



Quantifying Dexterity in Human Activities Through the Time-delay Embedded Phase Space Representation

Miguel Perez-Xochicale

18th May 2015

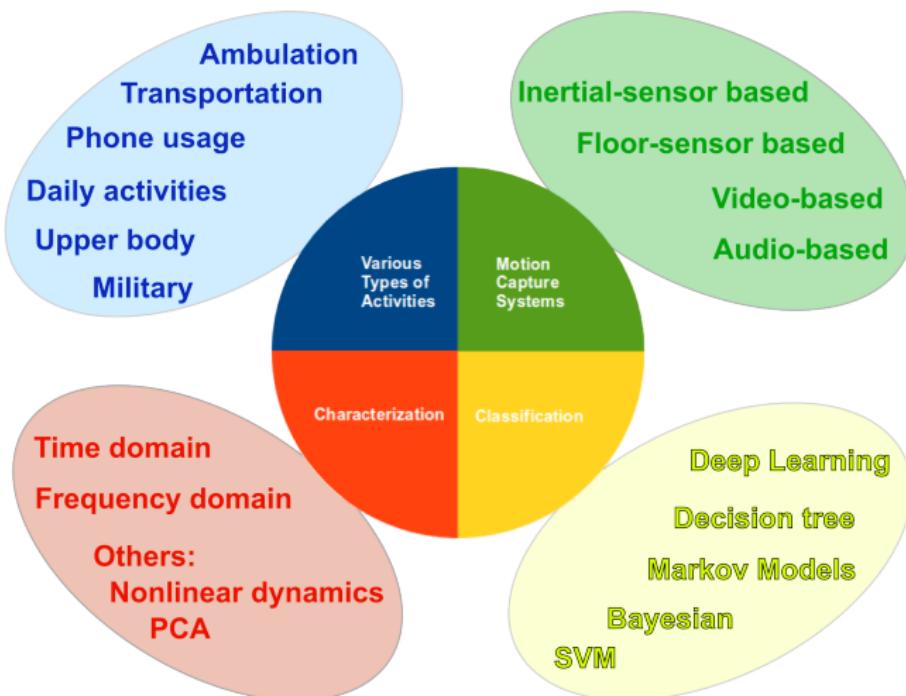


CONACYT
Consejo Nacional de Ciencia y Tecnología

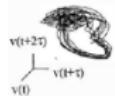


UNIVERSITY OF
BIRMINGHAM

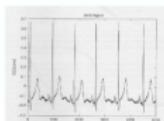
Why is HAR a challenging task?



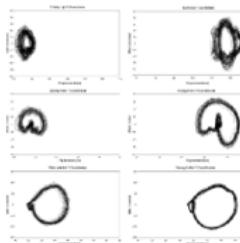
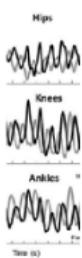
Study of electrophysiological time series



Korn et al (2003)



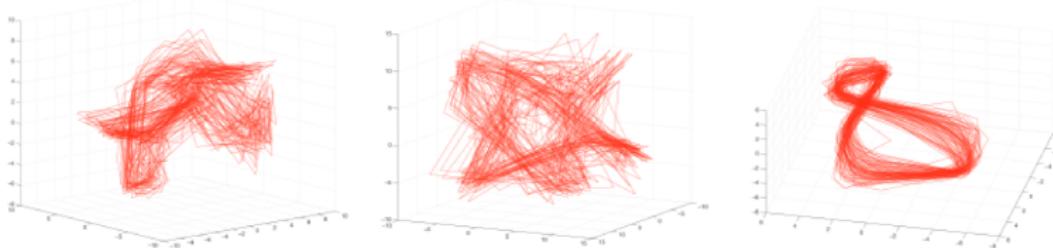
Fojt et al (1998)



Buzzi et al (2013)



Time Series Classification

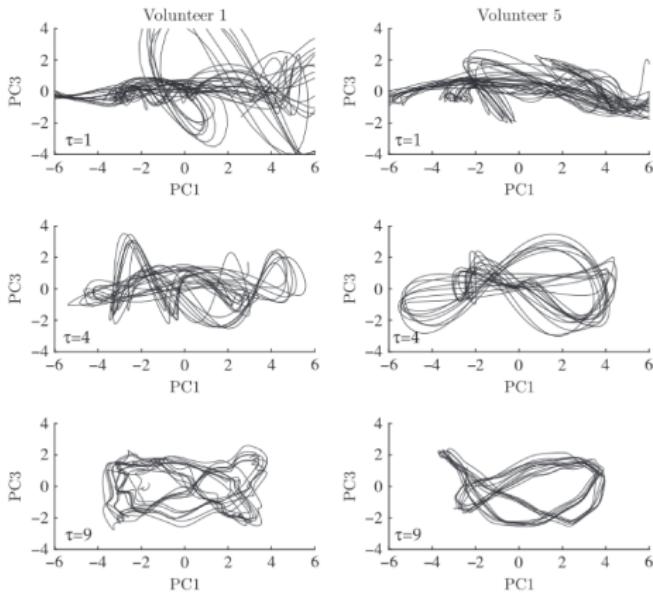


Jordan et al (2013)

Reconstructed state spaces for walking (left), running (middle), and biking (right) from noisy accelerometer data.



Gait Identification



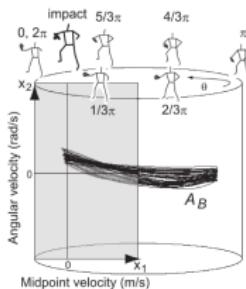
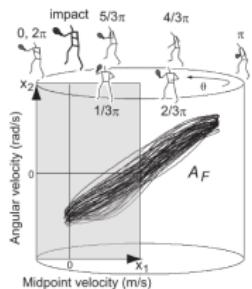
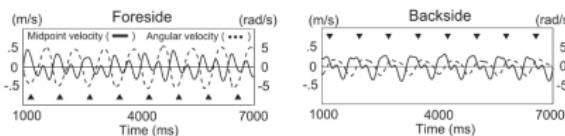
Sama et al (2013)

Reconstruction of the trajectory of the first and third PC for two individuals.

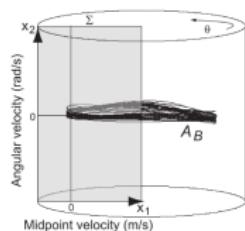
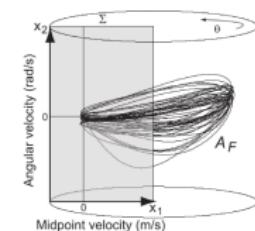
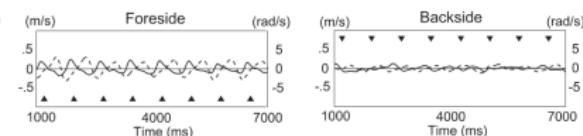


Dexterity of Tennis Players

Expert #5



Novice #4



Suzuki et al (2013)

Time series and hyper-cylindrical phase space from Expert #5 and Novice #4.

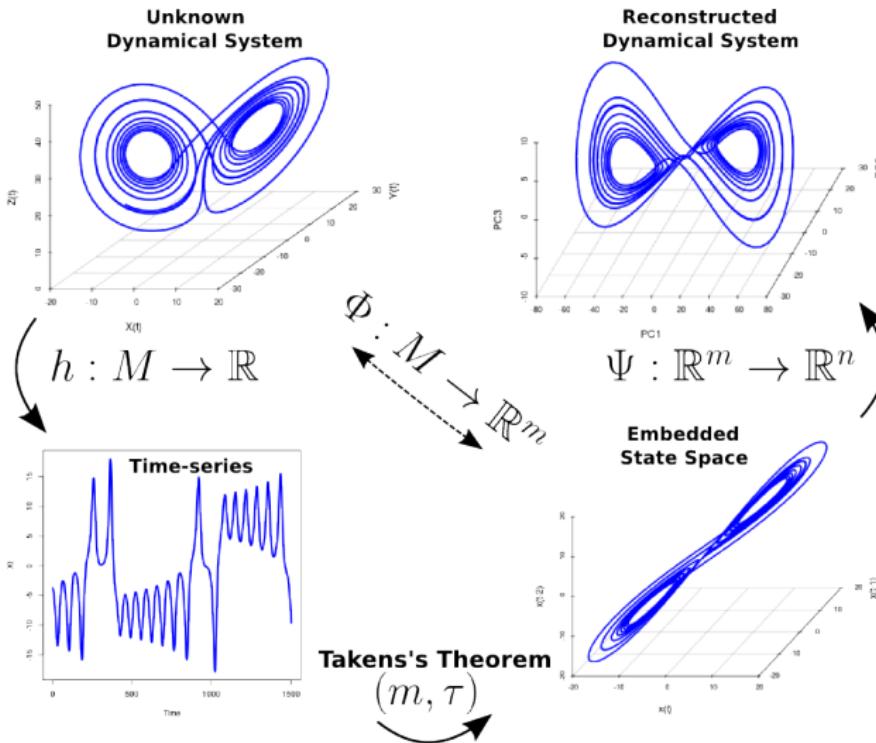


Research Questions

- How can phase space representation quantify the dexterity of human activities?
- How can we apply Takens's Theorem and Principal Component Analysis to characterise Human Activities?
- How do concepts from nonlinear dynamics help us to characterise dexterity in human activities?



Reconstructed State Space



Takens' Theorem (1981)

According to Takens' Theorem, the reconstructed state space in m **embedding dimension** with τ **embedding delay** of the original system is given by

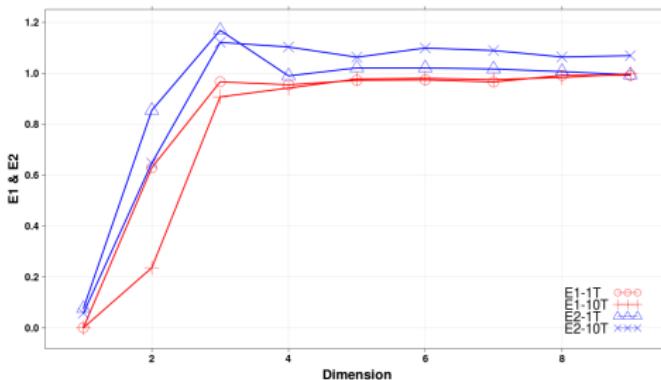
$$\overline{x(t)} = (x(t), x(t - \tau), x(t - 2\tau), \dots, x(t - (m - 1)\tau)).$$

Takens' Theorem, also known as time-delay embeddings method, states that for a large enough m to unfold the attractor and $\tau > 0$ chosen to maximize the information content of $x(t)$, this method provides a one-to-one reconstruction of the true dimension k system (\mathbb{R}^k).



Minimum embedding parameters

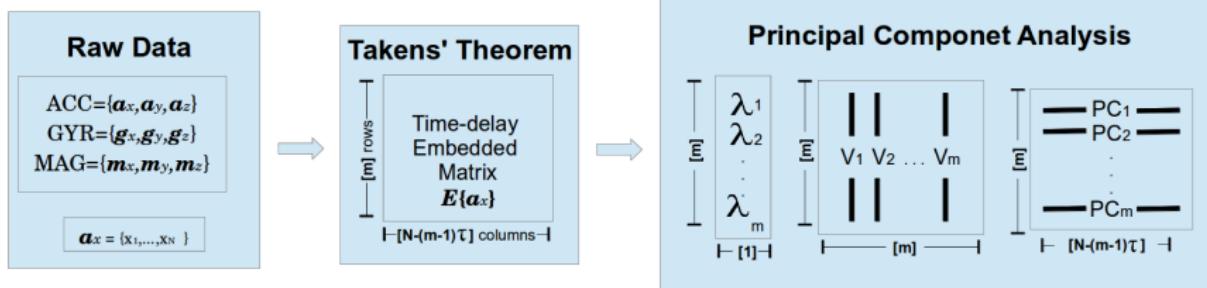
Cao (1997) proposed a method based on the false neighbor method to determine the minimum embedding dimension from time-series based on Taken's theorem.



The values E1 and E2 from Lorenz attractor Cao (1997)



Phase Space Reconstruction



Time-Delay Embedding Example

Lorenz System

$$\begin{aligned}\frac{dx}{dt} &= \sigma(x - y), \\ \frac{dy}{dt} &= x(\rho - z) - y, \\ \frac{dz}{dt} &= xy - \beta z.\end{aligned}$$

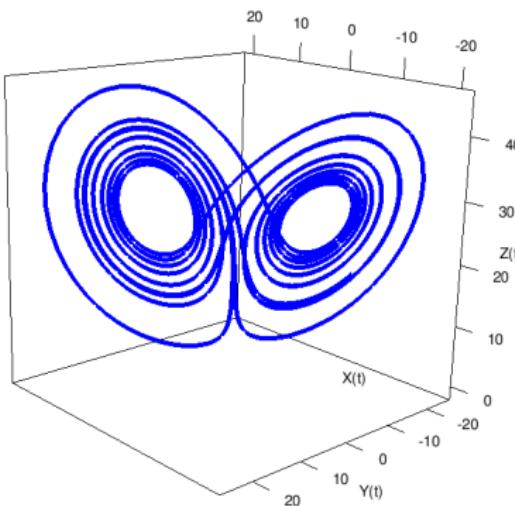


Figure : $\sigma = 10$, $\rho = 28$ and $\beta = 3/8$



Time-Delay Embedding Example

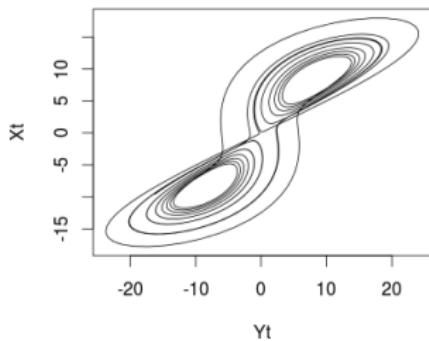
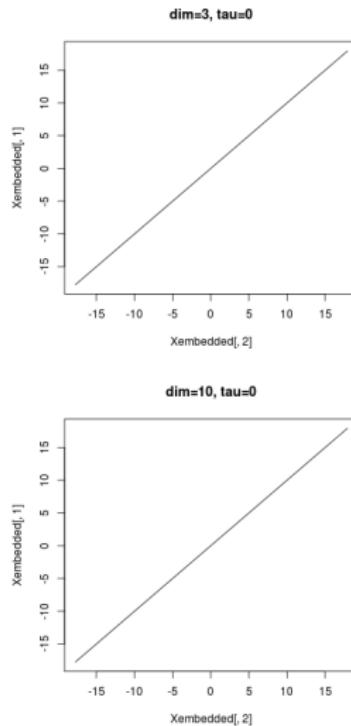


Figure : Original Manifold



Time-Delay Embedding Example

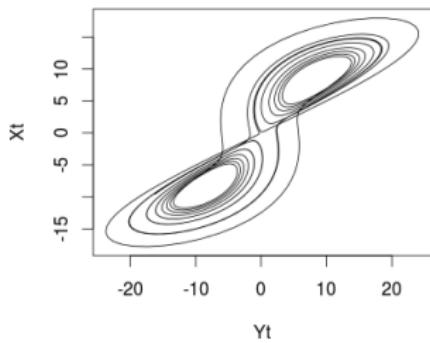
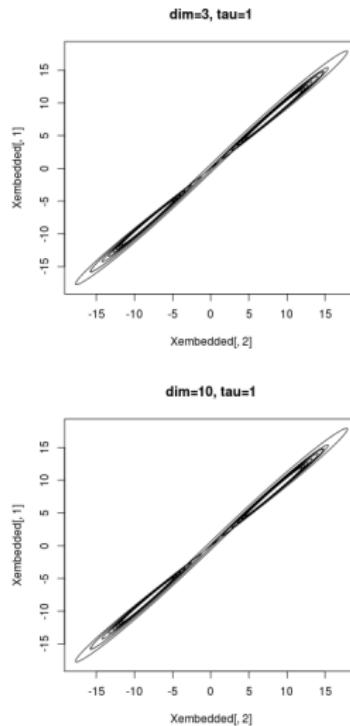


Figure : Original Manifold



Time-Delay Embedding Example

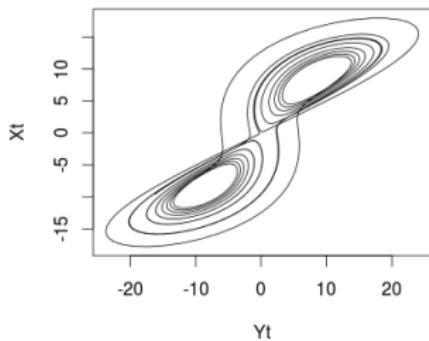
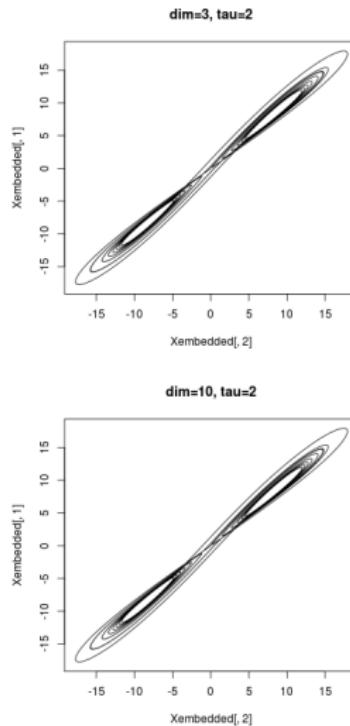


Figure : Original Manifold



Time-Delay Embedding Example

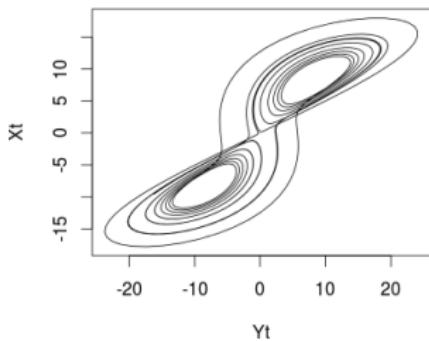
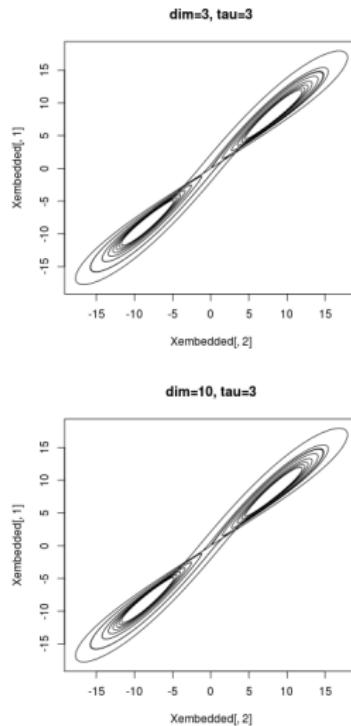


Figure : Original Manifold



Time-Delay Embedding Example

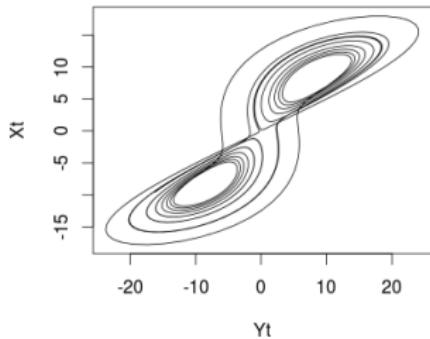
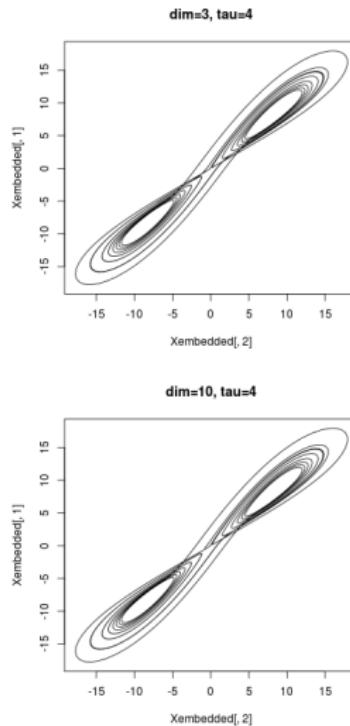


Figure : Original Manifold



Time-Delay Embedding Example

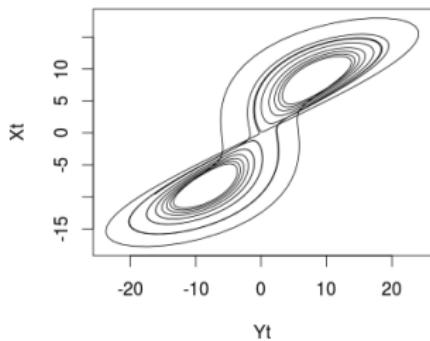
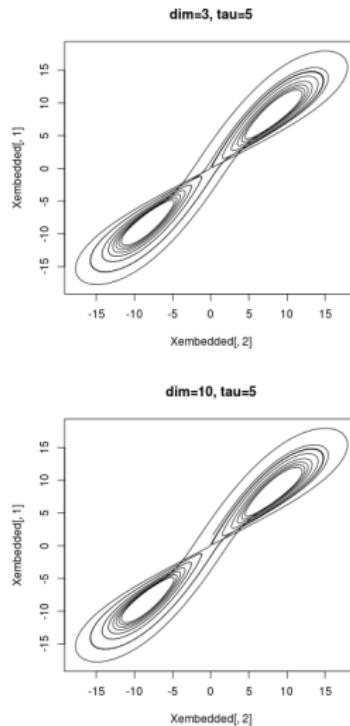


Figure : Original Manifold



Time-Delay Embedding Example

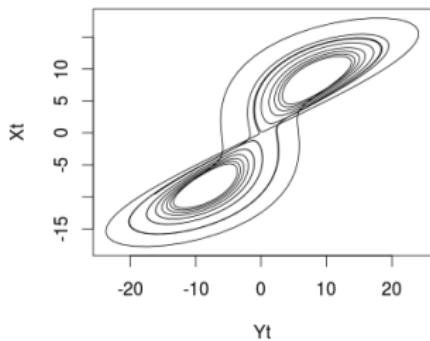
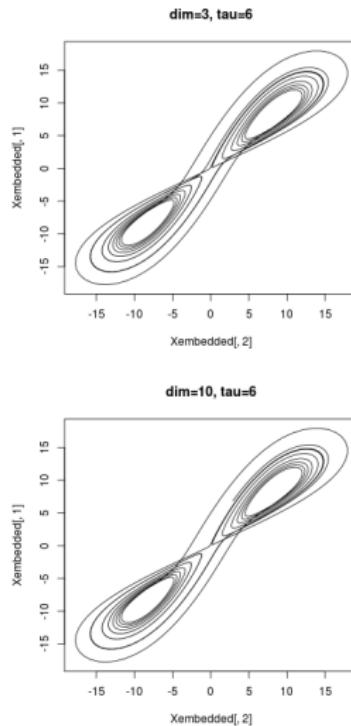


Figure : Original Manifold



Time-Delay Embedding Example

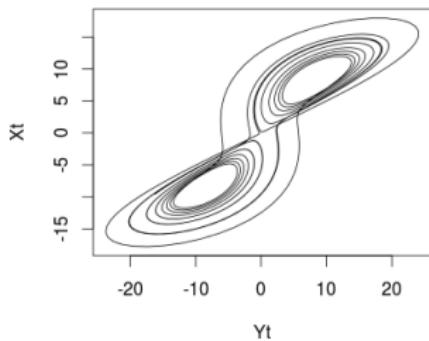
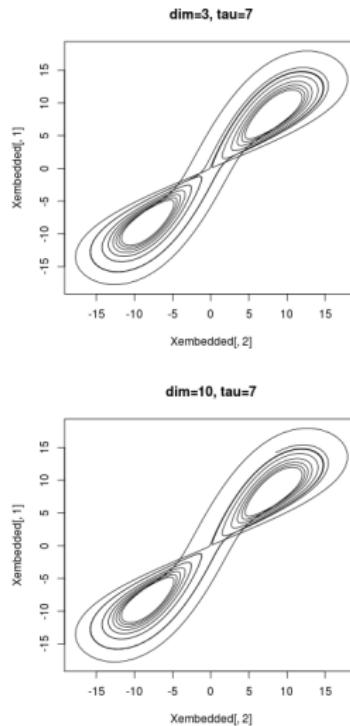


Figure : Original Manifold



Time-Delay Embedding Example

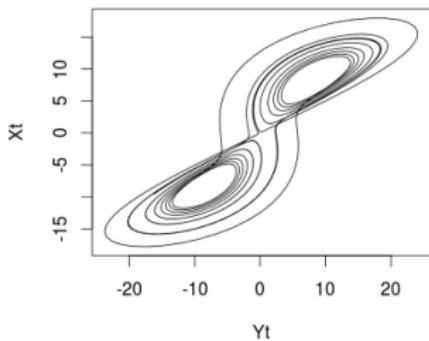
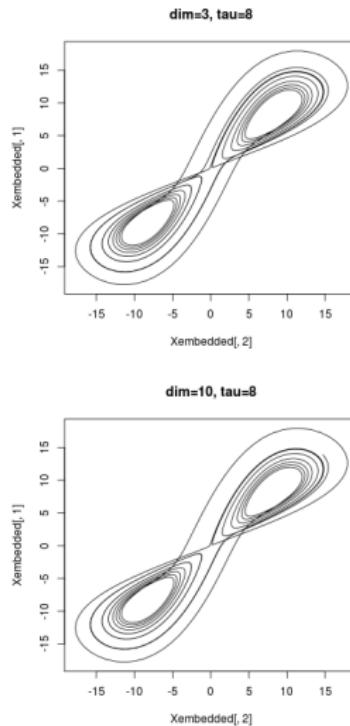


Figure : Original Manifold



Time-Delay Embedding Example

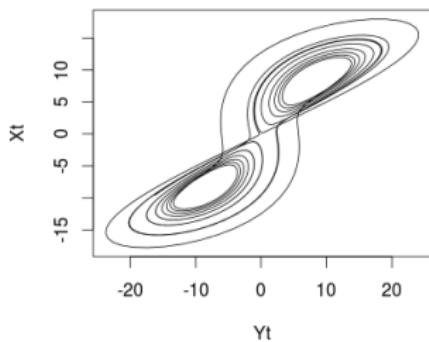
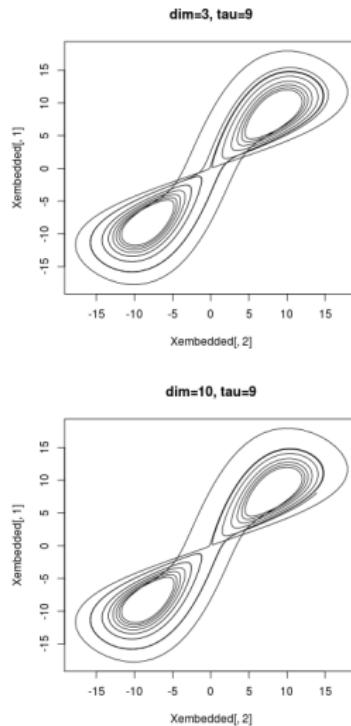


Figure : Original Manifold



Time-Delay Embedding Example

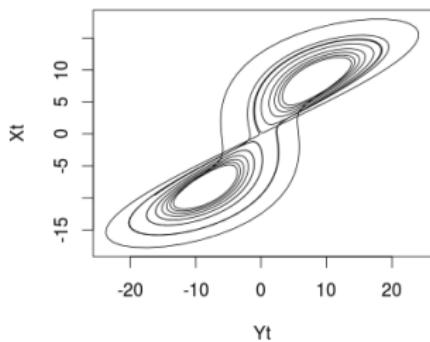
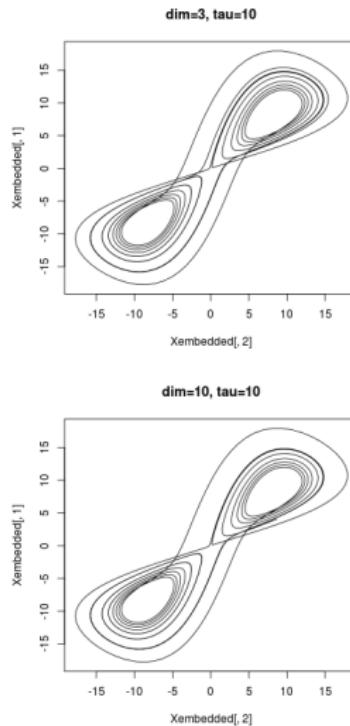


Figure : Original Manifold



Time-Delay Embedding Example

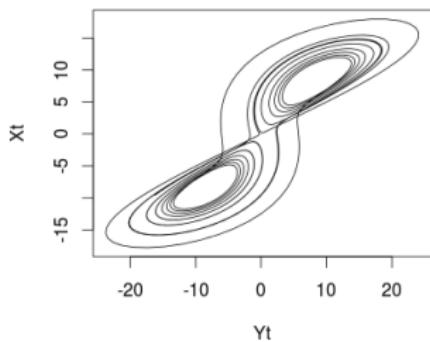
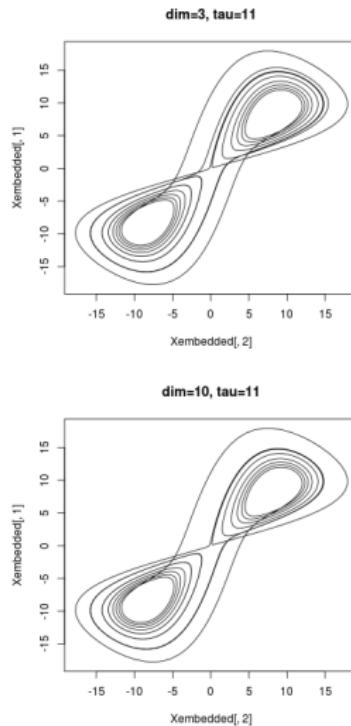


Figure : Original Manifold



Time-Delay Embedding Example

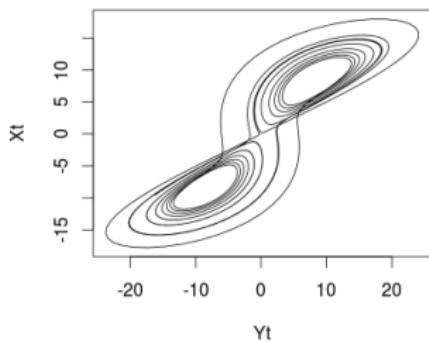
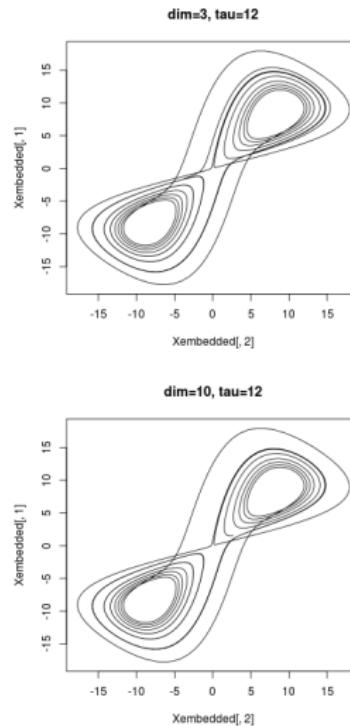


Figure : Original Manifold



Time-Delay Embedding Example

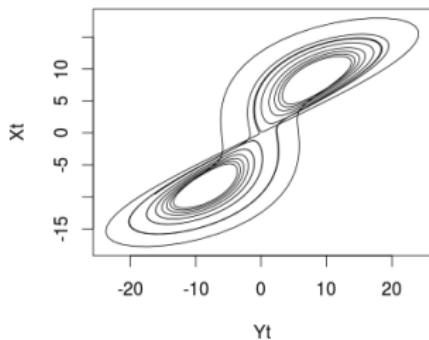
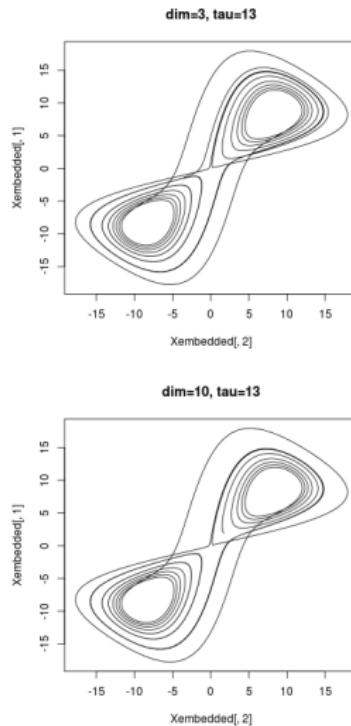


Figure : Original Manifold



Time-Delay Embedding Example

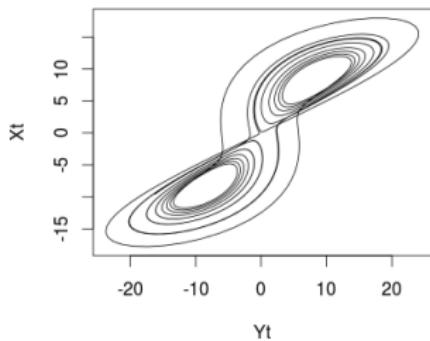
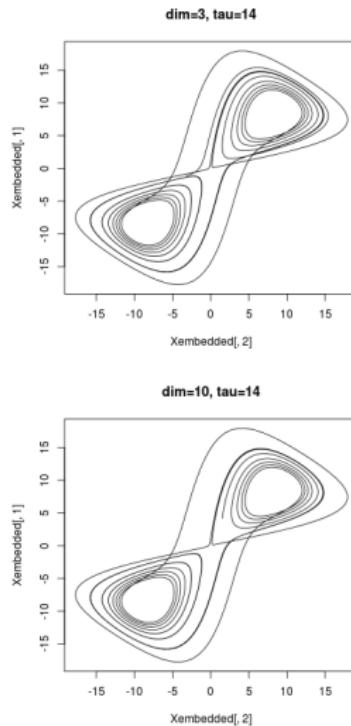


Figure : Original Manifold



Time-Delay Embedding Example

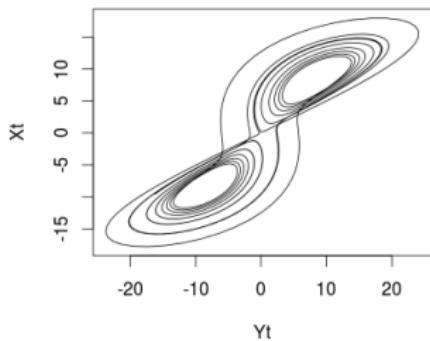
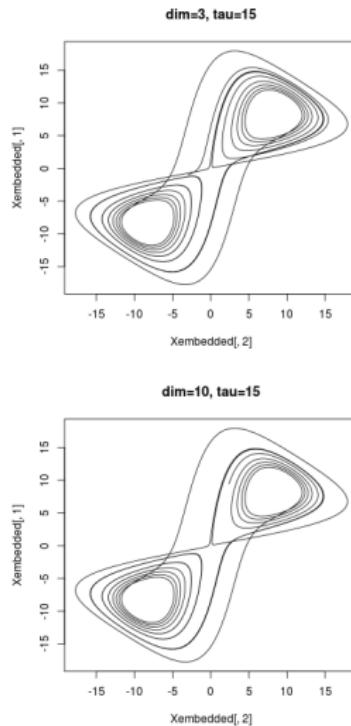


Figure : Original Manifold



Time-Delay Embedding Example

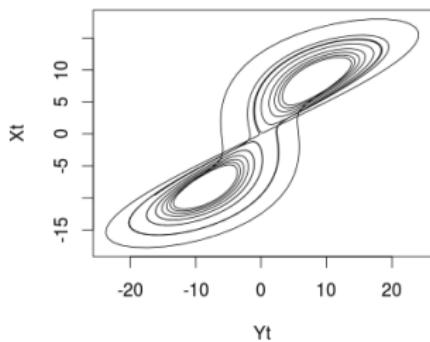
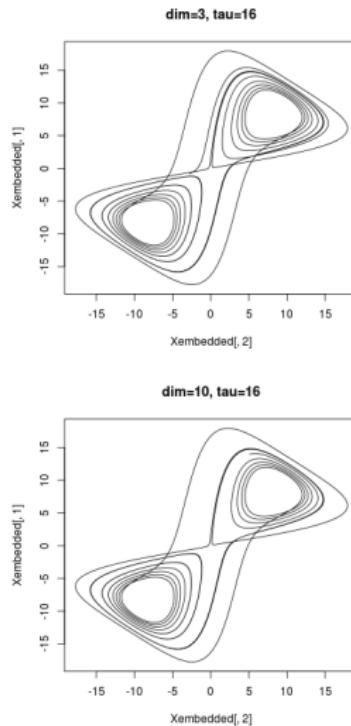


Figure : Original Manifold



Time-Delay Embedding Example

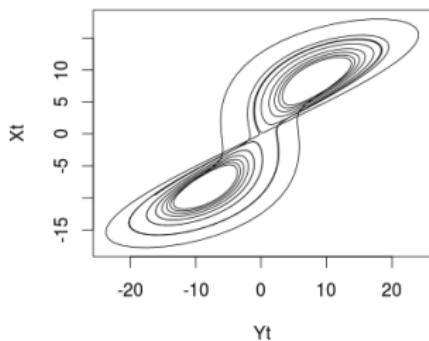
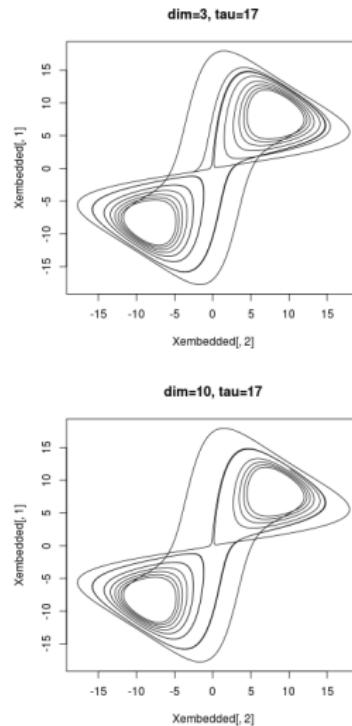


Figure : Original Manifold



Time-Delay Embedding Example

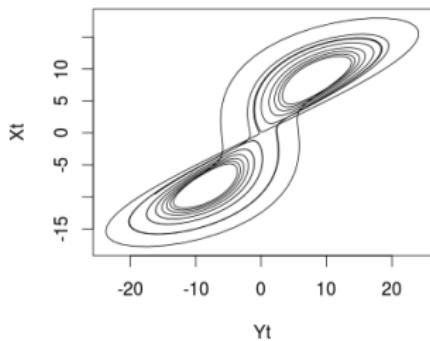
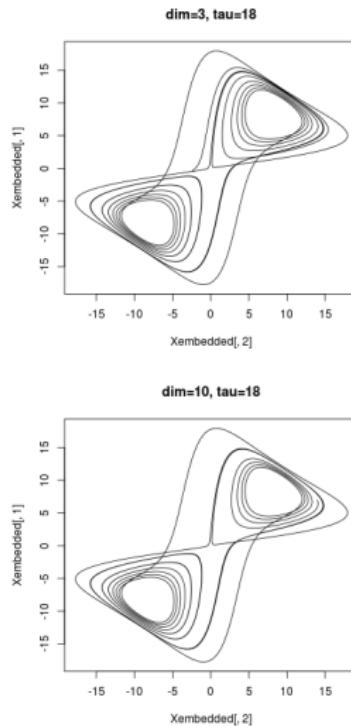


Figure : Original Manifold



Time-Delay Embedding Example

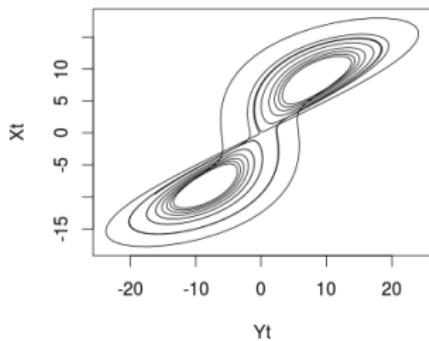
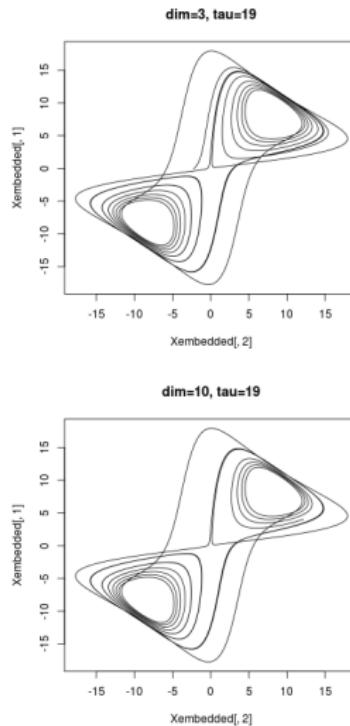


Figure : Original Manifold



Time-Delay Embedding Example

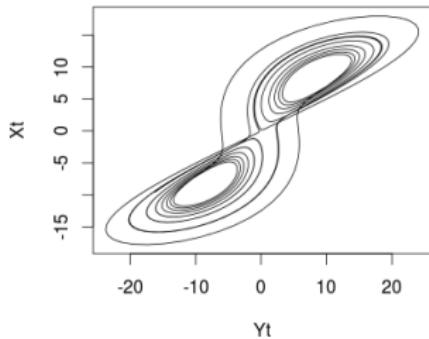


Figure : Original Manifold

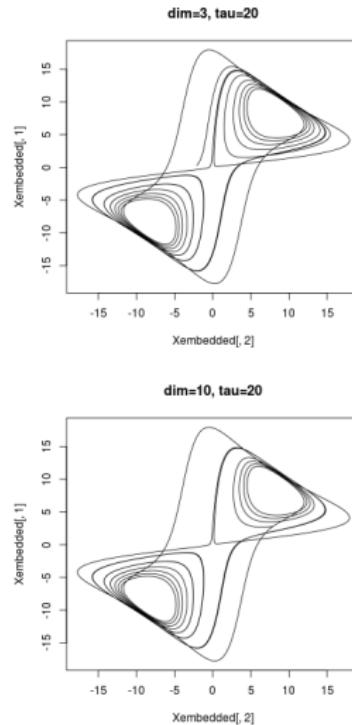
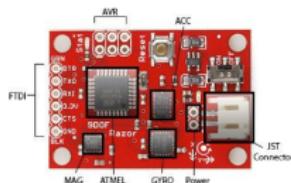


Figure : Reconstructed Manifold
Miguel Perez-Xochicale



DECIMUS Class in C++

9DOF Razor IMU



Accelerometer
X
Y
Z
[m]

Magnetometer
X
Y
Z
[m]

Gyroscope
X
Y
Z
[m]

Yaw
Pitch
Roll



DECIMUS Class in C++

*time-Delay Embedding theorem for
ReConstructing state spaces
Using Inertial Measurement Units*

Time-Delay Embedding Parameters

```
DataAnalysis.Set_SpaceReconstructionParameters(50,10,5); // (lengthwindowframe, dim, tau)
```

```
Decimus DataAnalysis
Kind: Variable definition
Defd: main.cpp:96 Show uses
DataAnalysis Object
```

Principal Component Analysis

```
mat A;
A= EmbeddedMatrix.t() * EmbeddedMatrix; // generate a symmetric matrix --- mat B = A.t() * A;

vec eigval_original, eigval;
mat eigvec_original, eigvec, transformedData;
eig_sym(eigval_original, eigvec, transformedData);
eig_sym(eigval_original, eigvec_original, A.t(), "dc"); // divide-and-conquer

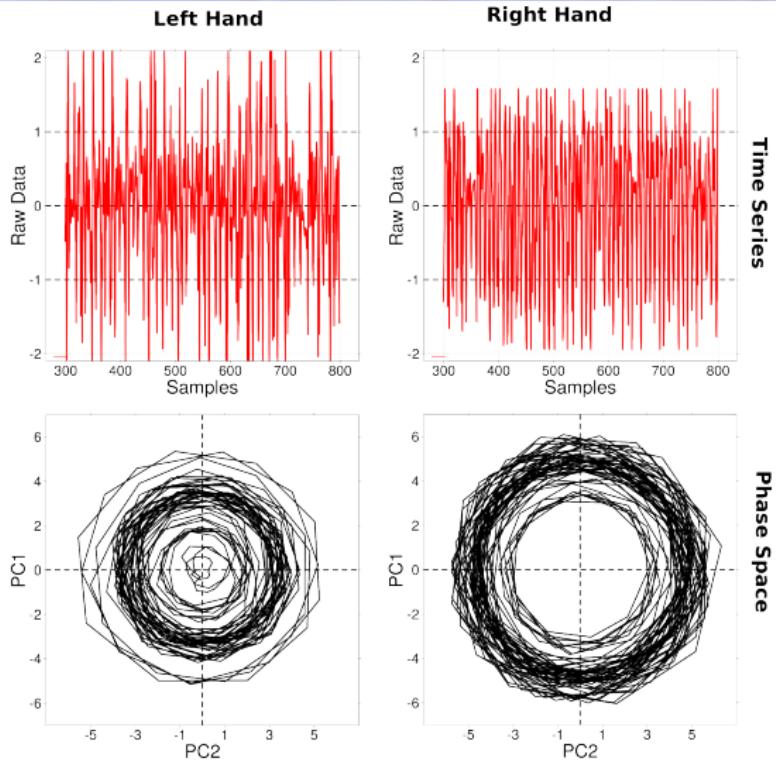
eigval = flipud(eigval_original);
eigvec = fliplr(eigvec_original);
transformedData = fliplr(eigvec_original).t() * EmbeddedMatrix.t();

cout << "DIY:eigenvalues \n" << eigval << endl;
cout << "DIY:eigenvectors \n" << eigvec << endl;
cout << "DIY:transformedData \n" << transformedData << endl;
```

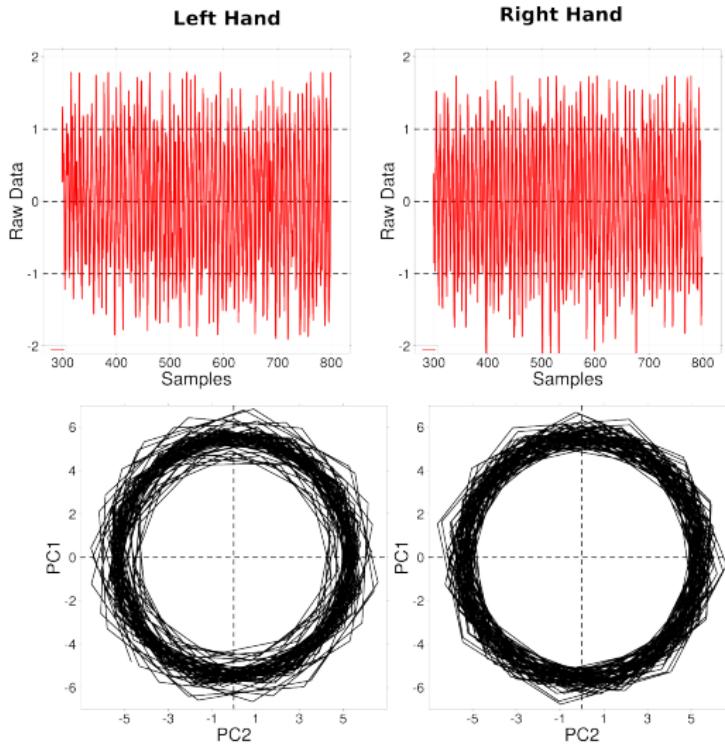
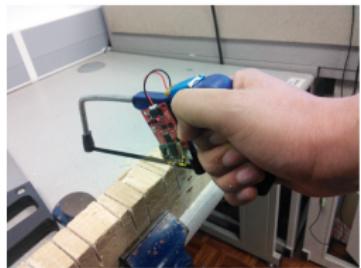
IMU, Axes and C++ Class



Sawing with a toolbox saw



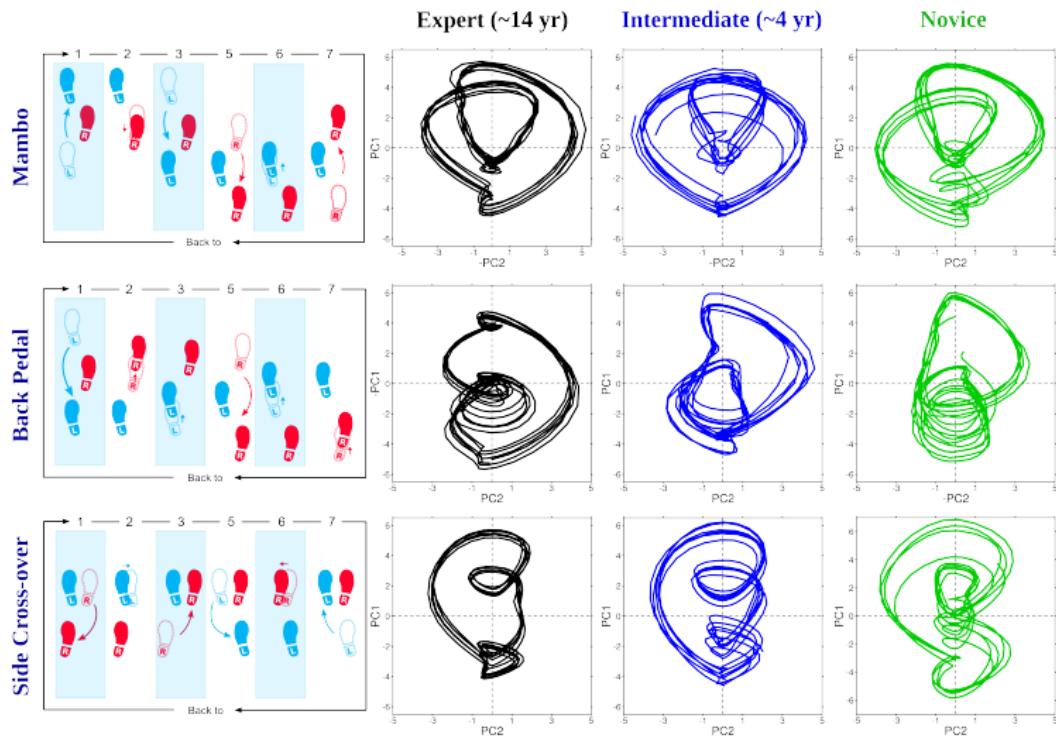
Sawing with a hack saw



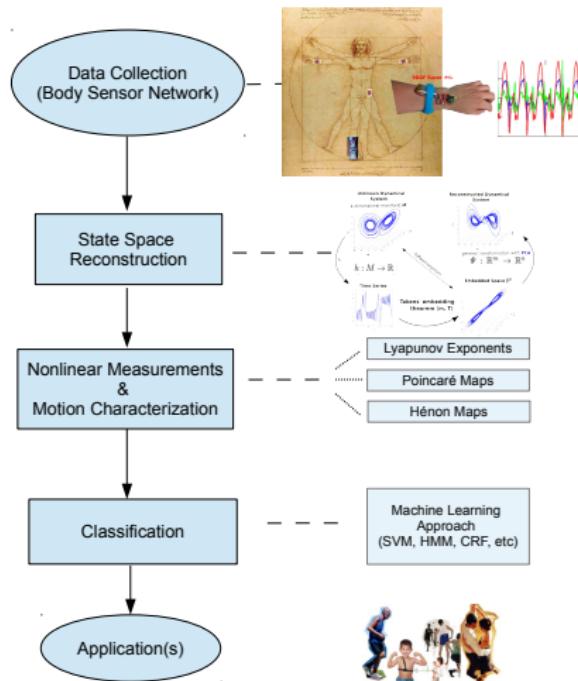
Quantifying Dexterity in Dance



Phase Spaces for Dancing



PhD Framework



QUESTIONS?

Miguel Perez-Xochicale

E-Mail: perez.xochicale@gmail.com

Homepage: <https://sites.google.com/site/perezxochicale/>



My own pictures are release under CC BY-NC 4.0

<http://creativecommons.org/licenses/by-nc/4.0/>

Give credits to: Miguel Perez-Xochicale

