An Occam's razor paradigm for the control of complex systems

Lionel Mathelin

LIMSI - CNRS

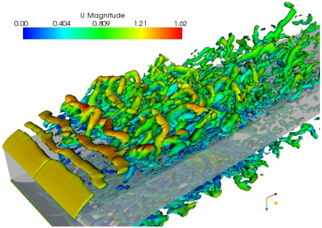
Joint work with Kévin Kasper & Hisham Abou-Kandil
(SATIE - ENS Cachan)

GDR CDD / MOSAR – Nov. 2014

Control of complex systems – Context

Some issues...

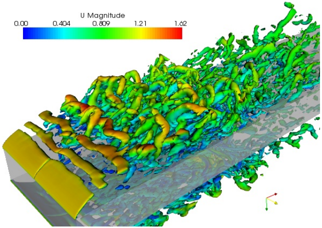
- more and more complex systems. Example of fluid flows: $> 10^7$ spatial DOFs, $> 10^5$ temporal DOFs
  - 3-D unsteady and turbulent,
  - high Reynolds number flows,
  - coupled physics (*e.g.*, fluid / structure / thermal).
- strict specifications $\rightarrow$ calls for efficient control algorithms and not just crude ones,
- real-time control...



Control of complex systems – Context

Some issues...

- more and more complex systems. Example of fluid flows: $> 10^7$ spatial DOFs, $> 10^5$ temporal DOFs
  - 3-D unsteady and turbulent,
  - high Reynolds number flows,
  - coupled physics (*e.g.*, fluid / structure / thermal).
- strict specifications $\rightarrow$ calls for efficient control algorithms and not just crude ones,
- real-time control...



Several non-cooperative players:

- Efficiency $\rightarrow$ should rely on some optimality principles (costly !),
- Real-time $\rightarrow$ should rely on a very simple model (high Reynolds number flows require high actuation frequency, $> kHz$).

Control of fluid flows (cont'd)

$\longrightarrow$ closed-loop control

One hence needs

- estimate the state of the system (filter),
- a low complexity, yet accurate, framework for (nonlinear) control.

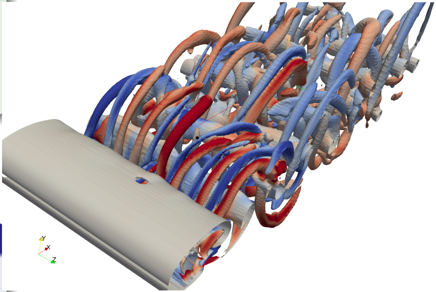
Outline

Outline

Context

Context

High-dimensional spatial field.
Limited number of sensors located at the surface of a bluff body.
Real-time (closed-loop) control.



Parameters

- **Number** of Sensors
- Sensor **locations**
- Reconstruction **method**

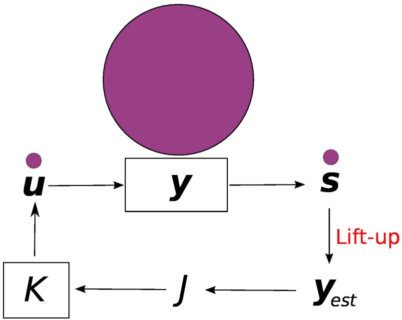
Formulation

Goal
Control a functional of the state of the system from a few wall-mounted sensors.

Formulation

Goal
Control a functional of the state of the system from a few wall-mounted sensors.

$\longrightarrow$ Severely ill-posed problem



Standard approach — POD-based state estimation

Derived a reduced-order model: $\boldsymbol{y} \approx \widehat{\boldsymbol{y}} = D_{\mathrm{POD}}\, \boldsymbol{x}$.
$\left\{ D_{\mathrm{POD}}^{(i)} \right\}_{i=1}^{n_D}$ typically are POD modes (topos). $\boldsymbol{x}$ is the associated coefficients vector.
From a "learning" (unsorted) sequence $Y = \left(\boldsymbol{y}^{(1)} \ldots \boldsymbol{y}^{(n_{\mathrm{snap}})} \right) \in \mathbb{R}^{n \times n_{\mathrm{snap}}}$:

$$D_{\mathrm{POD}}\, \Sigma\, V^* \overset{\text{thin SVD}}{\approx} Y.$$

For a given number n_D of retained modes, leads to the best approximation in the following sense:

$$\widehat{Y} = D_{\mathrm{POD}}\, X_{est} = \underset{\substack{\widetilde{Y} \in \mathbb{R}^{n \times n_{\mathrm{snap}}} \\ \mathrm{rank}\left[\widetilde{Y}\right] \leq n_D}}{\arg\min} \left\| Y - \widetilde{Y} \right\|_F \qquad \text{with } X_{est} = \Sigma\, V^*.$$

- $D \in \mathbb{R}^{n \times n_D}$ the approximation basis [Dictionary],
- $X_{est} \in \mathbb{R}^{n_D \times n_{\mathrm{snap}}}$ the basis coefficients as estimated from the n_s sensors,
- $\boldsymbol{s} \in \mathbb{R}^{n_s}$, sensor information,
- $C \in \mathbb{R}^{n_s \times n}$, restriction matrix such that $\boldsymbol{s} = C\, \boldsymbol{y}$.

Standard approach — POD-based state estimation

When online, what is measured is $\boldsymbol{s} = C\,\boldsymbol{y} \in \mathbb{R}^{n_s}$ only.

Observer such that

$$\widehat{\boldsymbol{x}} \in \underset{\widetilde{\boldsymbol{x}} \in \mathbb{R}^{n_D}}{\arg\min} \left\| \boldsymbol{s} - C\, D_{\mathrm{POD}}\, \widetilde{\boldsymbol{x}} \right\|_F,$$

or simply $\widehat{\boldsymbol{x}} = \left(C\, D_{\mathrm{POD}}\right)^+\ \boldsymbol{s}$.

- $D \in \mathbb{R}^{n \times n_D}$ the approximation basis [Dictionary],
- $X_{est} \in \mathbb{R}^{n_D \times n_{\mathrm{snap}}}$ the basis coefficients as estimated from the n_s sensors,
- $\boldsymbol{s} \in \mathbb{R}^{n_s}$, sensor information,
- $C \in \mathbb{R}^{n_s \times n}$, restriction matrix such that $\boldsymbol{s} = C\,\boldsymbol{y}$.

Standard approach — POD-based state estimation

When online, what is measured is $\boldsymbol{s} = C\,\boldsymbol{y} \in \mathbb{R}^{n_s}$ only.

Observer such that

$$\widehat{\boldsymbol{x}} \in \underset{\widetilde{\boldsymbol{x}} \in \mathbb{R}^{n_D}}{\arg\min} \left\| \boldsymbol{s} - C\,D_{\mathrm{POD}}\,\widetilde{\boldsymbol{x}} \right\|_F,$$

or simply $\widehat{\boldsymbol{x}} = (C\,D_{\mathrm{POD}})^+\ \boldsymbol{s}$.

The reconstructed field is finally:

$$\widehat{\boldsymbol{y}} = D_{\mathrm{POD}}\,\widehat{\boldsymbol{x}} = D_{\mathrm{POD}}\ (C\,D_{\mathrm{POD}})^+\ \boldsymbol{s}.$$

$\boxed{L_{\mathrm{POD}} := D_{\mathrm{POD}}\ (C\,D_{\mathrm{POD}})^+}$ is the POD-based lift-up operator.

- $D \in \mathbb{R}^{n \times n_D}$ the approximation basis [Dictionary],
- $X_{est} \in \mathbb{R}^{n_D \times n_{\mathrm{snap}}}$ the basis coefficients as estimated from the n_s sensors,
- $\boldsymbol{s} \in \mathbb{R}^{n_s}$, sensor information,
- $C \in \mathbb{R}^{n_s \times n}$, restriction matrix such that $\boldsymbol{s} = C\,\boldsymbol{y}$.

Standard approach — POD-based state estimation

When online, what is measured is $\boldsymbol{s} = C\,\boldsymbol{y} \in \mathbb{R}^{n_s}$ only.

Observer such that

$$\widehat{\boldsymbol{x}} \in \underset{\widetilde{\boldsymbol{x}} \in \mathbb{R}^{n_D}}{\arg\min} \left\| \boldsymbol{s} - C\, D_{\mathrm{POD}}\, \widetilde{\boldsymbol{x}} \right\|_F,$$

or simply $\widehat{\boldsymbol{x}} = \left(C\, D_{\mathrm{POD}}\right)^+ \boldsymbol{s}$. $\longleftarrow$ **requires $n_s \geq n_D$!**

The reconstructed field is finally:

$$\widehat{\boldsymbol{y}} = D_{\mathrm{POD}}\, \widehat{\boldsymbol{x}} = D_{\mathrm{POD}}\, \left(C\, D_{\mathrm{POD}}\right)^+ \boldsymbol{s}.$$

$\boxed{L_{\mathrm{POD}} := D_{\mathrm{POD}}\, \left(C\, D_{\mathrm{POD}}\right)^+}$ is the POD-based lift-up operator.

- $D \in \mathbb{R}^{n \times n_D}$ the approximation basis [Dictionary],
- $X_{est} \in \mathbb{R}^{n_D \times n_{\mathrm{snap}}}$ the basis coefficients as estimated from the n_s sensors,
- $\boldsymbol{s} \in \mathbb{R}^{n_s}$, sensor information,
- $C \in \mathbb{R}^{n_s \times n}$, restriction matrix such that $\boldsymbol{s} = C\,\boldsymbol{y}$.

Formulation

$\longrightarrow$ reformulate in a framework amenable to estimation from a few sensors,
$\longrightarrow$ basis learning approach.

Formulation

$$\text{Find } \{D, X_{est}, C\} \in \underset{\widetilde{D},\widetilde{C},\widetilde{X}_{est}}{\arg\min} \left\| Y - \widehat{Y}\left(\widetilde{D}, \widetilde{C}, \widetilde{X}_{est}\right) \right\|_F$$

A dictionary learning algorithm

⟶ derive an over-complete dictionary for sparse representation:

$$\text{Find } \{D, X\} \in \underset{\widetilde{D},\widetilde{X}}{\arg\min} \left\| Y - \widetilde{D}\,\widetilde{X} \right\|_F \quad \text{s.t.} \quad \left\| \widetilde{\mathbf{x}}^{(i)} \right\|_0 \leq n_S, \quad \forall i,$$
$$X = \left(\boldsymbol{x}^{(1)} \ldots \boldsymbol{x}^{(n_{\text{snap}})} \right).$$

A dictionary learning algorithm

$\longrightarrow$ derive an over-complete dictionary for sparse representation:

$$\text{Find } \{D, X\} \in \underset{\widetilde{D},\widetilde{X}}{\arg\min} \left\| Y - \widetilde{D}\,\widetilde{X} \right\|_F \quad \text{s.t.} \quad \left\| \widetilde{\mathbf{x}}^{(i)} \right\|_0 \leq n_S, \quad \forall i,$$
$$X = \left(\boldsymbol{x}^{(1)} \ldots \boldsymbol{x}^{(n_{\text{snap}})} \right).$$

Use the K-SVD algorithm
D, n_S, n_D (number of K-SVD modes)

Repeat

- **Sparse Coding** : $X = \underset{\widetilde{X}}{\arg\min} \left\| Y - D\,\widetilde{X} \right\|_F \quad \text{s.t.} \quad \left\| \mathbf{x}^{(i)} \right\|_0 \leq n_S, \forall i.$
- **CodeBook Update** : Update D and X in order to lower $\| Y - D\,X \|_F$ while maintaining the support of $\left\{ \boldsymbol{x}^{(i)} \right\}_i$.

A dictionary learning algorithm

$\longrightarrow$ derive an over-complete dictionary for sparse representation:

$$\text{Find } \{D, X\} \in \underset{\widetilde{D},\widetilde{X}}{\arg\min} \left\| Y - \widetilde{D}\,\widetilde{X} \right\|_F \quad \text{s.t.} \quad \left\| \widetilde{\mathbf{x}}^{(i)} \right\|_0 \leq n_S, \quad \forall i,$$

$$X = \left(\boldsymbol{x}^{(1)} \dots \boldsymbol{x}^{(n_{\text{snap}})} \right).$$

Use the K-SVD algorithm
D, n_S, n_D (number of K-SVD modes)

Repeat

- **Sparse Coding** : $X = \underset{\widetilde{X}}{\arg\min} \left\| Y - D\,\widetilde{X} \right\|_F \quad \text{s.t.} \quad \left\| \mathbf{x}^{(i)} \right\|_0 \leq n_S, \forall i.$
- **CodeBook Update** : Update D and X in order to lower $\|Y - D\,X\|_F$ while maintaining the support of $\left\{ \boldsymbol{x}^{(i)} \right\}_i$.

But typically $\boldsymbol{x}^{(i)} \in \mathbb{R}^{n_D}$ **cannot** be estimated from measurements.
$\longrightarrow$ determine D given a set of sensors for estimating $\boldsymbol{x}^{(i)}$ from $\boldsymbol{s}^{(i)}$ instead of $\boldsymbol{y}^{(i)}$.

$\longrightarrow$ Observability-oriented K-SVD: K-SVD*C*

Iteratively solve for

- **Sparse coding**

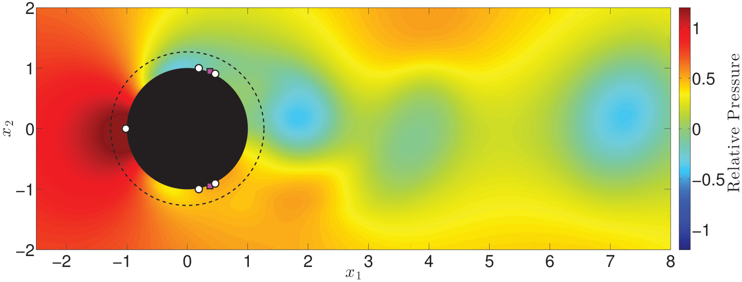
$$X_{est} = \underset{\widetilde{X}}{\arg\min} \left\| \boldsymbol{s} - C\, D\, \widetilde{X} \right\|_F \quad \text{s.t.} \quad \left\| \widetilde{\boldsymbol{x}}^{(i)} \right\|_0 \leq n_S, \quad \forall i.$$

- **CodeBook Update**
  Determine D and X in order to lower the learning error while maintaining the support of $\left\{ \boldsymbol{x}^{(i)} \right\}_i$:

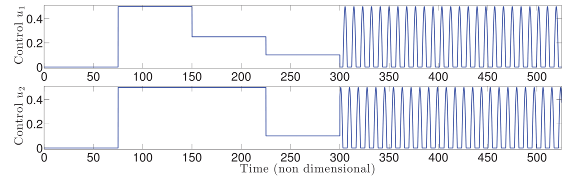
$$\{D, X\} = \underset{\widetilde{D}, \widetilde{X}}{\arg\min} \left\| Y - \widetilde{D}\ \widetilde{X} \right\|_F \quad \text{s.t.} \quad \text{supp}_\varepsilon \left(\widetilde{X} \right) = \text{supp}_\varepsilon \left(X_{est} \right).$$

$\longrightarrow$ consistent and as realistic as possible

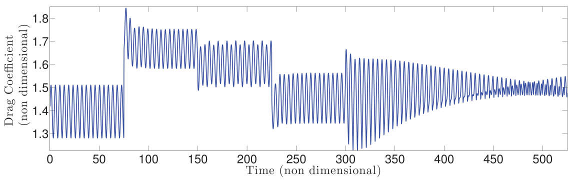
Test System : Flow around a circular cylinder



Command - Learning Sequence



Drag - Learning Sequence



The learning snapshots sequence must contain information on the dynamics that we want to reconstruct (some *a priori* knowledge is required).

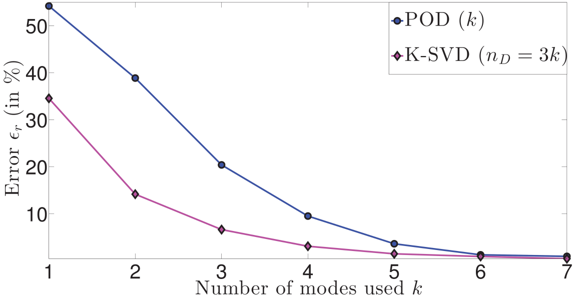
Algorithm

1. Basis learning (K-SVD$\boldsymbol{C}$) **[offline]**
   - Form a snapshot matrix Y of the QoI.
   - Use K-SVDC to obtain the dictionary D with the sensor matrix C.

2. Field reconstruction **[online]**
   - Use sparse recovery with the measure $\boldsymbol{s}$ to obtain $\widehat{\boldsymbol{x}}$.
   - Reconstruct the total field from $\widehat{\boldsymbol{y}} = D\widehat{\boldsymbol{x}}$.

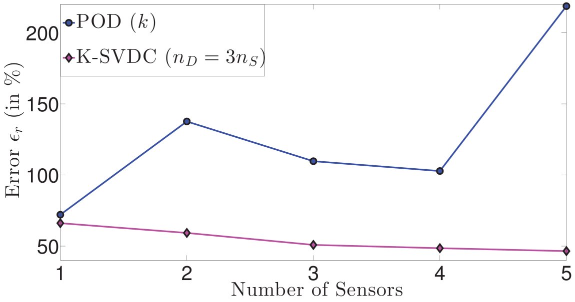
Online (real time) field estimation [filtering] is possible since sparse recovery algorithms are fast (recursive compressed sensing).

K-SVD*C* performance – Full information: $C \equiv I_n$



$\longrightarrow$ The K-SVD algorithm provides significantly better L^2-reconstruction performance than POD despite the Eckart-Young theorem.

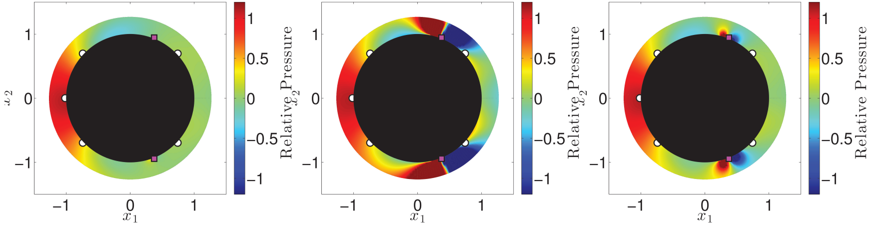
K-SVD***C*** performance – Sensor information only



⟶ The K-SVD***C*** algorithm provides significantly better L^2-reconstruction performance than POD.

⟶ POD coefficients are not accurately estimated from the sensors.

Reconstruction



Relative pressure field. Exact (left), estimated from POD (middle), estimated from K-SVD$\boldsymbol{C}$ (right).

$$n_S = 5 \longrightarrow n = 3,000+$$

$\longrightarrow$ Much better reconstruction performance with K-SVD$\boldsymbol{C}$.

Outline

Choosing the sensors location

The sensor location is of pivotal importance for estimation performance.

The best sensor location is the solution of a combinatorial (NP-hard) problem.

$\longrightarrow$ Heuristics and greedy algorithms

Choosing the sensors location

The sensor placement problem is formalized as

$$\text{Find } D, C \in \underset{\widetilde{D},\widetilde{C}\in\mathcal{M}_C}{\arg\min} \left\| Y - D\,\widehat{X}_{SR}\left(\widetilde{C}, Y\right)\right\|_F \quad \text{with } \widehat{X}_{SR} \leftarrow \texttt{Sparse Recovery}\left(\widetilde{C}, Y\right)$$

For a given D and, say, OMP for the sparse recovery, no closed form solution for C.

Choosing the sensors location

The sensor placement problem is formalized as

$$\text{Find } D, C \in \underset{\widetilde{D}, \widetilde{C} \in \mathcal{M}_C}{\arg\min} \left\| Y - D \widehat{X}_{SR} \left(\widetilde{C}, Y \right) \right\|_F \quad \text{with } \widehat{X}_{SR} \leftarrow \texttt{Sparse Recovery} \left(\widetilde{C}, Y \right)$$

For a given D and, say, OMP for the sparse recovery, no closed form solution for C.

$\longrightarrow$ Relax the recovery class $\longrightarrow$ sensor space.

$$\text{Find } \{C, L\} = \underset{\widetilde{C}, \widetilde{L}}{\arg\min} \left\| Y - \widetilde{L} \, \widetilde{C} \, Y \right\|_F .$$

$L : \mathbb{R}^{n_s} \rightarrow \mathbb{R}^{n_y}$ is the linear lift-up operator from the measurements to the field estimation. For a given C, it yields $L = Y \, (C \, Y)^+$ and C now solves:

$$\begin{aligned} C \quad & \in \underset{\widetilde{C} \in \mathcal{M}_C}{\arg\min} \left\| Y - Y \left(\widetilde{C} \, Y \right)^+ \widetilde{C} \, Y \right\|_F, \\ \text{or, equivalently} \quad & \in \underset{\widetilde{C} \in \mathcal{M}_C}{\arg\min} \left\| R - R \left(\widetilde{C} \, Y \right)^+ \widetilde{C} \, Y \right\|_F, \quad \text{with } Q \, R \overset{\text{thin QR}}{\approx} Y \quad \text{[faster]} \end{aligned}$$

$\longrightarrow$ Constrained optimization problem solved with a greedy technique.

Finally: sensor-based estimation [offline step]

Iteratively solve for

- **Sensor-based Sparse Coding**

$$X_{est} = \underset{\widetilde{X}}{\arg\min} \left\| C\, Y - D\, \widetilde{X} \right\|_F \quad \text{s.t.} \quad \left\| \widetilde{\boldsymbol{x}}^{(i)} \right\|_0 \leq n_S, \quad \forall i.$$

- **CodeBook Update**
  Determine D in order to lower the learning error in $C\,Y$:

$$D = \underset{\widetilde{D}}{\arg\min} \left\| C\, Y - \widetilde{D}\ X_{est} \right\|_F .$$

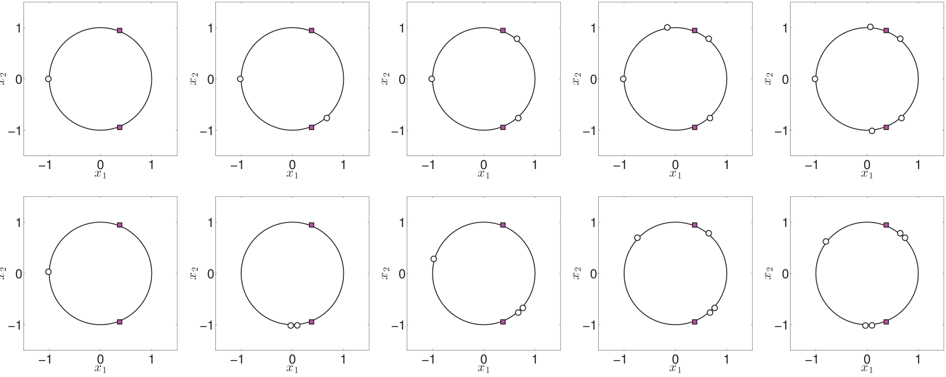
- **Goal-oriented CodeBook Update**

$$\{D_{QoI}, X_{QoI}\} = \underset{\widetilde{D}_{QoI}, \widetilde{X}_{QoI}}{\arg\min} \left\| QoI\,(Y) - \widetilde{D}_{QoI}\, \widetilde{X}_{QoI} \right\|_F \quad \text{s.t.} \quad \text{supp}_\varepsilon \left(\widetilde{X}_{QoI} \right) = \text{supp}_\varepsilon \left(X_{est} \right).$$

- **Sensor learning**
  Update sensors position:

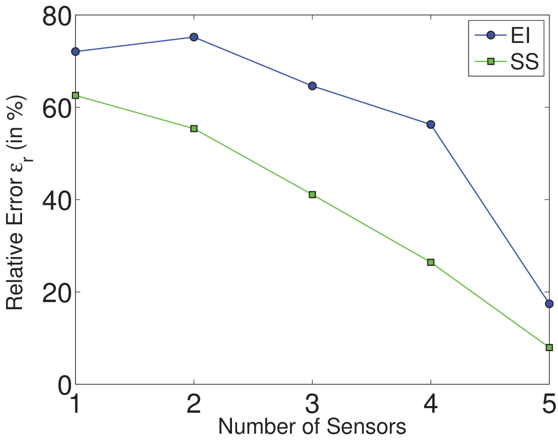
$$C = \underset{\widetilde{C} \in \mathcal{M}_C}{\arg\min} \left\| R - R \left(\widetilde{C}\, Y \right)^{+} \widetilde{C}\, Y \right\|_F \quad \text{s.t.} \quad \left\| \widetilde{\mathbf{c}}_i \right\|_0 = 1, \quad 1 \leq i \leq n_s, \quad Q\, R \overset{\text{thin QR}}{\approx} Y.$$

Effective Independence *vs* Sensor Space



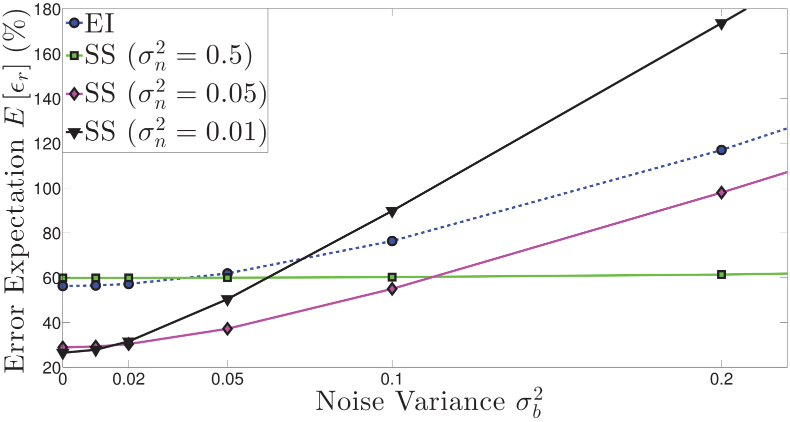
Effective Independence (top row) vs Sensor Space (bottom row).

Sensor Space – Noiseless environment



⟶ Significantly better performance than Effective Independence (EI, based on Fisher Information Matrix).

Sensor Space – Noisy environment – $n_S = 4$



$\longrightarrow$ With a suitably chosen robustness target σ_n, the present SS method allows better performance than EI in both noiseless and noisy environments.

Closing remarks

- Estimation/observation is a critical ingredient for the control of complex systems,
- Offline/Online strategy for deriving an observer. First learn about the system at hand, then exploit,
- Sparsity-exploitation / dictionary learning technique is one of the pivotal tools for a *realistic* and successful approach.

Wrapping-up

- Sparsity must be exploited whenever possible!
- Many (Most?) physics-related signals are compressible in standard functional bases,
- The interaction with the signal at hand is often very limited $\longrightarrow$ *strong restrictions on the sensing operator!*
- Basis learning philosophy is the key to achieve realistic and efficient applications.