# Gestion d'énergie dans un contexte éolien

## avec la Programmation Dynamique Stochastique

Pierre Haessig

EDF R&D LME, ENS Rennes SATIE
pierre.haessig@ens-rennes.fr

Supélec Rennes, 22 avril 2014

# Pierre Haessig
## situation actuelle

Doctorant EDF R&D, en fin de 3ème année, au laboratoire SATIE à l'ENS Rennes.

**Sujet de thèse**

Optimisation du **dimensionnement** et de la **gestion** d'un système de stockage intégré à la production éolienne.

soutenance prévue le 17 juillet 2014

4 activités principales :

- *modélisation énergétique* des systèmes (e.g. batteries),
- *modélisation statistique* des entrées incertaines (M2 ATSI)

# Pierre Haessig

## situation actuelle

Doctorant EDF R&D, en fin de 3ème année, au laboratoire SATIE à l'ENS Rennes.

**Sujet de thèse**

Optimisation du **dimensionnement** et de la **gestion** d'un système de stockage intégré à la production éolienne.

soutenance prévue le 17 juillet 2014

4 activités principales :

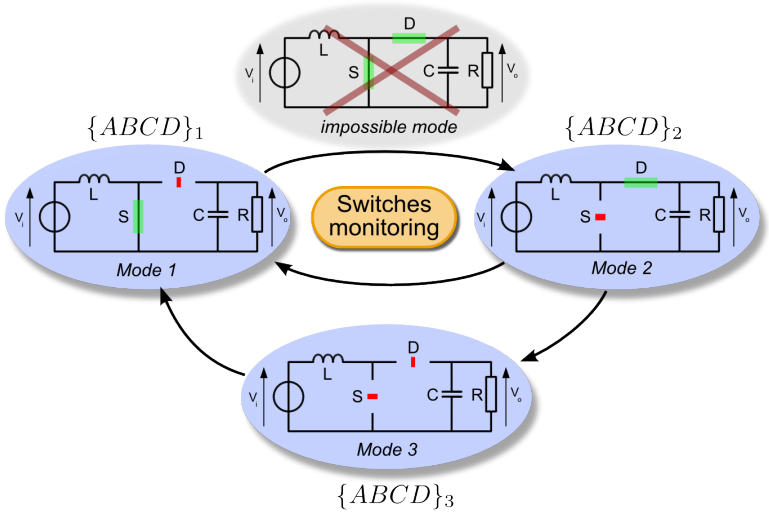
- *modélisation énergétique* des systèmes (e.g. batteries),
- *modélisation statistique* des entrées incertaines (M2 ATSI)
- optimisation dynamique stochastique pour la gestion d'énergie
- étude et optimisation du dimensionnement d'un stockage

# Pierre Haessig

## expérience de recherche précédente

2009 (RLE at MIT) : Modélisation des circuits d'électronique de puissance pour leur simulation en temps-réel ($\Delta_t = 1\mu$s).

Approche *hybrid systems*

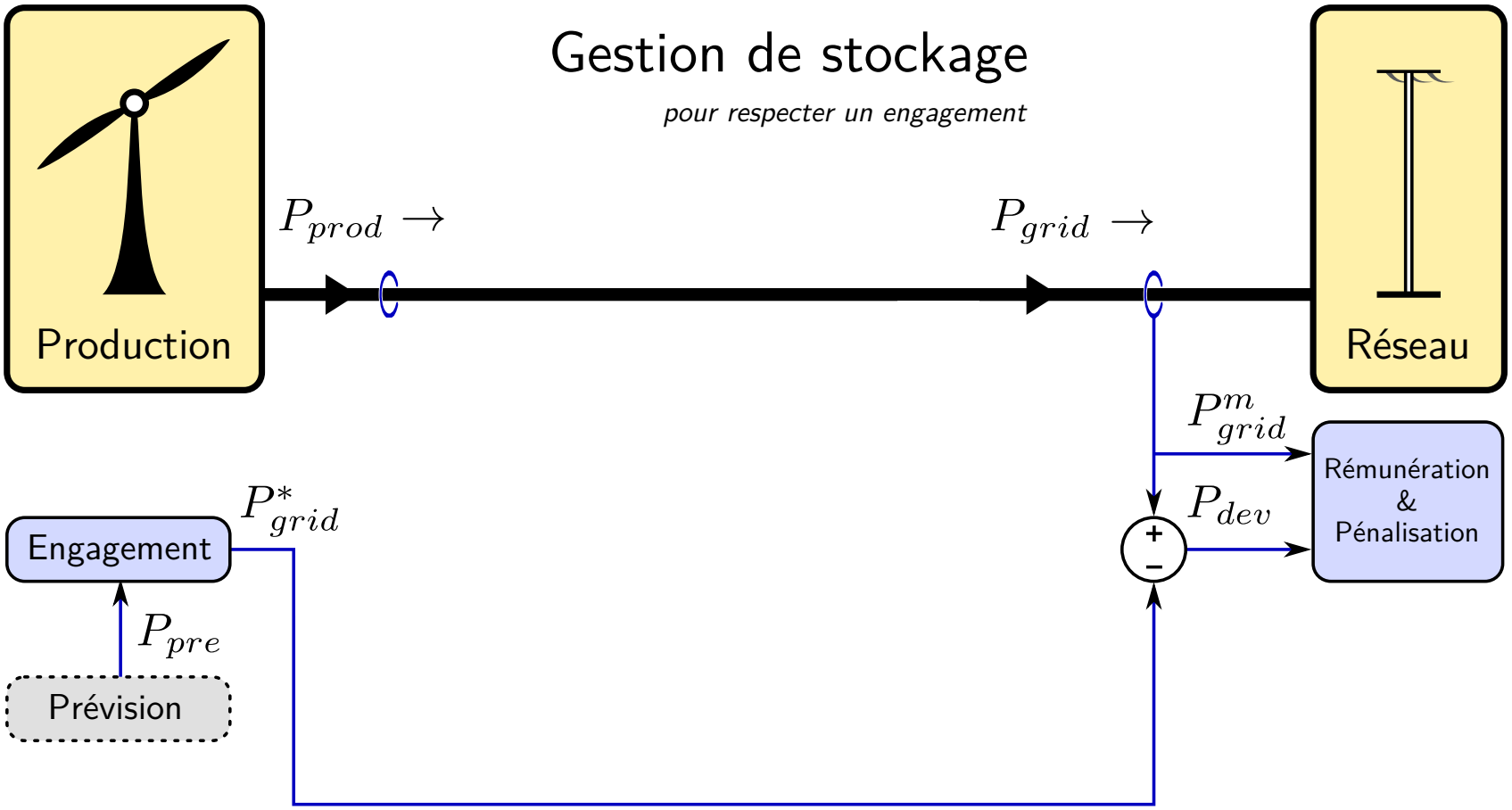



Aujourd'hui : simulateurs Typhoon HIL (Boston, Zürich, Novi Sad)

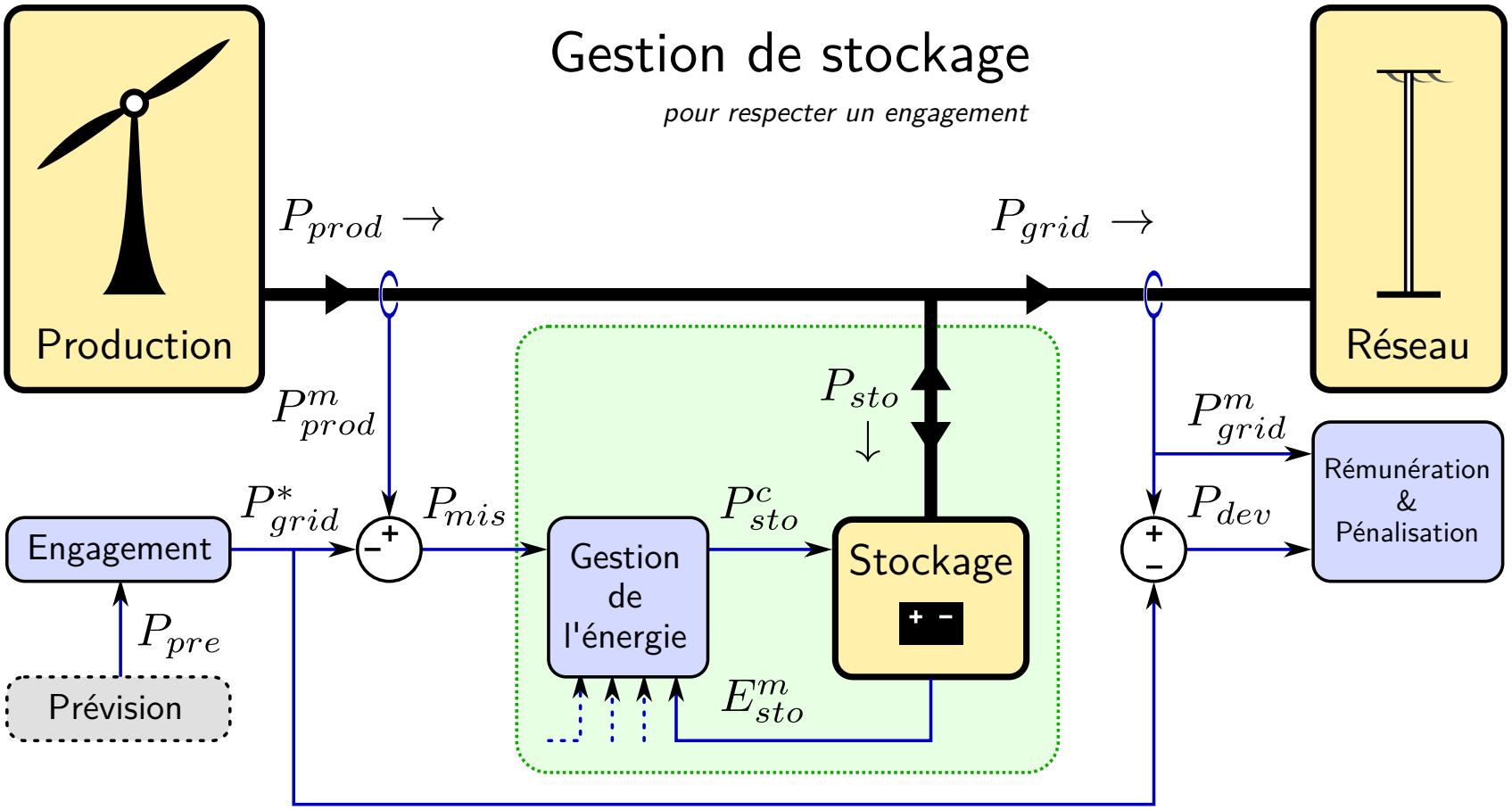
# Plan de la présentation

# Plan de la présentation

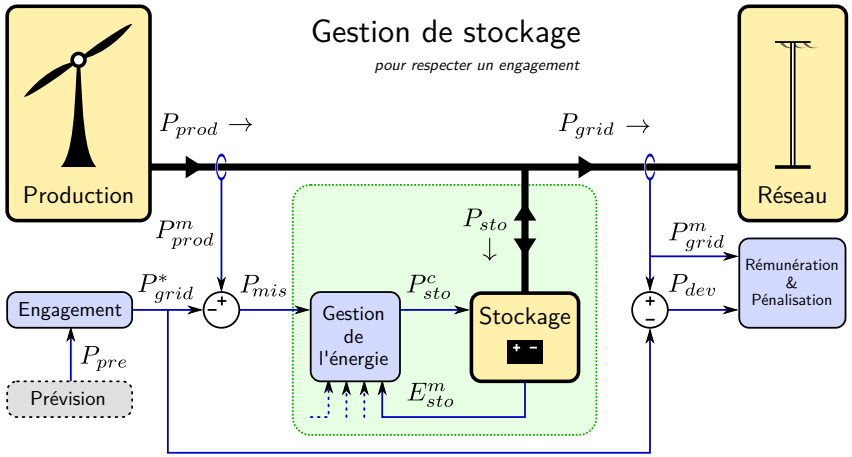
# Système de production éolien avec stockage



# Système de production éolien avec stockage



# Contexte industriel éolien-stockage

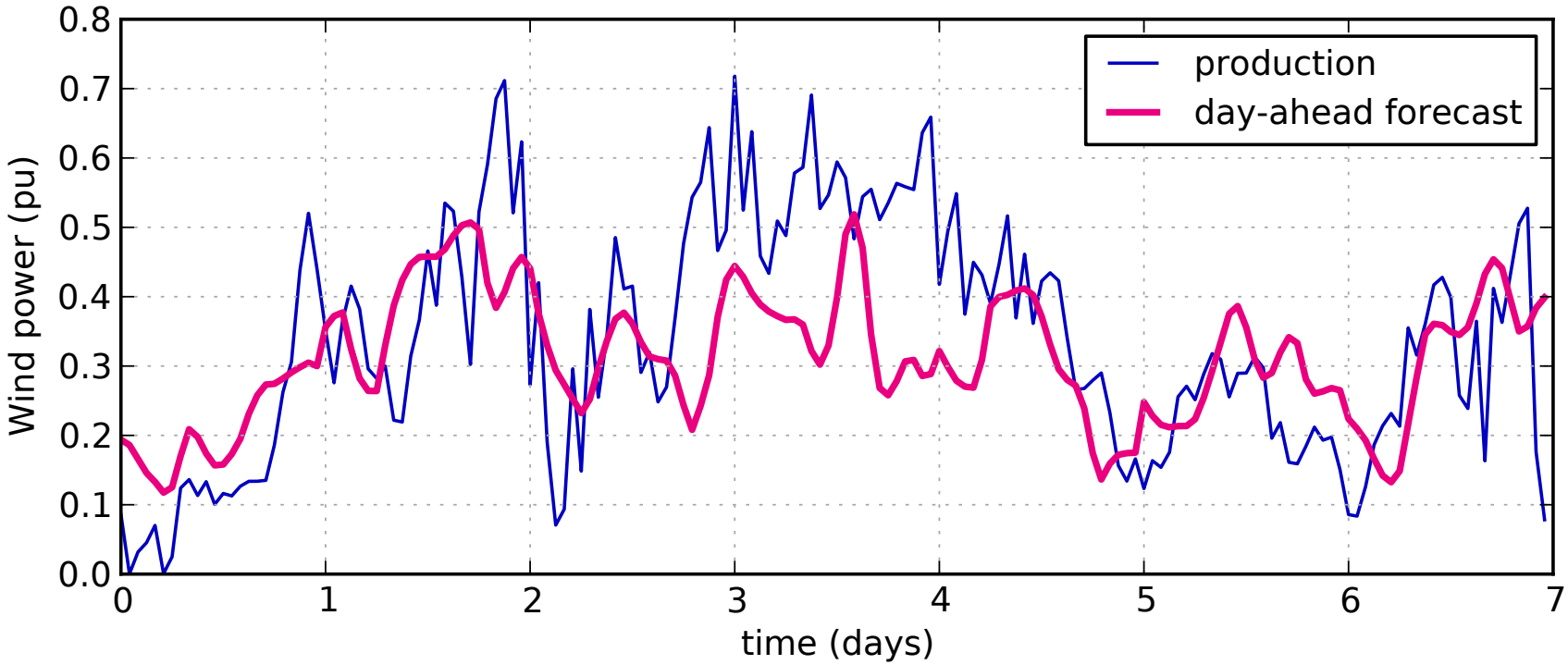



Appel d'offre de la Commission de Régulation de l'Énergie (CRE) pour des systèmes éoliens "avec services" :

- 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.

# Prévision de production J+1

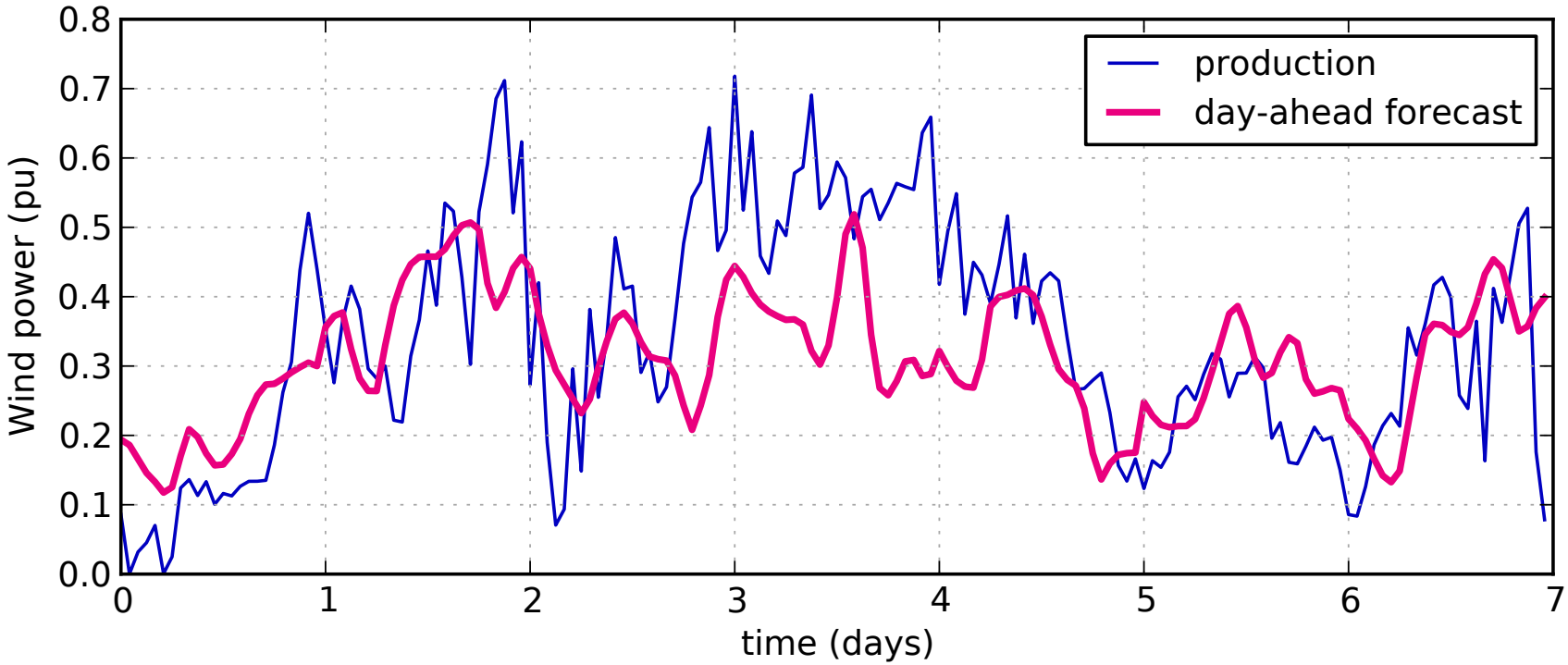
La production éolienne peut être prévue un jour à l'avance, grâce à des outils *météo et statistiques*.




une semaine de prévision/production (moyennes 1h), en Guadeloupe

# Prévision de production J+1

La production éolienne peut être prévue un jour à l'avance, grâce à des outils *météo et statistiques*.

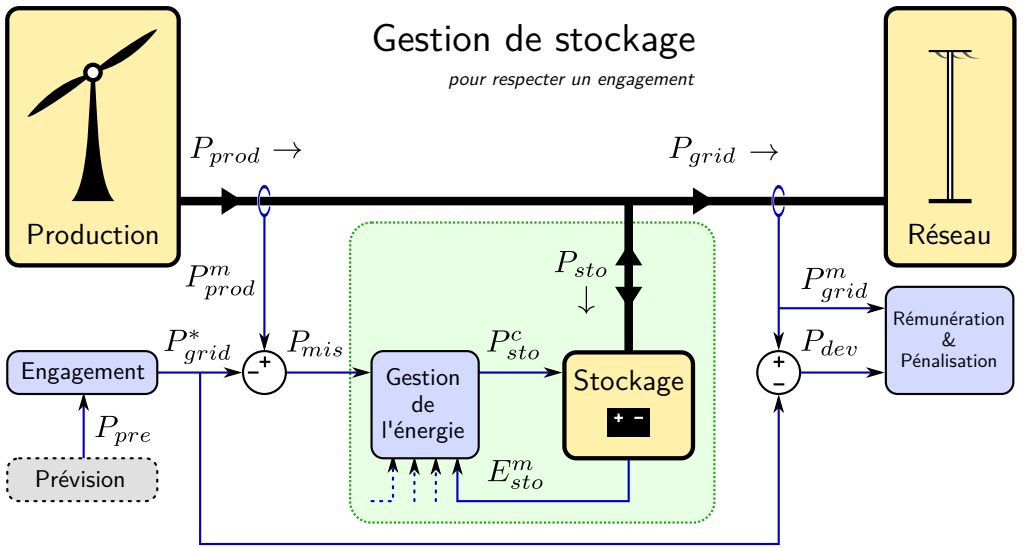



une semaine de prévision/production (moyennes 1h), en Guadeloupe

La prévision J+1 est imparfaite $\rightarrow$ erreur non nulle à compenser...

# Importance de l'erreur de prévision

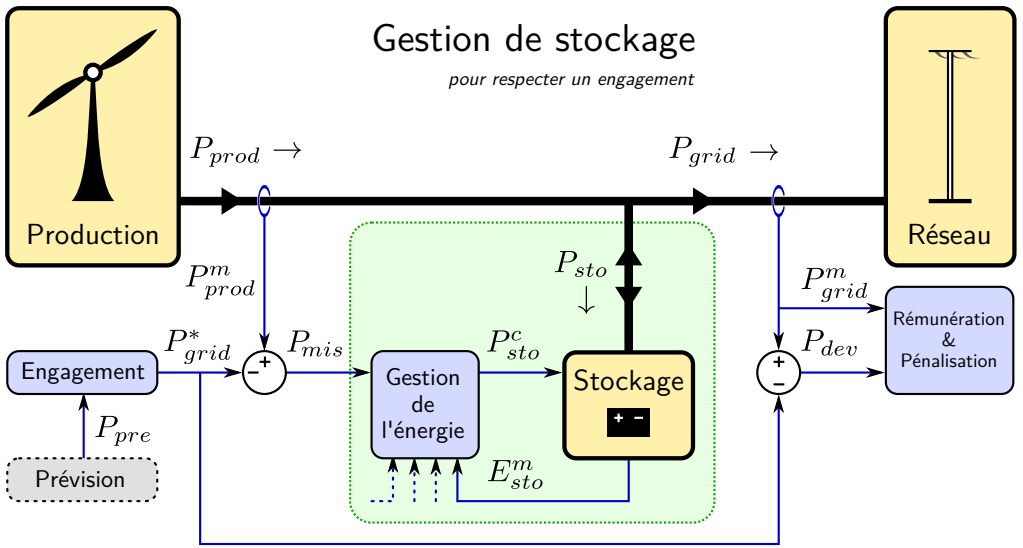
Le stockage est là pour *compenser les erreurs de prévision*.




(hypothèse "engagement J+1 = prévision J+1")

# Importance de l'erreur de prévision

Le stockage est là pour *compenser les erreurs de prévision*.



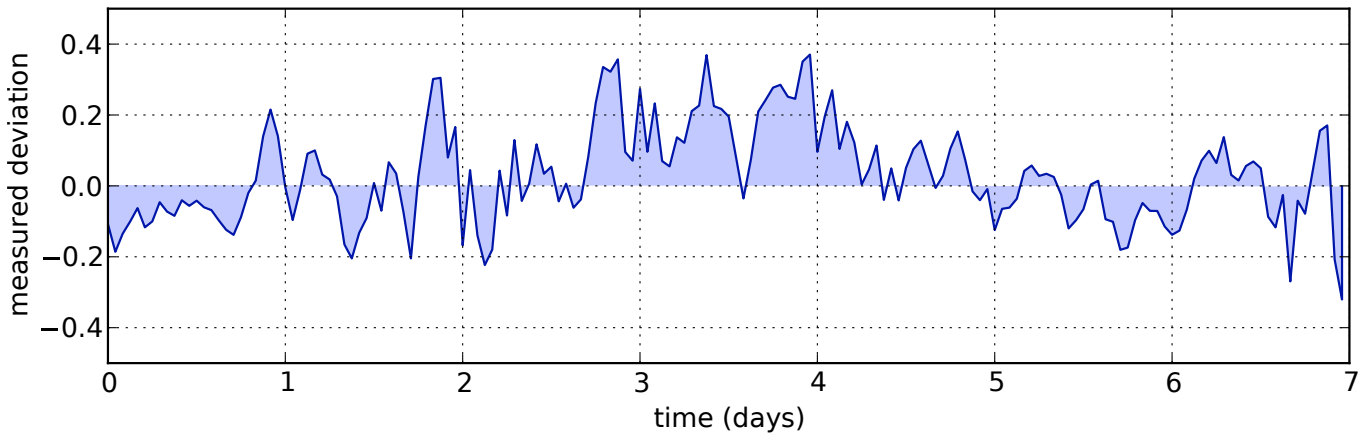

(hypothèse "engagement J+1 = prévision J+1")

## Besoin de modélisation

L'erreur de prévision J+1 est la principale entrée du problème.
Il importe donc de la caractériser.

# Caractérisation de l'erreur de prévision

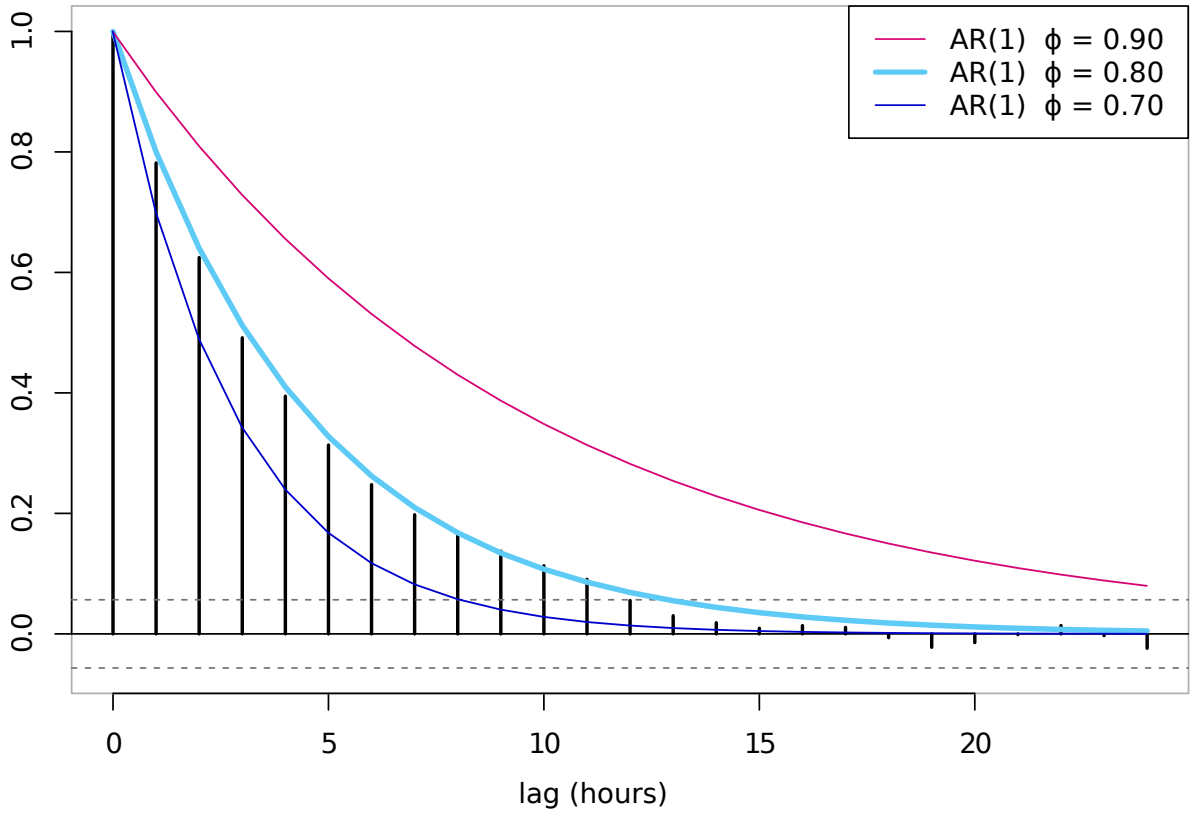
La qualité de la prévision dépend de la complexité du terrain, de l'horizon temporel de prédiction, ...
À l'échelle d'un parc en Guadeloupe : écart-type de **15%** de la puissance nominale.




**Structure temporelle** : les erreurs de prévision J+1, heure par heure ne sont pas *indépendantes*

# Autocorrélation de l'erreur de prévision

La dépendance temporelle (autocorrélation) des erreurs décroît de façon exponentielle



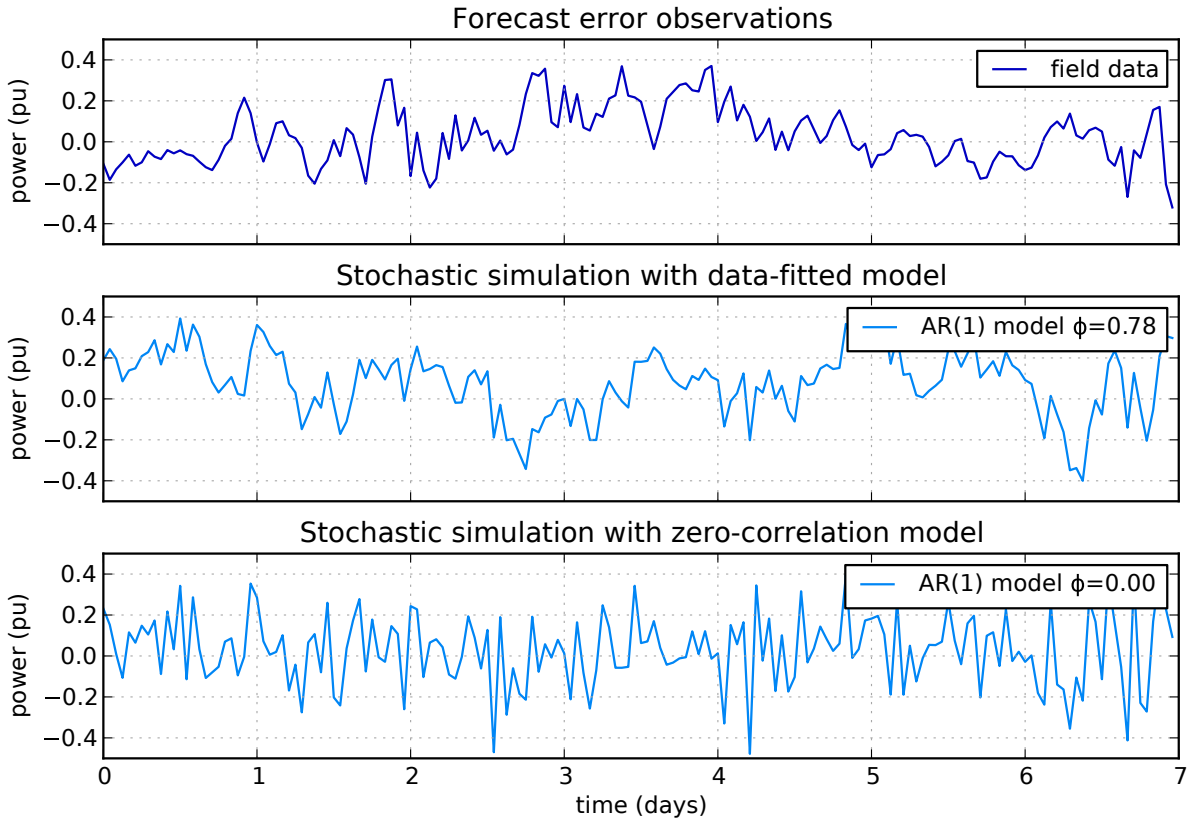

Cette forme d'autocorrélation correspond à un processus AR(1)

# Modèle autorégressif AR(1)

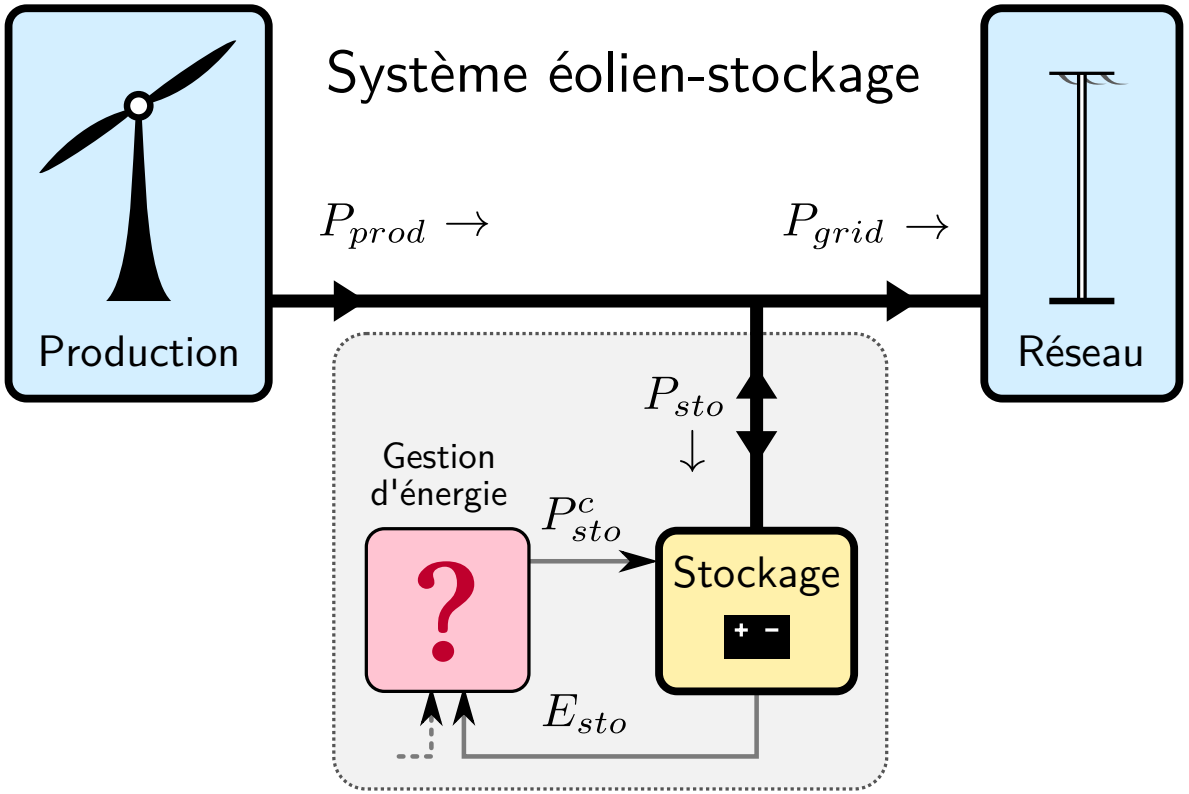
Modèle basé sur filtrage passe-bas d'un bruit blanc $\varepsilon(k)$ :

$$P_{err}(k+1) = \phi P_{err}(k) + \sigma_P \sqrt{1-\phi^2}\varepsilon(k+1)$$

**autorégressif** : chaque valeur est liée à la précédente (par $\phi$)

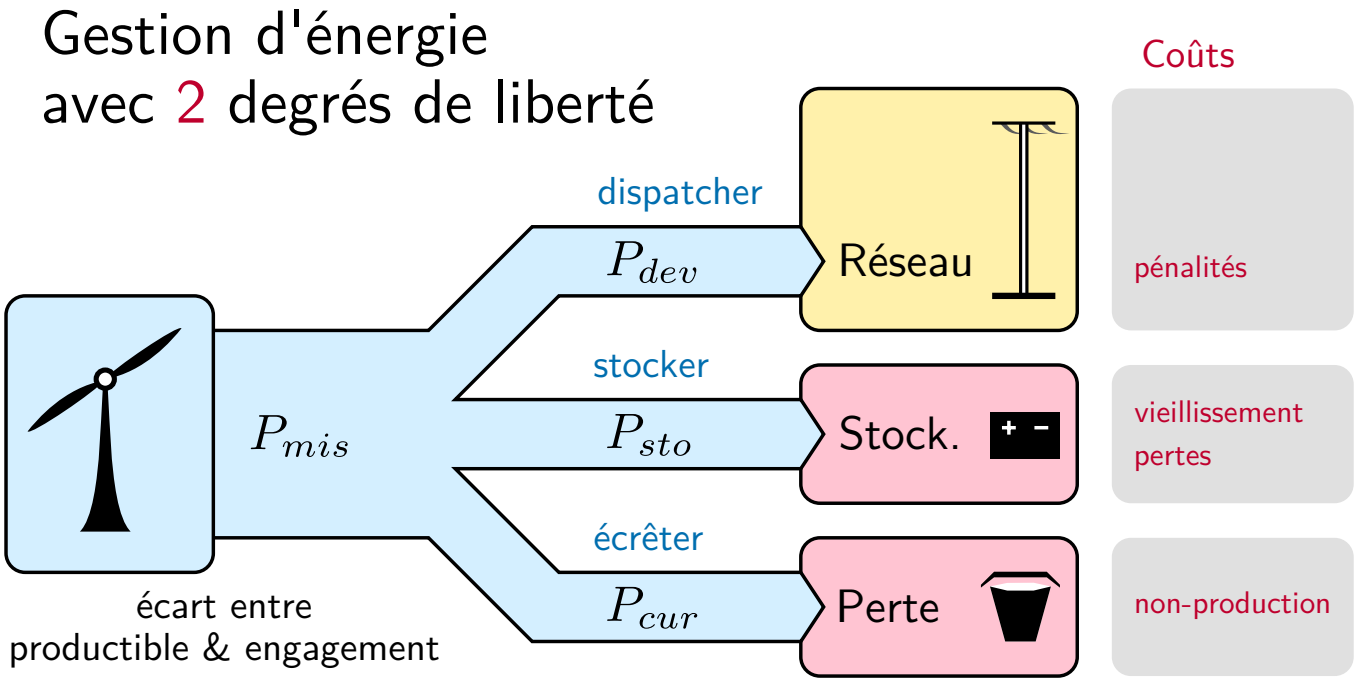


# Description du problème de contrôle




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.

# Plan de la présentation

# 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 $u = (P_{sto})$
- 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 $u = (P_{sto})$
- 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 :

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

Ces équations modélisent le stockage, 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 $0 \leq E + P_{sto}\Delta_t \leq E_{rated}$
- limite de puissance du stockage : $|P_{sto}| \leq P_{rated}$

# Coût de chaque instant

Coût de chaque instant $g(x, u, w)$ : pas besoin d'une forme particulière pour cette pénalisation (ni linéaire, ni quadratique, ni convexe, ...).

Exemple d'une pénalisation linéaire (valeur absolue) de l'écart à l'engagement :

$$g(E, P_{mis}, P_{sto}, w) = c_{dev}|P_{mis} - P_{sto}|$$

$\rightarrow$ *plusieurs exemples étudiés après*

# Programmation Dynamique

## équation de Bellman

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

## équation de Bellman

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 :

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

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 : $P_{sto} \in \texttt{linspace}(-\mathrm{P}_{\texttt{max}}, \mathrm{P}_{\texttt{max}}, 100)$

# 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 : $P_{sto} \in \texttt{linspace}(-\mathrm{P}_{\texttt{max}}, \mathrm{P}_{\texttt{max}}, 100)$

L'étape de minimisation aboutit à une *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* :

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

"Problème à horizon infini"

La résolution se fait itérativement comme pour l'horizon fini (algorithmes "Value Iteration" et "Policy Iteration").

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 fermée

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 $\mu_k^* : x_k \mapsto u_k^*$)

# Contrôle boucle fermée

Propriété importante de la Programmation Dynamique :

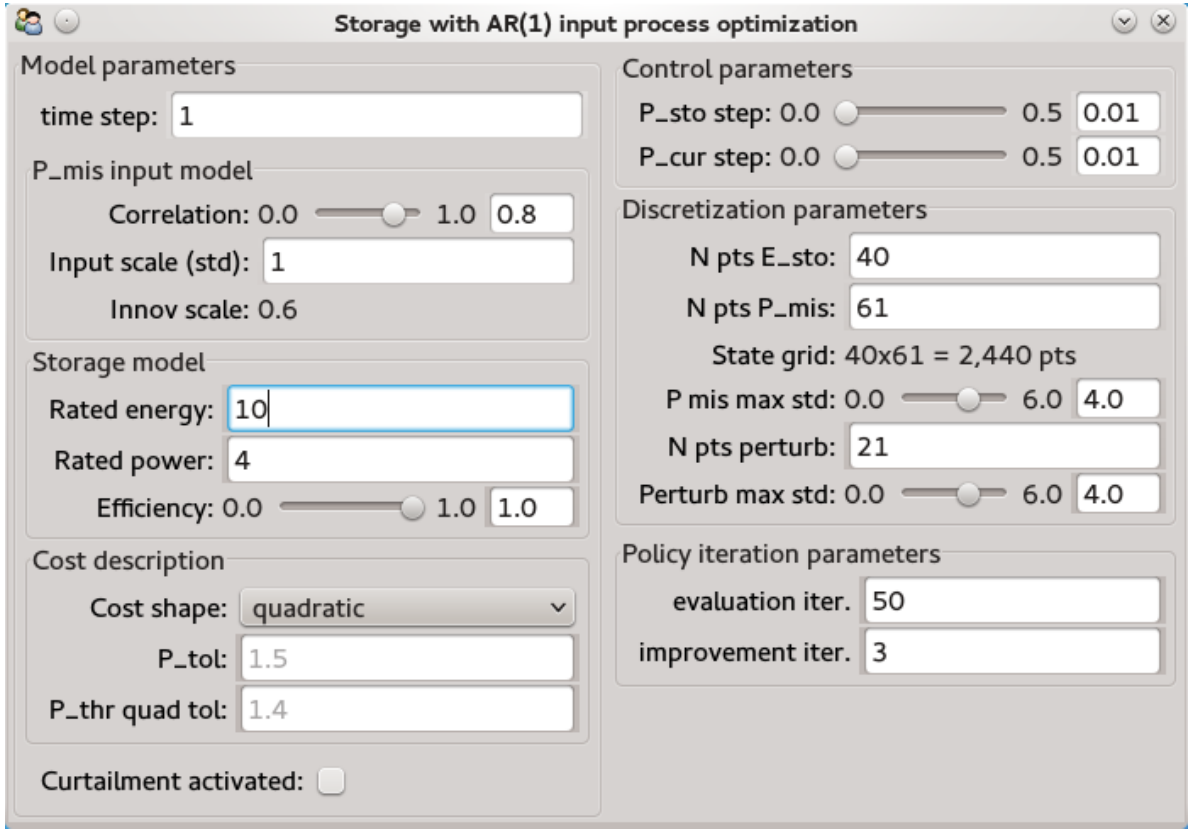
- elle ne donne pas une valeur de commande optimale (i.e. un nombre $u_k^*$)
- mais une loi de gestion optimale (i.e. une fonction de l'état $\mu_k^* : x_k \mapsto u_k^*$)

Le fait que le "produit" de l'optimisation soit une fonction, et pas un nombre, a ses avantages et ses inconvénients (calcul $++$).

Dans le cas déterministe, travailler avec une loi de gestion (un contrôle "boucle fermée") est *superflu*, mais dans le cas stochastique c'est *structurellement indispensable* pour atteindre l'optimum.

# Mise en œuvre logicielle

Paramètres du système et paramètres de résolution :




Résolution avec `StoDynProg`, logiciel présenté à EuroSciPy 2013

`https://github.com/pierre-haessig/stodynprog`

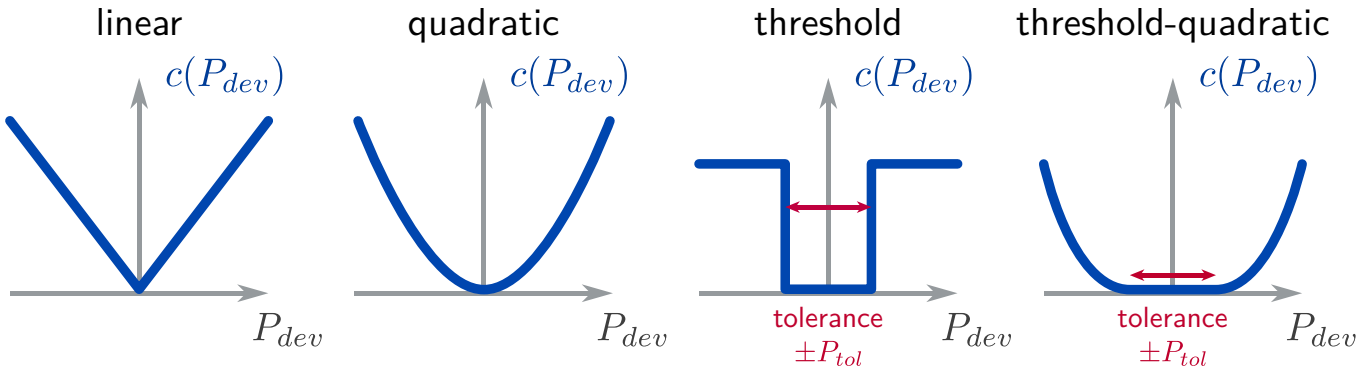
# Une étude : effet de la fonction coût

Coût à minimiser : $J = \mathbb{E}\{g(x_k, u_k, w_k)\}$

# Une étude : effet de la fonction coût

Coût à minimiser : $J = \mathbb{E}\{g(x_k, u_k, w_k)\}$

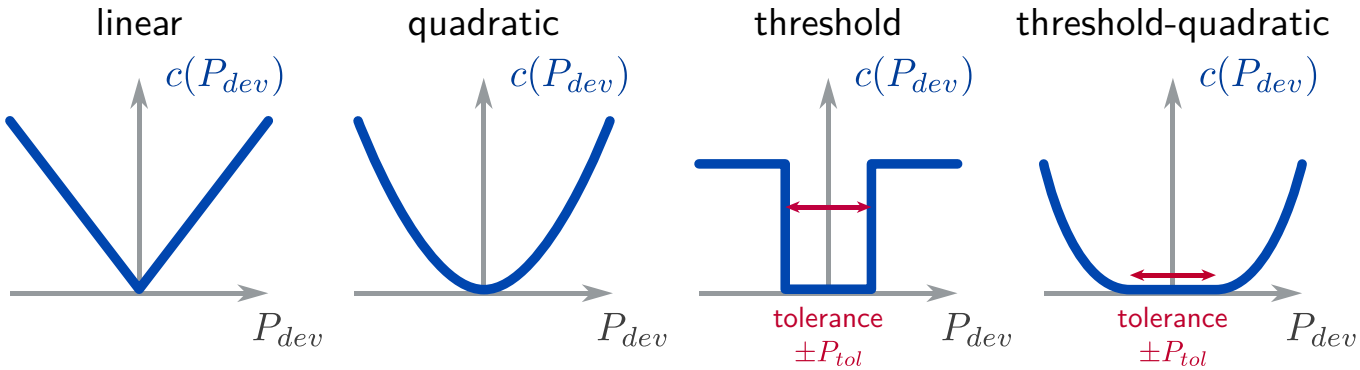
Comme le choix de la fonction coût instantané $g(\dots)$ est libre, on peut faire varier sa forme :



# Une étude : effet de la fonction coût

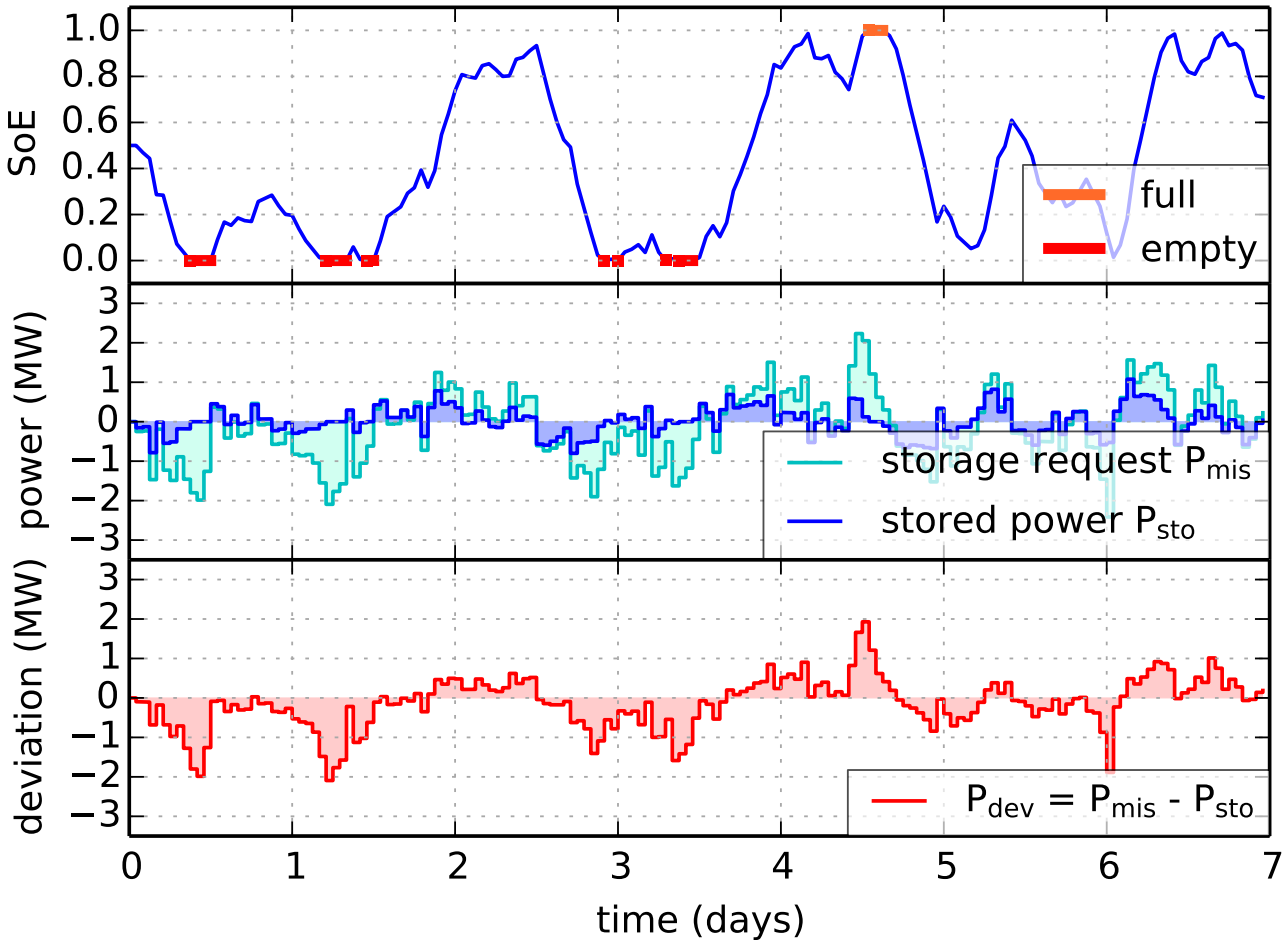
Coût à minimiser : $J = \mathbb{E}\{g(x_k, u_k, w_k)\}$

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



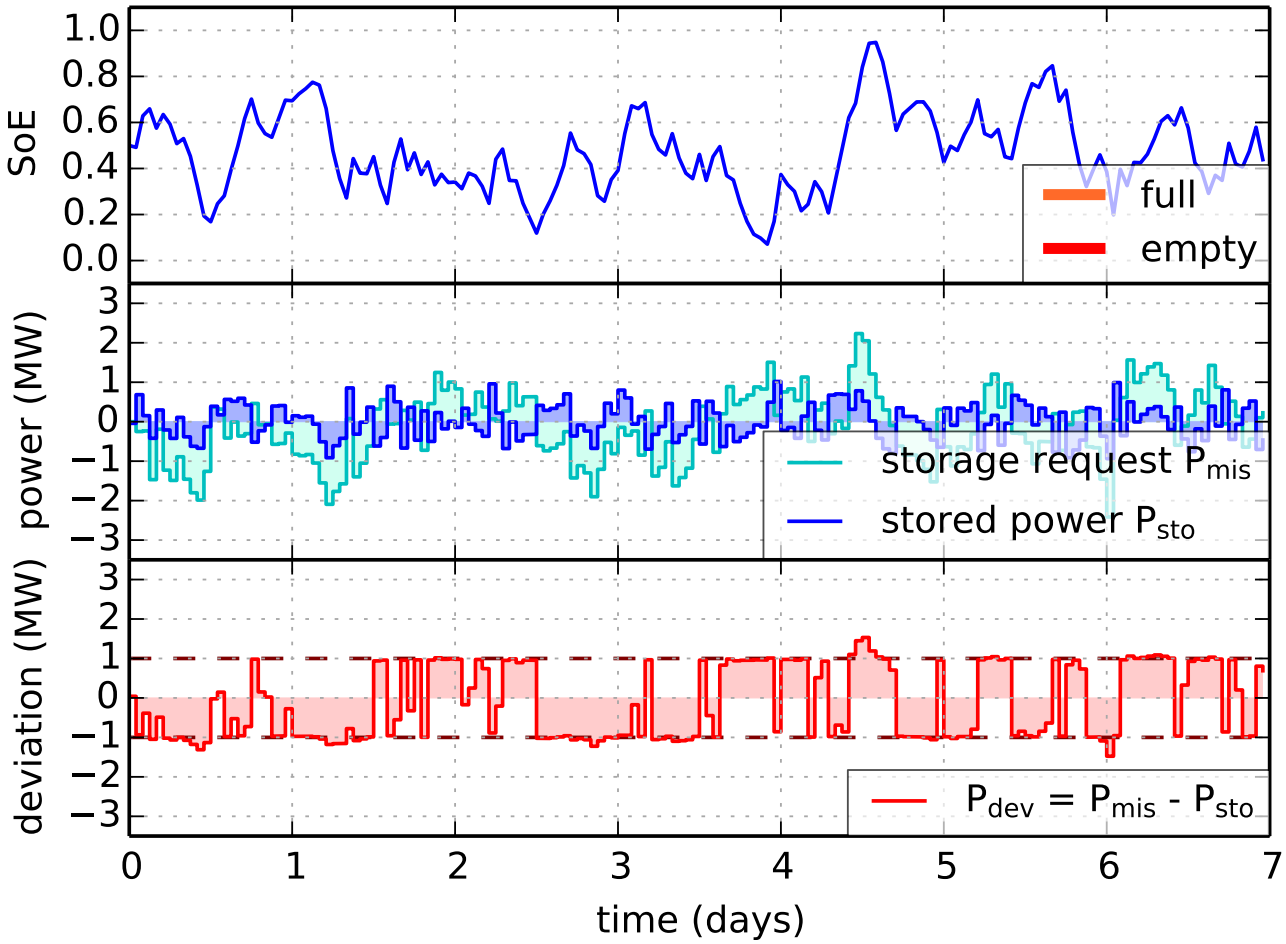

$\rightarrow$ Nous allons observer le résultat de l'optimisation du point de vue des trajectoires temporelles, puis de la loi de gestion.

# Trajectoires pour différentes formes de coûts



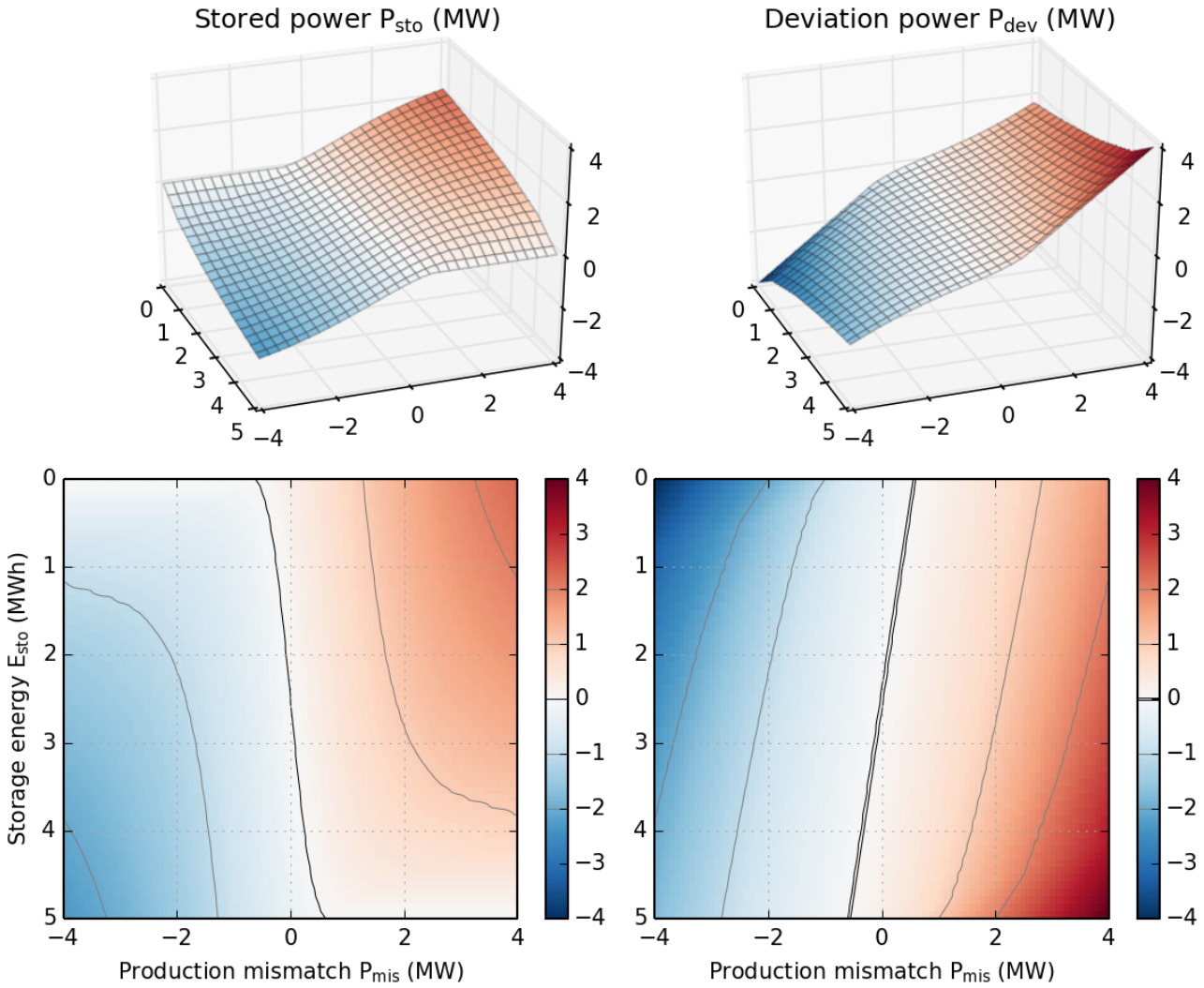

(coût quadratique)

# Trajectoires pour différentes formes de coûts



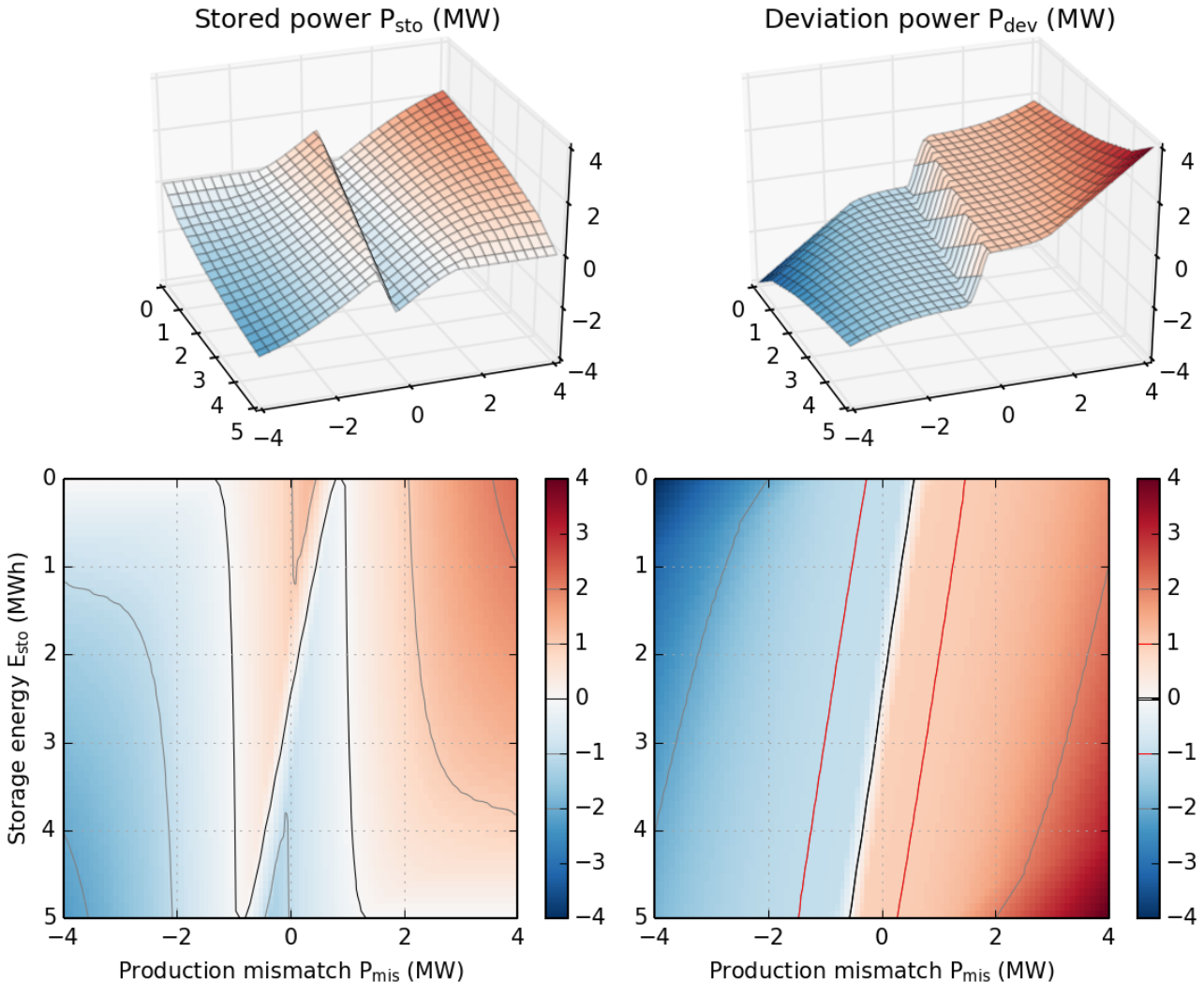

(coût seuil-quadratique à $\pm 1$ MW)

# Loi de gestion pour différentes formes de coûts




(coût quadratique)

# Loi de gestion pour différentes formes de coûts




(coût seuil-quadratique à $\pm 1\,\mathrm{MW}$)

# Effet du choix de la fonction coût

La programmation dynamique stochastique (SDP) permet de traiter une *large palette* de fonctions de pénalisation.

En comparant les résultats d'optimisation nous déduisons :

- la forme des pénalisation (e.g. non-convexité) un fort impact sur le comportement du système

# Effet du choix de la fonction coût

La programmation dynamique stochastique (SDP) permet de traiter une *large palette* de fonctions de pénalisation.

En comparant les résultats d'optimisation nous déduisons :

- la forme des pénalisation (e.g. non-convexité) un fort impact sur le comportement du système
- conséquence pratique : le règlement qui fixe les pénalités doit être rédigé avec soin pour éviter les stratégies "pirates" et encourager les comportements "grid-friendly".

# Plan de la présentation

# Flexibilité 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.

# Flexibilité 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, elle souffre de la *malédiction de la dimension* : vecteur d'état limité à la dimension 3 ou 4 !

# Flexibilité 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, elle souffre de la *malédiction de la dimension* : vecteur d'état limité à la dimension 3 ou 4 !

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)

# La question du *risque*

La SDP "de base" cherche à minimiser le coût *en espérance*. C'est une façon d'appréhender le risque (variabilité du coût) qui suppose une forme de compensation entre des coûts instantanés forts et faibles (attitude "risk neutral" en finance).

On pourrait chercher d'autres méthodes plus robustes, du type "minimax" où l'on cherche à minimiser le coût maximum, ou bien respecter des *contraintes en probabilité* (cf. en particulier les travaux de l'équipe Michel De Lara au CERMICS sur la *viabilité stochastique*).

# Perspectives pour le dimensionnement

La loi de gestion optimale doit être recalculée pour chaque dimensionnement du stockage (capacité $E_{rated}$), ce qui peut être coûteux en temps de calcul.

Solutions de contournement :

- utiliser une loi de gestion empirique sous-optimale pour le dimensionnement
  ($\rightarrow$ étude de l'interaction dimensionnement-gestion)
- *paramétrisation* de la loi de gestion optimale, d'après une structure obtenue par SDP.

# Publications associées à cette présentation

Gestion optimale de stockage appliquée à un houlogénérateur

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

(avec présentation de l'implémentation logicielle `StoDynProg` )

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", Wind Energy 2014 (online)