

# UNIVERSITÉ LILLE 1

Doctoral School SPI 072

University Department LML & CRISTAL

Thesis defended by **Linh Van NGUYEN**

on **28<sup>th</sup> September, 2016**

In order to become Doctor from Université Lille 1

Academic Field **Mechanics**

Speciality **Turbulence**

## **Reconstruction of finely resolved velocity fields in turbulent flows from low resolution measurements**

**Thesis supervised by** Jean-Philippe LAVAL  
Pierre CHAINAIS

### **Committee members**

<i>Referees</i>	Olivier MICHEL	Professor at Grenoble-INP
	Etienne MÉMIN	Senior Researcher at INRIA
<i>Examiners</i>	Luminita DANAILA	Professor at Université de Rouen
	Nicolas DOBIGEON	Professor at INP-ENSEEIH Toulouse
<i>Supervisors</i>	Jean-Philippe LAVAL	Researcher HDR at CNRS
	Pierre CHAINAIS	Associate Professor HDR at École Centrale de Lille



# UNIVERSITÉ LILLE 1

Doctoral School SPI 072

University Department LML & CRISTAL

Thesis defended by **Linh Van NGUYEN**

on **28<sup>th</sup> September, 2016**

In order to become Doctor from Université Lille 1

Academic Field **Mechanics**

Speciality **Turbulence**

## Reconstruction of finely resolved velocity fields in turbulent flows from low resolution measurements

**Thesis supervised by** Jean-Philippe LAVAL  
Pierre CHAINAIS

### Committee members

<i>Referees</i>	Olivier MICHEL	Professor at Grenoble-INP
	Etienne MÉMIN	Senior Researcher at INRIA
<i>Examiners</i>	Luminita DANAILA	Professor at Université de Rouen
	Nicolas DOBIGEON	Professor at INP-ENSEEIH Toulouse
<i>Supervisors</i>	Jean-Philippe LAVAL	Researcher HDR at CNRS
	Pierre CHAINAIS	Associate Professor HDR at École Centrale de Lille



# UNIVERSITÉ LILLE 1

École doctorale SPI 072

Unité de recherche LML & CRISTAL

Thèse présentée par **Linh Van NGUYEN**

le **28 septembre 2016**

En vue de l'obtention du grade de docteur de l'Université Lille 1

Discipline **Mécanique**

Spécialité **Turbulence**

## Reconstruction fine de champs de vitesses d'un écoulement turbulent à partir de mesures faiblement résolues

**Thèse dirigée par** Jean-Philippe LAVAL  
Pierre CHAINAIS

### Composition du jury

<i>Rapporteurs</i>	Olivier MICHEL	Professeur au Grenoble-INP
	Etienne MÉMIN	Directeur de recherche à l'INRIA
<i>Examineurs</i>	Luminita DANAILA	Professeur à l'Université de Rouen
	Nicolas DOBIGEON	Professeur à l'INP-ENSEEIH Toulouse
<i>Directeurs de thèse</i>	Jean-Philippe LAVAL	Chargé de Recherche HDR au CNRS
	Pierre CHAINAIS	Maître de Conférences HDR à l'École Centrale de Lille



This thesis has been prepared at the following research units.

### **LML**

Laboratoire de Mécanique de Lille  
FRE 3723  
Bvd Paul Langevin  
59655 Villeneuve d'Ascq,CEDEX  
France



(33)(0) 3 20 33 71 52



(33)(0) 3 20 33 71 53



[lml@univ-lille1.fr](mailto:lml@univ-lille1.fr)

Web Site <http://lml.univ-lille1.fr/lml/>



Laboratoire  
Mécanique  
Lille

### **CRIStAL**

Centre de Recherche en Informatique,  
Signal et Automatique de Lille  
UMR 9189 CNRS  
CS 20048  
59651 Villeneuve d'Ascq CEDEX  
France

Web Site <http://cristal.univ-lille.fr/>



CRIStAL  
Centre de Recherche en Informatique,  
Signal et Automatique de Lille





---

**RECONSTRUCTION OF FINELY RESOLVED VELOCITY FIELDS IN TURBULENT FLOWS FROM LOW RESOLUTION MEASUREMENTS**

This work lies in between the research domains of turbulence and image processing. The main objectives are to propose new methodologies to reconstruct small-scale turbulence from measurements at large-scale only. One contribution of this work is a review of existing methods. We also propose new models inspired from recent works in image processing to adapt them to the context of turbulence. We address two different problems. The first problem is to find an empirical mapping function between large and small scales for which regression models are a common approach. We also introduce the use of “dictionary learning” to train coupled representations of large and small scales for reconstruction. The second problem is to reconstruct small-scale information via the fusion of complementary measurements. The non-local means propagation model exploits the similarity of structures in the flow, while the Bayesian fusion model estimates the most probable fields given the measurements thanks to a maximum a posteriori estimate. All methods are validated and analyzed using numerical databases where fully resolved velocity fields are available. Performances of the proposed approaches are also characterized for various configurations. These results can be considered under the co-conception design framework where different experimental setups are designed to maximize the level of useful information after post-processing.

**Keywords:** reconstruction, turbulence, dictionary learning, non-local means, bayesian fusion, regression, small scales

---

**RECONSTRUCTION FINE DE CHAMPS DE VITESSES D'UN ÉCOULEMENT TURBULENT À PARTIR DE MESURES FAIBLEMENTS RÉVOLUES**

Ce travail est à la jonction de deux domaines de recherche que sont la turbulence et le traitement d'image. L'objectif principal est de proposer de nouvelles méthodologies pour reconstruire les petites échelles de la turbulence à partir de mesures grande échelle. Ce travail revisite des méthodes conventionnelles et propose de nouveaux modèles basés sur les travaux récents en traitement d'image pour les adapter à une problématique de turbulence. Le premier problème consiste à trouver une fonction de correspondance empirique entre les grandes et les petites échelles, une approche classique pour les modèles de type regression. Nous introduisons également une méthode appelée “apprentissage de dictionnaire” pour laquelle une représentation couplée des grandes et des petites échelles est déduite par apprentissage statistique. Le deuxième problème est de reconstruire les informations à petites échelles par fusion de plusieurs mesures complémentaires. Le modèle de type “propagation de la moyenne non-locale” exploite la similarité des structures de l'écoulement alors que les modèles bayésiens de fusion proposent d'estimer le champ le plus probable en fonction d'informations données, on parle d'estimateur maximum a posteriori. Les performances des différentes approches sont validées et analysées sur des bases de données numériques pour lesquelles les informations sont disponibles à toutes les échelles. Ces résultats peuvent être utilisés dans une approche de type co-conception où il s'agit d'imaginer différents dispositifs expérimentaux définis conjointement avec les traitements numériques pour maximiser la qualité des informations obtenues après traitement.

**Mots clés :** reconstruction, turbulence, apprentissage de dictionnaire, moyenne non-locale, fusion bayésienne, régression, petites échelles

---



# Acknowledgements

First and foremost I would like to express my special appreciation and thanks to my advisors, Dr. Jean-Philippe LAVAL and Dr. Pierre CHAINAIS. You have been wonderful mentors, who are always there to encourage my work and guide me through when things do not work well. Your advices on both research and life have been priceless. I have learned from you tremendously, especially the critical and analytical mindset.

I would also like to thank my committee members, Prof. Olivier MICHEL, Dr. Etienne Mémin, Prof. Nicolas DOBIGEON and Prof. Luminita DANAILA for serving as my committee members even at hardship. I thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions.

I would also like to all my colleagues, Ilkay, Sri, Raoul, Fabio, Cédric, Christophe, Robin, Tai, Vinshai, Phuong and Clément for your amazing friendship during my three years. I have learned so much from you, both in research and daily life. Your accompany really made my time in Lille enjoyable and memorable.

At the end, I would like to expresse my greatest appreciation to my family. I would like to thank my wife who is always there to share all happy and tough moments, and my parents who always encourage me to learn and experience new things. Without you this work would be really impossible. Thank you.



# Abbreviations

CV	<b>Cross Validation</b>
DCT	<b>Discrete Cosine Transform</b>
DL	<b>Dictionary Learning</b>
DNS	<b>Direct Numerical Simulation</b>
HF	<b>High Frequency</b>
HR	<b>High Resolution</b>
HTHS	<b>High-Time-High-Space resolution</b>
HTLS	<b>High-Time-Low-Space resolution</b>
HWA	<b>Hot Wire Anemometer</b>
KRR	<b>Kernel Ridge Regression</b>
LARS	<b>Least Angle Regression</b>
LASSO	<b>Least Absolute Shrinkage and Selection Operator</b>
LF	<b>Low Frequency</b>
LOOCV	<b>Leave One Out Cross Validation</b>
LR	<b>Low Resolution</b>
LSE	<b>Linear Stochastic Estimation or Least Square Estimation</b>
LTHS	<b>Low-Time-High-Space resolution</b>
MAP	<b>Maximum A Posteriori</b>
MLE	<b>Maximum Likelihood Estimator</b>
MSE	<b>Mean Square Error</b>
NLM	<b>Non-Local Means</b>
NRMSE	<b>Normalized Root Mean Square Error</b>
N-S	<b>Navier-Stokes</b>
ODL	<b>Online Dictionary Learning</b>

PCA	<b>P</b> rincipal <b>C</b> omponent <b>A</b> nalysis
PIV	<b>P</b> article <b>I</b> mage <b>V</b> elocimetry
POD	<b>P</b> roper <b>O</b> rthogonal <b>D</b> ecomposition
Re	<b>R</b> eynolds number
RMSE	<b>R</b> oot <b>M</b> ean <b>S</b> quare <b>E</b> rror
RR	<b>R</b> idge <b>R</b> egression
SPIV	<b>S</b> tereo <b>P</b> article <b>I</b> mage <b>V</b> elocimetry
SR	<b>S</b> uper- <b>R</b> esolution
SR1	<b>S</b> uper- <b>R</b> esolution: coupling low and high resolution patches
SR2	<b>S</b> uper- <b>R</b> esolution: coupling interpolated low and high resolution patches
SR3	<b>S</b> uper- <b>R</b> esolution: coupling derivatives with residuals patches
STB	<b>S</b> hake- <b>T</b> he- <b>B</b> ox PIV calibration
SVD	<b>S</b> ingular <b>V</b> alue <b>D</b> ecomposition

# Latin symbols

$H$	half the channel size (for channel flow)
$\tau$	(local coordinate in) time
$\alpha$	(local coordinate in) vertical direction
$\beta$	(local coordinate in) spanwise direction
$t$	(coordinate in) time
$x$	(coordinate in) streamwise direction
$y$	(coordinate in) vertical direction
$z$	(coordinate in) spanwise direction
$D_f$	dimension of the feature vectors
$\Delta T$	the size of the element block in time direction
$\Delta y$	the size of the element block in vertical direction
$\Delta z$	the size of the element block in spanwise direction
$P$	spatial dimension of one high resolution field
$Q$	spatial dimension of one low resolution field
$N$	total number of snapshots at high temporal resolution
$M$	total number of snapshots at low temporal resolution
$(.)^e$	superscript for samples outside the training data
$n_t$	a white Gaussian noise term at time $t$
$x$	velocity vector at LTHS resolution of size $PM \times 1$
$y$	velocity vector at HTLS resolution of size $QN \times 1$
$z$	velocity vector at HTHS resolution of size $PN \times 1$
$\mathbf{X}$	velocity at LTHS resolution of size $M \times P$
$\mathbf{Y}$	velocity at HTLS resolution of size $N \times Q$
$\mathbf{Z}$	velocity at HTHS resolution of size $N \times P$
$x_t$	a velocity snapshot of $\mathbf{X}$ at time $t$ of size $P \times 1$
$y_t$	a velocity snapshot of $\mathbf{Y}$ at time $t$ of size $Q \times 1$
$z_t$	a velocity snapshot of $\mathbf{Z}$ at time $t$ of size $P \times 1$
$\mathcal{R}_a^k$	operator to extract a patch centered at $k$ -th pixel to accumulate the estimate (non-local means filter)

$\mathcal{R}_h^k$	operator to extract a high-resolution patch centered at $k$ -th pixel
$\mathcal{R}_\ell^k$	operator to extract a low-resolution patch centered at $k$ -th pixel
$\mathcal{R}_s^k$	operator to extract a patch centered at $k$ -th pixel to estimate the similarity (non-local means filter)
$\mathbb{I}_s$	2D cubic spline interpolator in space
$\mathbb{I}_t$	1D cubic spline interpolator in time
$\mathbb{L}_s^{loc}$	low-pass filter operator on local patches
$\mathbb{L}_s$	2D low-pass filter, either ideal Fourier or 5th-order least-square spline
$\mathbb{L}_t$	1D low-pass filter, either ideal Fourier or 5th-order least-square spline
$\mathbb{S}_s^{loc}$	subsampling operator on local patches
$\mathbb{S}_s$	subsampling in space from $P$ to $Q$ snapshots, $\mathbb{S}_s \mathbf{z} = \mathbf{y}$
$\mathbb{S}_t$	subsampling in time from $N$ to $M$ snapshots, $\mathbb{S}_t \mathbf{z} = \mathbf{x}$
$\Delta\kappa$	loss of kinetic energy due to the subsampling or downsampling
$\epsilon$	normalized root mean square error
$\gamma$	KRR kernel parameter
$\lambda$	regularization parameter
$\mathcal{L}(f)$	a loss function of $f$ to minimize
$\mathbf{d}_i$	the $i$ -th atom of dictionary $\mathbf{D}$
$\mathbf{a}_i$	the sparse coefficient corresponding to $\mathbf{d}_i$
$\mathbf{p}_h^k$	a high-resolution patch centered at $k$ -th pixel
$\mathbf{p}_\ell^k$	a low-resolution patch centered at $k$ -th pixel
$\mathbf{D}$	A dictionary
$\mathbf{A}$	a sparse matrix to represent data via a dictionary
$p$	dimension of a high-resolution patch
$q$	dimension of a low-resolution patch
$L$	number of nonzero coefficients in the sparse coefficient vector $\mathbf{a}_i$ used to reconstruct a vector via a dictionary $\mathbf{D}$ , $L = \ \mathbf{a}_i\ _0$
$K$	number of atoms in a dictionary
$m$	number of extracted patches to train the dictionary
$\mathbf{P}_h$	a matrix of high-resolution patches
$\mathbf{P}_\ell$	a matrix of low-resolution patches



---

$r$	size of accumulation patches in NLM-based propagation model
$\mathbf{b}_{t_o}$	small scales at $t_o$ -th key frame to propagate via NLM model
$\mathcal{N}_k$	set of neighboring pixels of $k$ -th pixel in a 2D
$\mathcal{N}_{t_o}$	set of neighboring snapshots in time of the $t_o$ -th one
$s$	size of searching region in NLM-based propagation model
$\sigma$	NLM global filter parameter
$w[k, i, t]$	weight (intepreted as the probability) of $k$ -th pixel of $\hat{\mathbf{h}}_t$ coming from the $i$ -th pixel of $\mathbf{b}_{t_o}$
$\mathbf{h}_s$	conjugate part of the term $\mathbb{I}_s \mathbf{y}$ in space
$\mathbf{h}_t$	small scales at time $t$ to estimate (chapter 4) or conjugate part of the term $\mathbb{I}_t \mathbf{x}$ in time (chapter 5)
$\lambda_1$	regularization parameter in the learning stage
$\lambda_2$	regularization parameter in the reconstruction stage
$\Sigma$	covariance matrix
$\sigma$	vector form of the covariance matrix $\Sigma$



# Math symbols

$\operatorname{argmax}_x(f(x))$	argument of the maximum, which is $x$ for which the function $f(x)$ attains its maximum
$\operatorname{argmin}_x(f(x))$	argument of the minimum, which is $x$ for which the function $f(x)$ attains its minimum
$ \cdot $	matrix determinant
$\ \mathbf{u}\ _2^2$	square of a Euclidean distance: $\ \mathbf{u}\ _2^2 = \mathbf{u}^\top \mathbf{u}$
$\ \mathbf{u}\ _\Sigma^2$	square of a Mahalanobis distance: $\ \mathbf{u}\ _\Sigma^2 = \mathbf{u}^\top \Sigma^{-1} \mathbf{u}$
$\ \mathbf{u}\ _p$	$L^p$ norm, defined as $\left(\sum_i  \mathbf{u}_i ^p\right)^{1/p}$
$p(\mathbf{u})$	Probability of a variable $\mathbf{u}$
$p(\mathbf{u} \mathbf{v})$	Conditional probability of $\mathbf{u}$ given $\mathbf{v}$
$E[\cdot]$	expectation value
$\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}_u, \Sigma_u)$	Gaussian distribution of variable $\mathbf{u}$ that takes the mean $\boldsymbol{\mu}_u$ and fluctuates due to covariance $\Sigma_u$
$\mathcal{N}(\mathbf{u} \mathbf{v})$	distribution of $\mathbf{u}$ conditioning on $\mathbf{v}$
$\mathbf{u}^\dagger$	pseudo-inverse of $\mathbf{u}$ , defined as $\mathbf{u}^\top (\mathbf{u} \mathbf{u}^\top)^{-1}$
$\mathcal{O}(d)$	Time complexity as the order of $d$
$\mathbf{u}^\top$	transpose of vector $\mathbf{u}$



# Table of contents

<b>Abstract</b>	<b>ix</b>
<b>Acknowledgements</b>	<b>xi</b>
<b>Abbreviations</b>	<b>xiii</b>
<b>Latin symbols</b>	<b>xv</b>
<b>Math symbols</b>	<b>xix</b>
<b>Table of contents</b>	<b>xxi</b>
<b>List of Tables</b>	<b>xxiii</b>
<b>List of Figures</b>	<b>xxvii</b>
<b>Introduction</b>	<b>1</b>
<b>1 Problem definition</b>	<b>5</b>
1.1 Background . . . . .	5
1.2 Limitations of turbulence research tools . . . . .	8
1.3 Problem definition . . . . .	11
1.4 Datasets . . . . .	15
1.5 Concluding remarks . . . . .	21
<b>2 Velocity reconstruction using regression</b>	<b>23</b>
2.1 Regression in turbulence studies . . . . .	23
2.2 Ordinary least squares (OLS) . . . . .	24
2.3 Regularized linear regression . . . . .	25
2.4 Nonlinear regression . . . . .	28
2.5 A framework to select model and parameters . . . . .	30
2.6 Regression models for reconstruction of isotropic turbulence . . . . .	34

2.7	Concluding remarks . . . . .	38
<b>3</b>	<b>Learning coupled dictionaries</b>	<b>41</b>
3.1	From bases to dictionaries . . . . .	41
3.2	Proper Orthogonal Decomposition (POD) as a representation of turbulent fields . . . . .	43
3.3	Dictionary learning as a new representation . . . . .	45
3.4	Learning coupled dictionaries . . . . .	49
3.5	Dictionary learning for isotropic turbulence fields . . . . .	56
3.6	Concluding remarks . . . . .	66
<b>4</b>	<b>Non-local similarity-based propagation model</b>	<b>69</b>
4.1	Non-local means . . . . .	69
4.2	Similarity-based model to propagate small scales . . . . .	72
4.3	Applying the propagation model to isotropic turbulence . . . . .	75
4.4	Concluding remarks . . . . .	82
<b>5</b>	<b>Bayesian fusion model</b>	<b>85</b>
5.1	A probabilistic model . . . . .	85
5.2	Bayesian fusion model . . . . .	87
5.3	Equivalent models . . . . .	91
5.4	Model simplifications . . . . .	93
5.5	Applying the Bayesian fusion model to DNS data of isotropic turbulence	95
5.6	Concluding remarks . . . . .	99
<b>6</b>	<b>Comparisons of model performances on various datasets</b>	<b>101</b>
6.1	Reconstruction results on DNS data of isotropic turbulence . . . . .	102
6.2	Reconstruction results on DNS data of channel flow . . . . .	109
6.3	Concluding remarks . . . . .	113
	<b>Conclusions and perspectives</b>	<b>117</b>
	<b>Bibliography</b>	<b>123</b>
<b>A</b>	<b>Bayesian framework</b>	<b>131</b>
A.1	Derivation of Linear Gaussian models . . . . .	131
A.2	Derivation of generalized Millman formula . . . . .	135
	<b>Contents</b>	<b>139</b>

# List of Tables

- 1.1 Configuration parameters of three subsampling cases in space and three in time for DNS channel flow data. The subsampling ratios of HTLS measurements are  $\sqrt{P/Q}$  and equal in both spatial directions. The ratios of LTHS measurements in time are  $N/M$ . The equivalent spacing in spanwise direction is normalized by half channel height as  $\Delta z/H$  and the spacing in time is  $\Delta t$ . The normalized energy losses in space  $\Delta \kappa_s$  and in time  $\Delta \kappa_t$  are defined in equation 1.2. . . . . 16
- 1.2 Configuration parameters of three subsampling cases in space (spanwise+vertical direction) and three in time (streamwise direction) for isotropic turbulence data. The subsampling ratios of HTLS measurements are  $\sqrt{P/Q}$  and equal in both spatial directions. The ratios of LTHS measurements in time (spanwise in this case) are  $N/M$ . The normalized energy losses in space  $\Delta \kappa_s$  and in time  $\Delta \kappa_t$  are defined in equation 1.2. The slopes are purely to qualify the steepness of the spectrum and have no physical meaning. . . . . 20
- 2.1 Common basis functions for kernel methods. . . . . 30
- 3.1 Notations for three different methods of coupled dictionaries learning as proposed by Yang et al., (2010) and Zeyde et al., (2012). The dimension of LR and HR patches are  $q$  and  $p$  respectively. . . . . 54
- 3.2 Set of parameters for coupled dictionaries approach (notations are consistent with Algorithm 4): sparsity constrains  $\lambda_1$  for learning and  $\lambda_2$  for reconstruction; dimension of LR patches  $q$  and HR patches  $p$ ; number of atoms  $K$ ; number of training patches  $m$  . . . . . 56
- 4.1 Set of parameters to do simple grid search (try out all possible combinations from this set): the global filter parameter  $\sigma$ , size of similarity patches  $p$ , size of accumulation patches  $r$ , and size of searching region  $s$ . The search is performed on the DNS data of isotropic turbulence, with the subsampling ratios are  $P/Q = 3 \times 3$  in space and  $N/M = 4$  in time. 75

4.2	Configuration parameters of six testing cases. The subsampling ratios of HTLS measurements are $\sqrt{P/Q}$ and equal in both spatial directions. The ratios of LTHS measurements in time are $N/M$ . The normalized energy losses in space $\Delta\kappa_s$ and in time $\Delta\kappa_t$ are defined in Eq. (1.2). . . . .	81
5.1	Configuration parameters of seven testing cases. The subsampling ratios of HTLS measurements are $\sqrt{P/Q}$ and equal in both spatial directions. The ratios of LTHS measurements in time are $N/M$ . The normalized energy losses in space $\Delta\kappa_s$ and in time $\Delta\kappa_t$ are defined in equation 1.2. . . . .	97
5.2	NRMSEs of all scales reconstruction errors for the seven cases presented in table 5.1. $\bar{\epsilon}$ and $\epsilon_{max}$ are the average and maximum NRMSE defined by equation 2.28. $\bar{\epsilon}$ is averaged over all fields, while $\epsilon_{max}$ is computed for the most difficult positions in space and time (the most remote from all nearby measurements at local coordinates $(\Delta y/2, \Delta z/2, \Delta T/2)$ , see figure 1.6). These errors are shown for reconstructed fields by spatial interpolation $\mathbb{I}_s \mathbf{y}$ , time interpolation $\mathbb{I}_t \mathbf{x}$ , linear Gaussian model (LG) using formula 5.36 and Bayesian fusion model (BF) using formula 5.35. The smallest errors in each cases are boldfaced. . . . .	97
5.3	Average NRMSEs $\bar{\epsilon}$ of large and small scales (same notations in table 5.2). . . . .	98
6.1	Isotropic turbulence: configuration parameters of three testing cases with balanced energy losses. The subsampling ratios of HTLS measurements are $\sqrt{P/Q}$ and equal in both spatial directions. The ratios of LTHS measurements in time (streamwise in this case) are $N/M$ . The normalized energy losses in space $\Delta\kappa_s$ and in time $\Delta\kappa_t$ are defined in Eq. (1.2). . . . .	103
6.2	NRMSEs of all scales reconstruction errors for three cases in table 6.1. $\bar{\epsilon}$ and $\epsilon_{max}$ are the mean and max NRMSE defined in Eq. (2.28). $\bar{\epsilon}$ is averaged over all fields, while $\epsilon_{max}$ is computed for the most difficult positions in space and time (the most remote from all nearby measurements). The smallest errors in each cases are boldfaced. . . . .	104
6.3	Channel flow: configuration parameters of seven testing cases. The subsampling ratios of HTLS measurements are $\sqrt{N/M}$ and equal in both spatial directions. The ratios of LTHS measurements in time are $P/Q$ . The spacing in spanwise direction is normalized by half channel height as $\Delta z/H$ and the spacing in time is $\Delta t$ , normalized by $H/U_{max}$ ( $U_{max}$ is the central velocity of the flow). The normalized energy losses in space $\Delta\kappa_s$ and in time $\Delta\kappa_t$ are defined in Eq. (1.2). . . . .	109



- 
- 6.4 Channel flow: NRMSEs of all scales reconstruction errors for three cases in Table. 6.3.  $\bar{\epsilon}$  and  $\epsilon_{max}$  are the mean and max NRMSE defined in Eq. (2.28).  $\bar{\epsilon}$  is averaged over all space-time positions in the outer region  $y/H \in [0.25, 1.75]$ , while  $\epsilon_{max}$  is computed for one of the most difficult position in space and time (the most remote from all nearby measurements). The smallest errors in each cases are boldfaced. . . . 110



# List of Figures

- 1.1 Schematic representation of turbulence energy cascade (Davidson et al., 2015), following the sketch by Frisch, (1995). . . . . 7
- 1.2 Turbulence energy spectrum with relative separation between large and small scales by a cutoff wave number  $k_c$ . . . . . 12
- 1.3 Spectrum of models ordered by their level of nonlinearity (from left to right). Methods for mapping large-small scales are in blues, while those for fusion of measurements are in reds. Notations: linear regression (LR), proper orthogonal decomposition (POD), Kalman filter (KF), kernel regression (KR), similarity-based (S-B), dictionary learning (DL), Bayesian network (BayesNet), neural network (NN). . . . . 12
- 1.4 Turbulence energy spectrum with relative separation between large and small scales at wave number  $k_c$ . . . . . 13
- 1.5 Sketch of the inverse problem, with the two sources of measurements: the LTHS (color images) and a coarse grid of HTLS (red dots among black ones of LTHS). The inverse problem of HTLS data reconstruction is to fill in the space-time data-cube. . . . . 14
- 1.6 Sketch of an element block with local coordinates  $(\alpha, \beta, \tau)$ . LTHS time steps are at  $\tau/\delta t = 0$  and  $\tau/\delta t = N/M$ . HTLS measurements are represented by red dots and LTHS measurements by black ones. . . . . 16
- 1.7 A sample 2D streamwise velocity of simulated channel and its subsampled field by a factor of 10 (left). This subsampling ratio corresponds to the cut-off wave number  $k_2$  in the plot of spectrum (right) for horizontal lines at the center of the channel. Other wave number  $k_1$  and  $k_3$  correspond to the subsampling ratio in space of 5 and 10 respectively. . . . . 17
- 1.8 A sample 3D streamwise velocities (left) and vortices magnitude (right) of DNS isotropic turbulence at  $384^2$ . The streamwise direction is considered as “time” dimension in this case. . . . . 18
- 1.9 A sample isotropic 2D streamwise velocity (left) and its subsampled field by a factor of 3 (right) on the plane normal to flow direction. The subsampling is a forward problem, while going from a subsampled to fully-resolved field is an inverse problem. . . . . 19

1.10	2D energy spectral of the reference DNS of resolution $396^3$ . The maximum real wave number is $k = 128$ after removing the aliasing contents. Different cutoff wave numbers are shown: $k_{max} = 128/4$ corresponding to the pre-downsampling, $k_1 = k_{max}/3$ , $k_2 = k_{max}/4$ and $k_3 = k_{max}/6$ for subsampling ratios of $\sqrt{P/Q} = 3, 4$ and $6$ respectively. . . . .	20
2.1	Typical behavior of prediction error for testing (red) and training (blue) dataset as a function of a hyper-parameter (Hastie et al., 2009a). Different curves are for various datasets. The solid ones are expected errors (average of all curves). From left to right can be the direction of decreasing regularization or increasing model complexity. . . . .	32
2.2	Schematic view of the behavior of bias and variance (Hastie et al., 2009a). The blue-shaded, centered at the truth, is the realization space, where maximum distance is the irreducible error. All possible models are bounded by the red/magenta model space curves. Bias is shown by the black lines as the distance to the truth. Best models are shown as black dots, and circled by model variances. . . . .	33
2.3	Shrinkage effect: coefficients of OLS (left) vs RR (right) models correspond to an input LR measurement at the center of the field. The coefficients, which are rearranged in a 2D field of the size $96 \times 96$ , can be interpreted as the contributions of this input to the reconstruction of all other high-resolution points. The higher the coefficient, the stronger the impact. . . . .	35
2.4	Shrinkage effects of L2 (top) and L1 (bottom) penalty: coefficients corresponding to eight high-resolution outputs (in eight different colors) as functions of regularization parameters $\lambda$ . Higher $\lambda$ shrink the coefficients toward zeros differently depending on the penalty terms. . . . .	36
2.5	RR validation curve: errors as functions of regularization parameter $\lambda$ (left), and learning curve: errors as functions of training data size (right). Red curves are for the prediction, while blue ones are for the training data . . . . .	37
2.6	KRR validation curve (RBF kernel), with parameters are kernel parameter $\gamma$ and regularization parameter $\lambda$ . Red curves are for the prediction, while blue ones are for the training data . . . . .	38
3.1	An example of single image super-resolution using coupled dictionary learning and patch-wise approach. Photo credit: Yang et al., (2010). . . . .	49
3.2	Dictionaries leaned by PCA and ODL from the set of HR patches of size $p = 16 \times 16$ . With ODL, only 256 atoms are chosen from 512 atoms. . . . .	58

- 3.3 Sparsity vs error outside (left) and zoom in the region of low sparsity at semi-log scale (right) for different representations: wavelet (Daubechies), PCA, dictionary learning (ODL or KSVD). Wavelet transform is for the whole fields of size  $96 \times 96$ , while PCA and DL are for patches of size  $p = 16 \times 16$ . . . . . 59
- 3.4 Coupled dictionary of high and low resolution patches, with LR patches are of size  $4 \times 4$ , directly subsampled from their equivalent HR patches of size  $16 \times 16$ . The regularization parameter  $\lambda$  for join learning are chosen such that about 16 non-zero coefficients are retained for reconstructing the joint patches of the training. . . . . 60
- 3.5 Coupled dictionary of high and low resolution patches, where LR patches are of size  $4 \times 4$ , directly subsampled from their equivalent HR patches of size  $16 \times 16$ . The regularization parameter  $\lambda$  for join learning are chosen such that about 16 non-zero coefficients are retained for reconstructing the joint patches of the training. . . . . 61
- 3.6 Dictionary of the residual between HR and interpolated LR. . . . . 63
- 3.7 Means and standard deviations of NRMSEs estimated between reference and reconstructed fields of all middle planes (at the center of blocks bounded by the two LTHS planes). The reconstructions are by spline interpolation and super-resolution using three different methods: *SR1*, *SR2* and *SR3* (see table 3.1). NRMSEs are  $0.267 \pm 0.021$ ,  $0.235 \pm 0.019$ ,  $0.233 \pm 0.018$  and  $0.230 \pm 0.019$  respectively. The NRSME of spline interpolation in the equivalent subsampling case is 0.276 (dashed black line). . . . . 64
- 3.8 2D spectra of all planes used to computed the NRMSEs in figure 3.7, from reference, interpolation and SR by three different methods: *SR1*, *SR2* and *SR3* (see table 3.1). For scales from  $0.5k_c$  to  $1.5k_c$ , energy losses of SR fields compared to reference ones is 24%, 22% and 21% respectively, while that of interpolation is 85%. . . . . 64
- 3.9 (Left) 2D spectra of errors, which are the different between the reference and reconstruction by interpolation and three different SR methods as described in 3.4.3. (Right) 2D spectra of errors normalized by the energy spectrum of the reference (the black curve in figure 3.8). . . . . 65
- 4.1 Image denoising using non-local means. Four sample patches in red extracted from noisy image and centered at blue dots. The estimate of each point are weighted average of its neighbors, and weights are computed as the similarity of the moving patches compared to the reference patch. Photo credit: Foi and Boracchi, (2016). . . . . 70

- 4.2 A sketch of *greedy* and *non-greedy* propagation models. The two LTHS planes are at the border including small scales  $\mathbf{b}_{t_0} = \mathbf{h}_0$  and  $\mathbf{b}_{t_0+1} = \mathbf{h}_{N/M}$  defined in equation 4.5. All small scales to be estimated from other planes  $\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{P/Q-1}$  will be propagated either directly from LTHS planes (greedy, red arrows) or in a plane-by-plane manner (non-greedy, blue arrows). To simplify the plot, the propagation is shown from LTHS planes to the central plane only. . . . . 72
- 4.3 Average NRMSEs over the whole dataset: each estimate of NRMSE is computed by comparing the reconstructed plane with the reference at each  $t/\delta t$ . The bars show the average NRMSEs, while the error bars are standard deviations of NRMSE estimates. . . . . 77
- 4.4 Normalized error spectra of reconstructed planes at  $t/\delta t = 2$  (the most remote from the two neighboring LTHS planes). Three cases of varying  $p, r$  and  $s$ , all curves are almost collapsed. . . . . 78
- 4.5 Comparison of averaged NRMSEs as functions of time distances to the previous bounded LTHS plane by spatial/temporal interpolation, greedy and non-greedy propagation for the configuration where subsampling ratios are  $3 \times 3$  in space and 4 in time. . . . . 79
- 4.6 Comparison of energy spectra and error spectra for various reconstructions by spatial/temporal interpolation, greedy and non-greedy propagation. Subsampling ratios are  $3 \times 3$  in space and 4 in time. The spectra are averaged over all planes of the same distance to the closest LTHS plane, i.e.  $t/\delta t = 1$  (left) or  $t/\delta t = 2$  (right). . . . . 80
- 4.7 Averaged NRMSEs of reconstruction by spatial/temporal interpolation, greedy and non-greedy propagation models. The errors are shown for three cases, with a fixed ratio in space ( $P/Q = 3 \times 3$ ) and varied ratios in time ( $N/M = 4, 6, 8$ ). The reference plane is chosen to be the mid-plane, and  $\Delta t_c/\delta t$  is the distance of each other plane from the reference one. Different colors refer to different methods, while the marker shapes vary for different  $N/M$ . . . . . 82
- 4.8 Averaged NRMSEs of reconstruction by spatial/temporal interpolation, greedy and non-greedy propagation models. The errors are shown for different space spacing ( $P/Q = 3 \times 3, 4 \times 4, 6 \times 6$ ), while in time the ratio is fixed ( $N/M = 4$ ). The reference plane is chosen to be the mid-plane, and  $\Delta t_c/\delta t$  is the distance of each other plane from the reference one. Different colors refer to different methods, while the marker shapes vary for different  $P/Q$ . . . . . 83

- 6.1 Isotropic turbulence: NRMSEs are functions of time distances from the previous LTHS instant at the most difficult spatial location, i.e. at  $(\Delta y/2, \Delta z/2, \tau)$ . These errors are estimated between reference and reconstructed streamwise velocities by spatial and temporal interpolations, RR, KRR, non-greedy propagation and Bayesian fusion models. Results are shown for case 5 in table 6.1, with the subsampling ratios  $N/M = 4$  in time and  $P/Q = 3 \times 3$  in space. . . . . 105
- 6.2 Isotropic turbulence: similar plot as figure 6.1 for case 7 in table 6.1, with the subsampling ratios  $N/M = 8$  in time and  $P/Q = 6 \times 6$  in space. 105
- 6.3 Isotropic turbulence: NRMSEs are functions of spatial coordinates in an element block at the most difficult instant, i.e. at  $(\alpha, \beta, N\delta t/2M)$ . These errors are estimated between reference and reconstructed streamwise velocities by spatial and temporal interpolations, RR, KRR, non-greedy propagation and Bayesian fusion models. Results are shown for case 5 in table 6.1, with subsampling ratios of  $N/M = 4$  in time and  $P/Q = 3 \times 3$  in space. . . . . 106
- 6.4 Isotropic turbulence: NRMSEs are functions of spatial coordinates as in figure 6.3. Results are shown for case 7 in table 6.1, with subsampling ratios of  $N/M = 8$  in time and  $P/Q = 6 \times 6$  in space. . . . . 107
- 6.5 Isotropic turbulence: comparison of energy spectra and error spectra for reconstructions by interpolations (in space and time), KRR, non-greedy propagation and Bayesian fusion model in case 5 of subsampling ratios  $P/Q = 3 \times 3$  in space and  $N/M = 4$  in time (see table 6.1). The spectra are averaged over all planes of the same distance to the closest LTHS plane, i.e.  $t/\delta t = 1$  (left) or  $t/\delta t = 2$  (right). . . . . 108
- 6.6 Channel flow: NRMSEs between reference and reconstructed streamwise velocities by all methods as: functions of time distances from the previous LTHS instant at the most difficult spatial location, i.e. at  $(\Delta y/2, \Delta z/2, \tau)$ . Results are shown for case 5 in table 6.3, with the subsampling ratios  $N/M = 10$  in time and  $P/Q = 10 \times 10$  in space. . . . . 111
- 6.7 Channel flow: NRMSEs between reference and reconstructed streamwise velocities by all methods as: functions of spatial coordinates in an element block at the most difficult instant, i.e. at  $(\alpha, \beta, N\delta t/2M)$ . Results are shown for case 5 in table 6.3, with the subsampling ratios  $N/M = 10$  in time and  $P/Q = 10 \times 10$  in space. . . . . 111
- 6.8 Channel flow: a time evolution of fluctuating streamwise velocity at  $y/H = 1$  and  $z/H = 0$  of a point with the local coordinate  $(\Delta y, \Delta z, \tau)$  (see figure (1.6)). The whole evolutions of reference and fused velocity contain 900 samples. A region is zoomed in and compared with results from interpolations and RR. . . . . 112

---

6.9	Channel flow: spectra of the fluctuating velocity in figure 6.8. . . . .	112
6.10	Channel flow: probability distribution functions of velocity increments in (left) the original DNS and (right) the reconstructed field. . . . .	113
6.11	Channel flow: a sample snapshot of fluctuating streamwise velocity at one of the most difficult instant to estimate (in the middle of two LTHS time steps): Reconstruction of all scales (left) and large scales only (right). The figure is better viewed on screen. . . . .	114



# Introduction

Turbulence is very common in daily life, science and technology. Understanding its nature and increasing our prediction capability is crucial for many applications. One of the main engineering tasks is to calculate and predict the behaviors of turbulent flows. This is essentially done with numerical simulations, since experiments are much more expensive or sometimes even impossible. Exact simulations of turbulent flows without modeling are possible only in academic contexts and limited to simple cases due to huge computational demands. Approaches with turbulence models, i.e. approximate solutions, are used more often in industries. Some only calculate mean quantities, while others approximate only large scales and model the effects of small ones. However, they are not yet satisfactory, mainly due to their incapability of predicting several flow configurations, for instance flow separation. All fails due to the ignorance or crude modeling of small scales.

Small-scale turbulence is very important but not yet fully accessible. Despite advancements of computational resources and experimental techniques, having access to small-scale turbulence remains very challenging. Understanding its physics is the key to propose realistic models by parameterizing the effects of the small scales on the large ones, reducing the simulation efforts to much coarser grids. The physical properties of the flow at small-scale are also important in many applications such as combustion or biology. Unfortunately, accessing those information remains very difficult. Progressing further in experimental tools is one solution but will take very long time. It requires also advancements of infrastructure to handle such tremendous amount of data. Theoretical models generating small scales from large ones could help to progress in another direction. However, as discussed later, such a model is difficult to build from turbulence theories. Despite many efforts, all theories are not yet satisfactory to fully model the behaviors of small scales.

This thesis approaches the problem of estimating small-scale information from a different perspective. Empirical models and learning algorithms are used to propose computational methods to reconstruct full-scale information from available sparse measurements. They are further adapted or developed from works in signal and image

processing, taking into account the key similarities and differences between turbulence and natural images. This can be viewed as the inverse problem of seeking a higher information level out of available measurements.

Outcomes of this work may improve our understandings of turbulence by giving access to information that can not be directly measured. Further demonstrations of the empirical relation between scales are shown, demonstrating how far the model can reconstruct small scales given the large ones. This is beneficial for turbulence modeling, since models can be trained from available datasets and give access to small scales in new situations. Follow-up works may use those information and improve the prediction of large scales, avoiding the use of crude turbulence models.

This work is also beneficial for data compression or compressive sensing purposes. It deals with the quality of approximate reconstruction of physical quantities given a certain amount of measurements. This is directly connected to another active research domain of 3D data compression of large databases in fluid mechanics. This work characterizes the expected small-scale information level given a certain amount of measurements. Reversely, it also characterizes how many measurements are required for a certain level of desired information.

## Organization and contributions

The following parts of the thesis will be organized as follows:

**Chapter 1.** *Problem definition:* This chapter shortly reviews the importance of turbulence studies, especially the challenging problem of understanding and modeling small-scale turbulence. The shortage of research tools to access those information is addressed, leading to the main motivation of this thesis: *to propose computational methods to reconstruct small scales from available measurements*. Two problems are addressed and solved in this thesis: estimating small scales from large scales; and fusing available measurements to access higher information level. For each problem, a spectrum of potential methods are reviewed/proposed and compared to facilitate the usage in new situations.

**Chapter 2 and 3.** *Estimating small scales from large scales:* These two chapters tackle the problem of seeking a mapping function that permits to estimate small-scale information from measurements of large scales.

Chapter 2 discusses the family of regression models. It reviews conventional methods such as ordinary least squares and other regularized regression models such as ridge regression, LASSO and kernel methods. Parameter optimization via cross-validation is also discussed as the way to choose the optimal set of parameters for each model.

Chapter 3 introduces the approach based on *dictionary learning* method, which

generalizes principal component analysis to permit sparsity and redundancy properties. To find the mapping between large and small scales, coupled dictionaries are learned to represent large and small scales. When only large-scale information are accessible, its representation is estimated and then combined with the dictionary of small scales learned *a priori* to reconstruct those unknown details.

**Chapter 4 and 5.** *Fusion of complementary measurements:* These two chapters aim at proposing fusion models to combine information from multi-source measurements to estimate small-scale information. We will focus on the situation where two sources of measurements are available: the high-temporal-low-spatial and low-temporal-high-spatial resolution data.

Chapter 4 proposes a model to propagate small scales from the low-temporal-high-spatial planes to other instants in time. The model is based on a non-local means filter, where small scales are propagated in time based on the similarity between larger scales from different planes. The estimation is done using a patch-based overlapping approach.

Chapter 5 proposes a Bayesian fusion model to combine the measurements in space and time. This model is constructed based on a Bayesian framework where a prior knowledge about the flow can be introduced into the estimation. Bayesian estimators are used to find the most probable high-resolution fields given the measurements. The final model is a simplified version, leading to a linear fusion model. This model contains two important ingredients, the local structures of the flow and the statistical parameters learned from data to encode the flow physics.

**Chapter 6** compares performances of proposed models on various datasets. Detailed analyses are performed to demonstrate benefits of each models compared to simple interpolations. The configuration of complementary measurements in space and time are studied, which permit to investigate all proposed models.

This work has been resulted in the following publication:

Nguyen et al., (2015). “A Bayesian fusion model for space-time reconstruction of finely resolved velocities in turbulent flows from low resolution measurements”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2015.10, P10008.

and has been presented at the following international/national conferences:

- *15th European Turbulence conference*, Delft, Netherlands (August 2015)
- *GDR Turbulence*, Grenoble, France (June 2015)

All codes to reproduce the results of this work are available at <https://github.com/linhvannnguyen/PhDworks/>



# Problem definition

This chapter presents the context of the thesis, which aims at proposing empirical models to estimate small-scale information of turbulent flows from low resolution measurements. This work is addressed due to the shortage of research tools that could give a complete view of turbulent flows, with information at both large and small scales. The thesis brings forth two problems being solved. The first one is to find functions to estimate high-resolution fields, containing both large and small scales, from low-resolution measurements, essentially large-scale information. The second problem is to find fusion models to combine multiple sources of low-resolution measurements to reconstruct fully resolved velocity fields. The proposed methods will be tested on two direct numerical simulation datasets of either isotropic turbulence or turbulent channel flow at moderate Reynolds numbers. The data give access to the full information and permit to conduct various numerical experiments and study the performances of the reconstructions.

## 1.1 Background

This section briefly reviews the importance of turbulence studies. The main properties of turbulent flows are discussed, notably as a multi-scale phenomenon. The section further emphasizes the importance of studying small scales, which leads to the main motivation of the thesis.

### 1.1.1 Turbulence

Turbulence occurs very often in both nature and engineering applications. It appears in our surroundings, since most fluid flows are turbulent. Turbulence scales can be as extremely small as interior of biological cells or as large as flows in geophysics, astrophysics, oceans or atmospheres. In engineering applications, it dominates most

flows around vehicles such as cars, ships or airplanes. It also plays key roles in energy production and transformation or inside combustion chambers. A nice review on these applications of turbulence is given by George, (2009), McDonough, (2004), and Tennekes and Lumley, (1972).

Turbulence brings both desirable and undesirable effects. On the one hand, it enhances mixing and transporting matter, momentum and heat. This property is beneficial since it improves fuel-air mixing and accelerates cooling process in combustion chamber for instance. In civil engineering, turbulence helps releasing pollutant streams into the air or ocean more rapidly. It also enhances mass and heat transfer at solid-fluid or fluid-air interface. In aerospace engineering, it helps increasing wall shear stress on aircraft wing and therefore contributes to avoid separation. On the other hand, turbulence also causes serious consequences. Aerodynamics drag is one of the most troublesome problems. Airbus estimates that 10 % reduction of drags caused by turbulent boundary layer would result in one billion Euros fuel saving worldwide per year, along with the contribution to reduce pollution. Turbulence is also the main source of aircraft noises. Better knowledge will help to propose strategies of controlling turbulence in beneficial ways and reduce energy footprint in transportation.

Turbulence studies have mainly focused on theoretical aspects after the discovery of Navier-Stokes (N-S) equations. The experimental researches on fluid flows have only started since O. Reynolds experiment (Reynolds, 1883), motivating later works on non-dimensional parameters, closure models, and similarity theories for homogeneous and isotropic turbulence. More recent works have mainly focused on large-scale coherent structures and vortex organizations (Hutchins and Marusic, 2007). These works are only possible thanks to the rapid progresses in experimental techniques (mostly optical devices using micro and nano technologies) and computing power. These new results help in increasing our knowledge on turbulence nature, hence improving predictive capacity and proposing more accurate turbulence models for commercial software.

Despite constant efforts in the last two centuries, turbulence remains very challenging. The problem has been tackled by many great scientists such as Boussinesq, O. Reynolds (19th), Prandtl, Taylor, Kolmogorov (20th), to name only a few. However, it remains one of the last unsolved problems of classical mathematical physics. Analytical solutions are usually not accessible. The most recent works on N-S equations still discuss the existence of the solution (Chemin and Gallagher, 2009; Gallagher, 2015; Gallagher et al., 2003). The efforts to measure, calculate or model turbulence are also not fully successful in most practical problems due to its properties as a random, multi-scale and three-dimension irregular phenomenon.

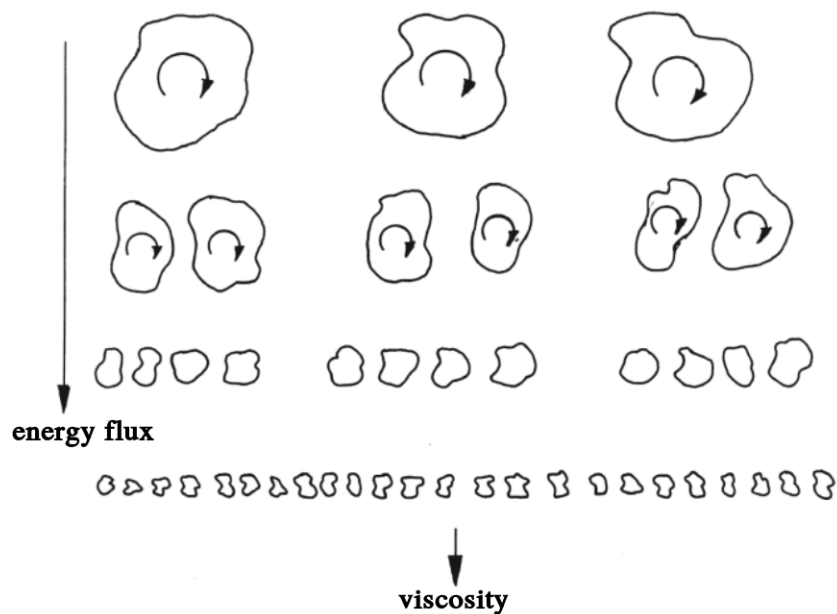


Figure 1.1 – Schematic representation of turbulence energy cascade (Davidson et al., 2015), following the sketch by Frisch, (1995).

### 1.1.2 Importance of small-scale turbulence

Turbulence is a multi-scale phenomenon, involving scales from the size of the domain till the Kolmogorov dissipative micro-scales. These scales co-exist and interact with each other. Large scales usually contain most of the kinetic energy of the flow, therefore responsible for the transportation of matter, momentum and heat. Small-scale turbulence contains less energy and is strongly involved in the dissipation. Turbulence kinetic energy usually cascades from large to small scales, as schematically represented in figure 1.1. The range of scales is continuous, and the width of this range depends on the Reynolds number of the flow.

All scales are important to characterize turbulent flows. For instance, an accurate description of all scales is required to estimate full Reynolds stress tensor to validate numerical simulations, turbulence models and scaling theories. This validation is usually performed at very low frequencies, demanding measurements with large field-of-view and long time records. Unsteady flow simulations resolve very small scales in space and time, predict the sharp gradient in the shear layer or near-wall turbulence. Validating these simulations requires small-scale information up to the Kolmogorov length scale.

Small-scale turbulence is crucial to understand the physics of the flow, since these scales are responsible for the breaking and reconnection of vortices. They are also more likely universal/quasi-universal compared to large scales. A good understanding could lead to turbulence models where the presence of small scales can be parametrized by more universal mathematical models. The efforts to simulate flows could be simplified to

large-scale turbulence only, reducing significantly computation costs. Moreover, small scales are crucial in some contexts such as combustion where matters are brought together at the molecular scales.

## 1.2 Limitations of turbulence research tools

Despite the constant progress in experiments and numerical simulations, a full access to small-scale turbulence remains impossible for flows at high Reynolds numbers. The section reviews briefly recent advancements and critical limitations.

### 1.2.1 Numerical simulations

In academic context, direct numerical simulation (DNS) is the only available tools to directly solve N-S equations without any additional model. It yields all information of the flow via the numerical simulation using an extremely fine grid to resolve all range of scales, from the largest (of the domain size) to the smallest eddies (of the order of Kolmogorov scales). High order schemes such as compact finite difference or spectral methods are used to discretize and solve the equations with a very high accuracy.

Constant progress in computational resources permits DNS of high Reynolds number ( $Re$ ) and more complex flows. The first attempt of DNS was to simulate isotropic turbulence on a very coarse grid of  $32^3$  (Fox and Lilly, 1972; Orszag and Patterson Jr, 1972), while recent simulations of the same flows were performed at  $4096^3$  (Ishihara et al., 2009; Kaneda et al., 2003). More complex geometries are also studied, such as channel flows (Lee and Moser, 2014; Sillero et al., 2013), flat-plate boundary layer under zero and favorable pressure gradients (Spalart, 1986, 1988) and backward facing step (Le et al., 1997), among many others.

DNS is limited uniquely to academic context due to computational constraints. The required number of grid points is  $O(Re^{9/4})$  in space and  $O(Re^{3/4})$  in time to resolve the smallest scales. This becomes a real burden for most engineering flows at high  $Re$  and with complex geometries. One of the largest DNS is a channel flow simulation at  $Re_\tau \approx 5200$  based on friction velocity  $u_\tau$  (Lee and Moser, 2015) with the same domain sizes as the well-known simulation by Hoyas and Jiménez, (2006) at  $Re_\tau \approx 2000$ . This is close to  $Re$  of most engineering flows, about  $Re_\tau \approx 10^3$  (Smits and Marusic, 2013). However, many flows are at  $Re$  one order of magnitude higher and in much more complex geometries. Even when computers become fast enough to simulate those flows, running DNS for engineering flows could produce petabytes of data, which may exceed our capability to analyze and draw useful conclusions.



## 1.2.2 Experimental techniques

Experiments can provide some information that are not as complete as numerical simulations but for more realistic flows at high  $Re$  and more complex geometries. The most common techniques are hot wire anemometry (HWA) and particle image velocimetry (PIV).

### Hot Wire Anemometry

Hot Wire Anemometry (HWA), first proposed by King, (1914), is a simple yet accurate technique to measure one-point velocities. The anemometer uses a thin wire at very high temperature submerged into the flow. The flow changes the heat equilibrium around the wire, altering the temperature of the wire. The electronic circuit of the device provide to the wire a controlled amount of electrical current to keep the wire temperature constant. The velocity is deduced from the voltage change. Since the change is analog without any information loss, the temporal frequency of velocity measurements can reach about 100 kHz, which corresponds to Kolmogorov time scale of most realistic flows. The main constraint is the spatial discretization, since only point-wise measurements are possible, and a rake of hot wires soon becomes intrusive.

### Particle Image Velocimetry

Particle Image Velocimetry (PIV) is currently the best technique to measure global/topological view of flows. The method aims at recording images of tracers seeded in the flow, which are necessarily light and neutrally buoyant, to measure instantaneous velocity fields (Adrian, 2005; Keane and Adrian, 1992). Laser light sheets pulsed at a fixed time interval are used to illuminate a slice of the flow, and pairs of images are recored at the same interval by CCD cameras. Velocity at one point is estimated as the averaged displacement of its neighboring tracers within a so-called “interrogation window”. The most likely position of a group of tracers inside this window at the next time step is estimated using a correlation technique (Sutton et al., 1983). Final velocity fields are computed knowing the displacements and the interval between two images of the same pair.

The conventional PIV technique measures two-dimension two-component (2D-2C) velocity fields. *Stereo-PIV (SPIV)* measures two-dimension-three-component (2D-3C) fields (Prasad, 2000). It uses a system of two synchronized cameras to capture images of a seeded flow. These images are simultaneously taken but with distinct off-axis views so that “out-of-plane” velocities can be extracted. These two cameras are usually set up in an angular configuration where the axes of the two cameras are rotated while assuring

that they intersect with the object plane along the same axis following the Scheimpflug principle.

*Tomographic PIV* can measure 3D-3C velocity fields. It gives the most complete view of a flow, including the velocity gradient tensor, a crucial measure to validate numerical approaches. Proposed recently by Elsinga et al., (2005) and Elsinga et al., (2006), the main principles remain analogous to SPIV with important modifications. The illumination of the 3D volume is done with a thicker laser sheet, and 3D velocities are estimated using 3D cross-correlations. Recent measurements can reach 3D-3C fields of a volume of about  $200\text{ cm}^3$  in water or  $50\text{ cm}^3$  in air at a frequency of 3 kHz (Scarano, 2013).

“*Shake-The-Box*” (STB) calibration technique aims at improving further the spatial resolution by combining Lagrangian and Eulerian views of a flow. The setup resembles tomographic PIV. The time spacing is required to be of the order of the Kolmogorov time scale to ensure an accurate tracking of particle trajectories. From the time-resolved sequence of tomographic images, STB uses particle image matching to find the Lagrangian trajectory of each particles. Interpolation is used to get velocities on the Euclidean regular grid. This technique ideally reaches the resolution till particle sizes. A recent work using STB permits a resolution of 21 pixels per  $\text{mm}^2$  in a cube of  $10\text{cm} \times 10\text{cm} \times 2\text{cm}$  at 1 kHz for a flow at  $Re = 33000$  (Schröder et al., 2015).

Despite above advancements, PIV is still limited in measuring realistic flows due to optical devices and data processing constraints. The state-of-the-art measurements are still a question of finding compromises among high temporal frequency, high spatial resolution and large field of view. This is due to the following critical limitations.

The spatial resolution of PIV measurements is limited due to the correlation-based calibration technique. The maximum resolution is constrained by *image density* (Keane and Adrian, 1992), which is the average number of tracers in an integral window (or interrogation cell). This parameter is very important to determine the measurement quality. Too high image density causes the problem of multiple correlation peaks, while too low image density reduces the possibility to find an image pair for each interrogation window. The optimal choice of image density is about 8-10 tracers per interrogation window, which is a compromise between good spatial resolution and quality of autocorrelation estimate.

STB can overcome the spatial resolution limits by resolving the fields down to the particle size, but requires a system with a very high sampling frequency. Current PIV systems only reach low frequencies compared to the smallest time scales. State-of-the-art systems can capture up to 10000 frames per second (fps) at a resolution of one mega-pixel (Willert, 2015). This fps is very large but still not sufficient for most engineering flows in

air. These flows sometimes require measurements at frequencies at least one order of magnitude higher. Moreover, a high speed PIV requires a laser that combines extremely high pulsing frequencies and high power.

## 1.3 Problem definition

Complete views of turbulent flows at both large and small scales are desired. However, current facilities are not yet available to measure such information. Large-scale information are gradually available, but small-scale turbulence remains out-of-reach. This work addresses the problem of empirically estimating small-scale information from large-scale measurements. It aims at exploiting available measured data and propose computational methods to reconstruct full fields in new situations. Such methods may permit to go beyond what one can measure, compute or store. The idea of the co-conception approach can be studied, where results from this work potentially paves the way to design experiments or simulations in such way that facilitates the post-processing to recover a maximum level of information.

Through out this thesis, cubic spline interpolation is used to go from low to high resolution. It is used also as the benchmark to compare with the proposed models. Interpolated fields contain no small scales with the frequency higher than the cut-off associated with the grid of measured points, neglecting the aliasing effects due to the subsampling. For this reason, through out this thesis, the term *small-scale* is referred also as *high-frequency* (HF) or *high-resolution* (HR), while *large-scale* has the same meaning as *low-frequency* (LF) or *low-resolution* (LR).

### 1.3.1 Estimating high-resolution turbulent fields from low-resolution measurements

Structures in a turbulent flow can be relatively separated into large and small scales as shown in figure 1.2. Energy content  $E(k)$  is shown as a function of each wave number  $k$  in log-log scale. This Fourier spectrum is continuous, and under certain conditions it has a  $k^{-5/3}$  region. The ranges of scales and their energy contents are very wide depending on  $Re$ . Large scales  $z_{lf}$  and small scales  $z_{hf}$  are separated at the cutoff wavenumber  $k_c$ . We consider the situation where  $z_{lf}$  - the low spatial resolution or low temporal frequency fields- is measured while  $z_{hf}$  is not accessible.

The question is *can small scales be estimated from large-scale measurements?* This can be considered as whether a mapping function  $f$  exists such that, given LF

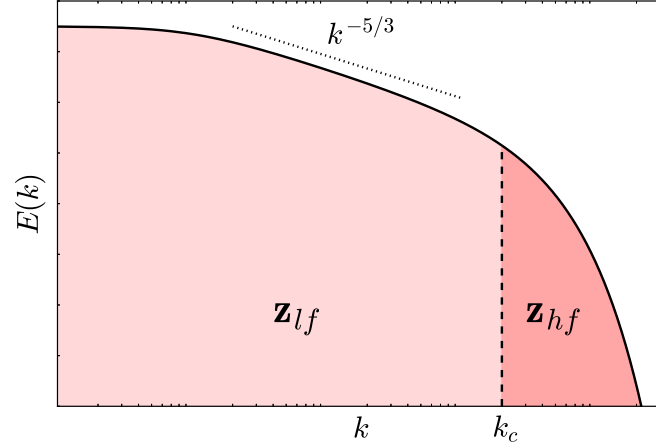


Figure 1.2 – Turbulence energy spectrum with relative separation between large and small scales by a cutoff wave number  $k_c$ .



Figure 1.3 – Spectrum of models ordered by their level of nonlinearity (from left to right). Methods for mapping large-small scales are in blues, while those for fusion of measurements are in reds. Notations: linear regression (LR), proper orthogonal decomposition (POD), Kalman filter (KF), kernel regression (KR), similarity-based (S-B), dictionary learning (DL), Bayesian network (BayesNet), neural network (NN).

measurements, the HF contents  $z_{hf}$  of the flow can be predicted by:

$$f : z_{lf} \mapsto \hat{z}_{hf} = f(z_{lf}) \quad (1.1)$$

No theoretical mapping function exists, although turbulent flows are governed by N-S equations. Despite constant attempts, researchers have not succeeded in proposing turbulence models to represent universal relations between large and small scales.

This thesis aims at proposing empirical functions, either linear and nonlinear, to estimate small-scale information given the large scales. The use of “empirical” models is in the sense of learning, where data at all scales are available *a priori*. The proposed algorithms learn from the training data an empirical relation between large and small scales. This relation is presented in the form of a function  $f(\cdot)$ , which permits to estimate small-scale information in new situations.

A large spectrum of methodologies is discussed and compared. Figure 1.3 lists such methods (in blue) as a function of the order of their nonlinearity. Pure linear models are conventional regression and dimensionality reduction (POD)-based methods.

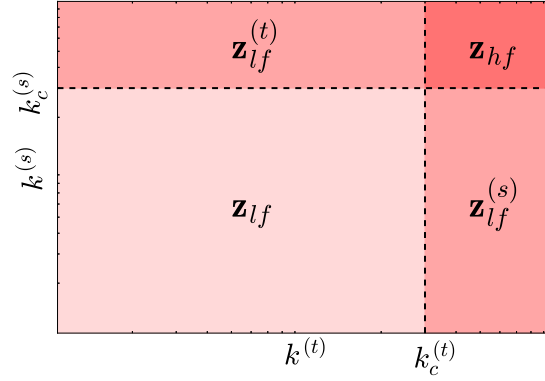


Figure 1.4 – Turbulence energy spectrum with relative separation between large and small scales at wave number  $k_c$ .

Nonlinearity level grows with model complexity. Simple nonlinear models, the so-called kernel regression, work in a fixed kernel space. The family of dictionary learning methods provides sparse representation of high dimensional data. Neural network, a broader and very active research field, finds nonlinear mapping functions with a more adaptive kernel space. A “deep” network permits to model highly nonlinear phenomena. However, we will leave this approach for future works.

### 1.3.2 Fusion of large-scale measurements

Turbulence is multi-scale both in space and in time. Usually large or small scales refer to either space (up to three dimensions) or time. However, LF measurements in space potentially contain some HF information in time and vice-versa. One idea is to measure those LF information by different measurement techniques and combine them to infer some HF contents. Such a fusion model takes benefit from each measurement by capturing all possible detailed information. The fused fields contain some small scales that are not accessible from each single measurement.

Figure 1.4 depicts schematically the above idea, where LF measurements in space and time are illustrated. In space, two sources of information  $\{\mathbf{z}_{lf}, \mathbf{z}_{lf}^{(s)}\}$  are available, where  $\mathbf{z}_{lf}$  contains purely LF information in space-time, and  $\mathbf{z}_{lf}^{(s)}$  captures LF in space but over the full range of temporal frequencies. An analogous interpretation holds for  $\{\mathbf{z}_{lf}, \mathbf{z}_{lf}^{(t)}\}$  in time, where  $\mathbf{z}_{lf}^{(t)}$  potentially gives some detailed information of the flow in space. By combining two sources of LF measurements in space and time, ideally one can hope to predict all scales in three blocks  $\mathbf{z}_{lf}$ ,  $\mathbf{z}_{lf}^{(t)}$  and  $\mathbf{z}_{lf}^{(s)}$ , while those within  $\mathbf{z}_{hf}$  are unreachable.

This work addresses a configuration as in figure 1.5 where two sources of measurements, the high-temporal-low-spatial (HTLS) and low-temporal-high-spatial (LTHS)

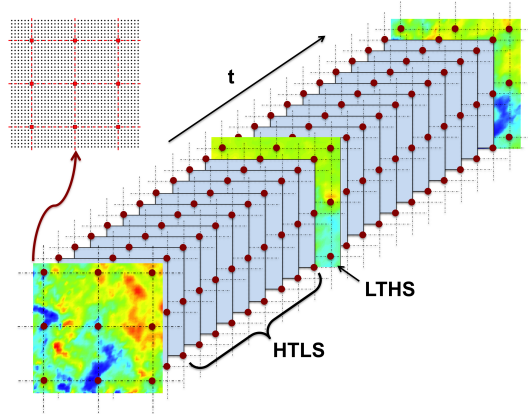


Figure 1.5 – Sketch of the inverse problem, with the two sources of measurements: the LTHS (color images) and a coarse grid of HTLS (red dots among black ones of LTHS). The inverse problem of HTLS data reconstruction is to fill in the space-time data-cube.

resolution velocity fields are available. LTHS measurements (black dots) at the three colorful planes are synchronized with HTLS measured positions at LR (red dots). HTLS captures HF contents of the flow dynamics, while LTHS measures highly resolved information in space. By combining these two sources, one hopes to capture more small-scale information in space and time.

This idea has been studied at *Laboratoire de Mécanique de Lille (LML)* in an European joint experiment (Coudert et al., 2011). A database of a boundary layer flow at high  $Re$  was measured using SPIV synchronized with a rake of HWA probes. PIV measurements (LTHS) are with a large field-of-view ( $30 \times 30cm$ ) and a high spatial resolution ( $143 \times 167$ ) but at a low acquisition frequency ( $4Hz$ ). HWA measurements in a rake of probes (HTLS) of the same field-of-view are sampled at an extremely high rate ( $50kHz$ ). However, the spatial discretization is very coarse ( $11 \times 13$ ) compared to the smallest scale. The post-processing step is to estimate velocity fields at the frequency  $50kHz$  and the resolution  $143 \times 167$ .

This work studies computational methods to combine HTLS and LTHS measurements to maximize the information level. The spectrum of possible methods as the order of nonlinearity is shown (in red) in figure 1.3. For completeness, the idea of data assimilation using linear/nonlinear Kalman filter is mentioned, but further discussion is outside the scope of this work. A fusion approach based on local similarity, the so-called similarity-based (S-B) scales propagation model, is proposed. This method further exploits the local information of the flow, with certain advantages and limitations compared to the linear fusion model. Another fusion method based on Bayesian framework is also proposed, which is further simplified as a linear model.

## 1.4 Datasets

Experimental data are available from the WALLTURB project (Coudert et al., 2011). However, there is no HR reference data in space-time to qualify and compare different reconstructions. This is the reason why DNS data are used in the present study to access fully resolved velocity fields. Data acquire the true properties of turbulent fields, since DNS simulates flows without any turbulence modeling. The data are also free from measurement uncertainties and noises, making possible a fair comparison of different models. LR measurements are virtually extracted from HR data without the presence of measurement noises. The proposed models are learned from these training data to reconstruct fully resolved fields, which are used to estimate the accuracy of different approaches by comparing to reference DNS data.

Two DNS datasets are used. The first data is from the DNS of a turbulent channel flow at a moderate  $Re$ . The designed numerical experiments on this data very well imitate the database from LML experiment. However, since the flow physics is strongly non-homogeneous in vertical direction, it is challenging for some models to perform. Data of isotropic turbulence is also used, with further assumptions of isotropy, periodicity and homogeneity, but still retains the principal properties of turbulence.

For both data, the configuration of HTLS and LTHS measurements as in figure 1.5 is studied, with the following notations to describe. HTHS, HTLS and LTHS snapshots at the  $t$ -th time step are denoted as  $\mathbf{z}_t \in \mathbb{R}^P$ ,  $\mathbf{y}_t \in \mathbb{R}^Q$  and  $\mathbf{x}_t \in \mathbb{R}^P$ , respectively. The dimension of HR fields ( $\mathbf{z}_t$  and  $\mathbf{x}_t$ ) is  $P$ , higher than the LR dimension  $Q$  of  $\mathbf{y}_t$ . There are  $N$  HTHS and HTLS snapshots, and only  $M$  LTHS ones ( $M < N$ ) sampled at  $t = 1, 1 + 1 \times N/M, 1 + 2 \times N/M, \dots, 1 + (M - 1) \times N/M$ . The matrix form of  $\{\mathbf{z}_t\}$ ,  $\{\mathbf{y}_t\}$  and  $\{\mathbf{x}_t\}$  are  $\mathbf{Z}$ ,  $\mathbf{Y}$  and  $\mathbf{X}$  of sizes  $N \times P$ ,  $N \times Q$  and  $M \times P$  respectively.

To describe all numerical experiments, it is useful also to introduce a sketch of an *element block* bounded by the measurements as in figure 1.6. This sketch is also used later to characterize performances of all models, since reconstruction accuracies depends on the relative positions in this block. A local coordinate system is introduced. The time position  $\tau$  is defined as a relative distance to two bounded LTHS snapshots (black dots) at  $\tau = 0$  and  $\tau = \Delta T = N\delta t/M$  ( $\delta t$  is the time distance between two neighboring HTLS planes). In space, each plane at  $\tau$  is bounded by the four HTLS measurements at the four corners (red points). The distances between two neighbors are  $\Delta y = \delta y \sqrt{P/Q}$  in vertical and  $\Delta z = \delta z \sqrt{P/Q}$  in spanwise direction.  $\delta y$  and  $\delta z$  are spatial grid sizes in vertical and spanwise direction respectively. We assumed also the sampling ratio are equally  $\sqrt{P/Q}$  in both directions. The most difficult positions to estimate are at local coordinate  $(\alpha, \beta, \tau) = (\Delta y/2, \Delta z/2, \Delta T/2)$ .

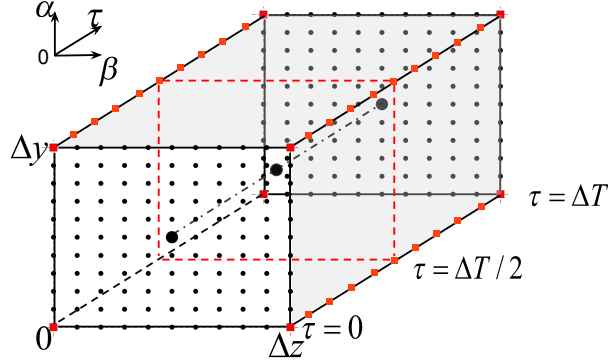


Figure 1.6 – Sketch of an element block with local coordinates  $(\alpha, \beta, \tau)$ . LTHS time steps are at  $\tau/\delta t = 0$  and  $\tau/\delta t = N/M$ . HTLS measurements are represented by red dots and LTHS measurements by black ones.

	Space			Time		
Subsampling ratio	5	10	20	4	10	20
Spacing	0.05	0.11	0.22	0.10	0.25	0.50
Energy loss $\Delta\kappa$ (%)	1.23	7.08	20.83	1.88	9.02	24.31

Table 1.1 – Configuration parameters of three subsampling cases in space and three in time for DNS channel flow data. The subsampling ratios of HTLS measurements are  $\sqrt{P/Q}$  and equal in both spatial directions. The ratios of LTHS measurements in time are  $N/M$ . The equivalent spacing in spanwise direction is normalized by half channel height as  $\Delta z/H$  and the spacing in time is  $\Delta t$ . The normalized energy losses in space  $\Delta\kappa_s$  and in time  $\Delta\kappa_t$  are defined in equation 1.2.

The following parts describe the two datasets in detail.

### 1.4.1 DNS data of channel flow

DNS database of a turbulent wall-bounded flow is designed the same way as experimental ones. The simulation uses the numerical procedure described in Marquillie et al., 2008. The flow is at a Reynolds number  $Re_\tau = 550$  based on the friction velocity  $u_\tau$ . Cartesian coordinates of the simulation in space are  $(x, y, z)$  for streamwise, vertical and spanwise directions respectively. The domain size normalized by half the channel height  $H$  is  $2\pi \times 2 \times \pi$ . Fully resolved fluctuating streamwise velocities in a plane normal to the flow direction are considered as HTHS data. This data includes  $N = 10000$  snapshots over a grid of  $P = 288 \times 257$  points and at a sampling frequency of  $f = \frac{H}{\delta t U_{max}} = 40$ ,



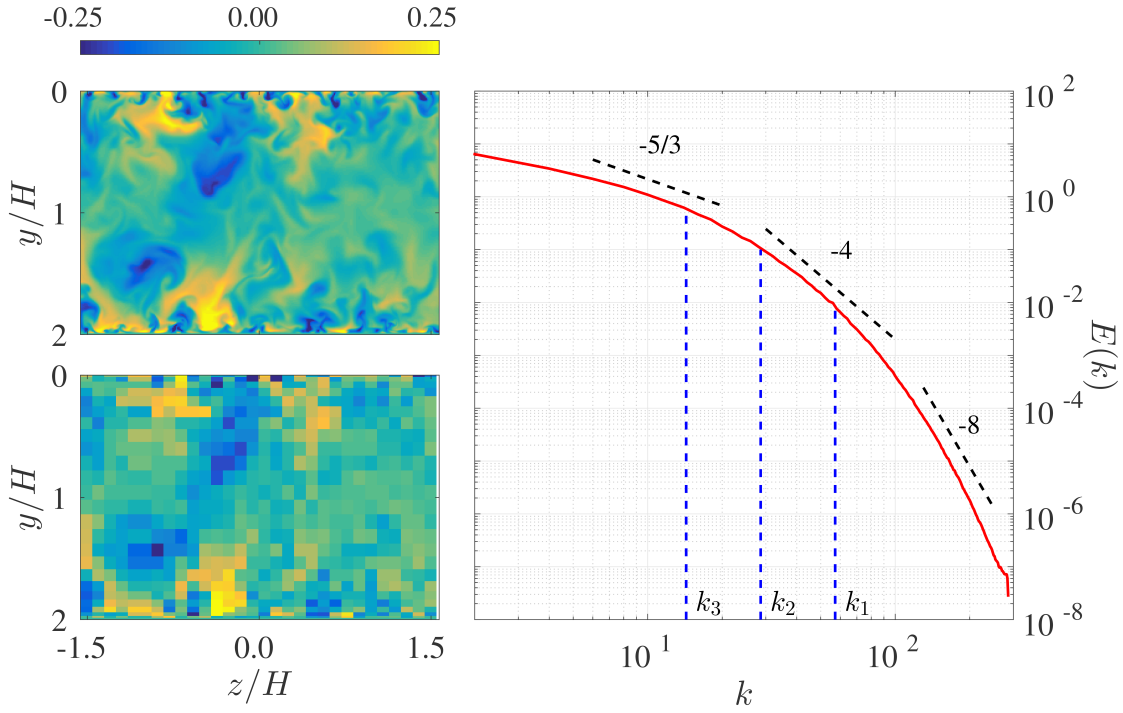


Figure 1.7 – A sample 2D streamwise velocity of simulated channel and its subsampled field by a factor of 10 (left). This subsampling ratio corresponds to the cut-off wave number  $k_2$  in the plot of spectrum (right) for horizontal lines at the center of the channel. Other wave number  $k_1$  and  $k_3$  correspond to the subsampling ratio in space of 5 and 10 respectively.

subsampled from the original simulation at  $f = 200$ .  $U_{max}$  is the central velocity of the flow. LTHS and HTLS measurements subsampled from HTHS data are used to train different prediction models. HTHS is used as the ground truth to estimate reconstruction errors. The extension to spanwise and vertical velocity components follows the same procedure.

Various subsampling ratios both in space and time are considered. In space, the LR fields are obtained by applying a direct subsampling ratio of  $\sqrt{P/Q} = 5, 10$  or  $20$  in both spatial directions. These ratios correspond to a number  $Q$  of virtual HTLS sensors of  $51 \times 57$ ,  $26 \times 29$  and  $13 \times 15$  respectively. Each ratio has a spacing between two successive HTLS points in spanwise and vertical directions of  $\Delta z$  and  $\Delta y$ . To test the idea of combining space-time sparse measurements, these subsamplings in space are used together with subsamplings in time of ratios  $\Delta T = N/M = 4, 10$  or  $20$ , corresponding to the total number of snapshots  $M = 2500, 1000$  and  $500$  respectively. Each ratio, both in space and time, corresponds to a certain amount of kinetic energy loss. This is essentially the energy of small scales separated from the large ones by a low pass filter (LPF)  $\mathbb{L}$ . In practice  $\mathbb{L}$  will be the  $5^{th}$ -order least square spline filter, either 1D in time ( $\mathbb{L}_t$ ) or 2D in space ( $\mathbb{L}_s$ ), using measurements as knots. This spline filter has the advantage of a sharp

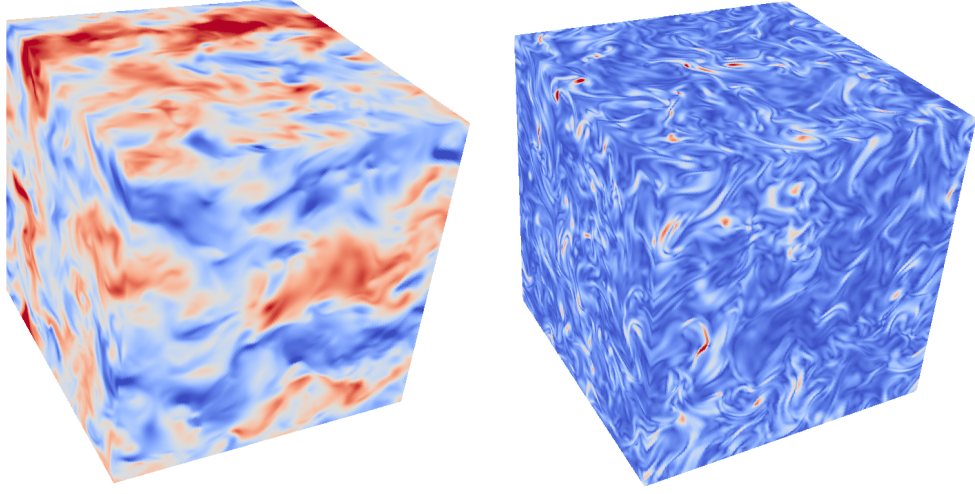


Figure 1.8 – A sample 3D streamwise velocities (left) and vortices magnitude (right) of DNS isotropic turbulence at  $384^2$ . The streamwise direction is considered as “time” dimension in this case.

cutoff response with a finite support (**de1978practical**). The energy loss is defined by comparing the filtered field  $\mathbb{L}z$  to the original field  $z$ :

$$\Delta\kappa = \frac{\sum_{j \in \mathbb{J}} z_j^2 - \sum_{j \in \mathbb{J}} [\mathbb{L}z]_j^2}{\sum_{j \in \mathbb{J}} z_j^2} \quad (1.2)$$

where  $\mathbb{J}$  is the considered set of points. Table 1.1 gathers the energy loss in time ( $\Delta\kappa_t$ ) and in space ( $\Delta\kappa_s$ ) estimated with  $\mathbb{L}_t$  and  $\mathbb{L}_s$  respectively. The set  $\mathbb{J}$  here contains all points at the center line of the channel, i.e.  $y/H = 1$ .

## 1.4.2 DNS data of isotropic turbulence

Channel flows with strong nonhomogeneity in vertical direction yields varying statistics in this direction. Numerical tests of different methods and comparisons of results are more challenging. This nonhomogeneity also calls for a very large database, since all the averaging can be done only in spanwise direction (roughly at the center of the channel) and time. The DNS database of *homogeneous isotropic turbulence* is used for simplifications yet retaining essential properties such as three-dimensionality, intermittency and multi-scale vortices. This flow is assumed to be incompressible inside a periodic box with nearly homogeneous and isotropic statistics. The data are recorded from a simulation over a  $384^3$  grid using the Fourier spectral method as originally proposed by Orszag and Patterson Jr, (1972). Figure 1.8 shows a sample of a 3D

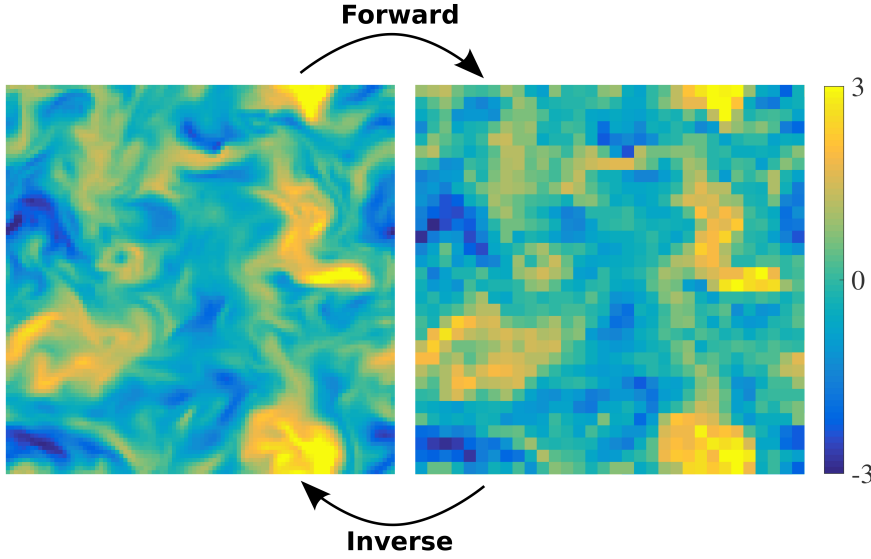


Figure 1.9 – A sample isotropic 2D streamwise velocity (left) and its subsampled field by a factor of 3 (right) on the plane normal to flow direction. The subsampling is a forward problem, while going from a subsampled to fully-resolved field is an inverse problem.

streamwise velocity and vorticity magnitude.  $Re$  is relatively low compared to the state-of-the-art  $4096^3$  simulations by (Ishihara et al., 2009; Kaneda et al., 2003). However, for the purpose of this work, the position of the cut-off frequency is the key parameter and the width of the inertial range is not crucial. Reconstruction accuracy is expected to depend more on the energy loss due to subsampling rather than on  $Re$ . Also, the accuracy depends on the slope of the energy spectra at the cut-off frequency, which represents the physics of the flow and implies the level of correlation between neighboring scales.

A total of 37 blocks of 3D fields at resolution  $384^3$  is recorded. Since a long record of time-resolved data is not available, the streamwise direction is virtually considered as the “time” dimension to homogenize the notations for the two datasets. These data are then downsampled by a factor of 4 in every dimension using an ideal Fourier filter, resulting in 3D fields on a  $96^3$  grid. The objective is to avoid working within the dissipative range where energy content is very low, leading to a too small energy loss for a given subsampling ratio. Also, the slope of the spectrum is rather high, implying very weak level of correlation between the measured scales and the ones to be reconstructed. Only a small subsampling ratio will now lead to a significant loss, therefore avoiding to work with large ratios and corresponding large numbers of uncertainty. The cubes of  $96^3$  are now considered as the reference HTHS fields. Only streamwise velocities in a plane normal to the flow direction are extracted. LR fields will be obtained by virtually subsampling from reference HR ones. A completely analogous procedure is applied to other velocity components.

	Space			Time		
Subsampling ratio	3	4	6	4	6	8
Energy loss $\Delta\kappa$ (%)	1.03	2.63	7.29	1.23	3.56	6.53

Table 1.2 – Configuration parameters of three subsampling cases in space (spanwise+vertical direction) and three in time (streamwise direction) for isotropic turbulence data. The subsampling ratios of HTLS measurements are  $\sqrt{P/Q}$  and equal in both spatial directions. The ratios of LTHS measurements in time (spanwise in this case) are  $N/M$ . The normalized energy losses in space  $\Delta\kappa_s$  and in time  $\Delta\kappa_t$  are defined in equation 1.2. The slopes are purely to qualify the steepness of the spectrum and have no physical meaning.

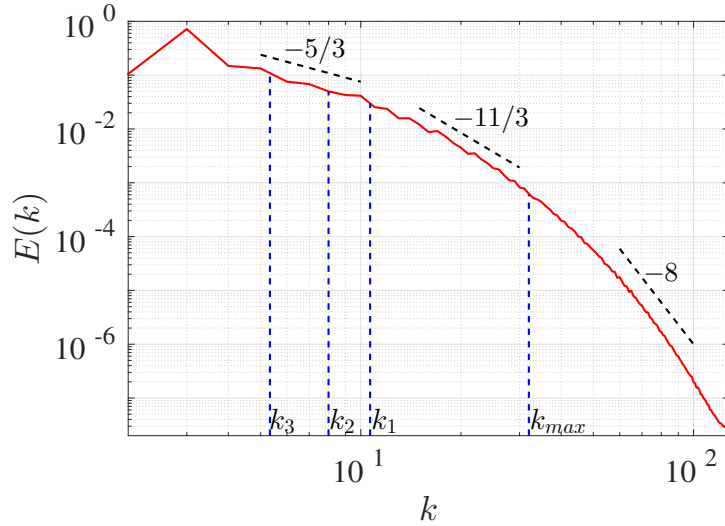


Figure 1.10 – 2D energy spectral of the reference DNS of resolution  $396^3$ . The maximum real wave number is  $k = 128$  after removing the aliasing contents. Different cutoff wave numbers are shown:  $k_{max} = 128/4$  corresponding to the pre-downsampling,  $k_1 = k_{max}/3$ ,  $k_2 = k_{max}/4$  and  $k_3 = k_{max}/6$  for subsampling ratios of  $\sqrt{P/Q} = 3, 4$  and  $6$  respectively.

Various cases with different ratios are investigated. The subsampling ratios  $\sqrt{P/Q}$  applied in each direction of space are 3, 4 and 6. These ratios correspond to a number  $Q$  of HTLS sensors of  $32 \times 32$ ,  $24 \times 24$  and  $16 \times 16$  respectively. A sampled HR field with its corresponding subsampled LR field of ratio  $3 \times 3$  is shown in figure 1.9. Subsampling ratios  $N/M$  in spanwise direction are 4, 6 and 8, corresponding to the number of training planes (or key frames) of  $M = 24$ ,  $M = 18$  and  $M = 12$ . Each ratio corresponds to a certain amount of energy loss due to small scales filtered out by an ideal Fourier filter. The energy loss is defined by comparing the filtered fields to the reference ones as in equation 1.2. The set  $\mathbf{J}$  now consists of all points thanks to isotropy and homogeneity properties. Table 1.2 gathers the energy loss due to the 2D or 1D subsamplings.

## 1.5 Concluding remarks

This chapter has briefly presented the context of the thesis, with the shortage of research tools to measure and compute small scales of turbulence. This problem call for computational methods that permit to estimate such scales from available measurements. We will tackle this problem by seeking: (i) empirical mapping functions between large and small scales, and (ii) fusion models to combine complementary measurements. A wide spectrum of methodologies will be discussed, and their performances will be analyzed using two DNS datasets of either channel flow or isotropic turbulence. Both sets give access to fully resolved fields and permit to design different numerical experiments of various subsampling ratios to obtain virtual measurements. Each configuration is characterized via the kinetic energy loss due to the subsampling.



## Velocity reconstruction using regression

This chapter presents regression models to address the problem of estimating high resolution fields from low resolution measurements as defined in section 1.3.1. It reviews the basic ordinary least squares model, regularized linear regression with the common L2 penalty as well as the L1 penalty, and kernel regression. These models are presented in matrix forms, which are more compact than previous works on turbulence. The problem of selecting model complexity and optimizing model parameters, a fundamental topic in learning theory, are also discussed.

The models are used to reconstruct high-resolution fields from low-resolution measurements. The DNS database of isotropic turbulence presented in section 1.4.2 are used for illustration. Only partial results on selecting the models and optimizing parameters are shown in this chapter. Optimal models will be used in chapter 6 when comparing performances of all proposed models.

### 2.1 Regression in turbulence studies

Regression is probably the earliest method in statistics for prediction. It aims at learning the relation between input variables and corresponding output targets. The model is learned from training samples where both input and output variables are available. the learned model will be used for prediction in situations where only input variables are given. Nice reviews can be found in Bishop, (2006) and Hastie et al., (2009a).

Least squares regression has been applied widely in turbulence studies under the name “*linear stochastic estimation*” (LSE) (Adrian, 1977, 1979). It has been further investigated (Adrian, 1992; Ewing and Citriniti, 1999; Guezennec, 1989) to estimate conditional eddies from the measurements. Later works introduced various extensions such as multi-time, nonlinear or higher-order LSE (Durgesh and Naughton, 2010; Meyer

and Tan, 2014; Mokhasi et al., 2009; Nguyen et al., 2010), which reconstruct velocity fields from pressure or shear-stress measurements. LSE can be also linked to proper orthogonal decomposition (POD), known also as “principle component analysis”, to reduce the order of reconstruction problems (Bonnet et al., 1994).

The idea of combining LR velocity measurements to obtain HR fields using regression has not been addressed until recently (Melnick and Thurow, 2012; Tu et al., 2013). Melnick and Thurow, (2012) used a POD-LSE model to get fully resolved 3D velocities of a flow over a flat plate by combining 3D smoke intensity and 2D PIV measurements. The POD-LSE model has been developed further by Tu et al., (2013) with a multi-time LSE reconstruction model. Kalman filter or Kalman smoother are used as real time estimation or data post-processing respectively. The model is tested using time-resolved PIV measurements of a bluff-body wake at a low  $Re$ .

Regression models in previous works remains rather simple, with the use of POD to reduce the order of the model. However, simple linear regression models suffer from certain limitations. The use of POD, acting as a low-pass filter, neglects certain small scales *a priori*. This work discusses further extensions of simple linear regression based on different regularizations or kernel methods. We aim at maximizing the amount of information that each method can recover, so POD is left aside throughout the thesis. Detailed analysis of reconstruction results such as spectra and errors will assess the performances of regression models as mapping functions between large and small scales.

## 2.2 Ordinary least squares (OLS)

Given a set of  $M$  training samples  $\{(\mathbf{y}_t, \mathbf{z}_t)\}, t = 1, 2, \dots, M$  corresponding to pairs of inputs  $\mathbf{y}_t \in \mathbb{R}^Q$  and outputs  $\mathbf{z}_t \in \mathbb{R}^P$ , regression models estimate  $\mathbf{z}_t$  as a function of  $\mathbf{y}_t$  via a mapping function  $f$ :

$$f : \mathbf{y}_t \mapsto \hat{\mathbf{z}}_t = f(\mathbf{y}_t) + \mathbf{n}_t \quad (2.1)$$

where  $\mathbf{n}_t \sim \mathcal{N}(0, \sigma_{\mathbf{n}_t}^2)$  is usually a white Gaussian noise.  $f$  is learned from the  $M$  pairs of training samples. This function should minimize the loss  $\mathcal{L}(f)$ , which measures how far the prediction  $f(\mathbf{y})$  and the reference  $\mathbf{z}$  are different within the training data.

The ordinary least squares model (OLS) assumes that  $f$  is an empirical linear function of  $\mathbf{y}_t$ . This model is widely used for its simplicity yet efficiency in situations where only a small training set with low signal-to-noise ratio is given. Assuming that  $\{\mathbf{z}_t\}$  and  $\{\mathbf{y}_t\}$  are mean-free and pre-normalized to one-standard deviation, OLS is expressed as:

$$\hat{\mathbf{z}}_t = f(\mathbf{y}_t) = \mathbf{B}^\top \mathbf{y}_t \quad (2.2)$$



where  $\mathbf{B}$  is the coefficient matrix of size  $Q \times P$ . The  $i$ -th column of  $\mathbf{B}$  tells how to weight each measurement  $\mathbf{y}_{t,j}$  to get an estimate of  $z_{t,i}$ . The unknown  $\mathbf{B}$  is learned from training data by minimizing the empirical loss:

$$\mathcal{L}(\mathbf{B}) = \sum_{t=1}^M \|\mathbf{z}_t - \mathbf{B}^\top \mathbf{y}_t\|_2^2 \quad (2.3)$$

or rearranged in the matrix form as:

$$\mathcal{L}(\mathbf{B}) = \|\mathbf{YB} - \mathbf{Z}\|_2^2 = (\mathbf{YB} - \mathbf{Z})^\top (\mathbf{YB} - \mathbf{Z}) \quad (2.4)$$

where  $\mathbf{Y}^\top \triangleq \{\mathbf{y}_t\}$  of size  $Q \times M$  and  $\mathbf{Z}^\top \triangleq \{\mathbf{z}_t\}$  of size  $P \times M$ .  $\|\cdot\|_2^2$  is the Euclidean distance, also called “ $L2$  norm”.  $\triangleq$  is the notation of “defined as”. The loss  $\mathcal{L}(\mathbf{B})$  is the square of errors, which is purely data-dependent. The optimization problem is to find  $\mathbf{B}$  such that:

$$\mathbf{B} = \underset{\mathbf{B}}{\operatorname{argmin}} \{\mathcal{L}(\mathbf{B})\} = \underset{\mathbf{B}}{\operatorname{argmin}} \{\|\mathbf{YB} - \mathbf{Z}\|_2^2\} \quad (2.5)$$

$\operatorname{argmin}\{\cdot\}$  is the argument of the minimum seeking  $\mathbf{B}$  such that  $\mathcal{L}(\mathbf{B})$  attains its minimum. The least square error solution of  $\mathbf{B}$  is by differentiating equation 2.4 with respect to  $\mathbf{B}$  and set to zero:

$$\mathbf{B} = \mathbf{Y}^\dagger \mathbf{Z} = (\mathbf{Y}^\top \mathbf{Y})^{-1} \mathbf{Y}^\top \mathbf{Z} \quad (2.6)$$

where  $(\cdot)^\dagger$  is Moore-Penrose pseudo inverse. Then, the output  $\mathbf{z}^e$  of any new input vector  $\mathbf{y}^e$  out of the training set is estimated as:

$$\mathbf{z}^e = \mathbf{B}^\top \mathbf{y}^e \quad (2.7)$$

## 2.3 Regularized linear regression

OLS suffers from critical problems. It requires the inverse of matrix  $\mathbf{Y}^\top \mathbf{Y}$  that can be singular or almost singular. The iterative solver can overcome this problem, but may result in a high-variance model with many large coefficients: a small change of  $\mathbf{y}^e$  leads to very different predictions of  $\mathbf{z}^e$ . Since learned purely from the training data, it often fits very well the training data but poorly performs otherwise. This phenomena is called *overfitting* and will be discussed later in this chapter. Regularized least squares are proposed to overcome this problem. The idea is to add a regularization term  $g(\mathbf{B})$  on  $\mathbf{B}$

to the data-dependent error term:

$$\mathcal{L}(\mathbf{B}, \lambda) = \|\mathbf{YB} - \mathbf{Z}\|_2^2 + \lambda g(\mathbf{B}) \quad (2.8)$$

where the regularization parameter  $\lambda$  controls the balance between the two terms. Different terms lead to different regression models.

### 2.3.1 Ridge Regression (RR): L2 penalty

The most common and simple regularization is L2 penalty, also called “weight decay”, which leads to the formulation of ridge regression (RR). The regularization term is on the norm of OLS coefficients, i.e.  $g(\mathbf{B}) = \|\mathbf{B}\|_2^2$ . The loss function becomes:

$$\mathcal{L}(\mathbf{B}, \lambda) = \|\mathbf{YB} - \mathbf{Z}\|_2^2 + \lambda \|\mathbf{B}\|_2^2 \quad (2.9)$$

The parameter  $\lambda$  controls the balance between the data misfit and regularization term, which is the sum of squares of all coefficients. It controls how far  $\mathbf{B}$  are shrunk towards zero, the larger  $\lambda$  the further. Setting the derivative with respect to  $\mathbf{B}$  to zero, the closed form to estimate  $\mathbf{B}$  is:

$$\mathbf{B} = (\mathbf{Y}^T \mathbf{Y} + \lambda \mathbf{I})^{-1} \mathbf{Y}^T \mathbf{Z} \quad (2.10)$$

This formula is similar to that of OLS except that some positive value  $\lambda$  is added to the diagonal elements of  $\mathbf{Y}^T \mathbf{Y}$  to ensure  $\mathbf{Y}^T \mathbf{Y} + \lambda \mathbf{I}$  is always invertible.

The mechanism of RR can be analyzed via the singular value decomposition (SVD) of  $\mathbf{Y}$ :

$$\mathbf{Y} = \mathbf{U} \mathbf{D} \mathbf{V}^T \quad (2.11)$$

where  $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_Q)$  is a  $M \times Q$  orthogonal matrix,  $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_Q)$  is a  $Q \times Q$  diagonal matrix ( $d_1 \geq d_2 \geq \dots \geq d_Q$ ), and  $\mathbf{V}^T = (\mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_Q^T)$  is a  $Q \times Q$  orthogonal matrix. The orthogonality implies that  $\mathbf{U}^T \mathbf{U} = \mathbf{I}$  or  $\mathbf{U}^T = \mathbf{U}^{-1}$ , similarly for  $\mathbf{V}$  and  $\mathbf{V}^T$ . The formula to estimate RR coefficients becomes:

$$\mathbf{B} = (\mathbf{Y}^T \mathbf{Y} + \lambda \mathbf{I})^{-1} \mathbf{Y}^T \mathbf{Z} = \mathbf{V} \text{diag} \left( \frac{d_j^2}{d_j^2 + \lambda} \right) \mathbf{U}^T \mathbf{Z} \quad (2.12)$$

Re-estimating training output variables  $\mathbf{Z}$  as a function of inputs, one obtains:

$$\hat{\mathbf{Z}} = \mathbf{YB} = \sum_{j=1}^Q \left( \mathbf{u}_j \frac{d_j^2}{d_j^2 + \lambda} \mathbf{u}_j^T \right) \mathbf{Z} \quad (2.13)$$

**Algorithm 1** Iterative shrinkage-thresholding algorithm (ISTA)

---

```

1: Set  $k=0$  and initialize  $\mathbf{B}^{(0)}$ ;
2: while not convergence do
3:    $\nabla f(\mathbf{B}^{(k)}) = \mathbf{Y}^\top(\mathbf{Z} - \mathbf{Y}\mathbf{B}^{(k)})$  ▷ Residual from step k
4:    $\mathbf{B}^{(k+1)} \leftarrow \text{prox}_{\lambda\|\cdot\|_1}[\mathbf{B}^{(k)} - t^{(k)} \nabla f(\mathbf{B}^{(k)})]$ 
5:    $k = k + 1$ 
6: end while
7: return  $\mathbf{B}^{(k+1)}$ 

```

---

This implies that RR projects  $\mathbf{Z}$  onto the principal components of  $\mathbf{Y}^\top\mathbf{Y}$  with large energy content (large  $d_j$ ) and shrinks the coefficients of low energy (small  $d_j$ ). This makes RR close to the *principal component regression* model discussed in Jolliffe, (1982), where one decomposes  $\mathbf{Y}^\top\mathbf{Y}$  using SVD and then set components with low energy content (small  $d_j$ ) to zeros in a handy manner. The approach also avoids matrix inversion and plays a role similar to regularization.

### 2.3.2 LASSO: L1 penalty

The nature of L2 penalty is to reduce model variance by shrinking coefficients corresponding to irrelevant events toward zero. However, they are not exactly zero. L1 penalty will precisely force some coefficients to zero. The model is called least absolute selection and shrinkage operator (LASSO) (Tibshirani, 1996) in statistics, or basis pursuit denoising (BPDN) (Chen et al., 1998) in signal processing. It is considered as an implicit subset selection step, where irrelevant events are neglected from the reconstruction. This property favors sparsity - output is estimated as a combination of some input variables only- that is beneficial in many applications.

The LASSO cost function is similar to that of RR with a subtle but very important modification:

$$F(\mathbf{B}) = \|\mathbf{Y}\mathbf{B} - \mathbf{Z}\|_2^2 + \lambda\|\mathbf{B}\|_1 \quad (2.14)$$

The data-dependent term is the same as OLS or RR. The different penalty term  $\lambda\|\mathbf{B}\|_1$  imposes a constraint on the sum of absolute values of coefficients. This modification leads to sparsity, the key difference between LASSO and RR. It also makes the problem nonlinear and there exists no closed-form solution. Gradient-based methods are usually used to solve this optimization problem. Iterative shrinkage-thresholding algorithms (ISTA) is one of them, where  $\mathbf{B}$  is solved iteratively by *soft-thresholding* as in Algorithm 1 (Daubechies et al., 2004). In the pseudo code, *prox*- the shrinkage operator- is the

element-wise soft-thresholding:

$$\left(\text{prox}_{\lambda\|\cdot\|_1}[\mathbf{B}]\right)_i = \begin{cases} b_i - \lambda \text{sign}(b_i) & \text{if } |b_i| > \lambda, \\ 0 & \text{otherwise} \end{cases} \quad (2.15)$$

where  $t^{(k)}$  is the gradient step size,  $b_i$  is the  $i$ -th coefficient of  $\mathbf{B}$ . The *sign function*  $\text{sign}(b_i)$  is defined as:

$$\text{sign}(b_i) = \begin{cases} -1 & \text{if } b_i < 0, \\ 0 & \text{if } b_i = 0, \\ 1 & \text{if } b_i > 0. \end{cases} \quad (2.16)$$

Faster schemes have been proposed to accelerate the convergence time (Beck and Teboulle, 2009; Vonesch and Unser, 2008), but further discussions are outside the scope of this work.

LASSO often outperforms RR and subset selection methods for several reasons. Compared to RR, LASSO takes most of the advantages, including the stability of the solution and the shrinkage feature. These features give a low-variance model compared to subset selection method. Moreover, LASSO favors sparsity, which can be considered as an implicit subset selection (Hastie et al., 2005, 2009b). By forcing some of the coefficients to zeros, the irrelevant predictors are suppressed in final estimation.

## 2.4 Nonlinear regression

Linear regression aims at mapping the output as a linear combination of input variables. This implies a real constraint on performances of this family of models. In many cases including turbulence, linear functions are too simple to describe the undergoing phenomenon. Nonlinear regression models are beneficial, and kernel feature mapping is often used.

### 2.4.1 Feature mapping

Kernel methods are used to introduce nonlinearity into the model. The idea is to project the original input vector  $\mathbf{y}$  onto a fixed feature space:

$$\mathbf{y}_t \mapsto \boldsymbol{\phi}_t = \varphi(\mathbf{y}_t) \quad (2.17)$$

and perform least square regression in this space.  $\mathbf{y}_t \in \mathbb{R}^Q$  is the  $t$ -input vector. The feature vector  $\boldsymbol{\phi}_t \in \mathbb{R}^{D_f}$ , where  $D_f \gg Q$  is the dimension in feature space. The least

square problem becomes:

$$\mathbf{B} = \underset{B}{\operatorname{argmin}} \left\{ \|\mathbf{Z} - \Phi \mathbf{B}\|_2^2 \right\} \quad (2.18)$$

where  $\Phi = \{\phi_t\}$  ( $t = 1, \dots, M$ ), the so-called *design matrix*, is of size  $M \times D_f$ . Adding L2 regularization term  $\|\mathbf{B}\|_2^2$  and deriving analogously as RR model, the solution is:

$$\mathbf{B} = (\Phi^\top \Phi + \lambda \mathbf{I})^{-1} \Phi^\top \mathbf{Z} \quad (2.19)$$

Then the prediction of a new input variable  $\mathbf{y}^e$  is:

$$\mathbf{z}^e = \mathbf{B}^\top \varphi(\mathbf{y}^e) \quad (2.20)$$

## 2.4.2 Kernel ridge regression

Involving nonlinearity via feature mapping significantly increases computational costs. The kernel trick is proposed (Saunders et al., 1998) to overcome this problem, resulting in the so-called kernel ridge regression (KRR) model. This trick appears when solving ridge regression using Lagrange dual optimization.

### Dual form of ridge regression

RR in equation 2.9 can be re-expressed as a dual Lagrangian optimization problem:

$$\mathbf{B} = \underset{B}{\operatorname{argmin}} \left\{ \sum_{t=1}^M \|\mathbf{e}_t\|_2^2 + \lambda \|\mathbf{B}\|_2^2 \right\} \quad s.t. \quad \mathbf{z}_t - \mathbf{B}^\top \mathbf{y}_t = \mathbf{e}_t, \quad t = 1, 2, \dots, M \quad (2.21)$$

where *s.t.* stands for “*subject to*”. Introducing Lagrange multipliers  $\mathbf{A}[M \times P] \triangleq \{\mathbf{a}_t^\top\} (t = 1, 2, \dots, M)$ ,  $\mathbf{a}_t \in \mathbb{R}^P$ , the optimization problem 2.21 is equivalent to the problem of finding the saddle point of the function:

$$\sum_{t=1}^M \|\mathbf{e}_t\|_2^2 + \lambda \|\mathbf{B}\|_2^2 + \sum_{t=1}^M \mathbf{a}_t^\top (\mathbf{z}_t - \mathbf{B}^\top \mathbf{y}_t - \mathbf{e}_t) \quad (2.22)$$

The solution as shown in Saunders et al., (1998) is:

$$\mathbf{A} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{Z} \quad (2.23)$$

Function name	$k(\mathbf{u}, \mathbf{v})$
Linear	$\mathbf{u}^\top \mathbf{v}$
Polynomial	$(r + \mathbf{u}^\top \mathbf{v})^d$ for $r, d \geq 0$
Radial basis function (RBF)	$\exp(-\gamma \ \mathbf{u} - \mathbf{v}\ _2^2)$ , $\gamma > 0$

Table 2.1 – Common basis functions for kernel methods.

where  $\mathbf{K} \triangleq \mathbf{Y}\mathbf{Y}^\top$  is the matrix of dot products,  $\mathbf{K}_{m,n} = \mathbf{y}_m^\top \mathbf{y}_n$ . The prediction for a new input variables  $\mathbf{y}^e$  is:

$$\hat{\mathbf{z}}^e = \left( \sum_{t=1}^M \mathbf{a}_t^\top \mathbf{y}_t \right) \mathbf{y}^e = \mathbf{A}^\top \mathbf{k}^e \quad (2.24)$$

where  $\mathbf{k}^e \triangleq \{k_t\} = \{\mathbf{y}_t^\top \mathbf{y}^e\} \in \mathbb{R}^M$ .

### Kernel trick

Solving RR in the dual form offers no improvement of accuracy, speed or stability. However, the beauty of this approach is the presence of only dot products among input variables in the final prediction step. This implies an analogous procedure when working in a nonlinear feature space. The transformation into such a space is unnecessary if the dot products of the transformed variables can be estimated using the kernel:

$$k(\mathbf{u}, \mathbf{v}) = \varphi(\mathbf{u})^\top \varphi(\mathbf{v}) \quad (2.25)$$

This explicit transformation is computed in time  $O(Q^2)$ , where  $Q$  is the dimension of the input vectors. The *kernel trick* permits to estimate directly  $k(\mathbf{u}, \mathbf{v})$  without computing  $\varphi(\mathbf{u})$  and  $\varphi(\mathbf{v})$ , reducing the computational time into  $O(Q)$  only. Not every kernel has this property. Table 2.1 gathers three common functions.

## 2.5 A framework to select model and parameters

A model is assessed via its generalization performance, i.e. the prediction capability on a new dataset independent from the training set. In all above models, there exists one or more hyper-parameters that are directly linked to model performances. The arising question is how to optimize those parameters using training data only in a systematic manner. This step is the so-called *parameter optimization*, while the step to test performances of the model on independent data is the so-called *model assessment*.

It is worth also mentioning the definition of different datasets: *training*, *validation*

and *test* sets. The training set contains data from which a model is learned. The validation set is used to estimate prediction errors, and from which the best model is chosen. This model is the one that gives the most accurate prediction on the validation set. The testing set is finally used to give an estimate of the prediction error of this optimal model. This error is approximately the generalization error on independent datasets.

To optimize the generalization capability of a model, one needs to understand the idea of *bias* and *variance*. Next sections will discuss this topic, with the technique called “*cross-validation*” to find the trade-off between bias and variance using the training data only.

### 2.5.1 Bias-variance trade-off

The above regression models can be interpreted as seeking a mapping function  $f$ :

$$f : \mathbf{y} \mapsto \hat{\mathbf{z}} = f(\mathbf{y}) + \mathbf{n} \quad (2.26)$$

where  $\mathbf{n} \sim \mathcal{N}(0, \sigma_n^2)$  is assumed to be a white Gaussian noise and be conditionally independent of  $\mathbf{z}$ . The expected prediction error of this model for an input vector  $\mathbf{y}_o$  is:

$$\begin{aligned} \epsilon(\mathbf{y}_o) &= \mathbb{E}[(z_o - f(\mathbf{y}_o))^2] \\ &= \underbrace{\sigma_n^2}_{\text{Irreducible Error}} + \underbrace{(\mathbb{E}[\hat{z}_o] - z_o)^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}[(\hat{z}_o - \mathbb{E}[\hat{z}_o])^2]}_{\text{Variance}} \end{aligned} \quad (2.27)$$

where  $\mathbb{E}[\cdot]$  is the expectation of a variable. The first term comes from the noise, which depends only on the data and is irreducible. The second term is the square bias, showing how far the average of the estimates is different from the true mean. The last term is the expected variance of the estimate around its mean. We can interpret the bias as the average prediction error over different data sets from the true mean, and the variance as how this error is sensitive to a particular choice of data set.

After the decomposition, a model is said to be *underfitting* or *overfitting* depending on the contribution of each term in the total error. An underfitting model is too simple to capture all details of the underlying phenomenon. This model is high-bias and low-variance: it poorly fits the training data and performs similarly in the testing data. An overfitting model is over-complex: it closely fits the training data, including noises, and gives poor performance on new testing data. This model has low bias and high variance. Ideal models should have both low bias and variance. In practice, a good model must satisfy its *bias-variance trade-off* when it does not suffer from either the problems of overfitting or underfitting.

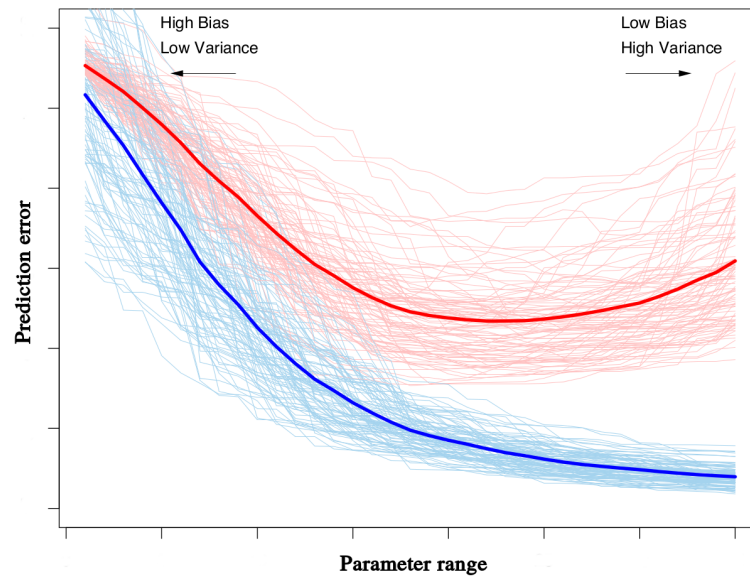


Figure 2.1 – Typical behavior of prediction error for testing (red) and training (blue) dataset as a function of a hyper-parameter (Hastie et al., 2009a). Different curves are for various datasets. The solid ones are expected errors (average of all curves). From left to right can be the direction of decreasing regularization or increasing model complexity.

The idea of underfitting and overfitting can be visualized in figure 2.1 as modified from Hastie et al., (2009a). It shows the prediction errors for training and testing datasets as a function of a parameter. This parameter can be either the model complexity in an increasing order or the regularization parameter  $\lambda$  in a decreasing order. Models toward the left (low complexity model, or regression with a strong regularization) are underfitting. It does not fit well both the training and testing data. Toward the right, errors on the training set decrease, while those of the testing set decrease and then increase again. Models in the far right are overfitting. They fit well the training set but not the testing set. The desired model is the one such that the error on the testing set is the lowest. At this point, the model reaches its bias-variance trade-off: bias and variance do not necessarily reach their minima, but a compromise ensures the minimal error on testing data.

The idea of bias-variance trade-off can be further visualized schematically as in figure 2.2 (Hastie et al., 2009a). The blue-shaded region shows the irreducible error  $\sigma_n^2$  (due to random noise) of the training set, where the truth is at the center and all realizations are within. To fit the model, the non-regularized model space (red curve) contains all possible models, while the magenta one shows the restricted space of regularized models. Model variances are depicted as yellow curves centered at the so-called “closest fit”. The model bias is the distance of the closest fit and the truth, which can only be reduced as increasing the model complexity or adding more features. Regularized models add



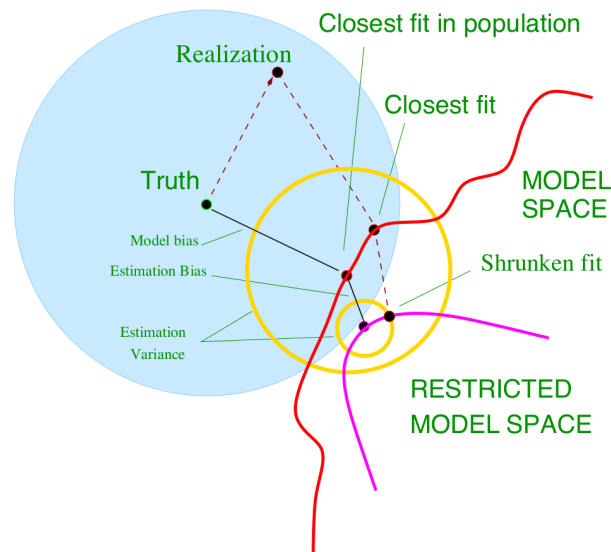


Figure 2.2 – Schematic view of the behavior of bias and variance (Hastie et al., 2009a). The blue-shaded, centered at the truth, is the realization space, where maximum distance is the irreducible error. All possible models are bounded by the red/magenta model space curves. Bias is shown by the black lines as the distance to the truth. Best models are shown as black dots, and circled by model variances.

estimation bias on top of the model bias but reduce the variance. The benefit of the trade-off is only when the variance is reduced more than the squared estimation bias.

The bias-variance trade-off is essentially estimated from the training data. If this set is large enough, it can be virtually divided into training, cross-validation and testing sets. A general advice for the size of these three sets is 50%, 25% and 25% respectively (Hastie et al., 2009a). However, learned models are usually improved when using more data. Cross-validation is another idea to find the trade-off while keeping the whole training data for learning the model.

### 2.5.2 Cross-validation

In many cases, the training data is limited. Setting aside 50% the data to quantify the model might lose all the benefits from regularization and nonlinearity compared to OLS. Cross validation (CV) is commonly used to select the model and optimize parameters using all given data.

The most popular CV technique is *k-fold CV*. The dataset is randomly split into  $K$  subsets of approximately equal size. For each  $K$ -th subset to estimate the prediction error, the model is trained on the remaining  $K - 1$  subsets. This procedure is repeated  $K$  times for all the subset, and the model error for the current setting is the average of its  $K$  estimates. A summary of *k-fold CV* is presented in algorithm 2.  $K$  is at most the number

---

**Algorithm 2** K-fold cross-validation for set of L parameters  $\gamma_1, \gamma_2, \dots, \gamma_L$

---

- 1: Divide the training samples into k folds randomly
  - 2: **for**  $i = 1, 2, \dots, L$  **do**
  - 3:     **for**  $j = 1, 2, \dots, K$  **do**
  - 4:         Train the model with  $\gamma_i$  using all data set except the  $j^{th}$
  - 5:         Estimate prediction error  $\epsilon(i, j)$  of the model on the  $j^{th}$  set
  - 6:     **end for**
  - 7: **end for**
  - 8: Estimate averaged error  $\epsilon(i) = \frac{1}{K} \sum_{j=1}^K \epsilon(i, j)$
  - 9: Return optimal parameter  $\gamma_m$  where  $\epsilon(m) = \min\{\epsilon(i)\}$ .
- 

of training samples (*leave one out cross validation*). When  $K$  is high, the selected model tends to have lower bias but higher variance, since many training samples are similar. A much heavier computation is also required, since the training/predicting is repeated  $K$  times. The popular choice of  $K$  is from 5 to 10 (Breiman and Spector, 1992; Kohavi et al., 1995).

## 2.6 Regression models for reconstruction of isotropic turbulence

We now apply regression models to reconstruct HR velocity fields from LR measurements. The data from DNS isotropic turbulence discussed in section 1.4.2 is used. Streamwise velocities from 37 data cubes of fully resolved data  $96^3$  are used as the reference. In each data cube, only the streamwise velocity component is considered. The training examples are the HR fields (in spanwise-vertical directions) and equivalent LR ones subsampled by a factor of  $\sqrt{P/Q} = 3$  in both vertical and spanwise directions. This ratio is equivalent to an energy loss of  $\Delta\kappa_s = 1.03\%$  (see table 1.2). In streamwise direction, the training planes are selected every 4 snapshots, corresponding to an energy loss of 1.23%. This configuration is chosen to mimic other problems investigated later in chapters 4 and 5.

Regression models are learned from  $37 \times 24$  pairs of LR ( $Q = 32 \times 32$ ) and corresponding HR ( $P = 96 \times 96$ ) fields. The learned models are then used to reconstruct all  $37 \times 96$  HR fields from  $37 \times 96$  measured LR planes. These results are presented later in chapter 6 when comparing various methods. Following sections only discuss the optimization of model parameters.

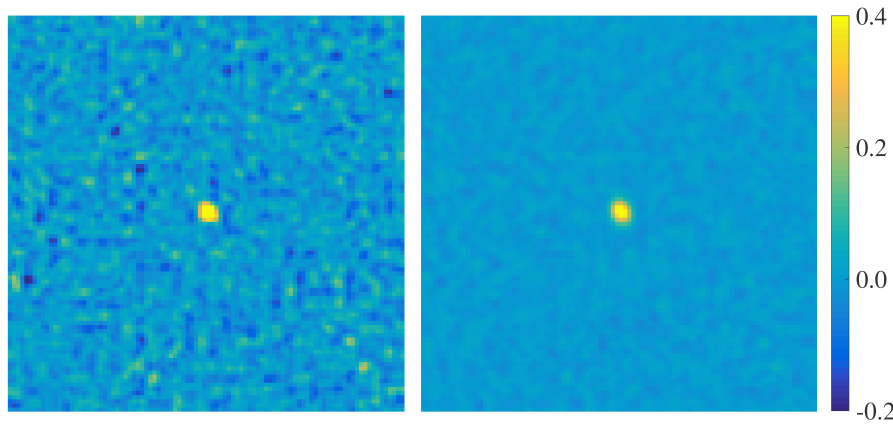


Figure 2.3 – Shrinkage effect: coefficients of OLS (left) vs RR (right) models correspond to an input LR measurement at the center of the field. The coefficients, which are rearranged in a 2D field of the size  $96 \times 96$ , can be interpreted as the contributions of this input to the reconstruction of all other high-resolution points. The higher the coefficient, the stronger the impact.

### 2.6.1 Regularization parameter and shrinkage effect

Ordinary least squares model inverts directly  $\mathbf{Y}^T \mathbf{Y}$ , which is usually ill-conditioned, leading to high variance models. A slight change of input variables can lead to very different predictions. Regularization is introduced to reduce this effect by imposing different penalty terms on regression coefficients. RR seeks a model with small sum-of-square coefficients, LASSO imposes a penalty of their absolute values (see section 2.3).

Figure 2.3 shows the weight of the input variable at the center of the field in reconstructing all other HR points by OLS and RR. These are the coefficients  $\mathbf{b}_j \triangleq [b_{j1}, b_{j2}, \dots, b_{jP}]^T$  computed in equation 2.6 for OLS or equation 2.10 for RR. Recall that  $\mathbf{B}$  is of size  $Q \times P$ . The figures are  $\mathbf{b}_j$  of OLS and RR models, where index  $j$  corresponds to the central point.  $\mathbf{b}_j$  is then rearranged to recover the 2D shape of size  $\sqrt{P} \times \sqrt{P}$  of the 2D velocity fields. In both plots, the weights remain significant in a neighborhood of several pixels. The coefficients drop very quickly when moving away from the center. This is due to the rapid decay of correlation between the central point and its neighbors. The difference between the two methods is that OLS (using direct inversion) gives large coefficients almost everywhere whereas RR shrinks the contributions of irrelevant points to small values.

The different shrinkage behaviors of RR and LASSO are visualized in figure 2.4. This figure shows the eight different coefficients corresponding to eight HR outputs as a function of the regularization parameter presented in equation 2.8. It confirms the effect of the penalty term, which shrinks coefficients associated with irrelevant inputs to zeros.

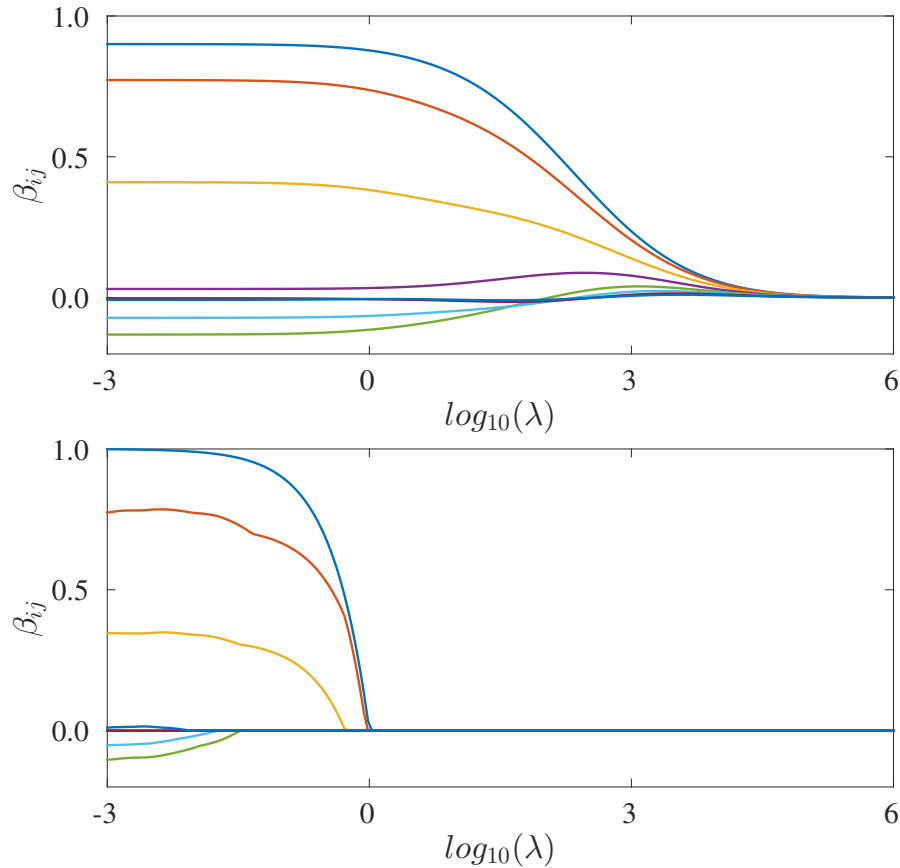


Figure 2.4 – Shrinkage effects of L2 (top) and L1 (bottom) penalty: coefficients corresponding to eight high-resolution outputs (in eight different colors) as functions of regularization parameters  $\lambda$ . Higher  $\lambda$  shrink the coefficients toward zeros differently depending on the penalty terms.

RR coefficients do not reach exact zeros even for extremely high  $\lambda$ , while LASSO rapidly shrinks some coefficients to exact zeros.

## 2.6.2 Optimizing regularization parameters and model complexity via ten-fold cross-validation

All regression models (except OLS) are parametric, i.e. at least one hyper-parameter controls the construction of the model and its performances. With RR, LASSO or KRR, the regularization parameter  $\lambda$  controls the balance between data misfit and regularization term. KRR model is also characterized by kernel parameters, which are the standard deviation of the distribution when using RBF kernel, or the polynomial order. All hyper-parameters are chosen from the training samples using ten-fold CV (see algorithm 2). The plot of the error as a function of model parameters is called the “validation curve”, from which the best model or parameter is selected.

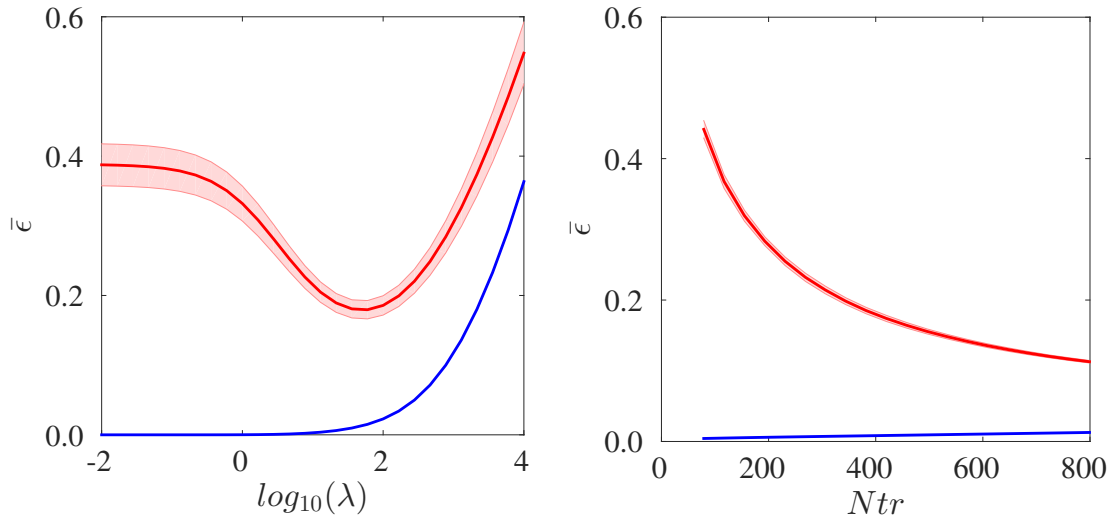


Figure 2.5 – RR validation curve: errors as functions of regularization parameter  $\lambda$  (left), and learning curve: errors as functions of training data size (right). Red curves are for the prediction, while blue ones are for the training data

Another important test is to see whether the training dataset is sufficient or not. This is examined via the so-called “learning curve”. To estimate this curve, a small subset of data is drawn out randomly from the training set. This set is retained to estimate the error at the last step. For the remaining data, models are optimized using different portions. Trained models are assessed by computing prediction errors on the testing data. The curve of errors as a function of different data size, the “learning curve”, will tell whether the current training data is sufficiently large to guarantee an accurate learning.

To plot validation and learning curves, the average normalized root mean square error (NRMSE)  $\bar{\epsilon}$  is used as the prediction error.  $\bar{\epsilon}$  represents an overall measure of the reconstruction accuracy. NRMSE is estimated between the reconstructed ( $\hat{z}_t$ ) and reference ( $z_t$ ) fields

$$\bar{\epsilon} = \left( \frac{\sum_t \sum_{j \in \mathbb{J}} (\hat{z}_{t,j} - z_{t,j})^2}{\sum_t \sum_{j \in \mathbb{J}} z_{t,j}^2} \right)^{1/2} \quad (2.28)$$

$\mathbb{J}$  is the considered set of points used to estimate the error. In the present case of isotropic turbulence,  $\mathbb{J}$  contains all points in each plane.

Figure 2.5 shows the validation and learning curve of RR model on the training set (blue) and validation set (red). The solid lines are the means, while a band of colors show the standard deviations of ten estimates of  $\bar{\epsilon}$  from ten folds. The validation curve (left) shows the model behavior for different  $\lambda$ . For the training set, the model fits accurately the data when no or small regularization is imposed. This effect of overfitting is shown on the validation set, where higher errors are obtained. The model has also high variance, i.e.

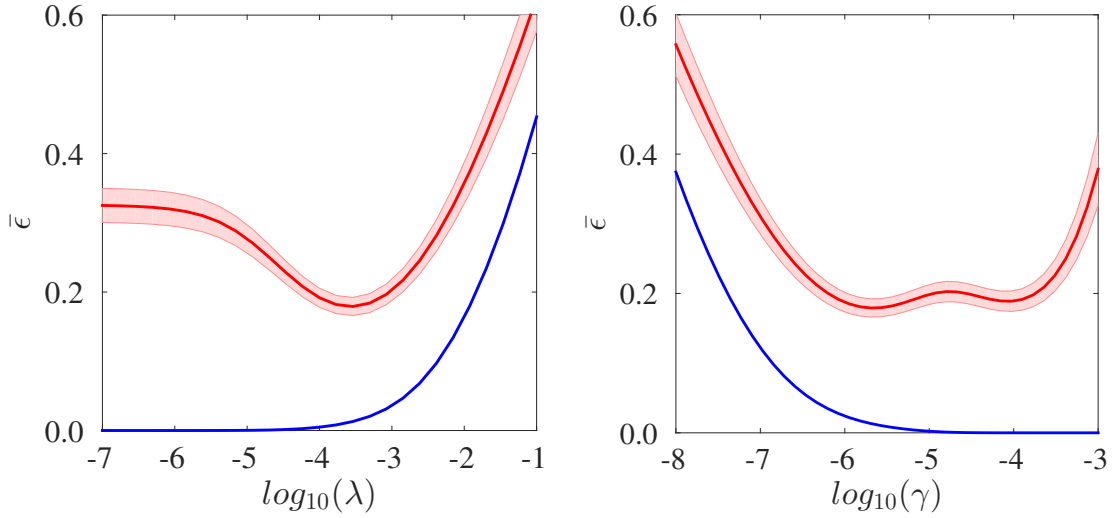


Figure 2.6 – KRR validation curve (RBF kernel), with parameters are kernel parameter  $\gamma$  and regularization parameter  $\lambda$ . Red curves are for the prediction, while blue ones are for the training data

prediction errors vary strongly from one set to the other. The optimal value of  $\lambda \approx 100$  is reached where the validation error is minimum. When a very strong regularization is imposed, errors on both training and validation sets increase, since all coefficients are shrunk toward zero. Errors will grow till their maximum value, which is the variance of the HR data. The learning curve (right) shows the effect of the training data size on the performance of the model (with optimal  $\lambda$ ). The figure shows that with larger training data, the prediction error on the validation set decrease sharply while this error on the training set gradually increase. With a good model and sufficiently large training data, these two curves will be very close. The curves also suggest that a larger dataset will lead to a better model in this case.

KRR model parameters are optimized using the same validation curve. Different models come with different kernel parameters (recall table 2.1), together with the regularization parameter  $\lambda$ . For RBF kernel,  $\gamma$  represents the kernel shape. Figure 2.6 shows that the optimal values of both  $\lambda$  and  $\gamma$  are both of the order of  $10^{-5}$ . For polynomial kernel, the polynomial order is to be tuned. The model with higher polynomial order will have lower bias but higher variance. An analogous procedure is used to find the optimal order of the polynomial fit, which is either 2 or 3 in general.

## 2.7 Concluding remarks

In this chapter, regression models are used to reconstruct high resolution fields of an isotropic turbulence from low resolution measurements. The models seek a mapping

function between low and high resolved fields. This function is learned from given training samples and used for prediction when only low-resolution measurements are available.

Both linear and nonlinear models have been discussed. The ordinary least squares model is the simplest one and is widely used in the literature of turbulence. Since suffering from some critical limitations, notably overfitting and ill-conditioning, various regularization methods have been introduced. Both L2 and L1 penalty have been discussed. While L2 penalty reduces model variance by shrinking coefficients of irrelevant events, L1 forces them to zeros by slightly change the penalty term and offers some potential benefits. Nonlinear regression is another approach to improve OLS. Instead of assuming the underlying phenomenon is linear, it introduces nonlinearity by projecting input vectors to a fixed feature space before performing standard OLS. This nonlinearity helps in improving reconstruction accuracy, since assuming a linear relation between large and small scales of turbulence implies a big constraint on model performances.

We have presented the procedure to optimize hyper-parameters of regression models. This step is crucial to ensure accurate predictions, since performances of all regression models (except OLS) depend on at least one parameter. The models are learned from the training data, but the performances are assessed on independent data only. Cross-validation is used to seek a compromise, the so-called bias-variance trade-off, from training data only. The dataset is virtually split into many subsets. The models are trained within some subsets, and their prediction generalized errors are estimated using the remaining sets.

The optimal set of parameters is found using the validation curve and ten-fold cross validation technique. This curve shows the prediction errors on the validation set as a function of model parameters. It gives an idea of the optimal parameter range. Through all the tests, it is shown that the gradient of this curve near its minima is not very sharp, implying that a good estimate of these parameters can be achieved from the training data only.

All model parameters are investigated. RR model is optimal when its regularization parameter  $\lambda$  is about 100. KRR model with RBF kernel is constructed with the optimal values of both  $\lambda$  and  $\gamma$  are of the order of  $10^{-5}$ . For polynomial KRR takes 2 or 3 as the optimal order of the kernel. All such models will be used later in chapter 6 to reconstruct high-resolution fields from low-resolution measurements and compared to other proposed methods.





## Learning coupled dictionaries

This chapter describes a novel approach to learn the relation between large and small scales of turbulence based on *dictionary learning* methods. The approach finds coupled dictionaries to represent the low and high resolution turbulent fields and to reconstruct the missing small-scale information. This method has been applied successfully in image processing, and remains a very active research subject. It potentially outperforms regression models, which are sometimes too simplistic to represent the underlying phenomenon in turbulence. Indeed learned dictionaries could encode part of the physics of the flow.

This chapter is organized as follows. First, different data representations are discussed, from predefined bases such as Fourier or wavelets to learned bases such as principal component analysis (PCA). Second, dictionary learning, a generalization of PCA, is discussed for the first time as a representation for turbulent fields. This section reviews also two algorithms to learn the dictionaries. Third, different approaches to learn coupled dictionaries of low and high resolution fields are presented. Last, the approach is tested on the DNS database of an isotropic turbulence presented in section 1.4.2.

### 3.1 From bases to dictionaries

This section briefly reviews conventional representations of turbulent signals. Given a vector  $\mathbf{x}_t$ , the main idea of all representation methods is to find a “*dictionary*”  $\mathbf{D}$ , a set of basis functions or the so-called “*atoms*” from which  $\mathbf{x}_t$  can be represented via a linear combination. This representation by coefficients  $\mathbf{a}_t$  is called a “*code*”. The representation can be exact:

$$\mathbf{x}_t = \mathbf{D}\mathbf{a}_t \tag{3.1}$$

or approximate:

$$\mathbf{x}_t = \mathbf{D}\mathbf{a}_t + \mathbf{n}_t \quad (3.2)$$

where  $\mathbf{n}_t$  is an estimation noise term. Seeking the representation of the signal is an optimization problem to estimate the dictionary  $\mathbf{D}$  and the code  $\mathbf{a}_t$ . The dictionary can be mathematically predefined or learned from the data. The estimation of the coefficients  $\mathbf{a}_t$  can be very simple with very fast algorithms for predefined or orthogonal bases, or more complex with redundant dictionaries.

### 3.1.1 Predefined dictionaries

Various types of dictionaries are studied to represent the data. They can be predefined *a priori* such as the well-known Fourier transform, the group of *wavelets* or *curvelets* (Mallat, 1999; Mallat, 1989). These predefined transforms are easy to compute thanks to fast algorithms. Fourier transform aims at describing the signal via its frequency content by decomposing signals as an infinite series of sines and cosines. This transformation is localized in frequency but not in space/time. The constraint is overcome using wavelet transforms, replacing the sines and cosines by wavelet functions localized both in space/time and frequency. This property permits wavelets to better represent the signal with discontinuities or sharp spikes. In such cases, the representations are much more compact with wavelet functions than sines and cosines. This makes wavelets standard in signal compression, for example JPEG 2000 for images, while Fourier transform is mainly used for spectral analysis. However, despite the large family of wavelets, all kinds of signals cannot be represented with high level of sparsity. Analogously for turbulence data, different flows or a single flow at various positions with respect to a wall can have significantly different physics, leading to signals with completely different properties.

### 3.1.2 Adaptive dictionaries

To further optimizing the representation, it is more appealing to learn dictionaries adaptively from the data (Bengio et al., 2013). The first and most common method is the *principal component analysis* (PCA). This approach learns a basis that maximizes the variance of the projected values. This is interesting for turbulence studies, since large scales of higher variances are of interest. This representation imposes orthogonal bases, which implies that the number of atoms is limited by the dimension of the input vectors. PCA is potentially subject to certain limitations due to this constraint. This chapter discusses the generalized version of PCA, which permits to learn a “*redundant*” dictionary. The method is called *dictionary learning*, which is widely used and remains an active

research topic in image processing. It will be discussed further in this section before applying to the problem of turbulent field reconstruction.

Adaptive dictionaries are useful in many applications (Tošić and Frossard, 2011). The first one is *dimension reduction* (Burges, 2010), which is useful to build models or visualize large amount of data. For modeling, high dimensional input variables potentially causes the problem of high variance as discussed in chapter 2, or intractable computation complexity. For visualization, it is a real constraint since the maximum number of dimensions one can observe and analyze is probably no more than three. The core idea is to project data onto a set of atoms in a dictionary. This projection is lossy, meaning that a certain amount of less informative variance is discarded. Another important application is to solve inverse problems such as super-resolution or denoising. These problems are very ill-posed and cannot be solved directly by least-squares methods. A good representation of the data can play the role as a regularization term, making the problem easier to solve. The choice of a dictionary defines the space in which we search for solutions.

## 3.2 Proper Orthogonal Decomposition (POD) as a representation of turbulent fields

PCA (Jolliffe, 2002), known also as *Karhunen–Loève* transform, is commonly used in many fields of signal and image processing. It was first introduced in turbulence studies by Lumley, (1967) under the name “*proper orthogonal decomposition*” (POD). It is then used as a standard approach for dimensionality reduction, which is largely beneficial in large scales reconstruction, flow control and coherent structure studies. PCA decomposes a sequence of snapshots into a dictionary  $\mathbf{D}$  to represent spatial structures, and a coefficient matrix  $\mathbf{A}$ , to capture temporal dynamics. The works on coherent structures and flow information extraction focus on the dictionary  $\mathbf{D}$  (Bonnet et al., 1994; Gordeyev and Thomas, 2000). Each atom  $\mathbf{d}_i$ , as ranked by its variance (an equivalent measure of the kinetic energy), represents the most energetic structures of the flow. Flow control and modeling works focus more on the use of projection coefficients  $\mathbf{A}$ . After learning the fixed dictionary  $\mathbf{D}$ , independent of time, from given training fields, the temporal dynamics of the flow is presented in the projection coefficients  $\mathbf{A}$  only. The efforts to model and control the flow are reduced tremendously by considering only several coefficients of high-variance atoms. This property is beneficial for reduced-order modeling works in flow control and reconstruction of large-scale velocity fields (Ly and Tran, 2001; Ravindran, 2000; Taylor and Glauser, 2004).

To derive POD or PCA, let first denote a data matrix  $\mathbf{X}$  of size  $P \times M$  containing a sequence of  $P$ -dimensional input vectors  $\mathbf{x}_t, t = 1, 2, \dots, M$ ,

$$\mathbf{X} \triangleq [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M], \quad (3.3)$$

assumed centered *a priori*:

$$\sum_{t=1}^M \mathbf{x}_t = \mathbf{0} \quad (3.4)$$

From  $\mathbf{X}$ , a dictionary  $\mathbf{D}$  of size  $P \times P$  is estimated, which contains  $P$  atoms  $\mathbf{d}_i \in \mathbb{R}^P, i = 1, 2, \dots, P$ . The objective is to project  $\mathbf{x}_t$  onto a reduced-order subspace of dimension  $L \leq P$  while maximizing the amount of variance. This idea, when applying to turbulent fields where variance represents the kinetic energy, is interpreted as determining the most energetic structures among a sequence of snapshots.

The first step is to find the most dominant basis function  $\mathbf{d}_1 \in \mathbb{R}^P$ , assuming  $\|\mathbf{d}_1\|_2^2 = 1$ . Each input vector is projected onto this direction as  $\mathbf{d}_1^\top \mathbf{x}_t$ . The variance of this projection is:

$$s_1 = \frac{1}{M} \sum_{t=1}^M (\mathbf{d}_1^\top \mathbf{x}_t)^2 = \mathbf{d}_1^\top \Sigma \mathbf{d}_1 \quad (3.5)$$

where  $\Sigma$  is the covariance matrix:

$$\Sigma = \frac{1}{M} \mathbf{X} \mathbf{X}^\top \quad (3.6)$$

The aim now is to find  $\mathbf{d}_1$  such that  $s_1$  is maximum, with the constraint that  $\|\mathbf{d}_1\|_2^2 = 1$ :

$$\mathbf{d}_1 = \underset{\mathbf{d}_1}{\operatorname{argmax}} \left\{ \mathbf{d}_1^\top \Sigma \mathbf{d}_1 \right\} \quad s.t. \quad \|\mathbf{d}_1\|_2^2 = 1 \quad (3.7)$$

This optimization problem can be rewritten in an unconstrained manner by introducing a Lagrange multiplier  $\sigma_1$ :

$$\mathbf{d}_1 = \underset{\mathbf{d}_1}{\operatorname{argmax}} \left\{ \mathbf{d}_1^\top \Sigma \mathbf{d}_1 + \sigma_1 (1 - \mathbf{d}_1^\top \mathbf{d}_1) \right\} \quad (3.8)$$

Setting the derivative with respect to  $\mathbf{d}_1$  to zero, one obtains:

$$\Sigma \mathbf{d}_1 = \sigma_1 \mathbf{d}_1 \quad (3.9)$$

which implies that  $\mathbf{d}_1$  is an eigenvector of  $\Sigma$ , and  $\sigma_1$  is the equivalent eigenvalue. Also,

multiplying both side with  $\mathbf{d}_1^\top$ , using  $\mathbf{d}_1^\top \mathbf{d}_1 = 1$ , one has:

$$\mathbf{d}_1^\top \Sigma \mathbf{d}_1 = \sigma_1 \quad (3.10)$$

The above expression implies that  $\sigma_1$  is the variance of the projection onto  $\mathbf{d}_1$ . This variance is maximized by selecting the maximum eigenvalue of the covariance matrix  $\Sigma$ . This process is repeated to find the second basis of the dictionary  $\mathbf{d}_2$  such that it is orthogonal to the first one, i.e.  $\mathbf{d}_1^\top \mathbf{d}_2 = 0$ . This is identical to find the second eigenvector corresponding to the second largest eigenvalue  $\lambda_2$ . The procedure of seeking all  $P$  basis vectors is identical to eigenvalue decomposition and SVD of the covariance matrix  $\Sigma$ . Sorting  $P$  eigenvalues in a descending order as  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_P \geq 0$ , one obtains the dictionary, or codebook, learned from the training samples  $\mathbf{X}$  by putting the equivalent eigenvectors together as  $\mathbf{D} \triangleq [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_P]$ . The projection matrix  $\mathbf{A}$  is found such that:

$$\mathbf{X} = \mathbf{D}\mathbf{A} \quad (3.11)$$

### 3.3 Dictionary learning as a new representation

Due to the orthogonality and variance maximization properties, PCA gains its success in many problems such as dimensionality reduction, low-order modeling and lossy compression. However, in solving different inverse problems, PCA is subject to several limitations. The number of atoms is at most the dimension of input vectors, leading to the “limited expressiveness” property (Tošić and Frossard, 2011). The representation learned by PCA can be efficient for training data, but a good generalization is not guaranteed. It is desirable to ignore the constraint on the number of basis functions and learn a redundant dictionary. Due to the redundancy, the sparsity constraint can be imposed and play the role of the regularization to solve ill-posed inverse problems.

#### 3.3.1 Redundant dictionary and sparse representation

Dictionary learning (DL) ignores the constraint on the number of basis functions, or atoms, permitting to find a *redundant* (or *overcomplete*) dictionary  $\mathbf{D} \in \mathbb{R}^{P \times K}$  to represent the data.  $K$  is the number of atoms, and redundancy means that  $K$  is potentially larger than  $P$ . These vectors are therefore not orthogonal. The companion of redundancy is sparsity, where the linear transformation matrix  $\mathbf{A}$  contains only a few nonzero coefficients. The representation then relies on the duality between redundancy and sparsity, which will be discussed further in this section. With more atoms, the representation using dictionary

learning is expected to be more adaptive to the signal and to better represent new data. This is the reason why the approach often gives the state-of-the-art results in most inverse problems in image processing (Elad et al., 2010; Yang et al., 2010).

The problem of representing data  $\mathbf{X}$  using the redundant dictionary  $\mathbf{D}$  and sparse coefficients  $\mathbf{A}$  includes two alternating optimization problems. The first one is to estimate the projection coefficients  $\mathbf{A}$ . PCA, with the orthogonality among atoms, simply estimates  $\mathbf{A}$  by dot products. Dictionaries, which are not necessarily orthogonal, can have more atoms than the dimension ( $K > P$ ). There are potentially many matrices  $\mathbf{A}$  such that  $\mathbf{X} = \mathbf{DA}$ . The sparsity constraint, meaning that  $\mathbf{A}$  has a minimal number of nonzero coefficients, is imposed to make the solution unique. The representation of  $\mathbf{X}$  becomes approximate, i.e.  $\mathbf{X} \approx \mathbf{DA}$ . Finding  $\mathbf{A}$  becomes an optimization problem of the form:

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{A}\|_p \quad \text{s.t.} \quad \mathbf{X} = \mathbf{DA} \quad (3.12)$$

The notation  $\hat{\cdot}$  is to distinguish between solution and variable in each optimization problem. The above problem can be re-arranged in the form of a regularized cost function as in chapter 2:

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{DA}\|_2^2 + \lambda \|\mathbf{A}\|_p \quad (3.13)$$

The common  $\ell^p$  norm  $\|\cdot\|_p$  is with  $0 \leq p \leq 1$ . Solving the problem with the  $\ell^0$  norm, i.e. counting the number of nonzero coefficients, leads to orthogonal matching pursuit (OMP) (Tropp and Gilbert, 2007), while solving with the  $\ell^1$  norm, summing the absolute values of the coefficients, leads to LASSO as discussed in chapter 2. Least Angle Regression (LARS) (Efron et al., 2004) is sometimes used as another efficient algorithm to solve  $\ell^1$  penalty problems and gives results very similar to LASSO. These regularizations help selecting a limited number of atoms that best approximate the input  $\mathbf{X}$ . The second problem is the choice of the dictionary, which is solved by efficient algorithms discussed in the next section.

### 3.3.2 Dictionary learning methods

Many algorithms have been proposed to learn the dictionary from data, starting with gradient descend method (Olshausen and Field, 1996), then K-SVD (Aharon et al., 2006), feature-sign approach (Lee et al., 2006) and online dictionary learning (Mairal et al., 2010). This section reviews some of those that are used later in this chapter.

### Alternate optimization

Dictionary learning is an alternate optimization problem to search the redundant dictionary  $\mathbf{D}$  and the sparse matrix  $\mathbf{A}$ . The approach has started by using a gradient descent approach (Olshausen and Field, 1996). It has recently gained popularity thanks to the progress in solving regularized optimization problems with  $\ell^p$  penalty (Donoho and Elad, 2003; Donoho et al., 2012; Efron et al., 2004; Tibshirani, 1996). One of the first efficient dictionary learning approach is the *method of optimal directions* (MOP) (Engan et al., 1999) where  $\mathbf{D}$  and  $\mathbf{A}$  are estimated by solving:

$$(\hat{\mathbf{D}}, \hat{\mathbf{A}}) = \underset{\mathbf{D}, \mathbf{A}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_2^2 \quad s.t. \quad \|\mathbf{a}_t\|_0 < L \quad \forall t \quad (3.14)$$

$L$  is the sparsity constraint- the maximum number of nonzero coefficients. This optimization problem is combinatorial and highly non-convex. The search for a local minimum is done by alternating two steps. With a fixed dictionary  $\mathbf{D}$ , the *sparse coding* step finds the representation of the input vectors by solving:

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \left\{ \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_2^2 + \lambda \|\mathbf{A}\|_p \right\} \quad (3.15)$$

This step uses OMP for  $\ell^0$  penalty and LASSO or LARS for  $\ell^1$  penalty. The *dictionary update* step re-estimates  $\mathbf{D}$  with fixed  $\mathbf{A}$  via the Moore-Penrose pseudo-inverse:

$$\mathbf{D} = \mathbf{X}\mathbf{A}^\dagger = \mathbf{X}\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1} \quad (3.16)$$

This scheme converges rapidly after some iterations, but requires complex matrix inversion, which is not efficient in most cases (Aharon et al., 2006).

### K-SVD

The *K-SVD* algorithm was proposed later by Aharon et al., (2006) and rapidly gained its popularity. The *dictionary update* is done by a block-relaxation approach. Instead of inverting the matrix, the algorithm updates each atom in an efficient way by generalizing the *k-means* clustering method (Bishop, 2006). The  $k$ -th atom is estimated by minimizing a quadratic error:

$$\{\hat{\mathbf{d}}_k, \hat{\mathbf{a}}_k\} = \underset{\mathbf{d}_k, \mathbf{a}_k}{\operatorname{argmin}} \|\mathbf{E}_k - \mathbf{d}_k \mathbf{a}_k^\top\|_2^2 \quad (3.17)$$

where  $\mathbf{E}_k$  is defined as the residual matrix:

$$\mathbf{E}_k \triangleq \mathbf{X}_k - \sum_{j \neq k} (\mathbf{d}_j \mathbf{a}_j^\top - \mathbf{d}_k \mathbf{a}_k^\top) \quad (3.18)$$

---

**Algorithm 3** Online dictionary learning algorithm by Mairal et al., (2010)

---

1: Input:

- a set of  $\mathcal{S}(\mathbf{x})$  of  $M$  input variable  $\mathbf{x}_t \in \mathbb{R}^P$ ;
- $\lambda \in \mathbb{R}$  : regularization parameter ( $\lambda \geq 0$ )
- $\mathbf{D}_0 \in \mathbb{R}^{P \times K}$ : initial dictionary

2:  $\mathbf{G}_0 \leftarrow 0, \mathbf{H}_0 \leftarrow 0$

3: **for**  $t = 1, 2, \dots, M$  **do**

4:   Draw  $\mathbf{x}_t$  from  $\mathcal{S}(\mathbf{x})$

5:   Sparse coding:

$$\hat{\mathbf{a}}_t = \underset{\mathbf{a} \in \mathbb{R}^K}{\operatorname{argmin}} \{ \|\mathbf{x}_t - \mathbf{D}_{t-1} \mathbf{a}\|_2 + \lambda \|\mathbf{a}\|_1 \} \quad (3.19)$$

6:    $\mathbf{G}_t \leftarrow \mathbf{G}_{t-1} + \hat{\mathbf{a}}_t \hat{\mathbf{a}}_t^\top$

7:    $\mathbf{H}_t \leftarrow \mathbf{H}_{t-1} + \hat{\mathbf{a}}_t \mathbf{x}_t^\top$

8:   Update  $\mathbf{D}_t$ :

$$\begin{aligned} \hat{\mathbf{D}}_t &= \underset{\mathbf{D} \in \mathbb{C}}{\operatorname{argmin}} \left\{ \frac{1}{t} \sum_{i=1}^t \frac{1}{2} \|\mathbf{x}_i - \mathbf{D} \hat{\mathbf{a}}_i\|_2^2 + \lambda \|\hat{\mathbf{a}}_t\|_1 \right\} \\ &= \underset{\mathbf{D} \in \mathbb{C}}{\operatorname{argmin}} \left\{ \frac{1}{t} \sum_{i=1}^t \frac{1}{2} \operatorname{Tr}(\mathbf{D}^\top \mathbf{D} \mathbf{G}_t) - \operatorname{Tr}(\mathbf{D}^\top \mathbf{H}_t) \right\} \end{aligned} \quad (3.20)$$

9: **end for**

10: Return  $\hat{\mathbf{D}}_M$

---

where  $\mathbf{X}_k$  gathers examples  $\mathbf{x}_t$  using  $\mathbf{d}_k$  in their representation. The update of both  $\mathbf{d}_k$  and  $\mathbf{a}_k$  is done simultaneously via a rank-1 approximation, i.e. considering only the first eigenvector of  $E_k$  after performing a SVD.

### Online dictionary learning (ODL)

K-SVD algorithm has certain constraints, mainly with a potential local minimum while iteratively learning the dictionary (Rubinstein et al., 2010). The algorithm is also limited to small numbers of samples due to its computational cost. Online dictionary learning (ODL) has been proposed by Mairal et al., (2010) to learn  $\mathbf{D}$  from massive datasets. The use of stochastic gradient descent to update the dictionary for each example (or a group of them), and LARS algorithm for sparse coding, permits the learning over millions of samples. Algorithm 3 summarizes this approach.



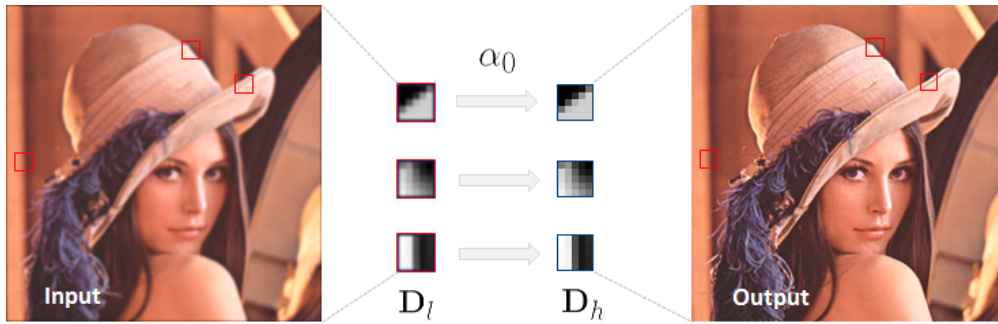


Figure 3.1 – An example of single image super-resolution using coupled dictionary learning and patch-wise approach. Photo credit: Yang et al., (2010).

## 3.4 Learning coupled dictionaries

In this section, DL is used to solve the problem of reconstructing HR fields from LR ones. The approach is inspired by the *single image super-resolution* application (Yang et al., 2010, 2012; Zeyde et al., 2012), i.e. estimating HR images with finer details from LR ones. The approach jointly learns coupled dictionaries at LR and HR from given training samples. These dictionaries are forced to represent the data using the same coefficients. In the prediction phase, the projection coefficients are estimated from the LR images and then combined with the learned HR dictionary to reconstruct the HR images.

### 3.4.1 Patch-based approach

DL requires a sparse coding step, i.e. solving the  $l^1$ -penalty optimization problem. The computation complexity is  $O(P^3 + MP^2)$  (Efron et al., 2004), where  $P$  is the dimension of the fields and  $M$  is the number of samples. Since  $P \gg M$  in most cases, the complexity is  $O(P^3)$ , which is a very heavy procedure. The computation is therefore intractable with the whole field. To overcome this difficulty, the “*patch-wise*” approach uses small patches instead of the whole fields. These patches are extracted from the original fields at all positions by moving pixel-by-pixel in both horizontal and vertical directions. Let  $m$  be the total number of patches, and  $p$  be their dimension ( $m \gg p$ ), the computation complexity becomes  $O(mp^2)$ , which is tractable when  $p$  is small. In the reconstruction phase, since one pixel belongs to several intersecting patches, the estimated field is reconstructed by aggregating all possible estimates of each pixel after putting them back into the whole scene.

The path-wise approach comes with several advantages. First, it localizes the information. A shared dictionary can be learned to represent all type of images with different contents. Second, it reduces tremendously the computation. Standard images are of dimension few millions, while typical patch sizes are around  $8 \times 8$ , which is a few orders

of magnitude smaller. Last, since the learning is on small patches, the number of samples required for the algorithm to converge is much smaller. From a single image, millions of small patches can be extracted for learning. Working with the original image, the number of training samples is necessary at least one order of magnitude higher than its dimension, which is already millions of pixels.

To formulate the patch-based approach, let  $z_t \in \mathbb{R}^P$  be a snapshot of a high-resolution field as a column vector. The operator  $\mathcal{R}_h^k : \mathbb{R}^P \mapsto \mathbb{R}^p$  is to extract the 2D patch and put them in a lexicographical order to form a column vector  $\mathbf{p}_h^k \triangleq \mathcal{R}_h^k z_t \in \mathbb{R}^p$ , where  $p$  is the size of 2D patches  $\mathbf{p}_h^k$ . If the overlapping is such that all patches are translated pixel-by-pixel, the total number of patches is  $(\sqrt{P} - \sqrt{p} + 1)^2$ . The reconstruction of the whole image is done as:

$$\hat{z}_t = \left[ \sum_k (\mathcal{R}_h^k)^\top \mathcal{R}_h^k \right]^{-1} \sum_k (\mathcal{R}_h^k)^\top \hat{\mathbf{p}}_h^k \quad (3.21)$$

where  $\hat{\mathbf{p}}_h^k$  and  $\hat{z}_t$  are the estimates of the reference  $\mathbf{p}_h^k$  and  $z_t$  respectively. The term  $(\mathcal{R}_h^k)^\top : \mathbb{R}^p \mapsto \mathbb{R}^P$  puts the equivalent patch into the global 2D scene and zero-pad elsewhere. The term  $(\mathcal{R}_h^k)^\top \mathcal{R}_h^k$  just counts the number of estimates for each pixels. Its inversion plays the role of a normalization factor. In practice, the whole process is done by estimating each pixel as its mean or median of all estimates from all patches that it belongs to.

### 3.4.2 The approach

Let assume that small patches from the field can be represented as a linear combination of several atoms from the learned dictionary. This assumption is the so-called “*Sparse-Land*” prior (Zeyde et al., 2012). The idea is to learn coupled dictionaries by imposing that the representation coefficients are the same at low and high resolution. The following section recalls the main ideas. More details can be found in Zeyde et al., (2012).

Suppose that  $z_t \in \mathbb{R}^P$  is the true high resolution velocity field. The corresponding low-resolution field  $y_t \in \mathbb{R}^Q$  ( $Q < P$ ) is obtained as:

$$\mathbf{y}_t = \mathbf{S}_s \mathbb{L}_s z_t + \mathbf{v}_t \quad (3.22)$$

where  $\mathbf{v}_t$  is a random noise omitted in this case,  $\mathbb{L}_s$  is an anti-aliasing low-pass filter, and  $\mathbf{S}_s$  is a subsampling operator. The presence of  $\mathbb{L}_s$  is to avoid the problem of aliasing when subsampling the field. In the case of direct subsampling, this filter is omitted from the above model. The goal is to find a HR estimate  $\hat{z}_t$  containing both large and small

scales. The most naive and simple idea is to interpolate from  $\mathbf{y}$ , i.e.  $\hat{\mathbf{z}}_t = \mathbb{I}_s \mathbf{y}_t \in \mathbb{R}^P$ . However, it will give no access to small-scale information above the cutoff frequency defined by the low-resolution grid. The coupled dictionary approach permits to learn small scales from training data and uses it for the reconstruction.

DL uses the patch-based approach, where the couples of LR and HR patches are extracted from the LR and HR fields respectively. Let  $\mathbf{p}_h^k = \mathcal{R}_h^k \mathbf{z}_t \in \mathbb{R}^p$  be the  $k$ -th HR patch. By assuming the Sparse-Land model for high and low resolution training fields, each patch can be estimated as a linear combination of atoms in a dictionary:

$$\mathbf{p}_h^k = \mathbf{D}_h \mathbf{a}^k + \boldsymbol{\epsilon}^k \quad (3.23)$$

where  $\mathbf{D}_h \in \mathbb{R}^{p \times K}$  is the HR dictionary,  $\mathbf{a}^k \in \mathbb{R}^K$  is the coefficient vector and  $\boldsymbol{\epsilon}^k$  is the reconstruction error. The dictionary is redundant ( $K > p$ ) and the coefficients are sparse ( $\|\mathbf{a}^k\|_0 < L$ ), with the sparsity constraint  $L \ll K$ . The corresponding LR patch  $\mathbf{p}_\ell^k \in \mathbb{R}^q$  is:

$$\mathbf{p}_\ell^k = \mathcal{R}_\ell^k \mathbf{y}_t = \mathcal{R}_\ell^k \mathbf{S}_s \mathbb{L}_s \mathbf{z}_t + \mathcal{R}_\ell^k \mathbf{v}_t \quad (3.24)$$

$\mathcal{R}_\ell^k$  is just an extraction operator, and  $\mathbf{S}_s \mathbb{L}_s$  is a transformation operator going from HR to LR fields. Since  $\mathbf{S}_s$  and  $\mathbb{L}_s$  are spatially independent operators, there exist local  $\mathbf{S}_s^{loc}$  and  $\mathbb{L}_s^{loc}$  that transform HR to LR patches:

$$\mathbf{p}_\ell^k = \mathbf{S}_s^{loc} \mathbb{L}_s^{loc} \mathbf{p}_h^k + \mathbf{v}_\ell^k \quad (3.25)$$

where  $\mathbf{v}_\ell^k$  is a random noise of the same dimension as the LR patches. From equations 3.23 and 3.25, one can write:

$$\begin{aligned} \mathbf{p}_\ell^k &= \mathbf{S}_s^{loc} \mathbb{L}_s^{loc} \mathbf{D}_h \mathbf{a}^k + \mathbf{S}_s^{loc} \mathbb{L}_s^{loc} \boldsymbol{\epsilon}^k + \mathbf{v}_\ell^k \\ &= \mathbf{S}_s^{loc} \mathbb{L}_s^{loc} \mathbf{D}_h \mathbf{a}^k + \tilde{\mathbf{v}}_\ell^k \end{aligned} \quad (3.26)$$

where  $\tilde{\mathbf{v}}_\ell^k \triangleq \mathbf{S}_s^{loc} \mathbb{L}_s^{loc} \boldsymbol{\epsilon}^k + \mathbf{v}_\ell^k$  is also a random noise term. Denoting  $\mathbf{D}_\ell \triangleq \mathbf{S}_s^{loc} \mathbb{L}_s^{loc} \mathbf{D}_h$ , the above equation illustrates that there exists also a Sparse-Land model for LR patches. These models for LR and HR patches also share the same sparse coefficient  $\mathbf{a}^k$ . The LR dictionary is also a downsampled version of the HR one.

### 3.4.3 Joint learning methods

In practice,  $\mathbf{S}_s \mathbb{L}_s$  or  $\mathbf{S}_s^{loc} \mathbb{L}_s^{loc}$  are usually not given. Yang et al., (2010) has addressed this problem by proposing the approach to jointly learn coupled dictionaries. Zeyde et al., (2012) follow the main idea with some modifications. All approaches contains two main

steps: *learning phase* and *reconstruction phase*. The learning phase can be done *offline*, i.e. training *a priori* from the data. With learned dictionaries, the reconstruction phase can be *online*.

Given the training HR velocity fields  $\mathbf{z}_t \in \mathbb{R}^P$ , corresponding LR fields are virtually extracted as  $\mathbf{y}_t = \mathbf{S}_s \mathbf{L}_s \mathbf{z}_t \in \mathbb{R}^Q$ . Couples of LR and HR patches  $\{\mathbf{p}_\ell^k, \mathbf{p}_h^k\}$  are extracted from the training fields as  $\mathbf{p}_\ell^k = \mathcal{R}_\ell^k \mathbf{y}_t \in \mathbb{R}^q$  and  $\mathbf{p}_h^k = \mathcal{R}_h^k \mathbf{z}_t \in \mathbb{R}^p$ , where  $q$  and  $p$  are the size of LR and HR patches respectively, and  $p/q = P/Q$ . Let  $\mathbf{P}_\ell$  and  $\mathbf{P}_h$  denote matrices of all LR and HR patches:

$$\begin{aligned} \mathbf{P}_\ell &= [\mathbf{p}_\ell^1 \ \mathbf{p}_\ell^2 \ \dots \ \mathbf{p}_\ell^m]_{q \times m} \\ \mathbf{P}_h &= [\mathbf{p}_h^1 \ \mathbf{p}_h^2 \ \dots \ \mathbf{p}_h^m]_{p \times m} \end{aligned} \quad (3.27)$$

where  $m$  is the total number of patches, and  $m = (\sqrt{Q} - \sqrt{q} + 1)^2 \times M$  at most, where  $M$  is the number of training planes. The LR patches are collected by one-pixel overlapping, while HR ones are obtained by overlapping  $p/q$  pixels to ensure the same number of LR and HR patches. Coupled dictionaries  $\{\mathbf{D}_h, \mathbf{D}_\ell\}$  are learned from the coupled training patches  $\{\mathbf{P}_h, \mathbf{P}_\ell\}$  imposing to share the same sparse representation:

$$\begin{cases} \mathbf{P}_h \approx \mathbf{D}_h \mathbf{A} \\ \mathbf{P}_\ell \approx \mathbf{D}_\ell \mathbf{A} \end{cases} \quad (3.28)$$

The first learning approach is proposed by Yang et al., 2010, which aims at solving an optimization problem:

$$\{\hat{\mathbf{D}}_h, \hat{\mathbf{D}}_\ell, \hat{\mathbf{A}}\} = \underset{\mathbf{D}_h, \mathbf{D}_\ell, \mathbf{A}}{\operatorname{argmin}} \left\{ \frac{1}{p} \|\mathbf{P}_h - \mathbf{D}_h \mathbf{A}\|_2^2 + \frac{1}{q} \|\mathbf{P}_\ell - \mathbf{D}_\ell \mathbf{A}\|_2^2 + \lambda_1 \left( \frac{1}{p} + \frac{1}{q} \right) \|\mathbf{A}\|_1 \right\} \quad (3.29)$$

$\mathbf{A}$  is the shared sparse coefficients, which ensures a compromise between small reconstruction errors of patches and sparsity constraint. The two normalization terms  $1/q$  and  $1/p$  are to balance the two mean-square error terms. The above cost function can be re-written as:

$$\{\hat{\mathbf{D}}, \hat{\mathbf{A}}\} = \underset{\mathbf{D}, \mathbf{A}}{\operatorname{argmin}} \left\{ \|\mathbf{P} - \mathbf{D} \mathbf{A}\|_2^2 + \lambda_1 \|\mathbf{A}\|_1 \right\} \quad (3.30)$$

where

$$\mathbf{P} = \begin{bmatrix} \frac{1}{\sqrt{p}} \mathbf{P}_h \\ \frac{1}{\sqrt{q}} \mathbf{P}_\ell \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \frac{1}{\sqrt{p}} \mathbf{D}_h \\ \frac{1}{\sqrt{q}} \mathbf{D}_\ell \end{bmatrix} \quad (3.31)$$

This problem can be solved using the standard DL algorithms.

The second approach is proposed by Zeyde et al., (2012), which contains a direct model to estimate the HR dictionary. The LR dictionary is first learned from LR patches:

$$\{\hat{\mathbf{D}}_\ell, \hat{\mathbf{A}}\} = \underset{\mathbf{D}_\ell, \mathbf{A}}{\operatorname{argmin}} \left\{ \|\mathbf{P}_\ell - \mathbf{D}_\ell \mathbf{A}\|_2^2 + \lambda_1 \|\mathbf{A}\|_1 \right\} \quad (3.32)$$

Assuming that the representation of HR patches  $\mathbf{P}_h$  via the HR dictionary  $\mathbf{D}_h$  will share the same sparse code  $\mathbf{A}$ , HR dictionary is estimated as:

$$\hat{\mathbf{D}}_h = \underset{\mathbf{D}_h}{\operatorname{argmin}} \left\{ \|\mathbf{P}_h - \mathbf{D}_h \hat{\mathbf{A}}\|_2^2 \right\} \quad (3.33)$$

for which the solution is easy to obtain by a pseudo-inverse:

$$\hat{\mathbf{D}}_h = \mathbf{P}_h \hat{\mathbf{A}}^\dagger = \mathbf{P}_h \hat{\mathbf{A}}^\top (\hat{\mathbf{A}} \hat{\mathbf{A}}^\top)^{-1} \quad (3.34)$$

As originally proposed for image super-resolution where sharp edges are important, Yang et al., (2010, 2012) and Zeyde et al., (2012) use different pre-processing techniques before learning coupled dictionaries. Interpolated images  $\mathbb{I}_s \mathbf{y}_t$  of the same dimension as HR ones are considered as LR images. Yang et al., (2010, 2012) couple HR patches, i.e.  $\mathbf{p}_h^k = \mathcal{R}_h^k z_t$ , with the derivatives of interpolated LR images  $\mathbb{F} * \mathbb{I}_s \mathbf{y}_t$ . The operator  $*$  is the convolution operator. These derivatives are obtained from the convolution of four different 1D kernels  $\mathbb{F}$  of first and second order:

$$\left[ -1 \ 0 \ 1 \right], \left[ -1 \ 0 \ 1 \right]^\top, \left[ 1 \ 0 \ -2 \ 0 \ 1 \right], \left[ 1 \ 0 \ -2 \ 0 \ 1 \right]^\top \quad (3.35)$$

LR patches are then extracted as  $\mathbf{p}_\ell^k = \mathcal{R}_\ell^k (\mathbb{F} * \mathbb{I}_s \mathbf{y}_t)$ . Zeyde et al., (2012) couples the same features  $\mathbb{F} * \mathbb{I}_s \mathbf{y}$ , but with the residual between HR and interpolated images, i.e.  $\mathbf{p}_h^k = \mathcal{R}_h^k (z_t - \mathbb{I}_s \mathbf{y}_t)$ . Moreover, the dimension of the patches is reduced using PCA, which corresponds to an adaptive low-pass filter. This step in practice is important, since applying the four filters to low resolution fields bring redundant and superfluous information and lead to unnecessary heavy computation.

Based on the above works by Yang et al., (2010) and Zeyde et al., (2012), we study three different approaches, namely “SR1”, “SR2” and “SR3”, to learn coupled dictionaries. Descriptions of LR and HR patches with their sizes are summarized in table 3.1. The dimension of LR patches when using derivatives is  $4p$ ; However in practice, it is reduced significantly via PCA while retaining 99.9% of energy content.

Notation	Patch extraction		Patch dimension	
	LR	HR	LR	HR
SR1	$\mathcal{R}_\ell^k \mathbf{S}_s \mathbf{L}_s \mathbf{z}_t$	$\mathcal{R}_h^k \mathbf{z}_t$	$q$	$p$
SR2	$\mathcal{R}_h^k \mathbf{I}_s \mathbf{S}_s \mathbf{L}_s \mathbf{z}_t$	$\mathcal{R}_h^k \mathbf{z}_t$	$p$	$p$
SR3	$\mathcal{R}_h^k (\mathbf{F} * \mathbf{I}_s \mathbf{y}_t)$	$\mathcal{R}_h^k \{\mathbf{z}_t - \mathbf{I}_s \mathbf{y}_t\}$	$4p$	$p$

Table 3.1 – Notations for three different methods of coupled dictionaries learning as proposed by Yang et al., (2010) and Zeyde et al., (2012). The dimension of LR and HR patches are  $q$  and  $p$  respectively.

### 3.4.4 Reconstruction using learned dictionaries

Having the coupled dictionaries at hand and given a LR field  $\mathbf{y}^e$ , the objective is to reconstruct the HR  $\mathbf{z}^e$ . The superscript “ $e$ ” stands for “*external*”, meaning outside of the training planes. First, all LR patches  $\mathbf{P}_\ell^e$  are extracted from  $\mathbf{y}^e$  with the overlapping of one pixel.  $\mathbf{P}_\ell^e$  is then centered by removing the mean of each patch  $m_l$ . Next, the sparse code  $\mathbf{A}$  are estimated by solving the regularized least-squares problem:

$$\hat{\mathbf{A}}^e = \underset{\mathbf{A}^e}{\operatorname{argmin}} \left\{ \|\mathbf{P}_\ell^e - \hat{\mathbf{D}}_\ell \mathbf{A}^e\|_2^2 + \lambda_2 \|\mathbf{A}^e\|_1 \right\} \quad (3.36)$$

HR patches are estimated using this shared code and putting back the mean  $m_l$ :

$$\hat{\mathbf{P}}_h^e = \sqrt{\frac{p}{q}} \hat{\mathbf{D}}_h \hat{\mathbf{A}}^e + m_l \quad (3.37)$$

Finally, the HR field is reconstructed by solving an optimization problem:

$$\hat{\mathbf{z}}^e = \underset{\mathbf{z}^e}{\operatorname{argmin}} \left\{ \sum_k \|\mathcal{R}_h^k \hat{\mathbf{z}}^e - \hat{\mathbf{p}}_h^k\|_2^2 \right\} \quad (3.38)$$

This problem aims to find the best compromise between all estimates, and the closed-form least-squares solution is:

$$\hat{\mathbf{z}}^e = \left[ \sum_k (\mathcal{R}_h^k)^\top \mathcal{R}_h^k \right]^{-1} \sum_k (\mathcal{R}_h^k)^\top \hat{\mathbf{p}}_h^k \quad (3.39)$$

This is the overlapping procedure as discussed in section 3.4.1. A pseudo algorithm of the whole process including the learning and reconstruction phases is summarized in algorithm 4.

---

**Algorithm 4** Coupled dictionary learning for reconstruction of high-resolution fields from low-resolution ones

---

1: **Input:**

- training high-resolution fields  $\{z_t\}, t = 1, 2, \dots, M$
- testing low-resolution field  $\mathbf{y}^e$

2: **Step 1:** *Learning phase* (offline)

- Extract virtual low-resolution fields the same way as  $\mathbf{y}^e$  is recorded:

$$\mathbf{y}_t \triangleq \mathbf{S}_s \mathbb{L}_s z_t + \mathbf{v}_t \quad (3.40)$$

- Extract and join coupled patches  $\{\mathbf{P}_l, \mathbf{P}_h\}$  from the fields  $\{\mathbf{y}_t, z_t\}, t = 1, 2, \dots, M$ :

$$\mathbf{P} \triangleq \left[ \frac{1}{\sqrt{p}} \mathbf{P}_h; \frac{1}{\sqrt{q}} \mathbf{P}_l \right] \quad (3.41)$$

- Learn the joint dictionary  $\mathbf{D} \triangleq \left[ \frac{1}{\sqrt{p}} \mathbf{D}_h; \frac{1}{\sqrt{q}} \mathbf{D}_l \right]$  as:

$$(\hat{\mathbf{D}}, \hat{\mathbf{A}}) = \underset{\mathbf{D}, \mathbf{A}}{\operatorname{argmin}} \left\{ \|\mathbf{P} - \mathbf{D}\mathbf{A}\|_2^2 + \lambda_1 \|\mathbf{A}\|_1 \right\} \quad (3.42)$$

3: **Step 2:** *Reconstruction phase* (online)

- Extract LR patches:  $\mathbf{P}_\ell^e \in \mathbb{R}^{q \times m}$  from  $\mathbf{y}^e$
- Estimate the sparse code:

$$\hat{\mathbf{A}}^e = \underset{\mathbf{A}^e}{\operatorname{argmin}} \left\{ \|\mathbf{P}_\ell^e - \hat{\mathbf{D}}_\ell \mathbf{A}^e\|_2^2 + \lambda_2 \|\mathbf{A}^e\|_1 \right\} \quad (3.43)$$

- Reconstruct HR patches  $\hat{\mathbf{P}}_h^e = \hat{\mathbf{D}}_h \hat{\mathbf{A}}^e \in \mathbb{R}^{p \times m}$

4: **Output:** Reconstruct HR field  $\mathbf{z}^e$  from  $\hat{\mathbf{P}}_h^e$  by overlapping.

---

$\lambda_1$	$\lambda_2$	$\sqrt{q}$	$\sqrt{p}$	$K$	$m$
0.1 ~ 0.2	$10^{-5}$	4	16	$2(q \times p)$	$10^5$

Table 3.2 – Set of parameters for coupled dictionaries approach (notations are consistent with Algorithm 4): sparsity constrains  $\lambda_1$  for learning and  $\lambda_2$  for reconstruction; dimension of LR patches  $q$  and HR patches  $p$ ; number of atoms  $K$ ; number of training patches  $m$

### 3.5 Dictionary learning for isotropic turbulence fields

The section applies dictionary learning approach to the DNS database of the isotropic turbulence discussed in chapter 1.4.2. This data is chosen because the fields are periodic, isotropic and homogeneous. These properties will facilitate all computations and the handling of boundary conditions. Two main problems are addressed. First, the efficiency of the representation using dictionary learning will be studied, comparing with other approaches such as wavelet transform or PCA. Second, the coupled dictionaries approach is discussed to solve the problem of reconstructing HR velocity fields from LR measurements.

To compare with other approaches, we consider a similar configuration as in chapter 2 to study regression models. The reference HTHS data is subsampled to obtain the measurements of HTLS  $\{y_t\}$  and LTHS  $\{x_t\}$  (see figure 1.5). The subsampling ratios are  $P/Q = 4 \times 4$  in space and  $N/M = 6$  in time, corresponding to a moderate amount of energy losses in every directions. We will use  $\{x_t\}$  only for the training, then uses  $\{y_t\}$  to predict  $\{z_t\}$ .

#### 3.5.1 On the choice of parameters

Results in the following sections are obtained with a consistent set of parameters presented in table 3.2. Notations are consistent with the algorithm 4 to learn coupled dictionaries. This section briefly discusses these choices.

The first set of parameters are sparsity constraints  $\lambda_1$  and  $\lambda_2$ . For the learning phase,  $\lambda_1$  is about 0.1 ~ 0.2 to learn either the coupled dictionaries in equation 3.29 or the LR dictionary in equation 3.32. This constraint imposes that the reconstruction of training patches uses only 10 ~ 15 nonzero coefficients in average, corresponding to the sparsity level of 0.90 ~ 0.95 (only 5 ~ 10% coefficients are nonzero). As will be seen later in figure 3.3, this is a strong constraint. However, decreasing  $\lambda_1$ , i.e. reducing the sparsity level, downgrades reconstruction results of coupled dictionaries. This is probably the *overfitting* problem discussed in chapter 2. With small  $\lambda_1$ , the dictionaries represent well



the training data but with a poor generalization capability. For reconstruction phase,  $\lambda_2$  is chosen to be close to zero. In such case, the sparsity level of  $\mathbf{A}^e$  in equation 3.43 is very low, i.e. the maximum number of atoms is used to reconstruct  $\mathbf{P}_\ell^e$ .

The second set of parameters are to describe the training patches. The sizes of the LR and HR patches are  $q = 4 \times 4$  and  $p = 16 \times 16$ , respectively. The LR patch size is small compared to the whole LR field of size  $Q = 16 \times 16$ , permitting to localize the information. Numerical experiments show a slight improvement when increasing this size, but it leads to a high computation requirements and a large number of training planes. Coupled patches are extracted from  $37 \times 16$  LTHS planes for training. From each snapshot, a total of  $(16 - 4 + 1) \times (16 - 4 + 1) = 169$  LR patches can be extracted using one-pixel overlapping, i.e. translating the patch in each of the two dimension in space by one pixel. The same number of corresponding HR patches are also extracted from HR training fields. Around  $m = 10^5$  couples of LR-HR patches are chosen randomly from the whole set of  $37 \times 16 \times 169$  patches in total and used to train the coupled dictionaries. The objective is to reduce the computational cost, since many patches are very similar.

The last parameter is the number of atoms. There is no general rule, but new findings using non-parametric approaches of learning dictionaries with an adaptive number of atoms (Dang and Chainais, 2016) show that the needed level of redundancy maybe not very high. Numerical experiments in this work show that a good size is about twice the dimension of input variables. For the case of coupled dictionary learning, the choice is  $K = 2(p + q)$ , where  $(p + q)$  is the dimension of the coupled LR and HR patches.

### 3.5.2 Efficiency of representations: a comparative study

Since the sparsity may play a key role, this section compares the efficiency of different representations. “*Efficiency*” means the quality of the approximation with respect to different sparsity levels. Mathematical predefined representation using wavelets, learned bases using PCA and dictionary learning methods are compared. Both KSVD (Aharon et al., 2006) and ODL (Mairal et al., 2010) are investigated. The representations are learned from the LTHS planes  $\{\mathbf{x}_i\}$  in the cases of PCA and DL. These representations are then tested on random subsampled fields from HTHS planes  $\mathbf{z}_i$  that are different from the set of  $\{\mathbf{x}_i\}$  used for learning. Reconstruction errors are estimated by comparing with reference fields.

Figure 3.2 shows the adaptive dictionaries learned by PCA (left) and ODL (right) as ranked by their energy contents. The redundant dictionary by ODL has twice more atoms compared to PCA, which contains exactly  $p = 16 \times 16$  atoms as the dimension of the input patches. Only 256 over 512 atoms obtained by ODL are shown to be comparable with the PCA dictionary. The atoms are completely different in the two dictionaries. PCA

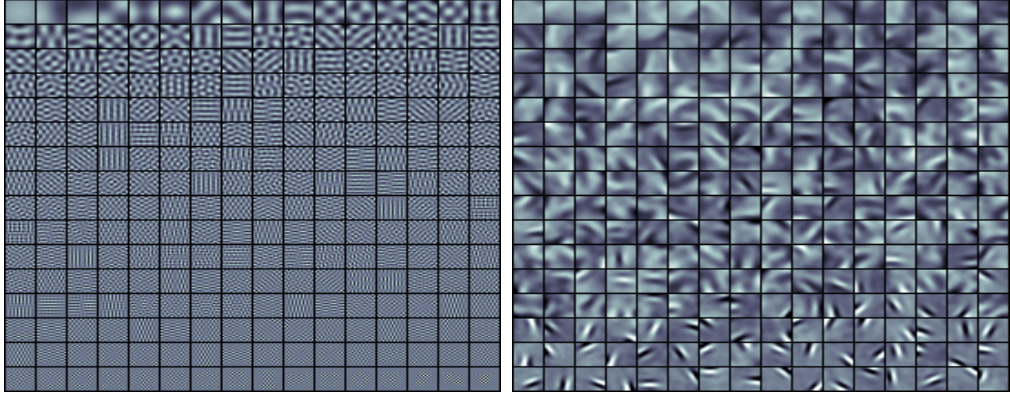


Figure 3.2 – Dictionaries learned by PCA and ODL from the set of HR patches of size  $p = 16 \times 16$ . With ODL, only 256 atoms are chosen from 512 atoms.

dictionary has a sharp decline of variance content. Also, since the number of training patches is sufficiently large, the atoms look similar to discrete cosine transform (DCT) basis functions, which have modulated sine-wave patterns. The redundant dictionary from ODL contains more “patterns” for each level of scale, which is expected to be more adaptive to the data for sparse representation.

Using a wavelet transform, there is no learning since the bases are predefined. The fields are decomposed into approximation and detail coefficients (horizontal, vertical and diagonal). The common Daubechies wavelet is used for its compact support and fast computation. The transform is performed on full fields. To test the sparsity effects, different thresholds are used. Detail coefficients larger than each threshold are retained while setting others to zeros, while approximation coefficients are kept unchanged. Inverse wavelet transform reconstructs the fields using these unchanged approximation and filtered detail coefficients. The sparsity is defined as the ratio of nonzero coefficients  $L$  (including both approximation and retained detail coefficients after filtering) and the dimension of the field  $P$ . The NRMSE between reconstructed fields  $\hat{z}_t$  and reference ones  $z_t$  is estimated as in equation 2.28 to qualify the reconstruction for each level of sparsity  $(1 - L/P)$ .

With PCA, the dictionary  $\mathbf{D}$  is learned from training patches  $\mathbf{P}_h$  extracted from all  $\{\mathbf{x}_t\}, t = 1, \dots, M$ . Its atoms  $\{\mathbf{d}_h^i\}$  are ranked by their variances  $\sigma_i$ , i.e.  $\sigma_1 > \sigma_2 > \dots > \sigma_P$ . The dimensionality is reduced such that the retained information from only the first  $L$  principal components ( $L < p$ ) is maximized. For a new field  $z_t$ ,  $\mathbf{P}_h$  are extracted and projected onto the first  $L$  vectors:

$$\mathbf{a}_i = \frac{\mathbf{P}_h \cdot \mathbf{d}_h^i}{\|\mathbf{d}_h^i\|_2^2}, i = 1, 2, \dots, L \quad (3.44)$$

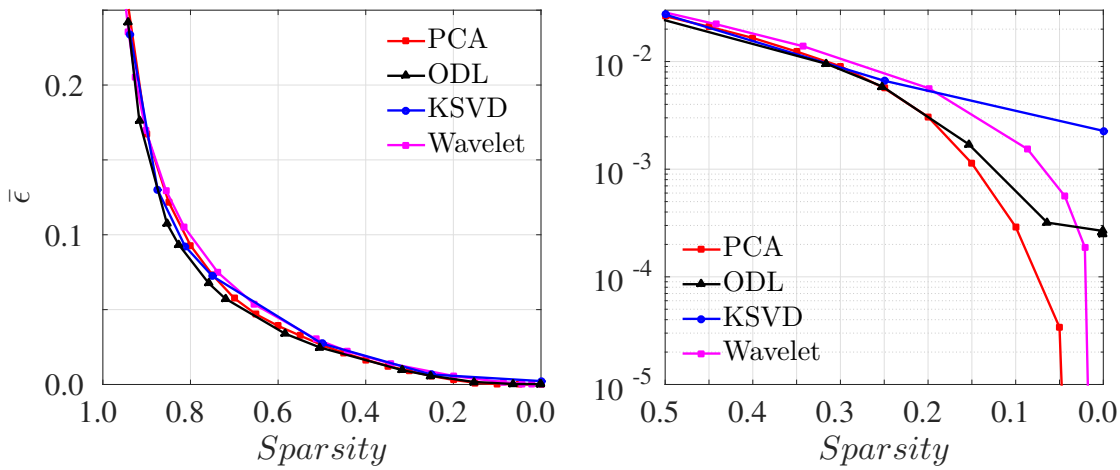


Figure 3.3 – Sparsity vs error outside (left) and zoom in the region of low sparsity at semi-log scale (right) for different representations: wavelet (Daubechies), PCA, dictionary learning (ODL or KSVD). Wavelet transform is for the whole fields of size  $96 \times 96$ , while PCA and DL are for patches of size  $p = 16 \times 16$ .

The filtered patches are estimated by combining the projected coefficients with the corresponding functions:

$$\hat{\mathbf{P}}_h = \sum_{i=1}^L \mathbf{a}_i d_h^i \quad (3.45)$$

Finally, the reconstructed field  $\hat{\mathbf{z}}_t$  is estimated by one-pixel overlapping as discussed in section 3.4.1. The sparsity level is defined as  $1 - L/p$ , where the patch size at HR is  $p = 16 \times 16$ . NRMSEs are estimated using equation 2.28.

KSVD and ODL learn  $\mathbf{D}$  from  $\mathbf{P}_h$  *a priori* with a high sparsity level ( $\lambda = 0.2$ , corresponding to about 15 non-zero coefficients). In the reconstruction step, all possible patches  $\mathbf{P}_h$  are extracted from each field  $\mathbf{z}_t$ . The sparse code is estimated using LARS algorithm to solve:

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \left\{ \|\mathbf{P}_h - \mathbf{D}\mathbf{A}\|_2^2 + \lambda \|\mathbf{A}\|_1 \right\} \quad (3.46)$$

The efficiency of the learned dictionary  $\mathbf{D}$  is studied by varying  $\lambda$  in this step. For each  $\lambda$ ,  $\hat{\mathbf{A}}$  is estimated and then used to re-estimate the patches  $\hat{\mathbf{P}}_h = \mathbf{D}\hat{\mathbf{A}}$  before reconstructing the global scene  $\hat{\mathbf{z}}_t$ . Sparsity is measured as the average of  $1 - \|\hat{\mathbf{a}}_t\|_0 / p$ , where  $\hat{\mathbf{a}}_t$  is the  $t$ -th row of  $\hat{\mathbf{A}}$ . NRMSEs are estimated the same way as of wavelet transform or PCA.

Figure 3.3 shows the curves of NRMSE as functions of sparsity levels. These errors are estimated from five testing planes equally far from neighboring LTHS snapshots. All curves behave similarly as reducing the error when sparsity decreases. At high levels (larger than 0.5), there is a clear benefit of using DL. With the same number of non-zero coefficients, both ODL and KSVD give slightly lower errors than PCA and wavelet transform. Few atoms from DL better represent the data than high variance principal

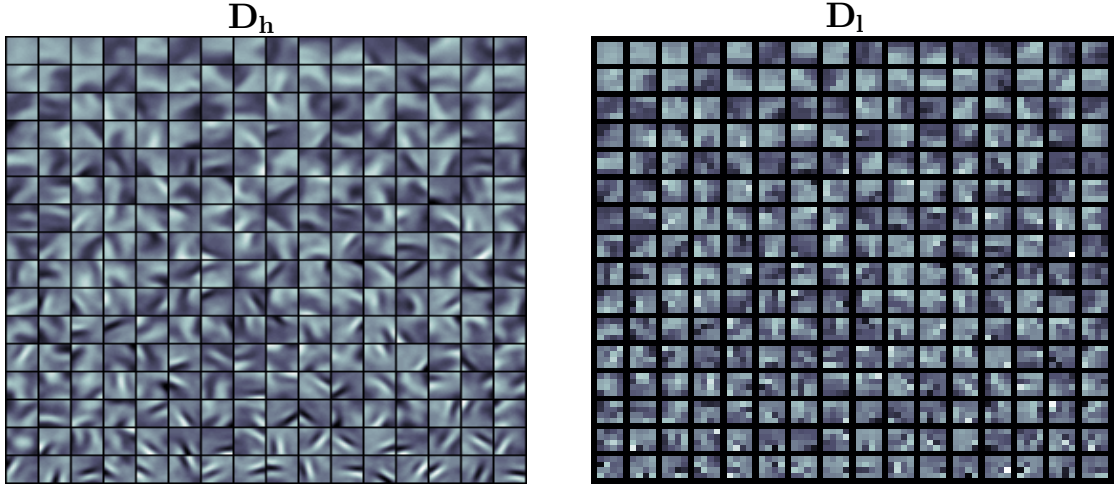


Figure 3.4 – Coupled dictionary of high and low resolution patches, with LR patches are of size  $4 \times 4$ , directly subsampled from their equivalent HR patches of size  $16 \times 16$ . The regularization parameter  $\lambda$  for joint learning are chosen such that about 16 non-zero coefficients are retained for reconstructing the joint patches of the training.

components of PCA or predefined ones. Comparing the two DL methods, ODL is better than KSVD. When using more non-zero coefficients, errors by DL methods saturate at nonzero values. Since DL provides only an approximate solution. Wavelet transform and PCA give zero NRMSEs when using all coefficients because the transforms are exact.

The above comparisons demonstrate the advantages of representations using learned dictionaries over predefined ones. Comparing redundant and orthogonal representations, i.e. DL versus PCA, redundant dictionaries represent the fields more efficiently. They achieve the same level of error using less atoms. This suggests also that the sparsity and redundancy priors can be good candidates to help solving the ill-posed inverse problem of reconstructing HR fields from LR ones. Comparing the two common DL methods, ODL shows clear advantages both in term of efficient representation and computation effort and will be used in the rest of this chapter.

### 3.5.3 Reconstruction of high resolution fields- subsampling cases

To compare to other methods, LR fields are first subsampled from HR ones by a factor of  $4 \times 4$ , i.e.  $y_t = \mathbf{S}_s z_t$ , where  $\mathbf{S}_s : \mathbb{R}^P \mapsto \mathbb{R}^Q$ ,  $P/Q = 4 \times 4$ . Coupled LR and HR patches are extracted to train the dictionaries. Due to direct subsampling, the aliasing, which has not been addressed in previous works, will play an important role. This section investigates the ability of dictionary learning approaches to handle this aliasing problem.

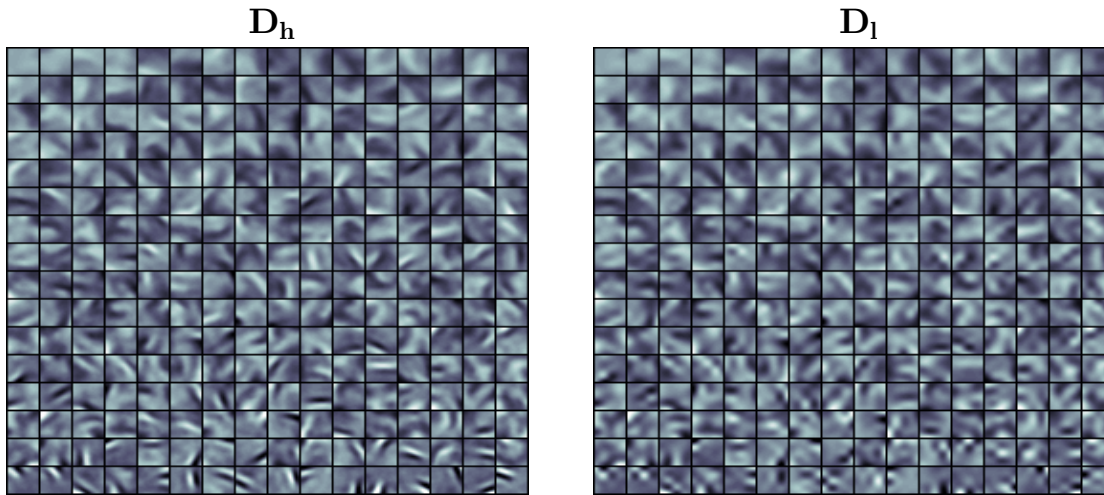


Figure 3.5 – Coupled dictionary of high and low resolution patches, where LR patches are of size  $4 \times 4$ , directly subsampled from their equivalent HR patches of size  $16 \times 16$ . The regularization parameter  $\lambda$  for joint learning are chosen such that about 16 non-zero coefficients are retained for reconstructing the joint patches of the training.

### Learning step

ODL (Mairal et al., 2010) is used to learn a joint dictionary of HR and LR from the LTHS planes. The subsampling ratio in space is  $P/Q = 4 \times 4$ , while the training samples are taken every  $N/M = 6$  snapshots in streamwise direction. From a total of  $37 \times 16$  training planes, small patches of size  $4 \times 4$  at LR are extracted via the extraction operator  $\mathcal{R}_l^k$ , coupled with HR patches of size  $16 \times 16$  by  $\mathcal{R}_h^k$ . The choice of parameters is discussed in section 3.5.1.

Three approaches *SR1*, *SR2* and *SR3* (table 3.1) are investigated. The interpolation step in *SR2* plays only the role of transforming the field from LR to the same dimension as the HR one. The content should be rather the same, since the interpolation does not introduce any small-scale information. However, starting from the interpolated fields can bring the benefit of having a good large-scale information *a priori*. The model will focus on small scales only. This approach has shown to be beneficial in many image processing problems.

Figure 3.4 shows dictionaries for HR and LR patches and figure 3.5 shows dictionaries for HR and interpolated LR patches. In both figures, atoms are sorted according to their variances  $\|\mathbf{a}_i\|_2^2$  when learning the dictionary (from the top-left). The most significant atoms contain mostly large scales, while less significant ones represent high-frequency contents. Coupled atoms show similar patterns. The relation between LR and HR patches are now encoded in the relation between LR and HR dictionaries. The assumption of coupled representations in this case also implies that LR atoms are approximately subsampled from the HR ones as shown in equation 3.26.

### Reconstruction step

Having learned dictionaries  $\{\mathbf{D}_h, \mathbf{D}_\ell\}$  at hand, from given  $\mathbf{y}^e$  different from the training data, the HR field  $\mathbf{z}^e$  is reconstructed by using algorithm 4.  $\mathbf{y}^e$  is also assumed to be directly subsampled from  $\mathbf{z}^e$  the same way as in the training data, i.e.  $\mathbf{y}^e = \mathbf{S}_s \mathbf{z}^e$ . From  $\mathbf{y}^e$ , all LR patches  $\mathbf{P}_l^e$  with one-pixel overlapping are extracted. The sparse code is estimated by solving the optimization problem in equation 3.43.  $\mathbf{A}^e$  is then used to reconstruct HR patches as  $\hat{\mathbf{P}}_h^e = \mathbf{D}_h \mathbf{A}^e$  and put back into the global scene of  $\hat{\mathbf{z}}^e$ .

LR and HR patches of sizes  $q = 4 \times 4$  and  $p = 16 \times 16$  respectively are extracted from all training planes. The sparse code  $\mathbf{A}^e$  is estimated from LR patches. Each row has at most  $\|\mathbf{A}^e\|_0 = q = 16$  nonzero coefficients. It means that HR patches are reconstructed from maximum 16 atoms within  $\mathbf{D}_h$ , a strong constraint on the reconstruction accuracy (see figure 3.3). The nature of the data also affects the accuracy in the sense how good is the Sparse-Land prior in this case of turbulence. Also, this constraint addresses the problem of designing more efficient representations of the data to reduce the error when using the same number of atoms.

Using all three approaches presented in table 3.1, the models can reconstruct the fields in the same accuracy as the spline interpolation but not better. This is due to the severe situation where the presence of aliasing makes equation 3.26 a very crude assumption. The subsampling of a small atoms brings a very strong aliasing effect that coupled dictionaries could not efficiently handle. The next section will study the capability of the present approach when the aliasing problem is absent from the LR data.

### 3.5.4 Reconstruction of high resolution fields- the downsampling case

This section investigates the possibility of the current approach to estimate the HR fields given the LR ones in the case of downsampling, i.e. with anti-aliasing prefiltering:

$$\mathbf{y}_t = \mathbf{S}_s \mathbf{L}_s \mathbf{z}_t \quad (3.47)$$

where  $\mathbf{S}_s$  and  $\mathbf{L}_s$  are the spatial subsampling and low pass filter respectively. Bicubic filter in Matlab built-in function *imresize* is used for the prefiltering and interpolation step. It was shown also in section 3.4.2 that the assumptions Sparse-Land model at HR also lead to the relation  $\mathbf{P}_\ell = \mathbf{S}_s^e \mathbf{L}_s^e \mathbf{P}_h$  between LR and HR dictionaries.  $\mathbf{S}_s^e$  and  $\mathbf{L}_s^e$  are local versions of  $\mathbf{S}_s$  and  $\mathbf{L}_s$  applying to patches.

We compare the three methods of coupled dictionary learning, the so-called *SRI*, *SR2* or *SR3*, presented in table 3.1. To recall *SRI* and *SR2* couple either LR or interpolated LR

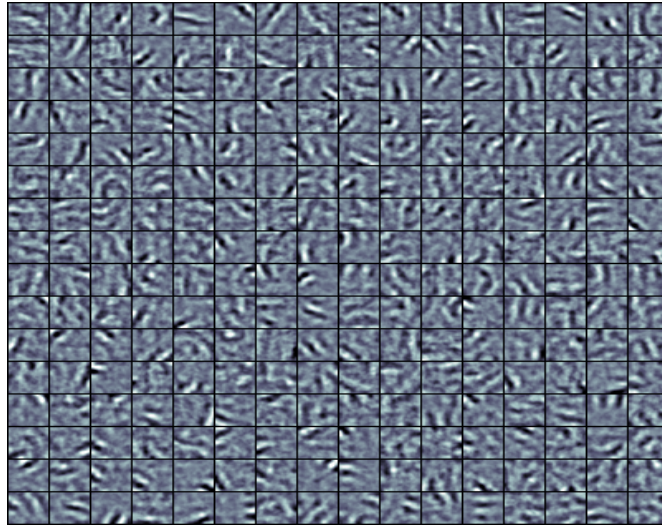


Figure 3.6 – Dictionary of the residual between HR and interpolated LR.

patches with HR ones, while *SR3* couples the residuals with the features of derivatives. The procedure follows the previous section of subsampled fields, with learning parameters in table 3.2. HR patches  $\mathbf{P}_h$  are extracted from LTHS fields and coupled with LR patches  $\mathbf{P}_\ell$ . The dictionaries are trained *offline* from  $\{\mathbf{P}_h, \mathbf{P}_\ell\}$ , and used in *online* reconstruction stage for all HTLS measurements to reconstruct HTHS fields.

We use three quantities for comparisons: the average NRMSE between reconstructed and reference velocity fields estimated using equation 2.28, the 2D energy spectra of the velocity fields and of the errors. These three quantities give a complete view to qualify different approaches. Only the most difficult planes, which are equally far from LTHS measurements, are used to estimate the errors. Error estimated using these planes will better represent the generalization capability of the approaches.

Figure 3.7 shows the average NRMSEs of different reconstructions, either interpolation or reconstruction by coupled dictionaries using *SRI*, *SR2* or *SR3* approaches. The three DL models reduce NRMSEs by 11.99%, 12.73% and 13.86% respectively compared to the simple interpolation. *SR3* gives the most accurate reconstructions by coupling the residuals, essentially contain only small scales, with derivatives of large-scale structures.

To further understand the quality of the reconstructed fields at different scales, figure 3.8 shows the 2D spectra of reference fields and different reconstructed ones. All methods capture good large scales till about  $0.5k_c$ , where  $k_c$  is the cutoff wave number defined by the subsampling ratio. The interpolation, starting from the downsampled fields with prefiltering step to avoid aliasing, loses already energy at large scales and capture almost no small scales. By coupling the dictionaries, the reconstructed fields recover the large-scale information with some small scales. If considering only the scales between  $0.5k_c$

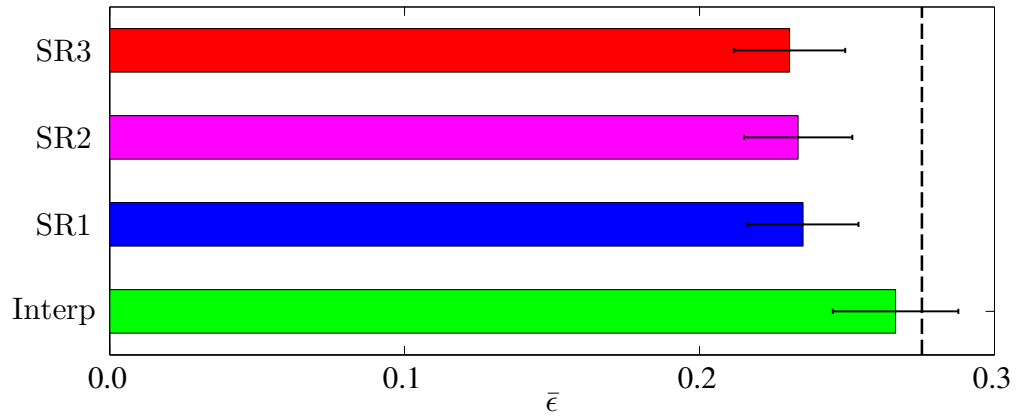


Figure 3.7 – Means and standard deviations of NRMSEs estimated between reference and reconstructed fields of all middle planes (at the center of blocks bounded by the two LTHS planes). The reconstructions are by spline interpolation and super-resolution using three different methods: *SR1*, *SR2* and *SR3* (see table 3.1). NRMSEs are  $0.267 \pm 0.021$ ,  $0.235 \pm 0.019$ ,  $0.233 \pm 0.018$  and  $0.230 \pm 0.019$  respectively. The NRSME of spline interpolation in the equivalent subsampling case is 0.276 (dashed black line).

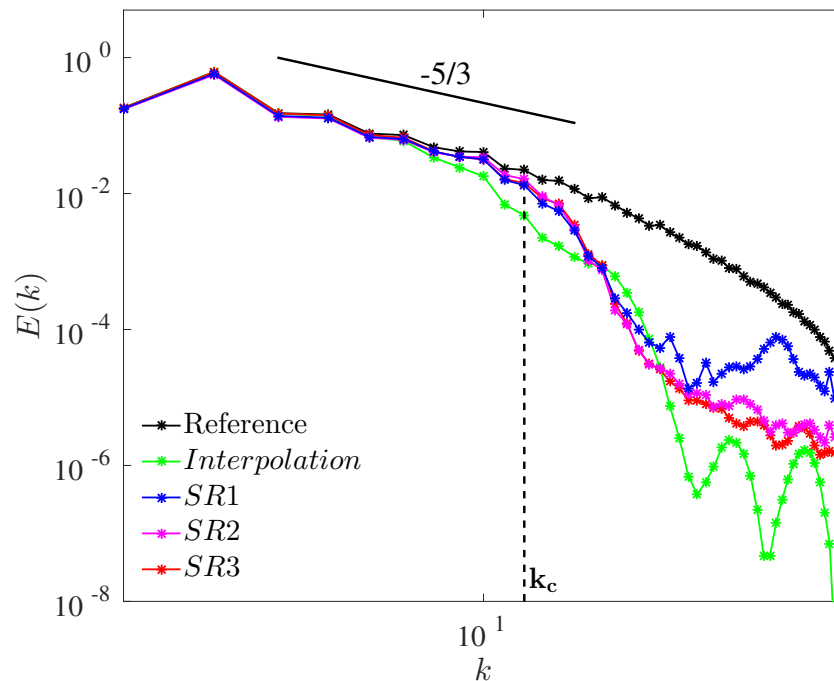


Figure 3.8 – 2D spectra of all planes used to computed the NRMSEs in figure 3.7, from reference, interpolation and SR by three different methods: *SR1*, *SR2* and *SR3* (see table 3.1). For scales from  $0.5k_c$  to  $1.5k_c$ , energy losses of SR fields compared to reference ones is 24%, 22% and 21% respectively, while that of interpolation is 85%.



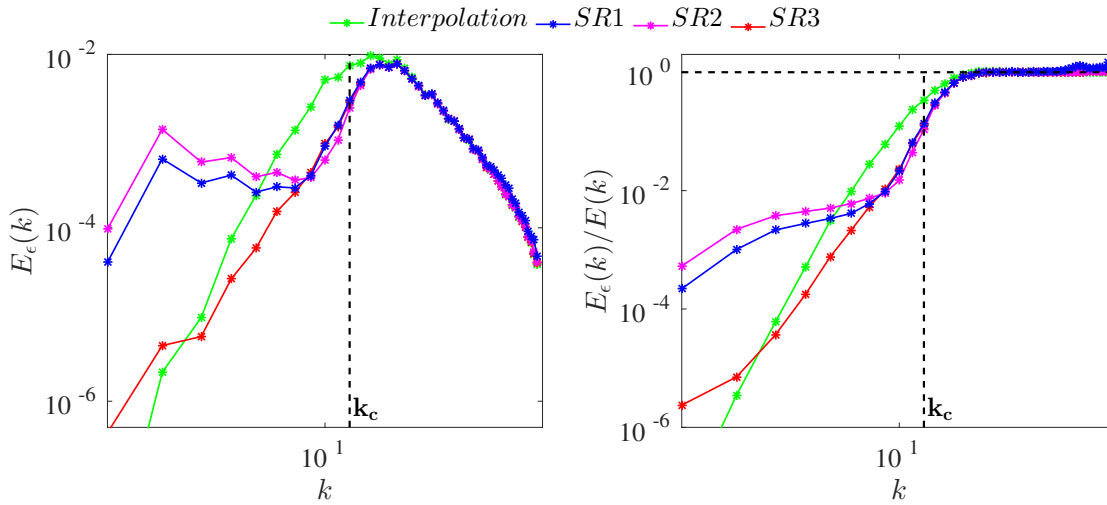


Figure 3.9 – (Left) 2D spectra of errors, which are the different between the reference and reconstruction by interpolation and three different SR methods as described in 3.4.3. (Right) 2D spectra of errors normalized by the energy spectrum of the reference (the black curve in figure 3.8).

and  $1.5k_c$ , the energy loss of interpolated fields is 85%, while those are 24%, 22% and 21% for  $SR1$ ,  $SR2$  and  $SR3$  respectively. The benefit of SR is significant in this most interesting range of scales. Larger than  $1.5k_c$ , all reconstructions are not reliable. Smaller than  $0.5k_c$ , spectra of all reconstructed fields are already very accurate.

Figure 3.9 (left) shows the spectra of the errors, which are the differences between the reference and reconstructed fields by all methods. Errors are very low at small wave numbers, reach a maximum around  $k_c$  and reduce at higher wave numbers. Integrals of these curves give the mean-square errors of each reconstruction method. To better qualify the error at each scale, the curves are normalized by the energy spectrum of the reference (the black curve in Figure 3.8). The normalized spectra of errors are interpreted as the percentage of error at each scale. The interpolation gives very small relative errors at large scales, but grow rapidly near  $k_c$ . Error spectra of all SR methods collapse at  $0.5k_c$  and reach a maximum of 100% error at around  $1.5k_c$ . Smaller than  $0.5k_c$ ,  $SR1$  and  $SR2$  gives slightly higher errors compared to  $SR3$ .  $SR3$  is more accurate since it starts from the interpolated large scales, while those information are re-estimated by  $SR1$  and  $SR2$ . Also,  $SR3$  focuses more on the missing information of small scales by using derivatives as features and following the patterns of larger ones.

From the above comparisons, coupled dictionary approaches demonstrate clear benefits compared to the simple interpolation, with errors reduced by about 15%. From the energy spectra or spectra of errors, the benefits mostly come from the range of scales between  $0.5k_c$  and  $1.5k_c$ , where  $k_c$  is the cutoff defined by the grid of LR measurements. The loss of energy in this range is reduced from 80% with interpolation to 20% by DL

approaches. The spectra of the errors also show that most of benefits come from this range of scales, before reaching 100% at around  $1.5k_c$ .

### 3.6 Concluding remarks

This chapter has discussed the possibilities of applying dictionary learning, a successful approach in the field of signal and image processing, to turbulence studies. The method finds a representation for the data by generalizing principal component analysis to sparse representation in a redundant dictionary. *Redundancy* means that the number of atoms can be larger than the dimension of input vectors, ignoring the orthogonality constraint of PCA. This implies the *sparsity*, i.e. the representation is composed by a linear combination of only a few atoms. These properties make the learned dictionary a more adaptive representation of the data. Sparsity can also play the role of a prior about the system when solving the inverse problem of HR field reconstruction.

To investigate the efficiency of DL in representing the data, the learned dictionaries by this approach have been compared against PCA and predefined wavelet dictionaries. Reconstruction errors as functions of sparsity, i.e. the number of atoms used, are shown as the measure of efficiency. Adaptive dictionaries show some superiority compared to the predefined ones. The benefits are mainly from the high sparsity levels, when less than half the number of coefficients are non-zero.

DL is then used to reconstruct HR velocity fields from LR measurements. The approach is called *coupled dictionary learning*, inspired from the single image super-resolution application Yang et al., (2010) and Zeyde et al., (2012). By learning coupled representations of LR and HR fields from the data, a nonlinear relation is established and generalized to perform the reconstruction task. With the same idea, different pre-processing techniques are tested. The coupling can be between HR and LR fields or their interpolation. Another approach focuses more on the small-scale information by coupling the small scales with the derivatives of interpolated fields.

The first attempt is for the configuration where LR fields are directly subsampled from HR ones without any anti-aliasing prefiltering step. This is *a priori* not a favorable case due to the presence of aliasing terms. However, it is worth studying since this setup mimics what would happen in a real experiment. It is interesting also to see how well learned dictionaries can handle aliasing. Results show that DL is inoperative to recover some small scales on top of the interpolated large-scale information.

To understand whether the failure comes from the approach or from the aliasing, another case is investigated where LR fields are downsampled from the HR ones with a prefiltering step. The same approach with identical parameters is used. Results

show significant improvements of reconstruction accuracy, with about 15% reduction of NRMSEs compared to interpolation. Spectral analyses also show that most benefits are at the frequency range of  $0.5k_c$  to  $1.5k_c$ , where  $k_c$  is the cutoff wave number corresponding to the downsampling ratio. In term of energy, simple interpolation loses 85% in this range scales, while this loss is reduced to about 20% with DL approaches.

The above results demonstrate the capability and limitations of DL approaches in solving the reconstruction problem in turbulence. The corresponding prior, which is the duality between sparsity and redundancy, is robust. The first attempt has not succeeded to reconstruct HR fields from direct subsampled LR ones due to the aliasing problem. This attempt addresses also the question on designing a good sensing system or post-processing techniques to deal with aliasing terms if exist.

This chapter has presented a similar configuration as regression models to learn the mapping function between large and small scales. However, the learning is localized thanks to the patch-wise approach. This property of dictionary learning approach can be beneficial in other configurations where the training and testing samples are of different scenes. For example, the training HR fields could be a small region of the whole field, while LR one can be larger. In such case, dictionary approach is the only candidate among all methods presented in this thesis.



# Non-local similarity-based propagation model

This chapter proposes a novel model that combines low-resolution measurements in space and time in order to reconstruct fully resolved turbulent fields. This method tackles the problem of fusing high-time-low-space and low-time-high-space resolution measurements mentioned in section 1.3.2. The model exploits local structures in large scale measurements to reconstruct small scales. The idea is to assume that small scales are essentially advected by large scales. This is based on the scale similarity hypothesis, where information at adjacent ranges of scales are strongly correlated. The model is further developed from the non-local means (NLM) denoising filter (Buades et al., 2005), which is very simple but efficient, and widely used in image processing.

The proposed model is applied to the streamwise velocity fields from DNS data of isotropic turbulence as described in section 1.4.2. Due to the absence of time resolved fields, *the streamwise direction is associated to the time dimension*, while the other two spatial directions are the spanwise and the vertical ones. Within this chapter, parameters of the model will be optimized. Model performances are also investigated by fixing one subsampling ratio in either space or time (spanwise) while varying the other. Detailed analysis of model performances are presented, but further comparisons with other methods will be investigated in chapter 6.

## 4.1 Non-local means

Non-local means (NLM) was originally proposed by Buades et al., (2005) to deal with the single image denoising problem. It has been then extended to solve other inverse problems of image reconstruction such as inpainting (to fill in missing pixels) or super-

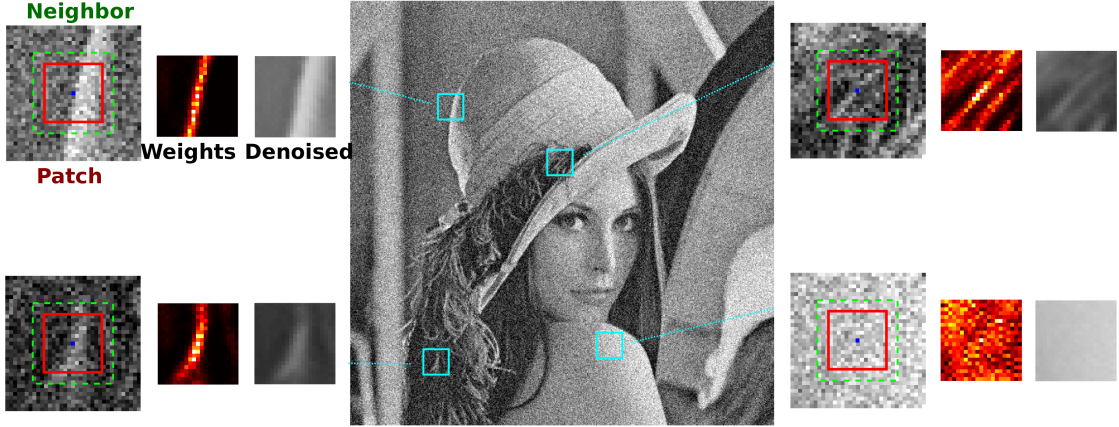


Figure 4.1 – Image denoising using non-local means. Four sample patches in red extracted from noisy image and centered at blue dots. The estimate of each point are weighted average of its neighbors, and weights are computed as the similarity of the moving patches compared to the reference patch. Photo credit: Foi and Boracchi, (2016).

resolution (to increase image resolution). The model assumes that image content is likely to repeat in many regions. The estimate of each pixel is a weighted average of its neighbors assigned with different weights. These weights are estimated as the similarity of the denoised/inpainted/super-resolved pixel with its neighbors. By proposing several estimates of a single pixel, NLM yields redundancy by using several observations of similar scenes obtained from different locations within the same image. These multiple observations of underlying ground truth help in solving the ill-posed inverse problem.

#### 4.1.1 Non-local means filter for denoising

The NLM denoising filter is simple but rather robust. Figure 4.1 illustrates how the NLM filter works on a sample image. Denoting  $z_t \in \mathbb{R}^P$  as the column vector of a 2D noisy image, the NLM filter denoises the  $k$ -th pixel as a weighted average of its neighboring noisy pixels:

$$\hat{z}_t[k] = \frac{\sum_{i \in \mathcal{N}_k} w[k, i] z_t[i]}{\sum_{i \in \mathcal{N}_k} w[k, i]} \quad (4.1)$$

$\mathcal{N}_k$  is the set of neighboring pixels of the  $k$ -th pixel. The local weight coefficient  $w[k, i]$  is the similarity between the  $k$ -th and  $i$ -th pixels:

$$w[k, i] = \exp\left(-\frac{\|\mathcal{R}_s^k z_t - \mathcal{R}_s^i z_t\|_2^2}{2\sigma^2}\right) \quad (4.2)$$

where the operator  $\mathcal{R}_s^k : \mathbb{R}^P \mapsto \mathbb{R}^P$  extracts 2D patches of size  $\sqrt{p} \times \sqrt{p}$  centered at the  $k$ -th pixel. These patches are then arranged in a column vector in lexicographical order. When  $p = 1$ , the NLM filter becomes a bilateral one. The global filter parameter  $\sigma$  regulates the decay of the exponential expression. It controls the contribution of the  $i$ -th pixel on the estimation of the  $k$ -th pixel. Geometrical priority can also be introduced, where high weights are given to closer neighbors than further ones.

### 4.1.2 Generalized non-local means for super-resolution

The idea of the NLM filter has been generalized to perform video super-resolution (Protter et al., 2009) by introducing time dimension. The aim is to estimate HR sequence of images  $z_t \in \mathbb{R}^P$ ,  $t = 1, 2, \dots, N$ , from corresponding LR sequence of images  $y_t \in \mathbb{R}^Q$  ( $Q < P$ ). NLM exploits the self-similarity property of natural images in the sequence. The model assumes that common scenes are shared by many snapshots in the sequence. Derivation and mathematical explanation can be found in Protter et al., (2009). The closed form solution is simplified, where each HR pixel is estimated as a weighted average of neighboring LR pixel in a space-time window:

$$\hat{z}_{t_o}[k] = \frac{\sum_{t \in \mathcal{N}_{t_o}} \sum_{i \in \mathcal{N}_k} w[k, i, t] y_t[i]}{\sum_{t \in \mathcal{N}_{t_o}} \sum_{i \in \mathcal{N}_k} w[k, i, t]} \quad (4.3)$$

where  $\mathcal{N}_{t_o}$  is the set of neighboring snapshots in time of the  $t_o$ -th one. The weight  $w[k, i, t]$ , defined as the probability of the  $k$ -th HR pixel estimation coming from the  $i$ -th LR pixel, is estimated as:

$$w[k, i, t] = \exp\left(-\frac{\|\mathcal{R}_s^k y_{t_o} - \mathcal{R}_s^i y_t\|_2^2}{2\sigma^2}\right) \quad (4.4)$$

This solution exists when there is at least one non-zero weight  $w[k, i, t]$ .

It is natural to test this idea to reconstruct HTHS turbulent fields  $\mathbf{Z}$  of size  $N \times P$  from the corresponding HTLS measurements  $\mathbf{Y}$  of size  $N \times Q$  ( $Q < P$ ) subsampled directly from  $\mathbf{Z}$ . However, the NLM model by Protter et al., (2009) is not able to improve the reconstruction accuracy compared to the single interpolation of HTLS velocity fields. This is potentially due to the differences between video of natural images and turbulence. The generalized NLM was originally illustrated most successful for video of human with small motions, where the same scene is obtained for successive snapshots. Turbulent fields are fundamentally different, where no sharp edge exists, sizes of structures make a continuous spectrum, and all scales move and distort randomly. Also, since turbulence fields have a decreasing power law spectrum, large scales contain most of the kinetic

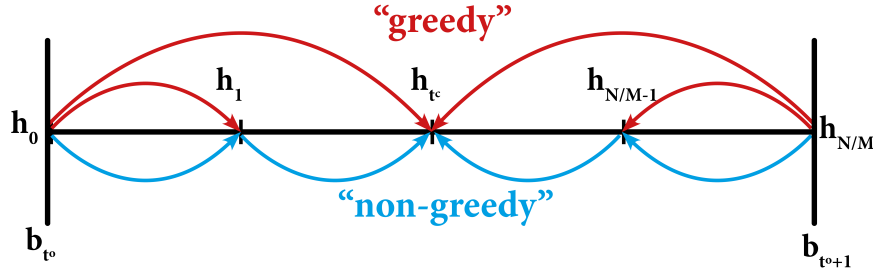


Figure 4.2 – A sketch of *greedy* and *non-greedy* propagation models. The two LTHS planes are at the border including small scales  $\mathbf{b}_{t_0} = \mathbf{h}_0$  and  $\mathbf{b}_{t_0+1} = \mathbf{h}_{N/M}$  defined in equation 4.5. All small scales to be estimated from other planes  $\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{P/Q-1}$  will be propagated either directly from LTHS planes (greedy, red arrows) or in a plane-by-plane manner (non-greedy, blue arrows). To simplify the plot, the propagation is shown from LTHS planes to the central plane only.

energy. The reconstruction of these information plays the dominant role in total error. The current NLM model re-estimates large scales, which are probably not as good as by spline interpolation.

## 4.2 Similarity-based model to propagate small scales

We propose a similarity-based model in the spirit of the NLM model for video super-resolution with important modifications. Since large scales are well reconstructed also with spline interpolation, we leave these information unchanged while propagating small scales only. Two models, the so-call “*greedy*” and “*non-greedy*” propagations, are proposed. Figure 4.2 illustrates the idea of the two propagation schemes. While the greedy model propagates small scales from LTHS planes directly toward other instant, the non-greedy one does the propagation plane-by-plane.

### 4.2.1 Greedy propagation of small scales from key planes

LTHS measured snapshots are crucial to recover some spatial HF contents, since these are the only place where small scales are available. In the configuration presented in figure 1.5, HTLS snapshots give only information of large scales in space, while small ones are only accessible from LTHS snapshots. The idea is to bring these small-scale information and aggregate them on top of the measured large scales. The propagation model is based on the similarity of large scales at different instants, while assuming that small scales are advected by large ones.

LTHS snapshots  $\{\mathbf{x}_{t_0}\} \in \mathbb{R}^P, t_0 = 1, 2, \dots, M$ , also called *key frames*, include both large



and small scales. HTLS snapshots  $\{\mathbf{y}_t\} \in \mathbb{R}^Q, t = 1, 2, \dots, N$  contain only large scales. The propagation model aims at propagating the small-scale content from the key frames and aggregating with the interpolation  $\mathbb{I}_s \mathbf{y}_t$  to reconstruct  $\hat{\mathbf{z}}_t$ , which contains information of all scales.  $\mathbb{I}_s : \mathbb{R}^Q \mapsto \mathbb{R}^P$  is the 2D cubic spline interpolator to reconstruct large scales from LR measurements.

The propagation is based on the similarity of the large scales between the key frames and the measured planes to reconstruct. Large and small scales could be separated from the key frames using ideal Fourier filter. However, these large scales are not consistent with those from the measured planes obtained by interpolations. It is more consistent to subsample and re-interpolate LTHS fields the same way as HTLS ones. Small scales of each key frame are then defined as:

$$\mathbf{b}_{t_0} = \mathbf{x}_{t_0} - \mathbb{I}_s \mathbf{S}_s \mathbf{x}_{t_0} \quad (4.5)$$

$\mathbf{S}_s$  is the subsampling operator in space,  $\mathbf{S}_s : \mathbb{R}^P \mapsto \mathbb{R}^Q$ , and  $\mathbf{b}_{t_0}$  should contain some aliasing. The greedy propagation model estimates small scales  $\hat{\mathbf{h}}_t$  at the  $t$ -th time step from the given small scales  $\mathbf{b}_{t_0}$  and  $\mathbf{b}_{t_0+1}$  of the two bounded LTHS planes, ones just before and after the  $t$ -th HTLS plane in time:

$$\hat{\mathbf{h}}_t[k] = \frac{\sum_{i \in N_k} w_0[k, i, t] \mathbf{b}_{t_0}[i] + \sum_{i \in N_k} w_1[k, i, t] \mathbf{b}_{t_0+1}[i]}{\sum_{i \in N_k} w_0[k, i, t] + \sum_{i \in N_k} w_1[k, i, t]} \quad (4.6)$$

The weight  $w_0[k, i, t]$  measures the similarity of  $k$ -th pixel of  $\hat{\mathbf{h}}_t$  and the  $i$ -th pixel of  $\mathbf{b}_{t_0}$ , and analogously for  $w_1[k, i, t]$ . It is estimated as the similarity of large scales in the two snapshots:

$$w_0[k, i, t] = \exp\left(-\frac{\|\mathcal{R}_s^k \mathbb{I}_s \mathbf{y}_t - \mathcal{R}_s^i \mathbb{I}_s \mathbf{S}_s \mathbf{x}_{t_0}\|_2^2}{2\sigma^2}\right) \quad (4.7)$$

The term  $\mathbb{I}_s \mathbf{S}_s \mathbf{x}_{t_0}$  is the conjugate large scales of  $\mathbf{b}_{t_0}$ , i.e.  $\mathbf{x}_{t_0} = \mathbb{I}_s \mathbf{S}_s \mathbf{x}_{t_0} + \mathbf{b}_{t_0}$ .  $w_1[k, i, t]$  is estimated analogously. The final reconstruction of all scales is:

$$\hat{\mathbf{z}}_t = \mathbb{I}_s \mathbf{y}_t + \hat{\mathbf{h}}_t \quad (4.8)$$

Since  $\mathbb{I}_s \mathbf{y}_t$  contains some aliasing, so does  $\hat{\mathbf{h}}_t$  due to the presence of aliasing in  $\mathbf{b}_{t_0}$  and  $\mathbf{b}_{t_0+1}$ , we could hope the two aliasing terms cancel and make the estimate of  $\hat{\mathbf{z}}_t$  more accurate. This can explain also the benefit when considering  $\mathbf{b}_{t_0}$  as the residual of the interpolation instead of ideal small scales.

The propagation in the formulation 4.6 is point-wise, where each point is estimated

as a weighted average of its neighbors. Instead, a small patch centered at  $i$ -th pixel of  $\mathbf{b}_{t_0}$  and  $\mathbf{b}_{t_0+1}$  can be accumulated over the patch centered at the  $k$ -th pixel of  $\hat{\mathbf{h}}_t$  using the same weights  $w_0[k, i, t]$  and  $w_1[k, i, t]$ . Since the same pixel belongs to several patches, the overlapping as in the patch-based approach (section 3.4.1) is used. The overlapping will ensure the continuity of neighboring points and increase the consistence of all existing structures. There will be also more degree of freedom, where the number of estimates at each location increases significantly. The accumulation of scales and weights are as:

$$\begin{aligned}\hat{\mathbf{h}}_t &= \sum_{k=1}^P \left( (\mathcal{R}_a^k)^\top \left( \sum_{i \in N_k} w_0[k, i, t] \mathcal{R}_a^i \mathbf{b}_{t_0}[i] + \sum_{i \in N_k} w_1[k, i, t] \mathcal{R}_a^i \mathbf{b}_{t_0+1}[i] \right) \right) \\ W_t &= \sum_{k=1}^P \left( (\mathcal{R}_a^k)^\top \mathcal{R}_a^k \mathbf{1} \left( \sum_{i \in N_k} w_0[k, i, t] + \sum_{i \in N_k} w_1[k, i, t] \right) \right)\end{aligned}\quad (4.9)$$

where  $\mathcal{R}_a^k$  and  $\mathcal{R}_a^i$  extract a small patch of size  $\sqrt{r} \times \sqrt{r}$  centered at the  $k$ -th and  $i$ -th pixel respectively. The transpose operator  $(\mathcal{R}_a^k)^\top$  is to put back this small patch into the large field, centered at  $k$ -th pixel.  $\mathbf{1}$  is the vector of size  $P \times 1$  of ones. The term  $(\mathcal{R}_a^k)^\top \mathcal{R}_a^k \mathbf{1}$  puts ones at the position of each patch. The whole procedure is to accumulate the sum of all small scales estimates and their corresponding weights. The final step is the normalization:

$$\hat{\mathbf{h}}_t[k] \leftarrow \frac{\hat{\mathbf{h}}_t[k]}{W_t[k]} \quad (4.10)$$

## 4.2.2 Non-greedy propagation of small scales

The greedy model aims at propagating directly small scales from the two key frames to the HTLS planes. The performance of such a model is very limited when the similarity level decays rapidly with time far from the LTHS planes. A non-greedy propagation of small-scale information plane-by-plane from the two neighboring LTHS fields toward further planes is expected to improve the reconstruction.

The main step to propagate the small scales from the two key frames at  $\mathbf{b}_{t_0}$  and  $\mathbf{b}_{t_0+1}$  is presented in algorithm 5.  $g(t, \mathbf{h}_1, \mathbf{h}_2)$  is denoted as the similarity-based estimation of small-scale information at  $t$ -th plane from the two bounded planes, with scales to propagate being  $\mathbf{h}_1$  and  $\mathbf{h}_2$ . The procedure starts by propagating small scales from these two frames to their first two neighboring planes,  $\mathbf{h}_1$  and  $\mathbf{h}_{N/M-1}$  in algorithm 5, following the formula 4.9. The next step considers these two estimated  $\mathbf{h}_1$  and  $\mathbf{h}_{N/M-1}$  as the new key frames and propagate them further. This procedure is repeated till the central planes that are equally far from  $\mathbf{b}_{t_0}$  and  $\mathbf{b}_{t_0+1}$ .

---

**Algorithm 5** Estimating small scales of  $N/M - 1$  planes by non-greedy NLM propagation from the two key frames at  $t_o$  and  $t_o + 1$

---

```

1: Input:  $\mathbf{h}_0 \triangleq \mathbf{b}_{t_o}, \mathbf{h}_{N/M} \triangleq \mathbf{b}_{t_o+1}$ 
2:  $t_c = \text{round}(N/2M)$  ▷ Centered snapshot
3: for  $i = 1, \dots, t_c - 1$  do
4:    $j = N/M - i$ 
5:    $\mathbf{h}_i = g(i, \mathbf{h}_{i-1}, \mathbf{h}_{j+1})$ 
6:    $\mathbf{h}_j = g(j, \mathbf{h}_{i-1}, \mathbf{h}_{j+1})$ 
7: end for
8: if  $\text{rem}(N/M - 1) == 0$  then ▷ Check if even, there are 2 centered snapshots
9:    $\mathbf{h}_{t_c} = g(t_c, \mathbf{h}_s[t_c - 1], \mathbf{h}_s[t_c + 2])$ 
10:   $\mathbf{h}_{t_c+1} = g(t_c + 1, \mathbf{h}_s[t_c - 1], \mathbf{h}_s[t_c + 2])$ 
11: else
12:   $\mathbf{h}_{t_c} = g(t_c, \mathbf{h}_{t_c-1}, \mathbf{h}_{t_c+1})$  ▷ Estimate centered snapshot
13: end if

```

---

$\sigma$	$\sqrt{p}$	$\sqrt{r}$	$\sqrt{s}$
0.05	9	1	5
0.1	17	3	11
0.2	25	13	17
0.4		25	

Table 4.1 – Set of parameters to do simple grid search (try out all possible combinations from this set): the global filter parameter  $\sigma$ , size of similarity patches  $p$ , size of accumulation patches  $r$ , and size of searching region  $s$ . The search is performed on the DNS data of isotropic turbulence, with the subsampling ratios are  $P/Q = 3 \times 3$  in space and  $N/M = 4$  in time.

## 4.3 Applying the propagation model to isotropic turbulence

### 4.3.1 The choice of model parameters

The propagation model uses four parameters (see equations 4.7 and 4.9), including the filter parameter  $\sigma$ , the size  $s$ ,  $p$ , and  $r$  of the searching region  $\mathcal{N}_s$ , the similarity patch extracted by  $\mathcal{R}_s$ , and the accumulation patch extracted by  $\mathcal{R}_a$  respectively. Optimization of these parameters is done by varying one while fixing the remaining three. The parameter-tuning is carried out by investigating the error between the reconstructed fields and the reference ones. These tests give an idea on the optimal set of parameters, which can be used in new situations where reference fields are not available.

Table 4.1 gathers all parameters to test. The DNS data of isotropic turbulence are

used. HTLS and LTHS measurements are virtually extracted from the fully resolved fields as discussed in section 1.3.2. The subsampling ratios are  $P/Q = 3 \times 3$  in space, equally distributed in spanwise and vertical direction, and  $N/M = 4$  in time (streamwise). These subsamplings correspond to energy losses of about 1% both in space and time (see table 1.2). The configuration is chosen to have a moderate loss of information to facilitate all numerical experiments. Other configurations with various losses are investigated later in this section.

Three measures are used to study the effects of different parameters on the reconstruction accuracy. One is the average normalized root mean square error (NRMSE)  $\bar{\epsilon}$ , which represents an overall measure of the reconstruction accuracy, as in equation 2.28. The two other quantities are the power spectra  $E(k)$  and the spectra of the error  $E^\epsilon(k)$ . Power spectra show streamwise energy contents, while error spectra show the squared errors at each frequency. These two measures give more insights about the model performances when reconstructing turbulent fields at different scales.

Figure 4.3 shows average NRMSEs for the four parameters as a function of  $t/\delta t$ , the time distance from the previous LTHS plane to the estimated one. Two LTHS planes are at  $t/\delta t = 0$  and  $t/\delta t = N/M = 4$ . Average NRMSEs are shown as bars, with the standard deviation of each estimate shown by the error bars. The average and standard deviation of NRMSEs at each  $t/\delta t$  are computed using  $37 \times 23$  estimates from all blocks between the two successive key frames. Parameters are varied by changing one and keeping the remaining three constant around a reference parameters set ( $\sigma = 0.1$ ,  $p = r = 25 \times 25$  and  $s = 11 \times 11$ ). This set is close to the optimal values used later in this section and chapter 6 when comparing with other models.

Figure 4.4 shows the 2D spectra of the errors  $E^\epsilon(k)$  normalized by the energy spectra of the reference  $E(k)$ . These spectra are estimated as an average of all planes at  $t/\delta t = 2$ , the most difficult planes to reconstruct. For all reconstructions, the spectra exhibit very small errors at large scales, which grow sharply when approaching the cutoff wave number defined by the spacing of the low sampling measurements. At very small scales, errors can reach 100%. This level of error means that no information has been reconstructed.

Among all parameters, the filter parameter  $\sigma$  is the most sensitive. The optimal value is  $\sigma \approx [0.1, 0.2]$ . Small values of  $\sigma$  narrow the searching region by shrinking weights of patches with low similarity levels to zero. Less estimates are taken into account when doing the weighted averaging. In the extreme situation  $\sigma = 0$ , the number of estimates is reduced to one pixel only, which is the center of the most correlated patch. The propagation model is simplified to the shifting of a single pixel. Otherwise, too large values of  $\sigma$  put weights on all the neighboring patches. Models with very large  $\sigma$  lead

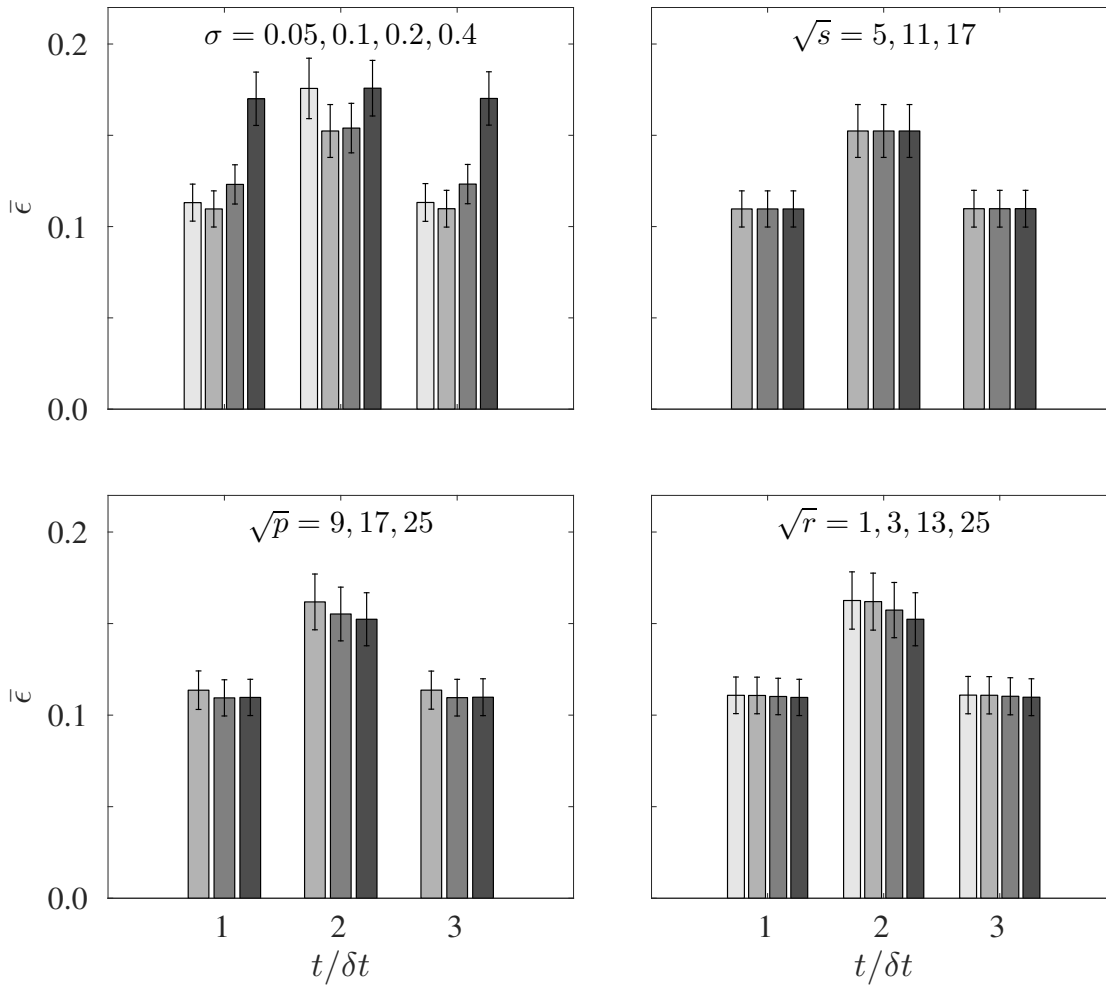


Figure 4.3 – Average NRMSEs over the whole dataset: each estimate of NRMSE is computed by comparing the reconstructed plane with the reference at each  $t/\delta t$ . The bars show the average NRMSEs, while the error bars are standard deviations of NRMSE estimates.

to a simple averaging with approximately equal weights. For these reasons, NRMSE is high for both small and large  $\sigma$ , while a moderate value of  $\sigma$  is a good compromise to minimize the reconstruction error. This is shown in the error spectra (figure 4.4) where the optimal  $\sigma$  gives similar large-scale reconstruction errors but lower ones at small scales.

Other model parameters are less sensitive. Searching region  $\mathcal{N}_k$  defines how large is the neighborhood to seek the small-scale information. However, the filter parameter  $\sigma$  plays a similar role by dumping the contribution of irrelevant patches to zero. Sizes of similarity and accumulation patches are more important. Similarity patches directly define the quality of weight estimates. Small patches are necessary to keep small-scale details of the flow. Larger patches improve the robustness of the models against noise and ill-posed conditions of the inverse problem. Ideally, we want the model to work well

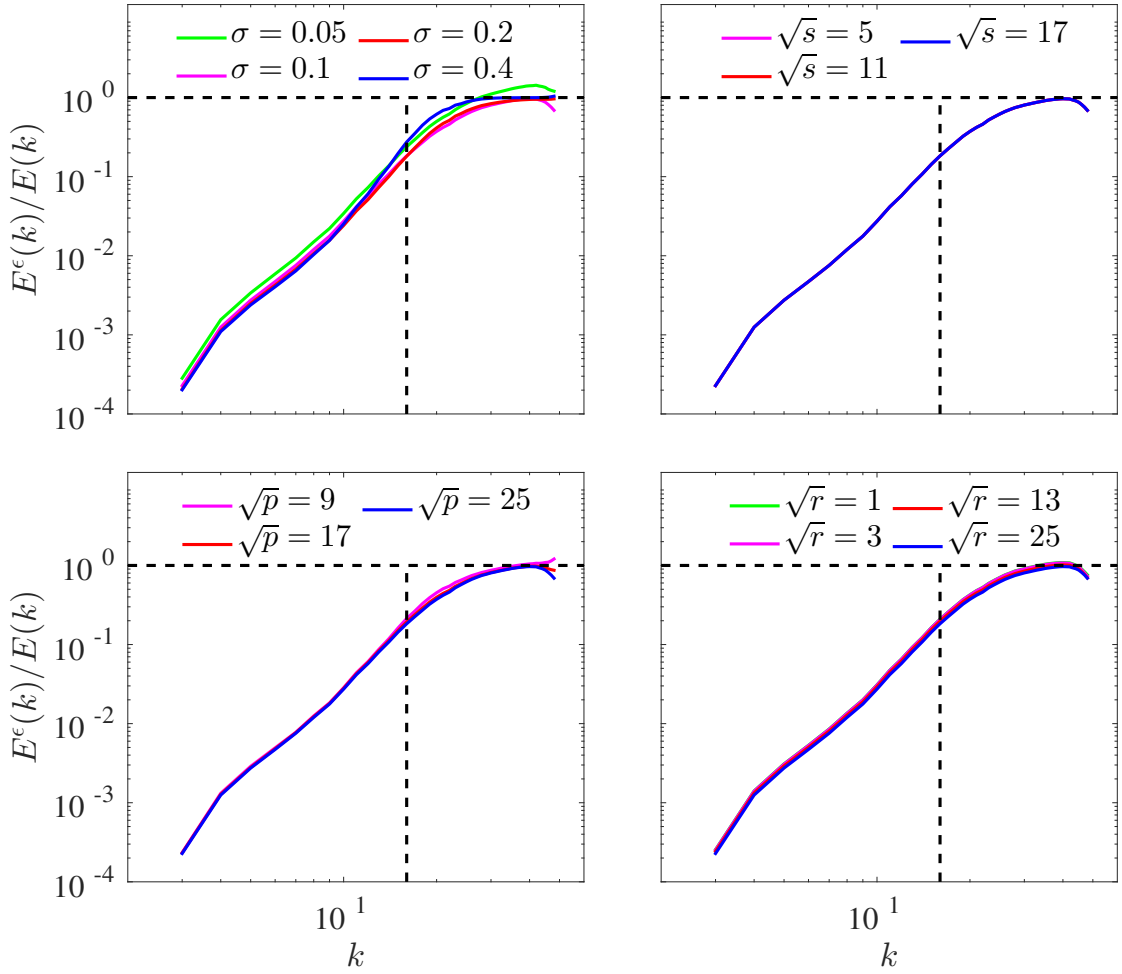


Figure 4.4 – Normalized error spectra of reconstructed planes at  $t/\delta t = 2$  (the most remote from the two neighboring LTHS planes). Three cases of varying  $p$ ,  $r$  and  $s$ , all curves are almost collapsed.

with small patches. However, experiments show slight improvements with larger patches. Similar results are observed when increasing the size of accumulation patches. Taking more pixels into the averaging helps in reducing the estimation noise, hence improving the reconstruction.

### 4.3.2 Comparing to spline interpolation

The proposed propagation models, either greedy or non-greedy algorithm in time, are compared to cubic spline interpolations either in time  $\mathbb{I}_t \mathbf{x}_t$  or in space  $\mathbb{I}_s \mathbf{y}_t$ . Apparently,  $\mathbb{I}_t \mathbf{x}_t$  contains no temporal small-scales information, while  $\mathbb{I}_s \mathbf{y}_t$  provides only large scales in space.

Figure 4.5 shows the mean NRMSEs by different methods as a function of time spacing to the previous LTHS plane  $t/\delta t$ . The error bars are also shown, telling how these

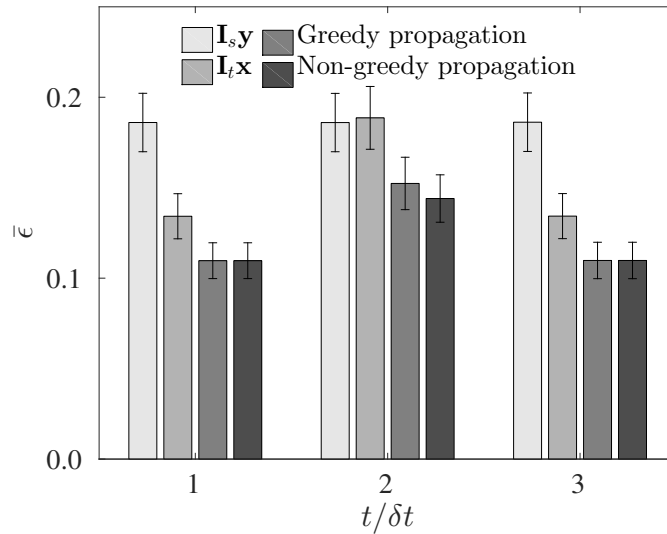


Figure 4.5 – Comparison of averaged NRMSEs as functions of time distances to the previous bounded LTHS plane by spatial/temporal interpolation, greedy and non-greedy propagation for the configuration where subsampling ratios are  $3 \times 3$  in space and 4 in time.

errors are different from one plane to another. These errors are estimated analogously as in figure 4.3. NRMSEs of reconstructed fields obtained by propagation models or time interpolation are necessarily small near the LTHS planes ( $t/\delta t = 1$  and  $t/\delta t = 3$ ), and increase toward the center. The error remains constant for spatial interpolation as it is only a function of the subsampling ratio in space. Greedy and non-greedy algorithms essentially give the same errors at the closest two planes from the LTHS ones. At these time steps, propagation models improve significantly (about 40%) compared to spatial interpolation. At the middle plane, errors by propagation increase, but remain 20% lower than both the interpolations. The non-greedy propagation model further decrease the error compared to the greedy one.

Figure 4.6 shows the spectral analysis of the reconstruction by different models. Energy spectra and error spectra are shown for the two positions, either close or far from LTHS planes. From the first two spectra, it is shown that spatial interpolation captures only large-scale information. Time interpolation gives an estimate with a good energy spectrum, while propagation schemes are subject to a certain energy loss at small scales. However, the error spectra shows that time interpolation gives good energy spectra with poorly reconstructed small scales. Propagation models yield compromise estimates. Certain amount of small scales are captured, while error spectra illustrate improvements at all frequencies compared to both interpolations.

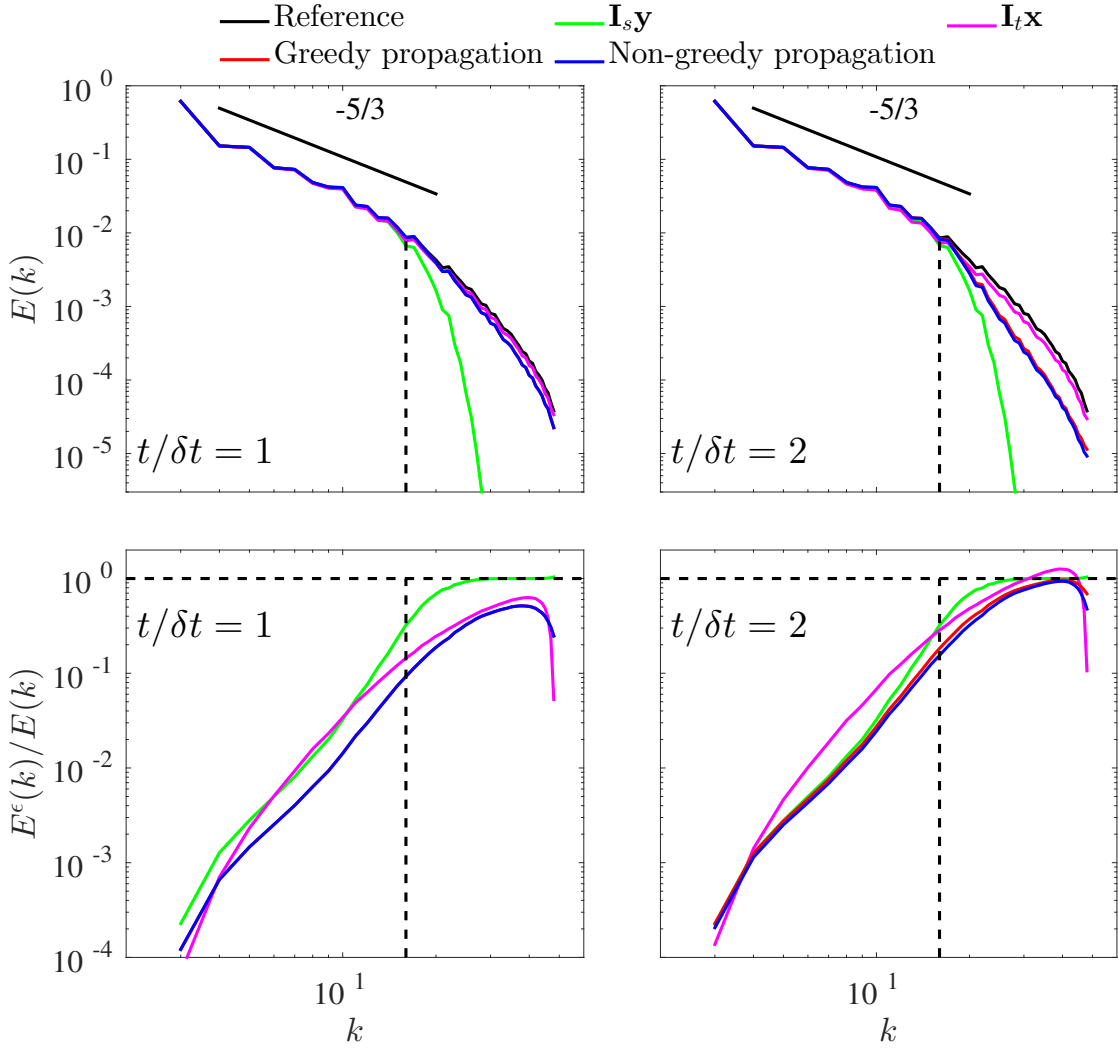


Figure 4.6 – Comparison of energy spectra and error spectra for various reconstructions by spatial/temporal interpolation, greedy and non-greedy propagation. Subsampling ratios are  $3 \times 3$  in space and 4 in time. The spectra are averaged over all planes of the same distance to the closest LTHS plane, i.e.  $t/\delta t = 1$  (left) or  $t/\delta t = 2$  (right).

### 4.3.3 Model performances in various configurations

To characterize the proposed propagation models, various configurations at different subsampling ratios in either space or time are investigated. Model performances depend on the ratio in space, since it defines how much information is missing. The performances also depend on the ratio in time, since small-scale information will necessarily be lost when similarity between large scales die off with increasing time spacings.

To test the model performances when increasing time distances between LTHS snapshots, the subsampling ratio  $P/Q = 3 \times 3$  in space, while varying those in time as  $N/M = 4, 6$  and  $8$ . The amount of missing information to recover due to the subsampling in space is about 1% of the total energy of the flow. Average NRMSEs as a function



Case	Subsampling ratios		Energy loss	
	$\sqrt{P/Q}$	$N/M$	$\Delta\kappa_s(\%)$	$\Delta\kappa_t(\%)$
1	3	4	1.03	1.23
2	3	6	1.03	3.56
3	3	8	1.03	6.53
4	3	4	1.03	1.23
5	4	4	2.63	1.23
6	6	4	7.29	1.23

Table 4.2 – Configuration parameters of six testing cases. The subsampling ratios of HTLS measurements are  $\sqrt{P/Q}$  and equal in both spatial directions. The ratios of LTHS measurements in time are  $N/M$ . The normalized energy losses in space  $\Delta\kappa_s$  and in time  $\Delta\kappa_t$  are defined in Eq. (1.2).

of the position in time are shown in figure 4.7. In all three cases, the central plane is at  $\Delta t_c/\delta t = 0$ , while other planes are at different distances  $\Delta t_c/\delta t$  to this plane, both negative (planes before) or positive (plane after).

In the figure 4.7, NRMSE for spatial interpolation remains constant, since it is independent of time. Errors by time interpolations or propagation models increase when moving toward the center. For time interpolations, these errors increase dramatically when the time subsampling ratio is high. The proposed propagation models are able to bring down the errors compared to interpolation for small to moderate distances between LTHS planes. For very large ones ( $N/M = 8$ , diamond shape), the propagated small scales even slightly downgrade the given large-scale information from the spatial interpolation.

Figure 4.8 shows NRMSEs of the proposed propagation models (both greedy and non-greedy), spatial and temporal interpolations when varying the subsampling ratio in space as  $P/Q = 3 \times 3, 4 \times 4$  and  $6 \times 6$  while keeping a fixed distance between LTHS planes of  $N/M = 4$ . Time interpolation gives the same error for all three cases, while errors of spatial interpolations are independent of the position in time. For a small subsampling ratio in time, temporal interpolation and propagation models are always better than spatial interpolation. For a small ratio of  $P/Q = 3 \times 3$ , the non-greedy propagation scheme give the most accurate reconstruction. In the case of moderate ratio ( $P/Q = 4 \times 4$ ), the energy loss in space is already three times larger than that in time (see table 4.2). Propagation models start with large scales from spatial interpolation and reconstruct small-scale information to give the same accuracy as time interpolation. For a large subsampling ratio of  $P/Q = 6 \times 6$ , propagation models improves significantly compared to spatial interpolation but remains worse than temporal interpolation. This is due to

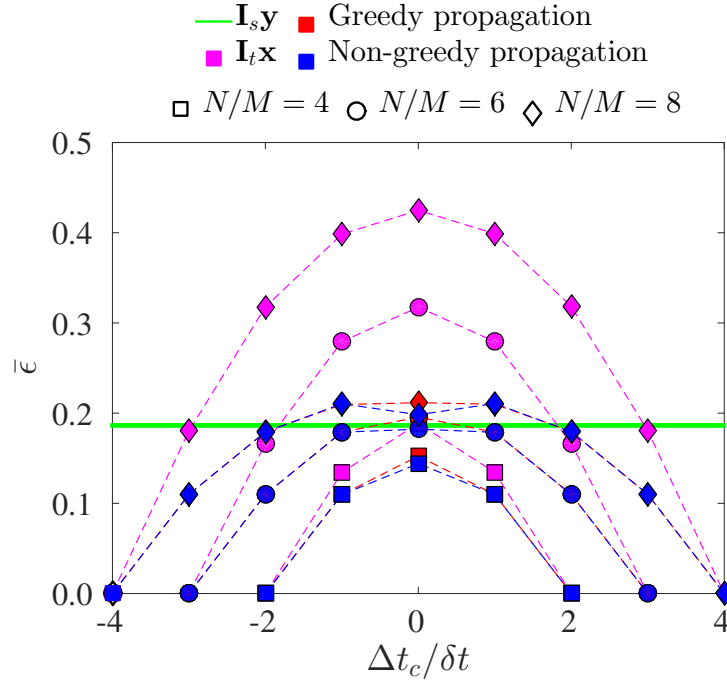


Figure 4.7 – Averaged NRMSEs of reconstruction by spatial/temporal interpolation, greedy and non-greedy propagation models. The errors are shown for three cases, with a fixed ratio in space ( $P/Q = 3 \times 3$ ) and varied ratios in time ( $N/M = 4, 6, 8$ ). The reference plane is chosen to be the mid-plane, and  $\Delta t_c / \delta t$  is the distance of each other plane from the reference one. Different colors refer to different methods, while the marker shapes vary for different  $N/M$ .

severe information losses after the subsampling in space, which cannot be completely recovered.

## 4.4 Concluding remarks

The similarity-based propagation model, also called NLM-based propagation model, is proposed to reconstruct small scales in a configuration where measurements of all scales are given at key frames (LTHS planes). Small scales from these planes are propagated toward other instants in time. The proposed model takes benefits from the local/nonlocal similarity among scales and among neighboring fields. Large-scale information is used as initial estimates and kept unchanged, while small scales are propagated as a weighted sum within a small neighborhood from one plane to another. The weights are estimated as the similarity of large scales in small patches. They are interpreted as the probabilities of an estimate to be its neighbors from nearby planes or key frames.

The model works very well in the situation where the subsampling ratios are not severe. In such case, it is able to bring small scales of the flow and add to initial

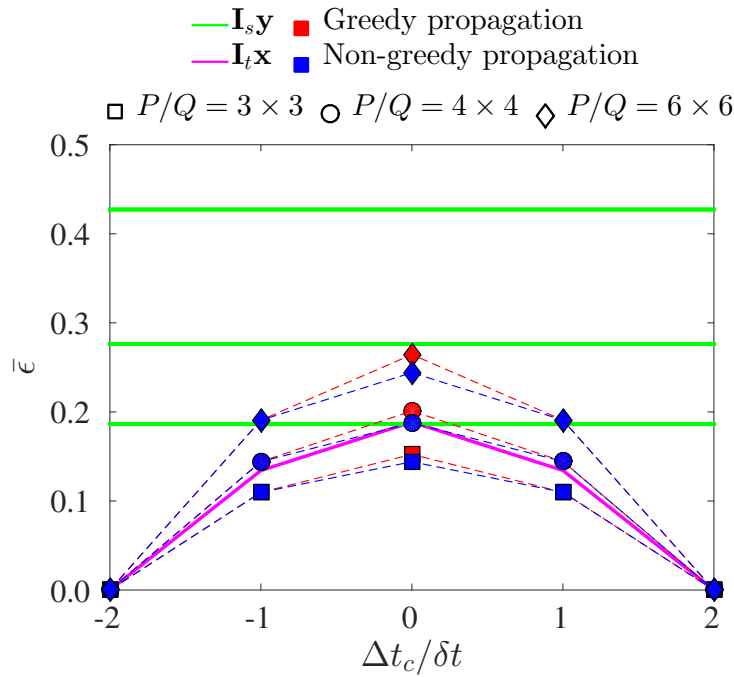


Figure 4.8 – Averaged NRMSEs of reconstruction by spatial/temporal interpolation, greedy and non-greedy propagation models. The errors are shown for different space spacing ( $P/Q = 3 \times 3, 4 \times 4, 6 \times 6$ ), while in time the ratio is fixed ( $N/M = 4$ ). The reference plane is chosen to be the mid-plane, and  $\Delta t_c / \delta t$  is the distance of each other plane from the reference one. Different colors refer to different methods, while the marker shapes vary for different  $P/Q$ .

large scales given by the coarse measurements. It also performs better than temporal interpolation from key frames, which is point-wise and disregards the spatial structures of the flow. Among the models, the non-greedy propagation works the most accurately by gradually bringing small-scale information from the bounded planes toward the center.

This model is not robust when the subsampling ratios remain moderate. When the time distance between LTHS planes increases, model performances are getting worse than spatial interpolation. This is because when the key frames are far from each other, similarities of large scales decay, making propagation models becomes inefficient. Reconstructed small-scale information even downgrades the initial large scales from spatial interpolation. The discrepancies are also found with large subsampling ratios in space. In such cases, the model suffers from severe information loss that can not be recovered completely.

More sophisticated models can be further investigated to improve the reconstruction. First, it's worth mentioning the idea of multi-scale propagation, which has been investigated during this work but gives no improvement. However, in the case where the range of scales is extremely wide, there might be potential gain from a non-greedy propagation in space as the one in time presented in this chapter. Second, since structures in turbu-

lence distort and rotate in a disorder manner, it maybe better to propose adapted methods that compute similarity of rotating patches. This idea has been started by Zimmer et al., (2008), but further investigation is required to adapt to our problem.

## Bayesian fusion model

This chapter proposes a Bayesian fusion model (Nguyen et al., 2015) to combine HTLS and LTHS measurements as described in section 1.3.2. The model takes benefit from both sources of information by searching for the most probable flow given the measurements thanks to a *maximum a posteriori* estimate. Better performances are expected since space and time correlations are equally important. The model also recovers flow details inaccessible from a single source of measurement via interpolations. By integrating directly the measurements, a compromise estimate is proposed such that detailed information of the flow close to the sensors is well preserved. This approach also overcomes the limitations of simple regression, which acts as a low pass filter due to the mean square error minimization. It potentially outperforms propagation models by using the space-time measurements in a balanced manner, permitting to take benefits equally from the two sources of information. In this chapter, DNS data of isotropic turbulence described in section 1.4.2 is used to investigate model performances. Further comparisons with other methods and demonstrations for DNS data of channel flow will be presented later in chapter 6.

### 5.1 A probabilistic model

Bayesian framework has been used widely from very early in communication problems (Sage and Melsa, 1971; Trees, 1967) and more recently in image processing, remote-sensing, data assimilation and data fusion ([wikle2007bayesian](#); Challa and Koks, 2004; Durrant-Whyte and Henderson, 2008; Joshi and Jalobeanu, 2010; Koks and Challa, 2003; Levitan and Herman, 1987; Ma et al., 2000; Zhang et al., 2009). The core of the framework is Bayes' rule, which describes how the probability of a system is altered by a new evidence or measurement. One of the main application of Bayes' rule is the

inference, with the aids of different Bayesian estimators.

### 5.1.1 Bayes' rule

Let  $\mathbf{d}$  denote a vector of observations/measurements/data, and  $\mathbf{h}$  denote a hypothesis. This hypothesis can be a parameter or an unknown variable to estimate, with a *prior distribution*  $p(\mathbf{h})$ . The probability of the observation vector, also called *evidence*, is  $p(\mathbf{d})$ . The joint probability  $p(\mathbf{d}, \mathbf{h})$  can be expressed as (Kendall et al., 1987):

$$\begin{aligned} p(\mathbf{h}, \mathbf{d}) &= p(\mathbf{h}|\mathbf{d})p(\mathbf{d}) \\ &= p(\mathbf{d}|\mathbf{h})p(\mathbf{h}) \end{aligned} \quad (5.1)$$

where  $p(.|.)$  is the conditional distribution of one variable given the other. Rearranging the above equation, the conditional distribution of the hypothesis knowing the data is:

$$p(\mathbf{h}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{h})p(\mathbf{h})}{p(\mathbf{d})} \quad (5.2)$$

where  $p(\mathbf{d})$  is a constant. This relation is referred as *Bayes's rule*, which can be rewritten as:

$$p(\mathbf{h}|\mathbf{d}) \propto p(\mathbf{d}|\mathbf{h})p(\mathbf{h}) \quad (5.3)$$

or in general:

$$[\text{posterior distribution}] \propto [\text{likelihood}] * [\text{prior distribution}] \quad (5.4)$$

The prior distribution  $p(\mathbf{h})$  describes our knowledge of  $\mathbf{h}$  without observing the data  $\mathbf{d}$ , while  $p(\mathbf{h}|\mathbf{d})$  is the corresponding knowledge after observing the data.  $p(\mathbf{d}|\mathbf{h})$  is *likelihood* function, which measures how far the hypothesis  $\mathbf{h}$  explain observations  $\mathbf{d}$ .

### 5.1.2 Bayesian estimates

Bayes's rule is very simple but very powerful. Inference is one of the most important applications. In most inference problem, either maximum likelihood estimate (MLE) or maximum a posteriori (MAP) estimate is used.

**Maximum likelihood estimate (MLE)**

The likelihood is the probability of the measurements  $\mathbf{d}$  under the hypothesis  $\mathbf{h}$ . For given measurements, MLE finds the hypothesis  $\mathbf{h}$  that maximizes the likelihood function:

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmax}} \{p(\mathbf{d}|\mathbf{h})\} \quad (5.5)$$

The resulting  $\hat{\mathbf{h}}$  can be considered as the one that makes the measurements the most “probable” or “likely”. Alternatively, the MLE is often obtained by maximizing the *log-likelihood*:

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmax}} \{\log(p(\mathbf{d}|\mathbf{h}))\} \quad (5.6)$$

Very often, this problem is ill-posed, while in some situations, an explicit expression of MLE can be obtained.

**Maximum a posteriori estimate (MAP)**

In many cases, some knowledge of undergoing phenomena is known *a priori*. Such knowledge can be from theory or empirical evidence, represented via the prior distribution  $p(\mathbf{h})$ . To find the most probable hypothesis  $\mathbf{h}$  given measurements  $\mathbf{d}$  and some prior knowledge of  $\mathbf{h}$ , the posterior probability  $p(\mathbf{h}|\mathbf{d})$  is maximized. Bayes’ rule tells us how to incorporate prior knowledge into the maximization problem as:

$$\begin{aligned} \hat{\mathbf{h}} &= \underset{\mathbf{h}}{\operatorname{argmax}} \{p(\mathbf{h}|\mathbf{d})\} \\ &= \underset{\mathbf{h}}{\operatorname{argmax}} \{p(\mathbf{d}|\mathbf{h})p(\mathbf{h})\} \end{aligned} \quad (5.7)$$

MAP is different from MLE due to the presence of the prior probability  $p(\mathbf{h})$ . This prior plays the role of a regularization when dealing with ill-posed inverse problems.

## 5.2 Bayesian fusion model

In a Bayesian approach, this section derives a fusion model to combine LTHS and HTLS measurements to reconstruct fully-resolved HTHS fields. This is done by considering all variables as random. The most probable HTLS is sought such that it maximizes a *posterior* probability given the two measurements via a MAP estimate.

### 5.2.1 The model

Let  $\mathbf{x}$  and  $\mathbf{y}$  denote LTHS and HTLS measurements, and  $\mathbf{z}$  denote HTHS data to reconstruct.  $\mathbf{x}$  and  $\mathbf{y}$  are directly subsampled from HTHS, i.e.  $\mathbf{x} = \mathbb{S}_t \mathbf{z}$  and  $\mathbf{y} = \mathbb{S}_s \mathbf{z}$ . The variables  $\mathbf{z}$ ,  $\mathbf{x}$  and  $\mathbf{y}$  are random, zero-mean of dimension  $PN$ ,  $PM$  and  $QN$  respectively.  $P$  and  $Q$  are numbers of spatial points in each snapshot, while  $N$  and  $M$  are numbers of snapshots. The present work is a challenging inverse problem since we consider  $Q \ll P$  and  $M \ll N$ . Let “s” denote the subscript of operators performing in space, and “t” denote the subscript of those in time.  $\mathbb{I}$  is an cubic spline interpolator,  $\mathbb{S}$  is for subsampling, and  $\mathbb{L}$  is a low-pass filter to separate large scales from small scales. In the case of isotropic turbulence,  $\mathbb{L}$  is the ideal Fourier filter thanks to the isotropic and periodic properties. In the case of channel flow,  $\mathbb{L}$  is a least-square spline filter for its sharp cutoff response.

Given LR measurements of either  $\mathbf{y}$  in space or  $\mathbf{x}$  in time, the fully resolved vector  $\mathbf{z}$  can be estimated by single interpolations. The 1D time interpolation goes from a  $PM$  to a  $PN$ - dimensional space, i.e.  $\mathbf{x} \mapsto \hat{\mathbf{z}} = \mathbb{I}_t \mathbf{x}$ , while the 2D space interpolation goes from a  $QN$  to a  $PN$ - dimensional space, i.e.  $\mathbf{y} \mapsto \hat{\mathbf{z}} = \mathbb{I}_s \mathbf{y}$ . Let  $PN$ - dimensional vectors  $\mathbf{h}_s$  and  $\mathbf{h}_t$  denote the information that cannot be recovered by simple interpolations  $\mathbb{I}_s$  or  $\mathbb{I}_t$ .  $\mathbf{z}$  can be modeled in two ways:

$$\mathbf{z} = \mathbb{I}_t \mathbf{x} + \mathbf{h}_t \quad (5.8)$$

$$\mathbf{z} = \mathbb{I}_s \mathbf{y} + \mathbf{h}_s \quad (5.9)$$

Missing information  $\mathbf{h}_t$  and  $\mathbf{h}_s$  essentially feature small scales. Using either  $\mathbf{x}$  or  $\mathbf{y}$ , it is not possible to estimate  $\mathbf{h}_t$  and  $\mathbf{h}_s$ . The idea of Bayesian fusion is to combine the two models by using  $\mathbb{I}_t \mathbf{x}$  in (5.8) to estimate the unknown  $\mathbf{h}_s$  in (5.9) and vice-versa.

Let  $\mathcal{N}(\mathbf{u} | \boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u)$  denote the multivariate Gaussian distribution of a  $PN$  dimensional random vector  $\mathbf{u}$  with mean value  $\boldsymbol{\mu}_u$  and covariance matrix  $\boldsymbol{\Sigma}_u$ . The  $PN \times PN$  matrix is the expectation of  $(\mathbf{u} - \boldsymbol{\mu}_u)(\mathbf{u} - \boldsymbol{\mu}_u)^\top$ . The probability density function (pdf) of  $\mathbf{u}$  with a multivariate Gaussian distribution  $\mathcal{N}(\mathbf{u} | \boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u)$  is:

$$p(\mathbf{u}) = \frac{1}{(2\pi)^{PN/2} |\boldsymbol{\Sigma}_u|^{1/2}} \exp\left(-\frac{1}{2} \|\mathbf{u} - \boldsymbol{\mu}_u\|_{\boldsymbol{\Sigma}_u}^2\right) \quad (5.10)$$

where  $|\cdot|$  denotes the matrix determinant, and  $\|\mathbf{u} - \boldsymbol{\mu}_u\|_{\boldsymbol{\Sigma}_u}^2$  is the Mahalanobis distance:

$$\|\mathbf{u} - \boldsymbol{\mu}_u\|_{\boldsymbol{\Sigma}_u}^2 = (\mathbf{u} - \boldsymbol{\mu}_u)^\top \boldsymbol{\Sigma}_u^{-1} (\mathbf{u} - \boldsymbol{\mu}_u) \quad (5.11)$$

Let assume that  $\mathbb{I}_t \mathbf{x}$  and  $\mathbf{h}_t$  are approximately independent;  $\mathbb{I}_t \mathbf{x}$  captures temporal large



scales of  $\mathbf{x}$ . Similarly,  $\mathbb{I}_s \mathbf{y}$  and  $\mathbf{h}_s$  are assumed to be approximately independent. This is not exact, since the subsamplings cause aliasing terms in each pairs of  $(\mathbb{I}_t \mathbf{x}, \mathbf{h}_t)$  and  $(\mathbb{I}_s \mathbf{y}, \mathbf{h}_s)$ . Assume also that  $\mathbf{h}_t$  and  $\mathbf{h}_s$  are zero mean Gaussian processes, i.e.  $\mathbf{h}_t \sim \mathcal{N}(0, \Sigma_{\mathbf{h}_t})$  and  $\mathbf{h}_s \sim \mathcal{N}(0, \Sigma_{\mathbf{h}_s})$ . Pdfs of these unknowns are modeled as:

$$p(\mathbf{h}_t) = \frac{1}{(2\pi)^{PN/2} |\Sigma_{\mathbf{h}_t}|^{1/2}} \exp\left(-\frac{1}{2} \|\mathbf{h}_t\|_{\Sigma_{\mathbf{h}_t}}^2\right) \quad (5.12)$$

and similarly for  $p(\mathbf{h}_s)$ . Posterior probabilities of  $\mathbf{z}$  knowing either  $\mathbf{x}$  or  $\mathbf{y}$  are then modeled as:

$$p(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mathbf{z}|\mathbb{I}_t \mathbf{x}, \Sigma_{\mathbf{h}_t}) \quad (5.13)$$

$$p(\mathbf{z}|\mathbf{y}) \sim \mathcal{N}(\mathbf{z}|\mathbb{I}_s \mathbf{y}, \Sigma_{\mathbf{h}_s}) \quad (5.14)$$

### 5.2.2 MAP estimation

The present Bayesian model aims to build an estimate of  $\mathbf{z}$  given  $\mathbf{x}$  and  $\mathbf{y}$  using the probability models (5.13) and (5.14). The model uses a MAP estimate to search the most probable  $\hat{\mathbf{z}}$  given  $\mathbf{x}$  and  $\mathbf{y}$  such that  $\hat{\mathbf{z}}$  maximizes the posterior pdf  $p(\mathbf{z}|\mathbf{x}, \mathbf{y})$ :

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\operatorname{argmax}} p(\mathbf{z}|\mathbf{x}, \mathbf{y}) \quad (5.15)$$

Using Bayesian rules, one has:

$$p(\mathbf{z}|\mathbf{x}, \mathbf{y}) \propto p(\mathbf{x}, \mathbf{y}|\mathbf{z}) p(\mathbf{z}) \quad (5.16)$$

Assuming that  $\mathbf{x}$  and  $\mathbf{y}$  are conditionally independent of  $\mathbf{z}$ , equation 5.16 becomes:

$$p(\mathbf{z}|\mathbf{x}, \mathbf{y}) \propto p(\mathbf{x}|\mathbf{z}) p(\mathbf{y}|\mathbf{z}) p(\mathbf{z}) \quad (5.17)$$

In equation 5.17, the likelihood functions  $p(\mathbf{x}|\mathbf{z})$ ,  $p(\mathbf{y}|\mathbf{z})$  and the prior pdf  $p(\mathbf{z})$  appear, while only the posterior probabilities  $p(\mathbf{z}|\mathbf{x})$  and  $p(\mathbf{z}|\mathbf{y})$  are available in (5.13) and (5.14).

To complete the model, the likelihood functions can be expressed in term of the posterior pdfs and prior of  $\mathbf{z}$  using Bayesian rules. Section 5.3.1 will introduce an alternative way to estimate these functions from posterior pdfs using a linear Gaussian model. Since there is no evidence of a proper prior for the fully resolved velocity field  $\mathbf{z}$ , it is natural to assume a “*noninformative*” prior (also referred as “*vague*” or “*flat*”). This prior assumes that all the values of  $\mathbf{z}$  are equally likely (Gelman et al., 2013). The

estimation of  $\hat{\mathbf{z}}$  is now solely based on the measurements and not influenced by external information. The prior distribution therefore has no influence on the posterior pdfs.

With the assumption of a noninformative prior,  $p(\mathbf{z})$  is constant. Using Bayes rules, the relation between the likelihood function and the posterior pdf is:

$$p(\mathbf{z}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \quad (5.18)$$

Since  $p(\mathbf{z})$  is replaced by a constant, one gets  $p(\mathbf{x}|\mathbf{z}) \propto p(\mathbf{z}|\mathbf{x})$  and  $p(\mathbf{y}|\mathbf{z}) \propto p(\mathbf{z}|\mathbf{y})$ . Equation 5.17 becomes:

$$p(\mathbf{z}|\mathbf{x}, \mathbf{y}) \propto p(\mathbf{z}|\mathbf{x})p(\mathbf{z}|\mathbf{y}) \quad (5.19)$$

The MAP estimate is therefore:

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\operatorname{argmax}} p(\mathbf{z}|\mathbf{x})p(\mathbf{z}|\mathbf{y}) \quad (5.20)$$

Logarithms of  $p(\mathbf{z}|\mathbf{x})$  and  $p(\mathbf{z}|\mathbf{y})$  are:

$$-\ln p(\mathbf{z}|\mathbf{x}) = \frac{1}{2} \|\mathbf{z} - \mathbb{I}_t \mathbf{x}\|_{\Sigma_{h_t}}^2 + C_1 \quad (5.21)$$

$$-\ln p(\mathbf{z}|\mathbf{y}) = \frac{1}{2} \|\mathbf{z} - \mathbb{I}_s \mathbf{y}\|_{\Sigma_{h_s}}^2 + C_2 \quad (5.22)$$

where  $C_1$  and  $C_2$  only depend on  $\Sigma_{h_t}$  and  $\Sigma_{h_s}$ , and are independent of  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{z}$ . Solving the optimization problem 5.20 is equivalent to minimize the cost function:

$$C(\mathbf{z}) = \frac{1}{2} \|\mathbf{z} - \mathbb{I}_t \mathbf{x}\|_{\Sigma_{h_t}}^2 + \frac{1}{2} \|\mathbf{z} - \mathbb{I}_s \mathbf{y}\|_{\Sigma_{h_s}}^2 \quad (5.23)$$

Setting the gradient of  $C(\mathbf{z})$  to zero:

$$\frac{\partial C(\mathbf{z})}{\partial \mathbf{z}} = \Sigma_{h_s}^{-1} (\mathbf{z} - \mathbb{I}_s \mathbf{y}) + \Sigma_{h_t}^{-1} (\mathbf{z} - \mathbb{I}_t \mathbf{x}) = 0 \quad (5.24)$$

yields:

$$\hat{\mathbf{z}} = \left( \Sigma_{h_t}^{-1} + \Sigma_{h_s}^{-1} \right)^{-1} \left( \Sigma_{h_s}^{-1} \mathbb{I}_s \mathbf{y} + \Sigma_{h_t}^{-1} \mathbb{I}_t \mathbf{x} \right) \quad (5.25)$$

Applying the matrix inversion lemma (Kay, 1993):

$$(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1} \quad (5.26)$$

Equation 5.25 can be rewritten as:

$$\hat{\mathbf{z}} = \Sigma_{h_t} \left( \Sigma_{h_t} + \Sigma_{h_s} \right)^{-1} \mathbb{I}_s \mathbf{y} + \Sigma_{h_s} \left( \Sigma_{h_t} + \Sigma_{h_s} \right)^{-1} \mathbb{I}_t \mathbf{x} \quad (5.27)$$

Equation 5.27 is the final form of the proposed Bayesian fusion model using a MAP estimate and assuming a noninformative prior of  $\mathbf{z}$ . Variance matrices  $\Sigma_{\mathbf{h}_t}$  and  $\Sigma_{\mathbf{h}_s}$  are parameters to be learned from the measurements.

## 5.3 Equivalent models

The above section presents a Bayesian fusion model to combine two sources of measurements. In the final formula, the prior distribution is omitted. The model is similar to some other models existing in the literature. The linear Gaussian model assumes that measurements follow Gaussian distributions to estimate the likelihood functions and prior probability. The final formula proposes slightly different weights compared to the Bayesian model assuming a non-informative prior. Another model is the generalized Millman formula (Shin et al., 2006), which is studied in the context of multi-sensor fusion. The model aims at combining information from various sensors. It becomes identical to the Bayesian fusion model when the cross-covariance between estimates are neglected. We briefly recall these models below.

### 5.3.1 Linear Gaussian model

Let consider equations 5.8 and 5.9 as the two linear models of  $\mathbf{z}$ . The posterior distributions  $\mathcal{N}(\mathbf{z}|\mathbb{I}_t \mathbf{x}, \Sigma_{\mathbf{h}_t})$  and  $\mathcal{N}(\mathbf{z}|\mathbb{I}_s \mathbf{y}, \Sigma_{\mathbf{h}_s})$  are given in equations 5.13 and 5.14. Distributions of the measurements  $p(\mathbb{I}_s \mathbf{y}) = \mathcal{N}(0, \Sigma_{\mathbb{I}_s \mathbf{y}})$  and  $p(\mathbb{I}_t \mathbf{x}) = \mathcal{N}(0, \Sigma_{\mathbb{I}_t \mathbf{x}})$  are also assumed Gaussian. Using Bayes' rule for these linear variables as described in Bishop, (2006), appendix A.1 derives the following formulas:

$$\begin{aligned} p(\mathbb{I}_t \mathbf{x} | \mathbf{z}) &= \mathcal{N}\left(\mathbb{I}_t \mathbf{x} | \Sigma_t \Sigma_{\mathbf{h}_t}^{-1} \mathbf{z}, \Sigma_t\right) \\ p(\mathbb{I}_s \mathbf{y} | \mathbf{z}) &= \mathcal{N}\left(\mathbb{I}_s \mathbf{y} | \Sigma_s \Sigma_{\mathbf{h}_s}^{-1} \mathbf{z}, \Sigma_s\right) \\ p(\mathbf{z}) &= \mathcal{N}(\mathbf{z} | 0, \Sigma_z) \end{aligned}$$

where:

$$\begin{aligned} \Sigma_t^{-1} &= \Sigma_{\mathbf{h}_t}^{-1} + \Sigma_{\mathbb{I}_t \mathbf{x}}^{-1} \\ \Sigma_s^{-1} &= \Sigma_{\mathbf{h}_s}^{-1} + \Sigma_{\mathbb{I}_s \mathbf{y}}^{-1} \\ \Sigma_z &= \Sigma_{\mathbf{h}_t} + \Sigma_{\mathbb{I}_t \mathbf{x}} = \Sigma_{\mathbf{h}_s} + \Sigma_{\mathbb{I}_s \mathbf{y}} \end{aligned}$$

Using the above Gaussian distributions, a MAP estimate of  $\mathbf{z}$  is:

$$\begin{aligned}\hat{\mathbf{z}} &= \underset{\mathbf{z}}{\operatorname{argmax}} \{p(\mathbb{I}_t \mathbf{x} | \mathbf{z})p(\mathbb{I}_s \mathbf{y} | \mathbf{z})p(\mathbf{z})\} \\ &= \underset{\mathbf{z}}{\operatorname{argmin}} \{-\ln p(\mathbb{I}_t \mathbf{x} | \mathbf{z}) - \ln p(\mathbb{I}_s \mathbf{y} | \mathbf{z}) - \ln p(\mathbf{z})\}\end{aligned}$$

Taking the logarithm of above probabilities yields:

$$\begin{aligned}-2 \ln p(\mathbb{I}_t \mathbf{x} | \mathbf{z}) &= (\mathbb{I}_t \mathbf{x} - \Sigma_t \Sigma_{\mathbf{h}_t}^{-1} \mathbf{z})^\top \Sigma_t^{-1} (\mathbb{I}_t \mathbf{x} - \Sigma_t \Sigma_{\mathbf{h}_t}^{-1} \mathbf{z}) \\ &= \|\Sigma_t \Sigma_{\mathbf{h}_t}^{-1} \mathbf{z} - \mathbb{I}_t \mathbf{x}\|_{\Sigma_t}^2 \\ -2 \ln p(\mathbb{I}_s \mathbf{y} | \mathbf{z}) &= (\mathbb{I}_s \mathbf{y} - \Sigma_s \Sigma_{\mathbf{h}_s}^{-1} \mathbf{z})^\top \Sigma_s^{-1} (\mathbb{I}_s \mathbf{y} - \Sigma_s \Sigma_{\mathbf{h}_s}^{-1} \mathbf{z}) \\ &= \|\Sigma_s \Sigma_{\mathbf{h}_s}^{-1} \mathbf{z} - \mathbb{I}_s \mathbf{y}\|_{\Sigma_s}^2 \\ -2 \ln p(\mathbf{z}) &= \mathbf{z}^\top \Sigma_z^{-1} \mathbf{z} = \|\mathbf{z}\|_{\Sigma_z}^2\end{aligned}\tag{5.28}$$

The cost function  $C(\mathbf{z})$  is defined as the sum of above terms:

$$C(\mathbf{z}) = \|\Sigma_t \Sigma_{\mathbf{h}_t}^{-1} \mathbf{z} - \mathbb{I}_t \mathbf{x}\|_{\Sigma_t}^2 + \|\Sigma_s \Sigma_{\mathbf{h}_s}^{-1} \mathbf{z} - \mathbb{I}_s \mathbf{y}\|_{\Sigma_s}^2 + \|\mathbf{z}\|_{\Sigma_z}^2\tag{5.29}$$

The gradient of  $C(\mathbf{z})$  is computed as:

$$\begin{aligned}\frac{\partial C(\mathbf{z})}{2 \partial \mathbf{z}} &= \Sigma_t^{-1} (\Sigma_t \Sigma_{\mathbf{h}_t}^{-1} \mathbf{z} - \mathbb{I}_t \mathbf{x}) + \Sigma_s^{-1} (\Sigma_s \Sigma_{\mathbf{h}_s}^{-1} \mathbf{z} - \mathbb{I}_s \mathbf{y}) + \Sigma_z^{-1} \mathbf{z} \\ &= (\Sigma_{\mathbf{h}_t}^{-1} + \Sigma_{\mathbf{h}_s}^{-1} + \Sigma_z^{-1}) \mathbf{z} - (\Sigma_t^{-1} \mathbb{I}_t \mathbf{x} + \Sigma_s^{-1} \mathbb{I}_s \mathbf{y})\end{aligned}$$

The estimate  $\hat{\mathbf{z}}$  is obtained by setting this gradient to zero so that:

$$\hat{\mathbf{z}} = (\Sigma_{\mathbf{h}_t}^{-1} + \Sigma_{\mathbf{h}_s}^{-1} + \Sigma_z^{-1})^{-1} (\Sigma_t^{-1} \mathbb{I}_t \mathbf{x} + \Sigma_s^{-1} \mathbb{I}_s \mathbf{y})\tag{5.30}$$

This fusion formula has the same form of a weighted sum, with the weights are slightly different from the formula 5.25.

### 5.3.2 Generalized Millman formula

In the context of multisensor fusion, the Millman and Bar-Shalom-Campo formula is very common to combine the two estimates with either correlated or uncorrelated errors. This formula is generalized in Shin et al., 2006, permitting to fuse multi-source measurements. This formula turns out to be similar in this context of Bayesian fusion when no prior knowledge is inserted into the model.

Let denote  $\{\hat{\mathbf{z}}_1, \dots, \hat{\mathbf{z}}_K\}$  the  $K$  estimates of an unknown random vector  $\mathbf{z} \in \mathbb{R}^n$ . The objective of the fusion model is to find the best linear estimate via a set of coefficient

matrices  $C_i$  ( $i=1,\dots,K$ ):

$$\hat{\mathbf{z}} = \sum_{i=1}^K C_i \hat{\mathbf{z}}_i \quad s.t. \quad \sum_{i=1}^K C_i = I_n \quad (5.31)$$

The final form of GMF, as derived in appendix A.2 is:

$$\hat{\mathbf{z}} = \sum_{i=1}^K C_i \hat{\mathbf{z}}_i \quad s.t. \quad \begin{cases} \sum_{i=1}^K C_i = I_n \\ \sum_{i,j=1}^{K-1} C_i (\Sigma_{ij} - \Sigma_{iK}) + C_K (\Sigma_{Kj} - \Sigma_{KK}) = 0, \quad j = 1, \dots, K-1 \end{cases} \quad (5.32)$$

In the case of fusing two estimates ( $K = 2$ ), this becomes the Bar-Shalom-Campo formula:

$$\hat{\mathbf{z}} = (\Sigma_{22} - \Sigma_{21})(\Sigma_{11} + \Sigma_{22} - \Sigma_{12} - \Sigma_{21})^{-1} \hat{\mathbf{z}}_1 + (\Sigma_{11} - \Sigma_{12})(\Sigma_{11} + \Sigma_{22} - \Sigma_{12} - \Sigma_{21})^{-1} \hat{\mathbf{z}}_2 \quad (5.33)$$

If  $\hat{\mathbf{z}}_1$  and  $\hat{\mathbf{z}}_2$  are uncorrelated, i.e.  $\Sigma_{12} = \Sigma_{21} = 0$ , the fusion formula becomes the Millman's formula

$$\hat{\mathbf{z}} = \Sigma_{22}(\Sigma_{11} + \Sigma_{22})^{-1} \hat{\mathbf{z}}_1 + \Sigma_{11}(\Sigma_{11} + \Sigma_{22})^{-1} \hat{\mathbf{z}}_2 \quad (5.34)$$

which is identical to the proposed Bayesian fusion model with a non-informative prior on  $\mathbf{z}$ .

## 5.4 Model simplifications

Though providing the full theoretical estimate of  $\hat{\mathbf{z}}$ , equation 5.27 is impractical to use as is for several reasons. The full covariance matrices  $\Sigma_{\mathbf{h}_t}$  and  $\Sigma_{\mathbf{h}_s}$ , representing all the sources of correlations in space and time, cannot be estimated from only the measurements  $\mathbf{x}$  and  $\mathbf{y}$ . This is because the unknown  $\mathbf{h}_t$  and  $\mathbf{h}_s$  are only accessible at the measured positions in space and time. Also, the covariance matrices of size  $PN \times PN$  are very large, making them very difficult to accurately estimate and to invert. Additional assumptions on the shape of  $\Sigma_{\mathbf{h}_t}$  and  $\Sigma_{\mathbf{h}_s}$  are necessary.

### 5.4.1 Simplified covariance matrices

A common approach to simplify the formula 5.27 is to assume  $\Sigma_{\mathbf{h}_s}$  and  $\Sigma_{\mathbf{h}_t}$  be diagonal. This simplification implies the independence between all elements of  $\mathbf{h}_t$  and  $\mathbf{h}_s$ . The

resulting simplified version of formula 5.27 is point-wise:

$$\hat{z}[i, t] = \frac{\sigma_{h_s}^2[i, t]}{\sigma_{h_s}^2[i, t] + \sigma_{h_t}^2[i, t]} \mathbb{I}_t \mathbf{x}[i, t] + \frac{\sigma_{h_t}^2[i, t]}{\sigma_{h_s}^2[i, t] + \sigma_{h_t}^2[i, t]} \mathbb{I}_s \mathbf{y}[i, t] \quad (5.35)$$

where  $[i, t]$  is the index of each point in space and time respectively. The variances  $\sigma_{h_s}^2$  and  $\sigma_{h_t}^2$  are functions of each position in space and time. Their estimation is detailed in the next section. Equation 5.35 will be used to reconstruct HTHS data. As a weighted average, it proposes a compromise estimate from the measurements. With a symmetrical form in space and time, the model uses information from both measurements to correct large scales reconstruction and recover partial information at smaller scales.

Assuming diagonal covariance matrices, the linear Gaussian model in equation 5.30 is simplified as:

$$\begin{aligned} \hat{z}[i, t] = & \frac{\sigma_{h_s}^2[i, t] + \frac{\sigma_{h_t}^2[i, t] \sigma_{h_s}^2[i, t]}{\sigma_{\mathbb{I}_t \mathbf{x}}^2[i, t]}}{\sigma_{h_t}^2[i, t] + \sigma_{h_s}^2[i, t] + \frac{\sigma_{h_t}^2[i, t] \sigma_{h_s}^2[i, t]}{\sigma_z^2[i, t]}} \mathbb{I}_t \mathbf{x}[i, t] \\ & + \frac{\sigma_{h_t}^2[i, t] + \frac{\sigma_{h_t}^2[i, t] \sigma_{h_s}^2[i, t]}{\sigma_{\mathbb{I}_s \mathbf{y}}^2[i, t]}}{\sigma_{h_t}^2[i, t] + \sigma_{h_s}^2[i, t] + \frac{\sigma_{h_t}^2[i, t] \sigma_{h_s}^2[i, t]}{\sigma_z^2[i, t]}} \mathbb{I}_s \mathbf{y}[i, t] \end{aligned} \quad (5.36)$$

The simplified linear Gaussian model slightly modifies formula 5.35 by introducing the three terms:  $\frac{\sigma_{h_t}^2[i, t] \sigma_{h_s}^2[i, t]}{\sigma_z^2[i, t]}$ ,  $\frac{\sigma_{h_t}^2[i, t] \sigma_{h_s}^2[i, t]}{\sigma_{\mathbb{I}_s \mathbf{y}}^2[i, t]}$  and  $\frac{\sigma_{h_t}^2[i, t] \sigma_{h_s}^2[i, t]}{\sigma_{\mathbb{I}_t \mathbf{x}}^2[i, t]}$ . These three terms are essentially small, since variances of  $\mathbf{h}_s$  and  $\mathbf{h}_t$  alone are relatively small compared to the variances of  $\mathbf{z}$ ,  $\mathbb{I}_s \mathbf{y}$  and  $\mathbb{I}_t \mathbf{x}$  due to the power law spectrum of turbulence.

## 5.4.2 Statistical parameters estimation

Let  $\mathbf{Z}$ ,  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{H}_t$ ,  $\mathbf{H}_s$ ,  $\mathbf{\Gamma}_{h_t}$  and  $\mathbf{\Gamma}_{h_s}$  be the matrix forms of  $\mathbf{z}$ ,  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{h}_t$ ,  $\mathbf{h}_s$ ,  $\sigma_{h_t}^2$  and  $\sigma_{h_s}^2$  respectively. Matrices  $\mathbf{Z}$ ,  $\mathbf{H}_t$ ,  $\mathbf{H}_s$ ,  $\mathbf{\Gamma}_{h_t}$  and  $\mathbf{\Gamma}_{h_s}$  are of size  $N \times P$ , while  $\mathbf{X}$  and  $\mathbf{Y}$  are of size  $M \times P$  and  $N \times Q$ . Matrices  $\mathbf{\Gamma}_{h_t}$  and  $\mathbf{\Gamma}_{h_s}$  correspond to empirical variances, which are functions of space and time  $(i, t)$ .

Variance matrices are estimated from  $\mathbf{H}_t$  and  $\mathbf{H}_s$ , which are available at the measure-

ments positions only. We use:

$$\mathbf{S}_t \mathbf{H}_s = \mathbf{X} - \mathbf{I}_s \mathbf{S}_s \mathbf{X} \quad (5.37)$$

$$\mathbf{S}_s \mathbf{H}_t = \mathbf{Y} - \mathbf{I}_t \mathbf{S}_t \mathbf{Y} \quad (5.38)$$

where  $\mathbf{S}_t$  subsamples in time from  $N$  to  $M$  time steps, and  $\mathbf{S}_s$  subsamples in space from  $P$  to  $Q$  points. These  $M$  instants and  $Q$  positions are the same as for LTHS and HTLS measurements. Since the flow is approximately stationary and spatial interpolation is independent of time,  $\mathbf{\Gamma}_{h_s}[i, t]$  becomes  $\mathbf{\Gamma}_{h_s}[i]$ , a function of spatial locations only. These variances are estimated by averaging over all time steps:

$$\mathbf{\Gamma}_{h_s}[i] = \frac{1}{M} \sum_{t=1}^M (\mathbf{X}[i, t] - \mathbf{I}_s \mathbf{S}_s \mathbf{X}[i, t])^2 \quad (5.39)$$

Variance in  $\mathbf{\Gamma}_{h_t}$  is a function of distances  $\tau$  to the previous LTHS time step only, where  $\tau/\delta t = 0, 1, 2, \dots, N/M$ , and  $\delta t$  is the time lag between two consecutive HTLS time steps.  $\mathbf{\Gamma}_{h_t}$  becomes a function of space and  $\tau$  only, i.e.  $\mathbf{\Gamma}_{h_t}[i, \tau]$ . It is estimated by averaging over  $M$  blocks (of  $N/M$  snapshots) bounded by two consecutive LTHS instants:

$$\mathbf{\Gamma}_{h_t}[i, \tau] = \frac{1}{M} \sum_{t_o} (\mathbf{Y}(i, t_o + \tau) - \mathbf{I}_t \mathbf{S}_t \mathbf{Y}(i, t_o + \tau))^2 \quad (5.40)$$

where  $t_o/\delta t = 1, 1+N/M, 1+2N/M, \dots, 1+(M-1)N/M$ . Along the homogeneous direction of the flow,  $\mathbf{\Gamma}_{h_s}[i]$  and  $\mathbf{\Gamma}_{h_t}[i, \tau]$  can be also averaged over all blocks defined by the four neighboring HTLS measurements (see figure 1.6). The variances are then a function of positions only in non-homogeneous direction and relative distances to the four closest HTLS sensors. These estimated variances are rearranged into a vector form  $\sigma_{h_t}^2[i, t]$  and  $\sigma_{h_s}^2[i, t]$  to complete the fusion model using the simplified formula in equation 5.35.

## 5.5 Applying the Bayesian fusion model to DNS data of isotropic turbulence

The Bayesian fusion model is applied to combine HTLS and LTHS measurements from simulated data of an isotropic turbulence as described in section 1.4.2. It is noticed that the streamwise direction is considered as time dimension, while the two spatial directions are spanwise and vertical. Both Bayesian fusion model (BF) with non-informative prior using formula 5.35 and the linear Gaussian model (LG) using formula 5.36 are tested.

### 5.5.1 Impact of subsampling ratios

A total of seven cases gathered in table 5.1 are investigated. These cases correspond to different subsampling ratios. In space, the ratios  $\sqrt{P/Q} = 3, 4$  and  $6$  are equally applied in spanwise and vertical direction, corresponding to resolutions of HTLS fields  $Q = 32 \times 32$ ,  $24 \times 24$  and  $16 \times 16$  respectively. Subsampling ratios  $N/M$  in time (streamwise) dimension are  $4$  ( $M = 37 \times 24$ ),  $6$  ( $M = 37 \times 16$ ) and  $8$  ( $M = 37 \times 12$ ). The energy losses due to these subsamplings are estimated as in section 1.4.2.

These seven cases can be categorized into three groups. Case 1 and case 2 are when the subsampling ratios in time are more critical than those in space, reversely in case 3 and case 4. In such cases, either HTLS or LTHS measurements are more informative than the other. The interpolation from the measurements with smaller ratio is more accurate. In the last three cases, the subsamplings in space and time are balanced as the energy losses are approximately equal. The fusion model in such cases is expected to out-perform both interpolations by combining useful information from both sources.

The fusion model uses equation 5.35 to reconstruct fully resolved velocities  $\hat{\mathbf{z}}$  from  $\mathbf{x}$  and  $\mathbf{y}$ . Reconstructed fields are compared with the original DNS data via the normalized root mean square error (NRMSE) (equation 2.28). The field  $z$  is more or less difficult to estimate depending on the considered instant and position with respect to available measurements. To qualify, two types of NRMSE, the mean NRMSE  $\bar{\epsilon}$  and the maximum NRMSE  $\epsilon_{max}$ , are estimated using equation 2.28.  $\bar{\epsilon}$  is estimated over the set of points  $\mathbf{J}$  including all space-time positions. It represents how far the reconstructed field departs from ground truth in order to evaluate reconstruction accuracy.  $\epsilon_{max}$  is estimated using all blocks (in time and in spanwise directions) between HTLS sensors (see figure 1.6). The set  $\mathbf{J}$  includes all points at the local coordinate  $(\Delta y/2, \Delta z/2, \Delta T/2)$ . Errors of single interpolations from either measurements of HTLS or LTHS are also estimated for comparisons.

Table 5.2 shows reconstruction errors in all cases. In cases 1 and 2, the energy losses due to subsampling in time are much higher than in space, and vice-versa in cases 3 and 4. In these cases, the model gives similar errors compared to the best interpolation, with smaller  $\bar{\epsilon}$  and comparable  $\epsilon_{max}$ . In cases 5 to 7, the losses are due to both the subsamplings in space and time in a balanced manner. The two proposed models give similar accuracy, with  $\bar{\epsilon}$  reduced by 15% to 25% and  $\epsilon_{max}$  reduced by 5% to 15% compared to the best single interpolation.

Improvements were expected from the weighted average in equation 5.35 since it uses variances  $\sigma_s^2[i, t]$  and  $\sigma_t^2[i, t]$  as parameters of the flow physics, and  $\mathbb{I}_t \mathbf{x}$  and  $\mathbb{I}_s \mathbf{y}$  as the specific flow information. The proposed model imposes the reconstruction to be consistent with measurements at nearby positions and proposes compromise estimates



Case	Subsampling ratios		Energy loss	
	$\sqrt{P/Q}$	$N/M$	$\Delta\kappa_s(\%)$	$\Delta\kappa_t(\%)$
1	3	8	1.03	6.53
2	4	8	2.63	6.53
3	6	4	7.29	1.23
4	6	6	7.29	1.23
5	3	4	1.03	1.23
6	4	6	2.63	3.56
7	6	8	7.29	6.53

Table 5.1 – Configuration parameters of seven testing cases. The subsampling ratios of HTLS measurements are  $\sqrt{P/Q}$  and equal in both spatial directions. The ratios of LTHS measurements in time are  $N/M$ . The normalized energy losses in space  $\Delta\kappa_s$  and in time  $\Delta\kappa_t$  are defined in equation 1.2.

Case	$\bar{\epsilon}$				$\epsilon_{max}$			
	$\mathbb{I}_s \mathbf{y}$	$\mathbb{I}_t \mathbf{x}$	LG	BF	$\mathbb{I}_s \mathbf{y}$	$\mathbb{I}_t \mathbf{x}$	LG	BF
1	0.19	0.31	<b>0.16</b>	<b>0.16</b>	0.22	0.42	0.23	<b>0.22</b>
2	0.28	0.31	<b>0.21</b>	<b>0.21</b>	0.35	0.42	0.33	<b>0.32</b>
3	0.43	0.13	<b>0.13</b>	<b>0.13</b>	0.53	<b>0.19</b>	0.20	<b>0.19</b>
4	0.43	0.23	0.22	<b>0.21</b>	0.53	0.32	0.33	<b>0.32</b>
5	0.19	0.13	<b>0.11</b>	<b>0.11</b>	0.22	0.19	<b>0.17</b>	<b>0.17</b>
6	0.28	0.23	<b>0.18</b>	<b>0.18</b>	0.35	0.28	<b>0.26</b>	<b>0.26</b>
7	0.43	0.31	0.27	<b>0.26</b>	0.52	0.42	0.41	<b>0.40</b>

Table 5.2 – NRMSEs of all scales reconstruction errors for the seven cases presented in table 5.1.  $\bar{\epsilon}$  and  $\epsilon_{max}$  are the average and maximum NRMSE defined by equation 2.28.  $\bar{\epsilon}$  is averaged over all fields, while  $\epsilon_{max}$  is computed for the most difficult positions in space and time (the most remote from all nearby measurements at local coordinates  $(\Delta y/2, \Delta z/2, \Delta T/2)$ , see figure 1.6). These errors are shown for reconstructed fields by spatial interpolation  $\mathbb{I}_s \mathbf{y}$ , time interpolation  $\mathbb{I}_t \mathbf{x}$ , linear Gaussian model (LG) using formula 5.36 and Bayesian fusion model (BF) using formula 5.35. The smallest errors in each cases are boldfaced.

Case	Large scales				Small scales			
	$\mathbb{I}_s \mathbf{y}$	$\mathbb{I}_t \mathbf{x}$	LG	BF	$\mathbb{I}_s \mathbf{y}$	$\mathbb{I}_t \mathbf{x}$	LG	BF
5	0.16	0.12	<b>0.09</b>	<b>0.09</b>	0.98	0.67	<b>0.64</b>	<b>0.64</b>
6	0.23	0.19	<b>0.14</b>	<b>0.14</b>	0.98	0.80	0.71	<b>0.70</b>
7	0.34	0.22	0.20	<b>0.19</b>	0.98	0.81	<b>0.72</b>	<b>0.72</b>

Table 5.3 – Average NRMSEs  $\bar{\epsilon}$  of large and small scales (same notations in table 5.2).

elsewhere. Simple interpolations use either HTLS or LTHS measurements only, losing information from the other sources.

### 5.5.2 Large and small scales reconstruction

In cases 1 to 4, both fusion models perform as the best interpolation with small improvements. This is expected since one measurement of HTLS or LTHS is much better resolved than the other. Cases 5 to 7 are the most interesting since energy losses due to subsampling in space and time are comparable. The model brings complementary information from both measurements and improves the reconstruction.

The reconstructions of large and small scales are investigated in details for these three cases. Spatial 2D ideal Fourier filters  $\mathbb{I}_s$  are used to separate large scales from small scales. These filters define different cutoff wave numbers based on the subsampling ratios. The reconstructed large scales by all methods are compared to the reference  $\mathbb{I}_s \mathbf{z}$ . Small scales are estimated using  $\mathbb{I} - \mathbb{I}_s$  where  $\mathbb{I}$  is the identity matrix. Table 5.3 shows NRMSEs estimated using equation 2.28 but normalized by the RMS of either  $\mathbb{I}_s \mathbf{z}$  or  $(\mathbb{I} - \mathbb{I}_s) \mathbf{z}$ .

The proposed fusion models recover part of small scales from complementary measurements. They correct also large-scale information compared to the spatial interpolation, since subsampling degrades those scales and introduces some aliasing effects. Results are shown in table 5.3 for the most favorable cases. Fusion models give smallest error in all cases and at both large and small scales. Errors of large scale reconstructions are reduced by 20 to 50 % compared to the best single interpolation, while these improvements are smaller for small scales reconstruction (5 to 25 %). Note that the reconstruction error remains very high for small scales. Also, a good reconstruction of large scales is surprisingly challenging despite the theoretical claims that large scales should be accurately reconstructed by interpolations.

## 5.6 Concluding remarks

This chapter proposes a Bayesian fusion model using a MAP estimate to reconstruct high resolution velocities of a turbulent channel flow from low resolution measurements in space and time. It searches for the most probable field given available measurements. This approach yields a simple but efficient weighted average formula in equation (5.35). Weighting coefficients are learned from measurements and encode the physics of the flow. The fusion of available measurements improves the large scale reconstruction and recovers some details at small-scale.

Numerical experiments using a DNS database of isotropic turbulence at a moderate Reynolds number illustrate the efficiency and robustness of the proposed method. Low resolution measurements are extracted to learn model parameters, while original data are used as the ground truth to estimate reconstruction errors. The model is tested in various cases with different subsampling ratios. Results are compared against cubic spline interpolations. Bayesian fusion always produces the most accurate reconstruction. The best results are obtained when missing spatial and temporal information are of the same order of magnitude. The fusion model provides a better large scale reconstruction compared to the interpolation. Small scales below the cutoff frequency are more difficult to recover. The error remains high, since only a certain amount of small scale details are reconstructed. This is one more reason in favor of using either some relevant physical model or some sophisticated learning procedure to "guess" small scales.

Next chapter will compare the performances of various methods we have presented. The analyses are on both the two datasets of isotropic turbulence and channel flow.



## Comparisons of model performances on various datasets

This chapter synthesizes results of reconstructions by the proposed models to compare their performances. Comparisons are shown for streamwise velocities from both DNS datasets of isotropic turbulence (section 1.4.2) and channel flow (section 1.4.1). Isotropic turbulence is rather ideal yet retains the fundamental characteristics of real turbulent flows. It facilitates all computations of proposed methods for its homogeneity and periodicity. The data of turbulent channel flow mimics better real flows. Results on these data can be generalized for other velocity components and for experimental data.

We study the setup where multi-source measurements are given, the HTLS and LTHS, as described in section 1.3.2. For each data, three favorable configurations of balanced energy losses in space and time are used to perform detailed analyses. These configurations permit to apply both families of approaches, the empirical models to learn the relation between large and small scales, and the fusion models to combine the two sources. With empirical approaches such as regressions (chapter 2) or dictionary learning (chapter 3), LTHS planes are considered as training samples, while HTLS measurements are used for reconstruction. Fusion models instead simultaneously combine both sources of measurements by either exploiting different hypotheses (chapter 4 and 5).

Comparisons between two families of methods are valid but bias. Empirical mapping models are not designed to take benefits of information in time, and reconstruction errors purely depend on the subsampling ratio in space. Performances of fusion models will depend on both ratios in space and time. Both results from the two proposed models suggest that more gains are expected when the subsamplings are in a balanced manner. In such cases, the models can get benefits from both sources of information. Otherwise, if the loss caused by one subsampling is much more severe than by another, the models tend to perform at best as accurate as the single interpolation from measurements with

the lower energy loss. Comparisons with regression models and dictionary learning also demonstrate how information from the measurements in time are beneficial in improving reconstruction accuracies.

Dictionary learning is omitted in all the analyses, since it works only when downsampled fields are considered as training low resolution planes. This configuration is different from all other chapters and the inspired experiment at LML. For the direct subsampling cases, it shows insignificant improvement compared to spatial interpolations.

## 6.1 Reconstruction results on DNS data of isotropic turbulence

Detailed analyses of proposed models are investigated on the DNS data of isotropic turbulence as described in section 1.4.2. The data facilitates numerical experiments of all proposed methods for its isotropy, periodicity and homogeneity. The computation complexity is first discussed. Then, detailed analyses of model performances are shown for the three cases of balanced energy losses.

### Computation time

All numerical experiments of the proposed methods are performed on the same personal computer using either Matlab (version 2014b) or python (version 2.7). Non-local means propagation models are programmed in C using *openMP* parallelization. Dictionary learning codes are based on *SParse Modeling Software* (SPAMS) package (<http://spams-devel.gforge.inria.fr/>). This package is written in C with Matlab and python interfaces. Regression codes are based on *scikit-learn* package (Pedregosa et al., 2011). The data contains  $37 \times 96$  HR planes at the resolution  $96 \times 96$  (about 400 Mb, extracted from 25 Gb of original simulation data).

Regression and Bayesian fusion models are the simplest and fastest model to test. To reconstruct all  $37 \times 96$  fields, Bayesian fusion model takes only a few minutes to interpolate, learn the covariance matrices and perform the weighted sum, thanks to the matrix vectorization. Regression models, both RR and KRR, take up to one minute to train the models. The time to reconstruct the whole  $37 \times 96$  fields is about 1 ~ 2 minutes. The most time consuming step is the cross-validation, which takes about 10 ~ 30 minutes. Coupled dictionary learning takes up to about 10 seconds to reconstruct one field in the case of *SR3* (see table 3.1) and even faster with *SR1* and *SR2*. The learning step, which is more expensive, can be done *offline*. Learning a dictionary of size  $K = 2 \times 16^2$  from  $m = 10^5$  patches of the dimension  $p = 16^2$  takes about one hour to converge. With

Case	Subsampling ratios		Energy losses	
	$\sqrt{P/Q}$	$N/M$	$\Delta\kappa_s(\%)$	$\Delta\kappa_t(\%)$
5	3	4	1.03	1.23
6	4	6	2.63	3.56
7	6	8	7.29	6.53

Table 6.1 – Isotropic turbulence: configuration parameters of three testing cases with balanced energy losses. The subsampling ratios of HTLS measurements are  $\sqrt{P/Q}$  and equal in both spatial directions. The ratios of LTHS measurements in time (streamwise in this case) are  $N/M$ . The normalized energy losses in space  $\Delta\kappa_s$  and in time  $\Delta\kappa_t$  are defined in Eq. (1.2).

NLM-based propagation models, the estimation of one field takes about 1 ~ 2 seconds without parallelization. All methods acquire a tractable computational time and therefore could be generalized to larger experiments. To recall, all codes and data are available at <https://github.com/linhvannnguyen/PhDworks/>.

## Model performance analyses

We study three cases of balanced energy losses in space (spanwise and vertical direction) and time, with of subsampling ratios and energy losses are summarized in table 6.1. Due to the lack of time-resolved fields, the streamwise direction is also considered as the virtual time direction. The first family of methods (regression and dictionary learning) does not depend on this condition. Bayesian fusion model also depends on the losses of energy but not really the properties of the flow in time. NLM model might be more sensitive, since the rapid distortion assumption is stronger with convective flows. It is expected to work better in the case where time-resolved sequence of snapshots are studied.

Table 6.2 gathers NRMSEs between reconstructed and reference fields of three cases in table 6.1. Both average and maximum errors are shown for all methods. Interpolations are 2D in space as  $\mathbb{I}_s \mathbf{y}$  and 1D in time as  $\mathbb{I}_t \mathbf{x}$ . Regression models are either linear (RR) or non-linear (KRR) using RBF kernel. NLM-based propagations of both greedy and non-greedy schemes are shown. As fusion models, linear Gaussian or Bayesian fusion of non-informative prior are studied. The “best scores”, i.e. lowest NRMSEs, are shown in bold at the bottom row. Fusion models by combining all sources of measurements give lowest errors. The accuracy improvements compared to single interpolations are significant.  $\bar{\epsilon}$  is reduced by 15% ~ 25%, and  $\epsilon_{max}$  is reduced 5% ~ 10% compared to the best single interpolation.

Method	$\bar{\epsilon}$			$\epsilon_{max}$		
	Case 5	Case 6	Case 7	Case 5	Case 6	Case 7
$\mathbb{I}_s \mathbf{y}$	0.19	0.28	0.43	0.22	0.35	0.52
$\mathbb{I}_t \mathbf{x}$	0.13	0.23	0.31	0.19	0.28	0.42
RR	0.14	0.23	0.35	0.20	0.34	0.50
KRR	0.13	0.23	0.34	0.20	0.34	0.49
Greedy propagation	0.11	0.20	0.32	0.18	0.33	0.51
Non-greedy propagation	0.11	0.19	0.30	0.17	0.31	0.46
Fusion (LG)	0.11	0.18	0.27	0.17	0.26	0.41
Fusion (BF)	0.11	0.18	0.26	0.17	0.26	0.40
Best scores	<b>0.11</b>	<b>0.18</b>	<b>0.26</b>	<b>0.17</b>	<b>0.26</b>	<b>0.40</b>

Table 6.2 – NRMSEs of all scales reconstruction errors for three cases in table 6.1.  $\bar{\epsilon}$  and  $\epsilon_{max}$  are the mean and max NRMSE defined in Eq. (2.28).  $\bar{\epsilon}$  is averaged over all fields, while  $\epsilon_{max}$  is computed for the most difficult positions in space and time (the most remote from all nearby measurements). The smallest errors in each cases are boldfaced.

Performances of fusion and NLM-based propagation models are similar at the low subsampling ratio (case 5). Fusion models are superior in cases 6 and 7 of higher ratios. They take into account all sources of measurements and find the best compromise estimates via weighted sums. NLM models propagate small-scale information from LTHS planes and put on top of initial spatial interpolations. The quality of propagations strongly depends on flow dynamics. When the distance between the LTHS planes is far, the assumption that small scales are advected by large ones can be rather crude.

Regression models give worse results than almost all methods except spatial interpolations. As discussed, their errors depends solely on the subsampling ratio in space. All regression models are not designed in such way to take benefits from information in time. LTHS planes are used only to train the models. The models use averaged coefficients to reconstruct all fields regardless the temporal dynamics of the flows. The performances should be compared only to the spatial interpolation  $\mathbb{I}_s \mathbf{y}$ . KRR gives slightly better reconstructions to RR. The model reduces  $\bar{\epsilon}$  by 18% ~ 32%, and  $\epsilon_{max}$  by 5% ~ 10% compared to interpolations.

Cases 5 and 7 are used to further analyze model performances. Case 5 corresponds to small energy losses due to subsamplings in both time (streamwise) and space (about 1%), while these losses of case 7 are more severe (about 7 %). The configuration with small losses is favorable for all models, and should be of interest when designing experiments such that the reconstructed information is reliable. Case 7 is more critical to highlight different behaviors and also limitations of present approaches.



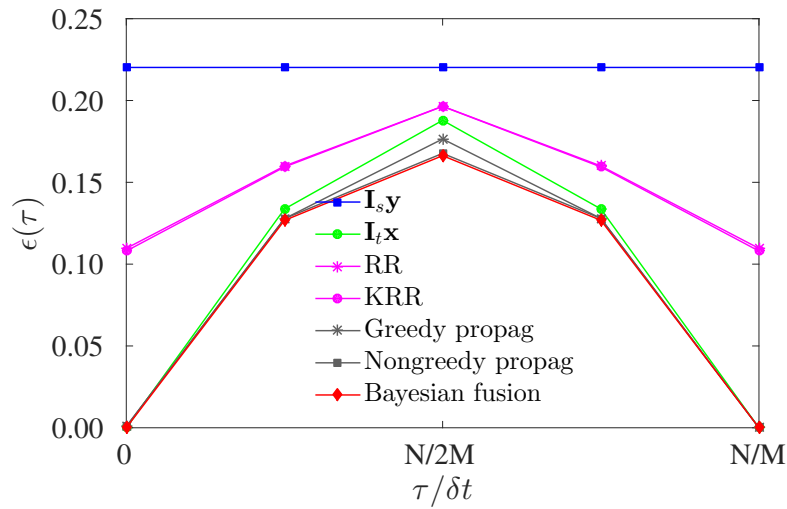


Figure 6.1 – Isotropic turbulence: NRMSEs are functions of time distances from the previous LTHS instant at the most difficult spatial location, i.e. at  $(\Delta y/2, \Delta z/2, \tau)$ . These errors are estimated between reference and reconstructed streamwise velocities by spatial and temporal interpolations, RR, KRR, non-greedy propagation and Bayesian fusion models. Results are shown for case 5 in table 6.1, with the subsampling ratios  $N/M = 4$  in time and  $P/Q = 3 \times 3$  in space.

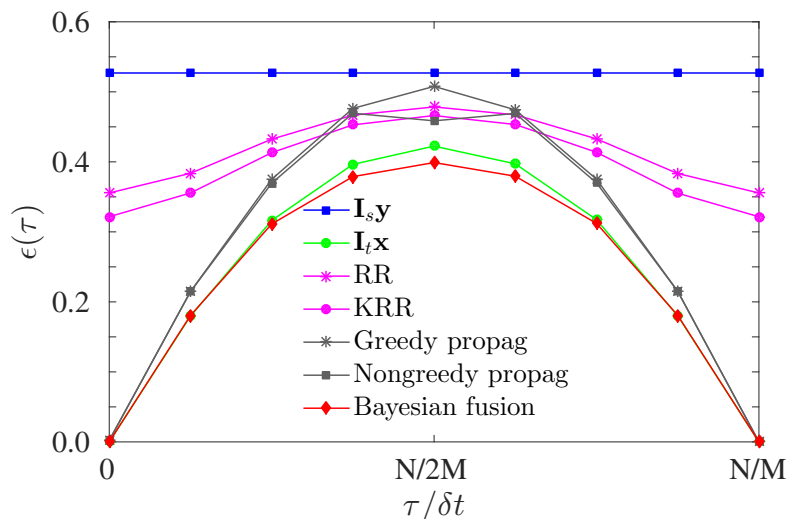


Figure 6.2 – Isotropic turbulence: similar plot as figure 6.1 for case 7 in table 6.1, with the subsampling ratios  $N/M = 8$  in time and  $P/Q = 6 \times 6$  in space.

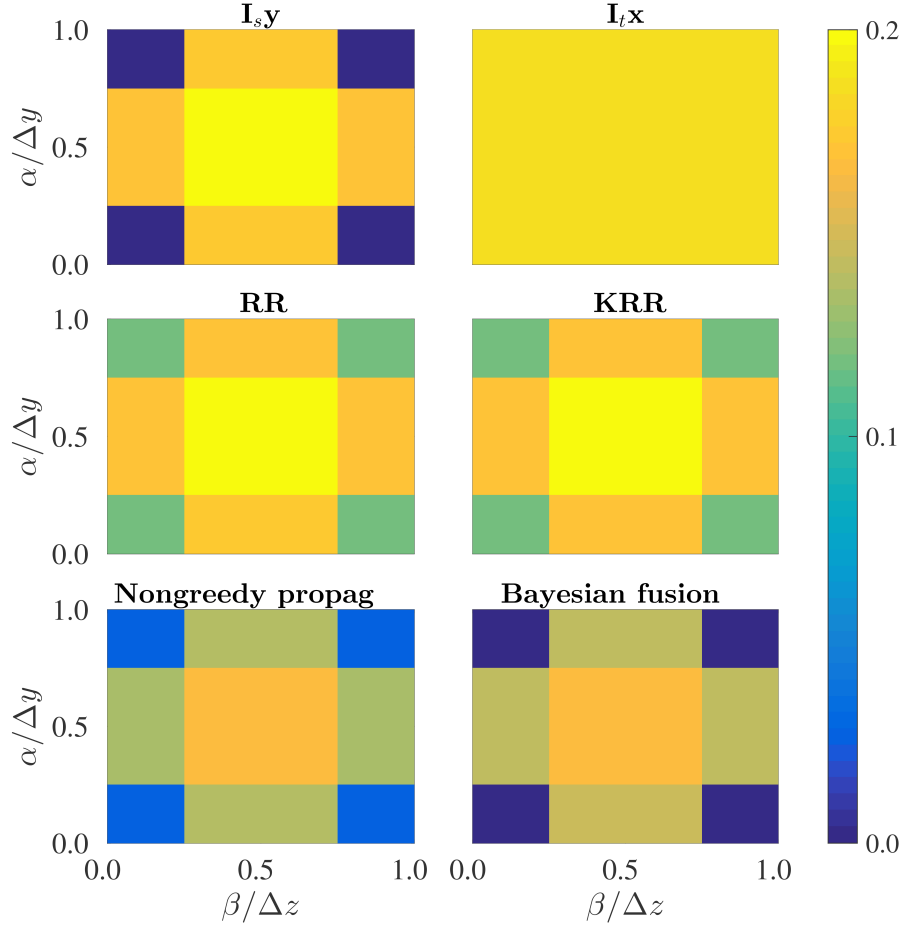


Figure 6.3 – Isotropic turbulence: NRMSEs are functions of spatial coordinates in an element block at the most difficult instant, i.e. at  $(\alpha, \beta, N\delta t/2M)$ . These errors are estimated between reference and reconstructed streamwise velocities by spatial and temporal interpolations, RR, KRR, non-greedy propagation and Bayesian fusion models. Results are shown for case 5 in table 6.1, with subsampling ratios of  $N/M = 4$  in time and  $P/Q = 3 \times 3$  in space.

To analyze the reconstructions in time, figure 6.1 shows NRMSEs as a function of distances  $\tau$  from the previous LTHS time step in cases 5. For each  $\tau$ , NRMSE is estimated using the set of point  $\mathbb{J}$  including points at local coordinates  $(\Delta y/2, \Delta z/2, \tau)$  of all blocks used to estimate  $\epsilon_{max}$ . NRMSEs are small close to the LTHS measurements ( $\tau/\delta t = 0$  and  $\tau/\delta t = N/M$ ) and increase when moving toward the middle ( $\tau/\delta t = N/2M$ ). Spatial interpolation behaves differently since the NRMSE is independent of time. Regression models illustrate the advantages of learning a model by significantly reducing the errors (about 15% for mid-planes at  $N/M$ ) compared to spatial interpolation. Time interpolation gives smaller error than RR and KRR, since those errors of regression models depend only on the subsampling ratio in space. These temporal information from LTHS planes are taken into account to further improve the reconstruction using fusion models. Both the

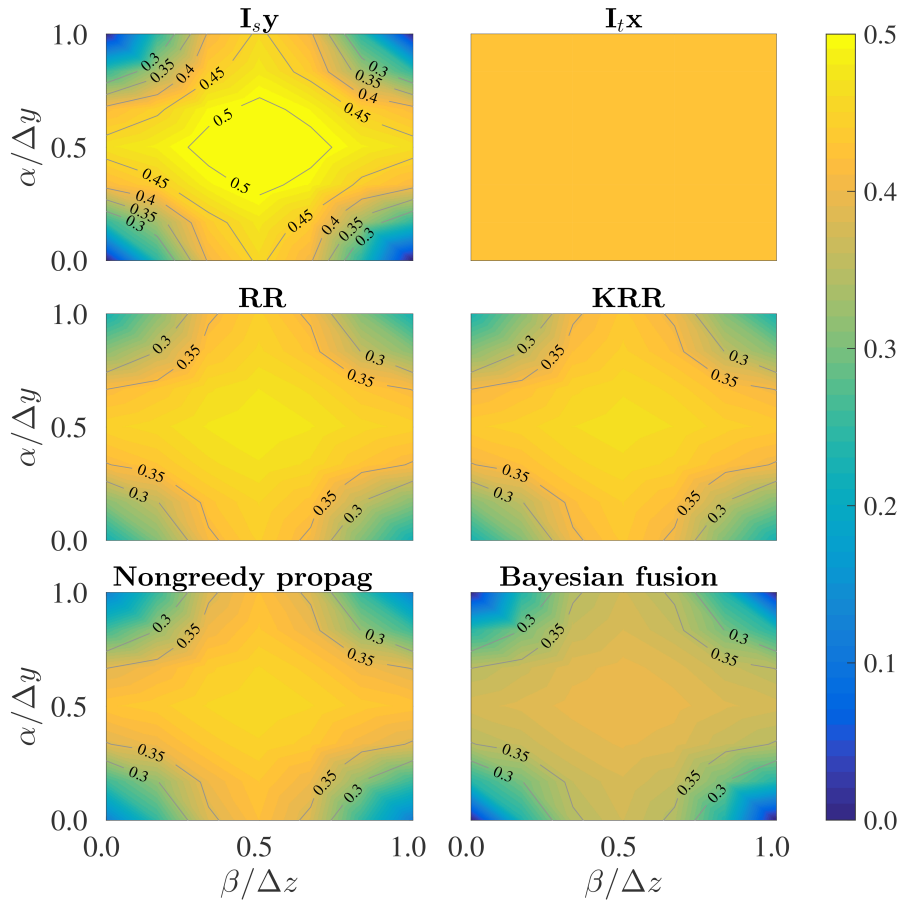


Figure 6.4 – Isotropic turbulence: NRMSEs are functions of spatial coordinates as in figure 6.3. Results are shown for case 7 in table 6.1, with subsampling ratios of  $N/M = 8$  in time and  $P/Q = 6 \times 6$  in space.

propagation models and fusion model give better reconstructions than all other models. Bayesian fusion and non-greedy model are slightly better than the greedy one.

Figure 6.2 shows similar results as in figure 6.1 for case 7 with very severe subsampling ratios in both space and time. Model performances are consistent with case 5, except that propagation models now give higher NRMSEs than time interpolation. This is because the propagation models start from a critical loss of information and could not recover completely. Also, this approach does not take advantages of temporal information from LTHS snapshots (except their small scales to propagate). Bayesian fusion model, which proposes a compromise estimate between the two interpolations. The model yields the minimum errors at all time steps. Even in the middle of two LTHS instants, the maximum fusion error remains lower than the one obtained with both interpolations.

To analyze the reconstructions in space, figures 6.3 and 6.4 show spatial NRMSE maps by all the methods as a function of local coordinates  $(\alpha, \beta)$  as described in figure 1.6, section 1.3.2. For each  $(\alpha, \beta)$ , NRMSE is estimated using equation (2.28), where  $\mathbf{J}$

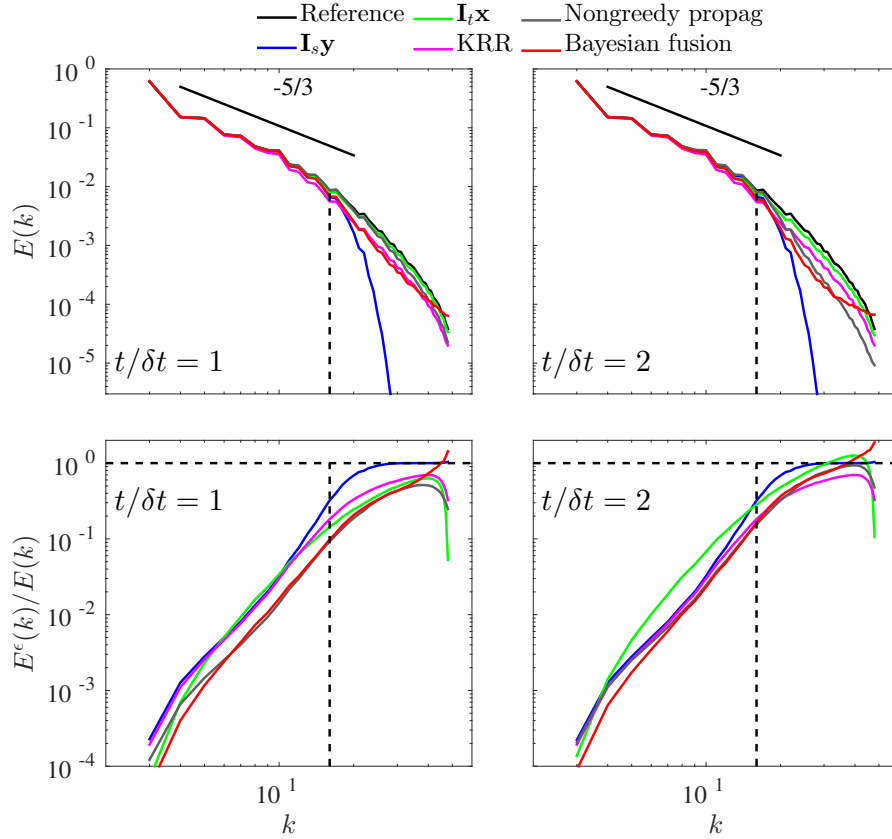


Figure 6.5 – Isotropic turbulence: comparison of energy spectra and error spectra for reconstructions by interpolations (in space and time), KRR, non-greed propagation and Bayesian fusion model in case 5 of subsampling ratios  $P/Q = 3 \times 3$  in space and  $N/M = 4$  in time (see table 6.1). The spectra are averaged over all planes of the same distance to the closest LTHS plane, i.e.  $t/\delta t = 1$  (left) or  $t/\delta t = 2$  (right).

includes points at  $(\alpha, \beta, N\delta t/2M)$  of all blocks used to estimate  $\epsilon_{max}$ . All methods give small NRMSEs close to the four HTLS positions in the corners. The errors increase when approaching the center. Time interpolation behaves differently since the errors are independent of spatial coordinates. Compared to interpolation in space, regression models do not give exact estimates at the HTLS positions. However, errors far from these four points are significantly reduced. The fusion models yields the best reconstruction results, where the Bayesian fusion model yields the smallest errors at all positions. At the most difficult positions (near the center of the block), it improves the reconstruction significantly compared to other methods.

Figure 6.5 shows the spectral analysis of the reconstruction by different models. To make the plot readable, RR, greedy propagation and linear Gaussian models are omitted. Energy spectra and error spectra are shown for the two positions, either close or far from LTHS planes. The first two spectra illustrate that spatial interpolation captures only

Case	Subsampling ratios		Spacings		Energy loss	
	$\sqrt{P/Q}$	$N/M$	$\Delta z/H$	$\Delta t$	$\Delta\kappa_s(\%)$	$\Delta\kappa_t(\%)$
5	5	4	0.05	0.10	1.23	1.88
6	10	10	0.11	0.25	7.08	9.02
7	20	20	0.22	0.50	20.83	24.31

Table 6.3 – Channel flow: configuration parameters of seven testing cases. The subsampling ratios of HTLS measurements are  $\sqrt{N/M}$  and equal in both spatial directions. The ratios of LTHS measurements in time are  $P/Q$ . The spacing in spanwise direction is normalized by half channel height as  $\Delta z/H$  and the spacing in time is  $\Delta t$ , normalized by  $H/U_{max}$  ( $U_{max}$  is the central velocity of the flow). The normalized energy losses in space  $\Delta\kappa_s$  and in time  $\Delta\kappa_t$  are defined in Eq. (1.2).

large-scale information, while time interpolation gives an estimate with a good energy spectrum. All models are subject to a certain energy loss at small scales. The error spectra shows that time interpolation gives good energy spectra with poorly reconstructed small scales. KRR also shows significant improvement compared to spatial interpolation, both in energy spectra and error spectra. Compared to those that have been shown in figure 4.6, the Bayesian fusion model shows similar results compared to the non-greedy propagation. Both models yield compromise estimates, where certain amount of small scales are captured. Error spectra illustrate improvements at all frequencies compared to both interpolations.

## 6.2 Reconstruction results on DNS data of channel flow

This section further analyses and compares performances of proposed methods on the DNS data of a turbulent channel flow with a moderate Reynolds number as described in section 1.4.1. This data mimics the experiments from WALLTURB project (Coudert et al., 2011), and simulates better practical flows compared to isotropic turbulence. Its characteristics are strongly non-homogeneous in vertical direction. The real “time” dimension is considered instead of a “virtual” one as in the previous dataset.

Based on the analyses on isotropic turbulence, and for simplicity, only RR and Bayesian fusion model are selected to represent the two families of approaches. Three cases of balanced subsamplings in space and time are considered. The configurations with subsampling ratios, spacings and amounts of energy loss are presented in table 6.3. The losses in space (2D) and time are similar in each case, from very small energy losses (less than 1 ~ 2% in case 5) to very severe ones (more than 20% in case 7). The average

Method	$\bar{\epsilon}$			$\epsilon_{max}$		
	Case 5	Case 6	Case 7	Case 5	Case 6	Case 7
$\mathbb{I}_y \mathbf{y}$	0.14	0.36	0.68	0.16	0.47	0.85
$\mathbb{I}_x \mathbf{x}$	0.11	0.32	0.54	0.18	0.55	0.85
RR	0.12	0.34	0.64	0.15	0.49	0.78
Fusion (BF)	0.08	0.25	0.46	0.13	0.43	0.73
Best scores	<b>0.08</b>	<b>0.25</b>	<b>0.46</b>	<b>0.13</b>	<b>0.43</b>	<b>0.73</b>

Table 6.4 – Channel flow: NRMSEs of all scales reconstruction errors for three cases in Table. 6.3.  $\bar{\epsilon}$  and  $\epsilon_{max}$  are the mean and max NRMSE defined in Eq. (2.28).  $\bar{\epsilon}$  is averaged over all space-time positions in the outer region  $y/H \in [0.25, 1.75]$ , while  $\epsilon_{max}$  is computed for one of the most difficult position in space and time (the most remote from all nearby measurements). The smallest errors in each cases are boldfaced.

and maximum NRMSEs of reconstructions by all methods for the three cases are shown in table 6.4. The best errors obtained by the fusion model are boldfaced and shown in the last row.

Case 6 of moderate energy loss (about 8%) is used for further visualization of the analysis. The energy losses are not too severe but are critical to highlight interests of the present approach.  $\bar{\epsilon}$  and  $\epsilon_{max}$  are reduced by 25 % and 35 % respectively for all scales reconstruction. Model performances in time and in space are shown in figures 6.6 and 6.7 respectively. They are estimated and plotted similarly as in figures 6.4 and 6.1. The fusion model clearly demonstrates the benefits of combining information in space and time by proposing a compromise estimate from the two single interpolations.

Figure 6.8 shows a time evolution of the point at  $y/H = 1$  and  $z/H = 0$  ( $\alpha = \Delta y/2$  and  $\beta = \Delta z/2$  in local coordinates, see figure 1.6), the most remote from neighboring HTLS sensors. A good agreement between fused and reference velocity is still obtained. A zoom-in period is shown for detailed comparisons with other methods. While time interpolation captures only low frequencies, spatial interpolation generates high frequencies but weakly correlated with the truth. The fusion model proposes a good compromise to improve both large and small scales reconstruction. It also captures detailed peaks much better than RR, since RR smooths these small scales out by minimizing the mean square errors.

Temporal spectra, estimated from the same points as those shown in figure 6.8, are compared in figure 6.9. Time interpolation fails to estimate the signal at higher frequencies than a certain cutoff. RR reconstructs both large and small scales, but the loss of large scale energy is critical. This loss is highlighted in the zoom-in picture of low frequencies. The present model improves the estimation at both low and high

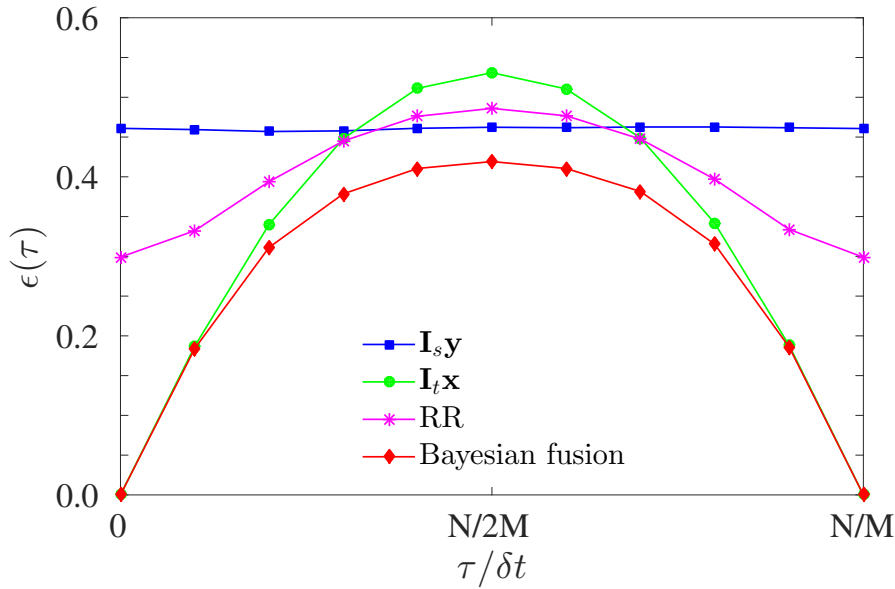


Figure 6.6 – Channel flow: NRMSEs between reference and reconstructed streamwise velocities by all methods as: functions of time distances from the previous LTHS instant at the most difficult spatial location, i.e. at  $(\Delta y/2, \Delta z/2, \tau)$ . Results are shown for case 5 in table 6.3, with the subsampling ratios  $N/M = 10$  in time and  $P/Q = 10 \times 10$  in space.

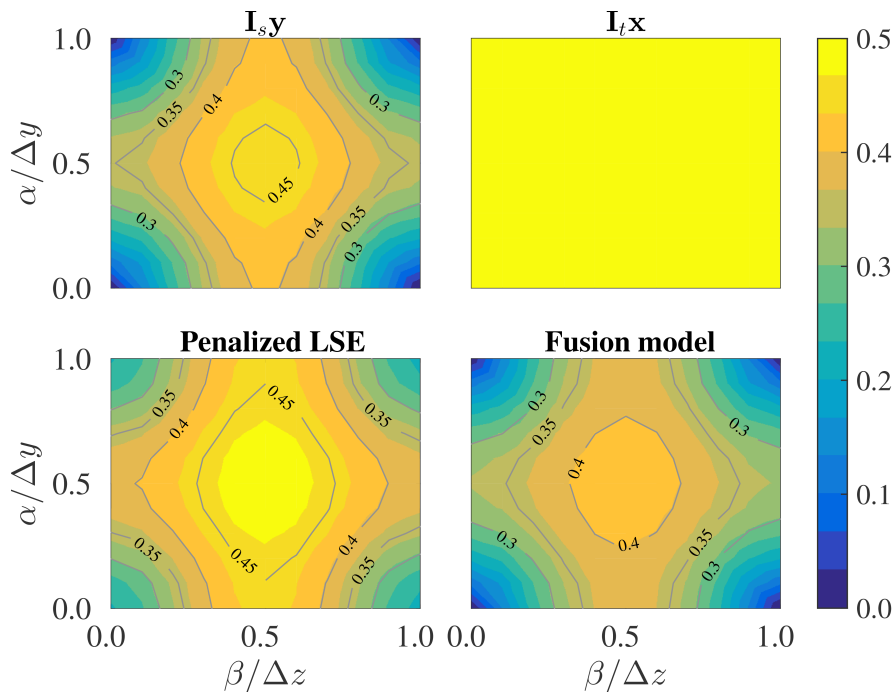


Figure 6.7 – Channel flow: NRMSEs between reference and reconstructed streamwise velocities by all methods as: functions of spatial coordinates in an element block at the most difficult instant, i.e. at  $(\alpha, \beta, N\delta t/2M)$ . Results are shown for case 5 in table 6.3, with the subsampling ratios  $N/M = 10$  in time and  $P/Q = 10 \times 10$  in space.

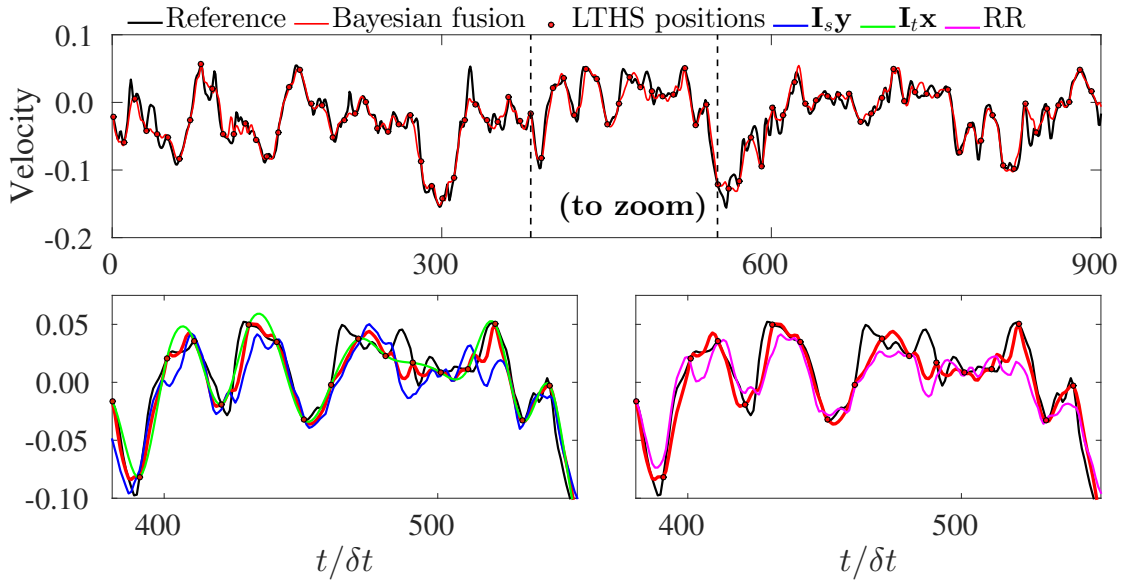


Figure 6.8 – Channel flow: a time evolution of fluctuating streamwise velocity at  $y/H = 1$  and  $z/H = 0$  of a point with the local coordinate  $(\Delta y, \Delta z, \tau)$  (see figure (1.6)). The whole evolutions of reference and fused velocity contain 900 samples. A region is zoomed in and compared with results from interpolations and RR.

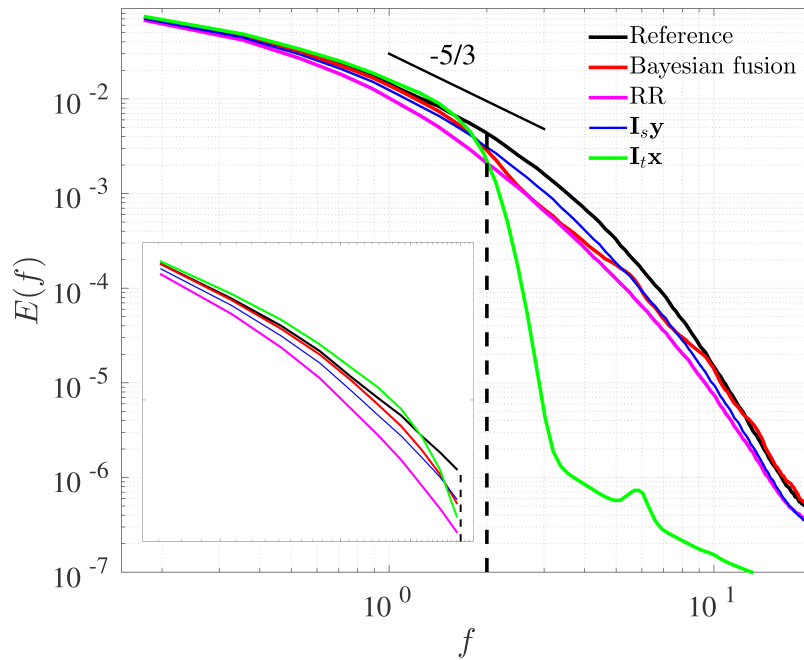


Figure 6.9 – Channel flow: spectra of the fluctuating velocity in figure 6.8.



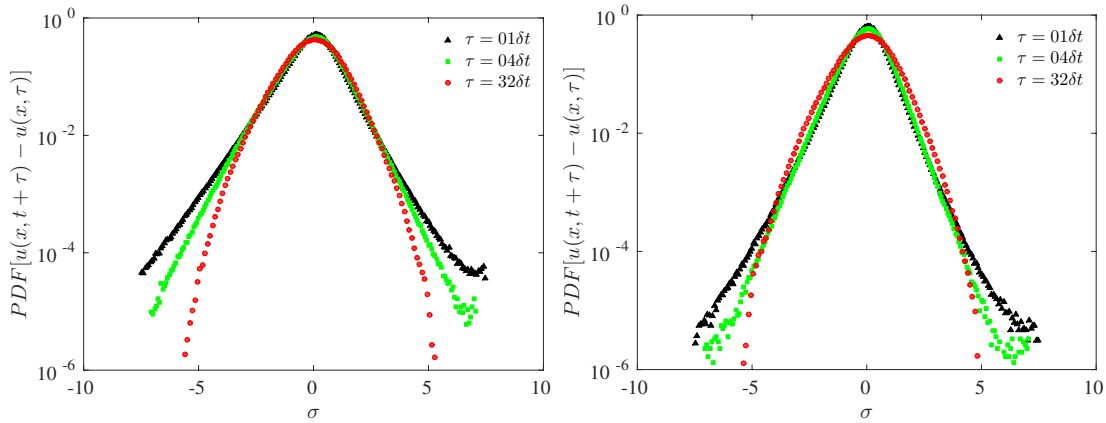


Figure 6.10 – Channel flow: probability distribution functions of velocity increments in (left) the original DNS and (right) the reconstructed field.

frequencies.

Figure 6.10 shows estimates of the probability density functions of time increments  $u(x, t + \tau) - u(x, t)$  for the original DNS field as well as for the reconstructed field. As expected, the original field displays intermittent non Gaussian distributions. More importantly, the reconstructed field, while less intermittent, still clearly exhibits non Gaussian increments at small scales. Note that the reconstruction error is essentially due to the difficulty to accurately reconstruct these small scales. It is expected that any reconstruction method will lead to fields that are less intermittent than the original one.

Figure 6.11 compares reconstructed snapshots by different methods. This snapshot is at the most remote instant from its two neighboring LTHS time steps. The model reconstructs correctly the velocity field with more flow details than spatial interpolation. It also recovers better large scales than RR and time interpolation methods.

## 6.3 Concluding remarks

This section analyzes performances of all the models discussed in this thesis. All reconstructions are gathered and compared to highlight the behaviors of different models. These comparisons also suggest further ideas to improve the models by combining several of them to take advantage of different sources of information.

Two datasets have been used, the DNS data of an isotropic turbulence and a turbulent channel flow at a moderate Reynolds number. Only the streamwise velocity component is studied, but all results can be reproduced for other components. In all configurations, HTLS and LTHS measurements are used as complementary information in space and time. Empirical models find the relation between large and small scales, while fusion

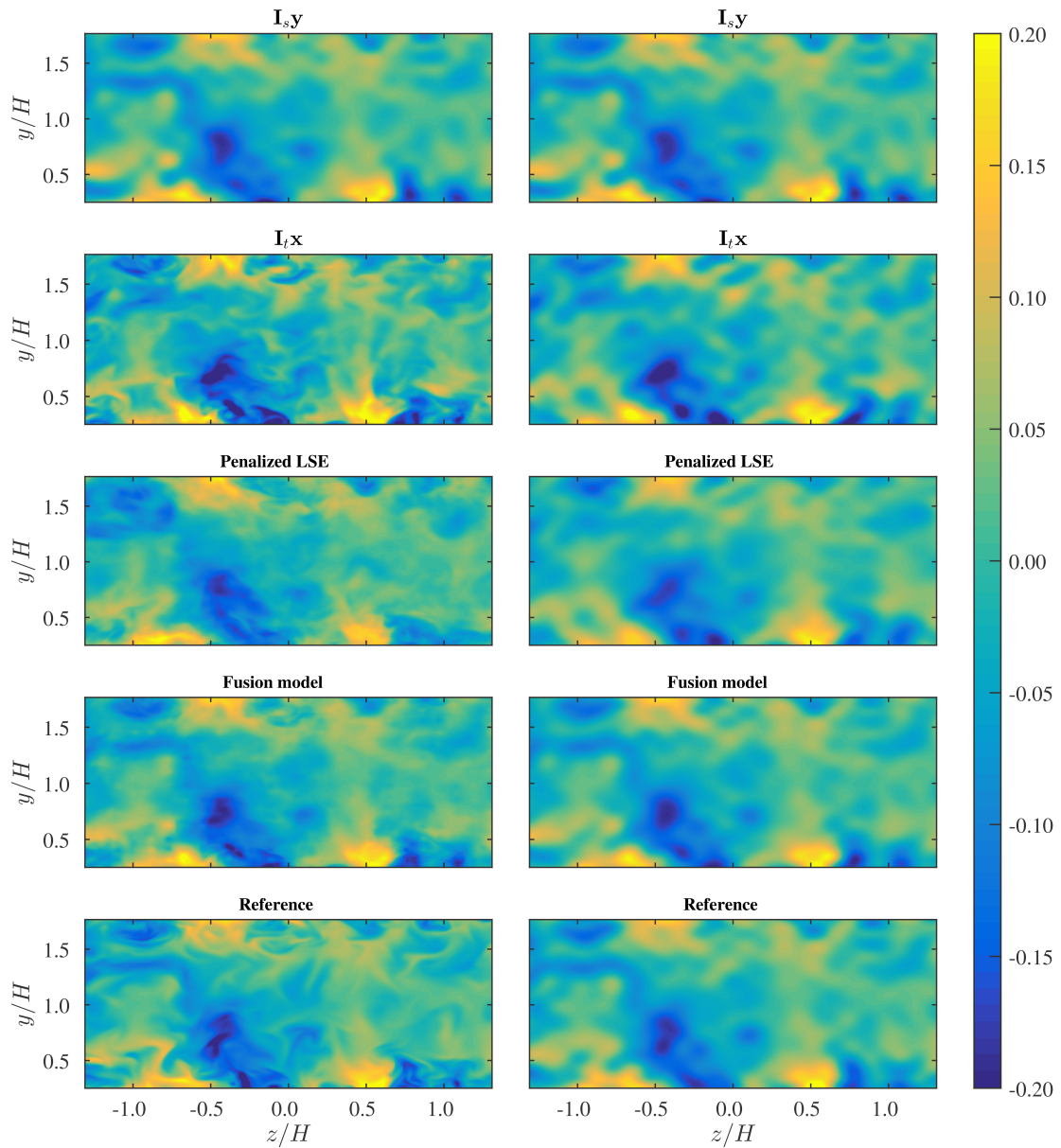


Figure 6.11 – Channel flow: a sample snapshot of fluctuating streamwise velocity at one of the most difficult instant to estimate (in the middle of two LTHS time steps): Reconstruction of all scales (left) and large scales only (right). The figure is better viewed on screen.

models combine two sources of measurements. Only the cases of balanced energy losses are investigated, since otherwise fusion models give results similar to single interpolation of the measurements with the lower energy loss.

Performances of regression models depend only on the subsampling ratio in space. This was expected since these models are not designed to take advantage of information in time. The benefits of such models are highlighted only in comparison to spatial interpolation. The improvements are significant. Kernel regression models also show their superiority compared to linear ones. This is certainly due to the non-linear relation between large and small scales in turbulence.

NLM-based propagation models reconstruct the small scales using a different prior. Small scales are assumed to be advected by larger ones. These small-scale information are then propagated along the time direction based on the similarity levels of corresponding large scales in space. Results show some advantages especially when the subsampling ratios are small in both space and time. A ratio of  $3 \times 3$  in space, corresponding to about 1% of energy loss, ensures that the model does not start from a too severe loss of information. In time, a ratio of 4, also about 1% of energy loss, ensures that the similarity level between large scales remains significant. Significant benefits are observed compared to spatial interpolation, even in rather crude subsampling ratios in space and time. A non-greedy scheme, by gradually propagating small scales from LTHS planes, gives slightly better reconstructions compared to the greedy one.

The analyses show advantages of further exploiting information in time. The Bayesian fusion model, despite its simplicity, demonstrates clear benefits compared to other methods. By simply proposing compromise estimates between the two interpolations in space and time, it leads to more accurate reconstruction at all positions in space and time. The compromise is achieved in the form of a weighted average, where weights are learned from measurements. These weights encode some information about the physics of the flow. Benefits are significant in cases of balanced energy losses. The 2D fields or 1D signal comparisons also demonstrate that the fusion model seeks a trade-off between retaining good large-scale information from a single interpolation and proposes more details from the other interpolation.

All models reconstruct the fields using different priors, i.e. exploiting information differently. Comparisons show clear advantages of combining different measurements. Further exploitations and combinations of such models can improve the reconstruction results. For instance, NLM-based propagation could start from initial large scales obtained from regression, which are more accurate than spatial interpolation. This propagated information, which exploits the local similarity of the flows, could combine with the time interpolation to further improve results of Bayesian fusion models.



# Conclusions and perspectives

This work lies in between the two research domains of turbulence and image processing. The main objective was to explore a large spectrum of approaches to estimate small-scale turbulence from given measurements at large scales only. One contribution of the thesis is to review conventional image-based methods. We have also adapted other models inspired from recent works in image processing and proposed new methods to our context.

The estimation of small-scale turbulence from measurements has been addressed via two problems described in chapter 1. The first problem is to find a relationship between large and small scales of turbulence given training samples at all scales. The second is to measure and combine complementary information, which are from space and time in this case. DNS data of an isotropic turbulence and a channel flow are used to setup different numerical experiments. These data give access to all scales of turbulence, which are used as the reference to qualify different approaches. Low-resolution fields of large scales are virtually extracted from the reference ones to which the reconstructions are compared.

For the two problems, we have proposed two families of approaches. The first one is to learn an empirical relationship between large and small scales through simple regression models or through learning coupled representations of different scales using dictionary learning. The second group of methods is based on the fusion of information. Different assumptions are exploited to propose schemes to combine available measured data. Two fusion models are developed, which use either similarity of structures in the flows or probabilistic models to find compromise estimates given the measurements.

Chapter 2 reviewed regression models, which find mapping functions between low-resolution and high-resolution fields. The function can be linear (via a set of coefficients) or nonlinear (using fixed kernel spaces). Model performances are usually sensitive to some hyper-parameters. This problem was addressed by optimizing a bias-variance trade-off using the cross-validation technique. Comparing reconstruction results, these models work better than standard spline interpolation. Nonlinear regressions also give more accurate reconstruction than linear ones.

Chapter 3 discussed *dictionary learning*, a generalization of principal component anal-

ysis (called *proper orthogonal decomposition* in turbulence studies) to learn redundant bases that better represent turbulent fields in a sparse manner. A couple of representations are learned and used to reconstruct the high-resolution fields from low-resolution measurements. In the case of direct subsampling that could happen in real experiments, the aliasing problem appears. We observe that the coupled dictionary learning method does not permit to de-alias. The model has not brought significant improvements compared to spatial interpolation. Once avoiding this problem by a prefiltering step, dictionary learning gives significant improvements compared to interpolation. In the range of scale between 0.5 and 1.5 the cutoff, it reduces the energy loss by about 60% and the total error by about 15%.

Chapter 4 discussed a non-local means-based propagation model as the first fusion model. The model was based on the hypothesis of *rapid distortion*, which assumes that small scales are advected by large ones. These small-scale information from the LTHS planes are propagated in time based on the similarity level of large scales in space. The model works very well at low subsampling ratios in space and time (corresponding to about 1% of energy loss) where it significantly improves the reconstruction accuracy (by about 40%) compared to single interpolation in space. However, the model is not very robust when the energy losses are more severe. With large subsampling ratios in space (for instance  $P/Q = 6 \times 6$ ), it suffers from severe losses and can not recover completely even at large-scale. With large ratios in time (for instance  $N/M = 8$ ), similarities of large scales decay rapidly and make propagation models less accurate.

To further exploit all available information from measurements, a simple yet efficient fusion model was proposed in chapter 5. The model estimates a high-resolution field that maximizes its *posterior* probability given the measurements. A Bayesian framework is used and further simplifications lead to a linear fusion formula, a weighted sum of the two single interpolations from the two measured data. Weighted coefficients are covariance matrices of unknown small scales, which are learned from measurements. The model reconstructs high-resolution fields with significant improvement compared to single interpolations from either sources. The superiority of this model is emphasized in the case where energy losses are balanced in space and time. In such cases, improvements are observed both at large and small scales.

Chapter 6 provided a more global view of all proposed methods. We synthesized reconstructions by all approaches for the same setups where measurements in space and time are subsampled in a balanced manner. Dictionary learning is omitted in this comparison because of its slightly different configuration of numerical experiments. Single models to solve the reconstruction problem can be predefined (interpolation) or learned adaptively from the training data (regression). Adaptive models reconstruct the

fields more accurately. More significant improvements are observed when combining complementary measurements in space and time. NLM-based model gives very accurate reconstructions at low energy losses. At higher subsampling ratios, this model is not able to over-perform temporal interpolation. Bayesian fusion model is simpler yet more efficient, and gives the best reconstruction in all cases.

## **Suggestions for future works**

The main purpose of this work was a first exploration of a set of methods for the reconstruction of fully resolved turbulent fields from low resolution measurements. Turbulent fields are highly disordered and much less regular than natural images, therefore the inverse problem of reconstruction is even more difficult in our case. This thesis has opened many new directions to develop tools for turbulence studies. Results have suggested also some new directions for signal/image processing studies to provide even more adapted tools. The present work gives rise to the following suggestions for future works.

### **Extending to the reconstruction of three-component velocity fields?**

The whole thesis has dealt with the reconstruction of streamwise velocity fields. All results can be reproduced for the other two components independently. However, all components are connected physically through Navier-Stokes equations. Further investigations are needed to exploit the information such as cross-correlation between components, vorticities and divergence of the velocity fields. One idea is to impose the prior of *divergence-free*. Another idea would be to use vorticities as features instead of derivatives to learn coupled dictionaries.

### **Ensemble of models as generalizing Bayesian fusion model?**

Chapter 6 has compared performances of all proposed models. It has shown that different models, though exploiting different sources of information or assumptions, give better results than single interpolations. Combining multi-sources of measurements through their interpolations gives more accurate reconstructions compared to one complex single model. This observation suggests to further ensemble different models and take advantage of each single model. Regressions, by learning an adaptive interpolator, could replace spline interpolations. NLM-based propagation models by exploiting the similarity information also appear as a good candidate to replace simple interpolation.

**Highly nonlinear mapping function between large and small scales of turbulence?**

This thesis has reviewed regression models, but they remain rather simplistic to describe this highly nonlinear relation between scales. This is suggested when observing advantages of kernel regressions in chapter 6. Other ideas, especially *neural network* and *deep learning* (Dong et al., 2014), could be studied. Such approaches permit much more complex and highly nonlinear mapping functions between input low-resolution and output high-resolution fields, not restricted to a fixed kernel space as the KRR model. However, such a model requires an extremely large amount of training samples. The patch-wise approach is potentially a good candidate to give access to more samples and to localize the information. Such models could be combined with sparse prior (Wang et al., 2015) discussed in chapter 3 and proven to further improve reconstruction accuracy.

**Dictionary learning for other inverse problems in turbulence?**

As a more efficient representation compared to PCA and predefined wavelets, dictionary learning has demonstrated its possibility to solve the inverse problem of reconstructing high-resolution velocities from low-resolution measurements. The improvement is significant when aliasing is handled carefully but inoperative in the case of direct subsampling. This approach is not limited to super-resolution only. Other inverse problems such as removing noise or estimating missing pixels could be studied. One particular idea is to apply dictionary learning to D time-resolved experimental data of “*Shake-The-Box*”-PIV (Schröder et al., 2015). By following the particles, the method resolves till pixel size. However, velocity fields in a uniform grid is estimated by simple interpolations. This could be done with dictionary learning. Viewing the fields with many missing pixels as random sensing, the approach could learn the missing small scales from the position it “*sees*” and propose better estimates than interpolation. Also, measurements noise could be separated from the true information in the same step.

**Dictionary learning to deal with aliasing from direct subsampling?**

Aliasing is a known problem when the sensing system directly subsamples the fields without the prefiltering step. The first attempt of dictionary learning to reconstruct the high-resolution fields from the subsampled measurements has failed potentially due to aliasing. Without its presence, the model gives clear improvements compared to single interpolation. The problem of removing aliasing could be addressed and solved using dictionary learning as well. Coupled dictionaries could be learned to represent the fields with and without aliasing. The model in chapter 3 would become a three-stage super-resolution model, with an intermediate step to remove aliasing.



### A more complete fusion model, a Bayesian way?

The fusion model proposed in this work uses only a very simple weighted sum formula. The prior of the estimated field is omitted from the formula due to the assumption of a non-informative prior. Full covariance matrices to represent all sources of space-time correlations are also simplified to diagonal ones. Some work remains to be done to further exploit the information from measurements by building more complete covariance matrices. Also, a good prior could further improve further the reconstruction. The prior could carry physical properties of the flow, for example divergence free, energy spectra or even full Navier-Stokes equations. In case the full fields are given as training data, covariance matrices could also be learned adaptively. In such cases, the weights of the fusion model could be learned to take the properties of turbulence into account.

### Combining fusion and dictionary learning?

Dictionary learning has shown to be a good representation of turbulent fields using a sparsity prior. This prior helps in solving the ill-posed problem of reconstructing high-resolution fields. Fusion models by combining multi-sources of measurements are simple but also very efficient to solve the problem. The idea of combining sparse prior and fusion could be studied, inspired by Wei et al., (2015). In chapter 5, the optimization problem is:

$$z = \operatorname{argmin}_z \left\{ \frac{1}{2} \|z - \mathbb{I}_t \mathbf{x}\|_{\Sigma_{h_t}}^2 + \frac{1}{2} \|z - \mathbb{I}_s \mathbf{y}\|_{\Sigma_{h_s}}^2 \right\} \quad (6.1)$$

By imposing the sparse prior, the problem could be rewritten as:

$$z = \mathbf{DA} \quad s.t. \quad \{\mathbf{D}, \mathbf{A}\} = \operatorname{argmin}_{\mathbf{D}, \mathbf{A}} \left\{ \frac{1}{2} \|\mathbf{DA} - \mathbb{I}_t \mathbf{x}\|_{\Sigma_{h_t}}^2 + \frac{1}{2} \|\mathbf{DA} - \mathbb{I}_s \mathbf{y}\|_{\Sigma_{h_s}}^2 + \lambda \|\mathbf{A}\|_1 \right\} \quad (6.2)$$

or avoiding the interpolations, hence not bringing aliasing terms into the system, as:

$$z = \mathbf{DA} \quad s.t. \quad \{\mathbf{D}, \mathbf{A}\} = \operatorname{argmin}_{\mathbf{D}, \mathbf{A}} \left\{ \frac{1}{2} \|\mathbf{S}_t \mathbf{DA} - \mathbf{x}\|_{\Sigma_{h_t}}^2 + \frac{1}{2} \|\mathbf{S}_s \mathbf{DA} - \mathbf{y}\|_{\Sigma_{h_s}}^2 + \lambda \|\mathbf{A}\|_1 \right\} \quad (6.3)$$

Solving the above highly non-convex problem could be addressed thanks to recent progresses in solving alternating optimization problems (Boyd et al., 2011; Parikh, Boyd, et al., 2014).

### Increase resolution of PIV measurements using different setups?

Regression and interpolation are restricted to reconstruct high-resolution fields from low-resolution ones at the same scene, while it is not the case for coupled dictionaries learning.

One very promising setup would be to combine high-resolution PIV with time-resolved PIV (Tr-PIV). High-resolution PIV can measure the flow at a small field-of-view but at very high spatial resolution. These measurements could be used to train the dictionaries. Then Tr-PIV measures the flow at a much larger field-of-view but at lower resolution. Using the trained dictionaries, one could estimate time-resolved velocity fields at large field-of-view and high spatial resolution. One should notice also the limitation of such an approach in de-aliasing to carefully design the sensing system. Another configuration could be to use two different PIV systems to measure HTLS and LTHS fields (see figure 1.5) and apply fusion models to maximize the level of useful information.

### **Co-conception design of experiments?**

The thesis has studied the performances of various models for different configurations at various subsampling ratios. This gives an idea of which method to choose for one particular setup. From a certain loss of energy due to subsamplings, this work suggests the maximum level of accuracy and scale one could expect to reconstruct. This information is very useful to design new challenging experiments in order to maximize the expected level of small scale content after post-processing. This thesis connects to the current research area of co-conception. The idea is to co-design the measurement system with respect to some pre-defined post-processing procedure.

# Bibliography

- Adrian, R. J. (1977). “On the role of conditional averages in turbulence theory”. In: *Turbulence in Liquids*. Vol. 1, pp. 323–332.
- Adrian, R. J. (1979). “Conditional eddies in isotropic turbulence”. In: *Physics of Fluids* 22, pp. 2065–2070.
- Adrian, R. J. (1992). “Stochastic estimation of conditional structure: a review”. In: *Applied scientific research*. Vol. 53. Kluwer Academic Publ, pp. 291–303.
- Adrian, Ronald J (2005). “Twenty years of particle image velocimetry”. In: *Experiments in fluids* 39.2, pp. 159–169.
- Aharon, Michal et al. (2006). “K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation”. In: *Signal Processing, IEEE Transactions on* 54.11, pp. 4311–4322.
- Beck, Amir and Marc Teboulle (2009). “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”. In: *SIAM journal on imaging sciences* 2.1, pp. 183–202.
- Bengio, Yoshua et al. (2013). “Representation learning: A review and new perspectives”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.8, pp. 1798–1828.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. ISBN: 0387310738.
- Bonnet, J. P. et al. (1994). “Stochastic estimation and proper orthogonal decomposition: complementary techniques for identifying structure”. In: *Experiments in fluids* 17.5, pp. 307–314.
- Boyd, Stephen et al. (2011). “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Foundations and Trends® in Machine Learning* 3.1, pp. 1–122.
- Breiman, Leo and Philip Spector (1992). “Submodel selection and evaluation in regression. The X-random case”. In: *International statistical review/revue internationale de Statistique*, pp. 291–319.
- Buades, Antoni et al. (2005). “A review of image denoising algorithms, with a new one”. In: *Multiscale Modeling & Simulation* 4.2, pp. 490–530.
- Burges, Christopher JC (2010). *Dimension reduction: A guided tour*. Now Publishers Inc.
- Challa, S. and D. Koks (2004). “Bayesian and Dempster-Shafer fusion”. In: *Sadhana - Academy Proceedings in Engineering Sciences* 29.2, pp. 145–174.
- Chemin, Jean-Yves and Isabelle Gallagher (2009). “Wellposedness and stability results for the Navier–Stokes equations in  $R^3$ ”. In: *Annales de l’Institut Henri Poincaré (C) Non Linear Analysis*. Vol. 26. 2. Elsevier, pp. 599–624.

- Chen, Scott Shaobing et al. (1998). “Atomic decomposition by basis pursuit”. In: *SIAM journal on scientific computing* 20.1, pp. 33–61.
- Coudert, S. et al. (2011). “Double large field stereoscopic PIV in a high Reynolds number turbulent boundary layer”. In: *Experiments in Fluids* 50.1, pp. 1–12.
- Dang, Hong-Phuong and Pierre Chainais (2016). “Towards Dictionaries of Optimal Size: A Bayesian Non Parametric Approach”. In: *Journal of Signal Processing Systems*.
- Daubechies, I et al. (2004). “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint”. In: *Communications on Pure and Applied Mathematics* 57.11, pp. 1413–1457.
- Davidson, P. et al. (2015). *Turbulence: an introduction for scientists and engineers*. Oxford University Press, USA.
- Dong, Chao et al. (2014). “Image super-resolution using deep convolutional networks”. In: *arXiv preprint arXiv:1501.00092*.
- Donoho, David L and Michael Elad (2003). “Optimally sparse representation in general (nonorthogonal) dictionaries via  $l_1$  minimization”. In: *Proceedings of the National Academy of Sciences* 100.5, pp. 2197–2202.
- Donoho, David L et al. (2012). “Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit”. In: *Information Theory, IEEE Transactions on* 58.2, pp. 1094–1121.
- Durgesh, V. and J. W. Naughton (2010). “Multi-time-delay LSE-POD complementary approach applied to unsteady high-Reynolds-number near wake flow”. In: *Experiments in fluids* 49.3, pp. 571–583.
- Durrant-Whyte, H. and T. C. Henderson (2008). “Multisensor data fusion”. In: *Springer handbook of robotics*. Springer, pp. 585–610.
- Efron, Bradley et al. (2004). “Least angle regression”. In: *The Annals of statistics* 32.2, pp. 407–499.
- Elad, Michael et al. (2010). “On the Role of Sparse and Redundant Representations in Image Processing”. In: *Proceedings of the IEEE- Special issue on applications of sparse representation and compressive sensing* 98.6, pp. 972–982. ISSN: 0018-9219. DOI: 10.1109/JPROC.2009.2037655.
- Elsinga, GE et al. (2005). “Assessment of Tomo-PIV for three-dimensional flows”. In: *6th International Symposium on Particle Image Velocimetry, Pasadena, California, USA*.
- Elsinga, Gerrit E et al. (2006). “Tomographic particle image velocimetry”. In: *Experiments in fluids* 41.6, pp. 933–947.
- Engan, Kjersti et al. (1999). “Method of optimal directions for frame design”. In: *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*. Vol. 5. IEEE, pp. 2443–2446.
- Ewing, D. and J. H. Citriniti (1999). “Examination of a LSE/POD complementary technique using single and multi-time information in the axisymmetric shear layer”. In: *IUTAM Symposium on Simulation and Identification of Organized Structures in Flows Fluid Mechanics and Its Applications*. Vol. 52. Springer, pp. 375–384.
- Foi, Alessandro and Giacomo Boracchi (2016). “Foveated Nonlocal Self-Similarity”. In: *International Journal of Computer Vision*, pp. 1–33. ISSN: 1573-1405. DOI: 10.1007/s11263-016-0898-1. URL: <http://dx.doi.org/10.1007/s11263-016-0898-1>.

- Fox, Douglas G and Douglas K Lilly (1972). “Numerical simulation of turbulent flows”. In: *Reviews of Geophysics* 10.1, pp. 51–72.
- Frisch, U. (1995). *Turbulence: the legacy of A.N. Kolmogorov*. Cambridge University Press.
- Gallagher, Isabelle (2015). “Some stability results on global solutions to the Navier–Stokes equations”. In: *Analysis* 35.3, pp. 177–184.
- Gallagher, Isabelle et al. (2003). “Asymptotics and stability for global solutions to the Navier–Stokes equations”. In: *Annales de l’Institut Fourier*. Vol. 53. 5, pp. 1387–1424.
- Gelman, A. et al. (2013). *Bayesian data analysis*. 3rd ed. Chapman and Hall/CRC. ISBN: 1439840954, 978-1439840955.
- George, William K (2009). “Lectures in Turbulence for the 21st Century”. In: *Chalmers University of Technology*.
- Gordeyev, Stanislav V and Flint O Thomas (2000). “Coherent structure in the turbulent planar jet. Part 1. Extraction of proper orthogonal decomposition eigenmodes and their self-similarity”. In: *Journal of Fluid Mechanics* 414, pp. 145–194.
- Guezennec, Y. G. (1989). “Stochastic estimation of coherent structures in turbulent boundary layers”. In: *Physics of Fluids A: Fluid Dynamics (1989-1993)* 1.6, pp. 1054–1060.
- Hastie, T. et al. (2009a). *The elements of statistical learning*. 2nd ed. New York, NY, USA: Springer. ISBN: 0387848576, 978-0387848570.
- Hastie, Trevor et al. (2005). “The elements of statistical learning: data mining, inference and prediction”. In: *The Mathematical Intelligencer* 27.2, pp. 83–85.
- Hastie, Trevor et al. (2009b). *Unsupervised learning*. Springer.
- Hoyas, Sergio and Javier Jiménez (2006). “Scaling of the velocity fluctuations in turbulent channels up to  $Re\tau = 2003$ ”. In: *Physics of Fluids (1994-present)* 18.1, p. 011702.
- Hutchins, N and Ivan Marusic (2007). “Evidence of very long meandering features in the logarithmic region of turbulent boundary layers”. In: *Journal of Fluid Mechanics* 579, pp. 1–28.
- Ishihara, Takashi et al. (2009). “Study of high-Reynolds number isotropic turbulence by direct numerical simulation”. In: *Annual Review of Fluid Mechanics* 41, pp. 165–180.
- Jolliffe, Ian (2002). *Principal component analysis*. Wiley Online Library.
- Jolliffe, Ian T (1982). “A note on the use of principal components in regression”. In: *Applied Statistics*, pp. 300–303.
- Joshi, M. and A. Jalobeanu (2010). “MAP estimation for multiresolution fusion in remotely sensed images using an IGMRF prior model”. In: *IEEE Transactions on Geoscience and Remote Sensing* 48.3, pp. 1245–1255.
- Kaneda, Yukio et al. (2003). “Energy dissipation rate and energy spectrum in high resolution direct numerical simulations of turbulence in a periodic box”. In: *Physics of Fluids* 15.2, p. L21.
- Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. ISBN: 0-13-345711-7.
- Keane, R. D. and R. J. Adrian (1992). “Theory of cross-correlation analysis of PIV images”. In: *Applied scientific research* 49.3, pp. 191–215.
- Kendall, M. G. et al., eds. (1987). *Kendall’s Advanced Theory of Statistics*. New York, NY, USA: Oxford University Press, Inc. ISBN: 0-195-20561-8.

- King, Louis Vessot (1914). “On the convection of heat from small cylinders in a stream of fluid: determination of the convection constants of small platinum wires with applications to hot-wire anemometry”. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, pp. 373–432.
- Kohavi, Ron et al. (1995). “A study of cross-validation and bootstrap for accuracy estimation and model selection”. In: *Ijcai*. Vol. 14. 2, pp. 1137–1145.
- Koks, D. and S. Challa (2003). *An introduction to bayesian and dempster-shafer data fusion*. Tech. rep. DSTO Systems Sciences Laboratory.
- Le, Hung et al. (1997). “Direct numerical simulation of turbulent flow over a backward-facing step”. In: *Journal of fluid mechanics* 330, pp. 349–374.
- Lee, Honglak et al. (2006). “Efficient sparse coding algorithms”. In: *Advances in neural information processing systems*, pp. 801–808.
- Lee, Myoungkyu and Robert D Moser (2014). “Direct numerical simulation of turbulent channel flow up to Re tau approx 5200”. In: *arXiv preprint arXiv:1410.7809*.
- Lee, Myoungkyu and Robert D Moser (2015). “Direct numerical simulation of turbulent channel flow up to”. In: *Journal of Fluid Mechanics* 774, pp. 395–415.
- Leviton, E. and G. T. Herman (1987). “A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography”. In: *IEEE Transactions on Medical Imaging* 6.3, pp. 185–192.
- Lumley, John (1967). “The structure of inhomogenous turbulent flows”. In: *Asmospheric turbulence and radio wave propagation*, pp. 166–178.
- Ly, Hung V and Hien T Tran (2001). “Modeling and control of physical processes using proper orthogonal decomposition”. In: *Mathematical and computer modelling* 33.1, pp. 223–236.
- Ma, B. et al. (2000). “Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion”. In: *IEEE Transactions on Intelligent Transportation Systems* 1.3, pp. 135–147.
- Mairal, Julien et al. (2010). “Online Learning for Matrix Factorization and Sparse Coding”. In: *Journal of machine learning research* 11, pp. 19–60. ISSN: 1532-4435.
- Mallat, Stéphane (1999). *A wavelet tour of signal processing*. Academic press.
- Mallat, Stephane G (1989). “A theory for multiresolution signal decomposition: the wavelet representation”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 11.7, pp. 674–693.
- Marquillie, M. et al. (2008). “Direct numerical simulation of a separated channel flow with a smooth profile”. In: *Journal of Turbulence* 9.1, pp. 1–23.
- McDonough, JM (2004). “Introductory lectures on turbulence physics, mathematics and modeling”. In:
- Melnick, M. B. and B. S. Thurow (2012). “Experimental Investigation of a Turbulent Boundary Layer Using Simultaneous 3-D Flow Visualization and 2-D PIV”. In: *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, pp. 07–52.
- Meyer, R. D. and G. Tan (2014). “Provide detailed and real-time indoor environmental information using POD–LSE and limited measurements”. In: *Energy and Buildings* 73, pp. 59–68.

- Mokhasi, P. et al. (2009). “Predictive flow-field estimation”. In: *Physica D: Nonlinear Phenomena* 238.3, pp. 290–308.
- Nguyen, Linh Van et al. (2015). “A Bayesian fusion model for space-time reconstruction of finely resolved velocities in turbulent flows from low resolution measurements”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2015.10, P10008.
- Nguyen, T. D. et al. (2010). “Proper orthogonal decomposition-based estimations of the flow field from particle image velocimetry wall-gradient measurements in the backward-facing step flow”. In: *Measurement Science and Technology* 21.11, p. 115406.
- Olshausen, Bruno A and David J. Field (1996). “Emergence of simple-cell receptive field properties by learning a sparse code for natural images”. In: *Nature* 381.6583, pp. 607–609.
- Orszag, Steven A and GS Patterson Jr (1972). “Numerical simulation of three-dimensional homogeneous isotropic turbulence”. In: *Physical Review Letters* 28.2, p. 76.
- Parikh, Neal, Stephen P Boyd, et al. (2014). “Proximal Algorithms.” In: *Foundations and Trends in optimization* 1.3, pp. 127–239.
- Pedregosa, F. et al. (2011). “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.
- Prasad, Ajay K (2000). “Stereoscopic particle image velocimetry”. In: *Experiments in fluids* 29.2, pp. 103–116.
- Protter, Matan et al. (2009). “Generalizing the nonlocal-means to super-resolution reconstruction”. In: *Image Processing, IEEE Transactions on* 18.1, pp. 36–51.
- Ravindran, SS (2000). “A reduced-order approach for optimal control of fluids using proper orthogonal decomposition”. In: *International journal for numerical methods in fluids* 34.5, pp. 425–448.
- Reynolds, O. (1883). “An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels”. In: *Proceedings of the royal society of London* 35.224-226, pp. 84–99.
- Roweis, Sam and Zoubin Ghahramani (1999). “A unifying review of linear Gaussian models”. In: *Neural computation* 11.2, pp. 305–345.
- Rubinstein, Ron et al. (2010). “Dictionaries for sparse representation modeling”. In: *Proceedings of the IEEE* 98.6, pp. 1045–1057.
- Sage, A. P. and J. L. Melsa (1971). *Estimation theory with applications to communications and control*. Tech. rep. DTIC Document.
- Saunders, Craig et al. (1998). “Ridge regression learning algorithm in dual variables”. In: *(ICML-1998) Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, pp. 515–521.
- Scarano, F (2013). “Tomographic PIV: principles and practice”. In: *Measurement Science and Technology* 24.1, p. 012001.
- Schröder, A et al. (2015). “Advances of PIV and 4D-PTV” Shake-The-Box” for Turbulent Flow Analysis—the Flow over Periodic Hills”. In: *Flow, Turbulence and Combustion* 95.2-3, pp. 193–209.
- Shin, Vladimir et al. (2006). “Generalized Millman’s formula and its application for estimation problems”. In: *Signal Processing* 86.2, pp. 257–266.

- Sillero, Juan A et al. (2013). “One-point statistics for turbulent wall-bounded flows at Reynolds numbers up to  $\delta^+$  approx 2000”. In: *Physics of Fluids (1994-present)* 25.10, p. 105102.
- Smits, Alexander J and Ivan Marusic (2013). “Wall-bounded turbulence”. In: *Phys. Today* 66.9, pp. 25–30.
- Spalart, Philippe R (1986). “Numerical study of sink-flow boundary layers”. In: *Journal of Fluid Mechanics* 172, pp. 307–328.
- Spalart, Philippe R (1988). “Direct simulation of a turbulent boundary layer up to  $R_\theta = 1410$ ”. In: *Journal of Fluid Mechanics* 187, pp. 61–98.
- Sutton, MA et al. (1983). “Determination of displacements using an improved digital correlation method”. In: *Image and vision computing* 1.3, pp. 133–139.
- Taylor, JA and MN Glauser (2004). “Towards practical flow sensing and control via POD and LSE based low-dimensional tools”. In: *Journal of fluids engineering* 126.3, pp. 337–345.
- Tennekes, H. and J. L. Lumley (1972). *A first course in turbulence*. MIT Press.
- Tibshirani, Robert (1996). “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288.
- Tošić, Ivana and Pascal Frossard (2011). “Dictionary learning”. In: *Signal Processing Magazine, IEEE* 28.2, pp. 27–38.
- Trees, H. L. Van (1967). *Detection, estimation, and modulation theory, optimum array processing*. Vol. 1. NY, USA: John Wiley & Sons.
- Tropp, Joel A and Anna C Gilbert (2007). “Signal recovery from random measurements via orthogonal matching pursuit”. In: *Information Theory, IEEE Transactions on* 53.12, pp. 4655–4666.
- Tu, J. H. et al. (2013). “Integration of non-time-resolved PIV and time-resolved velocity point sensors for dynamic estimation of velocity fields”. In: *Experiments in fluids* 54.2, pp. 1–20.
- Vonesch, Cédric and Michael Unser (2008). “A fast thresholded Landweber algorithm for wavelet-regularized multidimensional deconvolution”. In: *Image Processing, IEEE Transactions on* 17.4, pp. 539–549.
- Wang, Zhaowen et al. (2015). “Deep Networks for Image Super-Resolution with Sparse Prior”. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 370–378.
- Wei, Qi et al. (2015). “Hyperspectral and multispectral image fusion based on a sparse representation”. In: *IEEE Transactions on Geoscience and Remote Sensing* 53.7, pp. 3658–3668.
- Willert, Christian E (2015). “High-speed particle image velocimetry for the efficient measurement of turbulence statistics”. In: *Experiments in Fluids* 56.1, pp. 1–17.
- Yang, Jianchao et al. (2010). “Image Super-Resolution Via Sparse Representation”. In: *IEEE transactions on image processing* 19.11, pp. 2861–2873. ISSN: 1057-7149. DOI: 10.1109/TIP.2010.2050625.
- Yang, Jianchao et al. (2012). “Coupled Dictionary Training for Image Super-Resolution”. In: *IEEE transactions on image processing* 21.8, pp. 3467–3478. ISSN: 1057-7149. DOI: 10.1109/TIP.2012.2192127.
- Zeyde, Roman et al. (2012). “On single image scale-up using sparse-representations”. In: *Curves and Surfaces*. Springer, pp. 711–730.



- 
- Zhang, Y. et al. (2009). “Noise-resistant wavelet-based Bayesian fusion of multispectral and hyperspectral images”. In: *IEEE Transactions on Geoscience and Remote Sensing* 47.11, pp. 3834–3843.
- Zimmer, Sebastian et al. (2008). “A rotationally invariant block matching strategy improving image denoising with non-local means”. In: *Proc. 2008 International Workshop on Local and Non-Local Approximation in Image Processing*, pp. 135–142.



## Bayesian framework

### A.1 Derivation of Linear Gaussian models

#### A.1.1 Bayes' theorem for Gaussian variables

Let  $\mathbf{v}$  is the measurement, then the probability  $p(\mathbf{v})$  is known. Let  $\mathbf{u}$  is the desired variable, with the prior distribution  $p(\mathbf{u})$ . Suppose also there exists a linear observation model:

$$\mathbf{v} = A\mathbf{u} + \mathbf{n} \quad (\text{A.1})$$

where  $\mathbf{n}$  is a zero-mean random vectors and independent on  $\mathbf{u}$  and  $\mathbf{v}$  and has a precision matrix denoted as  $\Lambda_{\mathbf{v}|\mathbf{u}} = \Sigma_{\mathbf{v}|\mathbf{u}}^{-1}$ . One can express the joint probability  $p(\mathbf{w})$  as the product of the prior probability  $p(\mathbf{u})$  and the likelihood function of the measurement given the variable  $p(\mathbf{v}|\mathbf{u})$  as:

$$p(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\boldsymbol{\mu}_u, \Lambda_u^{-1}) \quad (\text{A.2})$$

$$p(\mathbf{v}|\mathbf{u}) = \mathcal{N}(\mathbf{v}|A\boldsymbol{\mu}_u, \Lambda_{\mathbf{v}|\mathbf{u}}^{-1}) \quad (\text{A.3})$$

To derive, let take the natural logarithm of both sides, using also the symmetrical shape of precision matrices, the left hand side gives:

$$\begin{aligned} \ln p(\mathbf{w}) &= -\frac{1}{2}(\mathbf{w} - \boldsymbol{\mu}_w)^T \Lambda_w (\mathbf{w} - \boldsymbol{\mu}_w) + \text{const} \\ &= -\frac{1}{2}\mathbf{w}^T \Lambda_w \mathbf{w} + \mathbf{w}^T \Lambda_w \boldsymbol{\mu}_w \\ &\quad -\frac{1}{2}\boldsymbol{\mu}_w^T \Lambda_w \boldsymbol{\mu}_w + \text{const} \end{aligned} \quad (\text{A.4})$$

and the right hand side gives:

$$\begin{aligned} \ln(p(\mathbf{u})p(\mathbf{v}|\mathbf{u})) &= (\mathbf{u} - \boldsymbol{\mu}_u)^T \Lambda_u (\mathbf{u} - \boldsymbol{\mu}_u) \\ &\quad -\frac{1}{2}(\mathbf{v} - A\mathbf{u})^T \Lambda_{\mathbf{v}|\mathbf{u}} (\mathbf{v} - A\mathbf{u}) \\ &\quad + \text{const} \end{aligned} \quad (\text{A.5})$$

In above equations, *const* represents all the terms that are independent on  $\mathbf{u}$  and  $\mathbf{v}$ . The first order terms in the above equation include:

$$\begin{aligned} 1^{st} \text{ order terms} &= \mathbf{u}^T \Lambda_u \boldsymbol{\mu}_u - \mathbf{u}^T A^T \Lambda_{v|u} \mathbf{v} \\ &= \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}^T \begin{pmatrix} \Lambda_u \boldsymbol{\mu}_u \\ 0 \end{pmatrix} \end{aligned} \quad (\text{A.6})$$

and the second order terms are:

$$\begin{aligned} 2^{nd} \text{ order terms} &= -\frac{1}{2} \mathbf{u}^T (\Lambda_u + A^T \Lambda_{v|u} A) \mathbf{u} - \frac{1}{2} \mathbf{v}^T \Lambda_{v|u} \mathbf{v} \\ &\quad + \frac{1}{2} \mathbf{v}^T \Lambda_{v|u} A \mathbf{u} + \mathbf{u}^T A^T \Lambda_{v|u} \mathbf{v} \\ &= -\frac{1}{2} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}^T \begin{pmatrix} A^T \Lambda_{v|u} A + \Lambda_u & A^T \Lambda_{v|u} \\ -\Lambda_{v|u} A & \Lambda_{v|u} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \\ &= -\frac{1}{2} \mathbf{w}^T \Lambda_w \mathbf{w} \end{aligned} \quad (\text{A.7})$$

Comparing above equations A.7, A.6 and A.4, Gaussian distribution of  $\mathbf{w}$  takes the precision matrix  $\Lambda_w$  of the form:

$$\Lambda_w = \begin{pmatrix} \Lambda_{uu} & \Lambda_{uv} \\ \Lambda_{vu} & \Lambda_{vv} \end{pmatrix} = \begin{pmatrix} A^T \Lambda_{v|u} A + \Lambda_u & A^T \Lambda_{v|u} \\ -\Lambda_{v|u} A & \Lambda_{v|u} \end{pmatrix} \quad (\text{A.8})$$

and the mean  $\boldsymbol{\mu}_w$  is:

$$\boldsymbol{\mu}_w = \begin{pmatrix} \boldsymbol{\mu}_u \\ \boldsymbol{\mu}_v \end{pmatrix} = \Lambda_w^{-1} \begin{pmatrix} \Lambda_u \boldsymbol{\mu}_u \\ 0 \end{pmatrix} = \Sigma_w \begin{pmatrix} \Lambda_u \boldsymbol{\mu}_u \\ 0 \end{pmatrix} \quad (\text{A.9})$$

The covariance matrix is obtained by taking the inverse of above:

$$\Sigma_w = \begin{pmatrix} \Sigma_{uu} & \Sigma_{uv} \\ \Sigma_{vu} & \Sigma_{vv} \end{pmatrix} = \begin{pmatrix} A^T \Lambda_{v|u} A + \Lambda_u & A^T \Lambda_{v|u} \\ -\Lambda_{v|u} A & \Lambda_{v|u} \end{pmatrix}^{-1} \quad (\text{A.10})$$

Using the inversion lemma:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} M & -MBD^{-1} \\ D^{-1} + D^{-1}CM & D^{-1}CMBD^{-1} \end{pmatrix} \quad (\text{A.11})$$

where  $M = (A - BD^{-1}C)^{-1}$  is the *Schur complement* of the matrix inversion with respect to  $D$ . The covariance matrix is finally given as:

$$\Sigma_w = \begin{pmatrix} \Sigma_{uu} & \Sigma_{uv} \\ \Sigma_{vu} & \Sigma_{vv} \end{pmatrix} = \begin{pmatrix} \Lambda_u^{-1} & \Lambda_u^{-1} A^T \\ A \Lambda_u^{-1} & \Lambda_{v|u}^{-1} + A \Lambda_u^{-1} A^T \end{pmatrix} \quad (\text{A.12})$$

Replacing this covariance matrix to equation (A.10), the expectation of  $\mathbf{w}$  is given as:

$$\boldsymbol{\mu}_{\mathbf{w}} = \begin{pmatrix} \boldsymbol{\mu}_{\mathbf{u}} \\ \boldsymbol{\mu}_{\mathbf{v}} \end{pmatrix} = \begin{pmatrix} \Lambda_{\mathbf{u}}^{-1} & \Lambda_{\mathbf{u}}^{-1} A^T \\ A \Lambda_{\mathbf{u}}^{-1} & \Lambda_{\mathbf{v}|\mathbf{u}}^{-1} + A \Lambda_{\mathbf{u}}^{-1} A^T \end{pmatrix} \begin{pmatrix} \Lambda_{\mathbf{u}} \boldsymbol{\mu}_{\mathbf{u}} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu}_{\mathbf{u}} \\ A \boldsymbol{\mu}_{\mathbf{u}} \end{pmatrix} \quad (\text{A.13})$$

From equation A.12 and A.13, the Gaussian distribution of  $\mathbf{v} = \mathcal{N}(\mathbf{v}|\boldsymbol{\mu}_{\mathbf{v}}, \Sigma_{\mathbf{v}})$  is:

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{v}} &= A \boldsymbol{\mu}_{\mathbf{u}} \\ \Sigma_{\mathbf{v}} &= \Lambda_{\mathbf{v}|\mathbf{u}}^{-1} + A \Lambda_{\mathbf{u}}^{-1} A^T \end{aligned} \quad (\text{A.14})$$

Using results of conditional Gaussian distribution as in equation ??, the posterior distribution  $\mathbf{u}|\mathbf{v}$  is  $\mathcal{N}(\mathbf{u}|\boldsymbol{\mu}_{\mathbf{u}|\mathbf{v}}, \Sigma_{\mathbf{u}|\mathbf{v}})$  where:

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{u}|\mathbf{v}} &= \boldsymbol{\mu}_{\mathbf{u}} - \Lambda_{\mathbf{u}\mathbf{u}}^{-1} \Lambda_{\mathbf{u}\mathbf{v}} (\mathbf{v} - \boldsymbol{\mu}_{\mathbf{v}}) \\ \Sigma_{\mathbf{u}|\mathbf{v}} &= \Lambda_{\mathbf{u}\mathbf{u}}^{-1} \end{aligned} \quad (\text{A.15})$$

Taking the partial precision matrices from equation A.8, covariance matrix  $\Sigma_{\mathbf{u}|\mathbf{v}}$  is given as:

$$\Sigma_{\mathbf{u}|\mathbf{v}} = (A^T \Lambda_{\mathbf{v}|\mathbf{u}} A + \Lambda_{\mathbf{u}})^{-1} \quad (\text{A.16})$$

and conditional mean  $\boldsymbol{\mu}_{\mathbf{u}|\mathbf{v}}$  is as:

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{u}|\mathbf{v}} &= \boldsymbol{\mu}_{\mathbf{u}} - (A^T \Lambda_{\mathbf{v}|\mathbf{u}} A + \Lambda_{\mathbf{u}})^{-1} (-A^T \Lambda_{\mathbf{v}|\mathbf{u}}) (\mathbf{v} - A \boldsymbol{\mu}_{\mathbf{u}}) \\ &= (A^T \Lambda_{\mathbf{v}|\mathbf{u}} A + \Lambda_{\mathbf{u}})^{-1} (A^T \Lambda_{\mathbf{v}|\mathbf{u}} \mathbf{v} + \Lambda_{\mathbf{u}} \boldsymbol{\mu}_{\mathbf{u}}) \end{aligned} \quad (\text{A.17})$$

Final results written in covariance matrices notation, given:

$$\begin{aligned} \mathbf{v} &= A \mathbf{u} \\ p(\mathbf{u}) &= \mathcal{N}(\mathbf{u}|\boldsymbol{\mu}_{\mathbf{u}}, \Sigma_{\mathbf{u}}) \\ p(\mathbf{v}|\mathbf{u}) &= \mathcal{N}(\mathbf{v}|A \mathbf{u}, \Sigma_{\mathbf{v}|\mathbf{u}}) \end{aligned} \quad (\text{A.18})$$

one gets:

$$\begin{aligned} p(\mathbf{v}) &= \mathcal{N}(\mathbf{v}|A \boldsymbol{\mu}_{\mathbf{u}}, \Sigma_{\mathbf{v}|\mathbf{u}} + A \Sigma_{\mathbf{u}} A^T) \\ p(\mathbf{u}|\mathbf{v}) &= \mathcal{N}(\mathbf{u}|\Sigma_{\mathbf{u}|\mathbf{v}} (A^T \Lambda_{\mathbf{v}|\mathbf{u}} \mathbf{v} + \Lambda_{\mathbf{u}} \boldsymbol{\mu}_{\mathbf{u}}), \Sigma_{\mathbf{u}|\mathbf{v}}) \\ \Sigma_{\mathbf{u}|\mathbf{v}} &= (A^T \Sigma_{\mathbf{v}|\mathbf{u}}^{-1} A + \Sigma_{\mathbf{u}}^{-1})^{-1} \end{aligned} \quad (\text{A.19})$$

## A.1.2 The model

As one of the example linear Gaussian model in Roweis and Ghahramani, 1999 and restated in Bishop, 2006, given the Gaussian distribution of measurements  $p(\mathbf{u})$  and posterior distribution  $p(\mathbf{u}|\mathbf{v})$ :

$$\begin{aligned} p(\mathbf{u}) &= \mathcal{N}(\mathbf{u}|\boldsymbol{\mu}_{\mathbf{u}}, \Sigma_{\mathbf{u}}) \\ p(\mathbf{v}|\mathbf{u}) &= \mathcal{N}(\mathbf{v}|A \mathbf{u} + \mathbf{b}, \Sigma_{\mathbf{v}|\mathbf{u}}) \end{aligned} \quad (\text{A.20})$$

where  $\mu_u$ ,  $A$  and  $b$  are parameters governing the mean of the distributions;  $\Sigma_{v|u}$  is not a function of  $u$ . Applying Bayes's theorem for these Gaussian variables, the joint distribution can be written using the prior and likelihood distributions given as:

$$\begin{aligned} p(v) &= \mathcal{N}(v | A\mu_u + b, \Sigma_{v|u} + A\Sigma_u A^T) \\ p(u|v) &= \mathcal{N}(u | \Sigma_{u|v} (A^T \Lambda_{v|u} (v - b) + \Lambda_u \mu_u), \Sigma_{u|v}) \\ \Sigma_{u|v} &= (A^T \Sigma_{v|u}^{-1} A + \Sigma_u^{-1})^{-1} \end{aligned} \quad (\text{A.21})$$

Using results in equation A.21, given the posterior distributions  $\mathcal{N}(z | \mathbb{I}_t \mathbf{x}, \Sigma_{h_t})$  and  $\mathcal{N}(z | \mathbb{I}_s \mathbf{y}, \Sigma_{h_s})$  in equations 5.13 and 5.14, and known distributions of measurements  $\mathcal{N}(\mathbb{I}_t \mathbf{x} | 0, \Sigma_{\mathbb{I}_t \mathbf{x}})$  and  $\mathcal{N}(\mathbb{I}_s \mathbf{y} | 0, \Sigma_{\mathbb{I}_s \mathbf{y}})$ . Results of above section show that one can get the likelihood functions and the prior distribution as:

$$\begin{aligned} \mathbb{I}_t \mathbf{x} | z &: \mathcal{N}(\mathbb{I}_t \mathbf{x} | \Sigma_t \Sigma_{h_t}^{-1} z, \Sigma_t) \\ \mathbb{I}_s \mathbf{y} | z &: \mathcal{N}(\mathbb{I}_s \mathbf{y} | \Sigma_s \Sigma_{h_s}^{-1} z, \Sigma_s) \\ z &: \mathcal{N}(z | 0, \Sigma_z) \end{aligned} \quad (\text{A.22})$$

where:

$$\begin{aligned} \Sigma_t^{-1} &= \Sigma_{h_t}^{-1} + \Sigma_{\mathbb{I}_t \mathbf{x}}^{-1} \\ \Sigma_s^{-1} &= \Sigma_{h_s}^{-1} + \Sigma_{\mathbb{I}_s \mathbf{y}}^{-1} \\ \Sigma_z &= \Sigma_{h_t} + \Sigma_{\mathbb{I}_t \mathbf{x}} = \Sigma_{h_s} + \Sigma_{\mathbb{I}_s \mathbf{y}} \end{aligned} \quad (\text{A.23})$$

Using above resulting Gaussian distributions, The MAP estimation of  $z$  is given as:

$$\begin{aligned} \hat{z} &= \underset{z}{\operatorname{argmax}} \{p(\mathbb{I}_t \mathbf{x} | z) p(\mathbb{I}_s \mathbf{y} | z) p(z)\} \\ &= \underset{z}{\operatorname{argmin}} \{-\ln p(\mathbb{I}_t \mathbf{x} | z) - \ln p(\mathbb{I}_s \mathbf{y} | z) - \ln p(z)\} \end{aligned} \quad (\text{A.24})$$

Logarithm of above probabilities are given as:

$$\begin{aligned} -2 \ln p(\mathbb{I}_t \mathbf{x} | z) &= (\mathbb{I}_t \mathbf{x} - \Sigma_t \Sigma_{h_t}^{-1} z)^T \Sigma_t^{-1} (\mathbb{I}_t \mathbf{x} - \Sigma_t \Sigma_{h_t}^{-1} z) \\ &= \|\Sigma_t \Sigma_{h_t}^{-1} z - \mathbb{I}_t \mathbf{x}\|_{\Sigma_t}^2 \\ -2 \ln p(\mathbb{I}_s \mathbf{y} | z) &= (\mathbb{I}_s \mathbf{y} - \Sigma_s \Sigma_{h_s}^{-1} z)^T \Sigma_s^{-1} (\mathbb{I}_s \mathbf{y} - \Sigma_s \Sigma_{h_s}^{-1} z) \\ &= \|\Sigma_s \Sigma_{h_s}^{-1} z - \mathbb{I}_s \mathbf{y}\|_{\Sigma_s}^2 \\ -2 \ln p(z) &= z^T \Sigma_z^{-1} z = \|z\|_{\Sigma_z}^2 \end{aligned}$$

The cost function  $C(z)$  is defined as the sum of above terms:

$$C(z) = \|\Sigma_t \Sigma_{h_t}^{-1} z - \mathbb{I}_t \mathbf{x}\|_{\Sigma_t}^2 + \|\Sigma_s \Sigma_{h_s}^{-1} z - \mathbb{I}_s \mathbf{y}\|_{\Sigma_s}^2 + \|z\|_{\Sigma_z}^2 \quad (\text{A.25})$$

The gradient of  $C(\mathbf{z})$  is computed as:

$$\begin{aligned}\frac{\partial C(\mathbf{z})}{2\partial \mathbf{z}} &= \Sigma_t^{-1} (\Sigma_t \Sigma_{\mathbf{h}_t}^{-1} \mathbf{z} - \mathbb{I}_t \mathbf{x}) + \Sigma_s^{-1} (\Sigma_s \Sigma_{\mathbf{h}_s}^{-1} \mathbf{z} - \mathbb{I}_s \mathbf{y}) + \Sigma_z^{-1} \mathbf{z} \\ &= (\Sigma_{\mathbf{h}_t}^{-1} + \Sigma_{\mathbf{h}_s}^{-1} + \Sigma_z^{-1}) \mathbf{z} - (\Sigma_t^{-1} \mathbb{I}_t \mathbf{x} + \Sigma_s^{-1} \mathbb{I}_s \mathbf{y})\end{aligned}\quad (\text{A.26})$$

The estimation of optimized  $\hat{\mathbf{z}}$  is obtained when setting this gradient to zeros:

$$\hat{\mathbf{z}} = (\Sigma_{\mathbf{h}_t}^{-1} + \Sigma_{\mathbf{h}_s}^{-1} + \Sigma_z^{-1})^{-1} (\Sigma_t^{-1} \mathbb{I}_t \mathbf{x} + \Sigma_s^{-1} \mathbb{I}_s \mathbf{y}) \quad (\text{A.27})$$

## A.2 Derivation of generalized Millman formula

Let denote  $\{\hat{\mathbf{z}}_1, \dots, \hat{\mathbf{z}}_K\}$  be  $K$  estimates of an unknown random vector  $\mathbf{z} \in R^n$ . The objective of the fusion model is to find the best linear estimate via a set of coefficient matrices  $C_i$  ( $i=1, \dots, K$ ):

$$\hat{\mathbf{z}} = \sum_{i=1}^K C_i \hat{\mathbf{z}}_i \quad s.t \quad \sum_{i=1}^K C_i = I_n \quad (\text{A.28})$$

where the weighting matrices are estimated as:

$$\{C_i\}_{i=1, \dots, K} = \underset{C_i}{\operatorname{argmin}} \left\| \mathbf{z} - \sum_{i=1}^K C_i \hat{\mathbf{z}}_i \right\|_2^2 \quad s.t \quad \sum_{i=1}^K C_i = I_n \quad (\text{A.29})$$

Define the loss function  $J$  as the error expectation:

$$J = E \left\{ \left\| \mathbf{z} - \sum_{i=1}^K C_i \hat{\mathbf{z}}_i \right\|_2^2 \right\} \quad (\text{A.30})$$

Define also the total error covariance:

$$\Sigma \triangleq \operatorname{cov} \{ \mathbf{z} - \hat{\mathbf{z}} \} \quad (\text{A.31})$$

$$= E \left[ \left( \mathbf{z} - \sum_{i=1}^K C_i \hat{\mathbf{z}}_i \right) \left( \mathbf{z} - \sum_{j=1}^K C_j \hat{\mathbf{z}}_j \right)^T \right] \quad (\text{A.32})$$

Since  $\sum_{i=1}^K C_i = I_n$ , the total error covariance is given as:

$$\Sigma = E \left[ \sum_{i=1}^K C_i (\mathbf{z} - \hat{\mathbf{z}}_i) \sum_{j=1}^K C_j (\mathbf{z} - \hat{\mathbf{z}}_j)^T \right] \quad (\text{A.33})$$

Define the local error term as  $\tilde{z}_i \triangleq z - \hat{z}_i$  and the local error covariance as  $\Sigma_{ij} \triangleq \text{cov}\{\tilde{z}_i, \tilde{z}_j\}$ , the total error covariance becomes:

$$\Sigma = E \left[ \sum_{i,j=1}^K C_i \tilde{z}_i (C_j \tilde{z}_j)^T \right] \quad (\text{A.34})$$

$$= \sum_{i,j=1}^K C_i \Sigma_{ij} C_j^T \quad (\text{A.35})$$

The loss function  $J$  is rewritten as:

$$J = E \left\{ \left\| z - \sum_{i=1}^K C_i \hat{z}_i \right\|_2^2 \right\} \quad (\text{A.36})$$

$$= \text{tr} \left\{ \sum_{i,j=1}^K C_i \Sigma_{ij} C_j^T \right\} \quad (\text{A.37})$$

Substituting  $C_K = I_n - C_1 - C_2 - \dots - C_{K-1}$  into above equation, one gets:

$$\begin{aligned} J = & \text{tr} \left\{ \sum_{i,j=1}^{K-1} C_i \Sigma_{ij} C_j^T + \sum_{i,j=1}^{K-1} (C_i \Sigma_{iK} + \Sigma_{Ki} C_i^T) - \sum_{i,j=1}^{K-1} (C_i \Sigma_{iK} C_j^T + C_j \Sigma_{Ki} C_i^T) \right\} \\ & + \text{tr} \left\{ \Sigma_{KK} - \left( \sum_{i,j=1}^{K-1} C_i \right) \Sigma_{KK} - \Sigma_{KK} \left( \sum_{i,j=1}^{K-1} C_j^T \right) + \sum_{i,j=1}^{K-1} C_i \Sigma_{KK} C_j^T \right\} \end{aligned} \quad (\text{A.38})$$

Using the symmetric property of covariance matrices, i.e.  $\Sigma_{ij} = \Sigma_{ji}^T$  and  $\Sigma_{ii} = \Sigma_{ii}^T$ , one gets:

$$\frac{\partial}{\partial C_i} (\text{tr}\{C_i \Sigma\}) = \Sigma^T \quad (\text{A.39})$$

$$\frac{\partial}{\partial C_i} (\text{tr}\{\Sigma C_i^T\}) = \Sigma \quad (\text{A.40})$$

$$\frac{\partial}{\partial C_i} (\text{tr}\{C_i \Sigma C_i^T\}) = C_i (\Sigma + \Sigma^T) \quad (\text{A.41})$$

Taking partial derivative  $\partial J / \partial C_i$  and set to zero, one gets:

$$\sum_{i,j=1}^{K-1} C_i (\Sigma_{ij} - \Sigma_{iK}) + C_K (\Sigma_{Kj} - \Sigma_{KK}) = 0 \quad \text{s.t.} \quad \sum_{i,j=1}^K C_i = I_n, \quad j = 1, \dots, K-1 \quad (\text{A.42})$$



The final GMF of fusing  $z$  from  $\{\hat{z}_1, \dots, \hat{z}_K\}$  is:

$$\hat{z} = \sum_{i=1}^K C_i \hat{z}_i \quad s.t. \quad \begin{cases} \sum_{i=1}^K C_i = I_n \\ \sum_{i,j=1}^{K-1} C_i (\Sigma_{ij} - \Sigma_{iK}) + C_K (\Sigma_{Kj} - \Sigma_{KK}) = 0, \quad j = 1, \dots, K-1 \end{cases} \quad (\text{A.43})$$

with a total error covariance of:

$$\Sigma = \sum_{i,j=1}^K C_i \Sigma_{ij} C_j^T \quad (\text{A.44})$$

### Bar-Shalom and Campo (1986)

In the case of fusing two estimates ( $K = 2$ ), the GMF becomes Bar-Shalom-Campo formula:

$$\hat{z} = C_1 \hat{z}_1 + C_2 \hat{z}_2 \quad (\text{A.45})$$

Weighting matrices  $C_1$  and  $C_2$  satisfy:

$$\begin{cases} C_1 + C_2 = I_n \\ C_1 (\Sigma_{11} - \Sigma_{12}) + C_2 (\Sigma_{21} - \Sigma_{22}) = 0 \\ C_1 (\Sigma_{12} - \Sigma_{12}) + C_2 (\Sigma_{22} - \Sigma_{22}) = 0 \end{cases} \quad (\text{A.46})$$

which lead to:

$$\begin{cases} C_1 = (\Sigma_{22} - \Sigma_{21})(\Sigma_{11} + \Sigma_{22} - \Sigma_{12} - \Sigma_{21})^{-1} \\ C_2 = (\Sigma_{11} - \Sigma_{12})(\Sigma_{11} + \Sigma_{22} - \Sigma_{12} - \Sigma_{21})^{-1} \end{cases} \quad (\text{A.47})$$

In case  $\tilde{z}_1$  and  $\tilde{z}_2$  are uncorrelated, i.e.  $\Sigma_{12} = \Sigma_{21} = 0$ , the fusion formula is known as the Millman's formula:

$$\hat{z} = \Sigma_{22} (\Sigma_{11} + \Sigma_{22})^{-1} \hat{z}_1 + \Sigma_{11} (\Sigma_{11} + \Sigma_{22})^{-1} \hat{z}_2 \quad (\text{A.48})$$

This formula is identical to the Bayesian fusion model assuming a non-informative prior on  $z$ .



# Contents

<b>Abstract</b>	<b>ix</b>
<b>Acknowledgements</b>	<b>xi</b>
<b>Abbreviations</b>	<b>xiii</b>
<b>Latin symbols</b>	<b>xv</b>
<b>Math symbols</b>	<b>xix</b>
<b>Table of contents</b>	<b>xxi</b>
<b>List of Tables</b>	<b>xxiii</b>
<b>List of Figures</b>	<b>xxvii</b>
<b>Introduction</b>	<b>1</b>
Organization and contributions . . . . .	2
<b>1 Problem definition</b>	<b>5</b>
1.1 Background . . . . .	5
1.1.1 Turbulence . . . . .	5
1.1.2 Importance of small-scale turbulence . . . . .	7
1.2 Limitations of turbulence research tools . . . . .	8
1.2.1 Numerical simulations . . . . .	8
1.2.2 Experimental techniques . . . . .	9
1.3 Problem definition . . . . .	11
1.3.1 Estimating high-resolution turbulent fields from low-resolution measurements . . . . .	11
1.3.2 Fusion of large-scale measurements . . . . .	13
1.4 Datasets . . . . .	15
1.4.1 DNS data of channel flow . . . . .	16

1.4.2	DNS data of isotropic turbulence . . . . .	18
1.5	Concluding remarks . . . . .	21
<b>2</b>	<b>Velocity reconstruction using regression</b>	<b>23</b>
2.1	Regression in turbulence studies . . . . .	23
2.2	Ordinary least squares (OLS) . . . . .	24
2.3	Regularized linear regression . . . . .	25
2.3.1	Ridge Regression (RR): L2 penalty . . . . .	26
2.3.2	LASSO: L1 penalty . . . . .	27
2.4	Nonlinear regression . . . . .	28
2.4.1	Feature mapping . . . . .	28
2.4.2	Kernel ridge regression . . . . .	29
2.5	A framework to select model and parameters . . . . .	30
2.5.1	Bias-variance trade-off . . . . .	31
2.5.2	Cross-validation . . . . .	33
2.6	Regression models for reconstruction of isotropic turbulence . . . . .	34
2.6.1	Regularization parameter and shrinkage effect . . . . .	35
2.6.2	Optimizing regularization parameters and model complexity via ten-fold cross-validation . . . . .	36
2.7	Concluding remarks . . . . .	38
<b>3</b>	<b>Learning coupled dictionaries</b>	<b>41</b>
3.1	From bases to dictionaries . . . . .	41
3.1.1	Predefined dictionaries . . . . .	42
3.1.2	Adaptive dictionaries . . . . .	42
3.2	Proper Orthogonal Decomposition (POD) as a representation of turbulent fields . . . . .	43
3.3	Dictionary learning as a new representation . . . . .	45
3.3.1	Redundant dictionary and sparse representation . . . . .	45
3.3.2	Dictionary learning methods . . . . .	46
3.4	Learning coupled dictionaries . . . . .	49
3.4.1	Patch-based approach . . . . .	49
3.4.2	The approach . . . . .	50
3.4.3	Joint learning methods . . . . .	51
3.4.4	Reconstruction using learned dictionaries . . . . .	54
3.5	Dictionary learning for isotropic turbulence fields . . . . .	56
3.5.1	On the choice of parameters . . . . .	56
3.5.2	Efficiency of representations: a comparative study . . . . .	57
3.5.3	Reconstruction of high resolution fields- subsampling cases . . . . .	60
3.5.4	Reconstruction of high resolution fields- the downsampling case . . . . .	62
3.6	Concluding remarks . . . . .	66

<b>4</b>	<b>Non-local similarity-based propagation model</b>	<b>69</b>
4.1	Non-local means . . . . .	69
4.1.1	Non-local means filter for denoising . . . . .	70
4.1.2	Generalized non-local means for super-resolution . . . . .	71
4.2	Similarity-based model to propagate small scales . . . . .	72
4.2.1	Greedy propagation of small scales from key planes . . . . .	72
4.2.2	Non-greedy propagation of small scales . . . . .	74
4.3	Applying the propagation model to isotropic turbulence . . . . .	75
4.3.1	The choice of model parameters . . . . .	75
4.3.2	Comparing to spline interpolation . . . . .	78
4.3.3	Model performances in various configurations . . . . .	80
4.4	Concluding remarks . . . . .	82
<b>5</b>	<b>Bayesian fusion model</b>	<b>85</b>
5.1	A probabilistic model . . . . .	85
5.1.1	Bayes' rule . . . . .	86
5.1.2	Bayesian estimates . . . . .	86
5.2	Bayesian fusion model . . . . .	87
5.2.1	The model . . . . .	88
5.2.2	MAP estimation . . . . .	89
5.3	Equivalent models . . . . .	91
5.3.1	Linear Gaussian model . . . . .	91
5.3.2	Generalized Millman formula . . . . .	92
5.4	Model simplifications . . . . .	93
5.4.1	Simplified covariance matrices . . . . .	93
5.4.2	Statistical parameters estimation . . . . .	94
5.5	Applying the Bayesian fusion model to DNS data of isotropic turbulence	95
5.5.1	Impact of subsampling ratios . . . . .	96
5.5.2	Large and small scales reconstruction . . . . .	98
5.6	Concluding remarks . . . . .	99
<b>6</b>	<b>Comparisons of model performances on various datasets</b>	<b>101</b>
6.1	Reconstruction results on DNS data of isotropic turbulence . . . . .	102
6.2	Reconstruction results on DNS data of channel flow . . . . .	109
6.3	Concluding remarks . . . . .	113
	<b>Conclusions and perspectives</b>	<b>117</b>
	Suggestions for future works . . . . .	119
	<b>Bibliography</b>	<b>123</b>

---

<b>A Bayesian framework</b>	<b>131</b>
A.1 Derivation of Linear Gaussian models . . . . .	131
A.1.1 Bayes' theorem for Gaussian variables . . . . .	131
A.1.2 The model . . . . .	133
A.2 Derivation of generalized Millman formula . . . . .	135
<b>Contents</b>	<b>139</b>



## RECONSTRUCTION OF FINELY RESOLVED VELOCITY FIELDS IN TURBULENT FLOWS FROM LOW RESOLUTION MEASUREMENTS

This work lies in between the research domains of turbulence and image processing. The main objectives are to propose new methodologies to reconstruct small-scale turbulence from measurements at large-scale only. One contribution of this work is a review of existing methods. We also propose new models inspired from recent works in image processing to adapt them to the context of turbulence. We address two different problems. The first problem is to find an empirical mapping function between large and small scales for which regression models are a common approach. We also introduce the use of “dictionary learning” to train coupled representations of large and small scales for reconstruction. The second problem is to reconstruct small-scale information via the fusion of complementary measurements. The non-local means propagation model exploits the similarity of structures in the flow, while the Bayesian fusion model estimates the most probable fields given the measurements thanks to a maximum a posteriori estimate. All methods are validated and analyzed using numerical databases where fully resolved velocity fields are available. Performances of the proposed approaches are also characterized for various configurations. These results can be considered under the co-conception design framework where different experimental setups are designed to maximize the level of useful information after post-processing.

**Keywords:** reconstruction, turbulence, dictionary learning, non-local means, bayesian fusion, regression, small scales

---

## RECONSTRUCTION FINE DE CHAMPS DE VITESSES D'UN ÉCOULEMENT TURBULENT À PARTIR DE MESURES FAIBLEMENTS RÉVOLUES

Ce travail est à la jonction de deux domaines de recherche que sont la turbulence et le traitement d'image. L'objectif principal est de proposer de nouvelles méthodologies pour reconstruire les petites échelles de la turbulence à partir de mesures grande échelle. Ce travail revisite des méthodes conventionnelles et propose de nouveaux modèles basés sur les travaux récents en traitement d'image pour les adapter à une problématique de turbulence. Le premier problème consiste à trouver une fonction de correspondance empirique entre les grandes et les petites échelles, une approche classique pour les modèles de type regression. Nous introduisons également une méthode appelée “apprentissage de dictionnaire” pour laquelle une représentation couplée des grandes et des petites échelles est déduite par apprentissage statistique. Le deuxième problème est de reconstruire les informations à petites échelles par fusion de plusieurs mesures complémentaires. Le modèle de type “propagation de la moyenne non-locale” exploite la similarité des structures de l'écoulement alors que les modèles bayésiens de fusion proposent d'estimer le champ le plus probable en fonction d'informations données, on parle d'estimateur maximum a posteriori. Les performances des différentes approches sont validées et analysées sur des bases de données numériques pour lesquelles les informations sont disponibles à toutes les échelles. Ces résultats peuvent être utilisés dans une approche de type co-conception où il s'agit d'imaginer différents dispositifs expérimentaux définis conjointement avec les traitements numériques pour maximiser la qualité des informations obtenues après traitement.

**Mots clés :** reconstruction, turbulence, apprentissage de dictionnaire, moyenne non-locale, fusion bayésienne, régression, petites échelles

---