2.160 Identification, Estimation, and Learning

Part 4 Machine Learning and Nonlinear System Modeling

Lecture 24

Koopman Operator Theory for Exact Linearization of Nonlinear Dynamical Systems

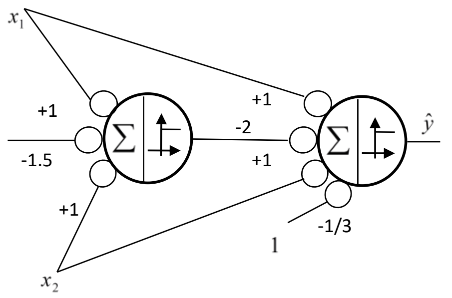
$$K_f \varphi = \varphi \circ f$$

H. Harry Asada
Department of Mechanical Engineering
MIT

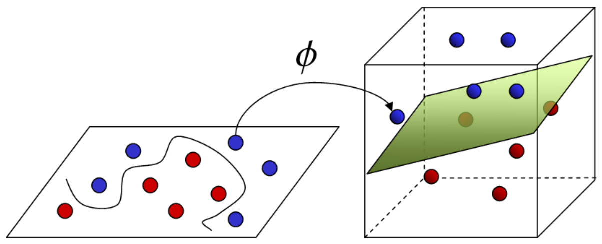
Augmenting / Lifting the Input Space

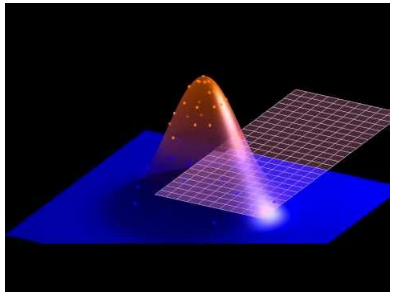
- A focal point of Part 4 Machine Learning and Nonlinear System Modeling: Lifting the input space
- Linearly separable classification: Not linearly separable problems, such as XOR, can be made linearly separable by augmenting the feature space.
- Hidden units of a neural network can create such internal variables to augment the space.
- Kernel methods recast the input space to a high dimensional space, including an infinite dimensional space.
- Gaussian Process exploits covariance kernels to indirectly deal with high-dimensional features.

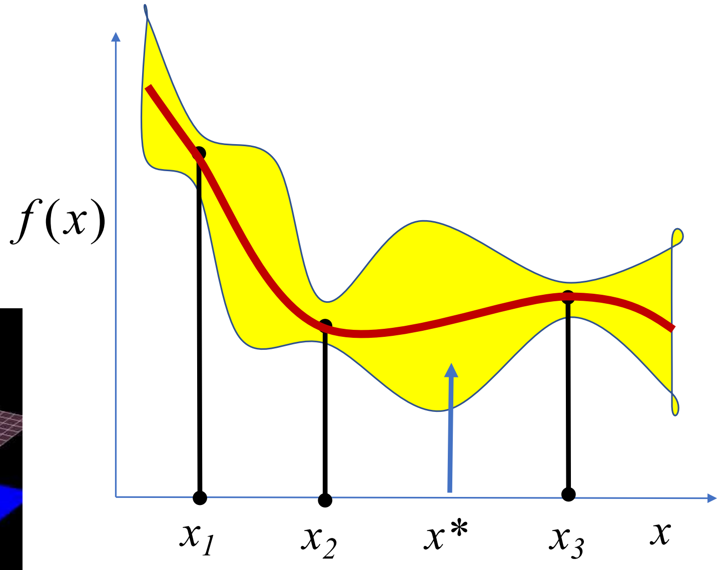
Input		Output
0	0	0
0	1	1
1	0	1
1	1	0
X_1	X_2	y



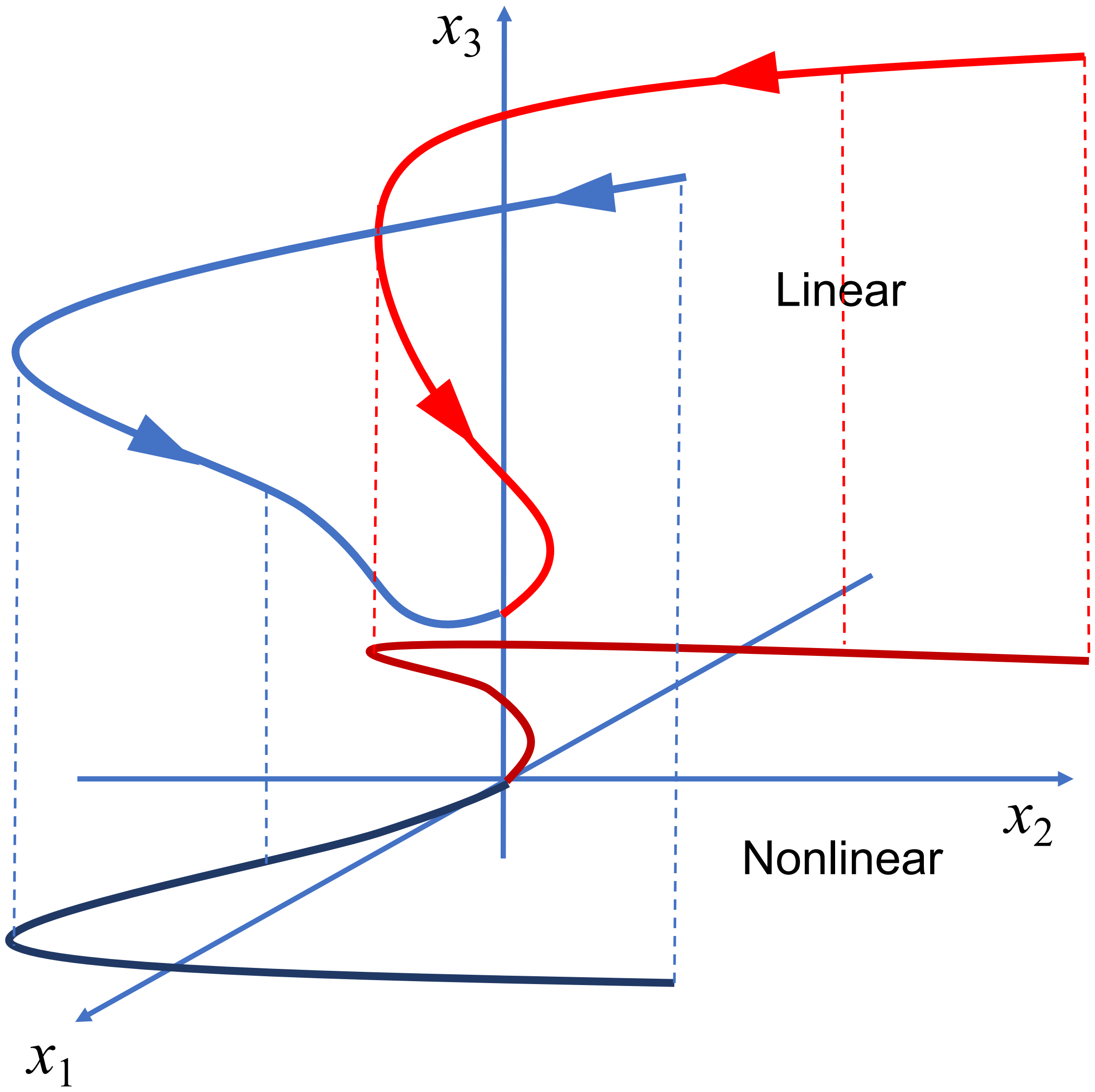
$$\varphi(\boldsymbol{x}) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ c \end{pmatrix}$$







The final two lectures of 2.160 aim to extend the methodology of input space augmentation to Linearization of Nonlinear Dynamical Systems through Lifting Dynamics.



Linear state equations

$$\frac{dz}{dt} = Az \qquad \frac{dz}{dt} = Az + Bu$$

z : High dimensional

Nonlinear state equations

$$\frac{dx}{dt} = f(x) \qquad \frac{dx}{dt} = f(x,u)$$

x : Low dimensional

A Lucky Example

- Consider the following 2nd-order nonlinear dynamical system:

$$\frac{dx_1}{dt} = ax_1$$

$$\frac{dx_2}{dt} = b(x_2 - x_1^{\,2})$$

- Introducing a new set of variables:

$$z_1 = x_1, z_2 = x_2, z_3 = x_1^{\,2}$$

- We can rewrite the original state equation as:

$$\frac{dz_1}{dt} = az_1$$

$$\frac{dz_2}{dt} = b(z_2 - z_3)$$

- The evolution of the third variable z_3 is given by differentiating it.

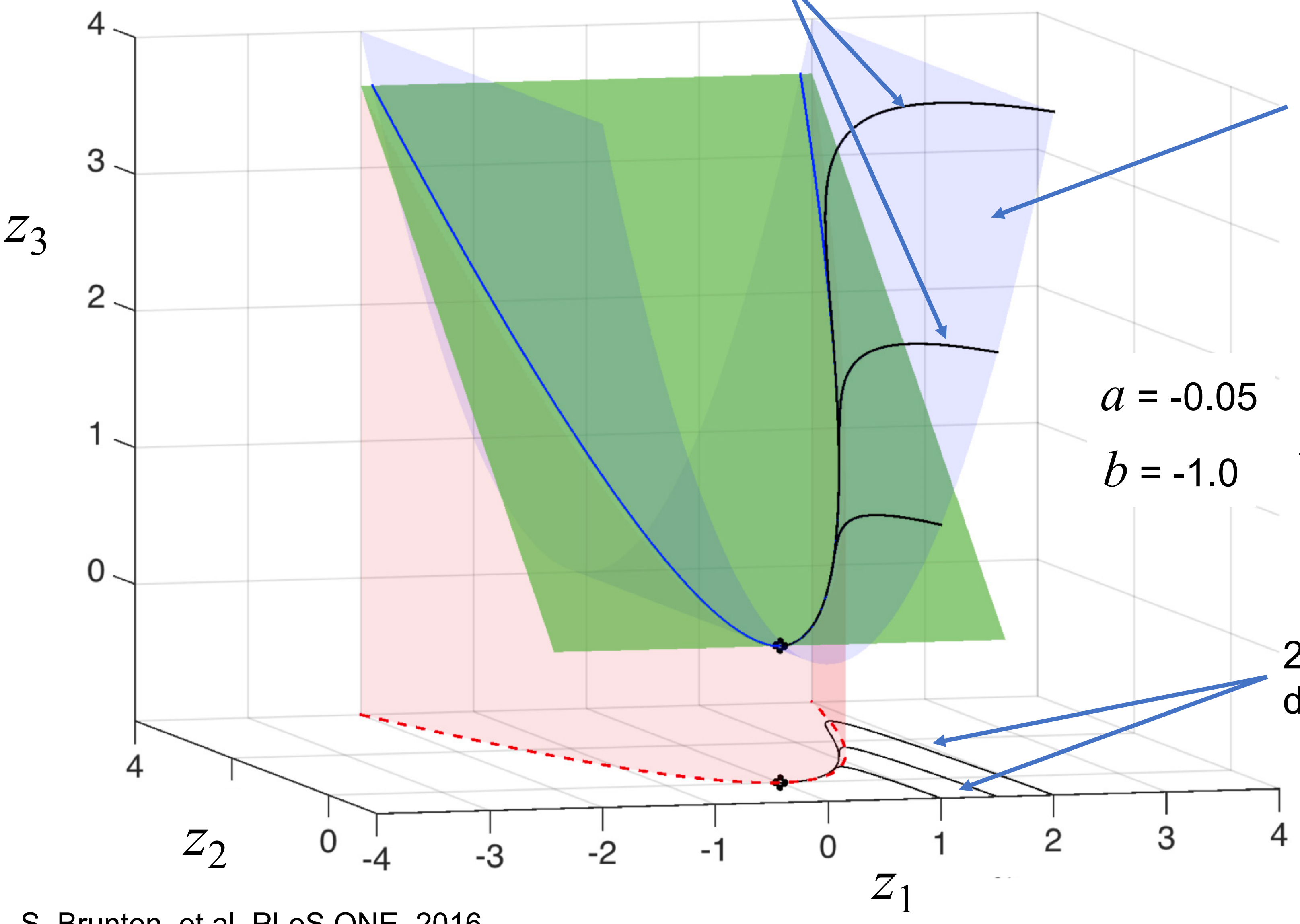
$$\frac{dz_3}{dt} = \frac{dx_1^{\,2}}{dt} = 2x_1\frac{dx_1}{dt} = 2x_1ax_1 = 2ax_1^{\,2} = 2az_3$$

- Therefore, the system is represented as a linear 3rd order system.

$$\frac{d}{dt}\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & b & -b \\ 0 & 0 & 2a \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

- Note that no approximation is used. The lifted system is linear and exact.

3D linear dynamics trajectories



The trajectories are constrained in this curve:

$$z_3 = z_1^{\,2}$$

a = -0.05

b = -1.0

$$\frac{d}{dt}\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & b & -b \\ 0 & 0 & 2a \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

2D nonlinear dynamics trajectories

S. Brunton, et al. PLoS ONE, 2016

A Motivating Example of Lifting Linearization

- Once linearized, the state equation can be applied to various nonlinear dynamics analysis and control design problems.
- Consider the above system with control input u.

$$\frac{d}{dt}\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & b & -b \\ 0 & 0 & 2a \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} u$$

- Let us apply Linear Quadratic Regulator (LQR) that optimizes the following cost functional.

$$J = \int_0^\infty \left(\boldsymbol{z}(t)^T \boldsymbol{Q}\, \boldsymbol{z}(t) + \boldsymbol{u}(t)^T \boldsymbol{R}\, \boldsymbol{u}(t) \right) dt$$

where

$$\boldsymbol{z}(t) = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \qquad \boldsymbol{u}(t) = u(t) \qquad \boldsymbol{Q} = \begin{pmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \boldsymbol{R} = 1$$

- Solving the above LQR problem, we can find an optimal state feedback law:

$$u(t) = -(k_1, k_2, k_3)\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = -(k_1 x_1 + k_2 x_2 + k_3 x_1^2)$$

- Note that this feedback law is a nonlinear feedback since x_1^2 is involved.

- Comparing the above LQR in the lifted space, let us consider a nonlinear optimal control for the original system.

- Minimize:

$$J = \int_0^\infty \left(\boldsymbol{x}(t)^T \boldsymbol{Q}_0\, \boldsymbol{x}(t) + \boldsymbol{u}(t)^T \boldsymbol{R}\, \boldsymbol{u}(t) \right) dt \qquad \boldsymbol{Q}_0 = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{12} & Q_{22} \end{pmatrix} \qquad \boldsymbol{R} = 1$$

Subject to $\dfrac{dx_1}{dt} = a x_1, \quad \dfrac{dx_2}{dt} = b(x_2 - x_1^2)$

- This optimization is difficult to solve; no longer convex optimization; a numerical solution may be at a local minimum, and the computation is more expensive.

Koopman Operator

- The above case study is a special case of lifting linearization, where simple embedding of nonlinear terms leads to a complete linear model. General nonlinear dynamical systems cannot be represented by exact linear equations of finite order.
- However, an arbitrary, autonomous nonlinear dynamical system can be represented by a linear system of infinite order in a Hilbert space, thanks to Bernard Koopman.

PNAS

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

HAMILTONIAN SYSTEMS AND TRANSFORMATIONS IN HILBERT SPACE

The Great Depression time

BY B. O. KOOPMAN

DEPARTMENT OF MATHEMATICS, COLUMBIA UNIVERSITY

Communicated March 23, 1931

In recent years the theory of Hilbert space and its linear transformations has come into prominence.[1] It has been recognized to an increasing

Koopman Operator

- ❑ We start with a discrete-time dynamical system, while the theory applies to a continuous-time system. Consider a nonlinear autonomous (no input) system:

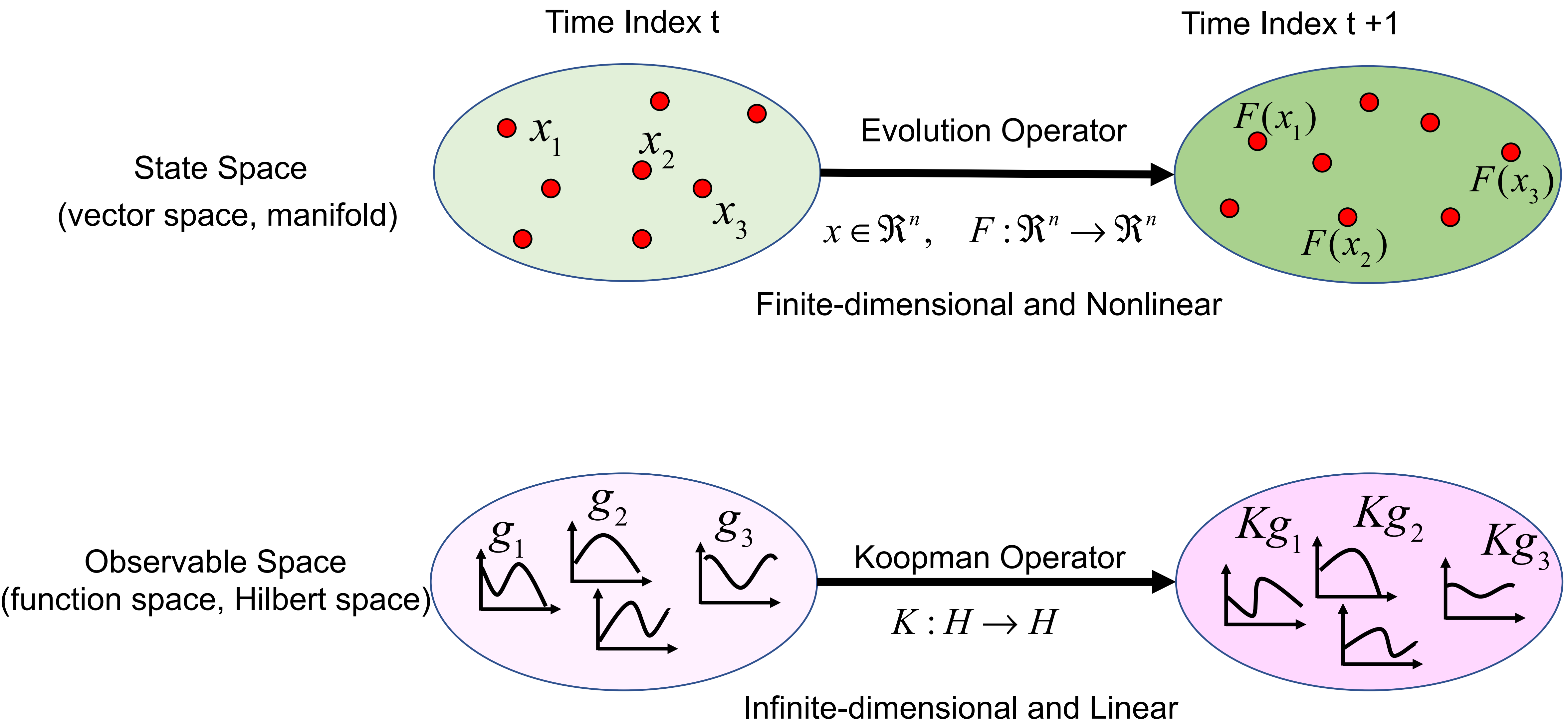
$$x_{t+1} = F(x_t) \qquad \text{where } x \in \Re^n, \quad F : \Re^n \to \Re^n \quad \text{continuous}$$

- ❑ Let $g(x) : \Re^n \to \Re$ be an observable, a scaler-valued function of state, which resembles output y. Here, $g(x)$ can be a sensor measurement, a nonlinear function of state variables, such as $z_3 = x_1{}^2$ in the previous example, or one of the state variables.
- ❑ Collection of all such observables form a linear vector space. Koopman Operator, denoted by ***K***, is a linear transformation on this vector space.

$$\boldsymbol{K} g(x) = g \circ F(x)$$

  - Here $\circ$ denotes a composition operation. In this case, the observable function g applies to $F(x)$, which represents the state of the next time step.
  - The Koopman operator is linear. That means, ***K*** is a type of matrix, but infinite dimension.
  - The Koopman Operator applies to the collection of observations, a vector of infinite dimension, that is, a function $g(x)$.

Schematic of Koopman Operator



A Brute-force Method for Obtaining a Linear State Equation in a Lifted Space

- Given a nonlinear state equation, find nonlinear terms in $F(x)$ and replace them by observables.

Example: $$x_{t+1} = ax_t + b\underbrace{x_t^2}_{g_1(x_t)} + c\underbrace{\sin \pi x_t}_{g_2(x_t)}$$

- This allows us to rewrite the state equation as a linear equation with a set of observables.

$$x_{t+1} = ax_t + bg_1(x_t) + cg_2(x_t)$$

- Formulate the transition of all the observables, $g_1(x_{t+1}), g_2(x_{t+1})$, as linear functions of observables and state, $g_1(x_t), g_2(x_t), x_t$.

$$g_1(x_{t+1}) = k_{10}x_t + k_{11}g_1(x_t) + k_{12}g_2(x_t)$$
$$g_2(x_{t+1}) = k_{20}x_t + k_{21}g_1(x_t) + k_{22}g_2(x_t)$$

A Brute-force Method (continued)

- Including other observables, write a set of augmented state equations, which represents "point-wise" transitions of state variables and observables.

$$
\begin{pmatrix} x_1(t+1) \\ \vdots \\ x_n(t+1) \\ g_{n+1}(t+1) \\ \vdots \\ g_{m-n}(t+1) \end{pmatrix}
=
\underbrace{\begin{pmatrix}
a_{11} & \cdots & a_{1,n} & a_{1,n+1} & \cdots & a_{1,m} \\
\vdots & \ddots & \vdots & & & \vdots \\
a_{n,1} & \cdots & a_{n,n} & & & \vdots \\
a_{n+1,1} & & & \ddots & & \vdots \\
\vdots & & & & \ddots & \vdots \\
a_{m,1} & \cdots & \cdots & \cdots & \cdots & a_{m,m}
\end{pmatrix}}_{A_m}
\begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \\ g_{n+1}(t) \\ \vdots \\ g_{m-n}(t) \end{pmatrix}
$$

- Note that the observables are renumbered so that the matrix is m by m.
- To differentiate time step t from the component of the state vector x, time is placed in (t).
- The first n rows of the matrix are known, if all the nonlinear terms of $F(x)$ are replaced by observables. The bottom $(m\text{-}n)$ rows are unknown and to be tuned.

A Brute-force Method (continued)

- Define Z_t , collect data for t = 0 through N., and set up 2 data matrices.

$$Z_t = \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \\ g_{n+1}(t) \\ \vdots \\ g_{m-n}(t) \end{pmatrix}$$

$$Z_{0|N-1} = \left(Z_0, Z_1, \cdots, Z_{N-1}\right) \in \Re^{m \times N}$$

$$Z_{1|N} = \left(Z_1, Z_2, \cdots, Z_N\right) \in \Re^{m \times N}$$

Note that $Z_{1|N}$ is one time step ahead of $Z_{0|N\text{-}1}$.

- The augmented state equation can be arranged for all the data collectively:

$$Z_{1|N} = A_m Z_{0|N-1}$$

- The least squares solution is given by using the pseudo-inverse of $Z_{0|N\text{-}1}$.

$$A_m = Z_{1|N} Z_{0|N-1}{}^{\#}$$

Limitations to the Brute-force Method

- The above brute-force method is limited in several aspects.
  - The selection of observables are ad hoc.
  - Koopman's theory does not say how to pick observables.
  - We do not know how many observables are required to better approximate the nonlinear dynamics.
- To answer these questions, let us better understand the Koopman Operator theory.

Interpretation of Koopman Operator

- Take transpose of the previous expression, and equate it to the following matrix product

$$Z_{1|N} = A_m Z_{0|N-1} \longrightarrow Z_{1|N}{}^T = Z_{0|N-1}{}^T A_m{}^T \longrightarrow Z_{1|N}{}^T = K_m Z_{0|N-1}{}^T$$

- Treating state variables, too, as observables, we can write the last expression as:

$$\begin{pmatrix} g_1(1) & \cdots & g_i(1) & \cdots \\ g_1(2) & \cdots & g_i(2) & \cdots \\ \vdots & \cdots & \vdots & \cdots \\ g_1(N) & \cdots & g_i(N) & \cdots \end{pmatrix} = \underbrace{\begin{pmatrix} k_{11} & k_{12} & \cdots & k_{1N} \\ k_{21} & k_{22} & \cdots & k_{2N} \\ \vdots & \cdots & \ddots & \vdots \\ k_{N1} & \cdots & \cdots & k_{NN} \end{pmatrix}}_{K_N} \begin{pmatrix} g_1(0) & \cdots & g_i(0) & \cdots \\ g_1(1) & \cdots & g_i(1) & \cdots \\ \vdots & \cdots & \vdots & \cdots \\ g_1(N-1) & \cdots & g_i(N-1) & \cdots \end{pmatrix}$$

$\hat{i}$ $g_i(F(x))$ $\qquad$ $g_i(x)$ $\hat{i}$

- Interestingly, the i^{th} column vector represents a trajectory of the i^{th} observable; the left trajectory: $\{g_i(t) \mid 1 \le t \le N\}$, while the one on the right hand side is $\{g_i(t) \mid 0 \le t \le N-1\}$
- This implies that the above linear transformation with matrix K_N transforms a trajectory to a trajectory, i.e. transformation of functions.

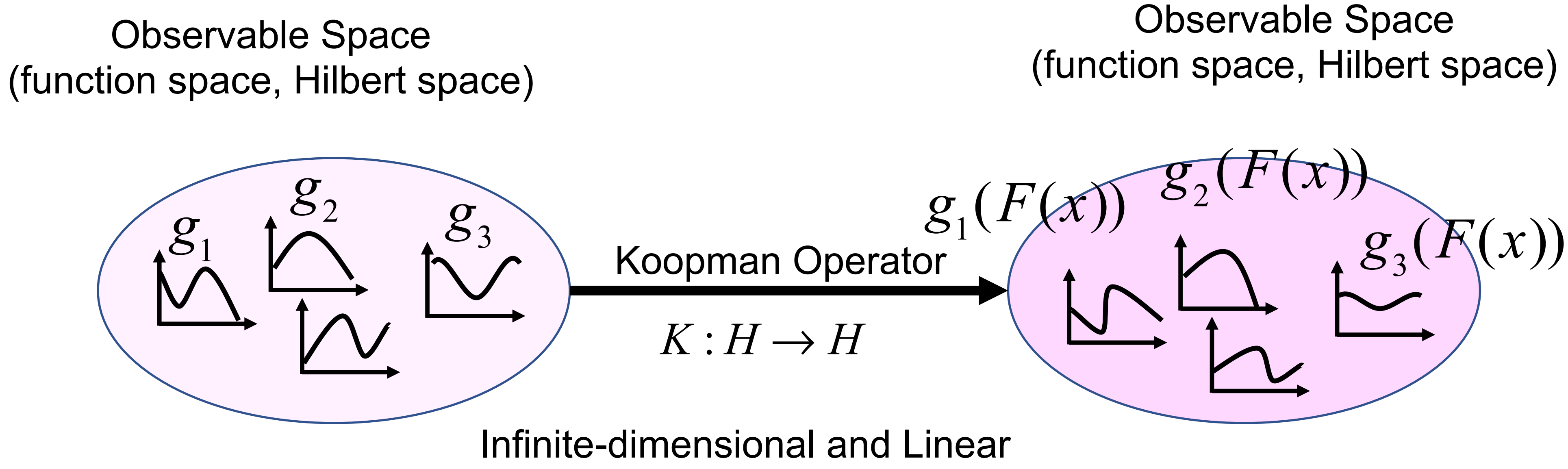
$$g_i(F(x)) = K\, g_i(x)$$

Revisiting the Schematic of Koopman Operator

- Extending the trajectory of each observable to infinite time steps, and the number of observables to infinite, matrix K_N becomes infinite dimensional. Let us denote the infinite-dimensional matrix by $\boldsymbol{K}$, and the observable trajectories as $g_1(x), g_2(x), \cdots$

- We can write the Koopman Operator as a linear transformation of a function to a function.

$$g_i(F(x)) = \boldsymbol{K}\, g_i(x), \quad i = 1, 2, \cdots$$



Comparison between Evolution Operator and Koopman Operator

- Evolution Operator $Z_{1|N} = A_m Z_{0|N-1}$

$$\begin{pmatrix} g_1(t+1) \\ \vdots \\ g_m(t+1) \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11} & \cdots & a_{1,m} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,m} \end{pmatrix}}_{A_m} \begin{pmatrix} g_1(t) \\ \vdots \\ g_m(t) \end{pmatrix}$$

- Koopman Operator $Z_{1|N}{}^T = K_N Z_{0|N-1}{}^T$

$$\underbrace{\begin{pmatrix} g_i(1) \\ g_i(2) \\ \vdots \\ g_i(N) \end{pmatrix}}_{g_i(F(x))} = \underbrace{\begin{pmatrix} k_{11} & k_{12} & \cdots & k_{1N} \\ k_{21} & k_{22} & \cdots & k_{2N} \\ \vdots & \cdots & \ddots & \vdots \\ k_{N1} & \cdots & \cdots & k_{NN} \end{pmatrix}}_{K_N} \underbrace{\begin{pmatrix} g_i(0) \\ g_i(1) \\ \vdots \\ g_i(N-1) \end{pmatrix}}_{g_i(x)} \Rightarrow Kg = g \circ F$$

Koopman Eigenvalues and Eigenfunctions

- The Koopman operator is a linear operator. Therefore, we can characterize it in terms of eigenvalues and eigenfunctions.
- Let λ_j be the j^{th} eigenvalue and $\varphi_j : \Re^n \to \Re$ be the corresponding eigenfunction of Koopman operator $\boldsymbol{K}$.

$$K\varphi_j(x) = \lambda_j \varphi_j(x), \quad j = 1, 2, \cdots$$

- Consider a vector-valued observable $\boldsymbol{g} : \Re^n \to \Re^p$. If each of the p components of $\boldsymbol{g}(x)$ lies in a function space spanned by the eigenfunctions, we can express $\boldsymbol{g}(x)$ as:

$$\boldsymbol{g}(x) = \sum_{j=1}^{\infty} \varphi_j(x) \boldsymbol{v}_j$$

where vector $\boldsymbol{v}_j$ is referred to as Koopman modes of the observable $\boldsymbol{g}(x)$.

- The temporal behaviors of observables can be represented with the Koopman eigenvalues, eigen-functions, and modes.

$$\boldsymbol{g}(x_k) = \sum_{j=1}^{\infty} \varphi_j(x_k) \boldsymbol{v}_j = \sum_{j=1}^{\infty} \varphi_j(F(x_{k-1})) \boldsymbol{v}_j = \sum_{j=1}^{\infty} K\varphi_j(x_{k-1}) \boldsymbol{v}_j = \sum_{j=1}^{\infty} \lambda_j \varphi_j(x_{k-1}) \boldsymbol{v}_j$$

$$= \cdots = \sum_{j=1}^{\infty} \lambda_j^{\,k} \varphi_j(x_0) \boldsymbol{v}_j$$

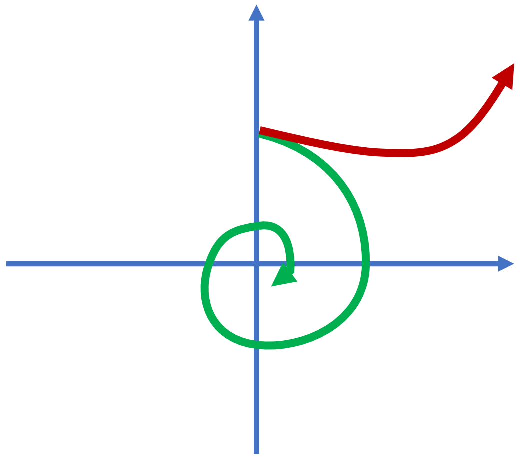
Koopman Eigenvalues and Eigenfunctions

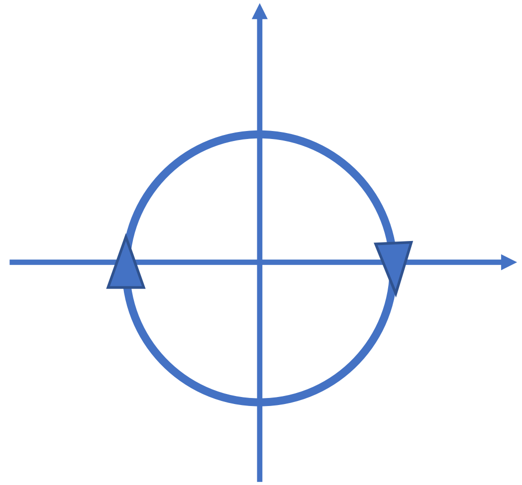
❑ The temporal behaviors of observables can be represented with the Koopman eigenvalues, eigen-functions, and modes.

$$\boldsymbol{g}(x_k) = \sum_{j=1}^{\infty} \lambda_j^{\,k} \varphi_j(x_0) \boldsymbol{v}_j$$

Mode:
Representing the observable w.r.t. eigen functions

Eigen-function:
Bases spanning the function space

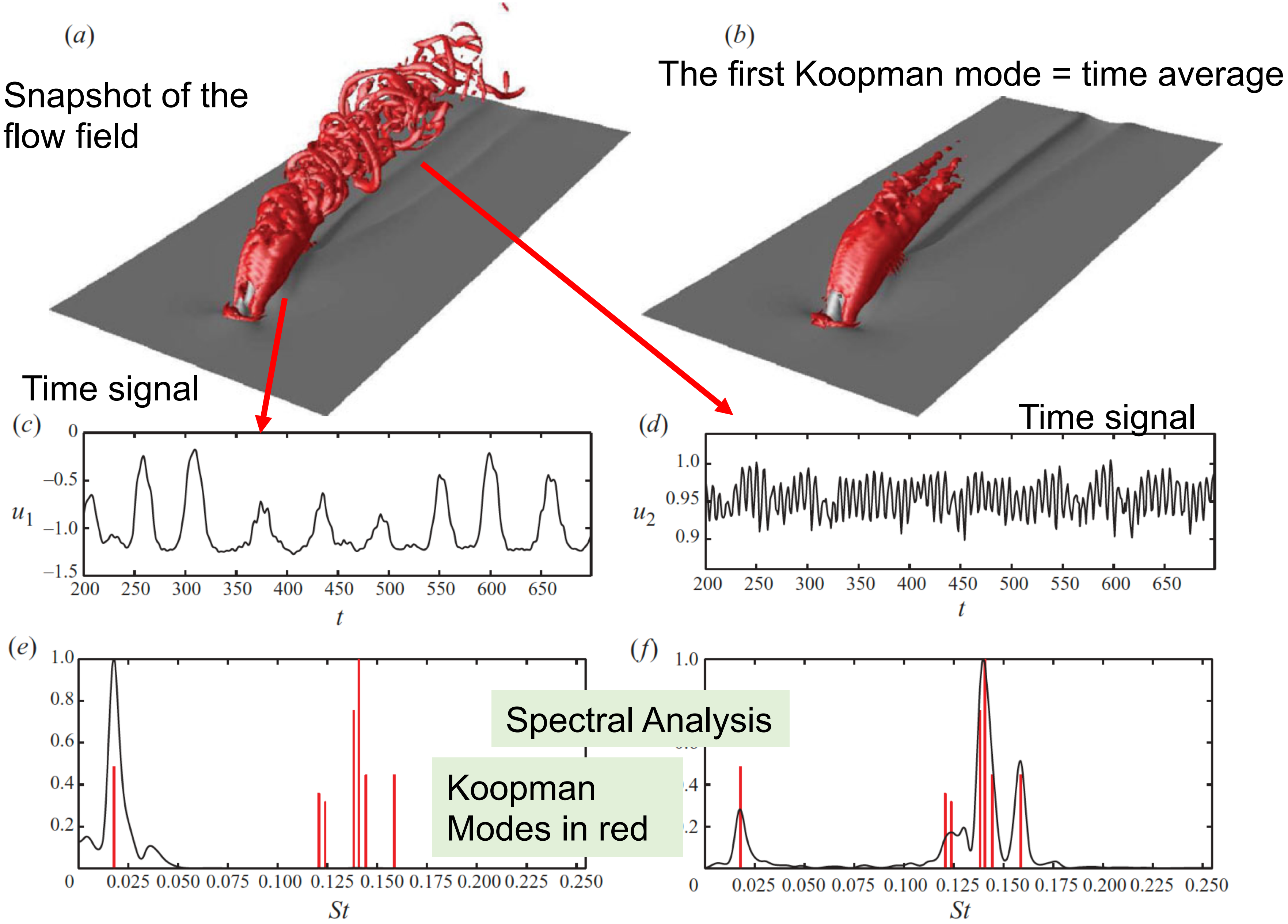




- If one of the eigenvalues is greater than 1, that mode diverges;
- Those modes of $|\lambda_j|<1$ converge; and
- The one on the unit circle evolves on an attractor (limit cycle).

Jet in Cross-Flow

- Koopman Operator was first successfully applied to fluid mechanics.
- Observables are flow velocities measured at various points in space.
- Data are directly analyzed with Koopman operator with regard to eigenvalues, eigen functions, and modes of the linear transform.



C.W. Rowley, et al, “Spectral Analysis of Nonlinear Flows”, Journal of Fluid-mechanics, 2009

Computation of Koopman Eigenvalues and Modes from Data

- Back to the Finite-dimensional Matrix K_m
- Suppose that we truncate the number of observables at m.
- Collecting data for time 0 through m,

$$Z_{1|m}{}^{T} = K_m Z_{0|m-1}{}^{T} \qquad Z_{0|m-1} = \left(Z_0, Z_1, \cdots, Z_{m-1}\right) \in \Re^{m\times m} \qquad Z_t = \begin{pmatrix} g_1(t) \\ \vdots \\ g_m(t) \end{pmatrix}$$

$$Z_{1|m} = \left(Z_1, Z_2, \cdots, Z_m\right) \in \Re^{m\times m}$$

- Note that $Z_{1|m}$ is one time step ahead of $Z_{0|m-1}$. Therefore, we can find

$$\begin{pmatrix} g_1(1) & g_2(1) & \cdots & g_m(1) \\ g_1(2) & g_2(2) & \cdots & g_m(2) \\ \vdots & \vdots & \vdots & \vdots \\ g_1(m) & g_2(m) & \cdots & g_m(m) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & 0 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & \vdots & & 0 & 1 \\ c_0 & c_1 & c_2 & \cdots & c_{m-1} \end{pmatrix}}_{C_m} \begin{pmatrix} g_1(0) & g_2(0) & \cdots & g_m(0) \\ g_1(1) & g_2(1) & \cdots & g_m(1) \\ \vdots & \vdots & \vdots & \vdots \\ g_1(m-1) & g_2(m-1) & \cdots & g_m(m-1) \end{pmatrix}$$

- Note that the Koopman operator is associated with a **Companion matrix** C_m.

$$K_m \leftrightarrow C_m$$

$$\begin{pmatrix} g_1(1) & g_2(1) & \cdots & g_m(1) \\ g_1(2) & g_2(2) & \cdots & g_m(2) \\ \vdots & \vdots & \vdots & \vdots \\ g_1(m) & g_2(m) & \cdots & g_m(m) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & 0 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & \vdots & & 0 & 1 \\ c_0 & c_1 & c_2 & \cdots & c_{m-1} \end{pmatrix} \begin{pmatrix} g_1(0) & g_2(0) & \cdots & g_m(0) \\ g_1(1) & g_2(1) & \cdots & g_m(1) \\ \vdots & \vdots & \vdots & \vdots \\ g_1(m-1) & g_2(m-1) & \cdots & g_m(m-1) \end{pmatrix}$$

- If there exist a set of coefficients c_i that satisfy the last row of the above relationship, the set of observables are complete, forming an Invariant Space.
- In general, the last row is an approximation with some residual r_i.

$$g_i(m) = \sum_{j=0}^{m-1} c_j g_i(j) + r_i, \quad i = 1, \cdots, m$$

- The squared residual $\sum r_i^2$ can be minimized by optimizing the coefficients c_i.

$$(c_1, \cdots, c_m) = \arg\min_{c_1, \cdots, c_m} \sum_{i=1}^{m} \left(g_i(m) - \sum_{j=0}^{m-1} c_j g_i(j) \right)^2$$

- We compute the eigenvalues of the optimized Companion matrix to obtain approximate Koopman eigenvalues.

Ritz Values and Ritz Vectors

- Let λ and w be an eigenvalue and the corresponding eigen vector of the transpose of the optimized Companion matrix C_m.

$$C_m^{\ T} w = \lambda w$$

- From the previous results,

$$Z_{1|m}^{\ T} = Z_{0|m-1}^{\ T} A_m^{\ T} \quad \text{and} \quad Z_{1|m}^{\ T} = K_m Z_{0|m-1}^{\ T} = C_m Z_{0|m-1}^{\ T} \quad \Rightarrow \quad A_m Z_{0|m-1} = Z_{0|m-1} C_m^{\ T}$$

- Post-multiply w to the last expression yields

$$A_m Z_{0|m-1} w = Z_{0|m-1} C_m^{\ T} w = \lambda Z_{0|m-1} w$$

- This implies that $\boldsymbol{v} = Z_{0|m-1}\, \boldsymbol{w}$ is an eigenvector of matrix A_m.
- Eigenvalue λ is called a Ritz value and eigenvector $\boldsymbol{v}$ is a Ritz vector.
- Collectively,

$$C_m^{\ T} = T^{-1} \Lambda T \qquad \text{where} \qquad T^{-1} = (w_1, \cdots, w_m), \Lambda = diag.(\lambda_1, \cdots, \lambda_m)$$

$$V = (v_1, \cdots, v_m) = Z_{0|m-1} T^{-1}$$

Modal Decomposition of Nonlinear Systems

- The Ritz values and vectors and related data-driven methods, such as Dynamic Mode Decomposition (DMD) were developed primarily for linear systems. We now extend them to nonlinear systems.
- Suppose that we have observed a sequence of observables,

$$\boldsymbol{g}(x(t)) \in \Re^m, \quad t = 0,1,2,\cdots,m$$

- Let λ_j*and $\boldsymbol{v}_j$* be the empirical Ritz values and vectors for the data. Then we can show

$$\boldsymbol{g}(x(t)) = \sum_{j=1}^{m} (\lambda_j^*)^t v_j *, \quad t = 0,1,\cdots,m-1$$

$$\boldsymbol{g}(x(m)) = \sum_{j=1}^{m} (\lambda_j^*)^m v_j * + r$$

- Where r is the residual after optimization, and $\boldsymbol{v}_j$* is scaled by the constant values $\varphi_j(x(0))$ in comparison to the previous expression. $\boldsymbol{g}(x_k) = \sum_{j=1}^{\infty} \lambda_j^k \varphi_j(x_0) \boldsymbol{v}_j$

Reflection

- Koopman Operator is weird, but powerful.
- Nonlinear autonomous systems can be linearized in an infinite dimensional space.
- It acts on functions. It is infinite dimensional.
- Since it is linear, spectral analysis with eigenvalues and eigenfunctions is applicable to Koopman Operator.
- Data-driven methods are available for obtaining eigenvalues, eigen vectors, and modes directly from data.

- The exact linearization has been guaranteed only for autonomous systems (no control inputs) in infinite dimensional spaces.
- Practical methods will be discussed in the final lecture this Wednesday.