Parsimonious Linear Fingerprinting for Time Series

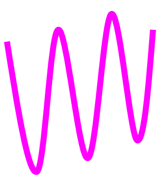
Lei Li, B. Aditya Prakash, Christos Faloutsos
School of Computer Science
Carnegie Mellon University

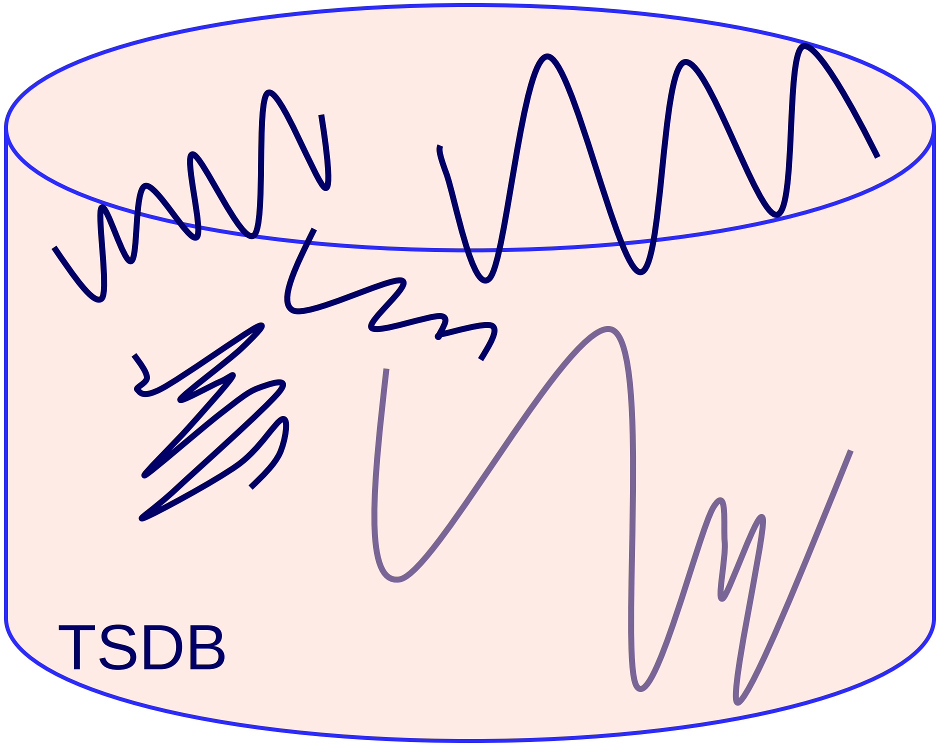
VLDB2010, 36th International Conference on Very Large Data Bases

Motivation

- Answering similarity queries in Time Series Databases

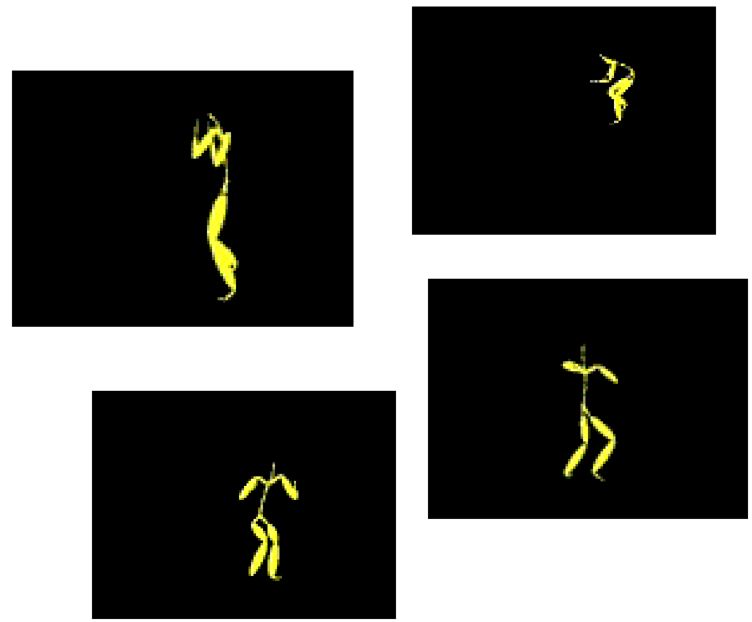
SELECT * FROM ***TSDB***
WHERE data
LIKE

“  ”

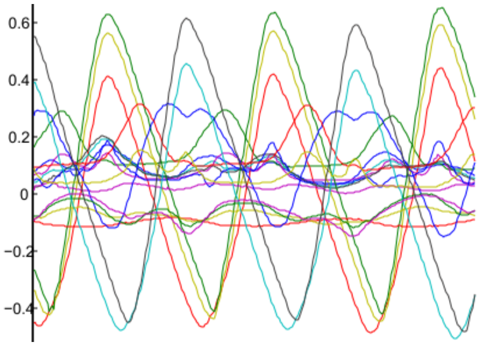


Motivation

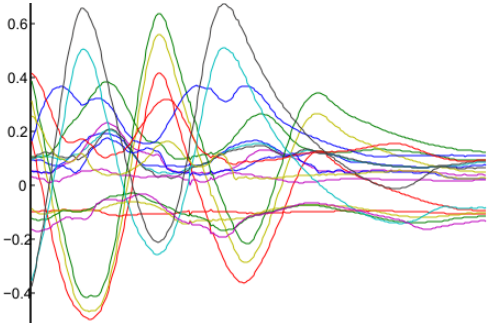
Similar motions ?

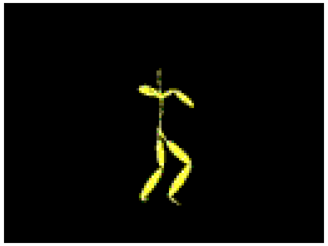


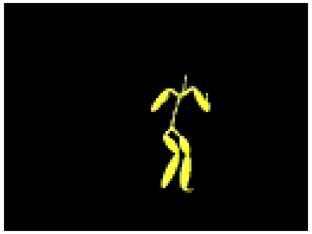
Motivation



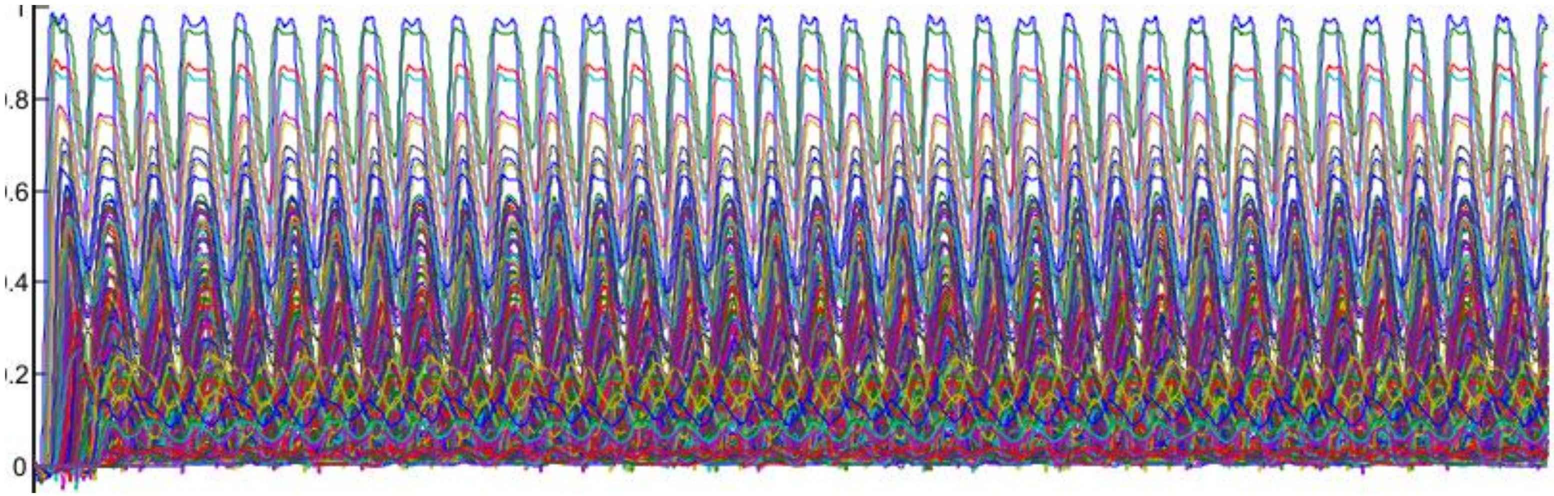
Automatic labeling of human motion sequences



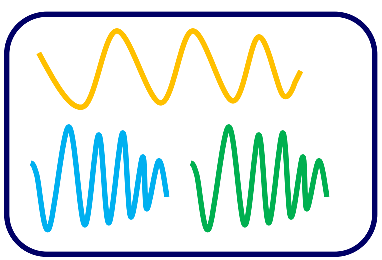




Motivation



Summarization / Compression



Outline

- Motivation
- Proposed Method: Intuition & Example
- Experiments & Results
- PLiF: Insight Details
- Conclusion

Intuition: Goals

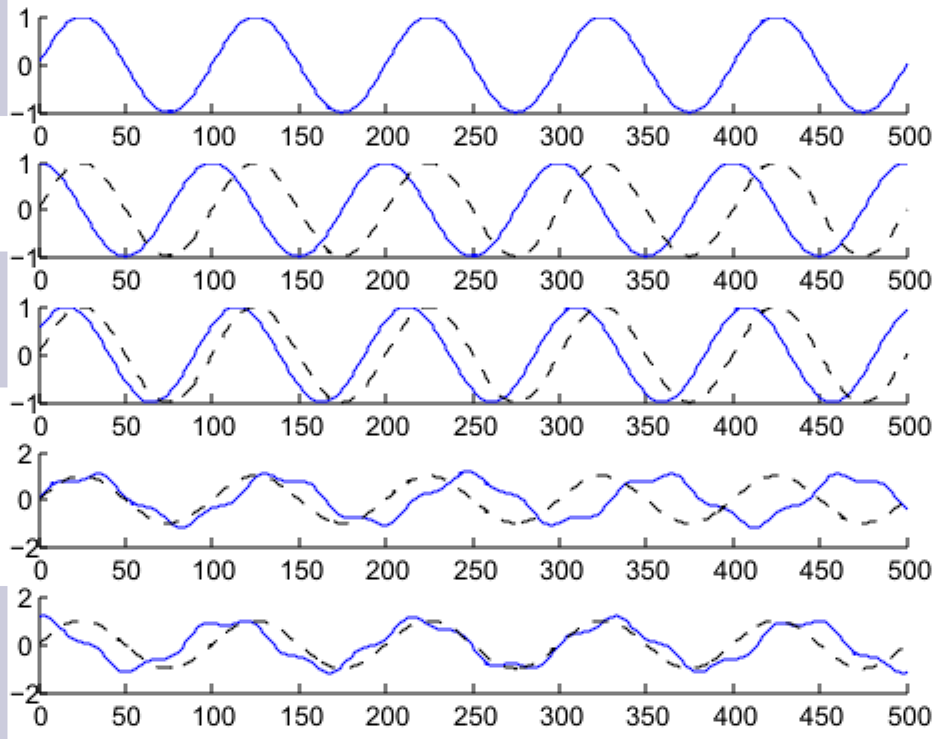
- G1 Good features/similarity function
- G2 Good compression
- G3 Ability to forecast
- G4 Scalability

Intuition: Goals

- **G1** Good features/similarity function
  - (1a) lag independent
  - (1b) frequency proximity
  - (1c) grouping harmonics
- **G2** Good compression
- **G3** Ability to forecast
- **G4** Scalability

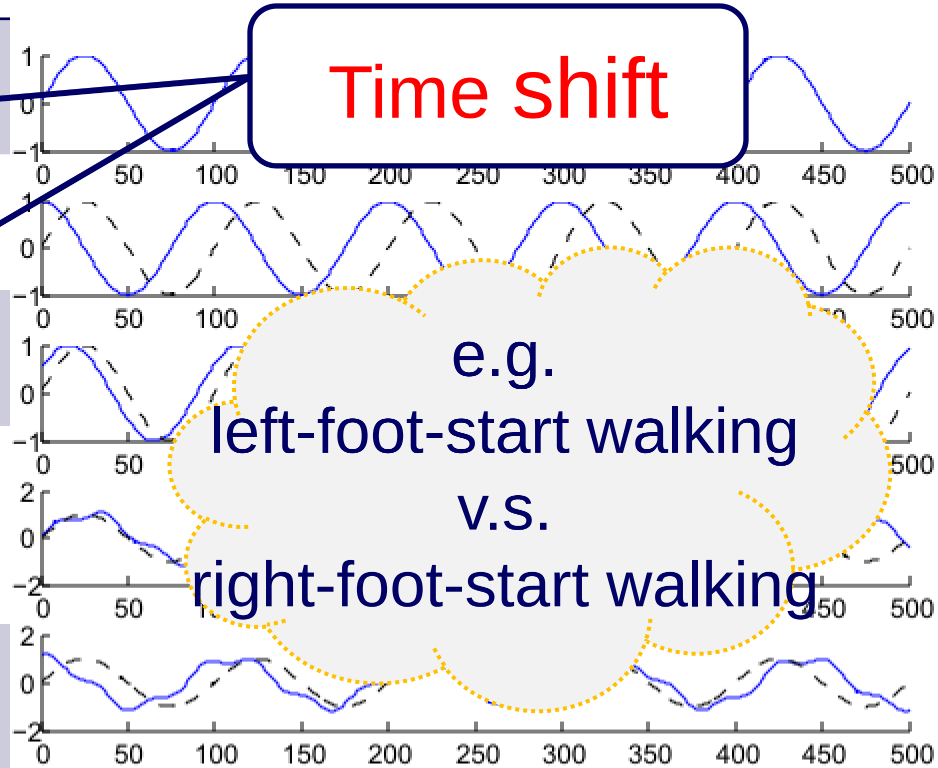
Example: synthetic signals

Equations	
(a)	$\sin(2\pi t/100)$
(b)	$\cos(2\pi t/100)$
(c)	$\sin(2\pi t/98 + \pi/6)$
(d)	$\sin(2\pi t/110) + 0.2\sin(2\pi t/30)$
(e)	$\cos(2\pi t/110) + 0.2\sin(2\pi t/30 + \pi/4)$



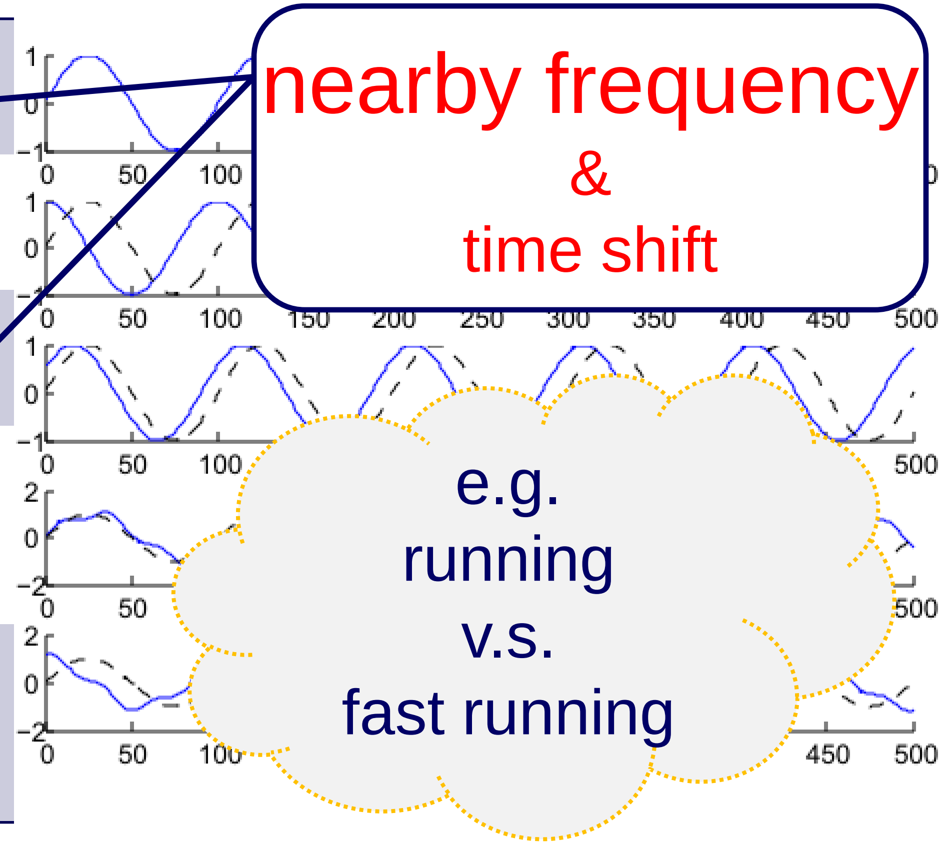
Intuition (1a)

Equations	
(a)	$\sin(2\pi t/100)$
(b)	$\cos(2\pi t/100)$
(c)	$\sin(2\pi t/98 + \pi/6)$
(d)	$\sin(2\pi t/110) + 0.2\sin(2\pi t/30)$
(e)	$\cos(2\pi t/110) + 0.2\sin(2\pi t/30 + \pi/4)$



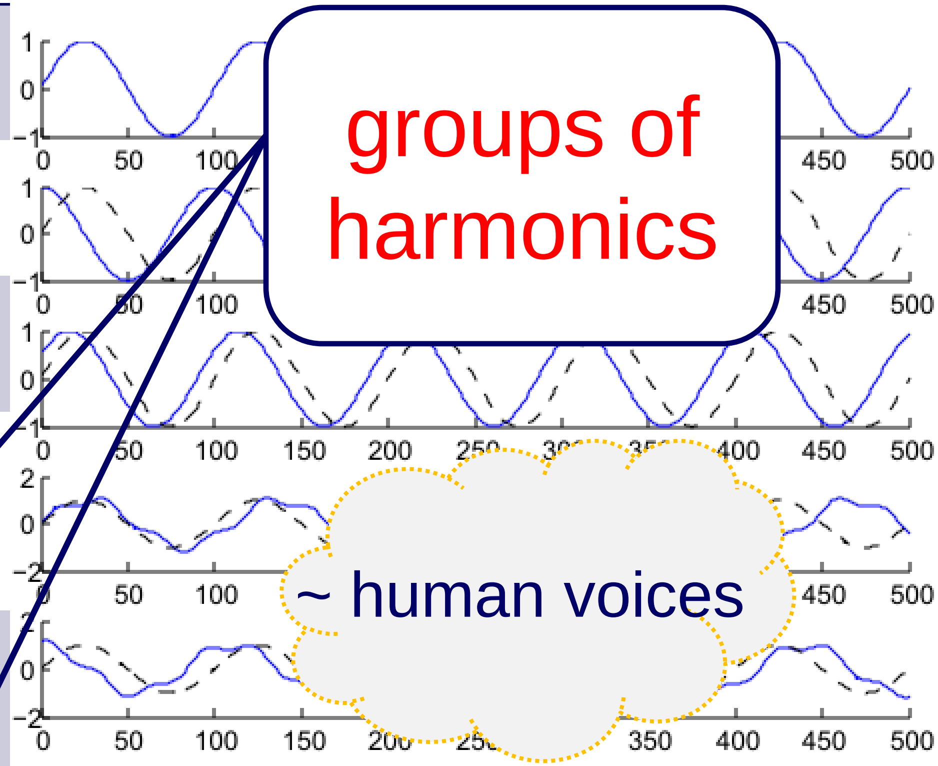
Intuition (1b)

Equations	
(a)	$\sin(2\pi t/100)$
(b)	$\cos(2\pi t/100)$
(c)	$\sin(2\pi t/98 + \pi/6)$
(d)	$\sin(2\pi t/110) + 0.2\sin(2\pi t/30)$
(e)	$\cos(2\pi t/110) + 0.2\sin(2\pi t/30 + \pi/4)$

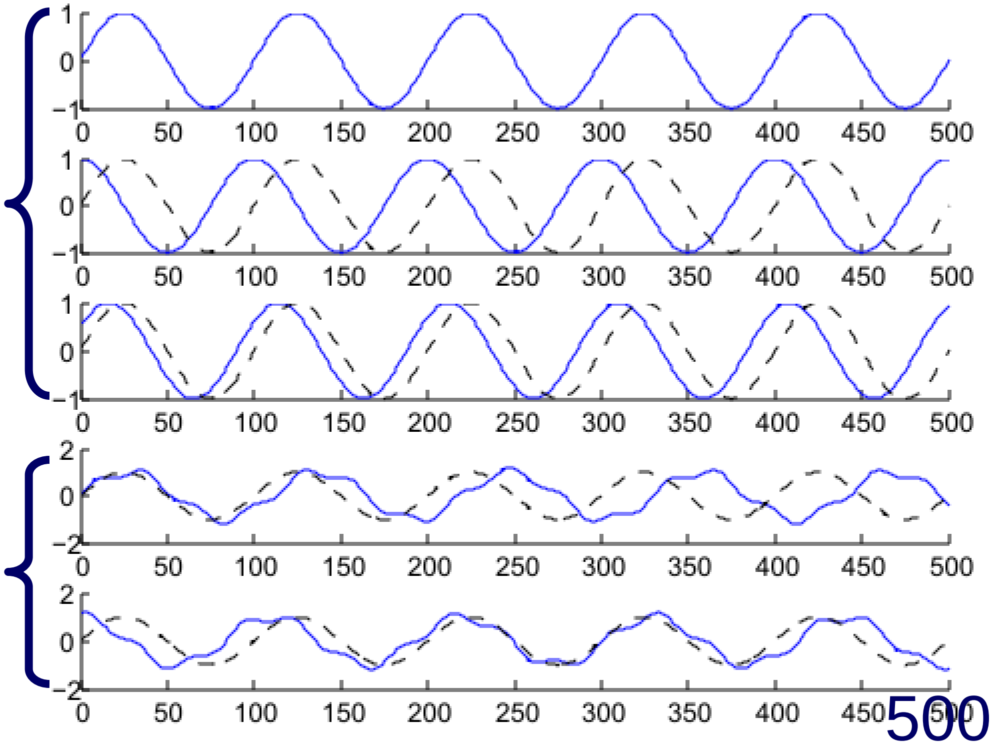


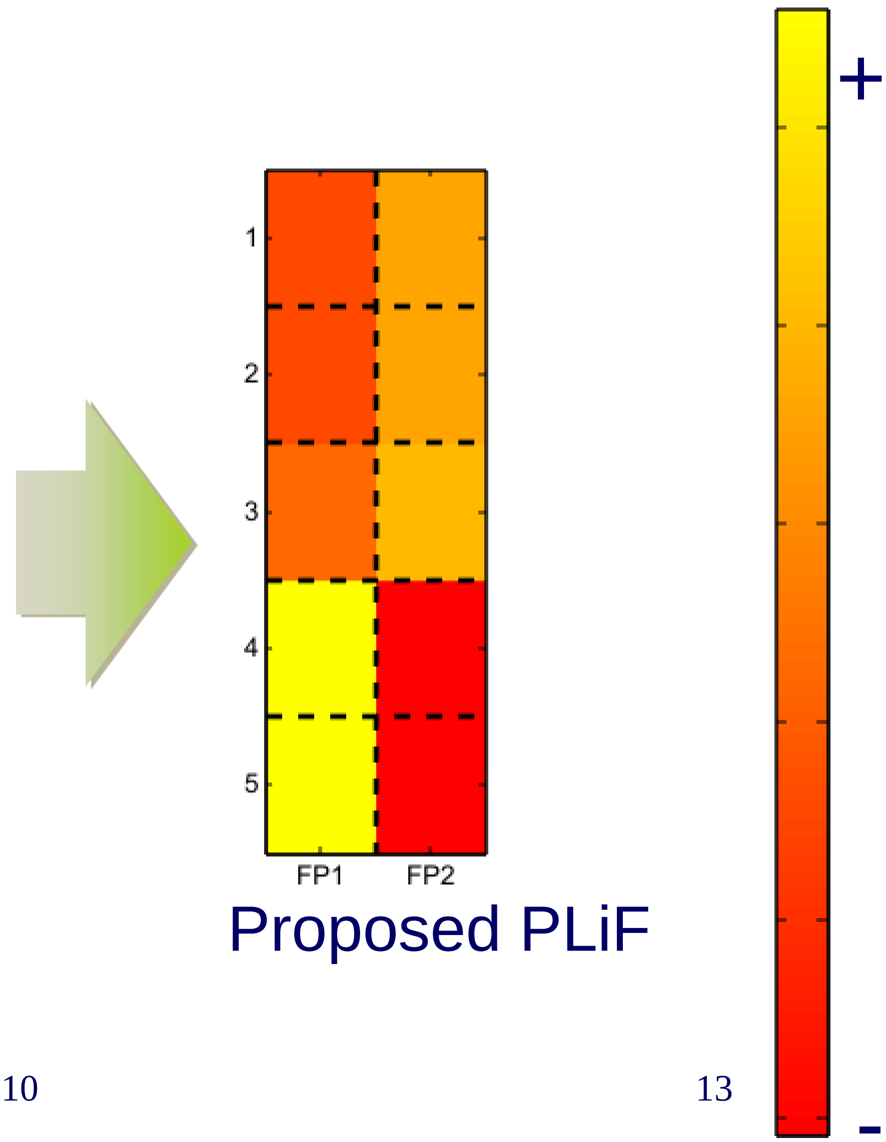
Intuition (1c)

Equations	
(a)	$\sin(2\pi t/100)$
(b)	$\cos(2\pi t/100)$
(c)	$\sin(2\pi t/98 + \pi/6)$
(d)	$\sin(2\pi t/\underline{110}) + 0.2\sin(2\pi t/\underline{30})$
(e)	$\cos(2\pi t/\underline{110}) + 0.2\sin(2\pi t/\underline{30} + \pi/4)$



Q: only two numbers to represent each!

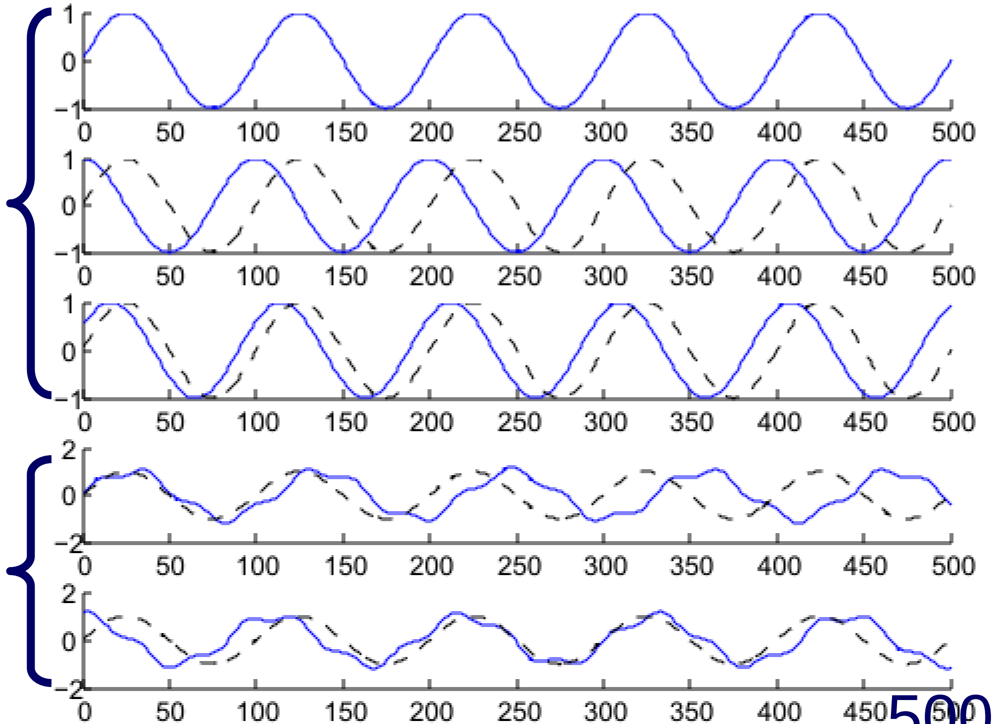


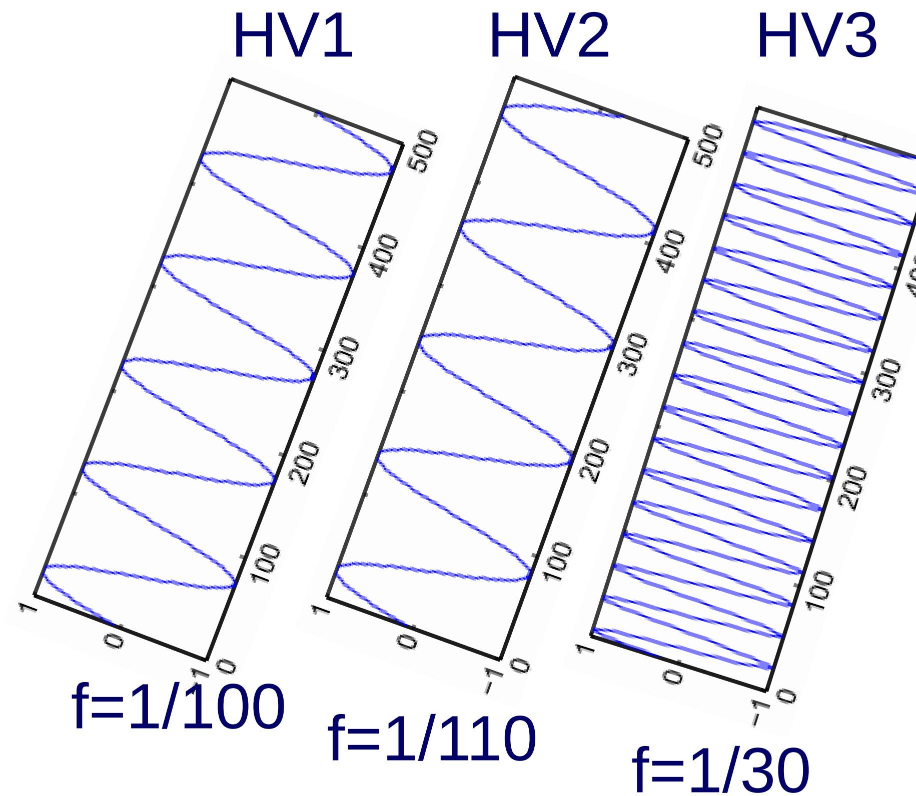


Proposed PLiF

Intuition: how it works

find hidden variable/pattern

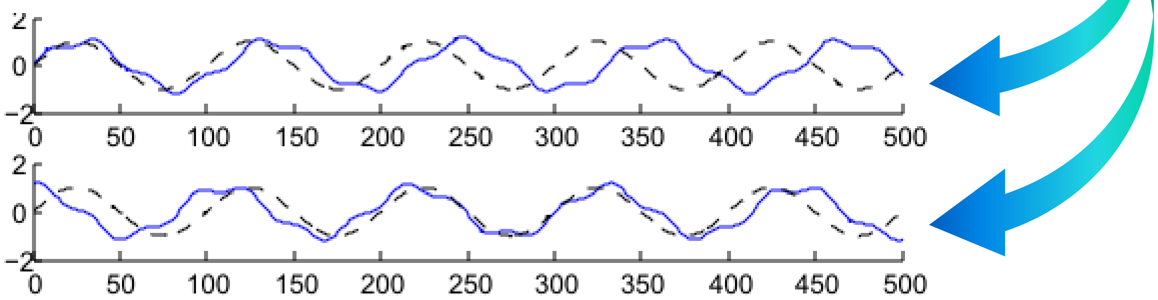




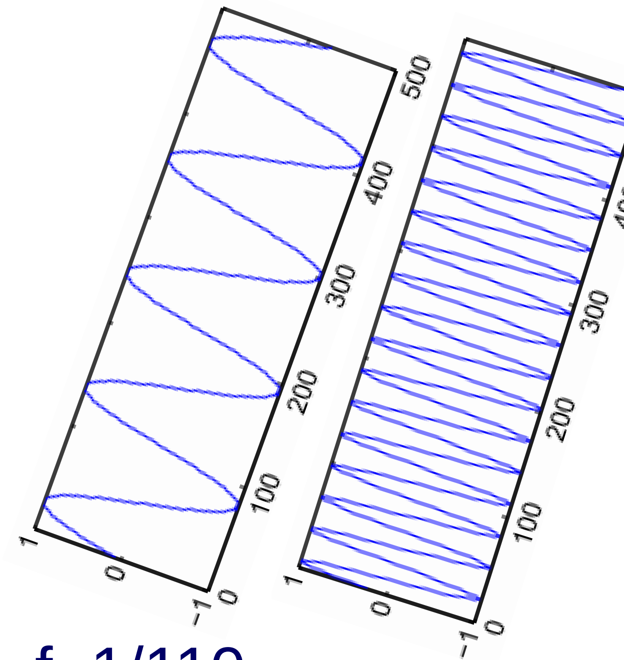
Intuition: how it works

find hidden variable/pattern

HV2’ = HV2 ⊕ HV3

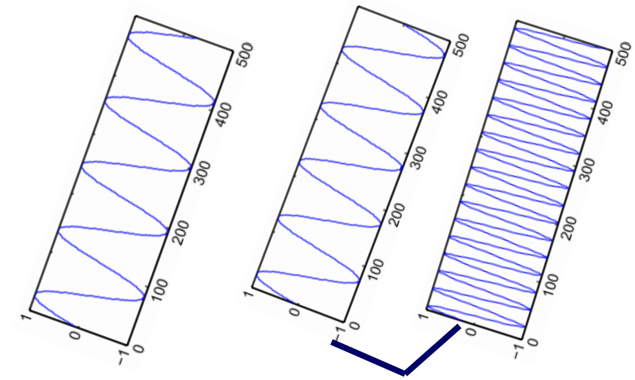


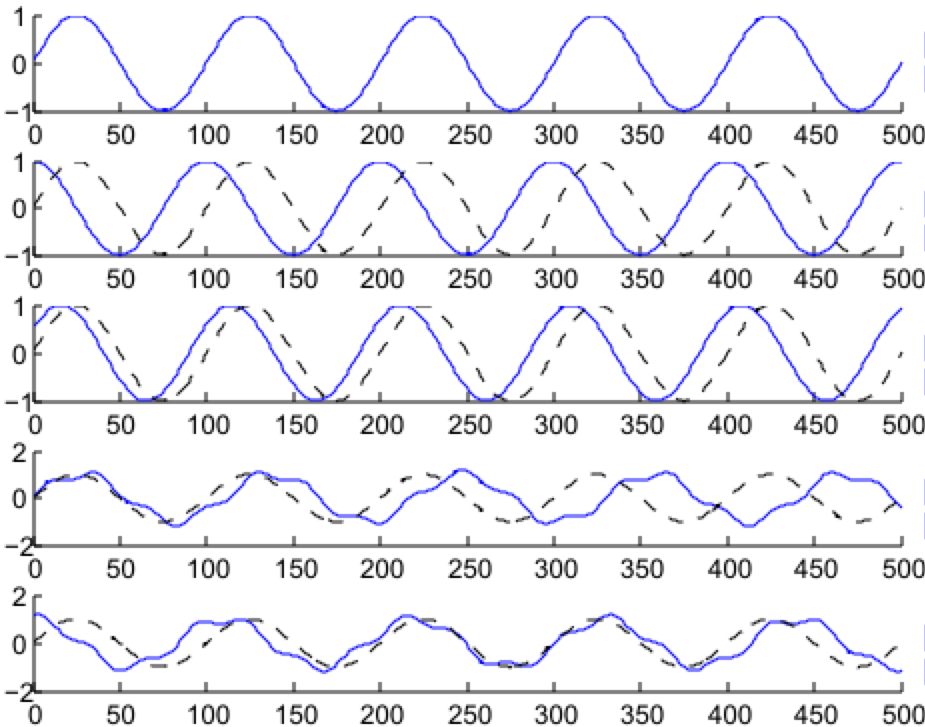
HV2 HV3



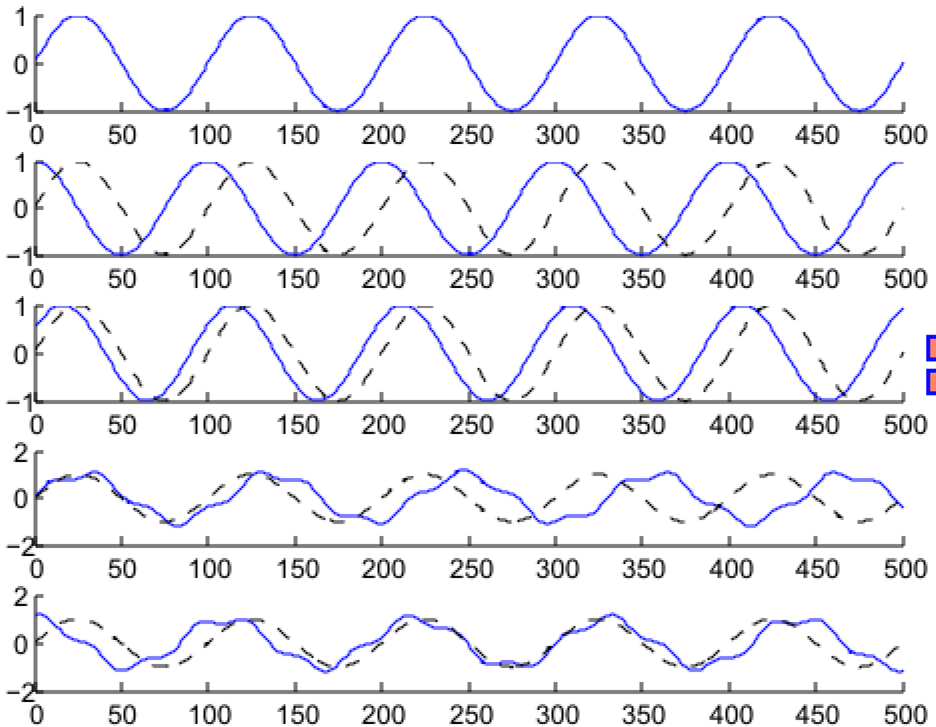
f=1/110 f=1/30

Co-occur





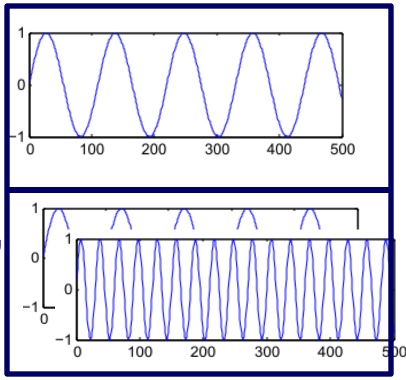
	HV1		HV2’
=	1.0	+	0
=	1.0	+	0
=	0.9	+	0
=	0	+	1.0
=	0	+	1.0



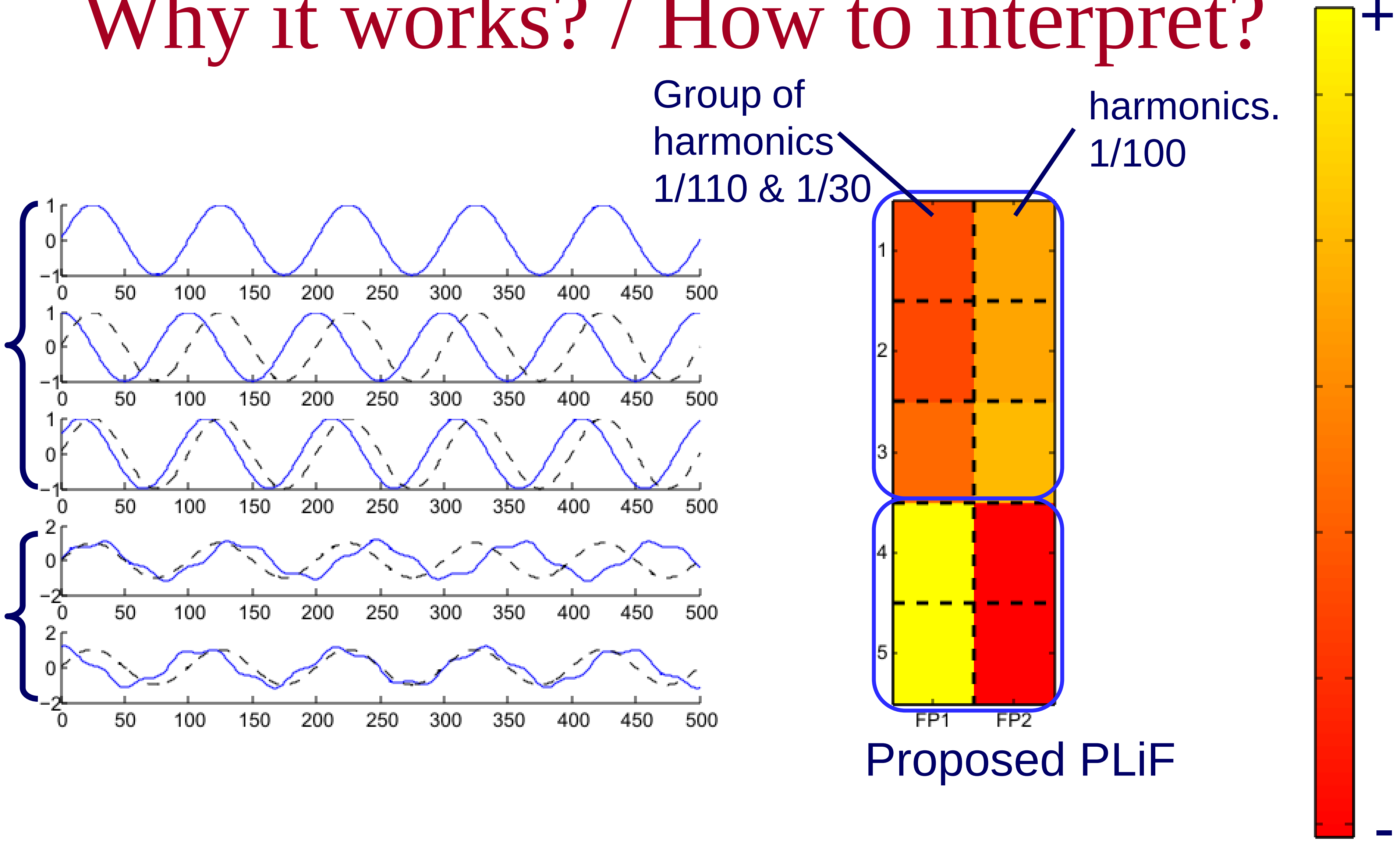
1.0	0
1.0	0
0.9	0
0	1.0
0	1.0

HV1

HV2’

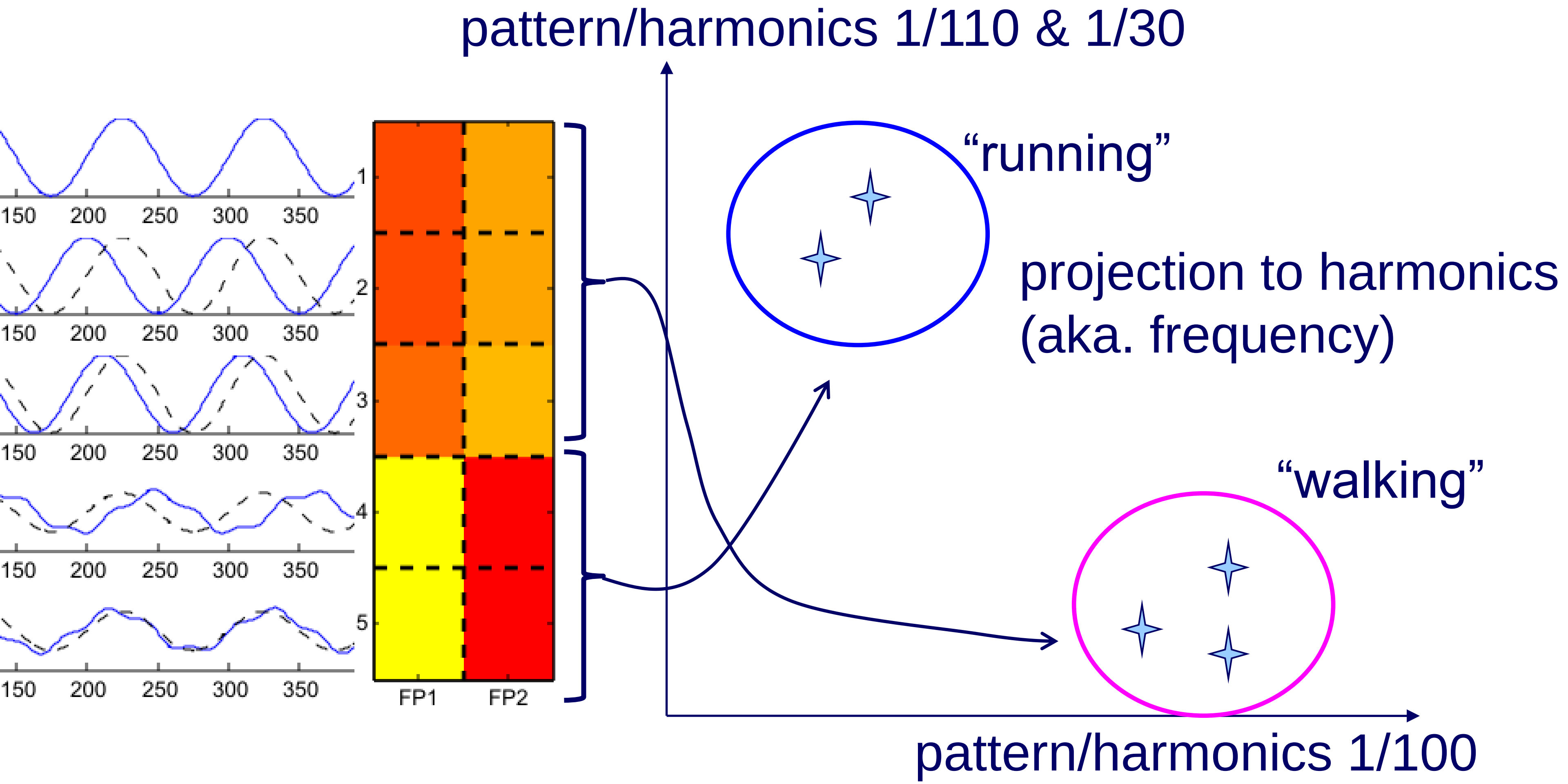


Why it works? / How to interpret?

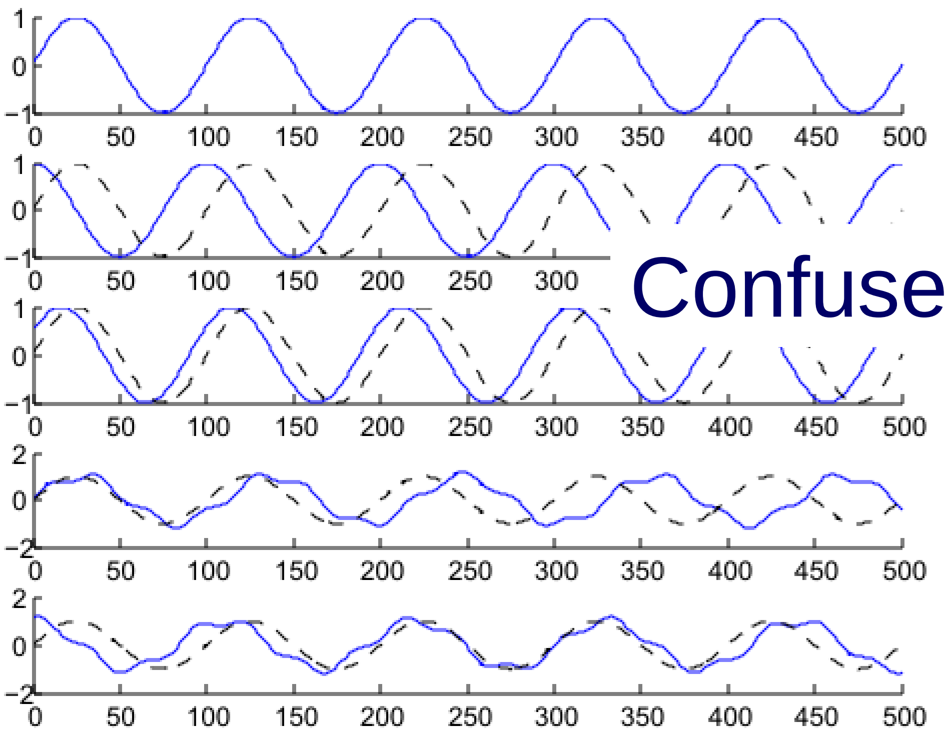


Proposed PLiF

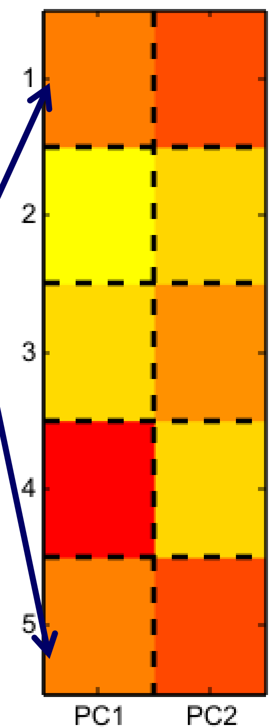
Basic Idea


pattern/harmonics 1/110 & 1/30
"running"
projection to harmonics
(aka. frequency)
"walking"
pattern/harmonics 1/100
150
200
250
300
350
1
2
3
4
5
FP1
FP2

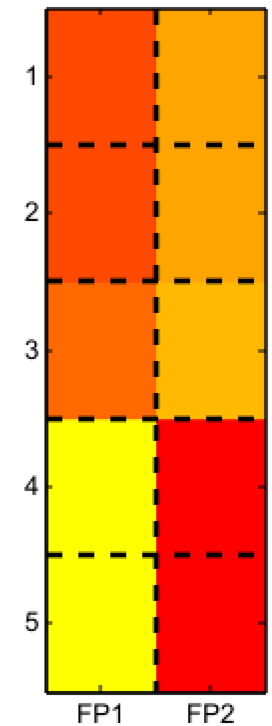
Why not SVD/PCA?



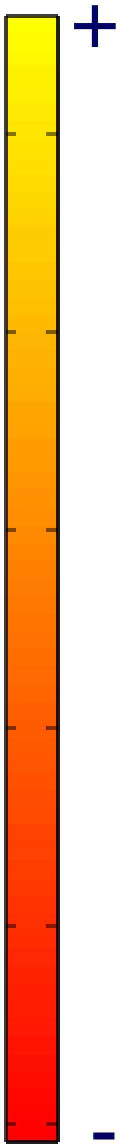
Confused!



PCA



PLiF



no clear grouping

Outline

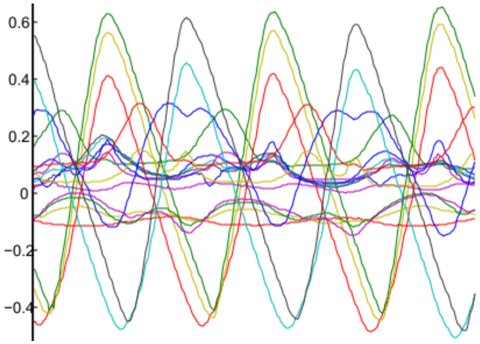
- Motivation
- Proposed Method: Intuition & Example
- Experiments & Results
- PLiF: Insight Details
- Conclusion

Experiment: Goals to Verify

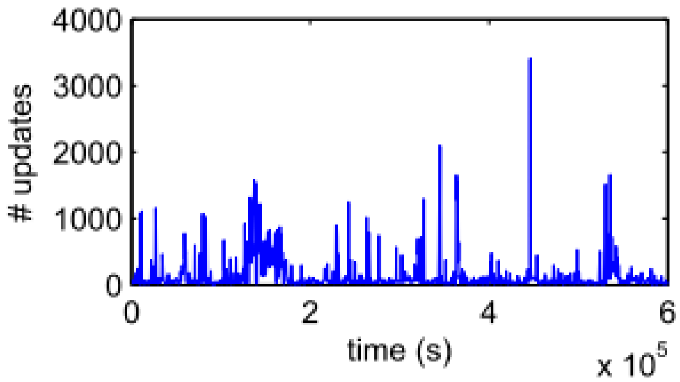
- G1 Good features (low dimensional)
- G2 Good compression
- G3 Ability to forecast
- G4 Scalability

Experiments

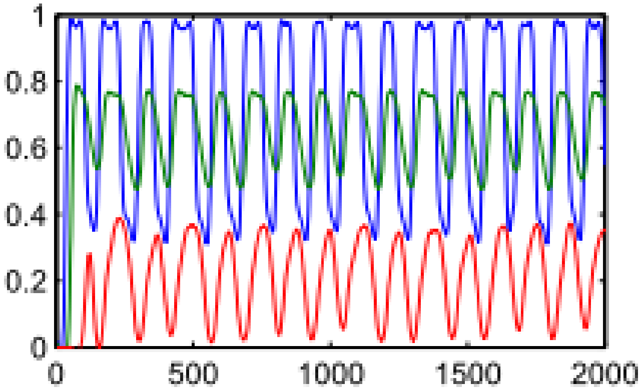
- Datasets:



Mocap 49 * 100-500



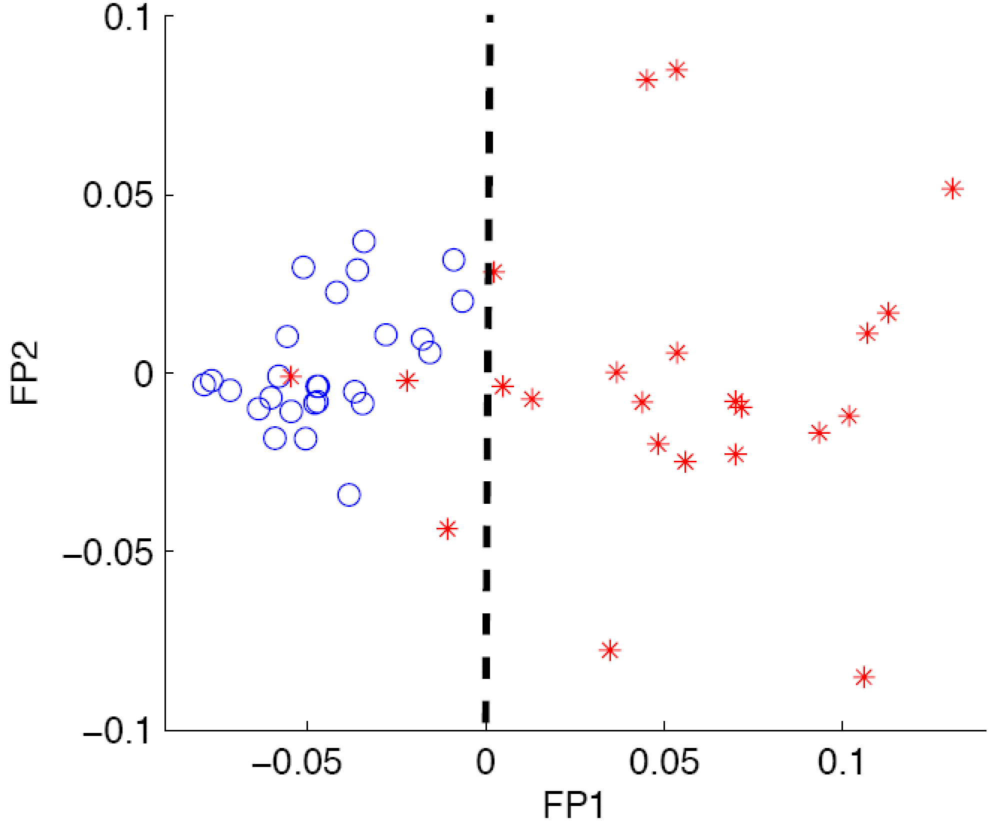
BGP: 10 * 103k



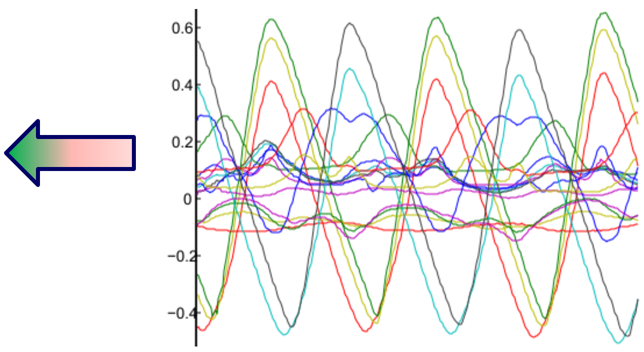
Chlorine:166 * 4k

Result – Visualization

Mocap PLiF first two "fingerprints"

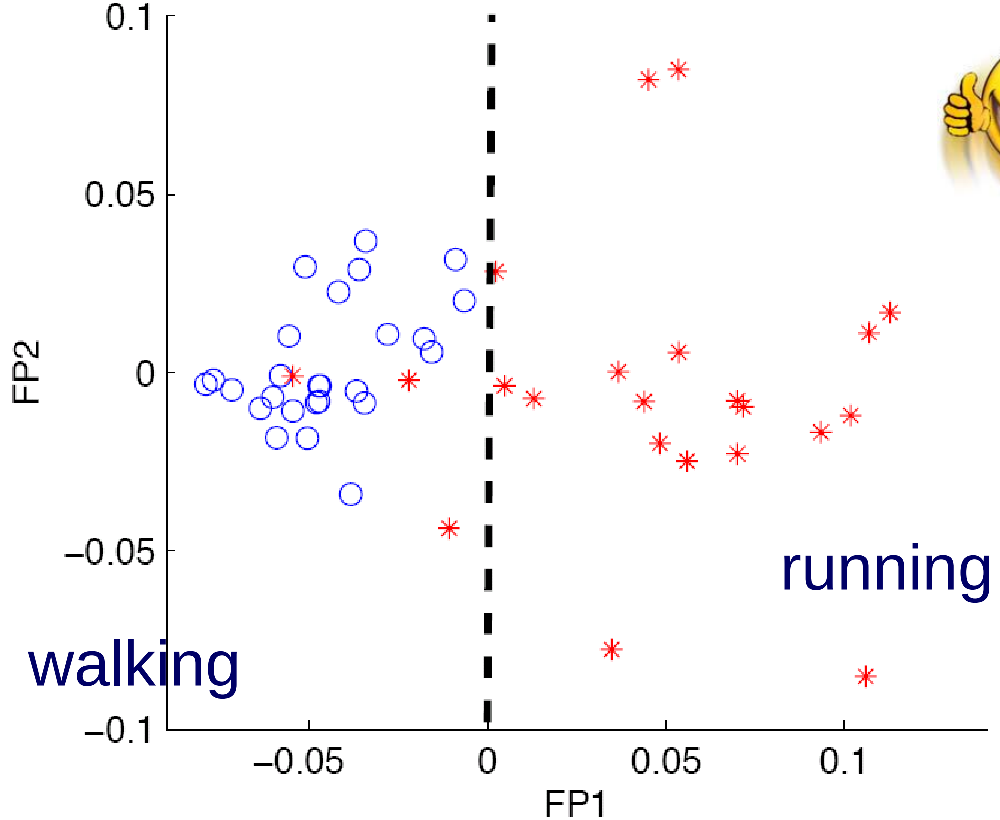


With PLiF,
now able to visualize
very high dimensional
time sequences



Result – Clustering

Mocap PLiF first two "fingerprints"



PLiF + thresholding

Pred.	walk	run
-1	26	3
1	0	20

Accuracy = 46/49

PCA + kmeans

Pred.	walk	run
-1	15	13
1	11	10

Accuracy = 25/49

Result – Clustering

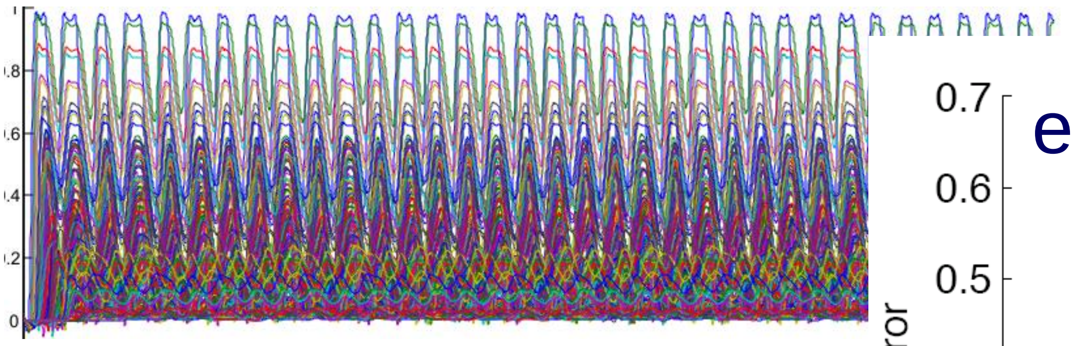
BGP data: PLiF + hierarchical clustering

Intuition: Goals

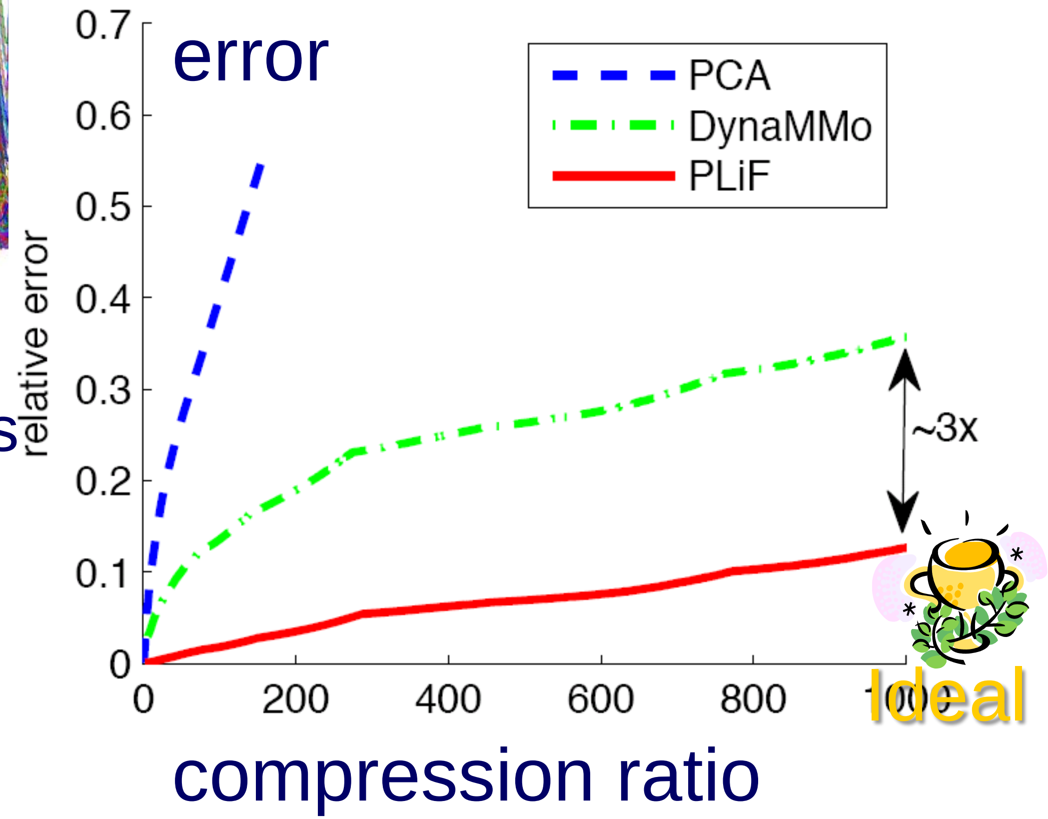
- G1 Good features/similarity function
- G2 Good compression
- G3 Ability to forecast
- G4 Scalability

Result - Compression

Chlorine 166 * 4k

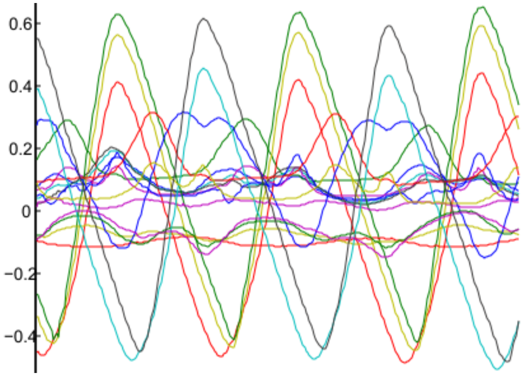


Storing only the PLiF features & sampling of hidden variables



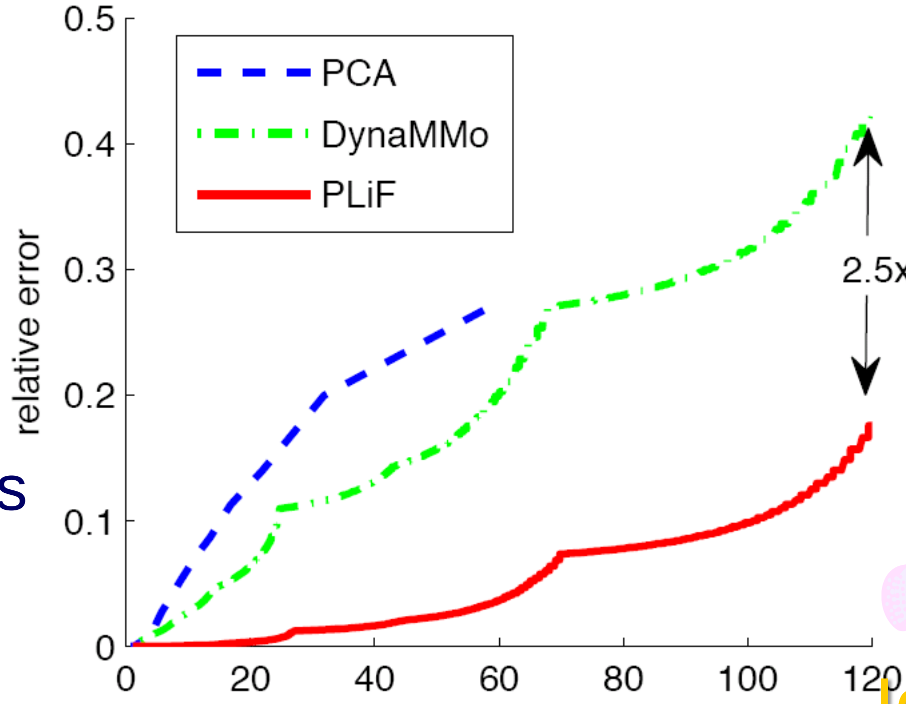
Result - Compression

Mocap: 93 * 300



Storing only the PLiF features & sampling of hidden variables

error



compression ratio

Intuition: Goals

✔ G1 Good features/similarity function

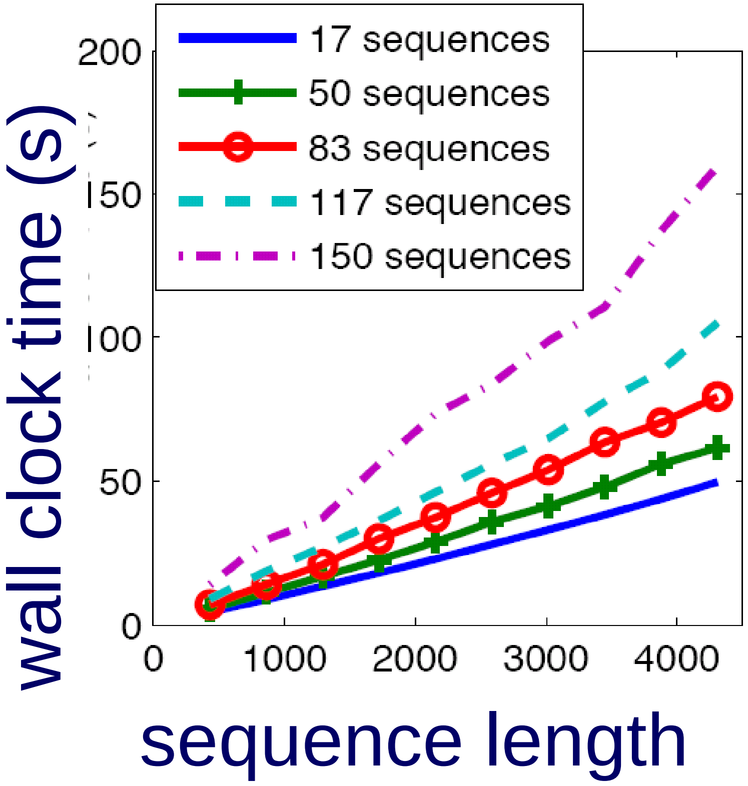
✔ G2 Good compression

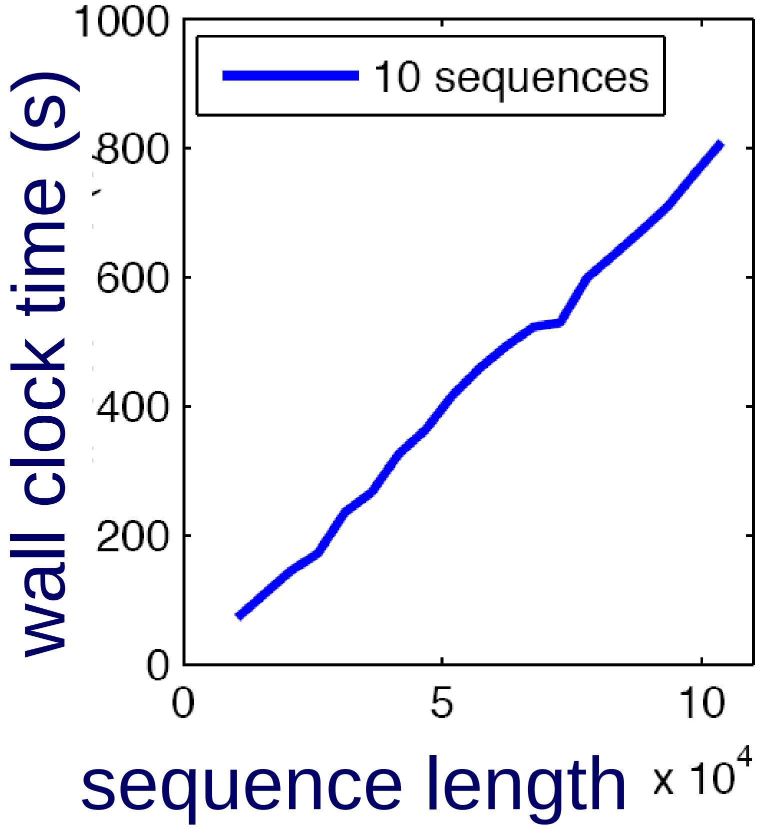
later G3 Ability to forecast

G4 Scalability

Scalability

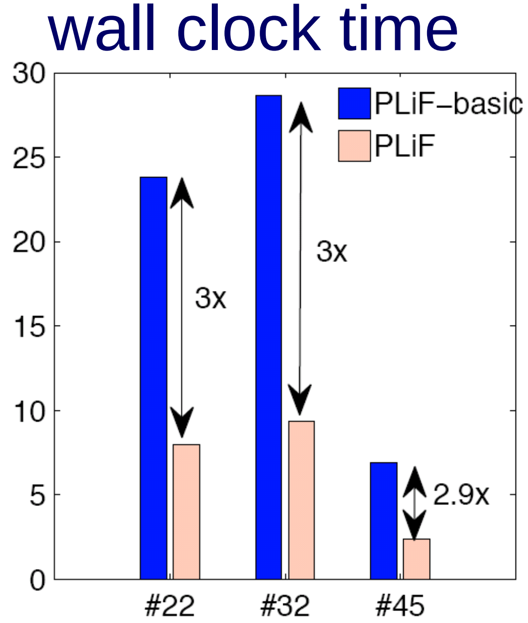
Linear ~ sequence length

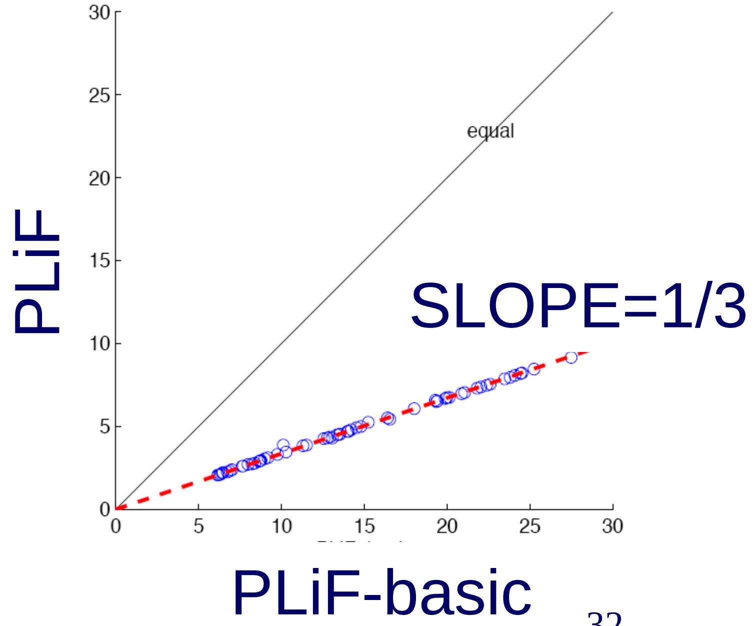




Scalability

- Optimized algorithm
- Details later





Intuition: Goals

- ✓ G1 Good features/similarity function
- ✓ G2 Good compression
- later G3 Ability to forecast
- ✓ G4 Scalability

Outline

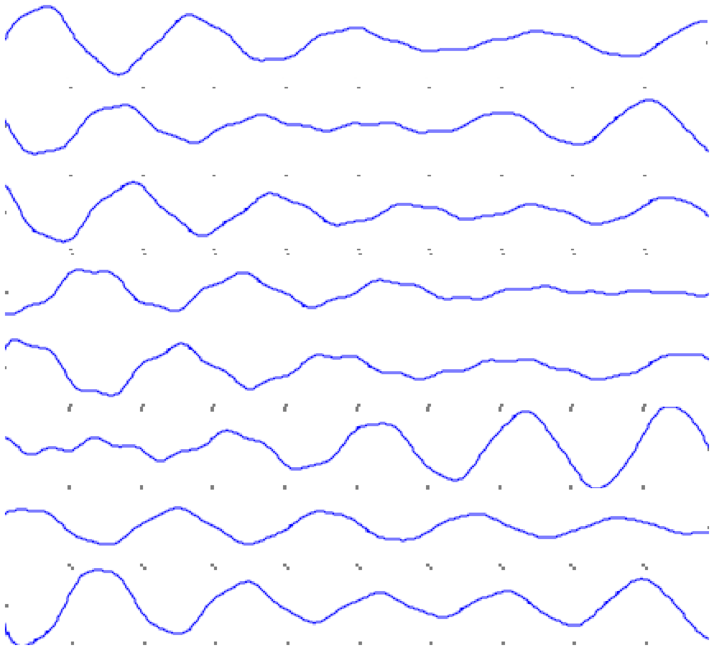
- Motivation
- Proposed Method: Intuition & Example
- Experiments & Results
- PLiF: Insight Details
- Conclusion

Proposed Method: PLiF

- **S1** Learning Dynamics
- **S2** Finding Canonical Form
- **S3** Handling the Lag
- **S4** Grouping Harmonics

Step 1. Learning Dynamics

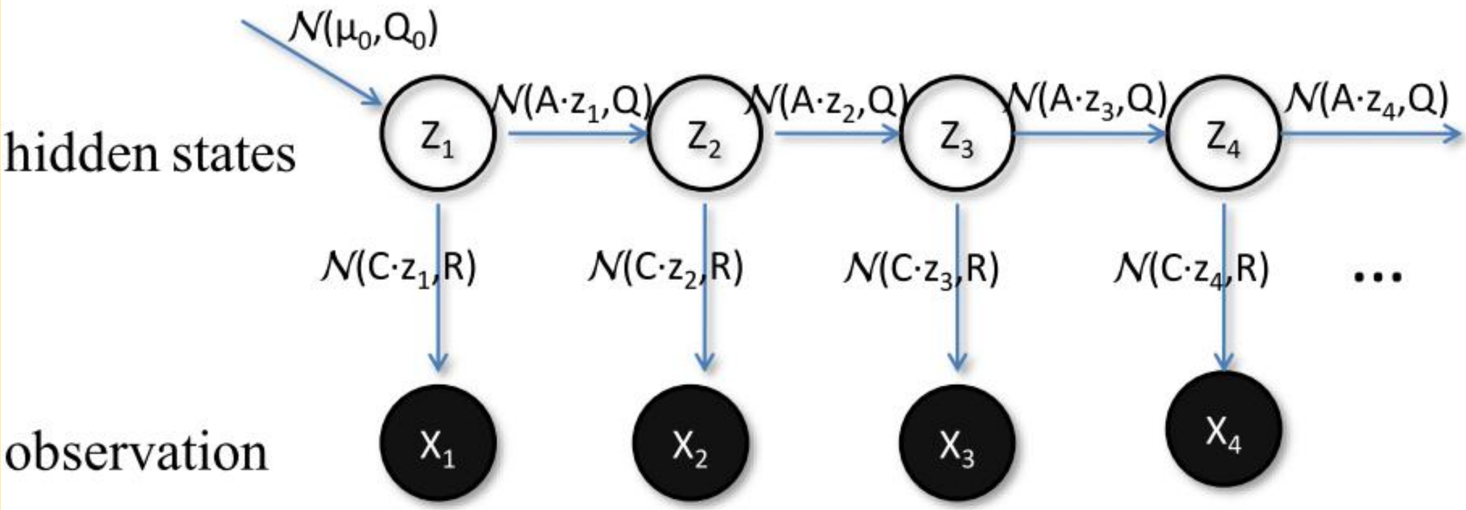
- Use machine learning to find:
  - “Transition” of Hidden Variables (HV): one time-tick to other
  - “Mixing” weights: HVs → observed data



Time series of hidden variables

Details

Underlying Model: Linear Dynamical Systems



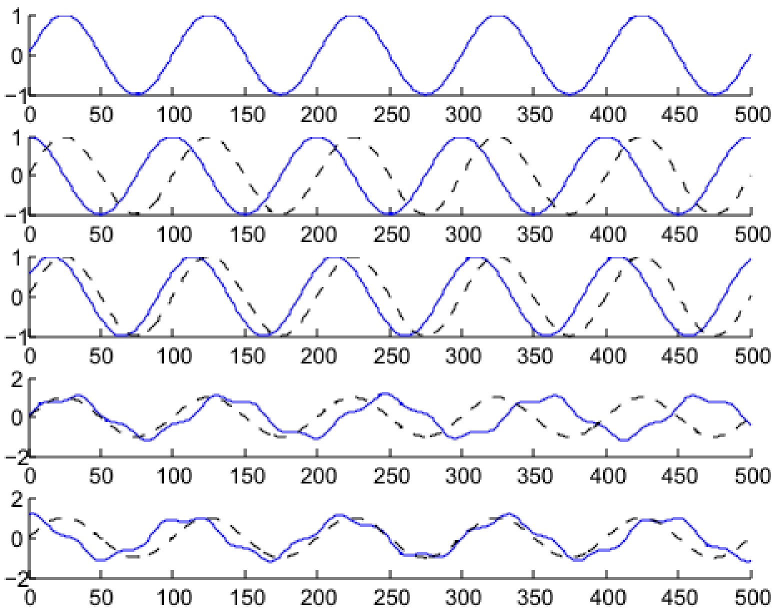
Model parameters:

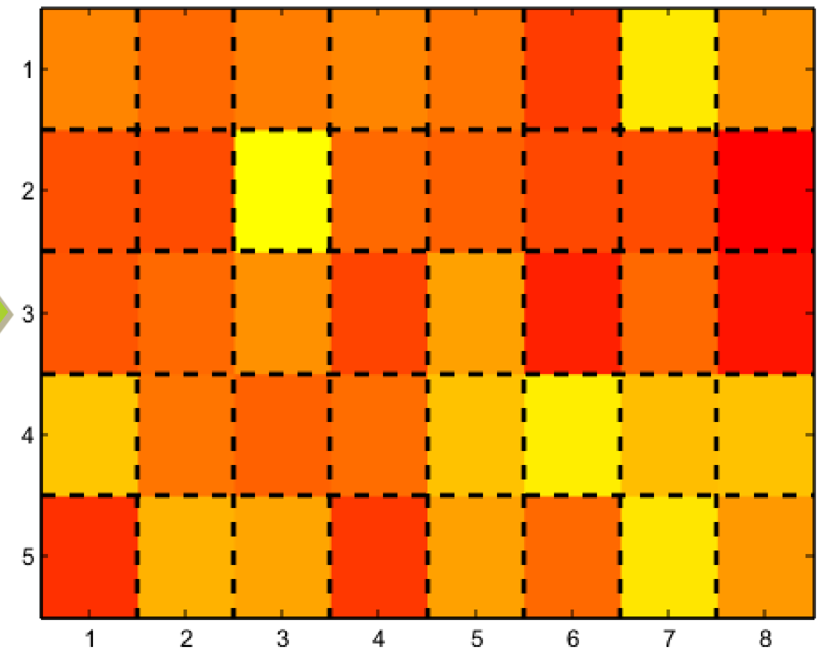
$\theta = \{\mu_0, Q_0, A, Q, C, R\}$

$$z_1 = \mu_0 + \omega_0$$
$$z_{n+1} = A \cdot z_n + \omega_n$$
$$x_n = C \cdot z_n + \varepsilon_n$$

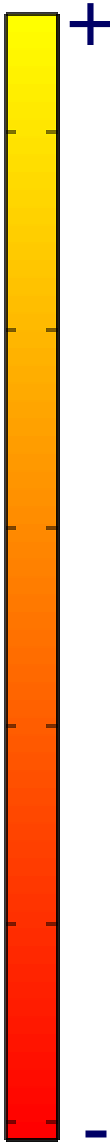
Dynamics/Transition in Hidden Variables

Mixing Weights





mixing/output matrix C



Learning the Parameters

- Expectation-Maximization
- maximizing the expected log likelihood:

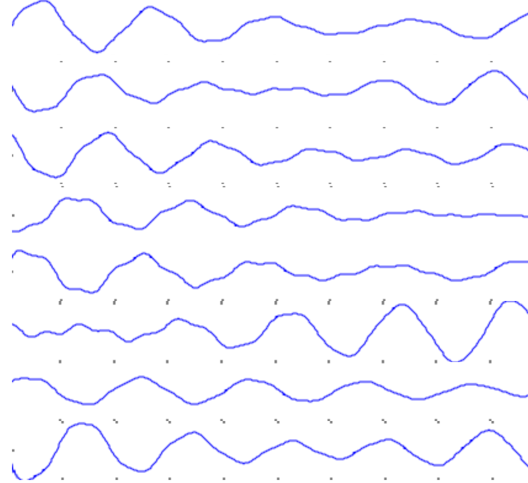
$$L(\theta;\mathcal{X}) = \mathbb{E}_{\mathcal{X},\mathcal{Z}|\theta}[-D(\vec{z}_1,\vec{\mu}_0,\mathbf{Q}_0) - \sum_{t=2}^{T} D(\vec{z}_t,\mathbf{A}\vec{z}_{t-1},\mathbf{Q}) - \sum_{t=1}^{T} D(\vec{x}_t,\mathbf{C}\vec{z}_t,\mathbf{R}) - \frac{1}{2}\log|\mathbf{Q}_0| - \frac{T-1}{2}\log|\mathbf{Q}| - \frac{T}{2}\log|\mathbf{R}|] \quad (13)$$

Standard EM: expensive!

Further speed optimization in our PLiF: matrix inversion using Woodbury matrix identity

Step 2: Canonicalization

- But, hidden variables
  - hard to interpret
  - non-unique: many combinations are essentially the same
- Intuition:
  - To make hidden variables compact and “uniquely” identified



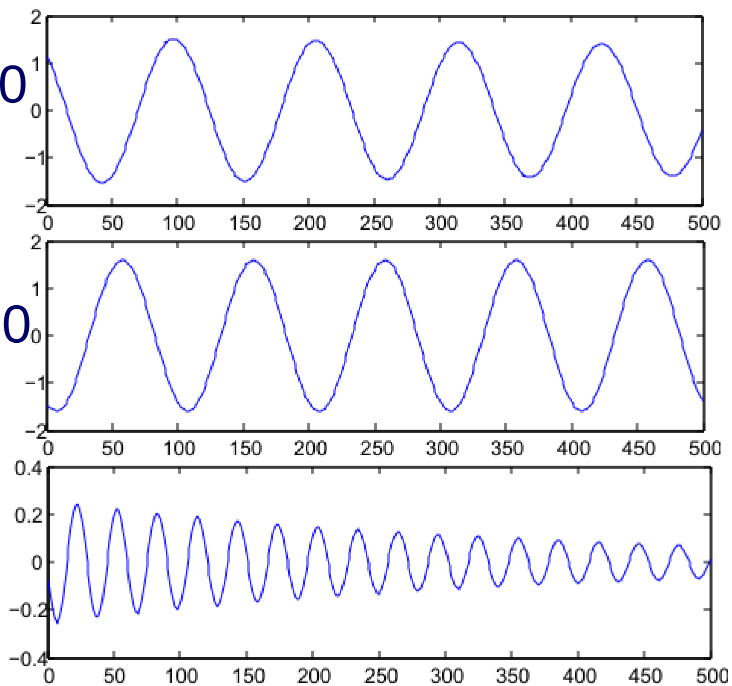
HV before

Canonicalization adds Interpretability

f=1/110

f=1/100

f=1/30



"Harmonics"

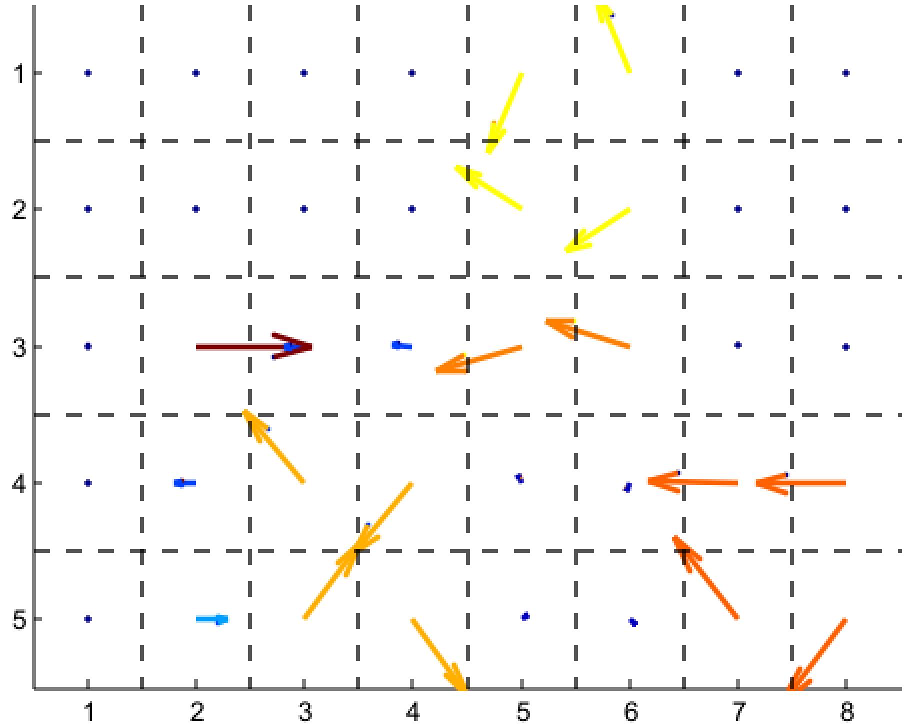
frequency

scaling (subtle)

Time series of HV after canonicalization (real part)

Step 2: Canonicalization

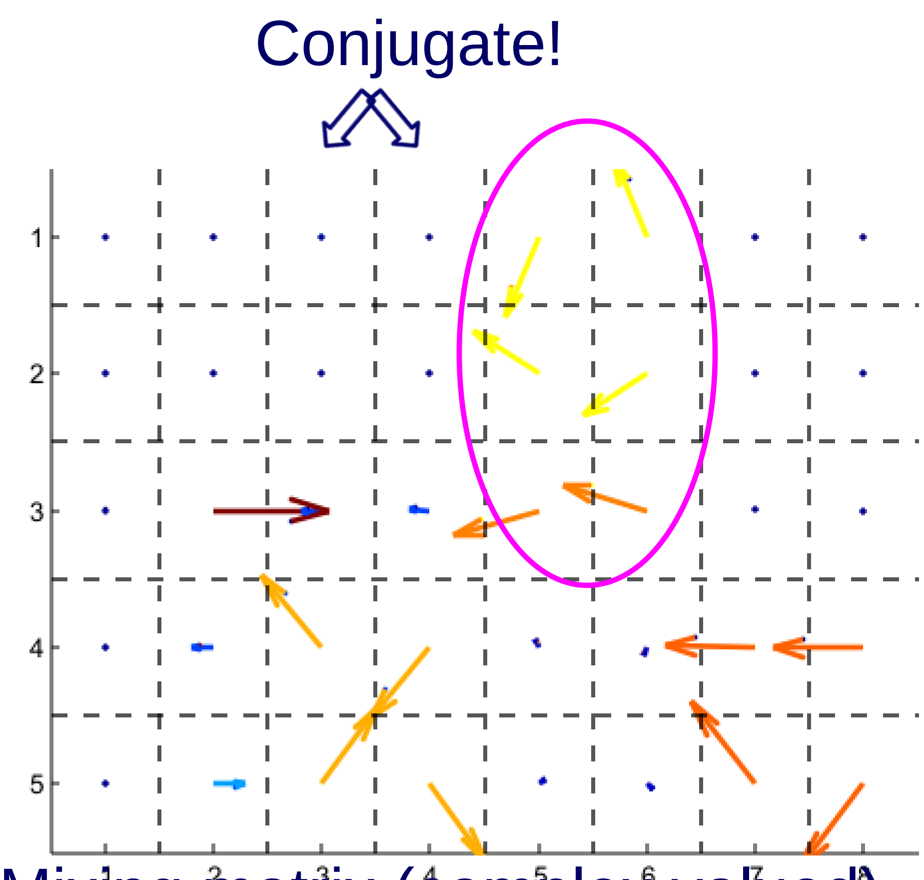
- Again,
  Estimating how each signal is composed of “harmonics”/patterns
- but, in complex space



Mixing matrix (complex valued)

Step 3:Handling Lag

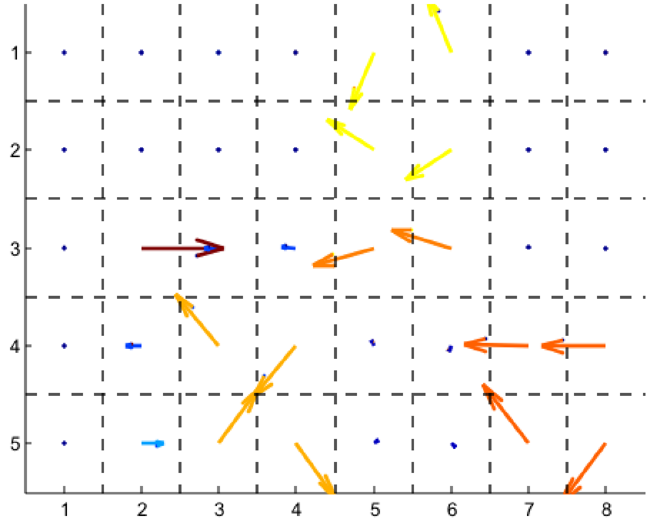
- Intuition:
  - Groups emerge..
  - reducing redundancy
  - eliminating phase shift

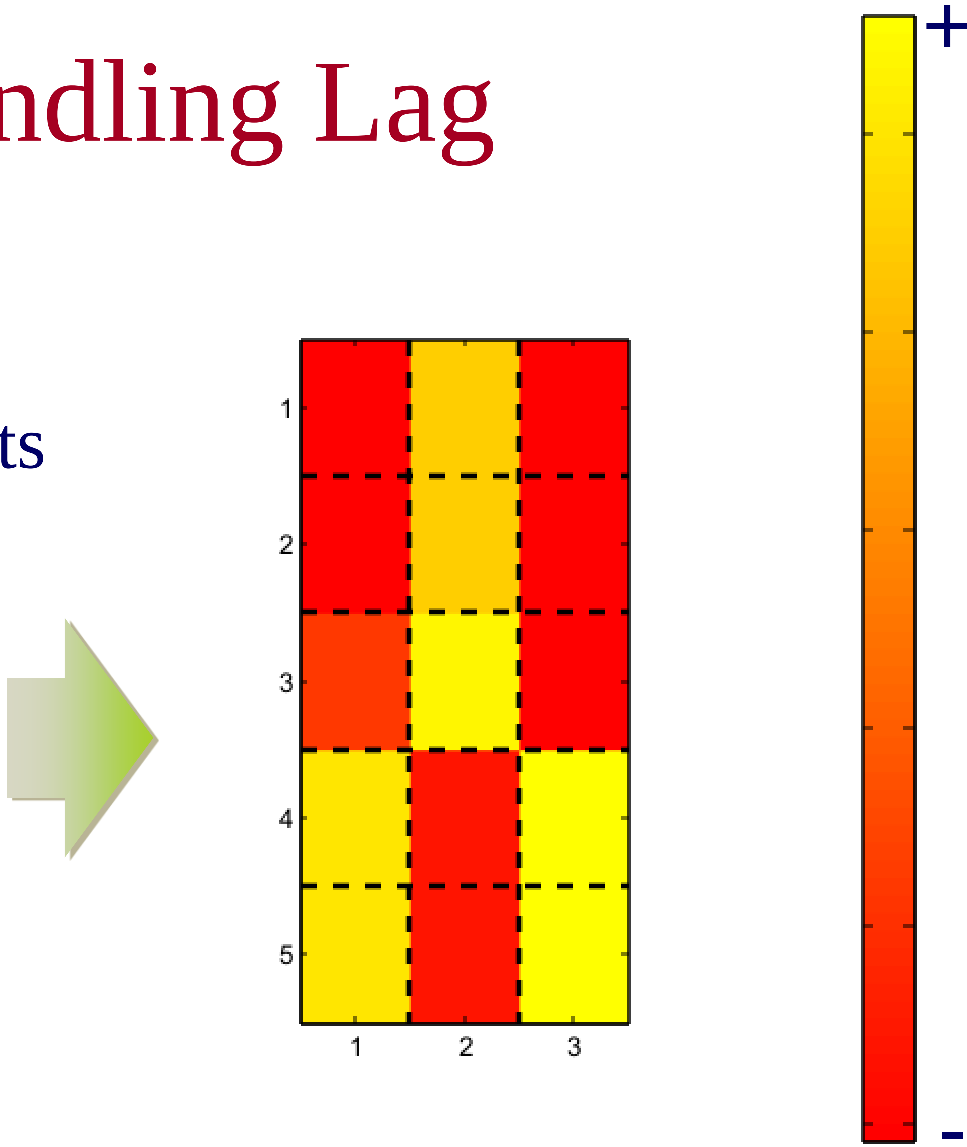


Mixing matrix (complex valued)

Step 3:Handling Lag

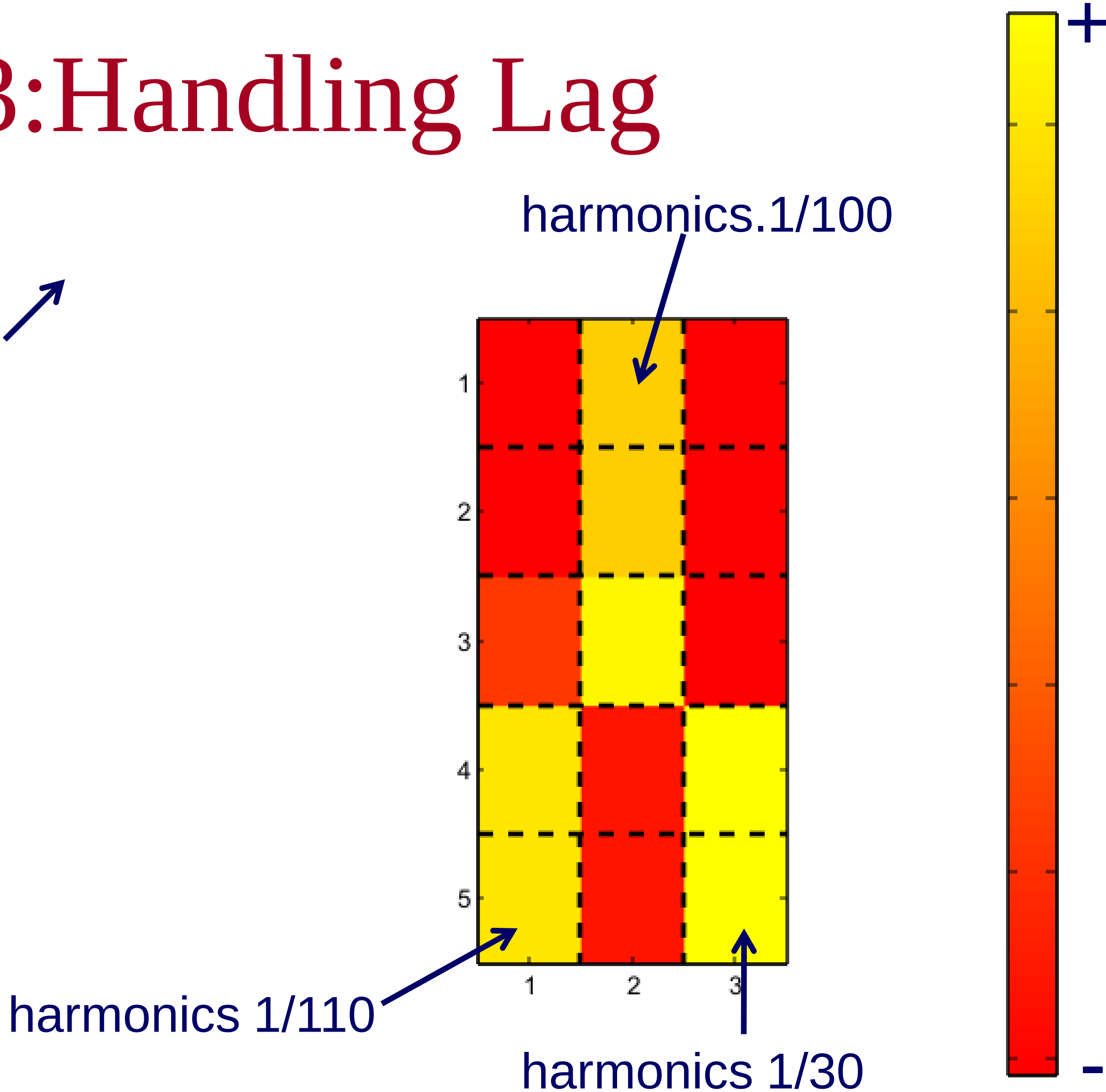
- Idea:
  – only magnitude counts
  – removing duplicates





Step 3:Handling Lag

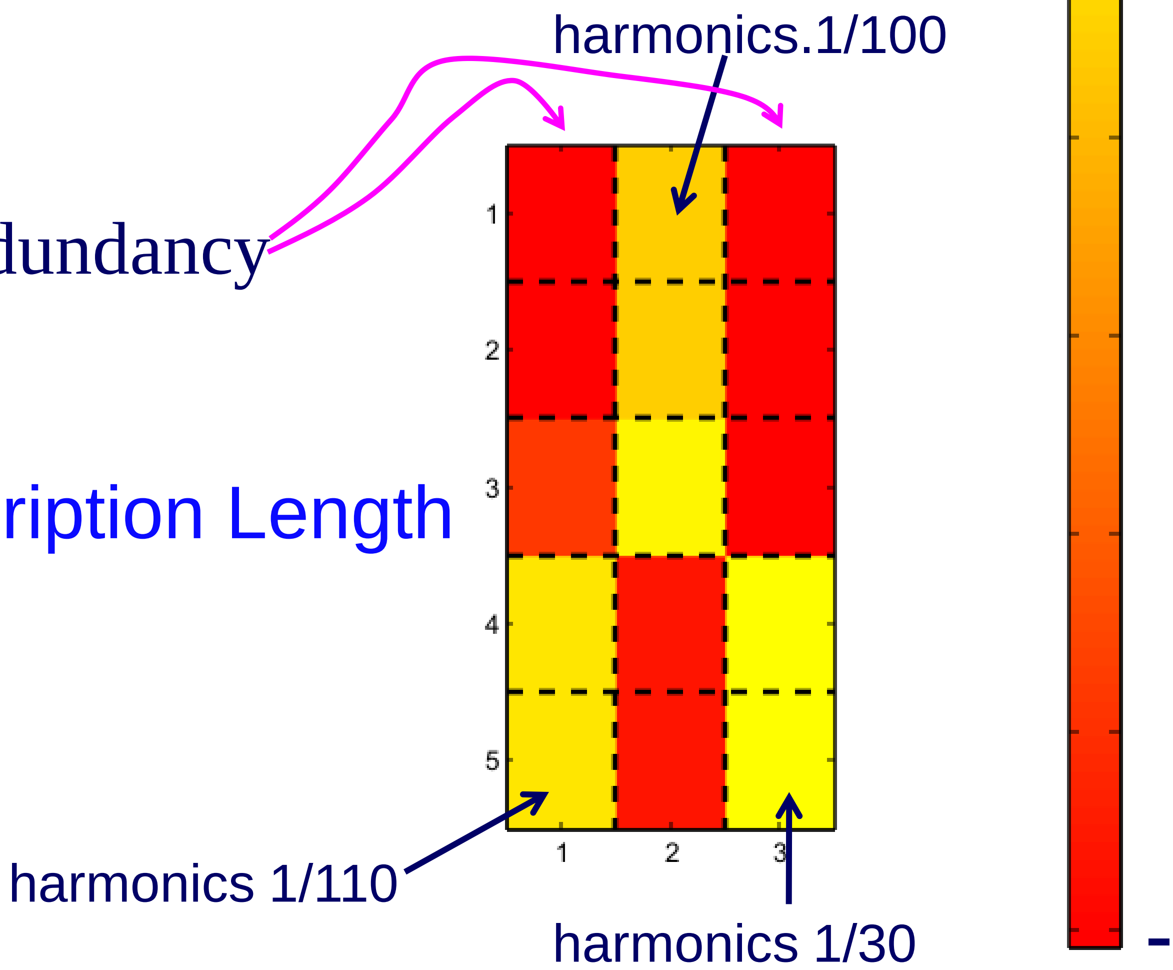
- interpretability ↗



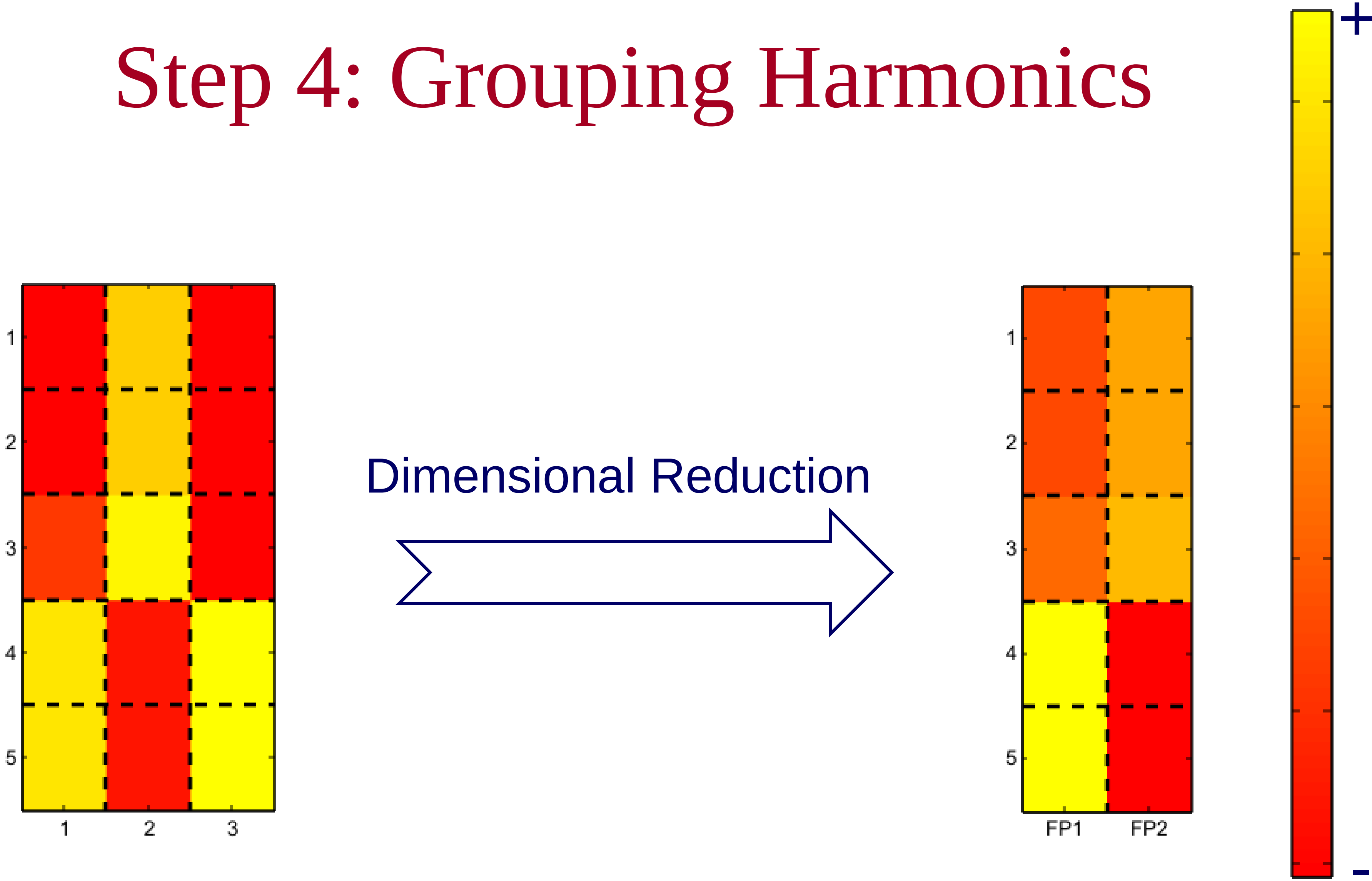
Step 4:Grouping Harmonics

- Intuition:
  - Still a little redundancy

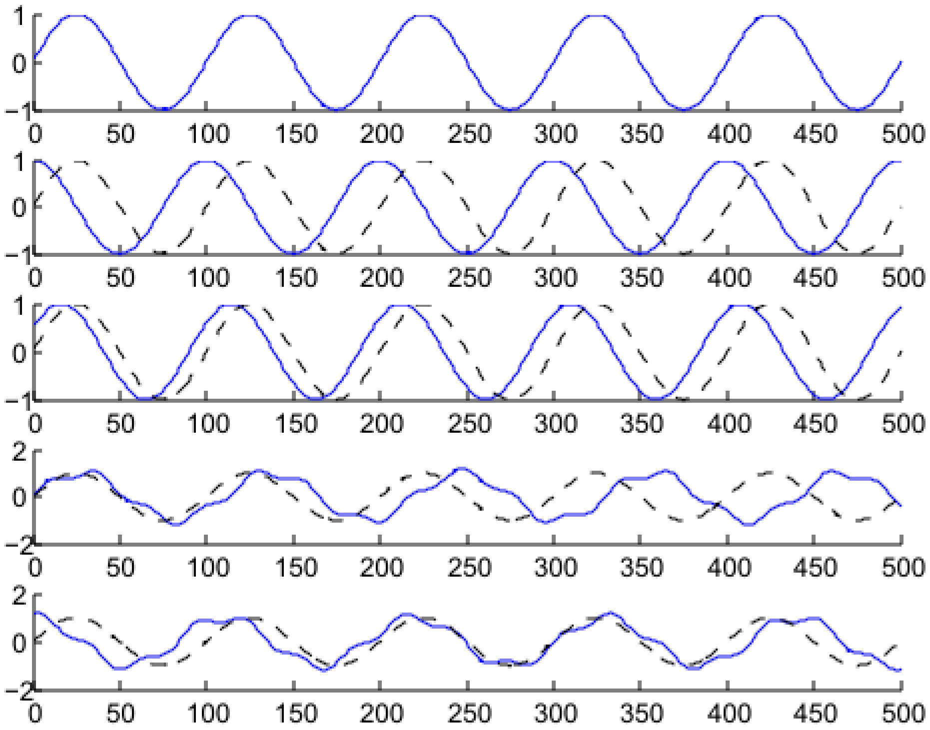
Think Minimum Description Length

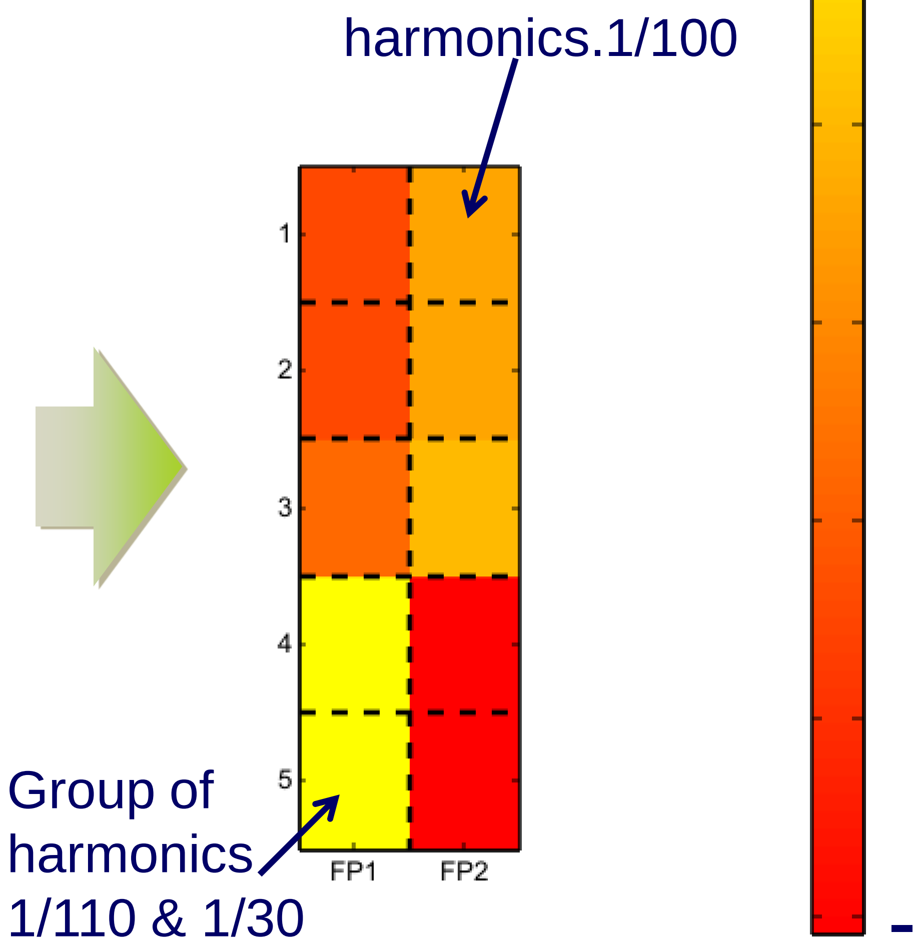


Step 4: Grouping Harmonics



Step 4: Grouping Harmonics





Parsimonious Linear Fingerprinting Goals ←→ steps

PLiF Goals

- G1 Good features/similarity function
  - (1a) lag independent
  - (1b) frequency proximity
  - (1c) grouping harmonics
- G2 Good compression
- G3 Ability to forecast
- G4 Scalability

PLiF alg. steps

- S1 Learning Dynamics
- S2 Canonical Form
- S3 Handling Lag
- S4 Grouping Harmonics

Outline

- Motivation
- Proposed Method: Intuition & Example
- Experiments & Results
- PLiF: Insight Details
- Conclusion

Conclusion

- Need for finding compact representation of time series data
- Intuition & Insights of PLiF
- Interpretation of PLiF & How it works
- Experiments on a diverse set of data
  - It really works!
  - It is fast & scalable.

Parsimonious Linear Fingerprinting

Goals ←→ steps

PLiF Goals

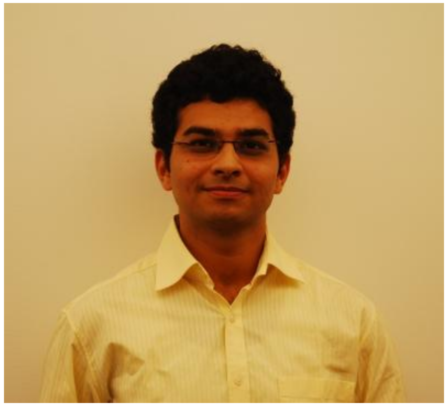
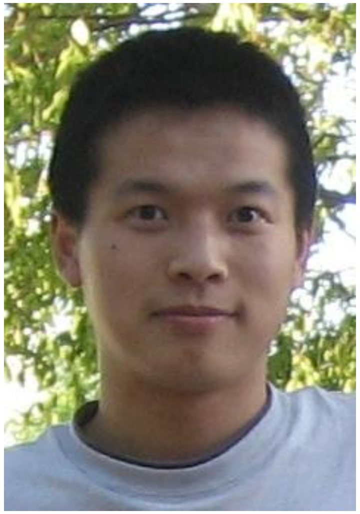
- G1 Good features/similarity function
  - (1a) lag independent
  - (1b) frequency proximity
  - (1c) grouping harmonics
- G2 Good compression
- G3 Ability to forecast
- G4 Scalability

PLiF alg. steps

- S1 Learning Dynamics
- S2 Canonical Form
- S3 Handling Lag
- S4 Grouping Harmonics

Question?

- Thanks!


B. Aditya Prakash

Lei Li

Christos Faloutsos

http://www.cs.cmu.edu/~leili/

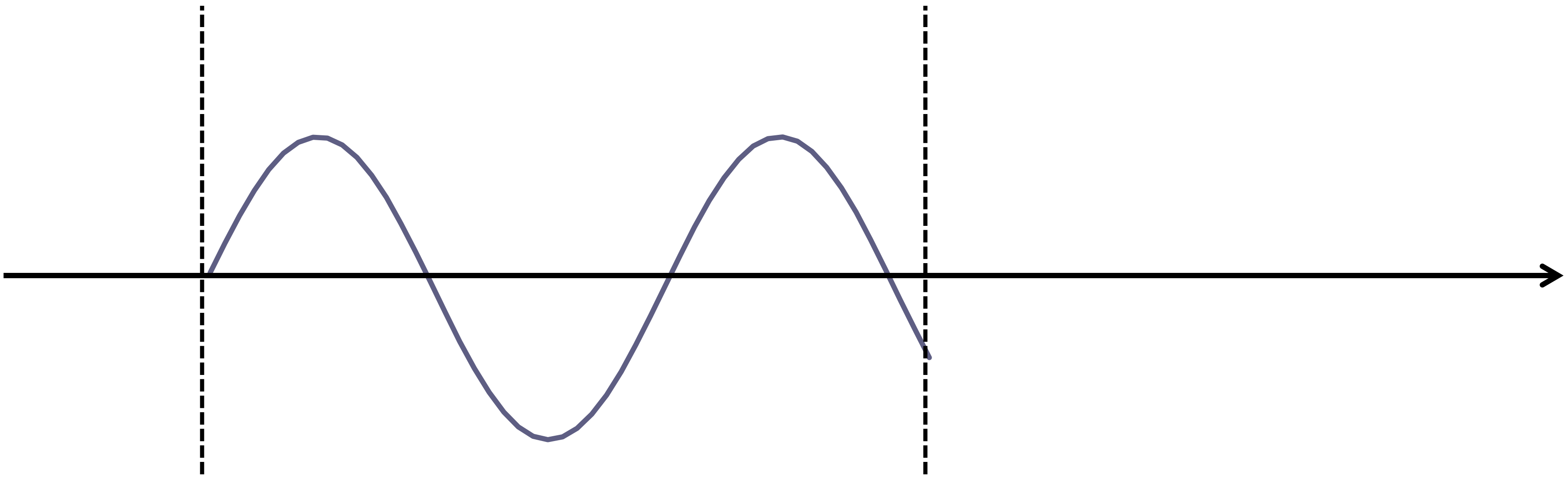
leili@cs.cmu.edu

appendix

BACKUP

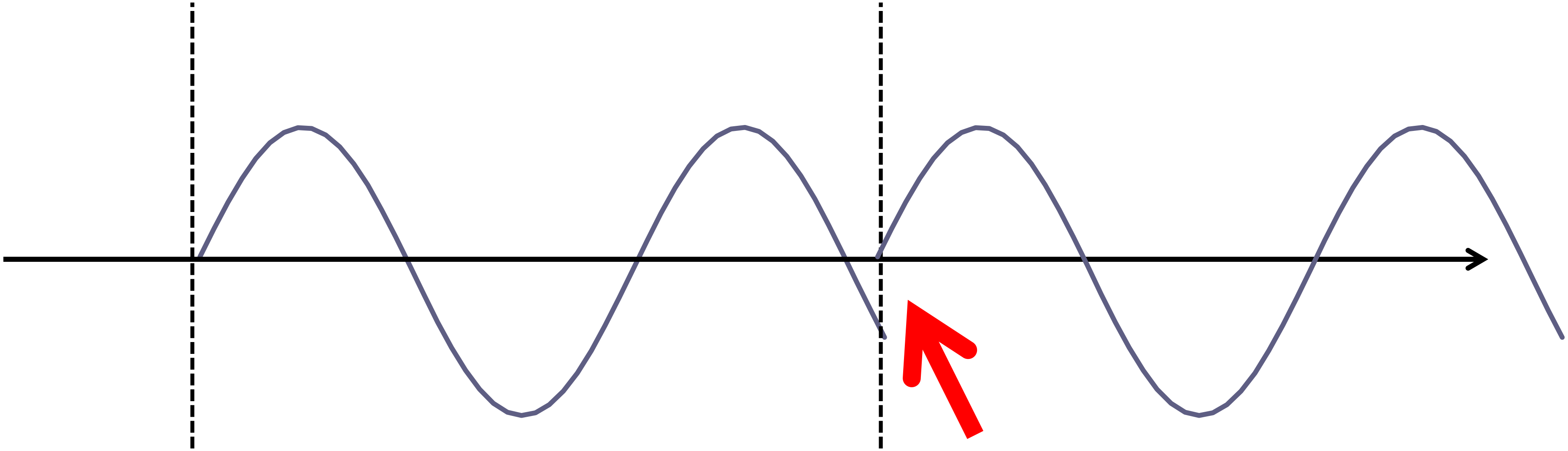
Why not Fourier (DFT)?

1. FT cannot do forecasting

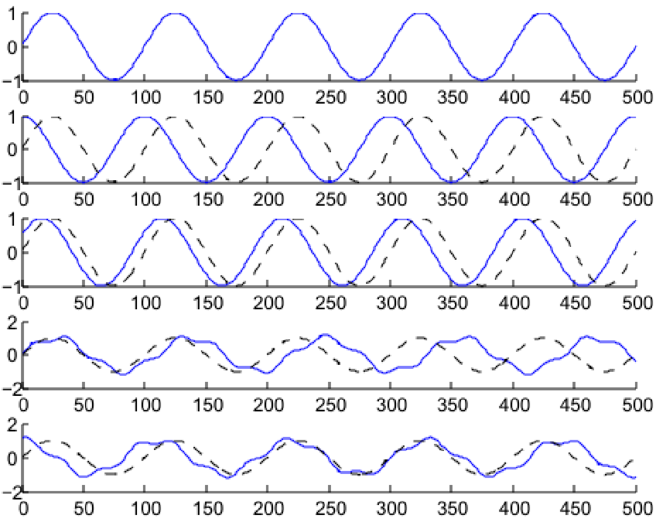


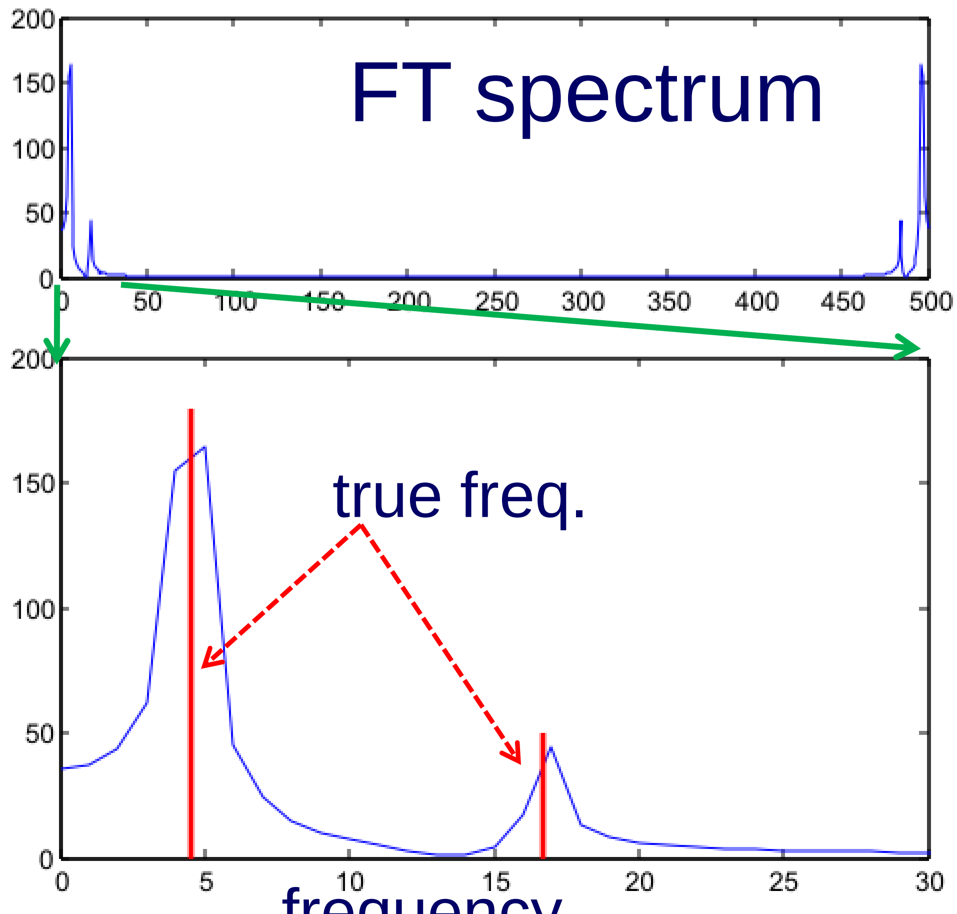
Why not Fourier (DFT)?

1. FT cannot do forecasting



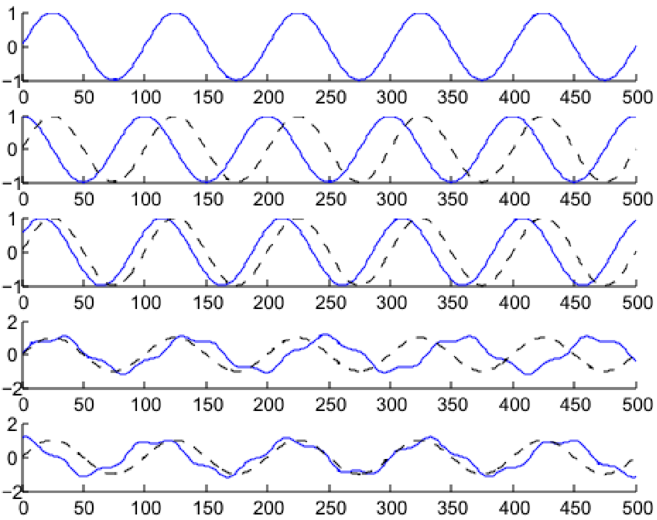
Why not Fourier (DFT)?



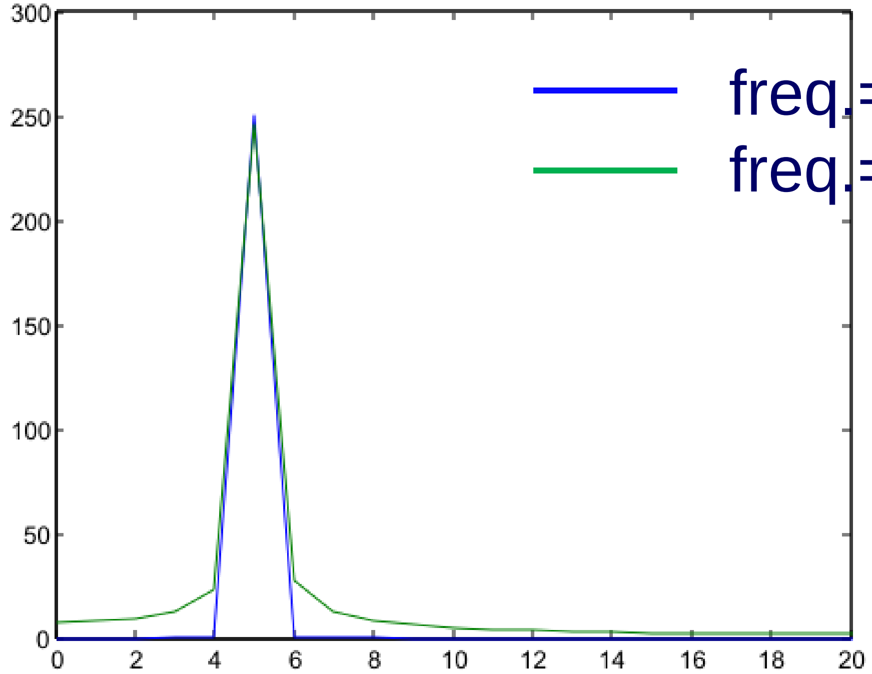


1. FT cannot do forecasting
2. No arbitrary frequency

Why not Fourier (DFT)?



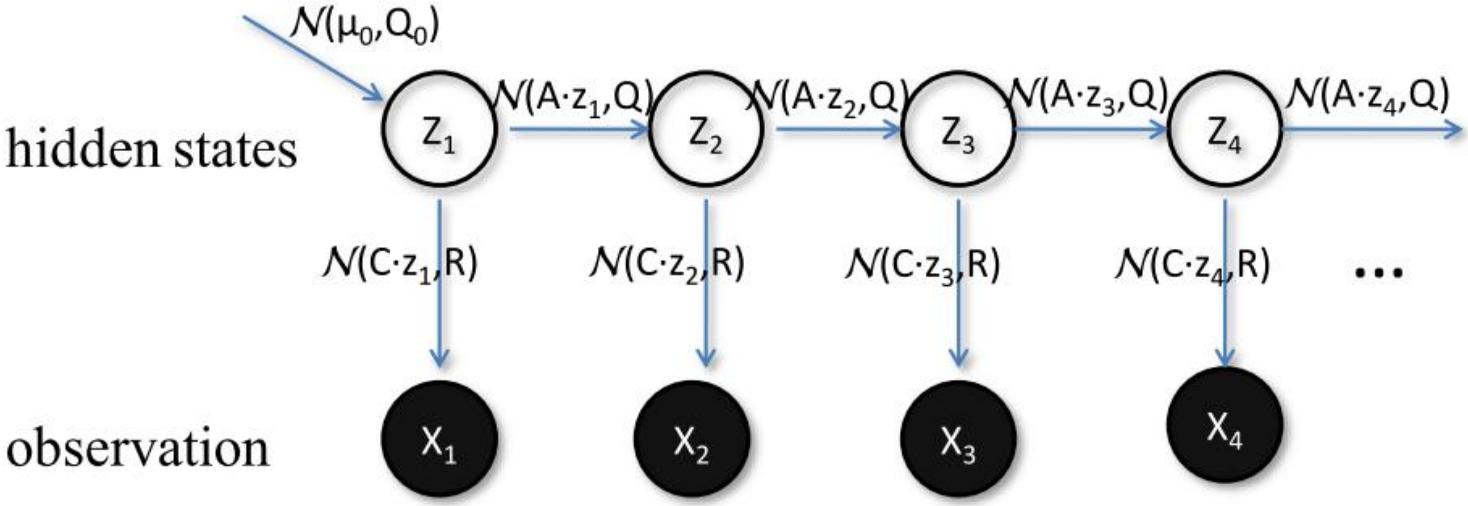
1. FT cannot do forecasting
2. No arbitrary frequency
3. nearby frequency treated differently, not suited for across signals



Details for Implementation

Read this only if you want to implement it

Modelling the data: Linear Dynamical Systems



Model parameters:

$\theta=\{\mu_0, Q_0, A, Q, C, R\}$

$$
\begin{aligned}
z_1 &= \mu_0 + \omega_0 \\
z_{n+1} &= A\cdot z_n + \omega_n \\
x_n &= C\cdot z_n + \varepsilon_n
\end{aligned}
$$

Linear Dynamical Systems: parameters

	name	meaning & example
μ_0	initial state for hidden variable	e.g. initial position, velocity & acceleration
A	transition matrix	how the states move forward, e.g. soccer flying in the air
C	transmission/ projection/ output matrix	hidden state → observation, e.g. camera taking picture of the soccer
Q_0	Initial covariance	
Q	transition covariance	how precision is the soccer motion
R	transmission/ projection covariance	i.e. observation noise; e.g. how accurate is the camera

Learning the Dynamics

- Expectation-Maximization
- maximizing the expected log likelihood

$$
\begin{aligned}
L(\theta;\mathcal{X}) \quad = \quad & \mathbb{E}_{\mathcal{X},\mathcal{Z}|\theta}[-D(\vec{z}_1,\vec{\mu}_0,\mathbf{Q}_0) \\
& -\sum_{t=2}^{\mathrm{T}} D(\vec{z}_t,\mathbf{A}\vec{z}_{t-1},\mathbf{Q}) - \sum_{t=1}^{\mathrm{T}} D(\vec{x}_t,\mathbf{C}\vec{z}_t,\mathbf{R}) \\
& -\frac{1}{2}\log|\mathbf{Q}_0| - \frac{\mathrm{T}-1}{2}\log|\mathbf{Q}| - \frac{T}{2}\log|\mathbf{R}|]
\end{aligned}
$$

Finding Canonical Form

- Intuition: find the canonical dynamics
- taking eigenvalue decomposition of the transition matrix A

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^*$$

- compensate C with

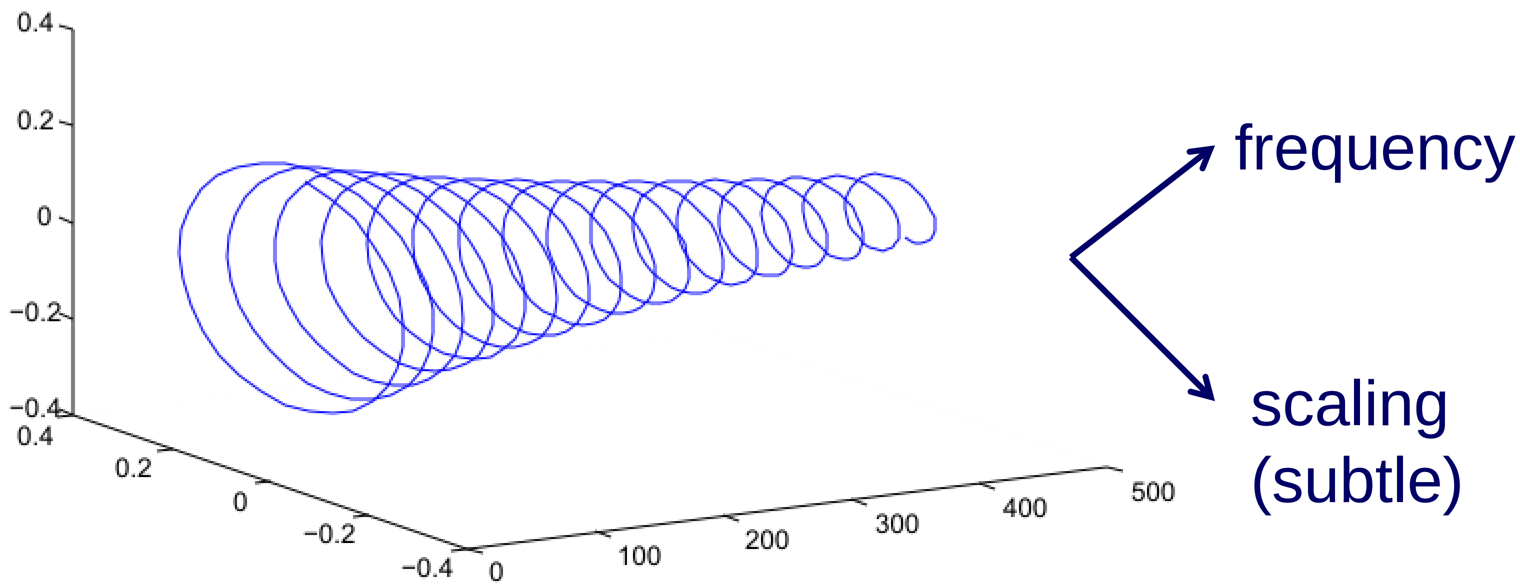
$$\mathbf{C}_h = \mathbf{C} \cdot \mathbf{V}$$

- C_h is a projection of the data to the dynamics
- but...

Lags and Harmonics group

- Handling the lag:
  - Intuition: phase/shift should not matter
  - step: eliminating duplicate conjugate in C_h, taking magnitude, ==> C_m
- Group harmonics
  - taking SVD or PCA on C_m
  - resulting fingerprints H_1

3D VIEW OF HIDDEN VARIABLES



Example: parsimonious HV after canonicalization

SPEEDUP OPTIMIZATION

Scalability

- Speedup the computation of matrix inverse using Woodbury matrix identity

$$(\mathbf{X}+\mathbf{Y}\mathbf{Z}\mathbf{Y}^T)^{-1} = \mathbf{X}^{-1}-\mathbf{X}^{-1}\mathbf{Y}(\mathbf{Z}^{-1}+\mathbf{Y}^T\mathbf{X}^{-1}\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}^{-1}$$