Physics of Data

UNIVERSITÀ DEGLI STUDI DI PADOVA

A review on HAVOK analysis for non-linear systems

Università degli studi di Padova
Physics of Data
Laboratory of Computational Physics
A.A. 2021/2022

Allegri Luca

De Masi Michele

Grilli Alessandro

Sbarbati Riccardo

Introduction

- HAVOK analysis for chaotic system:
    - Linearization of non-linear systems
    - Prediction of the dynamic behaviour of a system
- HAVOK critical issues
- Application of Neural Network
- Different integrable systems where HAVOK has a good response:
    - Lorenz attractor
    - Financial system
    - Magnetic Pendulum
- Open questions

HAVOK - Hankel Alternative View Of Koopman

A universal, data-driven decomposition of chaos as an intermittently forced linear system.

Combination of delay embedding and Koopman theory.

nature
COMMUNICATIONS

ARTICLE

DOI: 10.1038/s41467-017-00030-8 OPEN

Chaos as an intermittently forced linear system

Steven L. Brunton[1], Bingni W. Brunton[2], Joshua L. Proctor[3], Eurika Kaiser[1] & J. Nathan Kutz[4]

VOL. 17, 1931 *MATHEMATICS: B. O. KOOPMAN* 315

HAMILTONIAN SYSTEMS AND TRANSFORMATIONS IN HILBERT SPACE

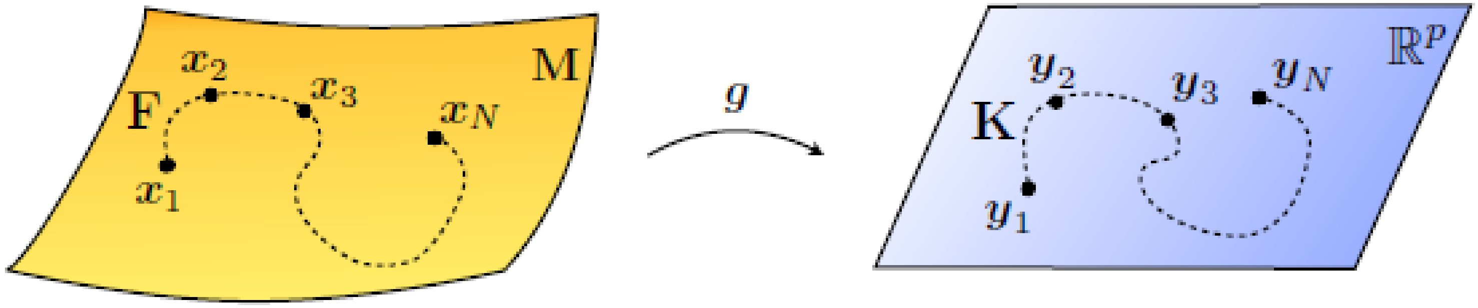
BY B. O. KOOPMAN

DEPARTMENT OF MATHEMATICS, COLUMBIA UNIVERSITY

Communicated March 23, 1931

Linear representation of non-linear dynamics

Transformation of a non-linear dynamic equation in the coordinate space in a linear equation in an infinite dimensional Hilbert space.

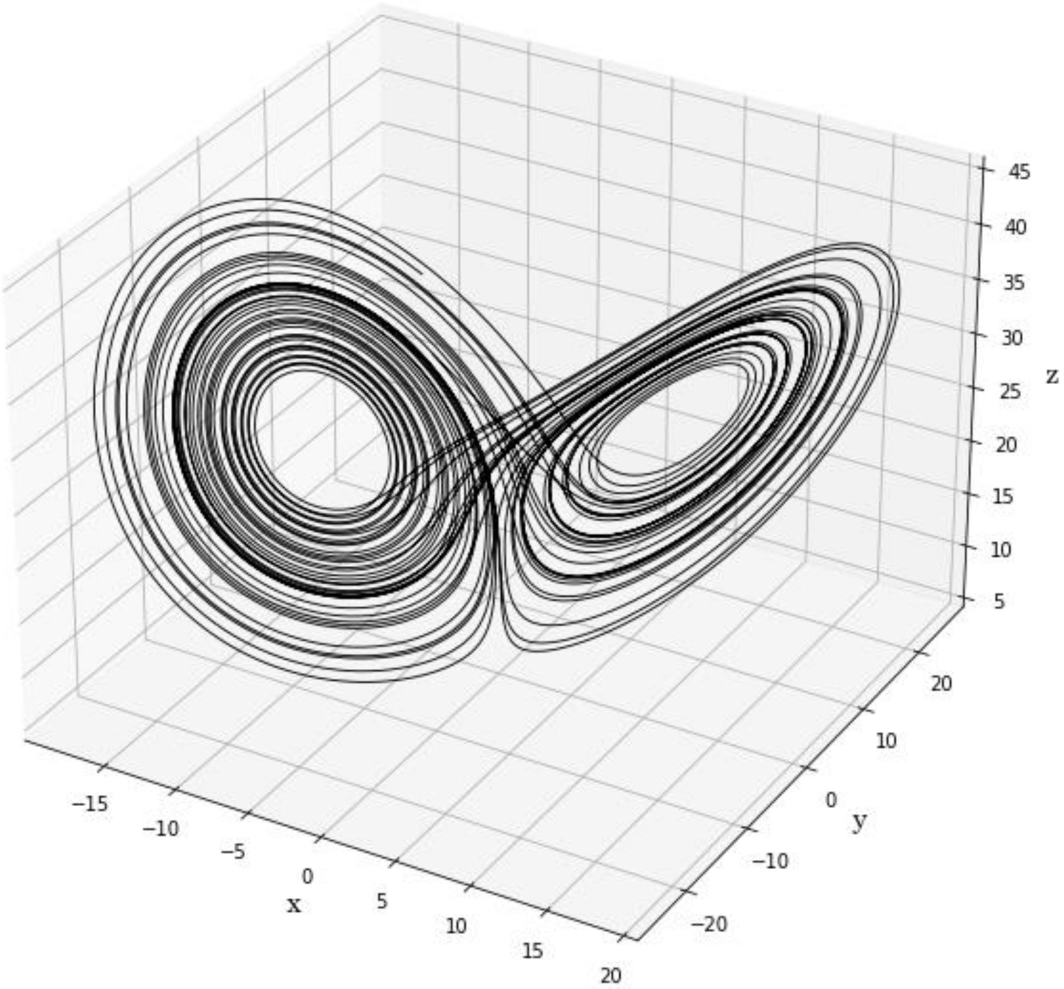


$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)),$$

$$\mathcal{K}g \triangleq g \circ \mathbf{F} \quad \Rightarrow \quad \mathcal{K}g(\mathbf{x}_k) = g(\mathbf{x}_{k+1}).$$

HAVOK Analysis

The most common chaotic system: *Lorenz attractor*



$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x),$$

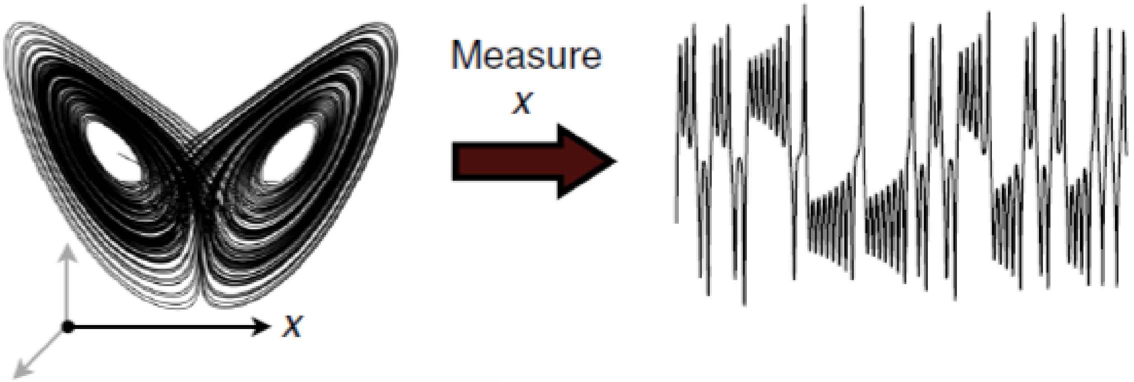
$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y,$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z.$$

$\beta = 8/3$ $\sigma = 10$ $\rho = 28$

HAVOK Analysis

From the 3D representation of the system we extract a time series from the most meaningful coordinate.

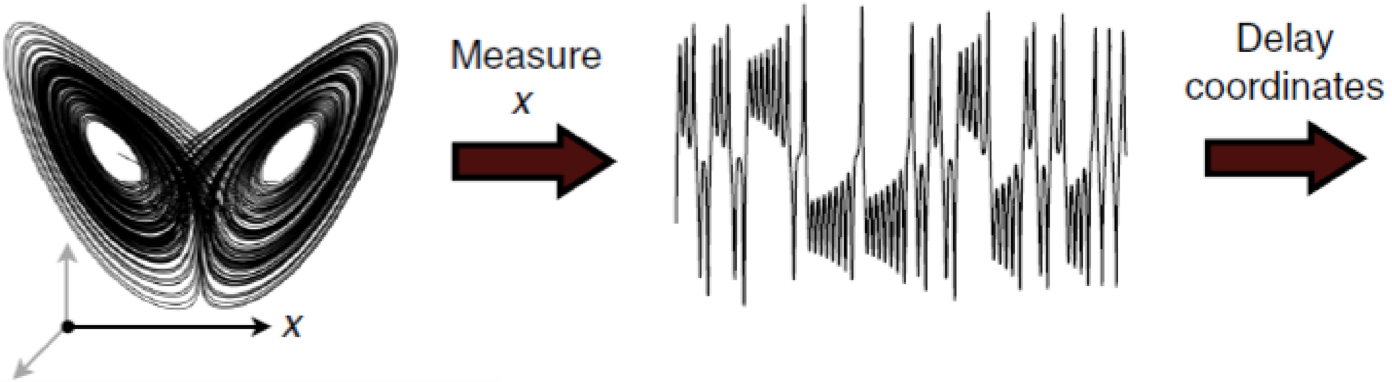


HAVOK Analysis

From the 3D representation of the system we extract a time series from the most meaningful coordinate.

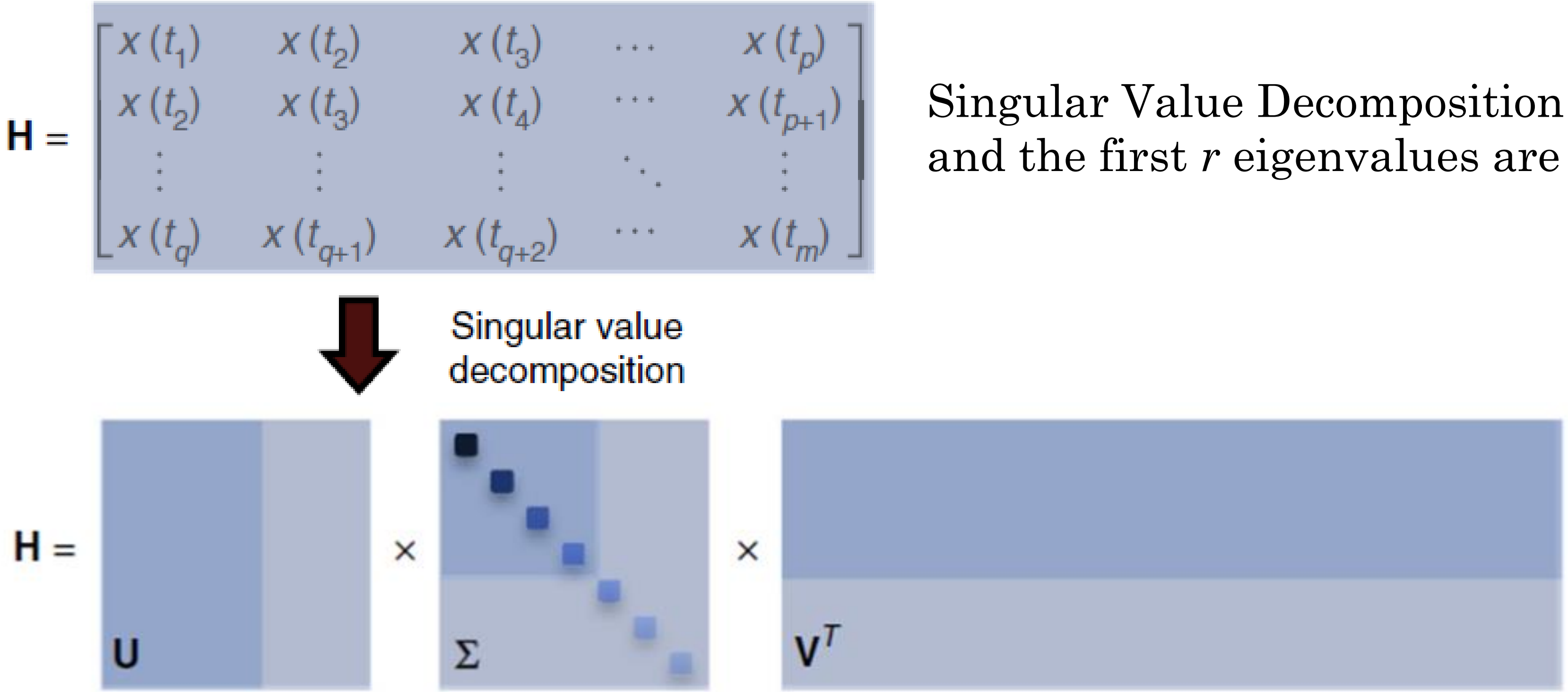
We set the time step dt and the dimension of a sliding window l to build a Hankel matrix from the time series.

```
l = 100
dt = 0.001
H = np.zeros((l,len(x_ts)-l+1))

for i in range(l):
    H[i,:] = x_ts[i:len(x_ts)-l+i+1]
```



$$H = \begin{bmatrix} x(t_1) & x(t_2) & x(t_3) & \cdots & x(t_p) \\ x(t_2) & x(t_3) & x(t_4) & \cdots & x(t_{p+1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(t_q) & x(t_{q+1}) & x(t_{q+2}) & \cdots & x(t_m) \end{bmatrix}$$

HAVOK Analysis

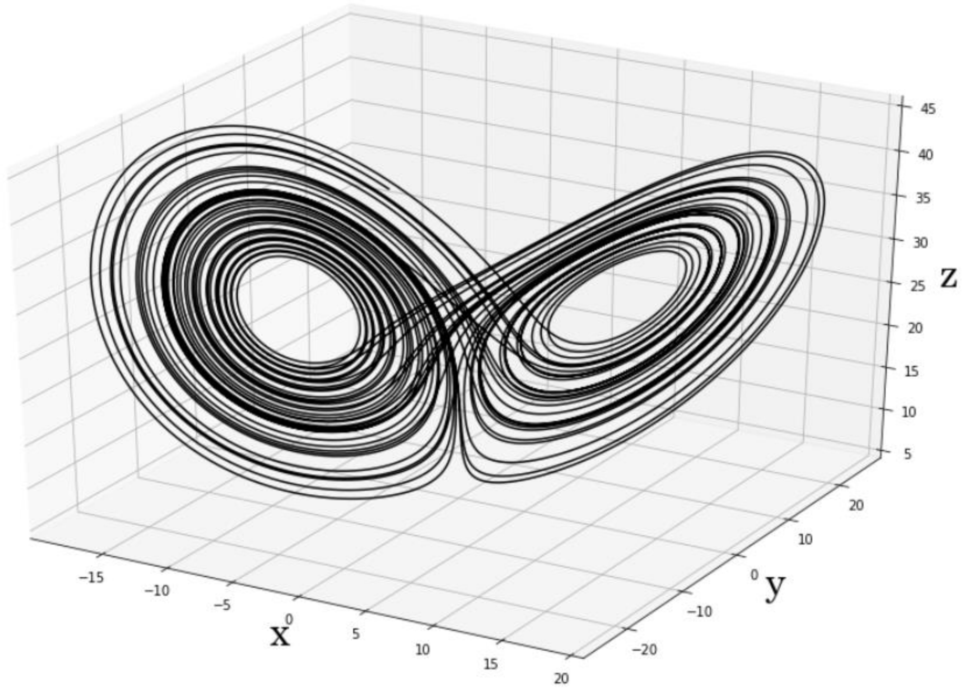


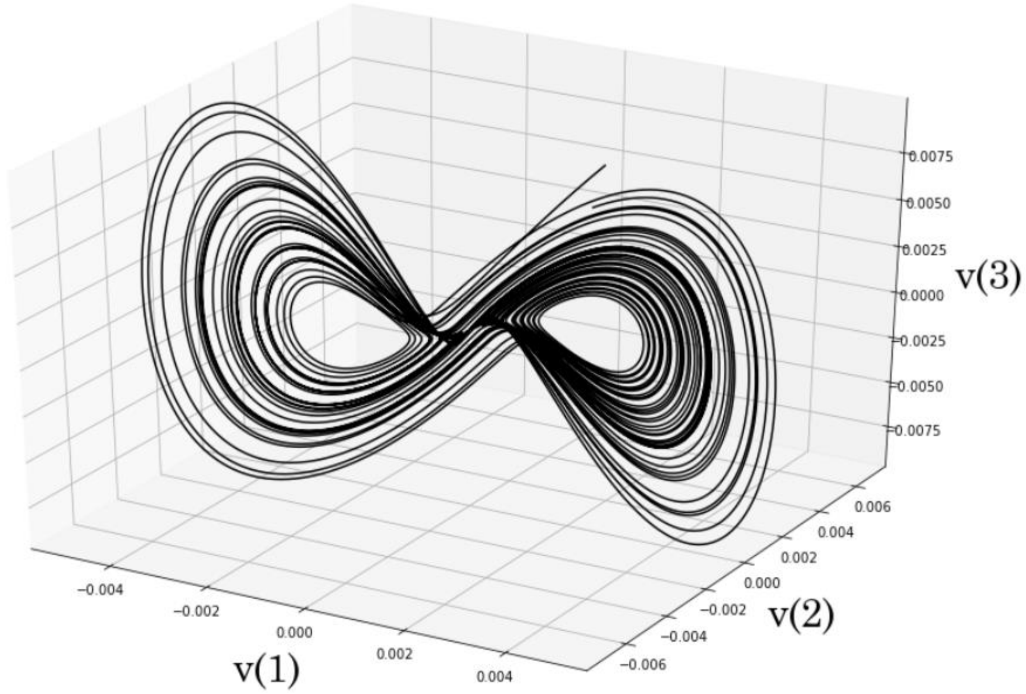
Singular Value Decomposition is applied and the first r eigenvalues are selected.

```python
U, spectrum, Vt = la.svd(H,full_matrices=False,compute_uv=True)
```

HAVOK Analysis

The first three columns of V create an embedded attractor.





Original Lorenz attractor

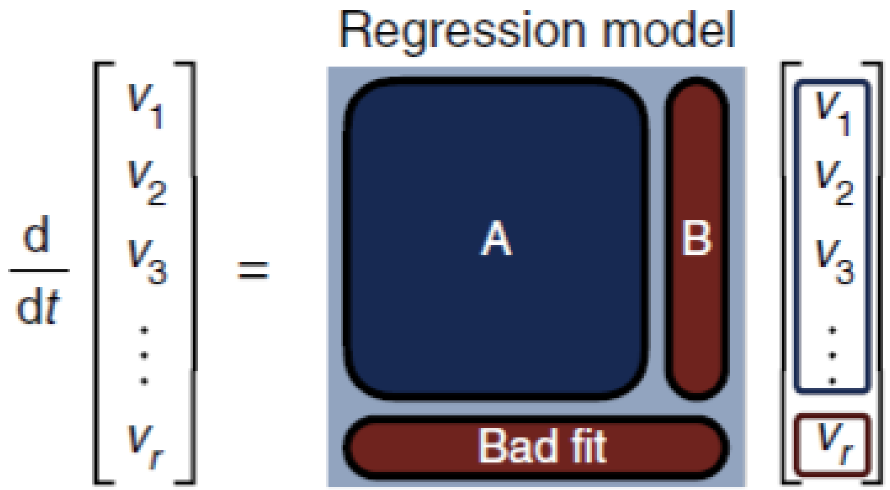
Embedded attractor

HAVOK Analysis

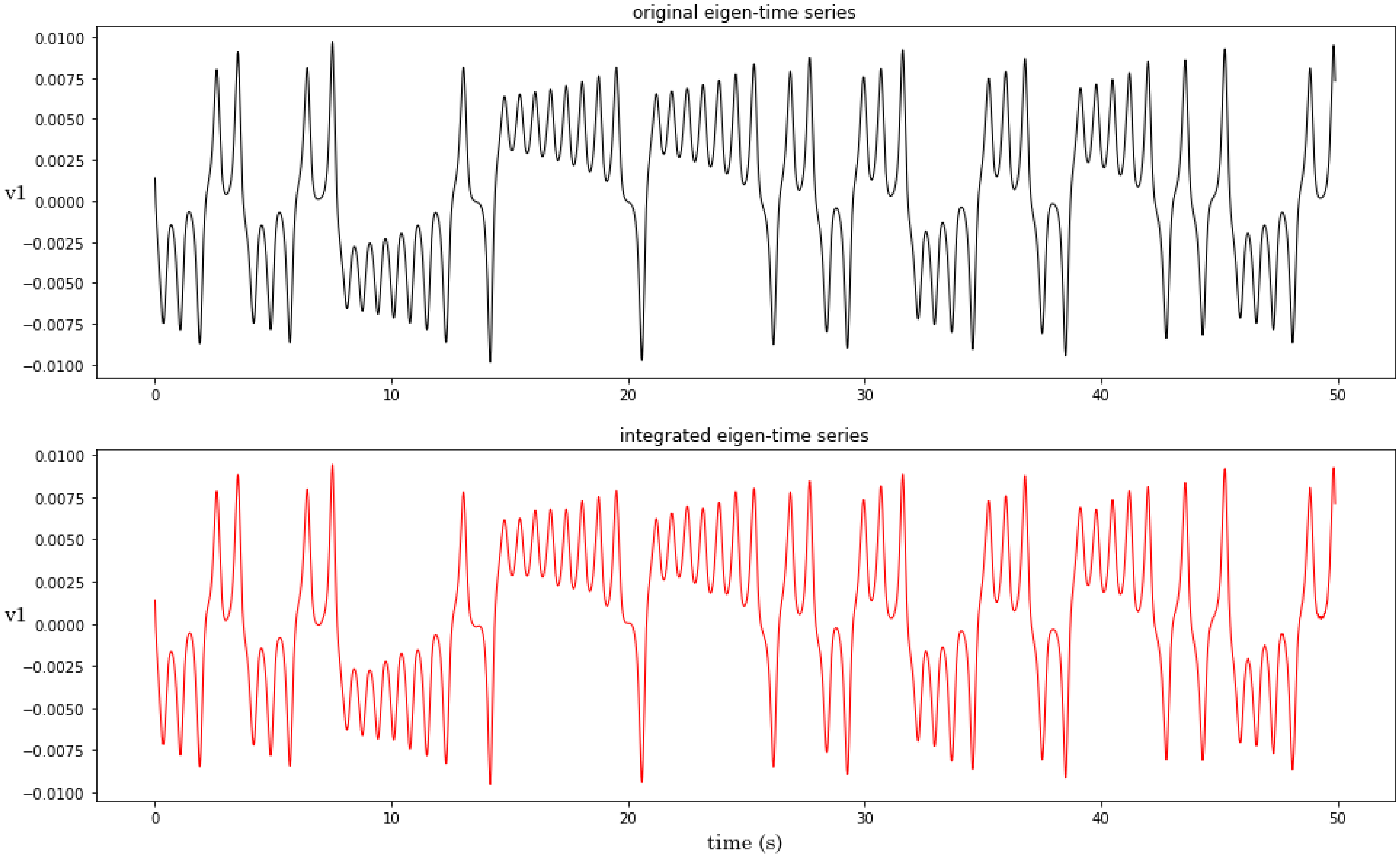
These *r* eigenvectors are used to train a linear model with a sparse regression.

The first *r-1* vectors are the linear coordinates of our model, $\mathbf{v_r}$ is a forcing term that influences the dynamics.

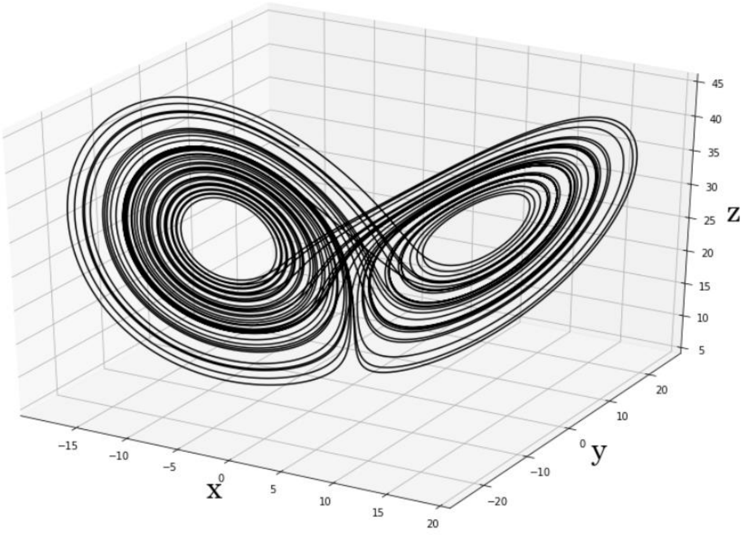
$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}(t) = \mathbf{A}\mathbf{v}(t) + \mathbf{B}v_r(t)$$

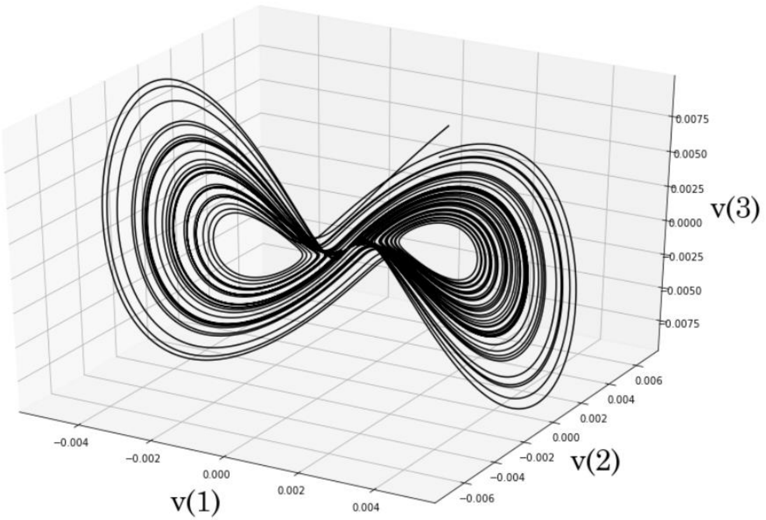


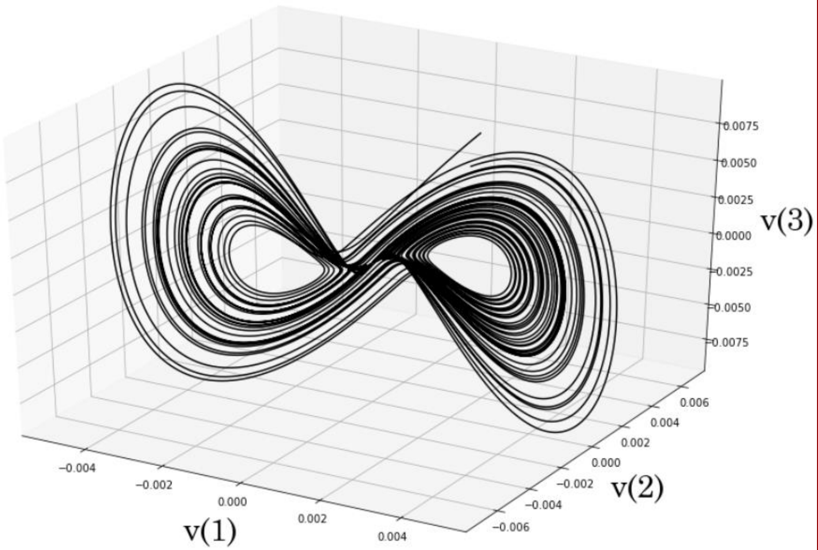
HAVOK Analysis



HAVOK Analysis







Original Lorenz system

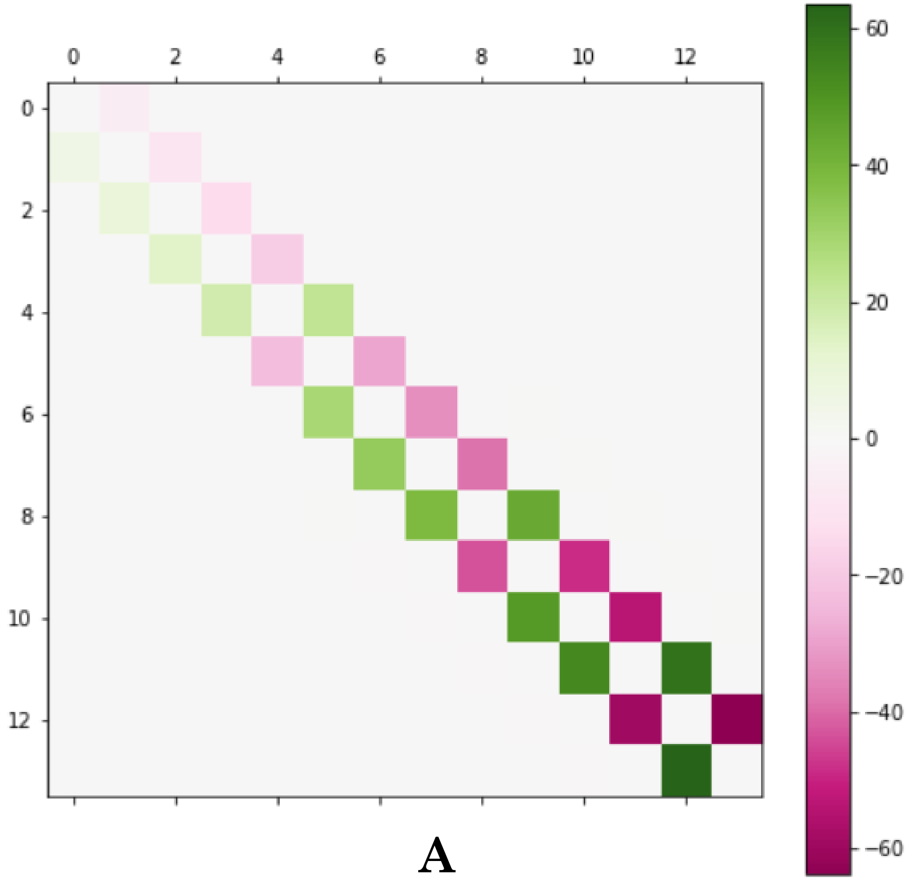
Embedded attractor

Integrated system

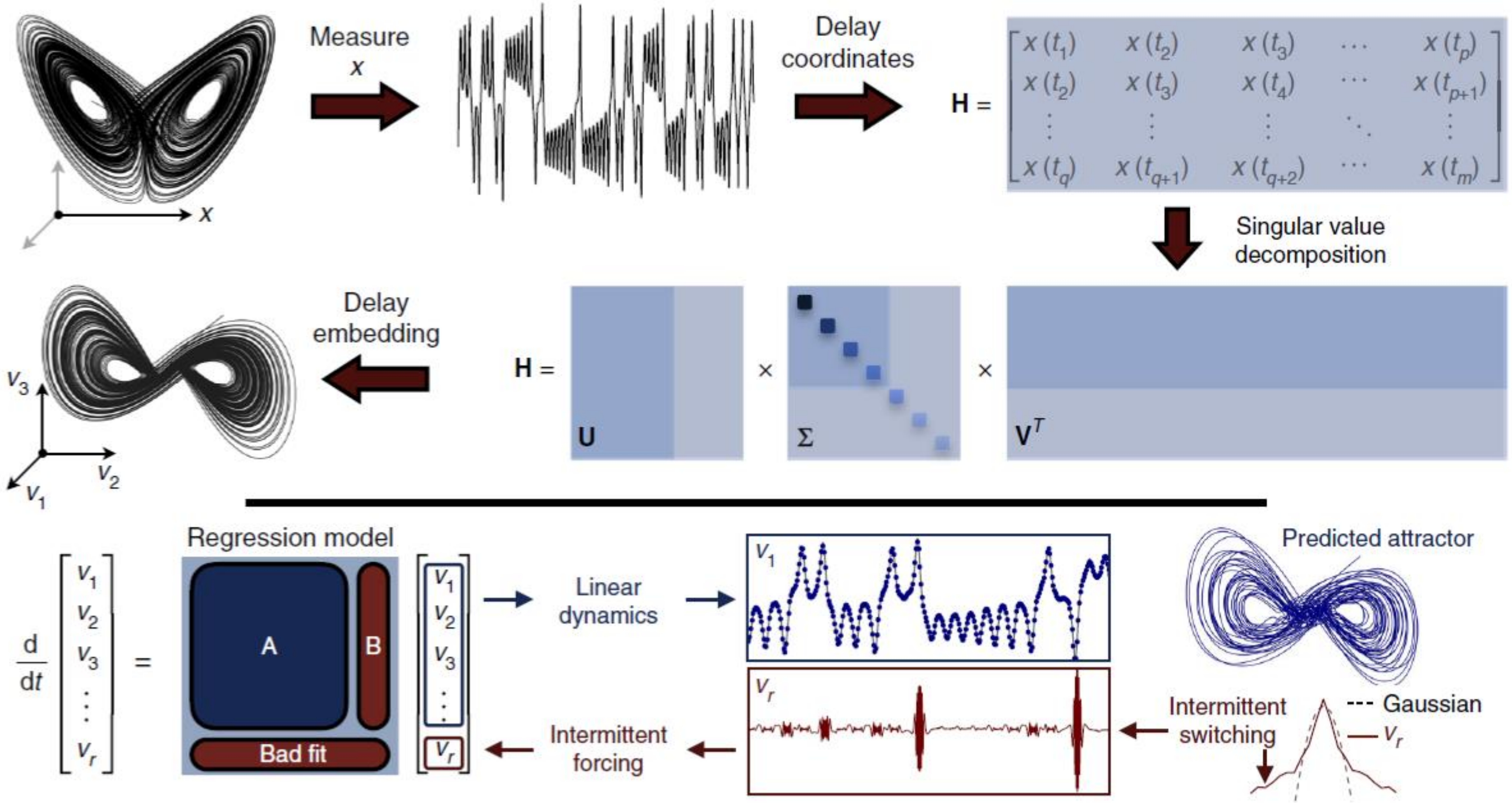
HAVOK Analysis

A curious pattern is created.

```
differentiation_method = ps.FiniteDifference(order=2)
feature_library = ps.PolynomialLibrary(degree=1)
optimizer = ps.STLSQ(threshold=0.02)

model = ps.SINDy(
    differentiation_method=differentiation_method,
    feature_library=feature_library,
    optimizer=optimizer
)

model.fit(v,t=time)

M = model.coefficients()
M = M[:,1:r+1]
A = M[:r-1,:r-1]
B = M[:r-1,r-1]
```



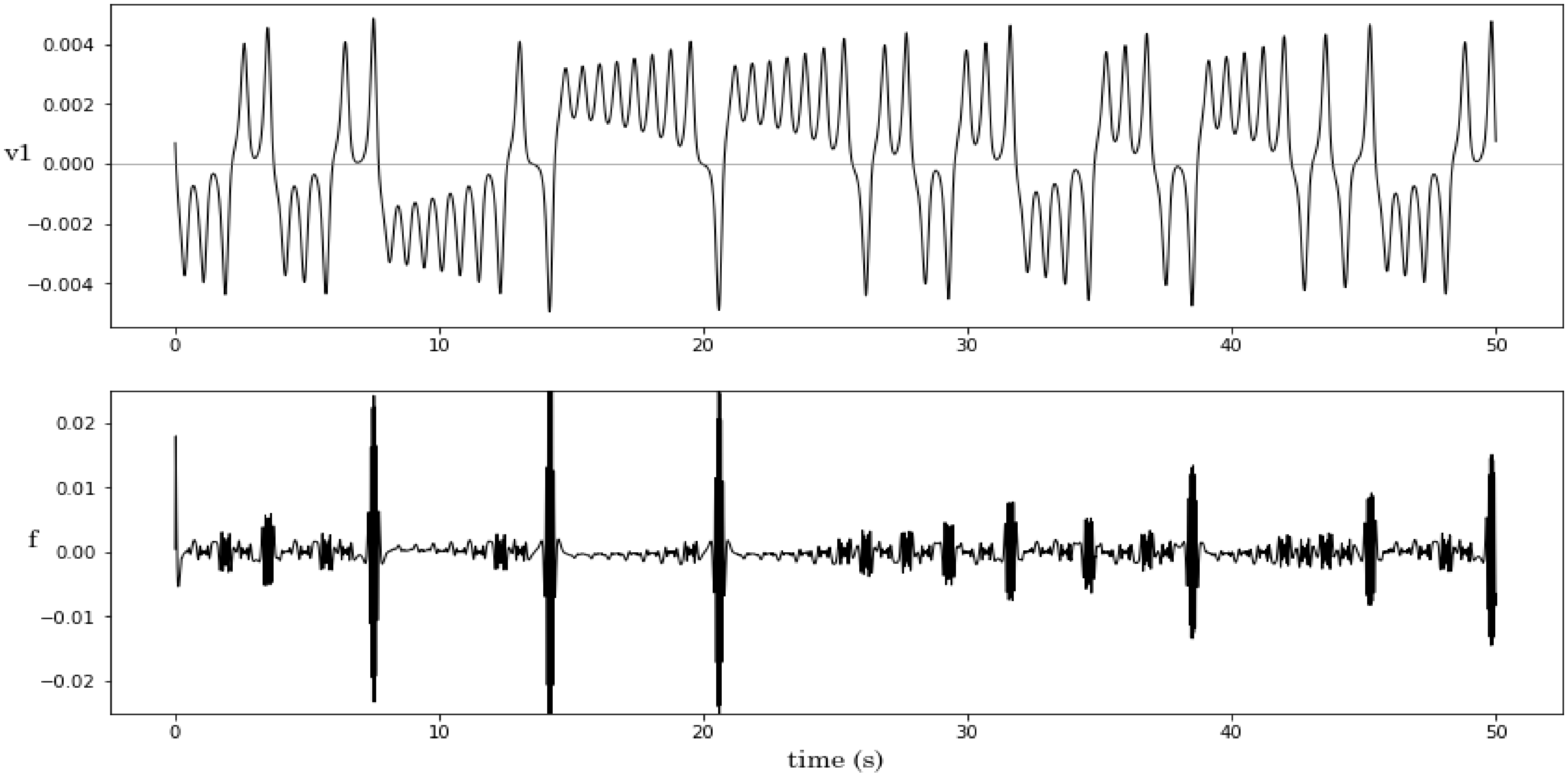
A



Measure
x
Delay
coordinates
H =
$\begin{bmatrix} x(t_1) & x(t_2) & x(t_3) & \cdots & x(t_p) \\ x(t_2) & x(t_3) & x(t_4) & \cdots & x(t_{p+1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(t_q) & x(t_{q+1}) & x(t_{q+2}) & \cdots & x(t_m) \end{bmatrix}$
Singular value
decomposition
Delay
embedding
H =
U
×
Σ
×
V^T
v_3
v_2
v_1
Regression model
$\frac{d}{dt}$
v_1
v_2
v_3
v_r
=
A
B
Bad fit
Linear
dynamics
Predicted attractor
Intermittent
forcing
Intermittent
switching
--- Gaussian
— v_r

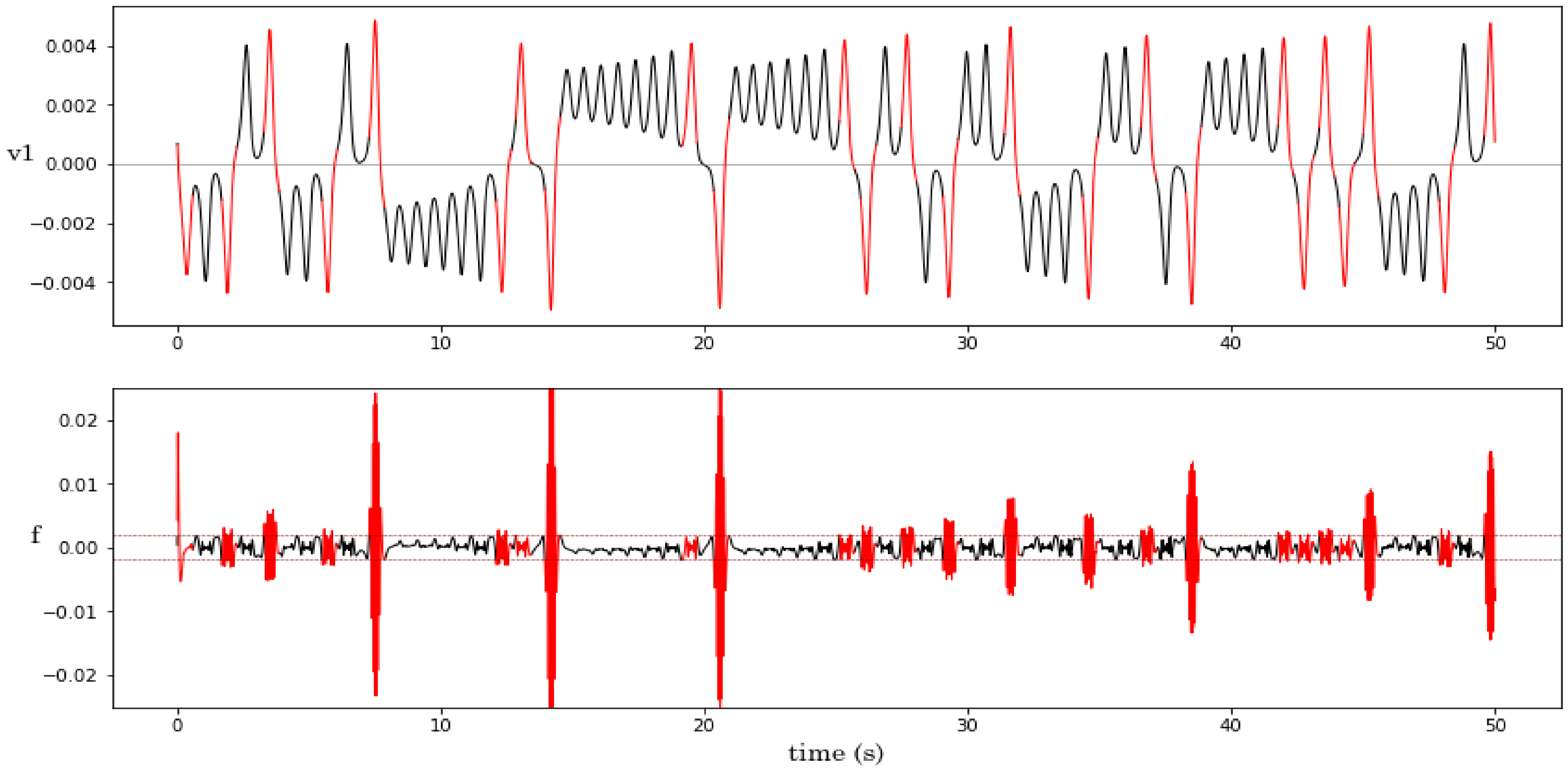
Role of the forcing term

There seems to be a correlation between peaks of the forcing term *f* and the chaotic switches of the system.



Role of the forcing term

This is verified by setting a threshold on *f*.



Training model for online prediction

- The knowledge of the forcing term would allow to predict the lobe switching for new incoming data.
- But we cannot have access to this information in advance.
- High peaks of v_r anticipate the lobe switching.
- Is it possible to gather this information before the actual change of the system?

Training model for online prediction

- v_r comes from applying SVD on the Hankel matrix, but this is not doable for real time computation.

- However, we can compute U from the SVD of a restricted dataset and then infer the subsequent principal coordinates point by point with a convolution:

$$V^T \propto \vec{x}^T \cdot U$$

```
Vt_conv[:,i] = np.dot(x_test[i:i+q],U1)
```

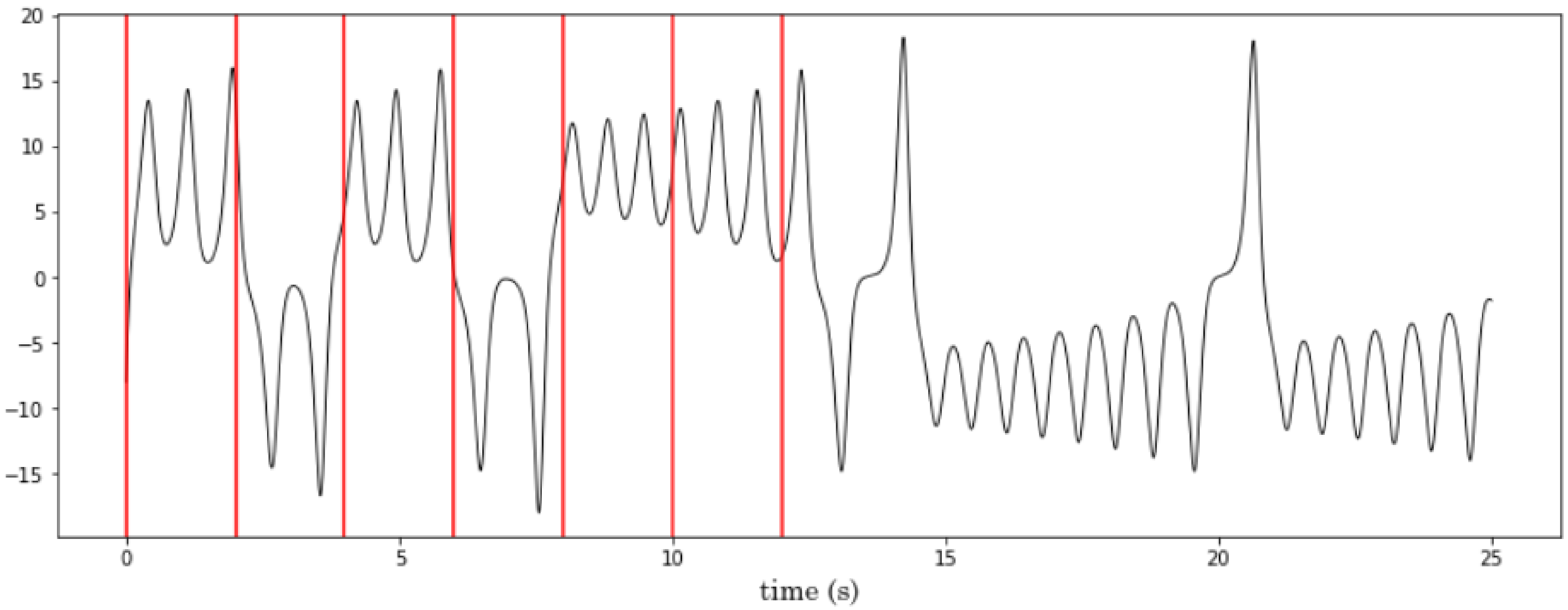
- The delay of this operation depends on the length of the sliding window and on the size of the timestep.

Critical issues – l & dt parameter estimation

When building the Hankel matrix, it is important to choose an appropriate size for the sliding window, such that it can represent the characteristics of the system without losing too much information.

The sampling timestep should not be wider than the typical period.

Excessively large sampling window does not represent the system!



Critical issues – l & dt parameter estimation

The parameter choice must be done by hand, such that the model does not diverge too early but does not have too much intrinsic delay in the computation of the forcing term.

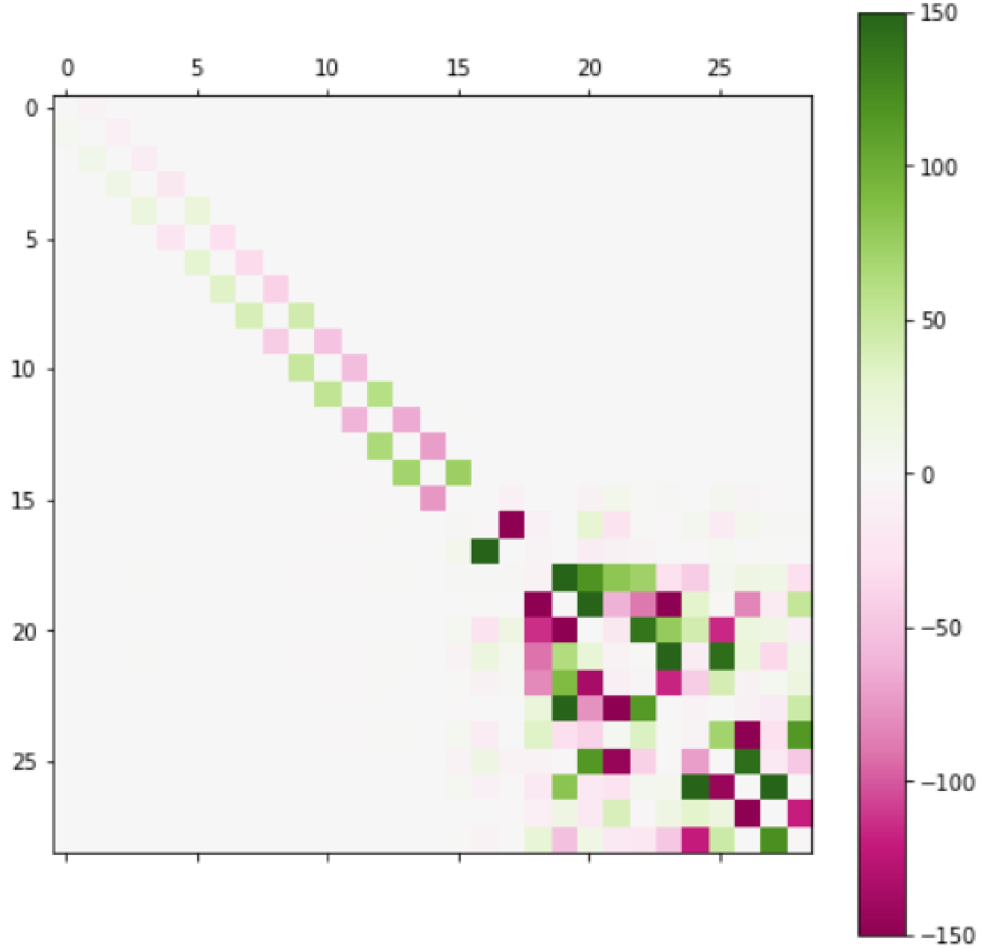
For the Lorenz system:

- $l = 100$
- $dt = 0.001s$

This allows a good model accuracy and equals to a delay of $0.1s$ for the online prediction, which is shorter than the period of a single loop (around $0.7s$).

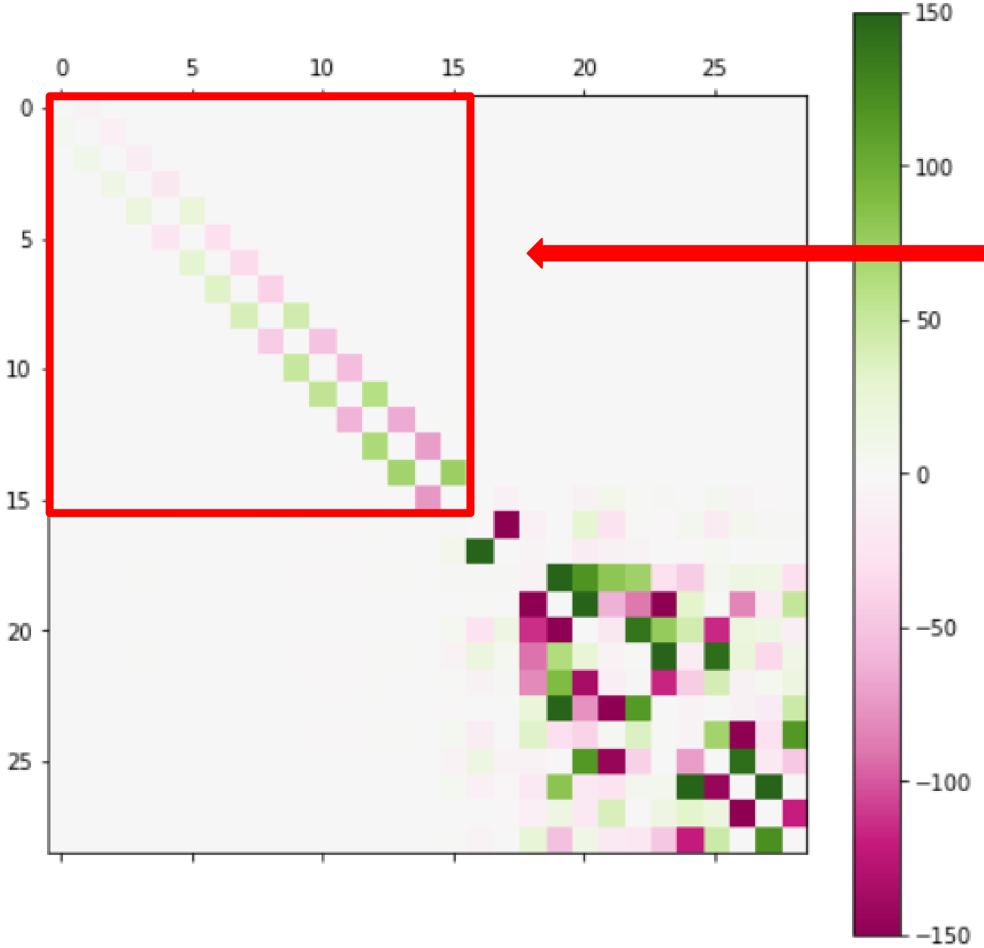
Critical issues - *r* parameter estimation

The number of principal vectors chosen strongly affects the efficiency of the model and is an operation that must be done by hand in a backward process, based on the pattern of matrix A.



Critical issues - *r* parameter estimation

The number of principal vectors chosen strongly affects the efficiency of the model and is an operation that must be done by hand in a backward process, based on the pattern of matrix A.

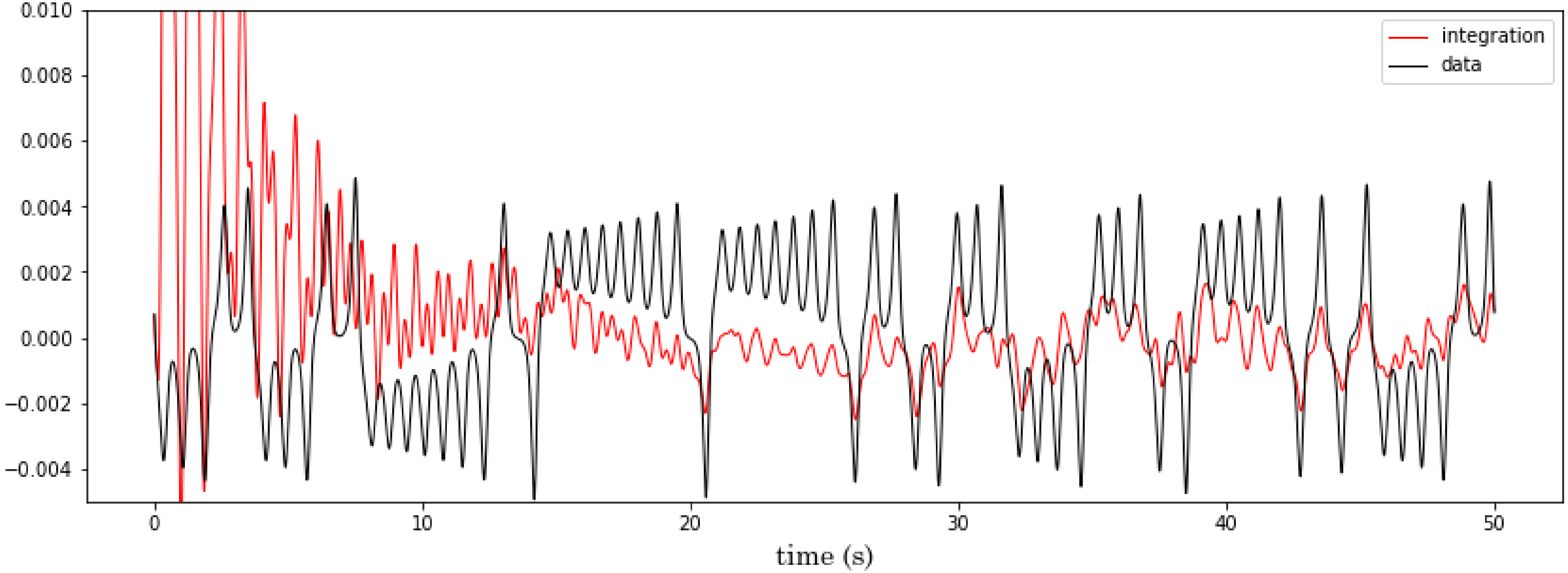


For the Lorenz attractor, the maximum number of principal vectors is $r = 15$

Critical issues - *r* parameter estimation

The number of principal vectors chosen strongly affects the efficiency of the model and is an operation that must be done by hand in a backward process, based on the pattern of matrix A.

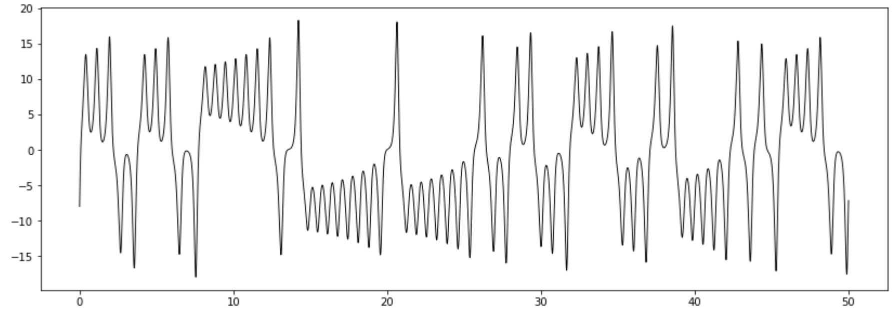
Increasing too much the dimensionality leads to a model that does not reproduce the dynamics of the system.



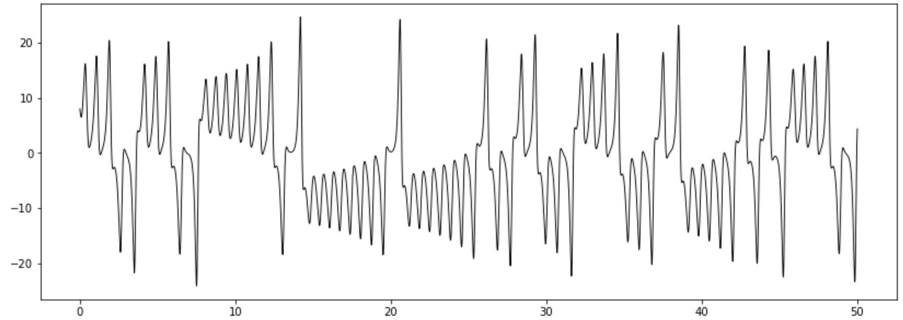
Critical issues – coherent time series

For a system with dimension > 1, it is important to look at the axis projection that gives you the most interesting time series, information-wise.

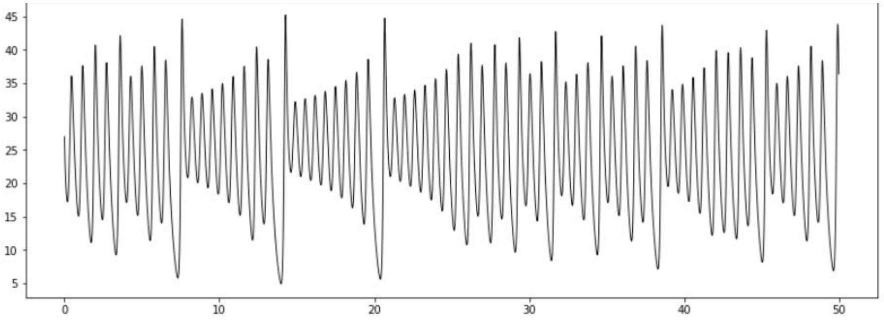
X

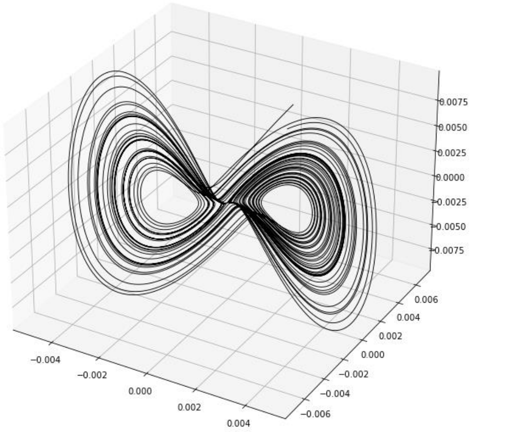


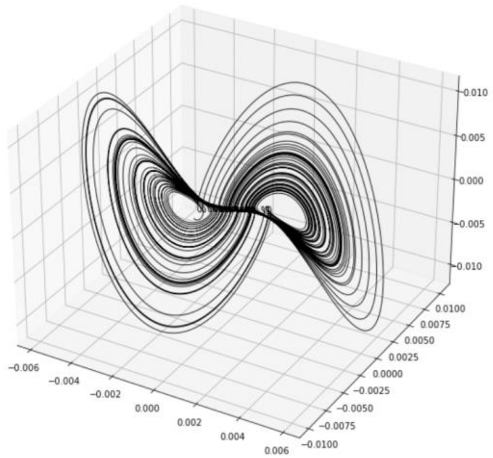
Y

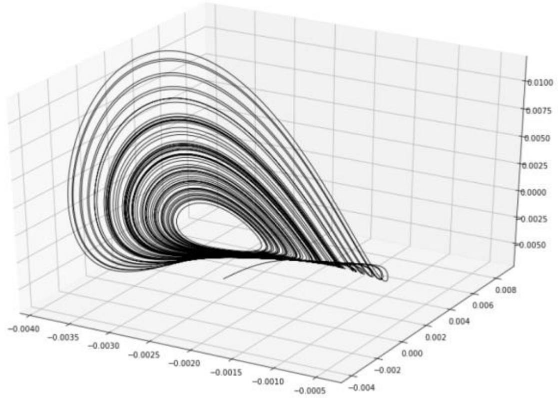


Z









Critical issues - behaviour prediction

By now, the prediction of the lobe switching is done setting a threshold on the forcing term.

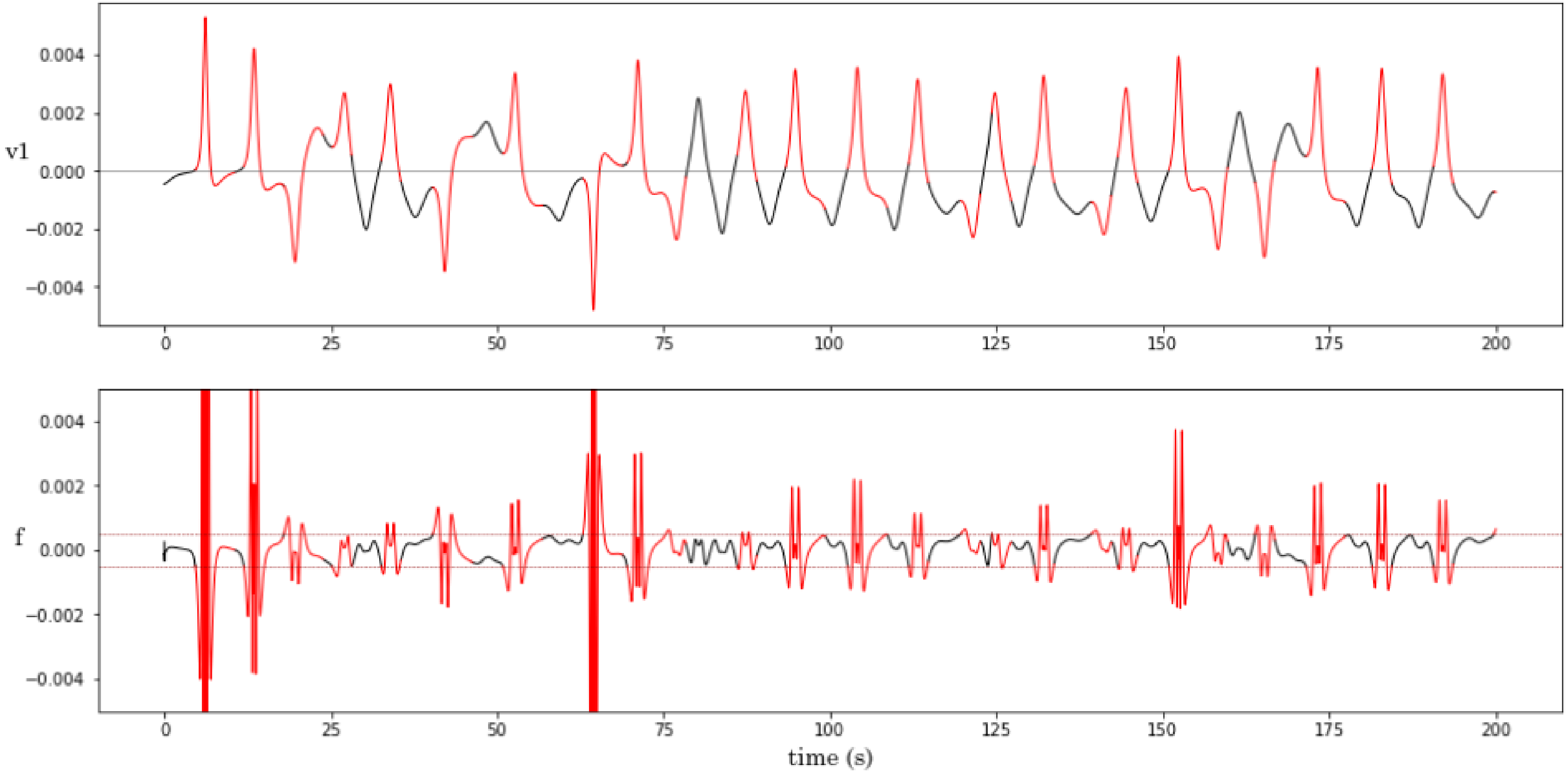
The accuracy of this method for the Lorenz system is around 99%, with a false positive rate of about 2.5%.

But the Lorenz system is one of the simplest examples of chaotic dynamics.

This percentages may not necessarily be true for other more complex chaotic models.

Critical issues - behaviour prediction

Sometimes the effect of the forcing term activation is not straightforward.

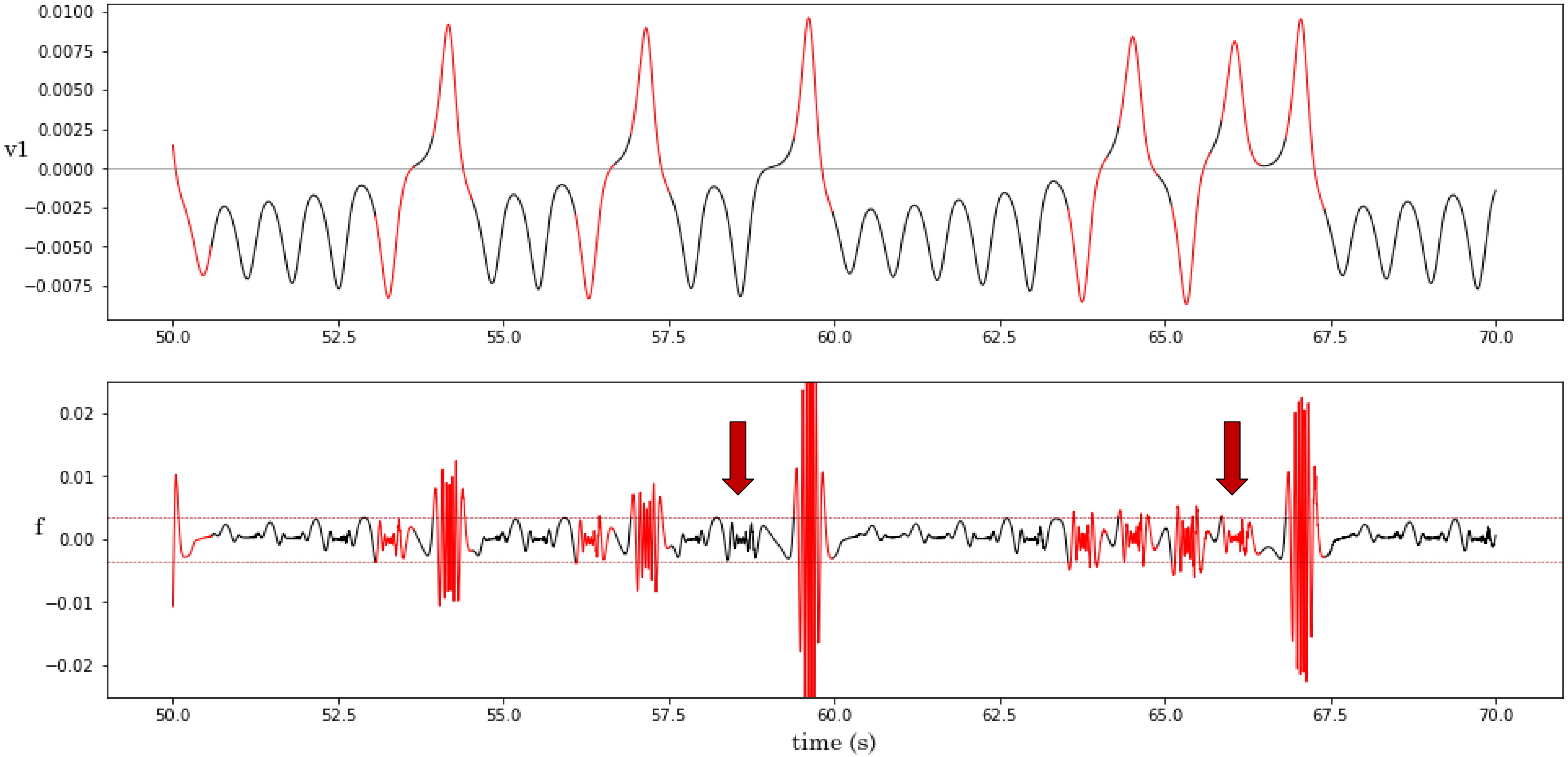


Chaotic model for a financial system.

l = *100*
dt = *0.001*
r = 9

Critical issues - behaviour prediction

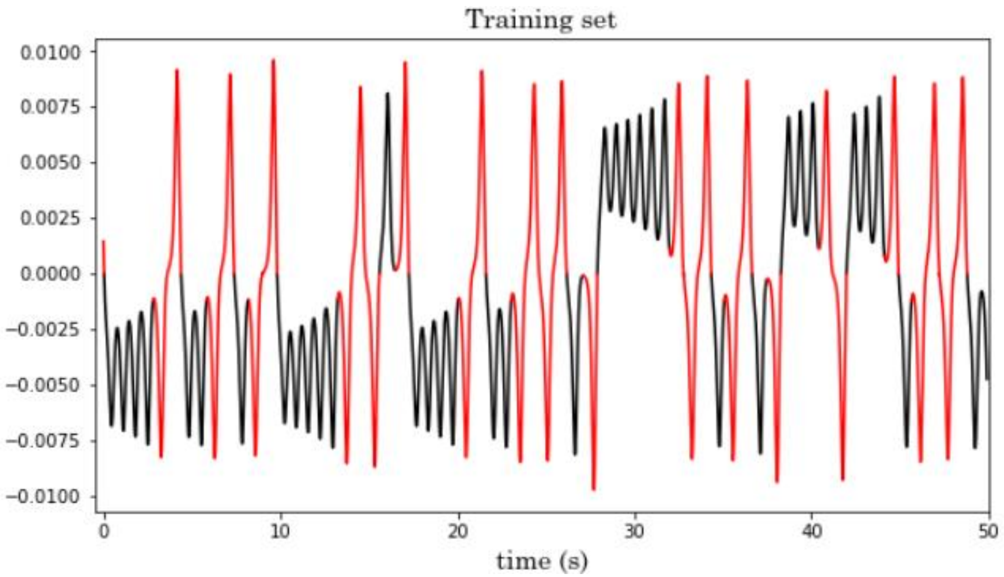
Even the Lorenz system shows ambiguity in some instances, suggesting that the lobe switching must not depend only on the forcing term.



Laboratory of Computational Physics

Neural Network for lobe switching

A Neural Network is implemented to avoid the threshold issues.

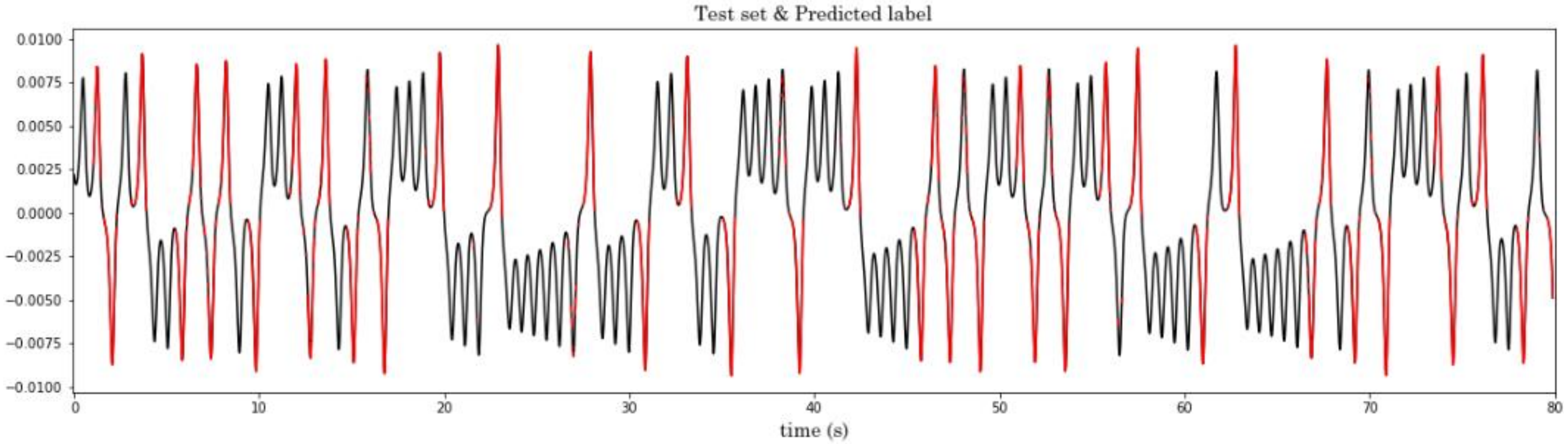


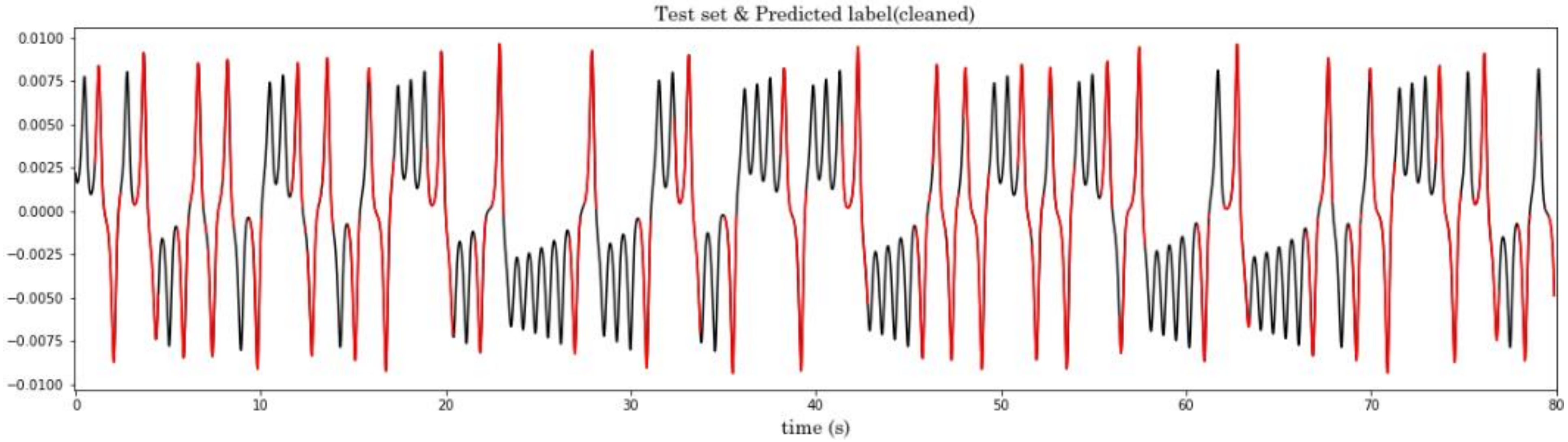

```
model = Sequential()
model.add(Dense(r,input_shape = (r,), activation = 'relu'))
model.add(Dense(50, activation = 'relu'))
model.add(Dense(1, activation = 'sigmoid'))

compile_model = model.compile(loss = 'binary_crossentropy',
                              optimizer = 'Nadam',
                              metrics = ['accuracy'])

fit = model.fit(x_train_NN, x_lab_NN, epochs = 50,
                batch_size = 50, verbose = 2)
```

The evaluated output of the NN, gives the following results:

```
49900/49900 [==============================] - 6s 116us/sample - loss: 0.1920 - accuracy: 0.9289
```

NN: prediction of the lobe switching





Laboratory of Computational Physics

Other systems – Financial

The HAVOK analysis has been performed on a chaotic model of a financial system proposed by Huang and Li.

$x(t)$ = interest rate

$y(t)$ = investment demand

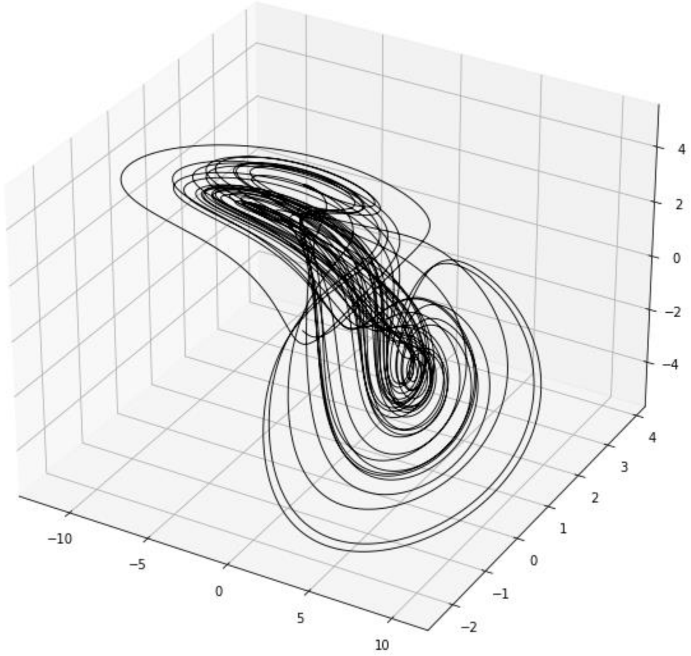
$z(t)$ = price index

$a = 0.3$ $b = 0.02$ $c = 1$ $r = 1$
$s = 0.1$ $p = 0.05$ $g = 1.2$ $\beta = 1$

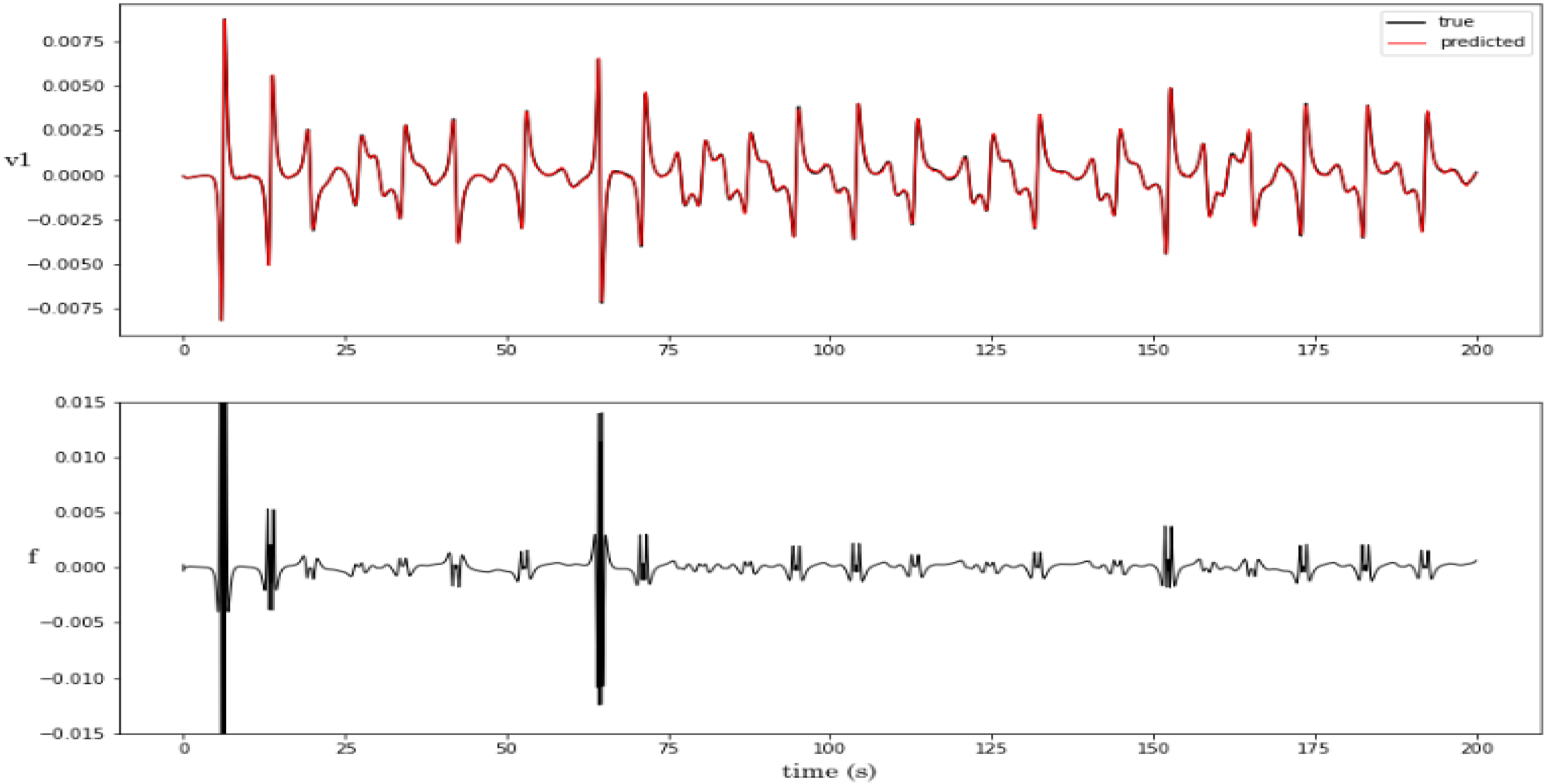
$$\frac{dx}{dt} = gz + (y - a)x,$$

$$\frac{dy}{dt} = -by^2 - sx^2 + r,$$

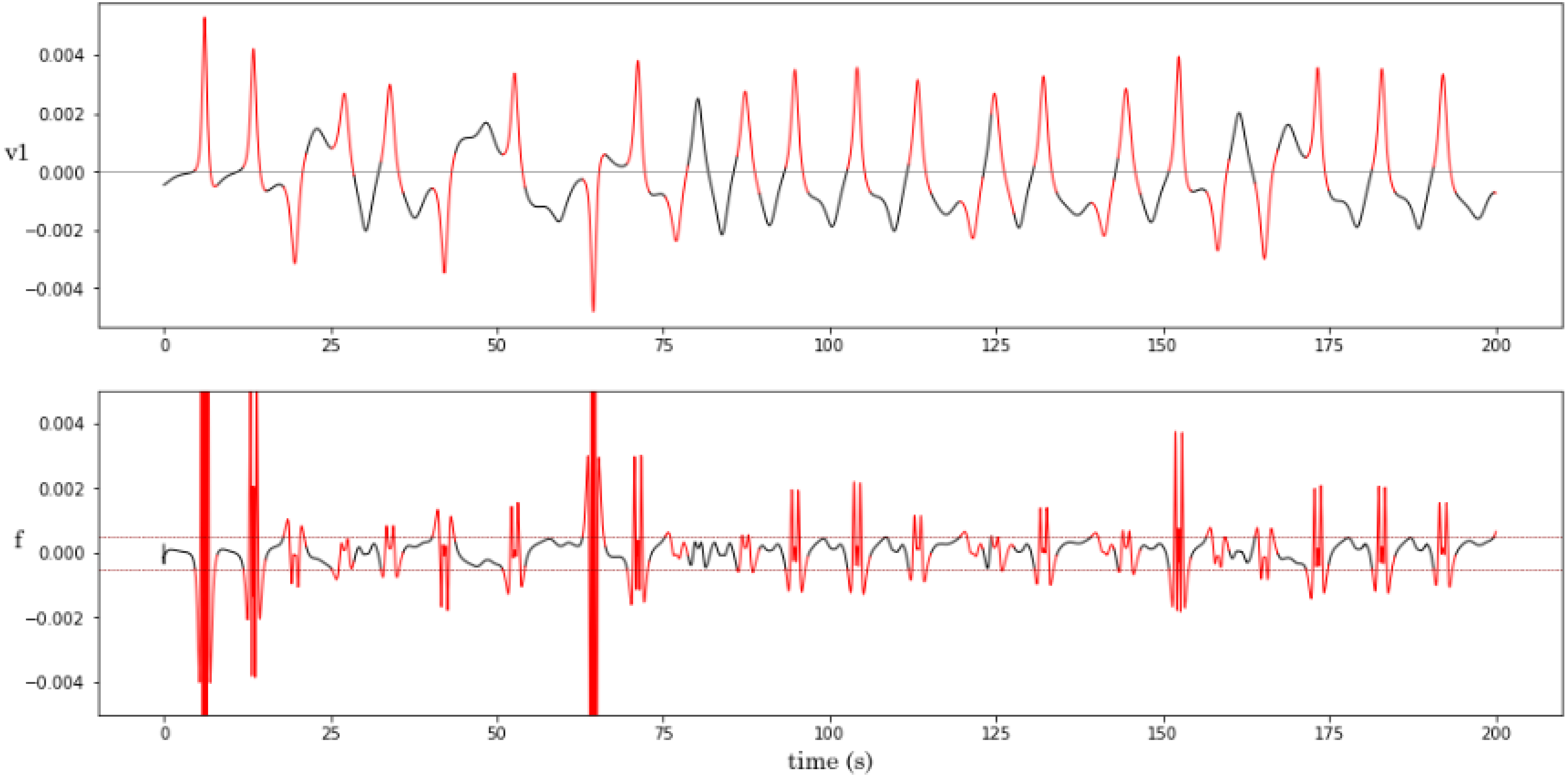
$$\frac{dz}{dt} = -cz - \beta x - py,$$



Other systems – Financial system

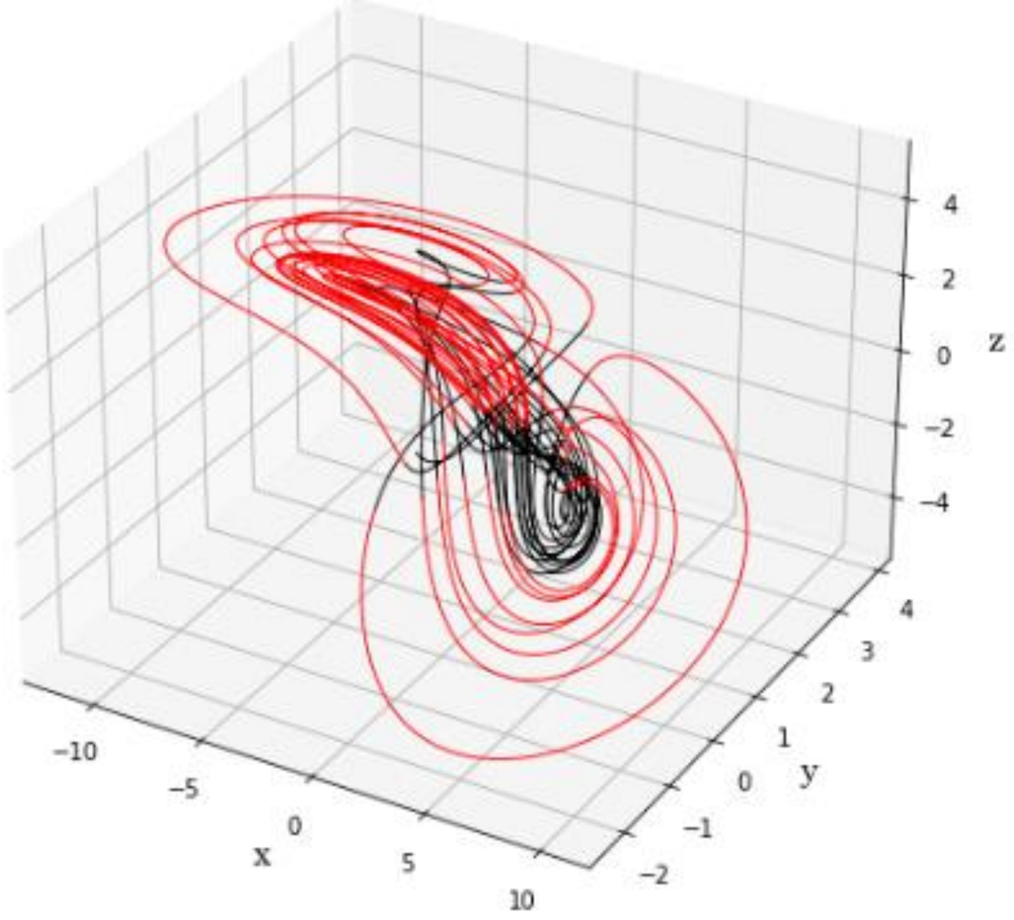


Other systems – Financial system

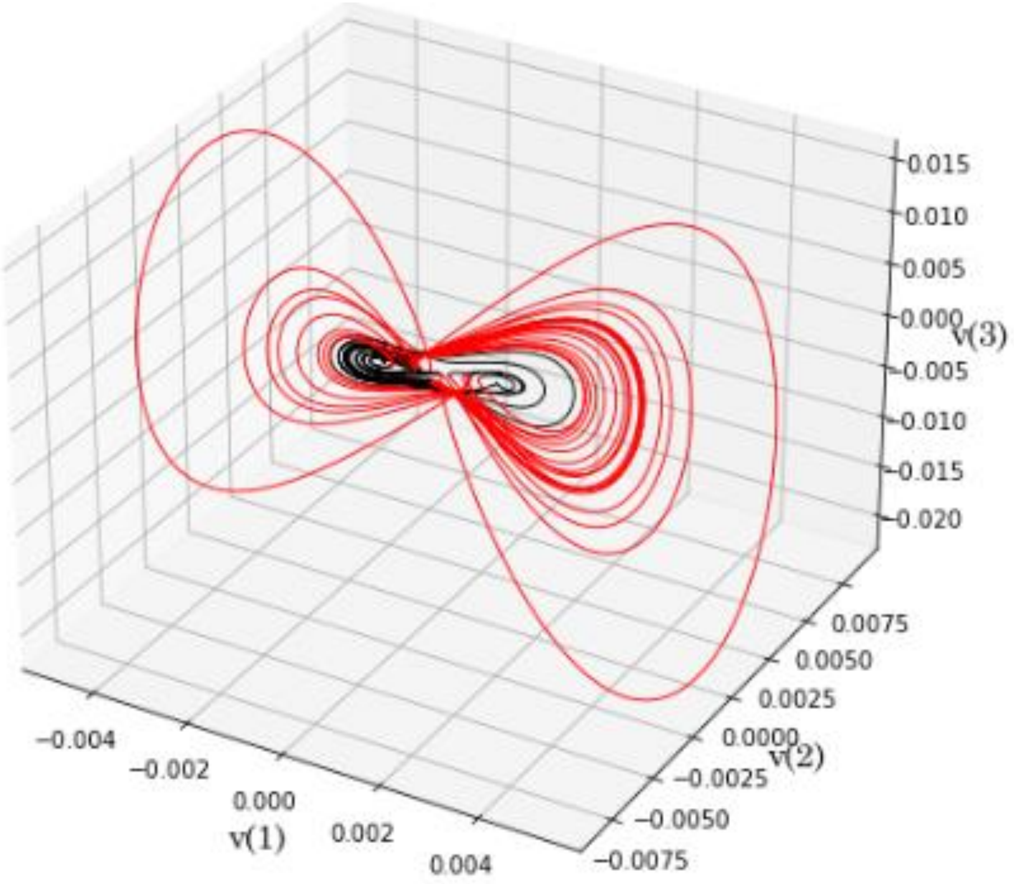


Laboratory of Computational Physics

Other systems – Financial system



Real attractor

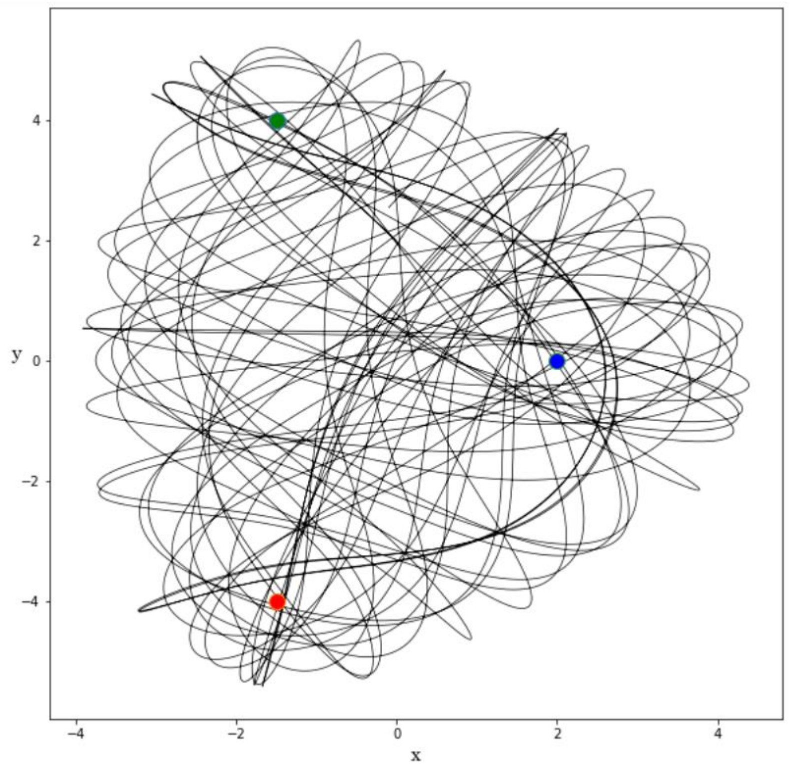


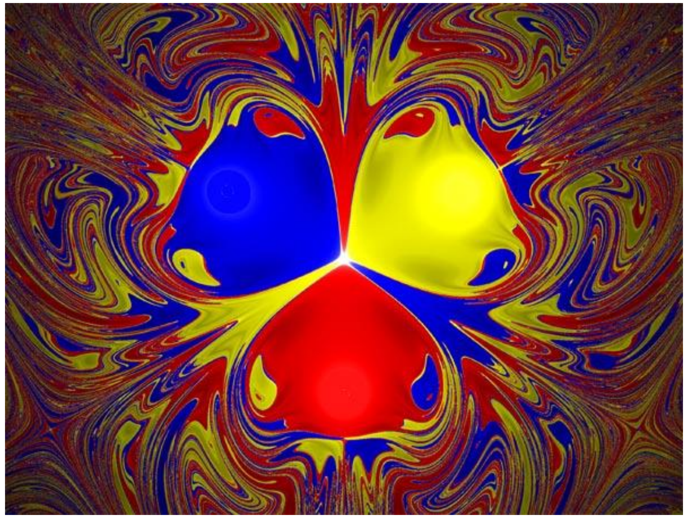
Embedded attractor

Other systems – Magnetic Pendulum

Does HAVOK work for more complex chaotic systems?
Is it able to reconstruct a dynamic based on three lobes?

Pendulum which oscillates above three magnets exerting an attractive force on it.



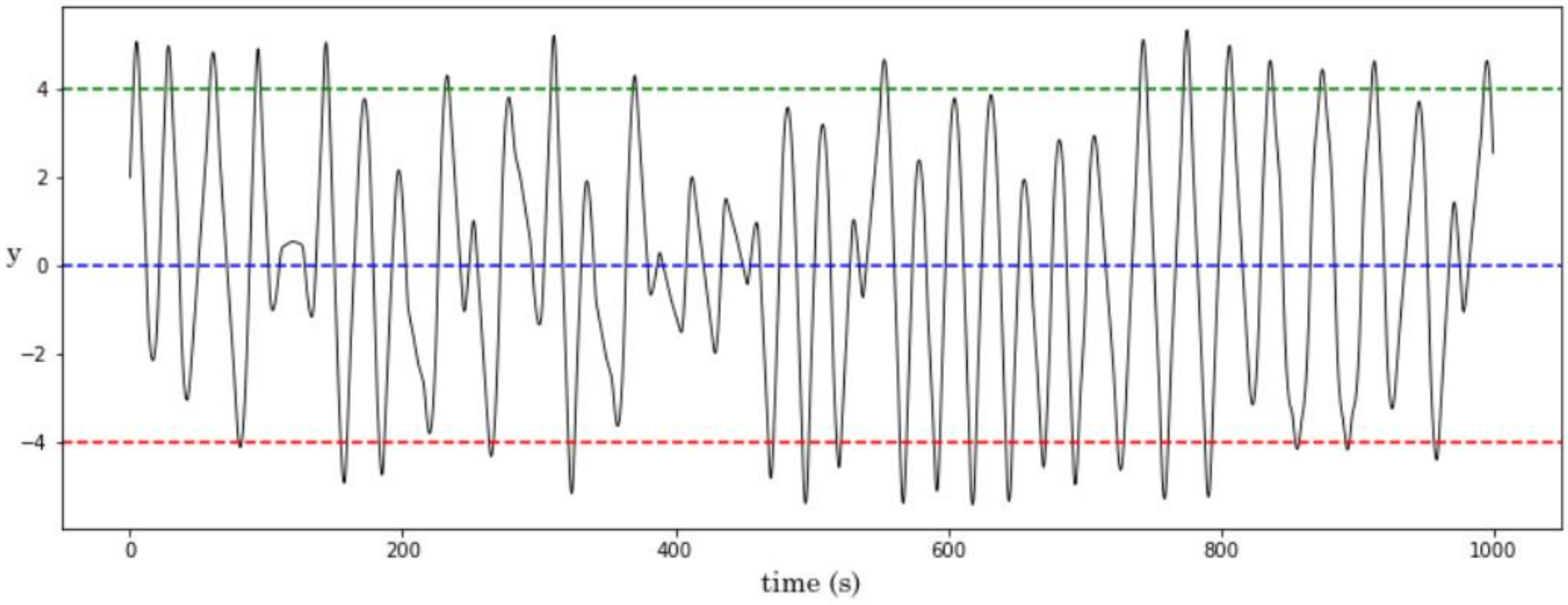


$$\begin{cases} \dfrac{\mathrm{d}^2x}{\mathrm{d}t^2} + R\dfrac{\mathrm{d}x}{\mathrm{d}t} - \displaystyle\sum_{i=1}^{3} \dfrac{x_i - x}{\left(\sqrt{(x_i - x)^2 + (y_i - y)^2 + D^2}\right)^3} + Cx = 0 \\ \dfrac{\mathrm{d}^2y}{\mathrm{d}t^2} + R\dfrac{\mathrm{d}y}{\mathrm{d}t} - \displaystyle\sum_{i=1}^{3} \dfrac{y_i - y}{\left(\sqrt{(x_i - x)^2 + (y_i - y)^2 + D^2}\right)^3} + Cy = 0 \end{cases}$$

$R = 0$	air resistance factor
$C = 0.03$	gravity factor
$D = 1$	pendulum – plane distance

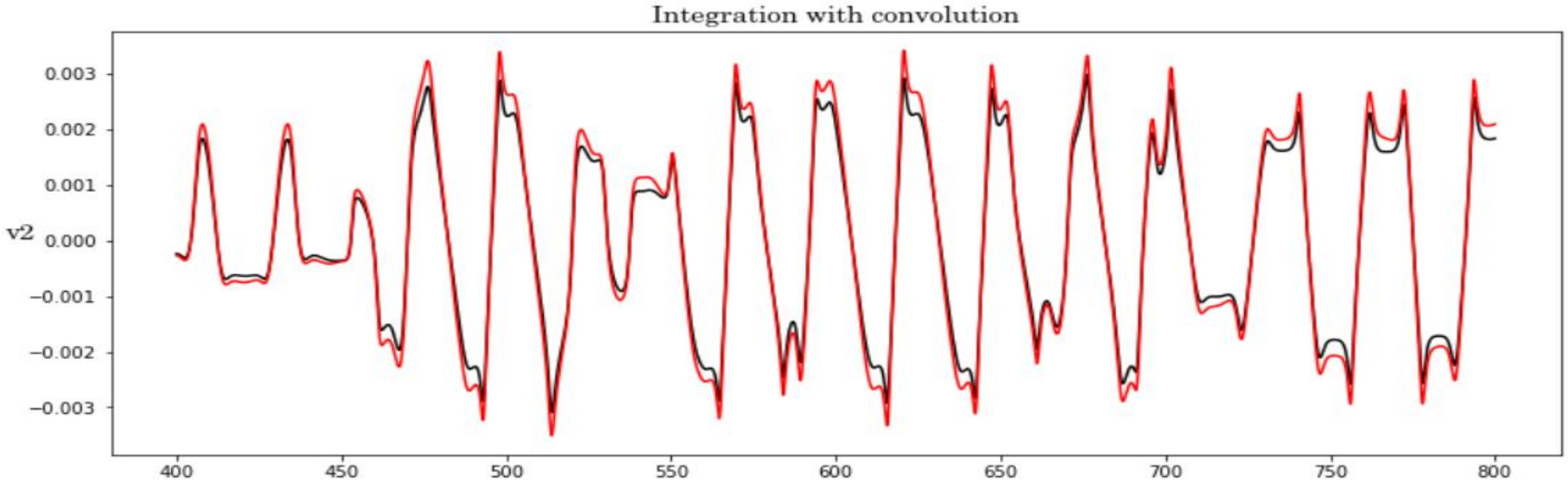
Other systems – Magnetic Pendulum

The time series is obtained by taking the Y coordinate.

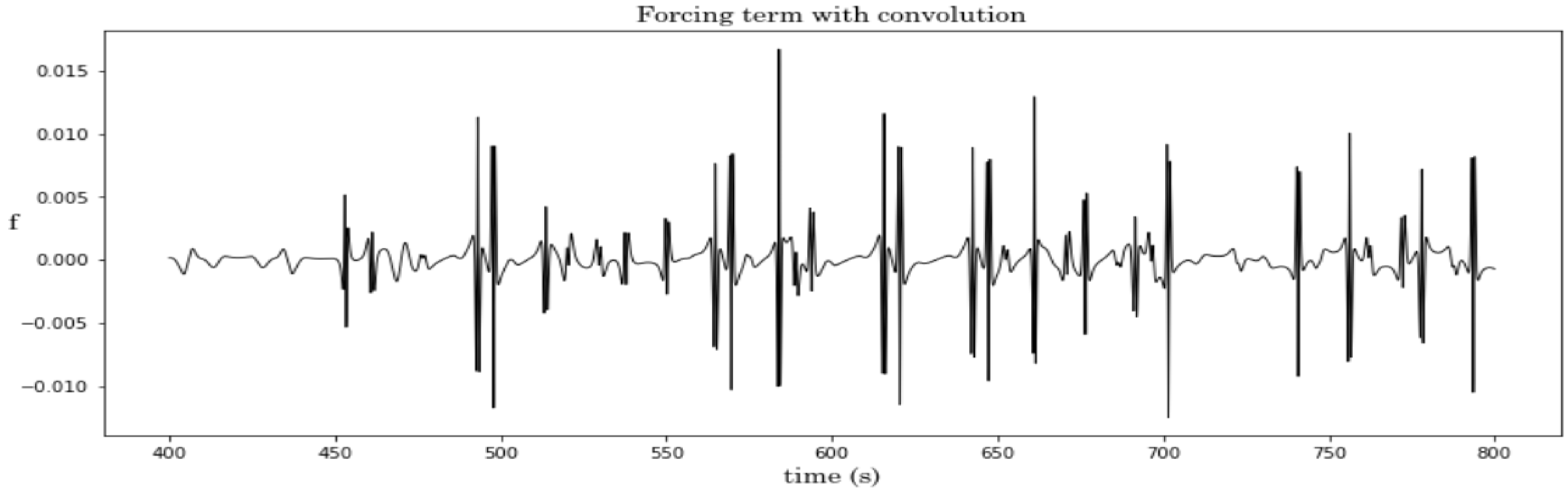


The states of the system are not easily defined, making predictions appear difficult.

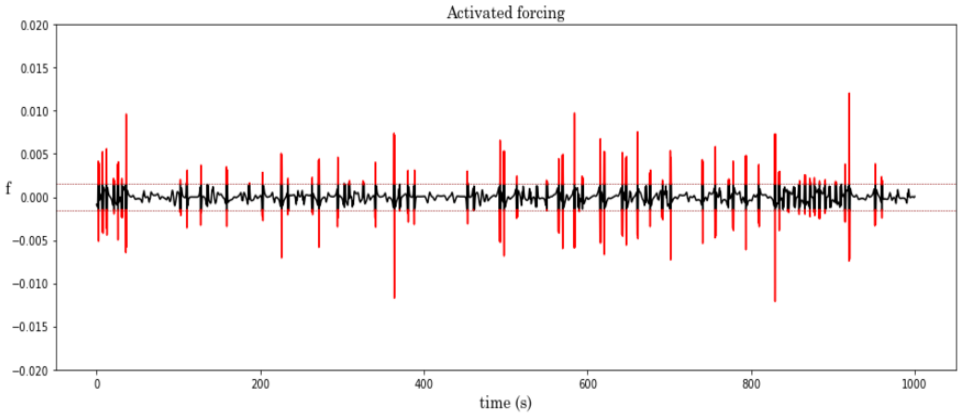
Other systems – Magnetic Pendulum



l = 100
dt = 0.001
r= 7

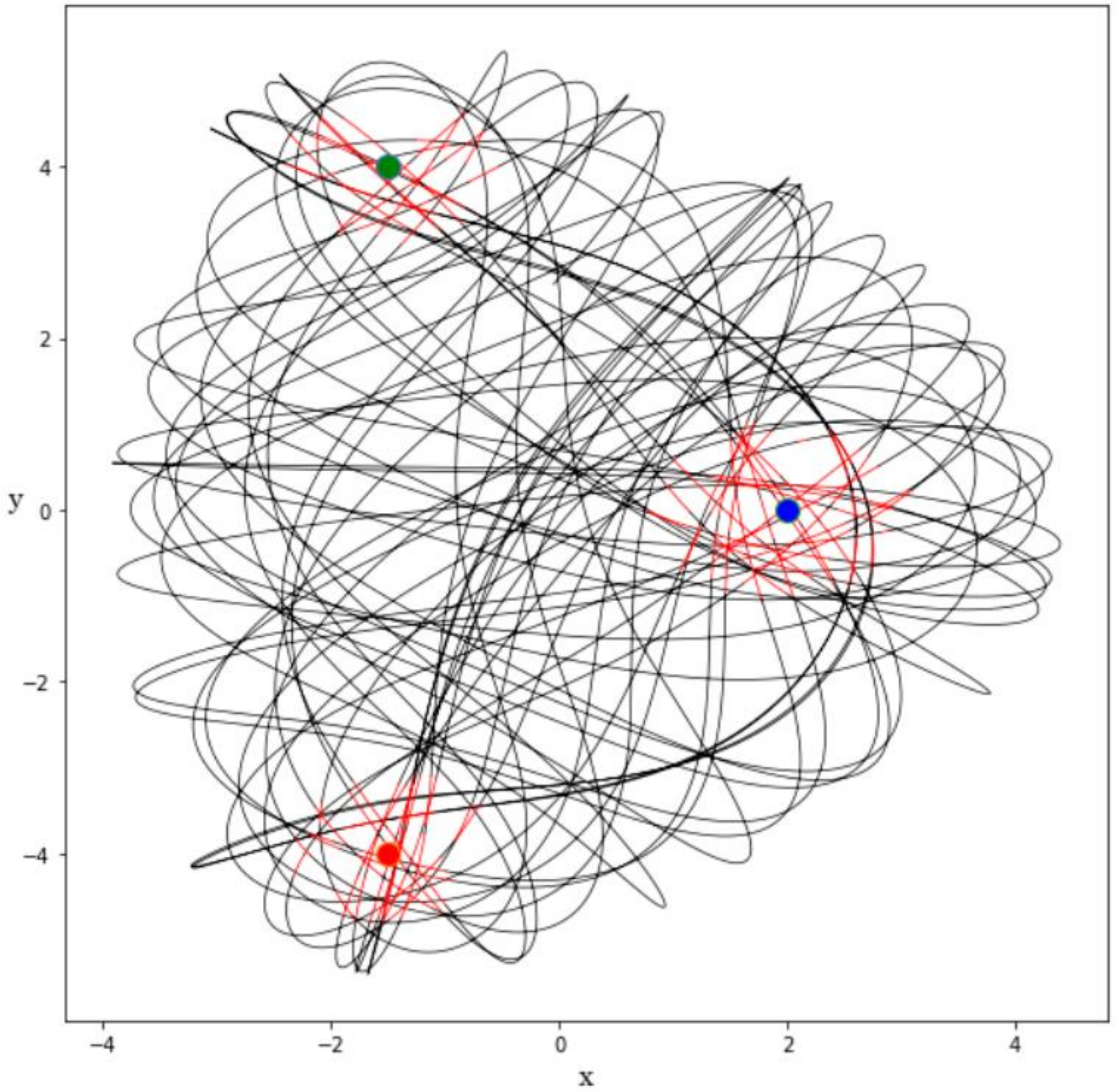


Other systems – Magnetic Pendulum



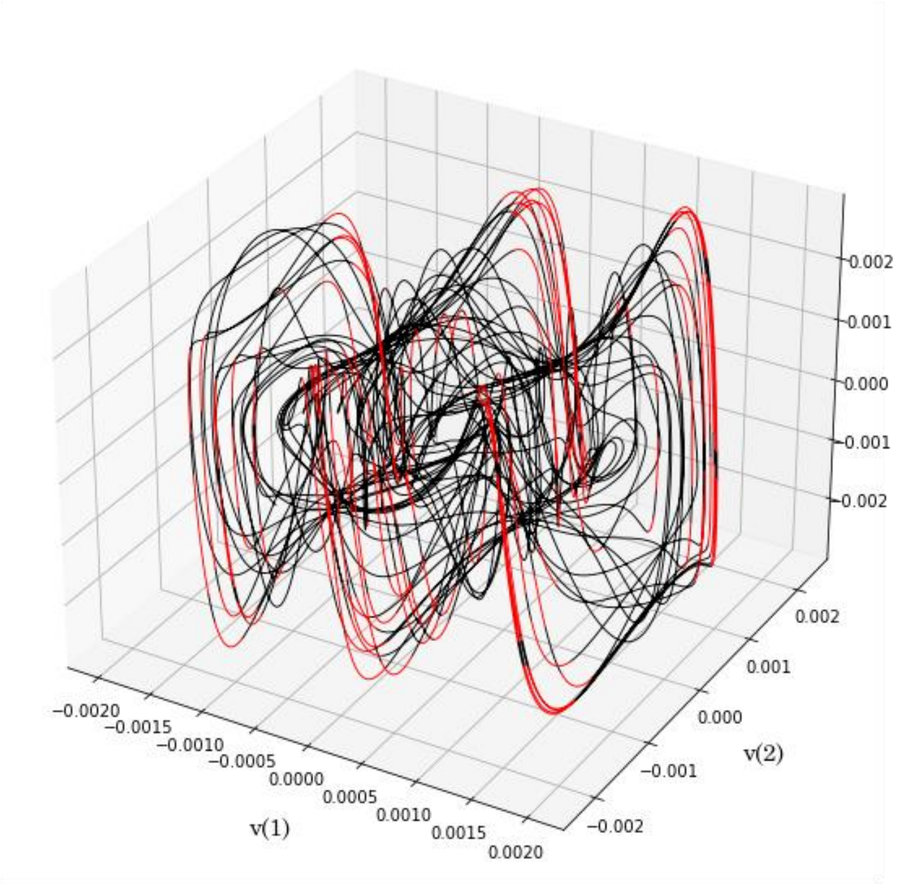
$$\begin{cases} \dfrac{\mathrm{d}^2x}{\mathrm{d}t^2} + R\dfrac{\mathrm{d}x}{\mathrm{d}t} - \boxed{\displaystyle\sum_{i=1}^{3} \frac{x_i - x}{\left(\sqrt{(x_i-x)^2+(y_i-y)^2+D^2}\right)^3}} + Cx = 0, \\ \dfrac{\mathrm{d}^2y}{\mathrm{d}t^2} + R\dfrac{\mathrm{d}y}{\mathrm{d}t} - \boxed{\displaystyle\sum_{i=1}^{3} \frac{y_i - y}{\left(\sqrt{(x_i-x)^2+(y_i-y)^2+D^2}\right)^3}} + Cy = 0, \end{cases}$$

The forcing indeed isolates the nonlinear components of the system

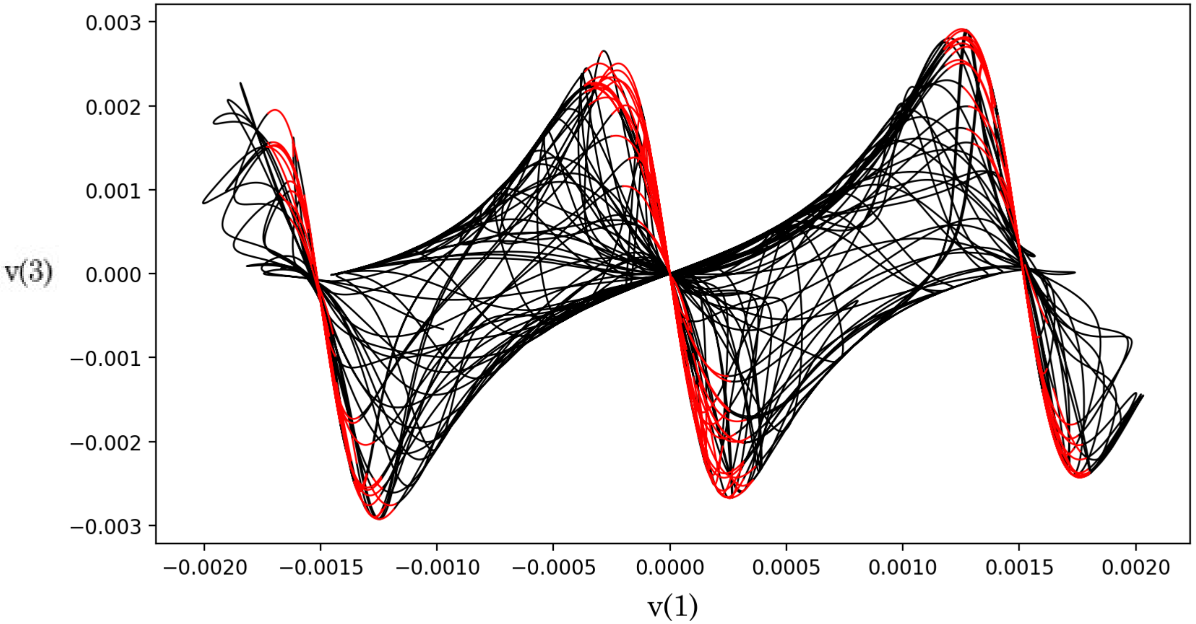


Laboratory of Computational Physics

Other systems – Magnetic Pendulum

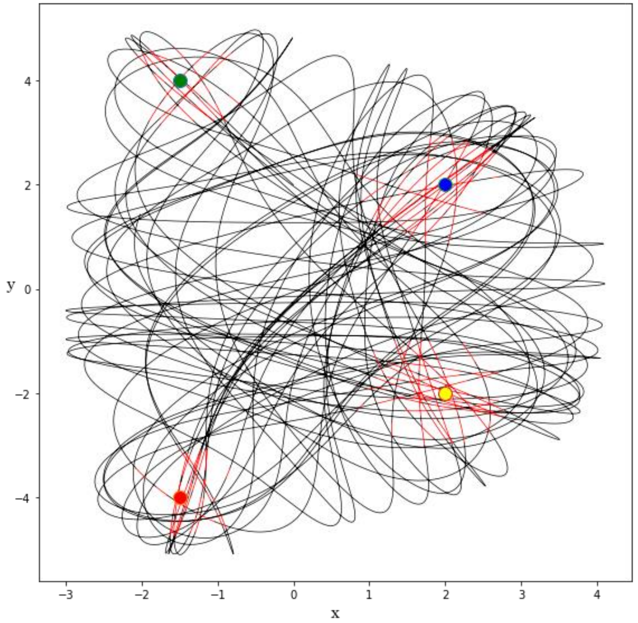


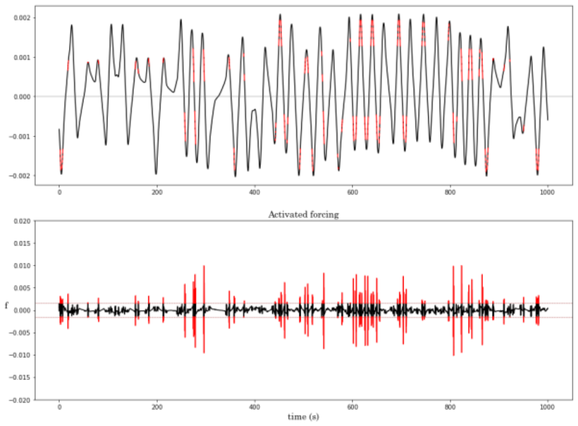
Embedded attractor

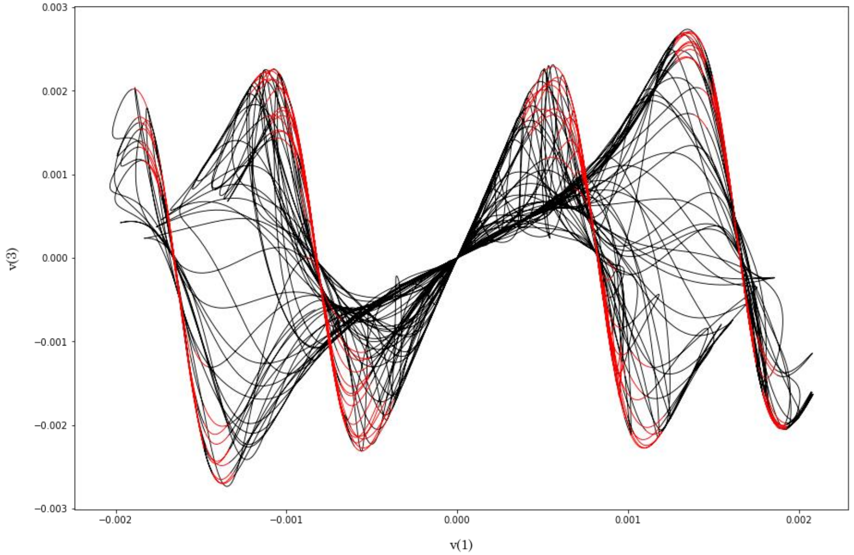


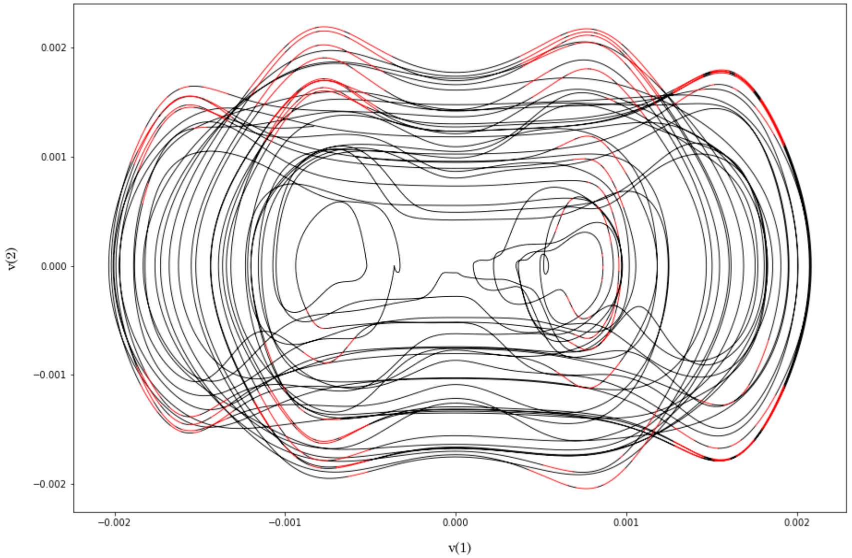
Projection on the x-z plane

Other systems – Magnetic Pendulum







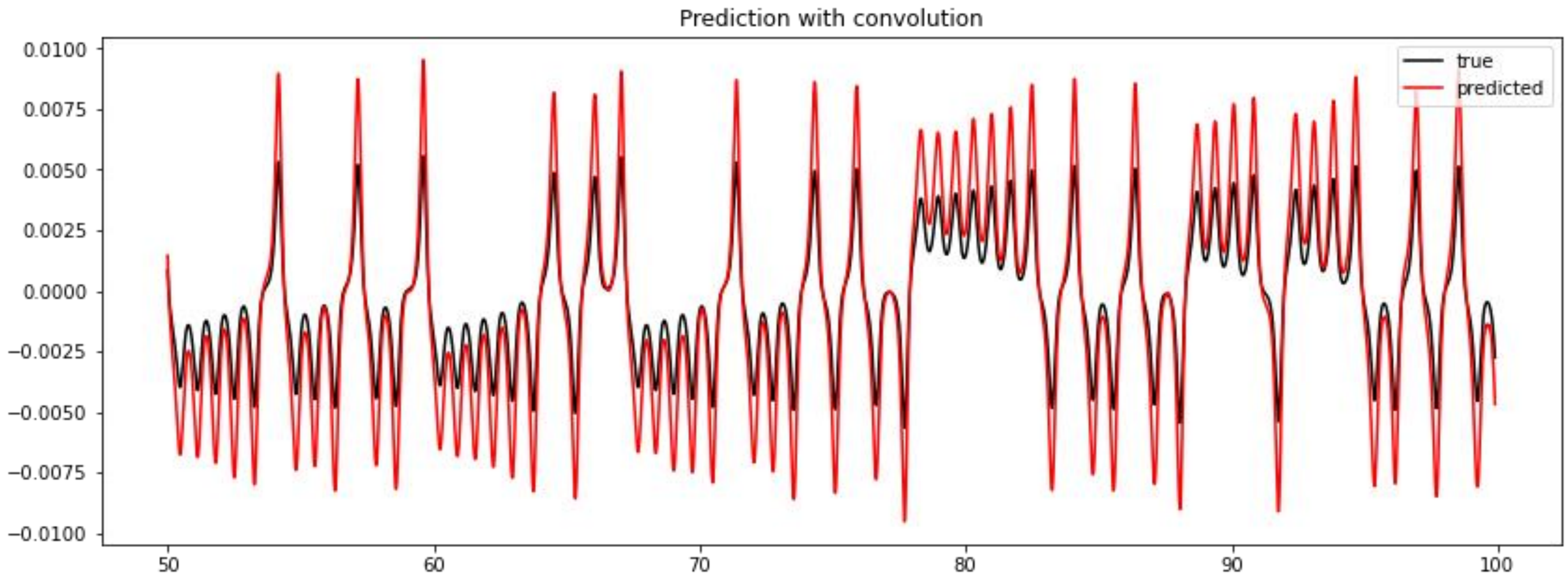


Laboratory of Computational Physics

Open questions

- Rescaling problem on the convolutional prediction:

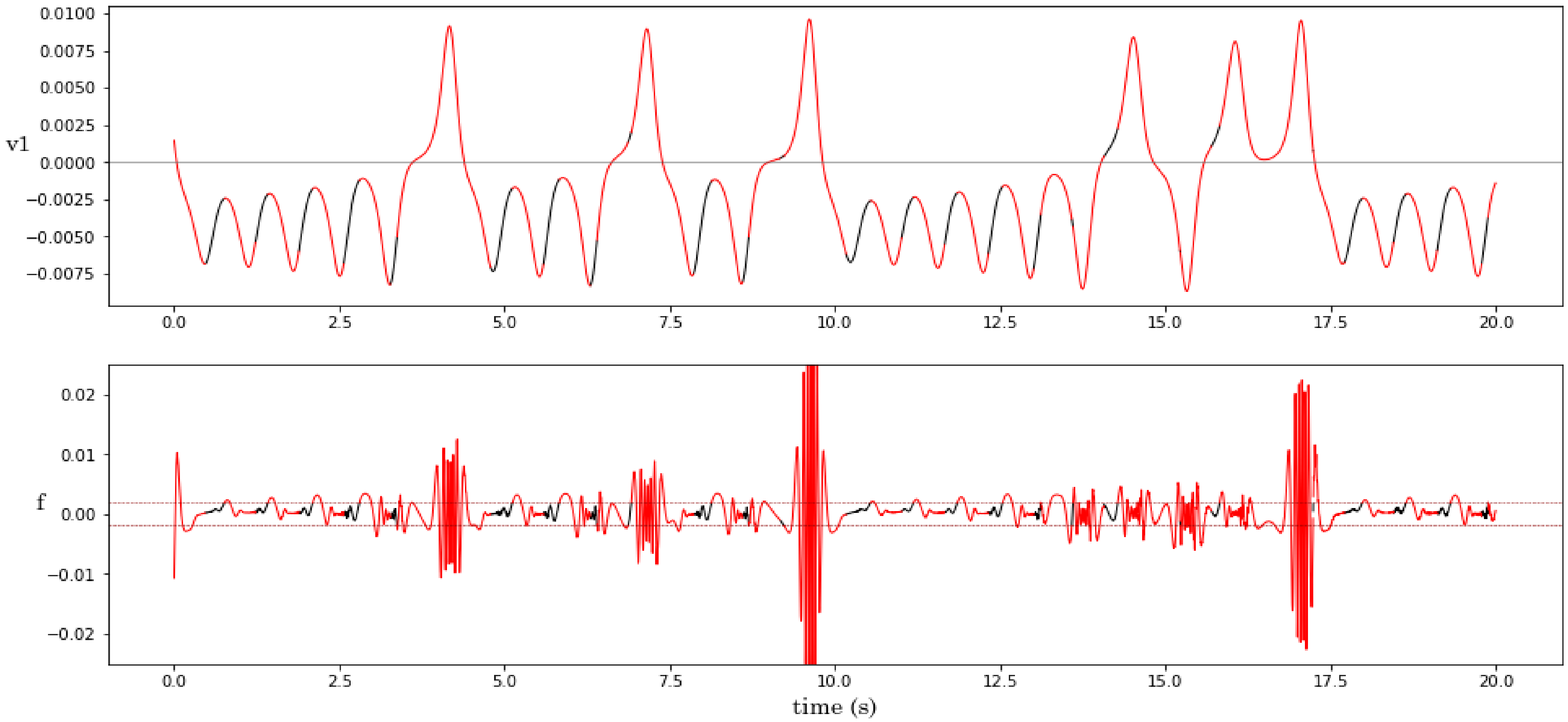
If the sizes of the train and the test are different, the prediction needs to be rescaled.



Open questions

- Rescaling problem on the convolutional prediction:

This can create problems when setting the threshold.

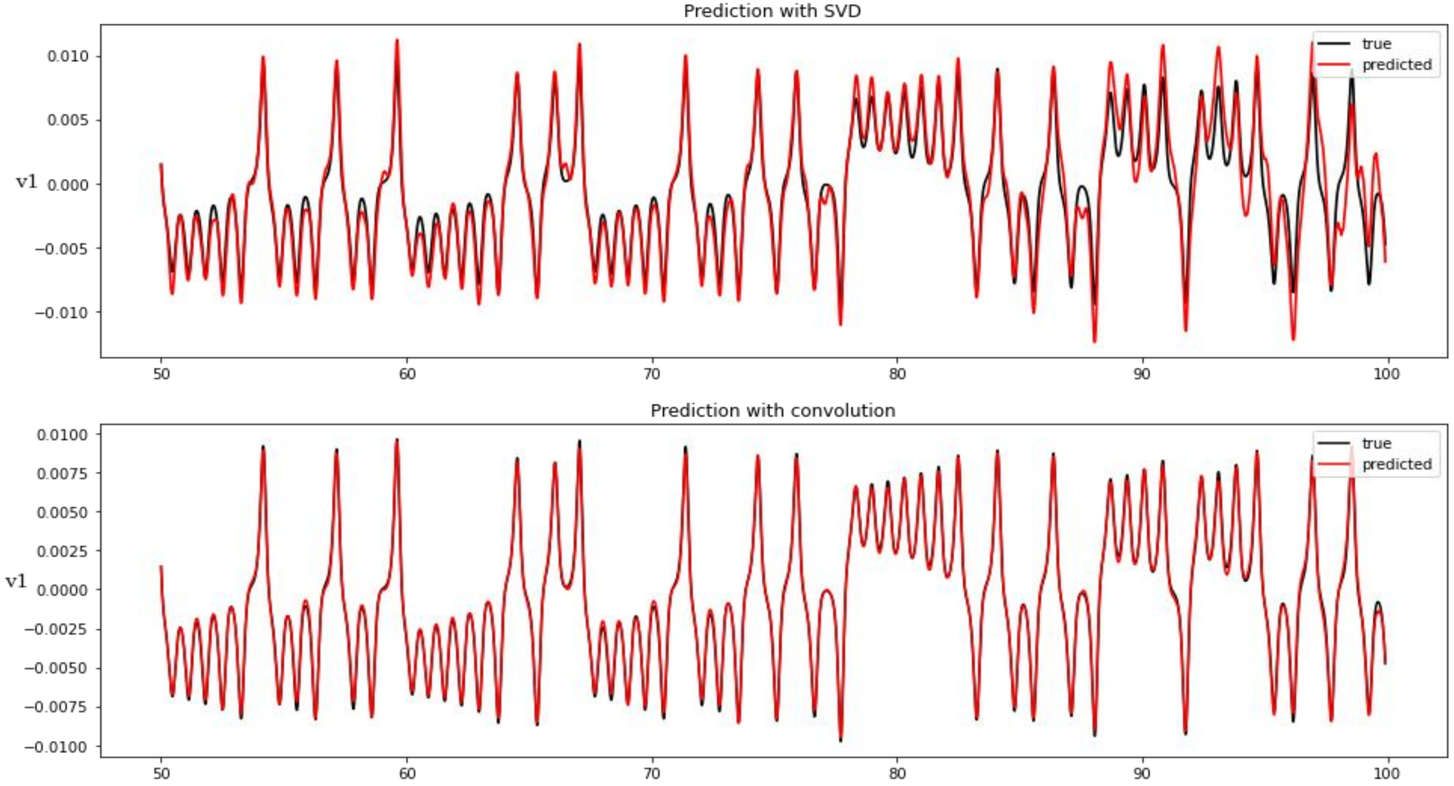


Laboratory of Computational Physics

Open questions

- SVD on the test set vs convolution

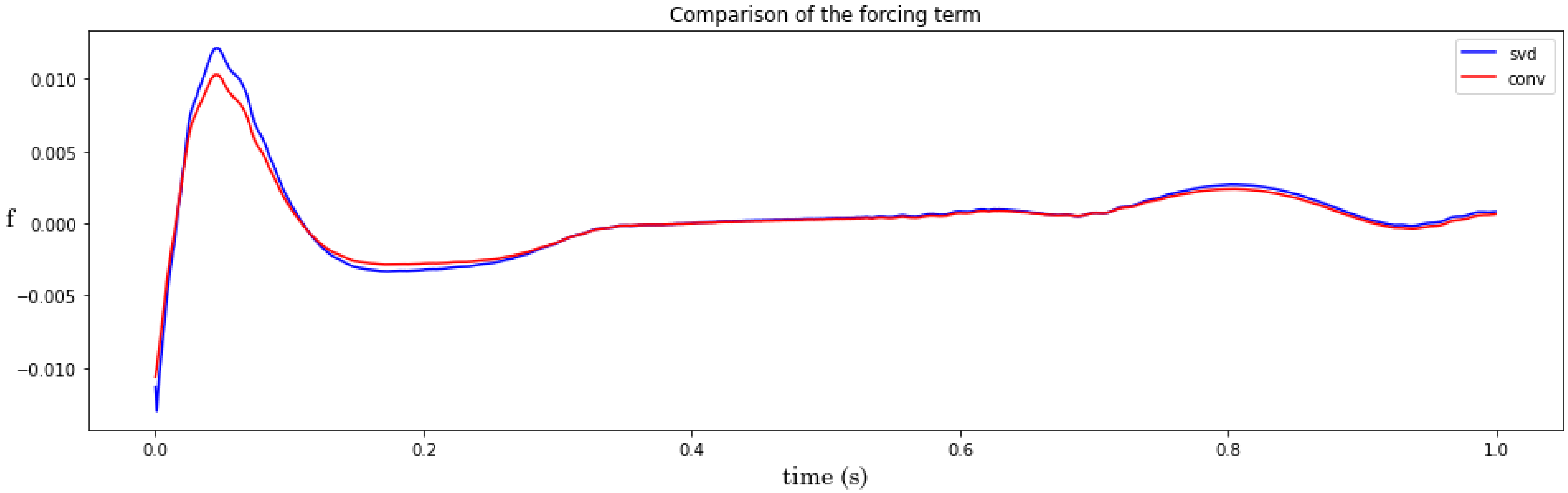
When looking for the forcing term, a straightforward application of an SVD to the test set gives worse results compared to the convolution procedure.



Open questions

- SVD on the test set vs convolution

A small difference in the forcing term creates divergences in the long run.

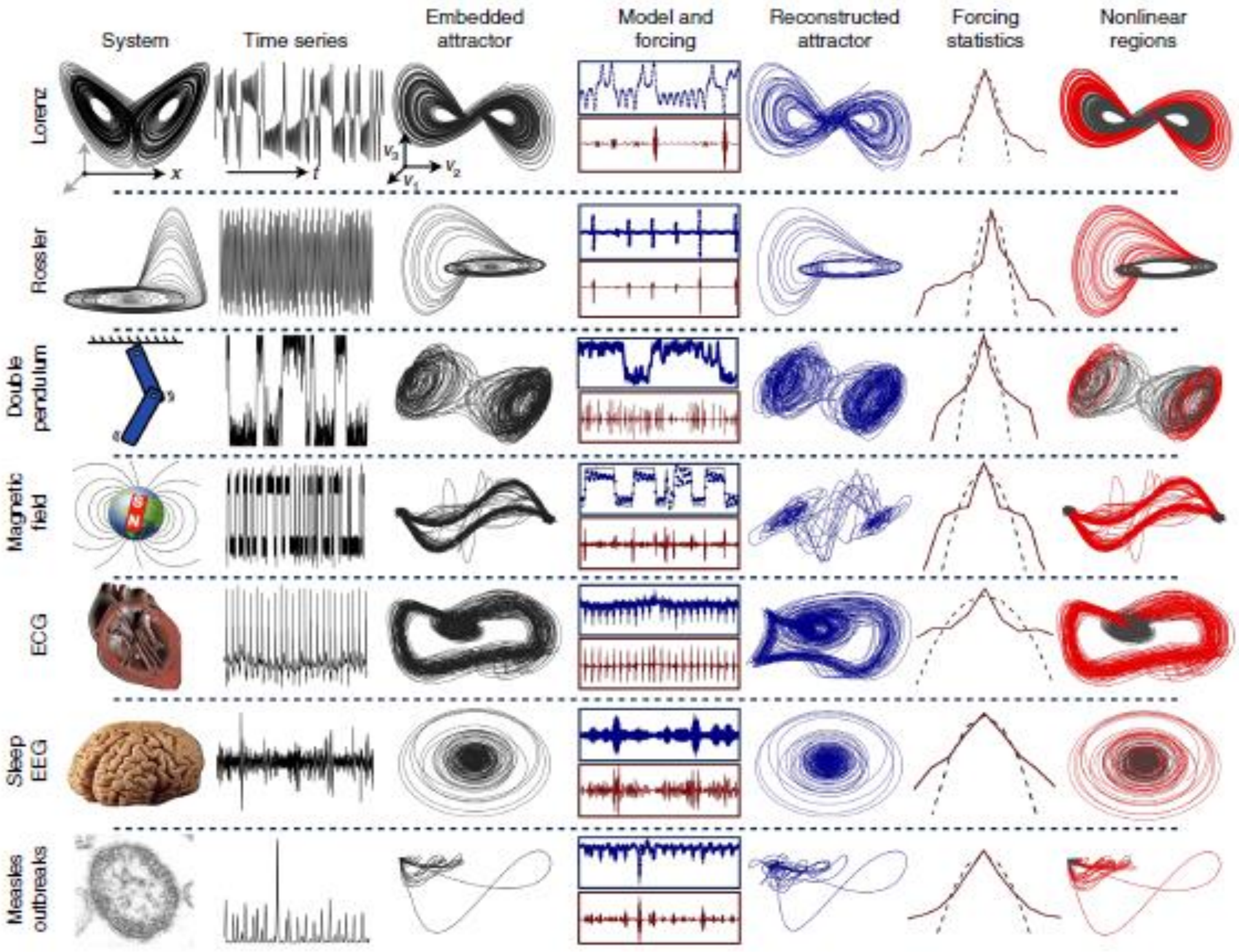


Conclusions

- HAVOK is able to linearize a non-linear system with a minimal loss of information
  - The non-linearity information is contained in the forcing term
- For chaotic systems, the forcing term can be useful to anticipate critical changes in the system behaviour
  - Many applications in control theory
- Yet, perform an HAVOK analysis is not an easy task
  - The model parameters must be set by hand with a trial-and-error process
  - The interpretation is not always straightforward

Backup slides

Other systems

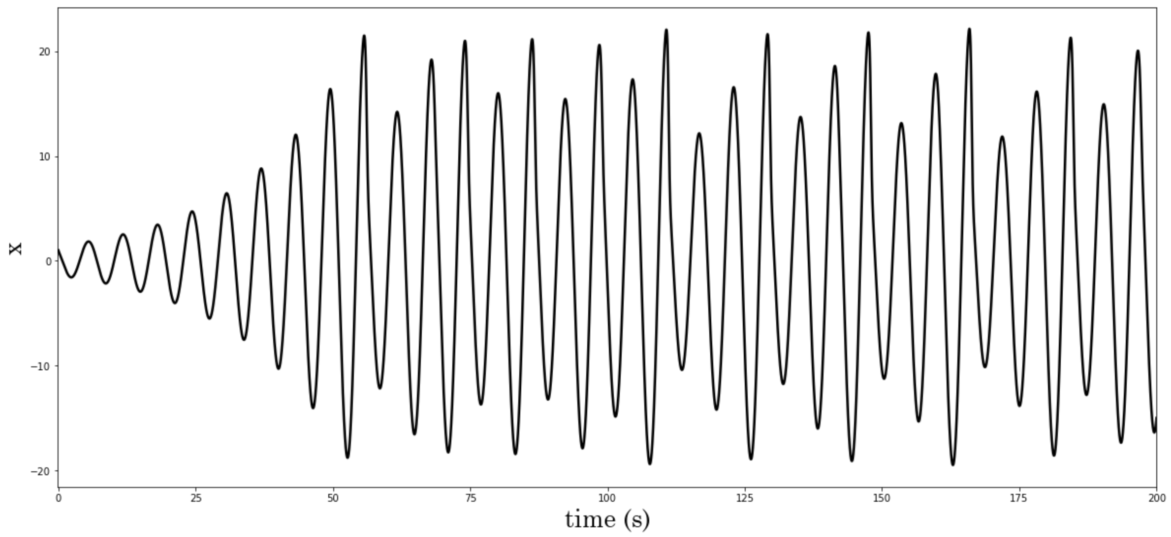


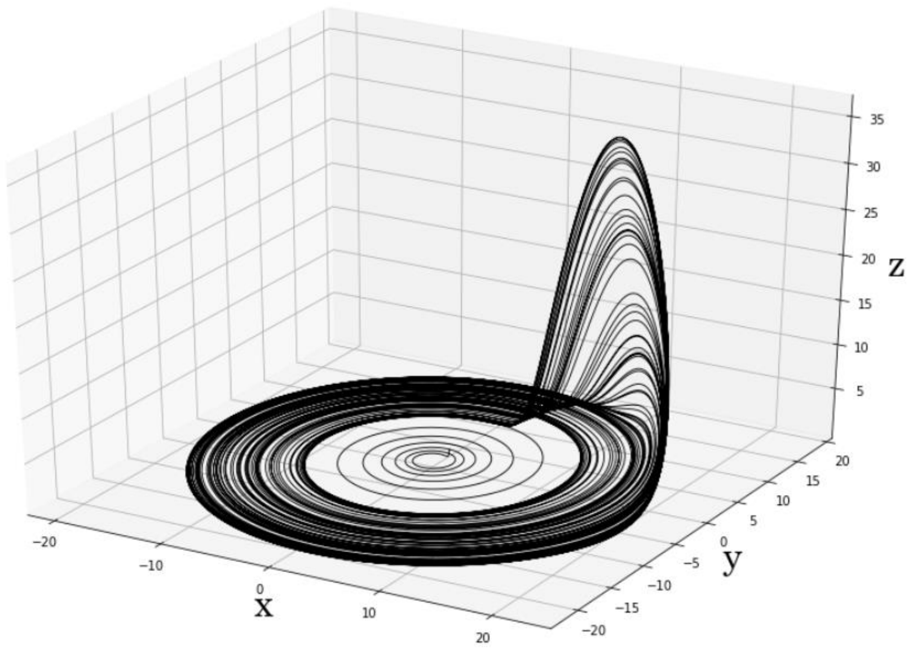
Other system – Rossler

Another simple example of chaotic model is the Rossler attractor.

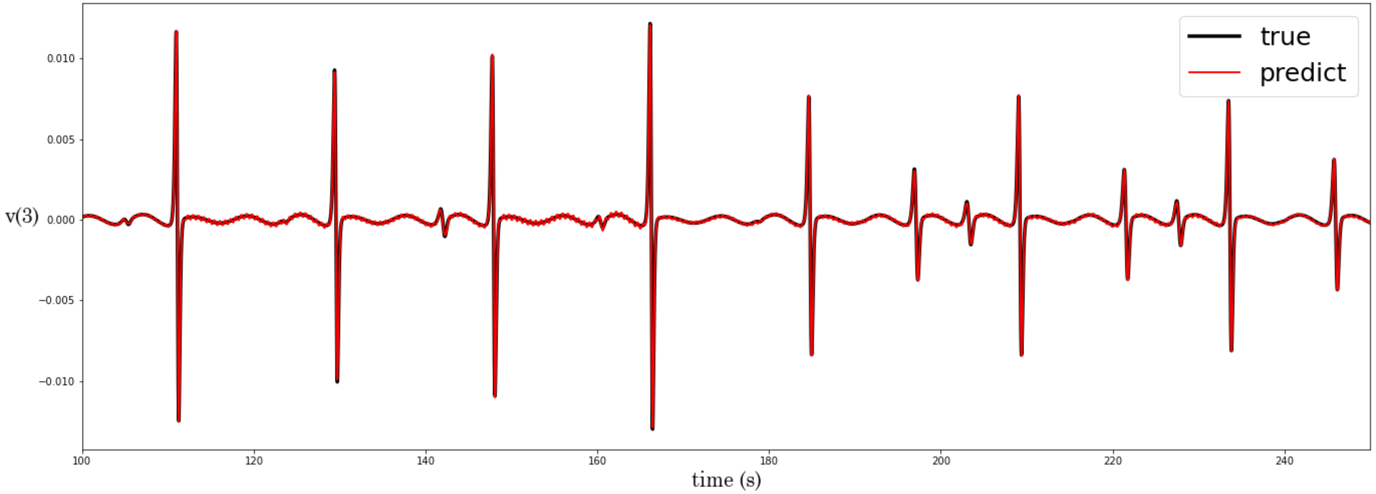
$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + z\,(x - c) \end{cases}$$

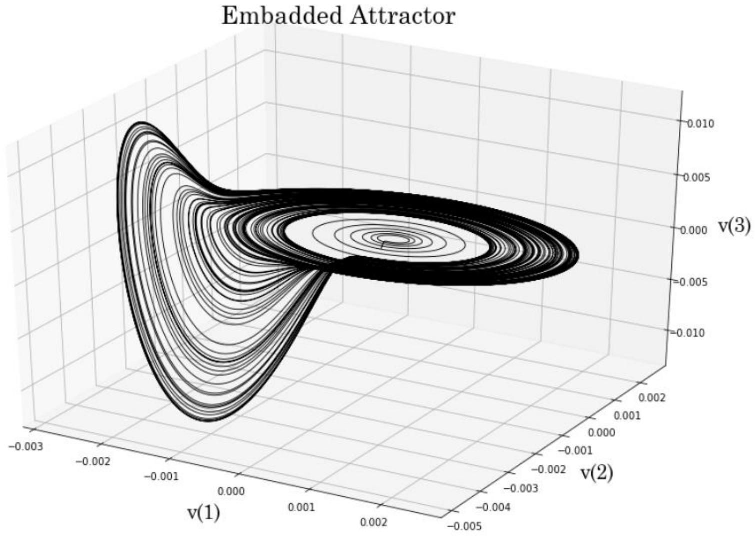
$a = 0.31 \quad b = 0.1 \quad c = 14$

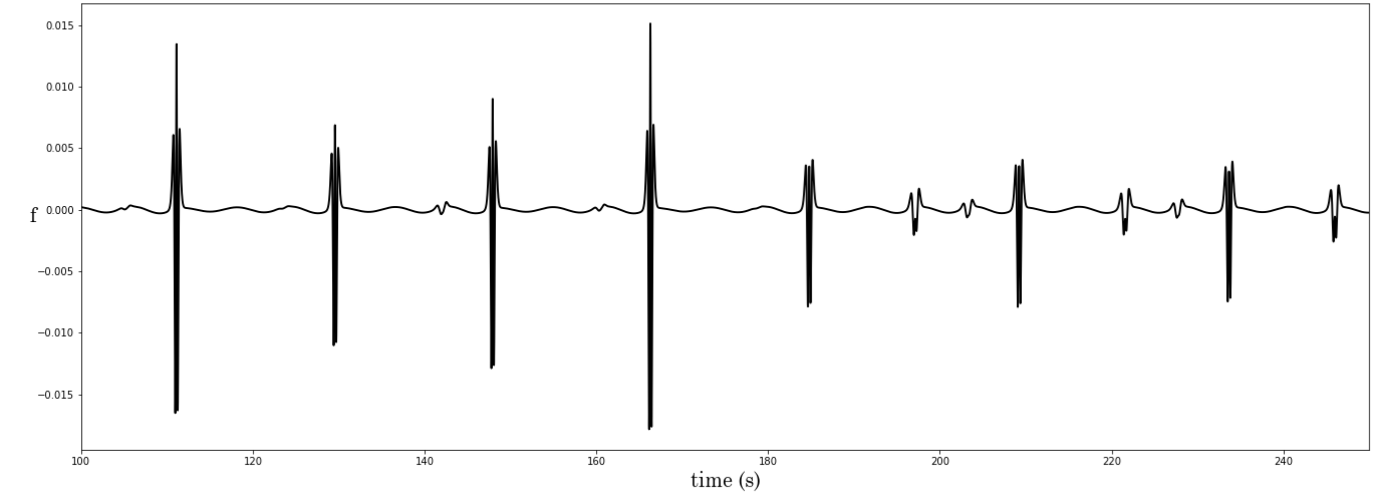


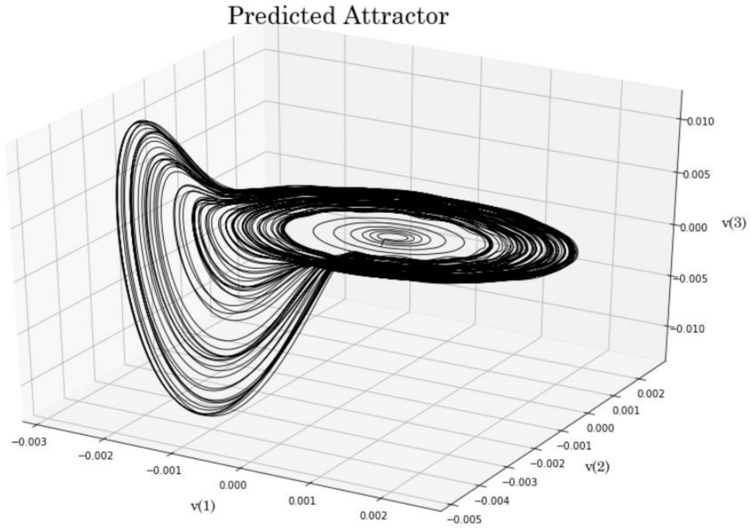


Other system–Rossler

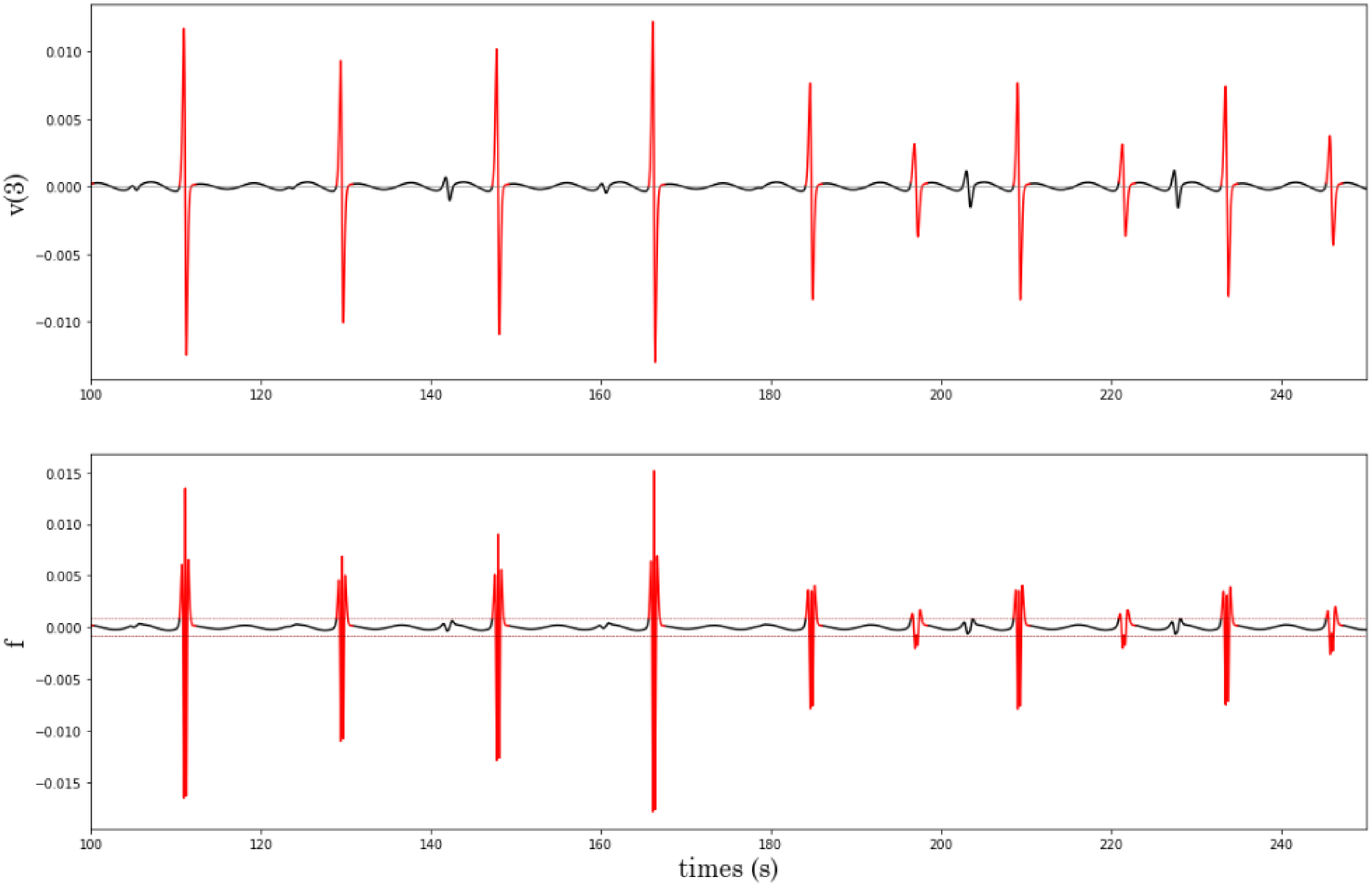




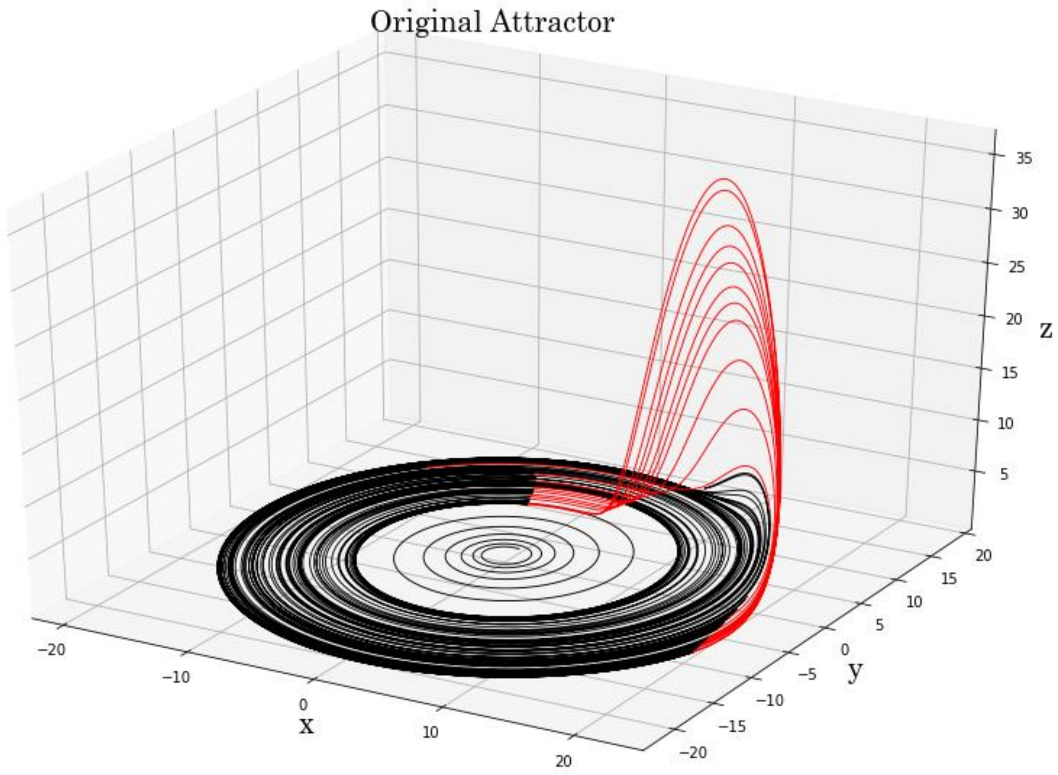


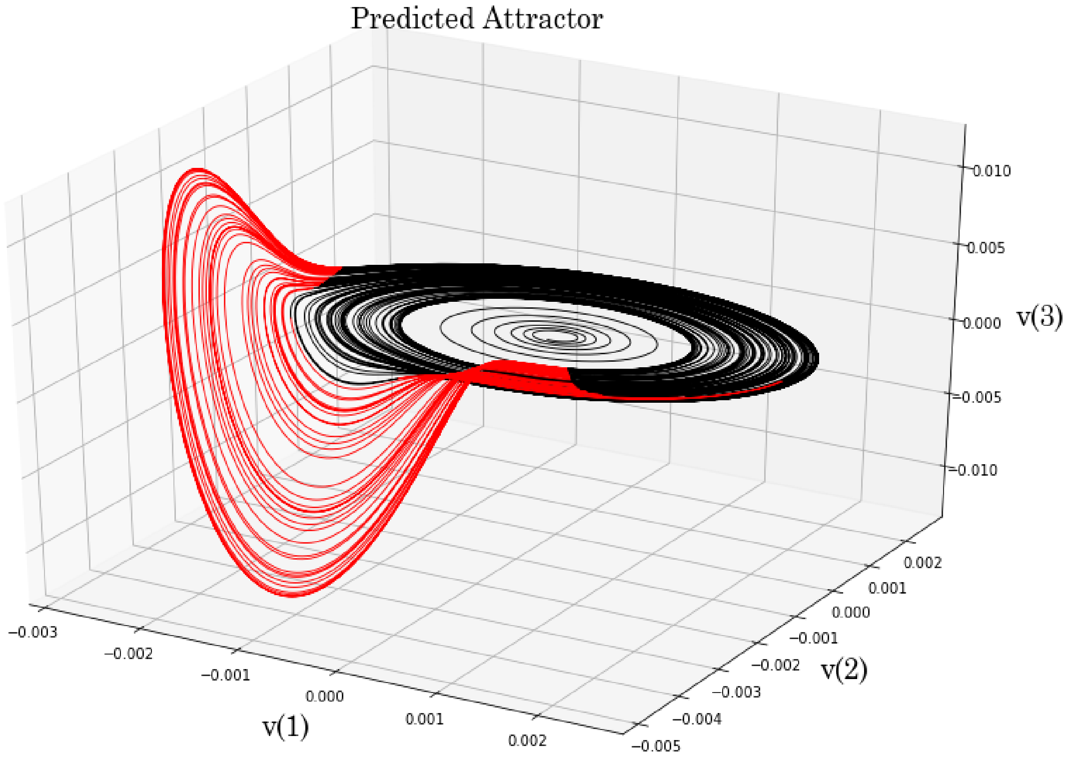


Other system–Rossler



Other system–Rossler





Laboratory of Computational Physics