Wavelet-based DMD in the context of Extended DMD

Cankat Tilki and Serkan Gugercin

Department of Mathematics, Virginia Tech

Nonlinear Model Reduction for Control

22-26 May 2023, Blacksburg, VA

Table of Contents

Table of Contents

Motivation

Let $\mathbf{x} \in \mathbb{R}^N$, $F : \mathbb{R}^N \to \mathbb{R}^N$ and consider the dynamical system given as

$$\mathbf{x}(t_{k+1}) = F(\mathbf{x}(t_k)). \tag{1}$$

Can we model the dynamics (1) if we are given *only some samples* of the state $\{\mathbf{x}(t_k)\}$ without knowing F?

- Koopman Operator Theory together with Extended Dynamic Mode Decomposition provide a powerful tool to achieve this goal.
- Trade-off between finite-dimensional nonlinearity vs infinite-dimensional linearity

Koopman Operator

Given $\mathbf{x}(t_{k+1}) = F(\mathbf{x}(t_k))$ and for $\psi : \mathbb{R}^N \to \mathbb{C}$ consider:

$$\psi(F(\mathbf{x}(t_k)) = \psi \circ F(\mathbf{x}(t_k)) = \psi(\mathbf{x}(t_{k+1})).$$

We project our dynamics from the *state space* $\mathbf{x}$ to *observable space* ψ.

Dynamical System on Observables

Let $\psi : \mathbb{R}^N \to \mathbb{C}$ and define a new dynamical system:

$$\psi(\mathbf{x}(t_{k+1})) = \psi \circ F(\mathbf{x}(t_k)) = \mathcal{K}[\psi](\mathbf{x}(t_k)).$$

We call $\mathcal{K}[\psi] := \psi \circ F$ as the *Koopman operator*.

Note: Koopman operator $\mathcal{K}$ is linear. Hence we lift a *finite dimensional nonlinear* problem to an *infinite dimensional linear* problem.

Recovering Dynamics via Koopman Operator

- Let (μ_i, ϕ_i) be the eigenpairs of the Koopman operator $\mathcal{K}$:

$$\mathcal{K}[\phi_i] = \mu_i \phi_i$$

- Define $\mathbf{g}(\mathbf{x}) = \mathbf{x}$ and assume $\mathbf{g} \in \text{span}\{\phi_i\}$. Let $\nu_i \in \mathbb{R}^N$ be such that

$$\mathbf{g}(\mathbf{x}) = \sum_{i=1}^{L} \nu_i \phi_i(\mathbf{x})$$

- Then we can reconstruct the original dynamics F as

$$F(\mathbf{x}) = \mathcal{K}[\mathbf{g}](\mathbf{x}) = \mathcal{K}\left[\sum_{i=1}^{L} \nu_i \phi_i\right](\mathbf{x}) = \sum_{i=1}^{L} \nu_i \mathcal{K}[\phi_i](\mathbf{x}) = \sum_{i=1}^{L} \nu_i \mu_i \phi_i(\mathbf{x}).$$

For F we need: *coefficients* ν_i (Koopman modes) and *eigenpairs* (μ_i, ϕ_i).

Approximating (ν_i, μ_i, ϕ_i)

- We are given the action of $\mathcal{K}$ on the observables $\psi_1, \ldots, \psi_K$
- For another observable φ, write $\varphi \approx \sum_{k=1}^{K} \psi_k a_k$. Then

$$\mathcal{K}[\varphi] = \varphi \circ F \approx \psi^T \mathbf{K}\mathbf{a}, \quad \mathbf{K} \in \mathbb{R}^{K \times K} \text{ and } \mathbf{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_K \end{bmatrix}^T.$$

- Let (λ_i, ξ_i) be the eigenpairs of $\mathbf{K}$.Then we approximate (μ_i, ϕ_i) as

$$\mu_i \approx \lambda_i \qquad \phi_i \approx \sum_{k=1}^{K} (\xi_i)_k \psi_k$$

- Similarly we can approximate coefficients ν_i by using the left eigenvectors of $\mathbf{K}$ and vectors $\mathbf{b}_i \in \mathbb{R}^K$ defined by the expansion

$$\mathbf{x}_i = \sum_{k=1}^{K} (\mathbf{b}_i)_k \psi_k(\mathbf{x})$$

Finding $\mathbf{K}$: EDMD Algorithm [WKR14]

Given observation data $\{\psi_k(\mathbf{x}(t_i))\}$ for $i = 0, 1, \ldots, T$ and $k = 1, \ldots, K$:

1. Construct $\mathbf{K}$ by solving the optimization problem

$$\mathbf{K} = \underset{\hat{\mathbf{K}} \in \mathbb{R}^{K \times K}}{\operatorname{argmin}} \left\| \mathcal{K}[\psi^T](\mathbf{x}(t_i)) - \psi(\mathbf{x}(t_i))^T \hat{\mathbf{K}} \right\|_F \qquad \psi = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_K \end{bmatrix}$$

$$= \underset{\hat{\mathbf{K}} \in \mathbb{R}^{K \times K}}{\operatorname{argmin}} \| \underbrace{\psi(\mathbf{x}(t_{i+1}))^T}_{T \times K} - \underbrace{\psi(\mathbf{x}(t_i))^T}_{T \times K} \hat{\mathbf{K}} \|_F$$

2. Recover Koopman modes and eigenpairs $(\nu_k, \mu_k, \phi_k)_{k=1}^K$.
3. Recover the original system $F(\mathbf{x}(t_i)) \approx \sum_{k=1}^K \nu_k \mu_k \phi_k(\mathbf{x}(t_i))$

<u>Original Dynamics:</u> $\mathbf{x}(t_{i+1}) = F(\mathbf{x}(t_i)) \approx \sum_{k=1}^{K} \nu_k \mu_k \phi_k(\mathbf{x}(t_i))$

Dynamic Mode Decomposition as a special case of EDMD

- Dynamic Mode Decomposition Algorithm: [Sch10, RMBSH09]
- Choose $\psi_k = e_k^T$ and $k = 1, \cdots, N$. So we are now given $\{\mathbf{x}_k(t_i)\}_{i=0}^T$.
- Then EDMD minimization problem becomes:

$$\begin{aligned}\mathbf{K} &= \underset{\hat{\mathbf{K}}\in\mathbb{R}^{K\times K}}{\operatorname{argmin}} \| \underbrace{\psi(\mathbf{x}(t_{i+1}))^T}_{T\times K} - \underbrace{\psi(\mathbf{x}(t_i))^T}_{T\times K} \hat{\mathbf{K}} \|_F \\ &= \underset{\hat{\mathbf{K}}\in\mathbb{R}^{K\times K}}{\operatorname{argmin}} \| \underbrace{\mathbf{x}(t_{i+1})^T}_{T\times K} - \underbrace{\mathbf{x}(t_i)^T}_{T\times K} \hat{\mathbf{K}} \|_F\end{aligned}$$

- This minimization problem is equivalent to

$$\mathbf{K} = \underset{\widetilde{\mathbf{K}}\in\mathbb{R}^{K\times K}}{\operatorname{argmin}} \left\| \mathbf{x}(t_{i+1}) - \widetilde{\mathbf{K}}^T \mathbf{x}(t_i) \right\|_F$$

- With these specific observables $\psi_k = e_k^T$ EDMD algorithm recovers the *Dynamic Mode Decomposition* algorithm.

ioDMD [BHM18]

Given time series data $\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{u}(t_i)$ from the unknown input/output dynamical system

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t),$$

ioDMD aims to to get the best linear least squares approximation

$$\mathbf{x}(t_{i+1}) \approx \mathbf{A}\mathbf{x}(t_i) + \mathbf{B}\mathbf{u}(t_i)$$
$$\mathbf{y}(t_i) \approx \mathbf{C}\mathbf{x}(t_i) + \mathbf{D}\mathbf{u}(t_i)$$

To do this, similar to DMD, we solve the minimization problem

$$\Gamma = \underset{\hat{\Gamma} \in \mathbb{R}^{(N+D)\times(N+M)}}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{Y}_0 \end{bmatrix} - \hat{\Gamma} \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{U}_0 \end{bmatrix} \right\|_F$$

and get the closed form solution

$$\Gamma = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{Y}_0 \end{bmatrix} \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{U}_0 \end{bmatrix}^{\dagger}.$$

Table of Contents

Why WDMD?

- Assume access to only the output data $\mathbf{y}(t_i) \in \mathbb{R}^D$ of an unknown dynamical system with forcing $\mathbf{u}(t) \in \mathbb{R}^M$ with state $\mathbf{x}(t) \in \mathbb{R}^N$:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t).$$

- Standard methods (e.g., ioDMD) usually require having the full state data $\mathbf{x}(t_i)$. But in most cases we only have $\mathbf{y}(t_i) = \mathbf{C}\mathbf{x}(t_i)$.

Solution: Construct *auxiliary* states $\mathbf{z}$ from the output samples $\mathbf{y}(t_i)$ and apply known methods to the auxiliary dynamical system

$$\mathbf{z}(t_{i+1}) = f_{\mathbf{z}}(\mathbf{z}(t_i), \mathbf{u}(t_i))$$
$$\mathbf{y}(t_i) = \mathbf{C}_{\mathbf{z}}\mathbf{z}(t_i).$$

How do we construct a *"good auxiliary state"*?

Wavelet Transform

- We will use wavelet transform of $\mathbf{y}(t)$ to create the auxiliary state.
- Assume we are given a mother wavelet $\Psi : \mathbb{R} \to \mathbb{R}$ and an L_2 function $f : \mathbb{R} \to \mathbb{R}$.
- Define $\Psi_j^k(t) = \Psi\left(\frac{t}{2^j} - k\Delta t\right)$ and $\omega_j^k(f)$:

$$\omega_j^k(f) = \langle f, \Psi_j^k \rangle_2 = \left\langle f, \Psi\left(\frac{t}{2^j} - k\Delta t\right)\right\rangle_2 = \frac{1}{2^{j/2}} \int_{-\infty}^{\infty} f(t)\Psi^*\left(\frac{t}{2^j} - k\Delta t\right) dt$$

- We can reconstruct f as

$$f(t) = \sum_{j,k} \omega_j^k(f)\Psi_j^k(t)$$

- Requires f but only have $\{f(t_i)\}$ since the dynamics is unknown.
- Need to approximate $\omega_j^k(f)$ from time series data $\{f(t_i)\}$.

Maximal Overlap Discrete Wavelet Transform [PW13]

For $f : \mathbb{R} \to \mathbb{R}$ assume we are given:

- time series data $\mathbf{f} = [\ f(t_0) \quad f(t_1) \quad \cdots \quad f(t_{T-1})\ ]^T \in \mathbb{R}^T$
- high/low pass filters $\mathbf{h}_j, \mathbf{g}_j \in \mathbb{R}^T$

1. Construct projections $\mathcal{W}_j, \mathcal{V}_j \in \mathbb{R}^{T\times T}$ from $\mathbf{h}_j, \mathbf{g}_j$ respectively.
2. j^{th} (discrete) wavelet and scaling coefficients of f:

$$\omega_j(\mathbf{f}) = \mathcal{W}_j\mathbf{f} \quad \text{and} \quad \theta_j(\mathbf{f}) = \mathcal{V}_j\mathbf{f}.$$

MODWT Recovery

Define $d_j^{(i)}(\mathbf{f}) = e_{i+1}^T\mathcal{W}_j^T\omega_j(\mathbf{f})$ and $s_j^{(i)}(\mathbf{f}) = e_{i+1}^T\mathcal{V}_j^T\theta_j(\mathbf{f})$.

$$\mathbf{f}(t_i) = \sum_{j=1}^{J} e_{i+1}^T\mathcal{W}_j^T\omega_j(\mathbf{f}) + e_{i+1}^T\mathcal{V}_J^T\theta_J(\mathbf{f}) = \sum_{j=1}^{J} d_j^{(i)}(\mathbf{f}) + s_J^{(i)}(\mathbf{f})$$

WDMD Algorithm [KGT21]

- Without loss of generality take $D = 1$ (number of outputs).

Given $\mathbf{Y} = \begin{bmatrix} \mathbf{y}(t_0) & \mathbf{y}(t_1) & \cdots & \mathbf{y}(t_{T-1}) \end{bmatrix} \in \mathbb{R}^T$ and $\mathbf{u}(t) \in \mathbb{R}^M$.

1. Construct $d_j(\mathbf{Y})$ and $s_J(\mathbf{Y})$ by MODWT.
2. Construct the samples of the lifted (auxiliary) state:

$$\mathbf{z}(t_i) = \mathbf{w}(t_i) = \begin{bmatrix} d_1^{(i)}(\mathbf{Y}) & \cdots & d_J^{(i)}(\mathbf{Y}) & s_J^{(i)}(\mathbf{Y}) \end{bmatrix}^T \in \mathbb{R}^{J+1}$$

3. Approximate the lifted dynamical system by a linear dynamical system

$$\begin{aligned} \mathbf{z}(t_{i+1}) &= f_{\mathbf{z}}(\mathbf{z}(t_i), \mathbf{u}(t_i)) \\ \mathbf{y}(t_i) &= \mathbf{1}_{J+1}\mathbf{z}(t_i) \end{aligned} \approx \begin{aligned} \mathbf{z}(t_{i+1}) &= \mathbf{A_z}\mathbf{z}(t_i) + \mathbf{B_z}\mathbf{u}(t_i)) \\ \mathbf{y}(t_i) &= \mathbf{C_z}\mathbf{z}(t_i) + \mathbf{D_z}\mathbf{u}(t_i) \end{aligned}$$

How to Obtain $\mathbf{A_z}, \mathbf{B_z}, \mathbf{C_z}, \mathbf{D_z}$

Given $\mathbf{y}(t_i), \mathbf{u}(t_i)$, form $\mathbf{z}(t_i)$, auxiliary state samples. Define

$$\begin{aligned}
\mathbf{Y}_0 &= \begin{bmatrix}\mathbf{y}(t_0) & \mathbf{y}(t_1) & \cdots & \mathbf{y}(t_{K-1})\end{bmatrix}.\\
\mathbf{U}_0 &= \begin{bmatrix}\mathbf{u}(t_0) & \mathbf{u}(t_1) & \cdots & \mathbf{u}(t_{K-1})\end{bmatrix}.\\
\mathbf{Z}_0 &= \begin{bmatrix}\mathbf{z}(t_0) & \mathbf{z}(t_1) & \cdots & \mathbf{z}(t_{K-1})\end{bmatrix}.\\
\mathbf{Z}_1 &= \begin{bmatrix}\mathbf{z}(t_1) & \mathbf{z}(t_2) & \cdots & \mathbf{z}(t_K)\end{bmatrix} = f_\mathbf{z}(\mathbf{Z}_0, \mathbf{U}_0).
\end{aligned}$$

Algorithm

Find the best $\mathbf{A_z}, \mathbf{B_z}, \mathbf{C_z}, \mathbf{D_z}$ such that

$$\begin{aligned}
\mathbf{z}(t_{i+1}) &= f_\mathbf{z}(\mathbf{z}(t_i), \mathbf{u}(t_i))\\
\mathbf{y}(t_i) &= \mathbf{1}_{J+1}\mathbf{z}(t_i)
\end{aligned}
\approx
\begin{aligned}
\mathbf{z}(t_{i+1}) &= \mathbf{A_z}\mathbf{z}(t_i) + \mathbf{B_z}\mathbf{u}(t_i))\\
\mathbf{y}(t_i) &= \mathbf{C_z}\mathbf{z}(t_i) + \mathbf{D_z}\mathbf{u}(t_i)
\end{aligned}$$

$$\Phi = \begin{bmatrix}\mathbf{A_z} & \mathbf{B_z}\\ \mathbf{C_z} & \mathbf{D_z}\end{bmatrix} = \underset{\hat{\mathbf{A}},\hat{\mathbf{B}},\hat{\mathbf{C}},\hat{\mathbf{D}}}{\operatorname{argmin}} \left\| \begin{bmatrix}\mathbf{Z}_1\\ \mathbf{Y}_0\end{bmatrix} - \begin{bmatrix}\hat{\mathbf{A}} & \hat{\mathbf{B}}\\ \hat{\mathbf{C}} & \hat{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{Z}_0\\ \mathbf{U}_0\end{bmatrix}\right\|_F \Longrightarrow \Phi = \begin{bmatrix}\mathbf{Z}_1\\ \mathbf{Y}_0\end{bmatrix}\begin{bmatrix}\mathbf{Z}_0\\ \mathbf{U}_0\end{bmatrix}^\dagger$$

Can we interpret WDMD as EDMD analytically?

Without loss of generality, let $t_{i+1} = t_i + \Delta t$.
Assume there is no input: $\mathbf{u}(t) = 0$.

- In WDMD we use the auxiliary state data $\mathbf{z}(t_i) = \left\{ \begin{bmatrix} d_1^{(i)}(\mathbf{Y}) \\ \vdots \\ d_J^{(i)}(\mathbf{Y}) \\ s_J^{(i)}(\mathbf{Y}) \end{bmatrix} \right\}$
- EDMD requires observable evaluations:

$$\{\psi_j(\mathbf{x}(t_i))\} \approx \{d_j^{(i)}(\mathbf{Y})\} \qquad (2)$$

How to choose the right observables ψ_j to satisfy (2)?

Observables for WDMD

- Wavelet Transform:

$$\omega_j^k(\mathbf{y}(t)) = \frac{1}{2^{j/2}} \int_{-\infty}^{\infty} \Psi_j^k(t)\mathbf{y}(t)dt \to \mathbf{y}(t) = \sum_k \sum_j \omega_j^k \Psi_j^k(t)$$

- Define the *WDMD observables*: $\psi_j(\mathbf{x}(t)) = \omega_j^0(\mathbf{y}(t))\Psi_j^0(t)$

$$\mathcal{K}[\psi_j](\mathbf{x}(t)) = \omega_j^0(\mathbf{y}(t+\Delta t))\Psi_j^0(t+\Delta t) = \omega_j^1(\mathbf{y}(t))\Psi_j^1(t)$$

- Approximate by MODWT:

$$\psi_j(\mathbf{x}(t)) = \omega_j^0(\mathbf{y}(t))\Psi_j^0(t) \approx d_j^{(0)}(\mathbf{Y}) \;\text{ and }\; s_J^{(0)}(\mathbf{Y})$$

Lemma: WDMD as EDMD

Assume we are given only the time series data $\mathbf{Y} = \begin{bmatrix} \mathbf{y}(t_0) & \cdots & \mathbf{y}(t_T) \end{bmatrix}$ from an unknown input-output dynamical system with forcing $\mathbf{u}(t) \in \mathbb{R}^M$ with state $\mathbf{x}(t) \in \mathbb{R}^N$.

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t).$$

Then WDMD can be interpreted as a specific case of EDMD algorithm where the observables are chosen as

$$\psi_j(\mathbf{x}) = \omega_j^0(\mathbf{y}(t))\Psi_j^0(t) = \left(\frac{1}{2^{j/2}} \int_{-\infty}^{\infty} \Psi_j^k(t)\mathbf{y}(t)dt\right) \Psi_j^0(t)$$

Moreover if we let $d_j(\mathbf{Y})$ and $s_j(\mathbf{Y})$ be the MODWT coefficients for the time series $\mathbf{Y}$ we can approximate these observables $\psi_j(\mathbf{x})$ as

$$\psi_j(\mathbf{x}) \approx d_j^{(0)}(\mathbf{Y}) \quad \text{and} \quad \psi_{J+1}(\mathbf{x}(t)) \approx s_j^{(0)}(\mathbf{Y}).$$

Table of Contents

SIR Model

The SIR model is given as

$$\frac{dS}{dt} = -\frac{\beta SI}{N}$$
$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$

- $S(0) = 90, I(0) = 10$ and $R(0) = 0$.
- $N = S(0) + I(0) + R(0) = 100$.
- We compare WDMD to delay DMD [YZWL21].

Numerical Results on the SIR Model

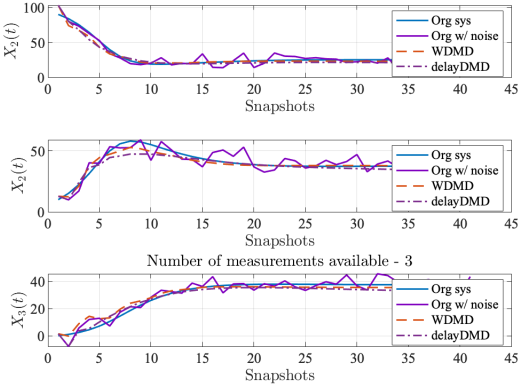

Figure: Fully Observed System with Noise, J = 2

Results

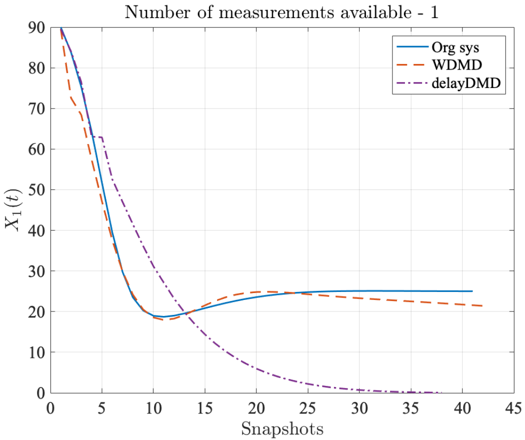

Figure: Partially Observed System with Noise, J = 5

Conclusions and Future Work

- We analytically connected WDMD to EDMD
- Observables are obtained via the Wavelet transform
- Future work:
  - A more through analysis of the impact of WDMD on noisy data
  - A more detailed comparison with other methods such as Delay DMD.
  - Incorporating the input $\mathbf{u}(t)$ and working with a bilinear model

References

- M. Krishnan, S. Gugercin, and P. Tarazaga (2022). Mechanical Systems and Signal Processing.
- J. Tu, C. Rowley, D. Luchtenburg, S. Brunton, and N. J. Kutz, "On dynamic mode decomposition: Theory and applications," Journal of Computational Dynamics, vol. 1, no. 2, pp. 391–421, 2014.
- D. B. Percival and A. T. Walden, Wavelet Methods for Time Series Analysis, vol. 4. Cambridge University Press, 2000.
- Williams, Matthew & Kevrekidis, Ioannis & Rowley, Clarence. (2014). A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. Journal of Nonlinear Science.

References

- P. J. Schmid. "Dynamic mode decomposition of numerical and experimental data." Journal of fluid mechanics, vol. 656, pp. 5–28, 2010.
- Rowley, Clarence W., Igor Mezić, Shervin Bagheri, Philipp Schlatter and Dan S. Henningson. "Spectral analysis of nonlinear flows." Journal of Fluid Mechanics 641 (2009): 115 - 127.
- P. Benner, C. Himpe, and T. Mitchell, "On Reduced Input-Output Dynamic Mode Decomposition," Advances in Computational Mathematics, vol. 44, pp. 1751–1768, Dec 2018.
- Y. Yuan, K. Zhou, W. Zhou, X. Wen, and Y. Liu, "Flow prediction using dynamic mode decomposition with time-delay embedding based on local measurement," Physics of Fluids, vol. 33, no. 9, p. 095109, 2021.

Recovering Eigenpairs of $\mathcal{K}$ (if $\mathbf{K}$ is given)

Assume we have $\phi_j \approx \sum_{k=1}^{K} \psi_k \xi_{kj} = \psi^T \xi_j$ and recall $\mathcal{K}[\varphi] \approx \psi^T \mathbf{K}\mathbf{a}$ so

$$\mathcal{K}[\phi_j] \approx \mathcal{K}\left[\sum_{k=1}^{K} \psi_k \xi_{kj}\right] = \sum_{k=1}^{K} \mathcal{K}[\psi_k]\xi_{kj} \approx \psi^T \mathbf{K} \xi_j$$

Since $\mathcal{K}[\phi_j] = \mu_j \phi_j \approx \psi^T \mu_j \xi_j$, we have

$$\psi^T \mathbf{K} \xi_j \approx \mathcal{K}[\phi_j] \approx \psi^T \mu_j \xi_j, \quad \psi = \begin{bmatrix} \psi_1 & \psi_2 & \cdots & \psi_K \end{bmatrix}^T.$$

(μ_j, ξ_j): eigenpairs of $\mathbf{K} \to (\mu_j, \psi^T \xi_j)$: approximate eigenpairs of $\mathcal{K}$.

In matrix form: $\Xi_{ij} = \xi_{ij}$, $\mathbf{w}_j :=$ left eigenvectors of $\mathbf{K}$ and $\mathbf{w}_i^* \xi_j = \delta_{ij}$.

$$\psi^T \Xi \approx \phi^T \quad \Longleftrightarrow \quad \psi^T \approx \phi^T \mathbf{W}^*, \quad \mathbf{W} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_K \end{bmatrix}$$

Recovering Koopman Modes

Approximate the full state observable, $\mathbf{g}(\mathbf{x}) = \mathbf{x}$, as

$$g_n = e_n^T \approx \sum_{k=1}^{K} b_{kn}\psi_k, \quad \mathbf{g} = \begin{bmatrix} g_1 & g_2 & \cdots & g_N \end{bmatrix}^T$$

In the matrix format we have

$$\mathbf{g} \approx \mathbf{B}^T\psi, \quad \mathbf{B}_{ij} = b_{ij}.$$

Use $\phi^T\mathbf{W}^* \approx \psi^T$ to recover the Koopman modes ν_i as rows of $\mathbf{W}^*\mathbf{B}$:

$$\mathbf{g} \approx \mathbf{B}^T\psi \approx \mathbf{B}^T\mathbf{W}\phi$$

Recall that ν_i is defined as $\mathbf{g} = \sum_{i=1}^{L} \nu_i\phi_i = \mathbf{V}\phi$.

But how do we obtain K?

Proof for WDMD as EDMD

We need to prove: $\{\psi_j(\mathbf{x}(t_i))\} \approx \{d_j^{(i)}(\mathbf{Y})\}$ and the minimization problems being equivalent.

- Observables conforming to data $\psi_j(\mathbf{x}(t_i)) = \mathcal{K}^i[\psi_j](\mathbf{x}(t_0))$: For $i = 0$ we have the approximation by MODWT. Then for $i = 1, \cdots, T$ we have

$$\omega_j^0(\mathbf{C}\mathbf{x}(t + i\Delta t)) = \int_{-\infty}^{\infty} \Psi^* \left(\frac{t}{2^j}\right) \mathbf{y}(t + i\Delta t) dt =$$

$$= \int_{-\infty}^{\infty} \Psi^* \left(\frac{\hat{t} - i\Delta t}{2^j}\right) \mathbf{y}(\hat{t}) d\hat{t} = \int_{-\infty}^{\infty} \Psi_j^i(\hat{t}) \mathbf{y}(\hat{t}) d\hat{t} = \omega_j^i(\mathbf{C}\mathbf{x}(t))$$

- Minimization problems are equivalent: The EDMD minimization problem for this choice of observables becomes

$$\underset{\hat{\mathbf{K}}}{\operatorname{argmin}} \|\mathbf{Z}_1^T - \mathbf{Z}_0^T \hat{\mathbf{K}}\|_F \equiv \underset{\widetilde{\mathbf{K}}}{\operatorname{argmin}} \|\mathbf{Z}_1 - \widetilde{\mathbf{K}}^T \mathbf{Z}_0\|_F$$