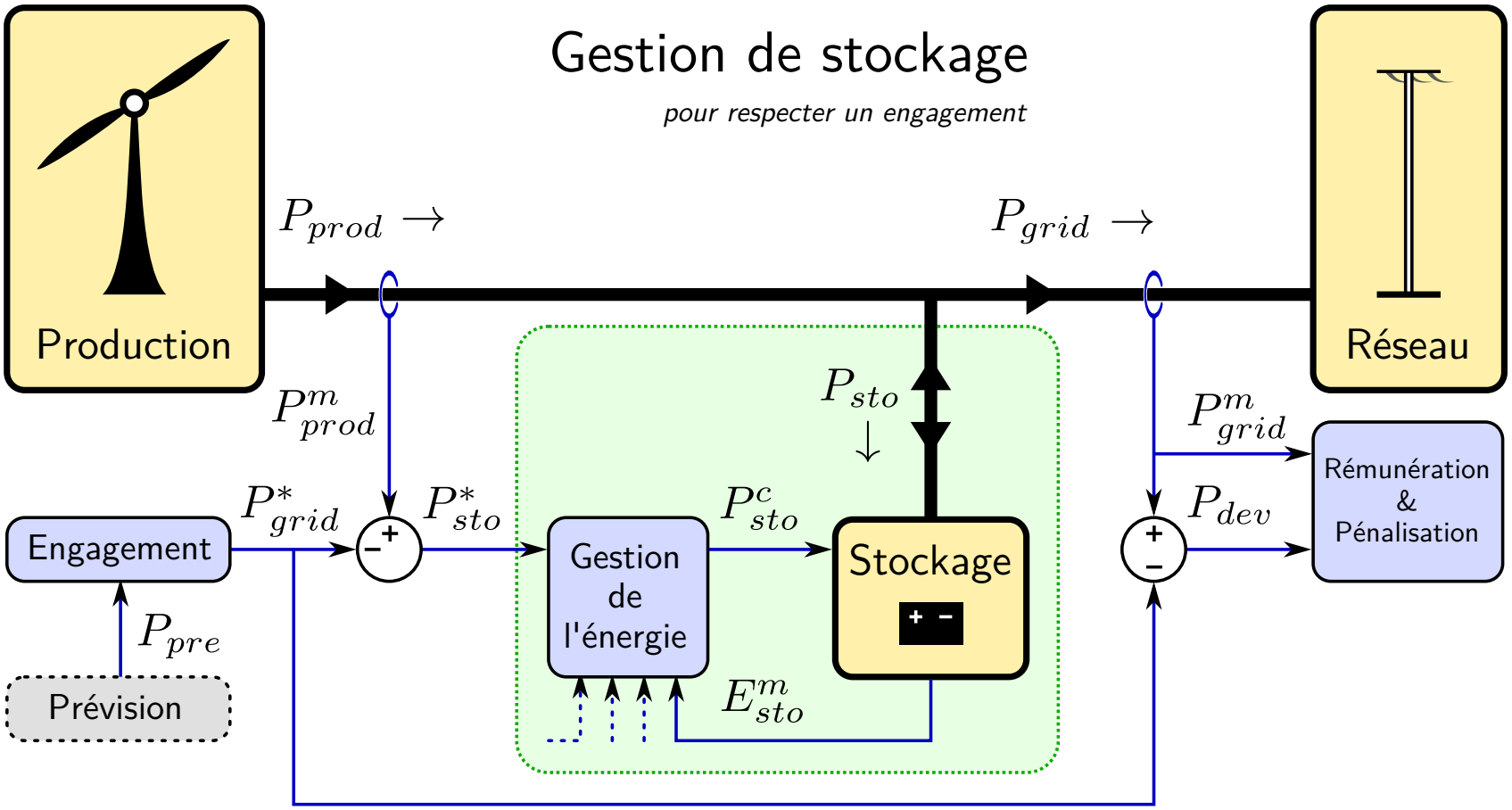
2 Application à un système éolien-stockage
- Contexte éolien-stockage
- Mise en oeuvre de la SDP
- Premiers résultats : convergence
- Effet du choix de la fonction coût

3 Conclusion

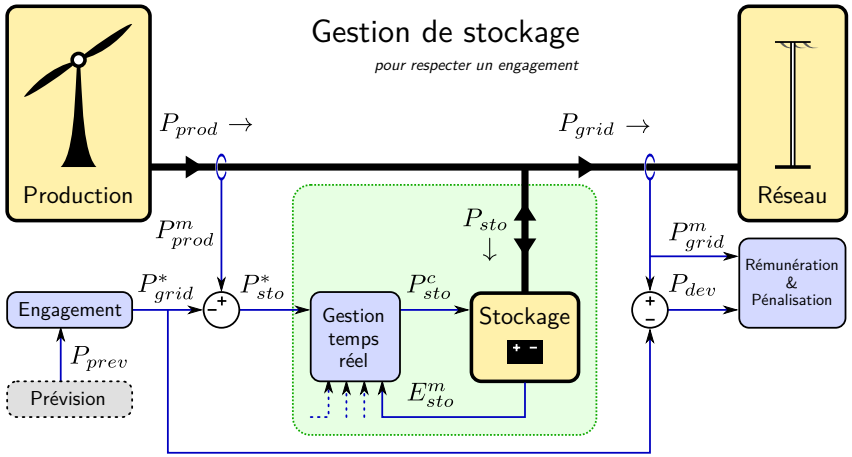
Système de production éolien avec stockage



Système de production éolien avec stockage



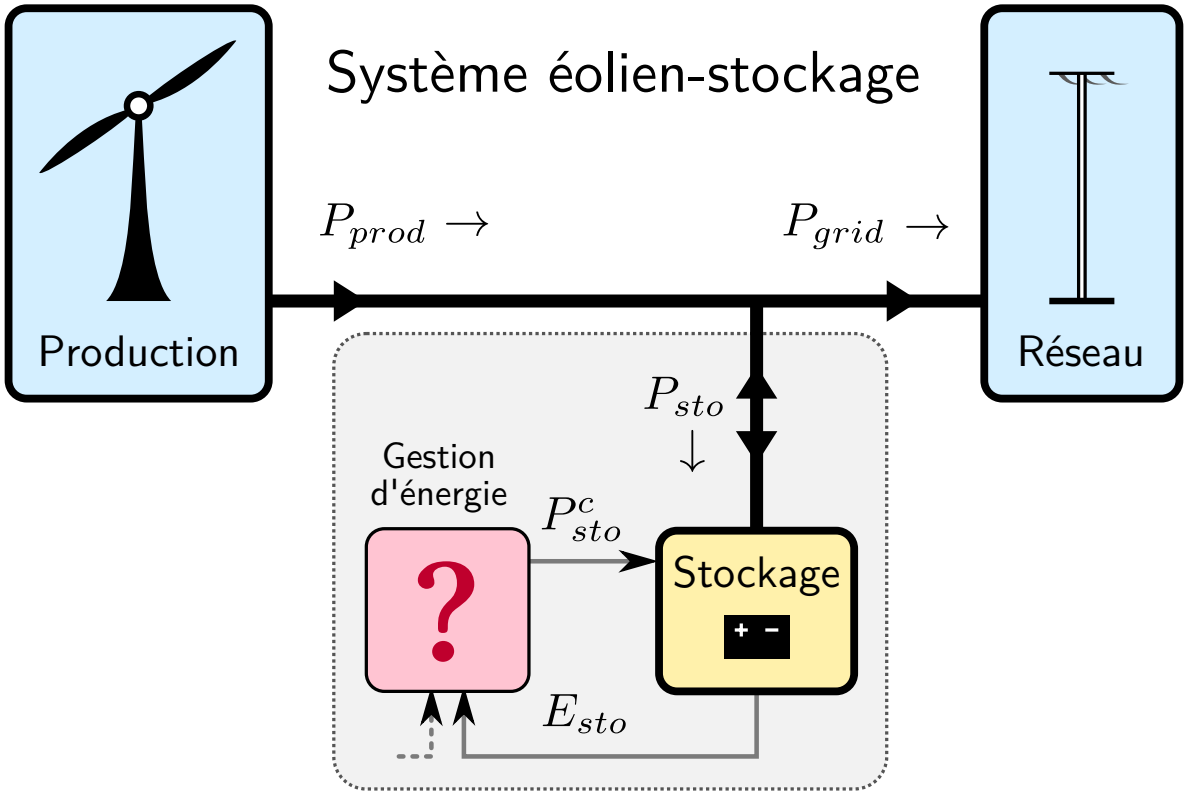
Contexte industriel éolien-stockage



Appel d'offre de la Commission de Régulation de l'Énergie (CRE) pour des systèmes éoliens "avec services" :

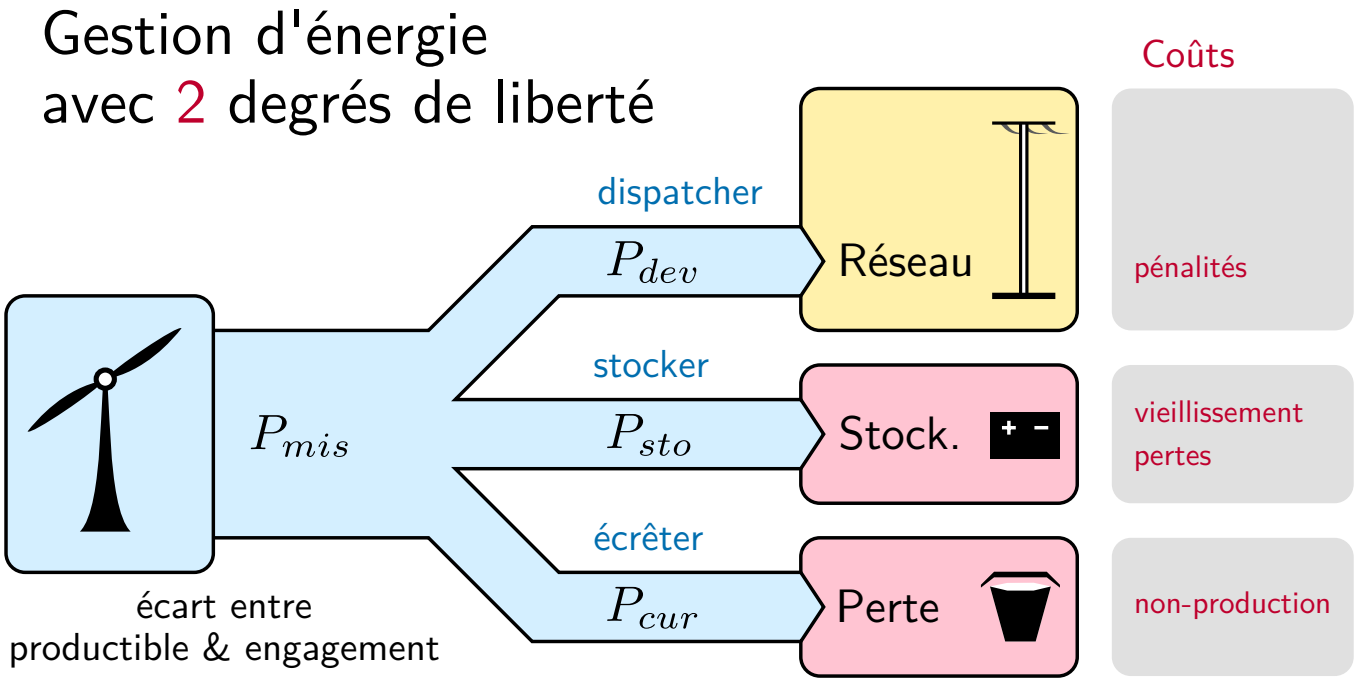
- réserve primaire (10 % de la puissance nominale libérable pendant 15 minutes)
- limitation des variations de la puissance
- **engagement** sur un plan de production 1 jour à l'avance.

Description du problème de contrôle



Comment gérer le stockage d'énergie ?

Description du problème de contrôle



On cherche à répartir l'écart P_{mis} ("mismatch") entre : le réseau, un stockage et une consigne d'écrêtage, au coût le plus bas.

Notations de la Programmation Dynamique

Objectif : **minimiser un coût additif** :

$$J = \sum_{k=0}^{K-1} g(x_k, u_k, w_k)$$

Notations des variables :

- état du système $x = (E, P_{mis})$
- commande $u = (P_{sto}, P_{cur})$
- perturbation w

Notation des fonctions :

- dynamique du système $x_{k+1} = f(x_k, u_k, w_k)$
- coût à chaque instant $g(x_k, u_k, w_k)$

Notations de la Programmation Dynamique

Objectif : **minimiser un coût additif** (en espérance) :

$$J = \mathbb{E}\Big\{ \sum_{k=0}^{K-1} g(x_k, u_k, w_k) \Big\}$$

Notations des variables :

- état du système $x = (E, P_{mis})$
- commande $u = (P_{sto}, P_{cur})$
- perturbation w

Notation des fonctions :

- dynamique du système $x_{k+1} = f(x_k, u_k, w_k)$
- coût à chaque instant $g(x_k, u_k, w_k)$

Dynamique du système

Dynamique du système $f(x_k, u_k, w_k)$ pour l'éolien-stockage :

$$E(k+1) = E(k) + P_{sto}(k) - a|P_{sto}(k)|$$
$$P_{mis}(k+1) = \phi P_{mis}(k) + w(k)$$

Ces équations modélisent le stockage (avec un facteur de pertes a), ainsi que l'écart P_{mis} avec un processus AR(1).

On a des *contraintes* sur la commande : $u \in U(x)$

- limites d'énergie du stockage $0 \leq E + P_{sto} - a|P_{sto}| \leq E_{rated}$
- limite de puissance du stockage : $|P_{sto}| \leq P_{rated}$
- positivité de l'écrêtage : $P_{cur} \geq 0$

Coût de chaque instant

Coût de chaque instant $g(x, u)$:

$$g(E, P_{mis}, P_{sto}, P_{cur}, w) =$$
$$\underbrace{c_{cycl}|P_{sto}|}_{\text{cyclage}} + \underbrace{c_{elec}(P_{cur} + a|P_{sto}|)}_{\text{pertes de production}} + \underbrace{c_{dev}|P_{mis} - P_{cur} - P_{sto}|}_{\text{écart à l'engagement}}$$

(ou bien n'importe quelle autre formule de pénalité !)

L'intérêt de la méthode choisie est de ne pas supposer de forme particulière de la fonction coût (ni linéarité, ni convexité, ...)

Programmation Dynamique

la méthode, en accéléré

Calcul récursif, partant du dernier instant :

$$J_K(x_K)^* = g(x_K) \quad \text{(coût terminal)}$$

puis on remonte le temps, pour $k = K-1, \ldots, 0$:

$$J_k(x_k) \;=\; \mathop{\mathbb{E}}_{w_k}\Big\{ \underbrace{g(x_k, u_k)}_{\text{coût de l'instant}} + \underbrace{J^*_{k+1}(f(x_k, u_k, w_k))}_{\text{coût du futur}} \Big\}$$

Programmation Dynamique
la méthode, en accéléré

Calcul récursif, partant du dernier instant :

$$J_K(x_K)^* = g(x_K) \quad \text{(coût terminal)}$$

puis on remonte le temps, pour $k = K-1, \ldots, 0$:

$$J_k(x_k)^* = \min_{u_k \in U(x_k)} \mathbb{E}_{w_k} \Big\{ \underbrace{g(x_k, u_k)}_{\text{coût de l'instant}} + \underbrace{J^*_{k+1}(f(x_k, u_k, w_k))}_{\text{coût du futur}} \Big\}$$

c'est l'équation de la Programmation Dynamique, ou éq. de Bellman.

Programmation Dynamique

l'optimisation proprement dite

L'équation de programmation dynamique inclut une minimisation :

$$J_k(x_k)^* = \min_{u_k \in U(x_k)} \mathbb{E}_{w_k} \Big\{ \underbrace{g(x_k, u_k)}_{\text{coût de l'instant}} + \underbrace{J^*_{k+1}(f(x_k, u_k, w_k))}_{\text{coût du futur}} \Big\}$$

Il s'agit de trouver, pour *chaque état* x_k, la commande u_k qui minimise $J_k(x_k, u_k)$.
Généralement, le nombre de variables de commande est faible (1 ou 2) et on peut optimiser par *énumération* des valeurs possibles.
Exemple pour la batterie : $P_{sto} \in \texttt{linspace}(-\mathrm{P}_{\texttt{max}}, \mathrm{P}_{\texttt{max}}, 100)$

Programmation Dynamique

l'optimisation proprement dite

L'équation de programmation dynamique inclut une minimisation :

$$J_k(x_k)^* = \min_{u_k \in U(x_k)} \mathop{\mathbb{E}}_{w_k} \Big\{ \underbrace{g(x_k, u_k)}_{\text{coût de l'instant}} + \underbrace{J^*_{k+1}(f(x_k, u_k, w_k))}_{\text{coût du futur}} \Big\}$$

Il s'agit de trouver, pour *chaque état* x_k, la commande u_k qui minimise $J_k(x_k, u_k)$.
Généralement, le nombre de variables de commande est faible (1 ou 2) et on peut optimiser par *énumération* des valeurs possibles.
Exemple pour la batterie : $P_{sto} \in \texttt{linspace}(-\texttt{P}_{\texttt{max}}, \texttt{P}_{\texttt{max}}, 100)$

L'étape de minimisation aboutit à un *loi de gestion* optimale :

$$u_k = \mu^*_k(x_k)$$

Programmation Dynamique

la méthode pour un coût moyen

On peut modifier la méthode pour minimiser non pas une somme mais une *moyenne temporelle* :

$$J = \frac{1}{K}\,\mathbb{E}\Big\{\sum_{k=0}^{K-1} g(x_k, u_k, w_k)\Big\} \quad \text{avec } K \to \infty$$

La résolution se fait récursivement, comme précédemment. On observe une convergence au bout d'un *certain nombre d'itérations*. (qq dizaines avec notre problème).

Au final, on obtient une loi de gestion optimale *stationnaire* :

$$u_k = \mu^*(x_k) \quad (\mu \text{ ne dépend pas de l'instant } k)$$

Contrôle boucle fermé

Propriété importante de la Programmation Dynamique :

- elle ne donne pas une valeur de commande optimale (i.e. un nombre u_k^*)
- mais une loi de gestion optimale (i.e. une fonction de l'état $\mu_k^* : x_k \mapsto u_k^*$)

loi de gestion = policy en anglais

Contrôle boucle fermé

Propriété importante de la Programmation Dynamique :

- elle ne donne pas une valeur de commande optimale (i.e. un nombre u_k^*)
- mais une loi de gestion optimale (i.e. une fonction de l'état $\mu_k^* : x_k \mapsto u_k^*$)

loi de gestion = policy en anglais

Le fait que le "produit de l'optimisation" soit une fonction et non pas un nombre a ses avantages et ses inconvénients.
Notons que dans le cas stochastique, travailler avec une loi de gestion (un contrôle "boucle fermé") est *inévitable*.

Convergence de l'algorithme

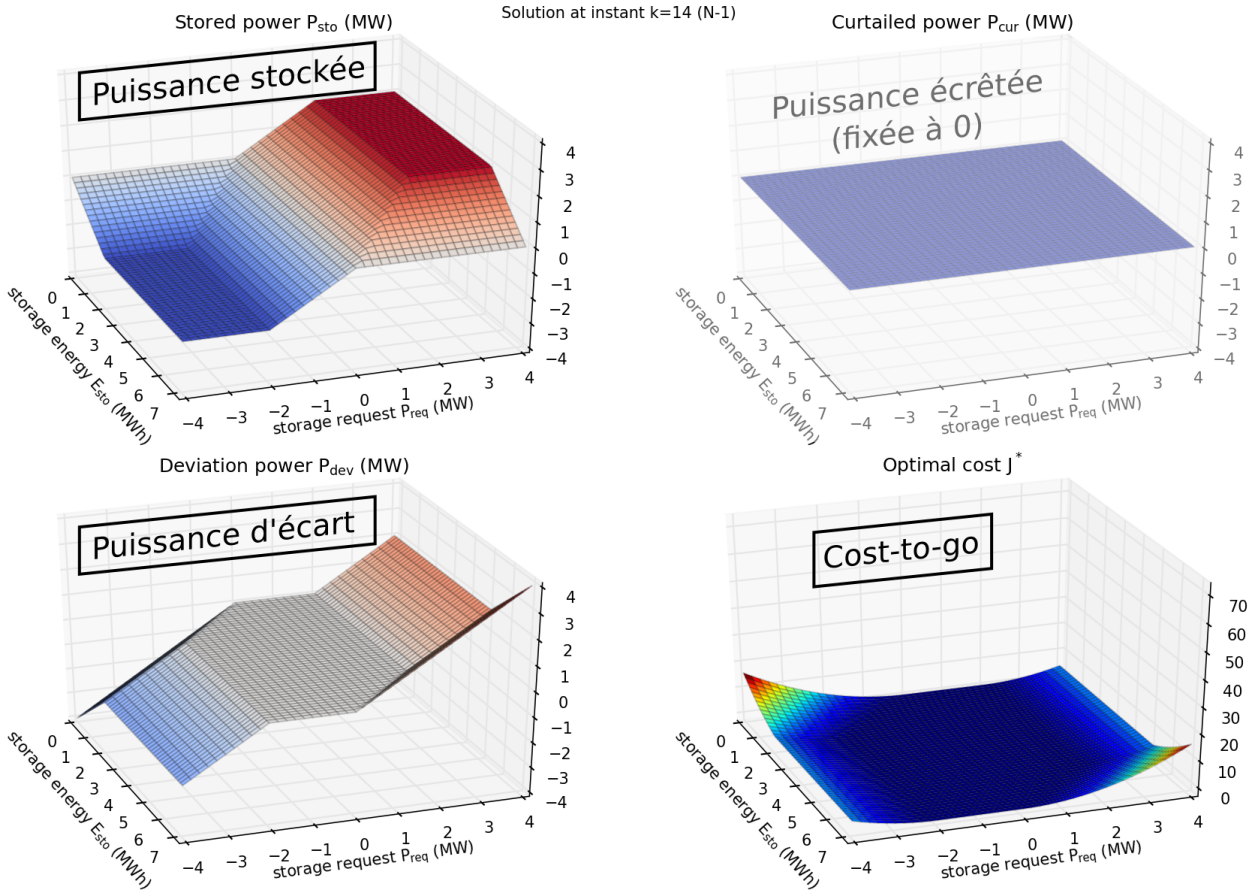
Observons chaque itération de l'algorithme
Point de départ :

$$J_K(x_K) = 0 \quad \text{(coût terminal nul)}$$

puis on remonte le temps, pour $k = K-1, \ldots, 0$:

$$J_k(x_k) = \min_{u_k \in U(x_k)} \mathbb{E}_{w_k} \Big\{ \underbrace{g(x_k, u_k)}_{\text{coût de l'instant}} + \underbrace{J_{k+1}(f(x_k, u_k, w_k))}_{\text{coût du futur}} \Big\}$$

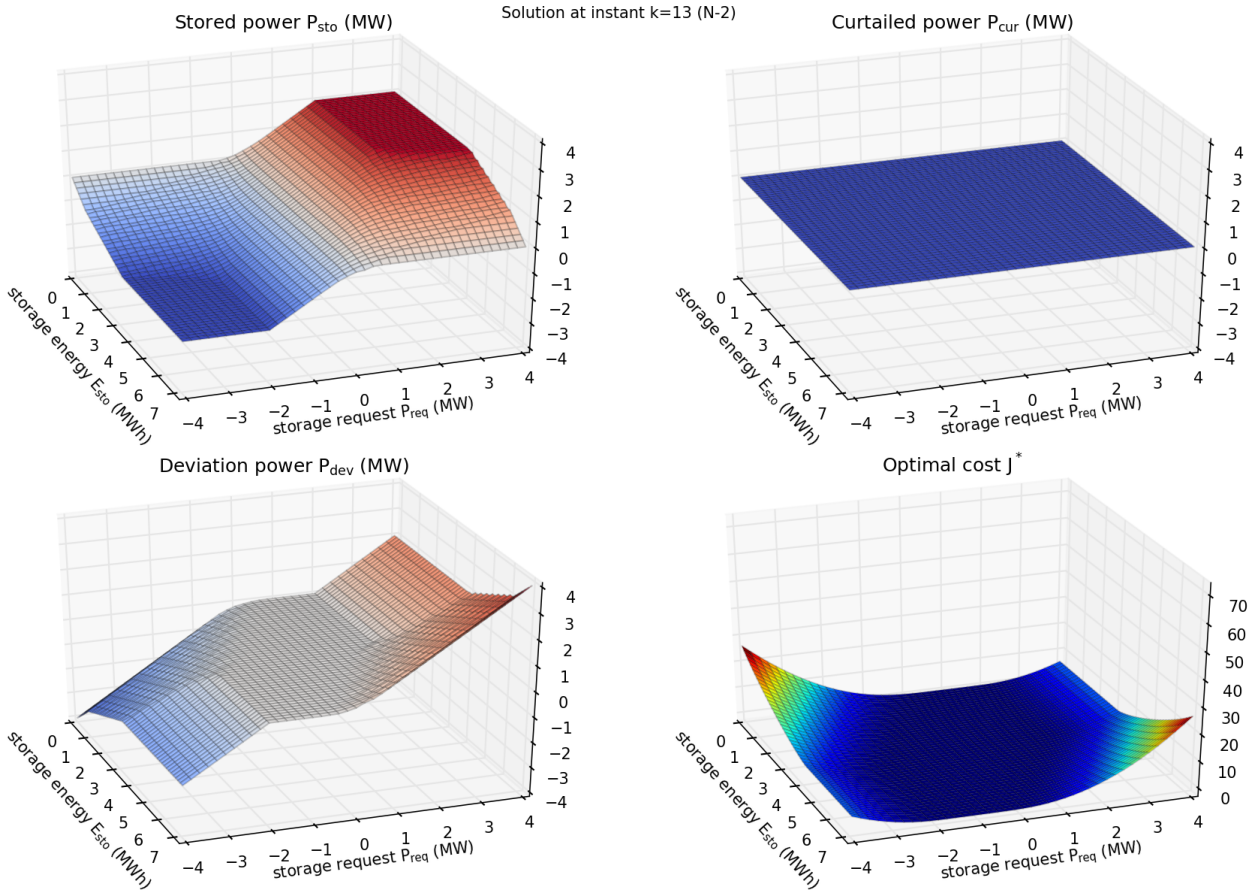
Convergence de l'algorithme : observation



$$k = N - 1$$

(optimisation pour un coût quadratique $\sum_k P_{dev}^2(k)$)

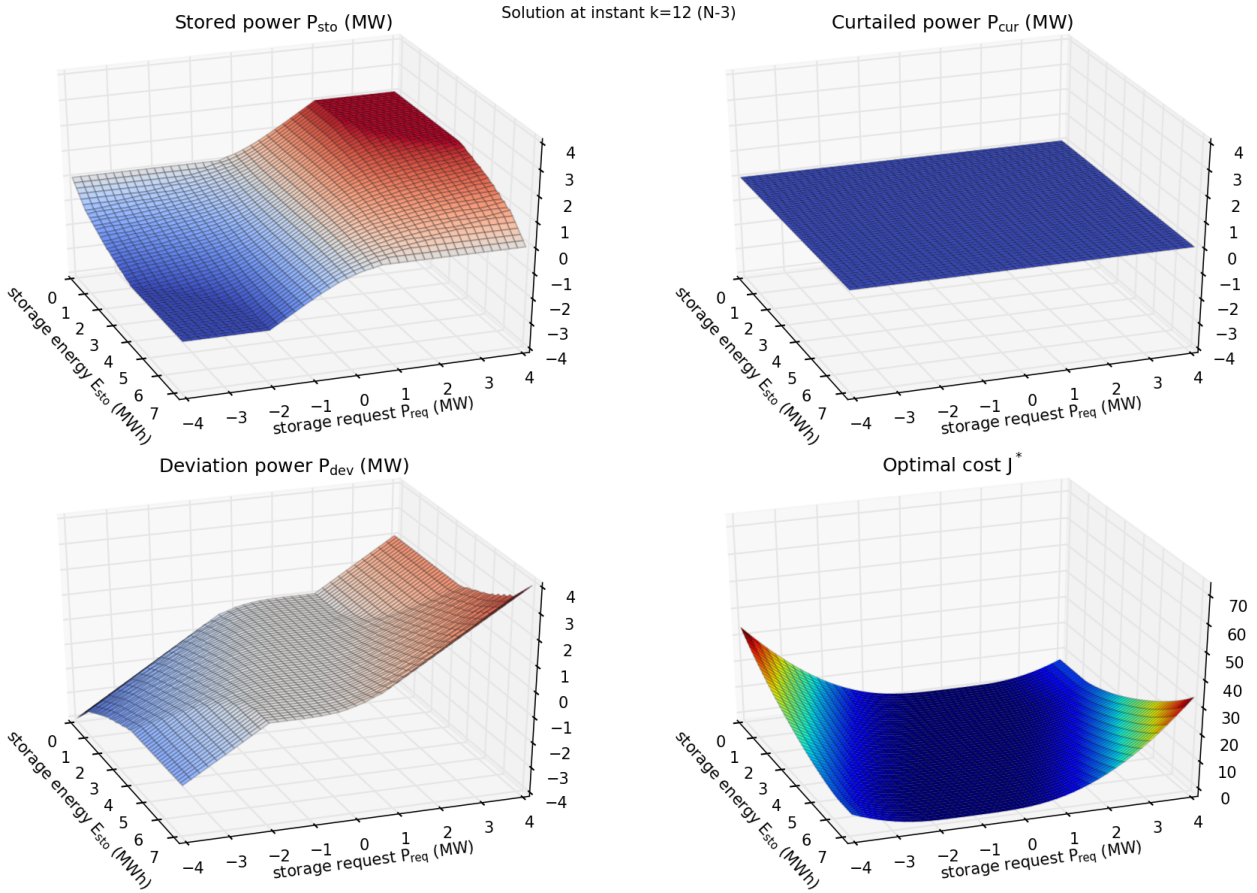
Convergence de l'algorithme : observation



$$k = N - 2$$

(optimisation pour un coût quadratique $\sum_k P_{dev}^2(k)$)

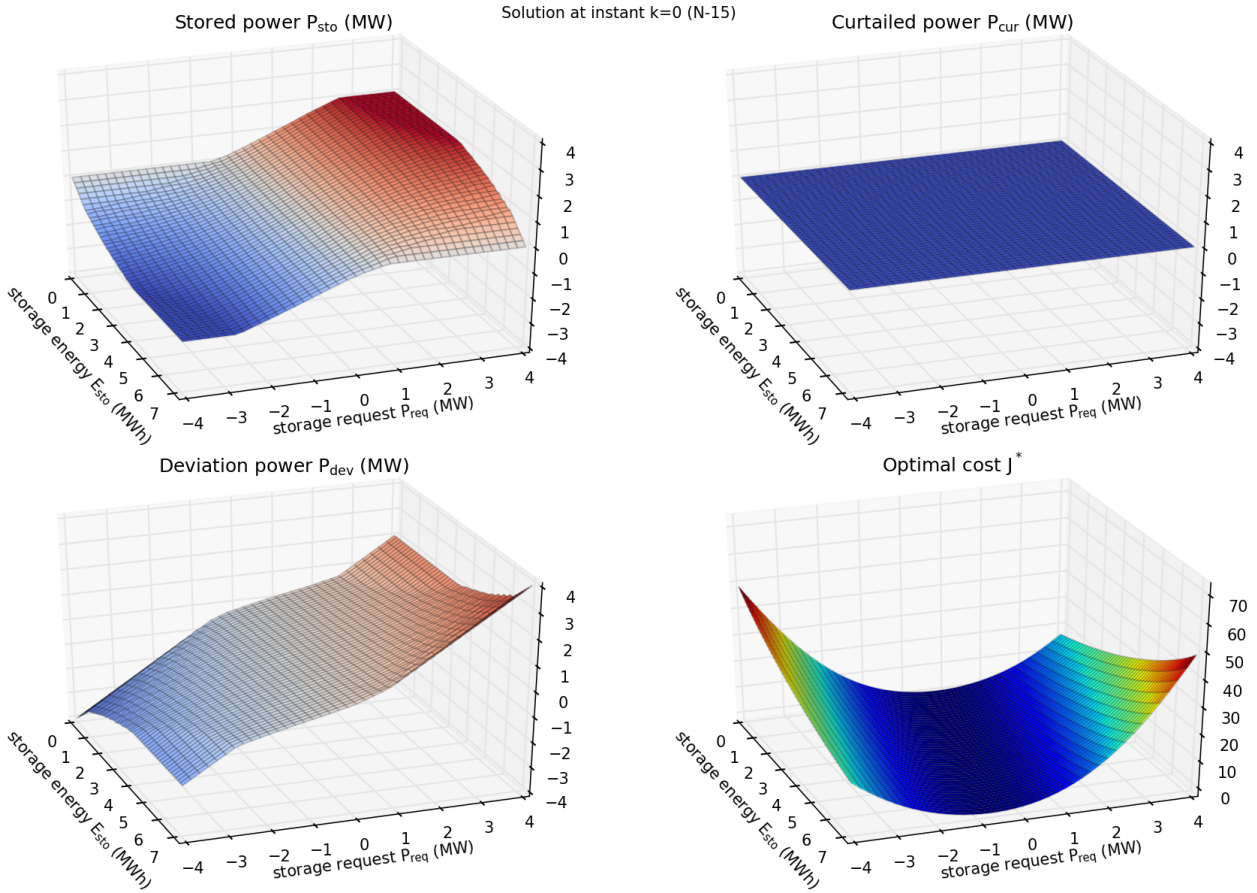
Convergence de l'algorithme : observation



$$k = N - 2$$

(optimisation pour un coût quadratique $\sum_k P_{dev}^2(k)$)

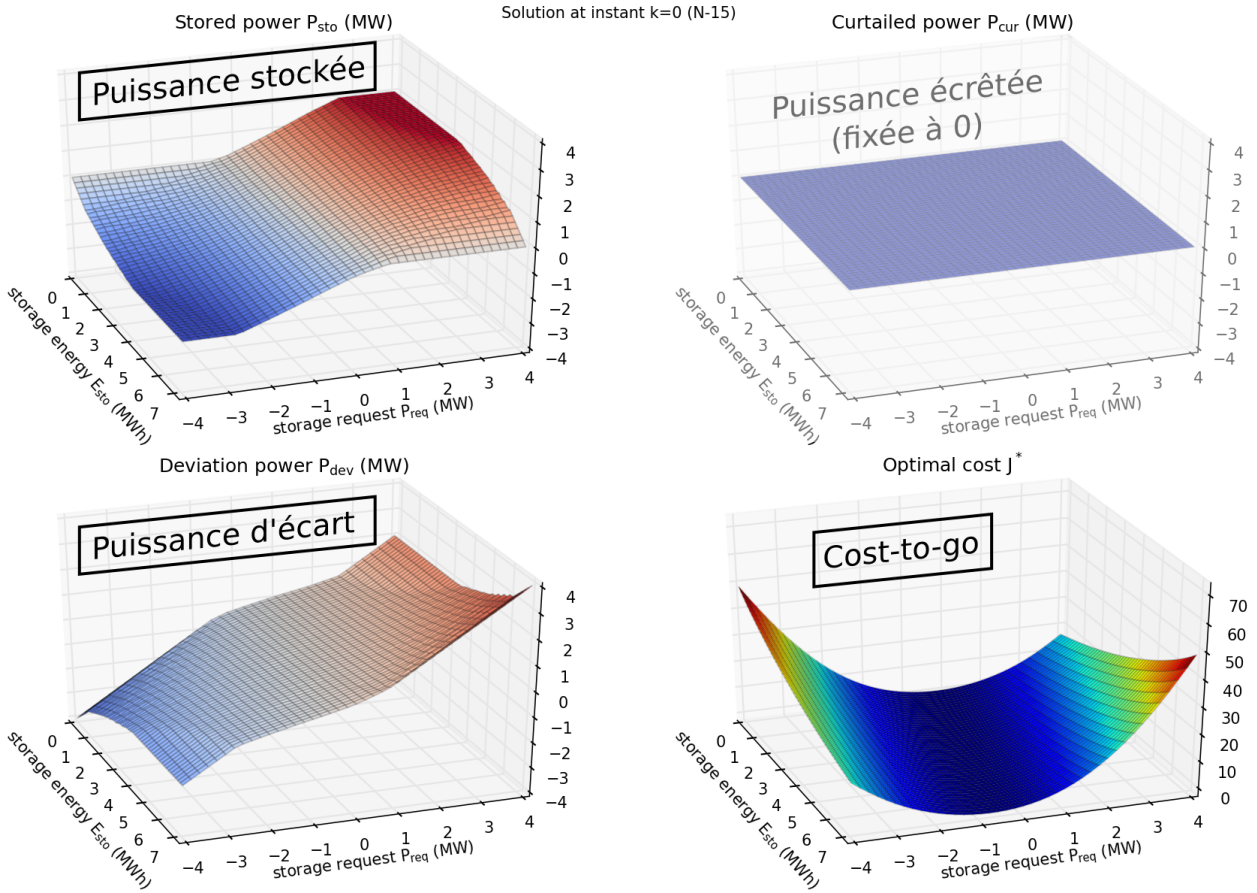
Convergence de l'algorithme : observation



$$k = N - 15$$

(optimisation pour un coût quadratique $\sum_k P_{dev}^2(k)$)

Convergence de l'algorithme : observation



$$k = N - 15$$

(optimisation pour un coût quadratique $\sum_k P_{dev}^2(k)$)

Effet de la fonction coût

$$J = \frac{1}{K}\,\mathbb{E}\Big\{ \sum_{k=0}^{K-1} g(x_k, u_k, w_k) \Big\} \quad \text{avec } K \to \infty$$

Comme le choix de la fonction coût instantané $g(\dots)$ est libre, on peut faire varier sa forme :

- coût valeur absolue $|P_{dev}|$
- coût quadratique P_{dev}^2
- coût avec seuil de tolérance

Effet de la fonction coût

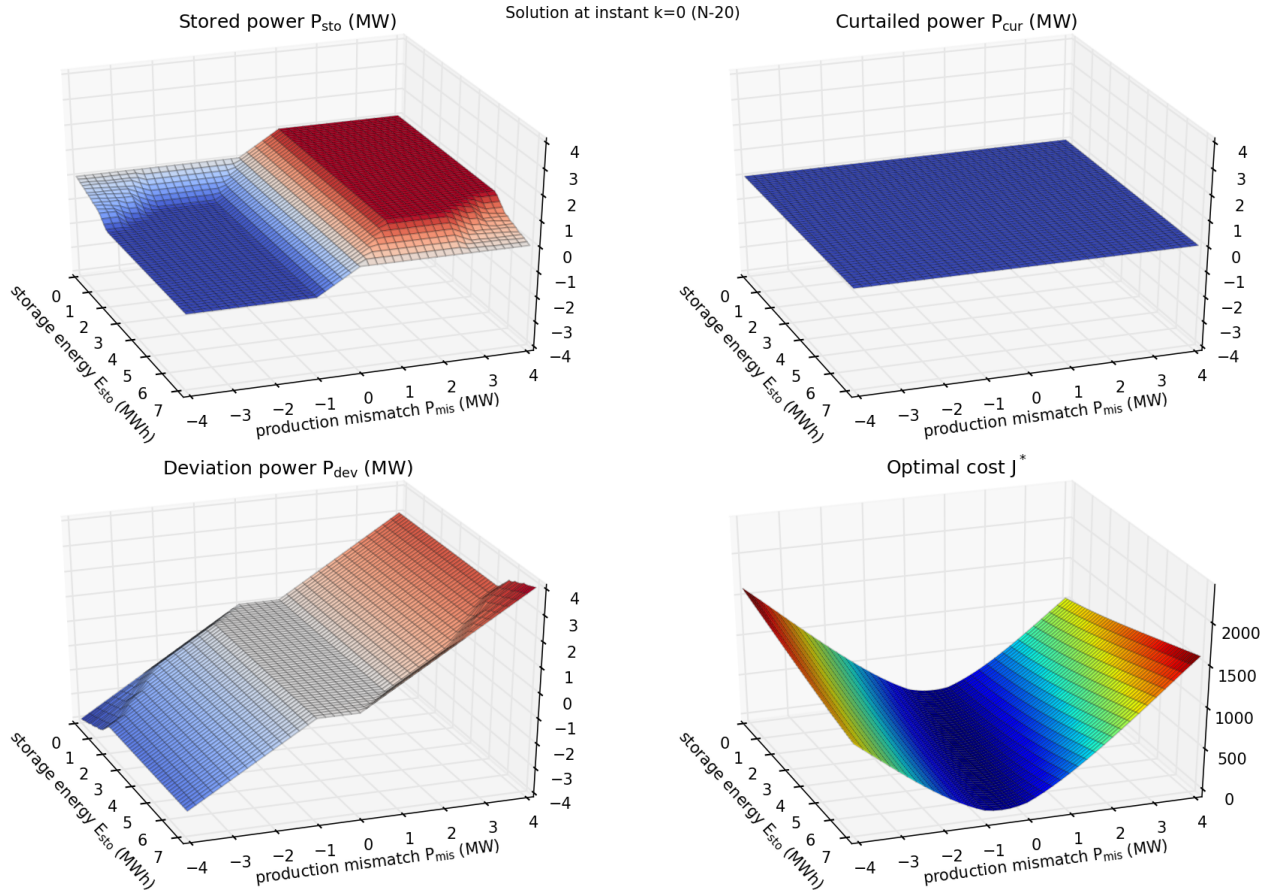
$$J = \frac{1}{K}\,\mathbb{E}\Big\{ \sum_{k=0}^{K-1} g(x_k, u_k, w_k) \Big\} \quad \text{avec } K \to \infty$$

Comme le choix de la fonction coût instantané $g(\dots)$ est libre, on peut faire varier sa forme :

- coût valeur absolue $|P_{dev}|$
- coût quadratique P_{dev}^2
- coût avec seuil de tolérance

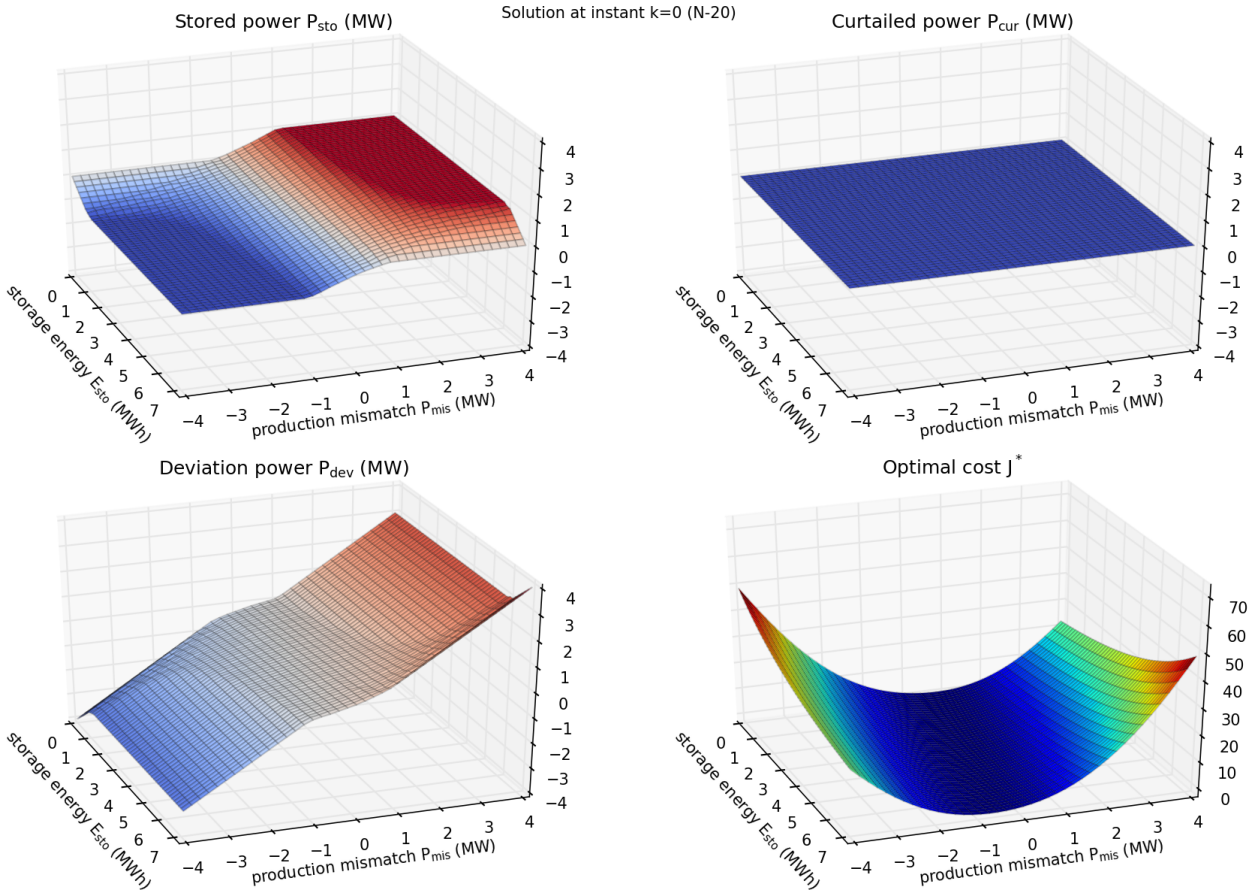
On observe le résultat du point de vue de la loi de gestion, puis par des trajectoires temporelles.

Loi de gestion pour différentes formes de coûts



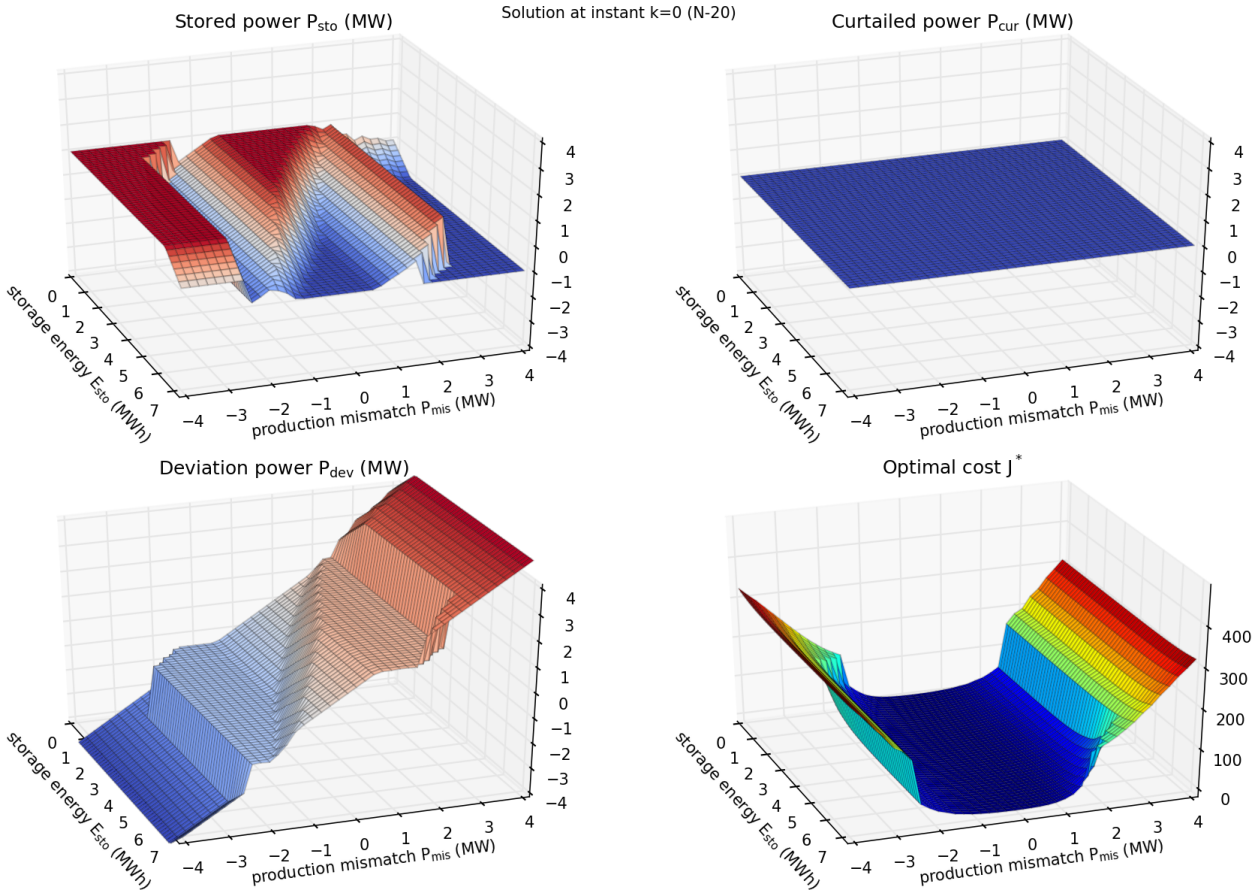
(coût linéaire, prenant en compte pertes & vieillissement)

Loi de gestion pour différentes formes de coûts



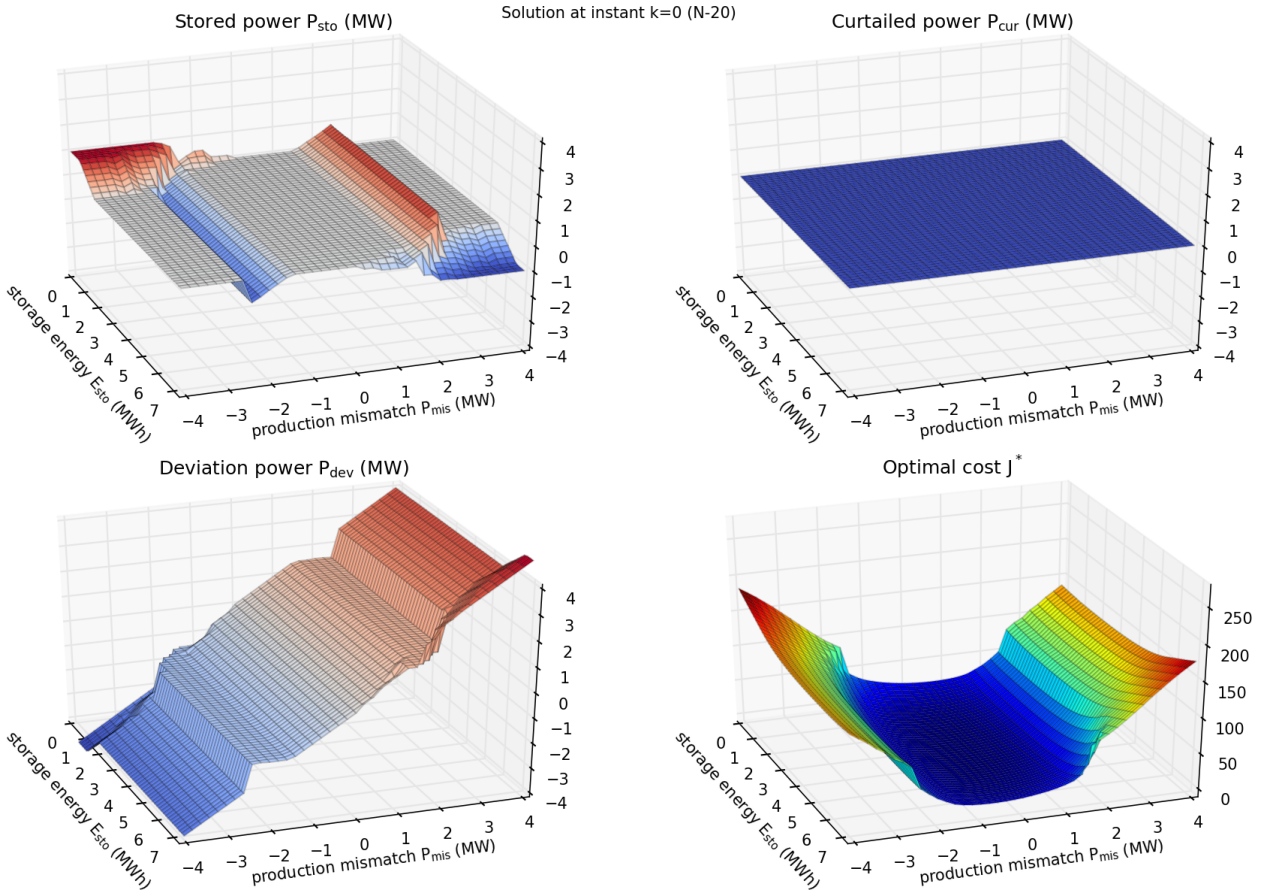
(coût quadratique)

Loi de gestion pour différentes formes de coûts



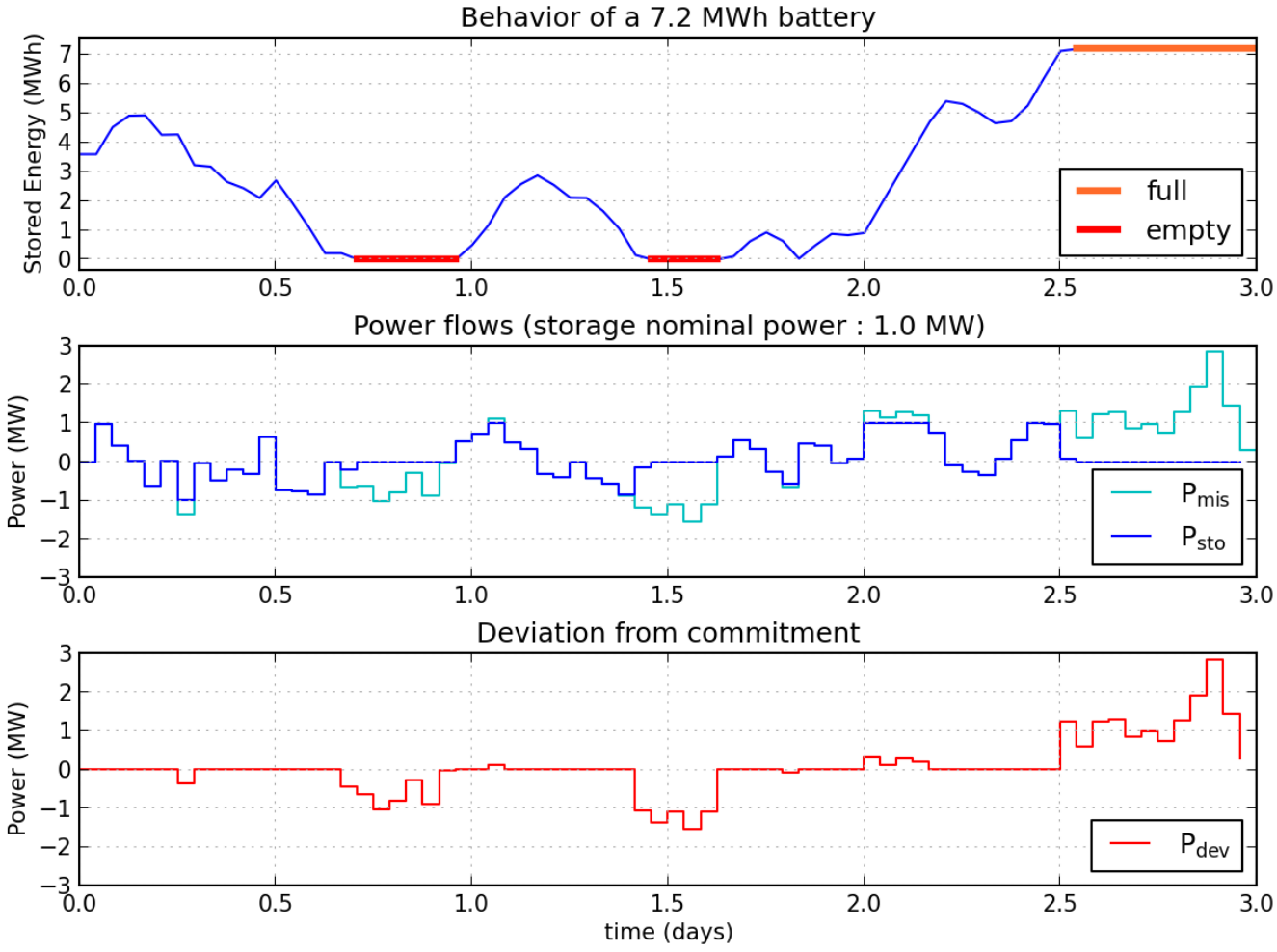
(coût seuil à ± 1.5 MW)

Loi de gestion pour différentes formes de coûts



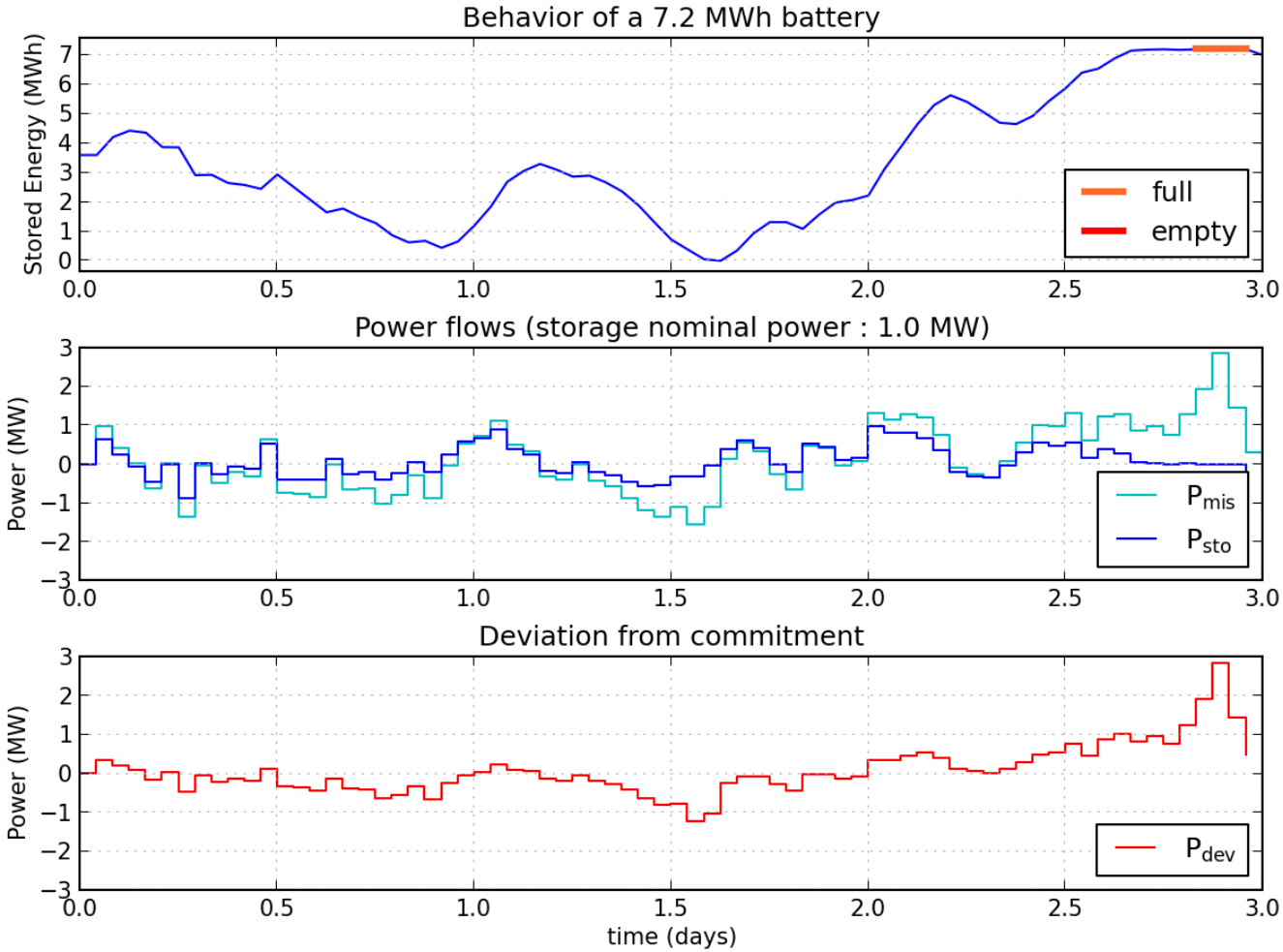
(coût seuil à ± 1.5 MW, prenant en compte pertes & vieillissement)

Trajectoires pour différentes formes de coûts



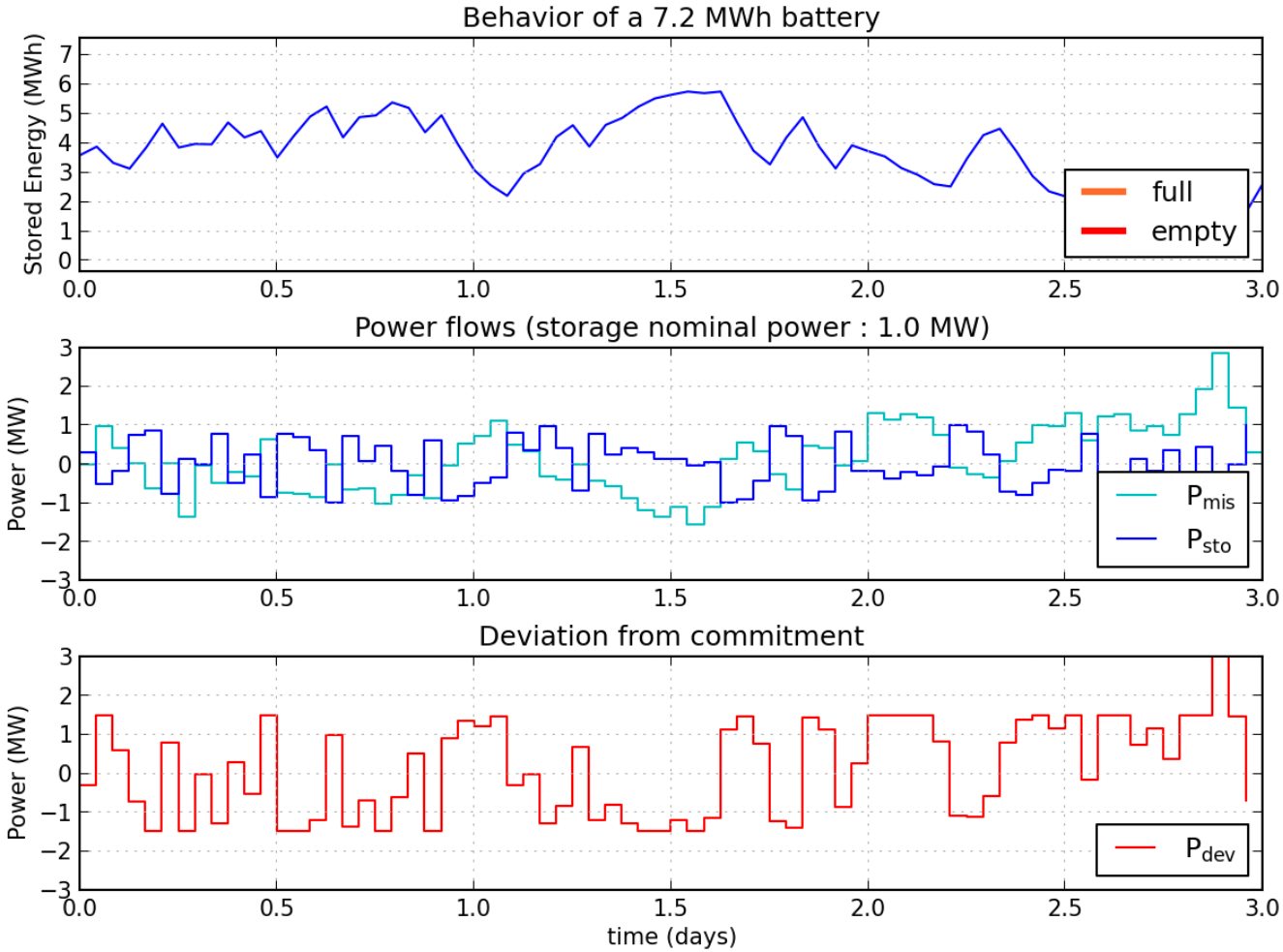
(coût linéaire, prenant en compte pertes & vieillissement)

Trajectoires pour différentes formes de coûts



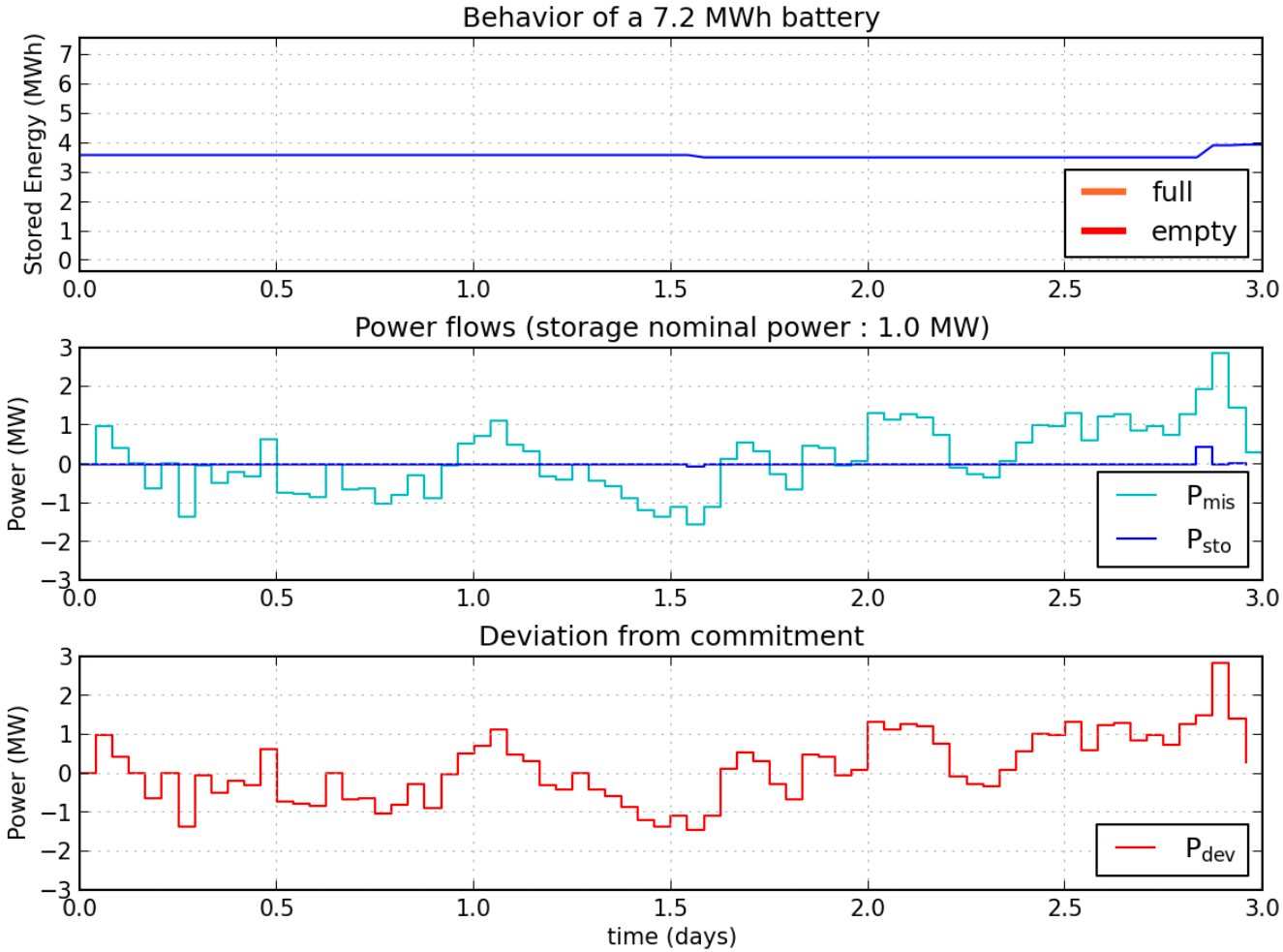
(coût quadratique)

Trajectoires pour différentes formes de coûts



(coût seuil à ± 1.5 MW)

Trajectoires pour différentes formes de coûts



(coût seuil à ± 1.5 MW, prenant en compte pertes & vieillissement)

Effet du choix de la fonction coût

Pour un problème donné, la SDP permet de comparer une *large palette* de fonctions de pénalisation. Pas de restrictions sur la forme des pénalités.

On constate sur ces résultats préliminaires un très fort impact du choix de la fonction coût. Il faudra en particulier faire attention :

- à la façon de pénaliser l'écrêtage P_{cur}
  (résultats non présentés sur les graphiques précédents)
- aux effets de seuil sur la pénalisation

Plan de la présentation

1. Qu'est-ce que la (S)DP ?
2. Application à un système éolien-stockage
3. Conclusion
   - Avantages-inconvénients de la méthode
   - Références

Avantages de la méthode SDP

La SDP ne nécessite pas d'hypothèses sur :

- la dynamique du système (pas nécessairement linéaire)
- la fonction coût (pas nécessairement convexe, ou quadratique)

Elle est donc très flexible dans ses cas d'applications.

Avantages de la méthode SDP

La SDP ne nécessite pas d'hypothèses sur :

- la dynamique du système (pas nécessairement linéaire)
- la fonction coût (pas nécessairement convexe, ou quadratique)

Elle est donc très flexible dans ses cas d'applications.

Cependant, toute *hypothèse supplémentaire* peut être utilisée pour rendre la méthode *plus efficace*. Exemples :

- système linéaire, coût quadratique : contrôle LQ
- système linéaire, coût convexe : SDDP (Pereira & Pinto 1991)

Inconvénients de la méthode SDP

the "Curse of Dimensionality"

La complexité temporelle du problème dynamique a été cassée par l'introduction de la fonction *cost-to-go* :

$$J_k(x), \quad x \text{ étant l'état}$$

Cependant, cette fonction n'a en générale *pas de forme particulière* (e.g. quadratique).
On calcule alors $J_k(x)$ sur une grille en *discrétisant l'état*.

Inconvénients de la méthode SDP

the "Curse of Dimensionality"

La complexité temporelle du problème dynamique a été cassée par l'introduction de la fonction *cost-to-go* :

$$J_k(x), \quad x \text{ étant l'état}$$

Cependant, cette fonction n'a en générale *pas de forme particulière* (e.g. quadratique).
On calcule alors $J_k(x)$ sur une grille en *discrétisant l'état*.
Dans l'exemple éolien-stockage précédent, l'état x vit dans $\mathbb{R}^2$. Si discrétise 100 pts par dimension, on obtient une grille à 100^2 pts.

Inconvénients de la méthode SDP

the "Curse of Dimensionality"

La complexité temporelle du problème dynamique a été cassée par l'introduction de la fonction *cost-to-go* :

$$J_k(x), \quad x \text{ étant l'état}$$

Cependant, cette fonction n'a en générale *pas de forme particulière* (e.g. quadratique).
On calcule alors $J_k(x)$ sur une grille en *discrétisant l'état*.
Dans l'exemple éolien-stockage précédent, l'état x vit dans $\mathbb{R}^2$. Si discrétise 100 pts par dimension, on obtient une grille à 100^2 pts.

La taille de grille augmente **exponentiellement** avec la dimension de l'état.

En pratique, on est donc limité à la dimension **1 à 3**.

Les palliatifs au problème de la dimension

Notons que le problème du la Malédiction de la Dimension n'est pas un problème de la méthode SPD, mais un *problème des problèmes* d'optimisation dynamique stochastique.

Les palliatifs au problème de la dimension

Notons que le problème du la Malédiction de la Dimension n'est pas un problème de la méthode SPD, mais un *problème des problèmes* d'optimisation dynamique stochastique.

Il existe beaucoup de "soins palliatifs" pour réanimer la SPD sur des problèmes de dimension $n > 3$.
Sous le terme "Programmation Dynamique Approximée" (ADP) on trouve :

- Neuro-Dynamic Programming
- Q-learning
- Rollout Policies, Limited Lookahead Policies
- Model Predictive Control
- . . .*liste non-exhaustive (cf. livre de D. Bertsekas)*

(cela reste des approximations, résultats non garantis)

Quelques références
sur la méthode

Livre sur la Programmation Dynamique (écrit par un "électricien")

- D. P. Bertsekas, "Dynamic Programming and Optimal Control" Athena Scientific, 3rd ed., 2005.

À propos des méthodes particulières

- M. Pereira and L. Pinto, "Multi-stage stochastic optimization applied to energy planning" Mathematical Programming, vol. 52, no. 1-3, p. 359–375, 1991.
- W. B. Powell, "Perspectives of approximate dynamic programming" Annals of Operations Research, p. 1–38, 2012.

Quelques références
sur des applications

Gestion optimale de barrage

- J.-C. Alais and M. De Lara, "Chance-Constrained and Stochastic Viable Management of an Hydroelectric Dam"
in CLAIO-SBPO, Rio de Janeiro, 24-28 Sept. 2012.

Gestion de stockage avec un houlogénérateur

- P. Haessig, T. Kovaltchouk, B. Multon, H. Ben Ahmed, and S. Lascaud, "Computing an Optimal Control Policy for an Energy Storage", *accepté à* EuroSciPy 2013, Bruxelles août 2013

Modélisation *autorégressive* des erreurs de prévision

- P. Haessig, B. Multon, H. Ben Ahmed, S. Lascaud, and P. Bondon, "Energy storage sizing for wind power : impact of the autocorrelation of day-ahead forecast errors", article *soumis à* Wind Energy

- P. E. de Mello, N. Lu, and Y. Makarov,
"An optimized autoregressive forecast error generator for wind and load uncertainty study", Wind Energy, vol. 14, no. 8, p. 967–976, 2011