Learning Dynamics Models with Stable Invariant Sets

Naoya Takeishi (HES-SO/RIKEN), Yoshinobu Kawahara (KyushuU/RIKEN)

The 35th AAAI, February 2021

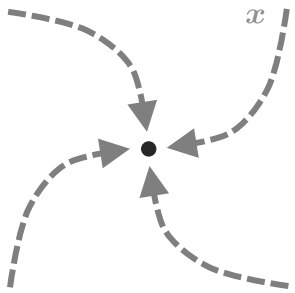
Problem Setting

- Estimate a model of (continuous-time) dynamical system $\dot{\boldsymbol{x}} \approx \boldsymbol{f}(\boldsymbol{x}), \quad \boldsymbol{f} : \mathcal{X} \to \mathbb{R}^d$
  i. given sequence $(\boldsymbol{x}_{t_1}, \boldsymbol{x}_{t_2}, \ldots, \boldsymbol{x}_{t_m}), \ \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$; OR
  ii. given pairs of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$

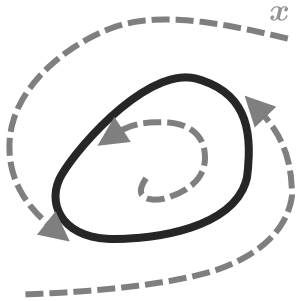
Problem Setting

- Estimate a model of (continuous-time) dynamical system $\dot{\boldsymbol{x}} \approx \boldsymbol{f}(\boldsymbol{x}),\ \ \boldsymbol{f} : \mathcal{X} \to \mathbb{R}^d$
  - i. given sequence $(\boldsymbol{x}_{t_1}, \boldsymbol{x}_{t_2}, \ldots, \boldsymbol{x}_{t_m}),\ \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$; OR
  - ii. given pairs of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$

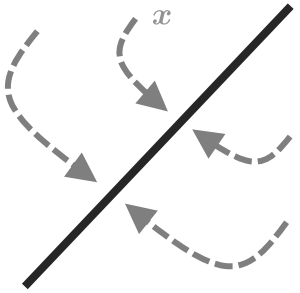
- Ensuring the model's invariance & (asymptotic) stability



stable equilibrium



stable limit cycle

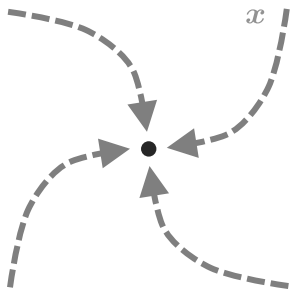


line attractor

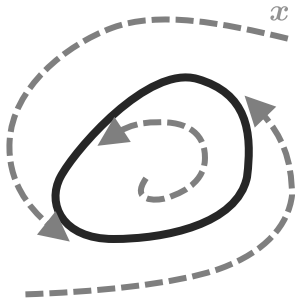
Problem Setting

- Estimate a model of (continuous-time) dynamical system $\dot{\boldsymbol{x}} \approx \boldsymbol{f}(\boldsymbol{x}),\ \ \boldsymbol{f} : \mathcal{X} \to \mathbb{R}^d$
  - i. given sequence $(\boldsymbol{x}_{t_1}, \boldsymbol{x}_{t_2}, \ldots, \boldsymbol{x}_{t_m}),\ \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$; OR
  - ii. given pairs of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$

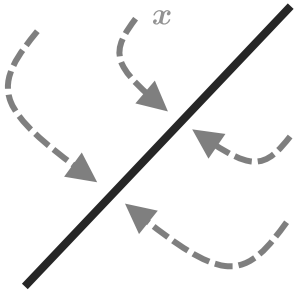
- Ensuring the model's invariance & (asymptotic) stability



stable equilibrium



stable limit cycle



line attractor

- useful if we have some prior knowledge of the target dynamics
- e.g., model should have a stable limit cycle for self-oscillating phenomena

Review: Dynamics model with stable equilibrium

[Manek & Kolter 2019]

- Let $\dot{x} = \hat{f}(x)$ be a base dynamics model (e.g., $\hat{f}$: neural net)

Review: Dynamics model with stable equilibrium

[Manek & Kolter 2019]

- Let $\dot{\boldsymbol{x}} = \hat{f}(x)$ be a base dynamics model (e.g., $\hat{f}$: neural net)
- Modify $\hat{f}$ so that equilibrium $\boldsymbol{x} = 0$ becomes asymptotically stable:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{cases} \hat{f}(x) - \frac{\nabla V(\boldsymbol{x})^\mathsf{T} \hat{f}(x)}{\|V(\boldsymbol{x})\|_2^2} \nabla V(\boldsymbol{x}), & \text{if } \nabla V(\boldsymbol{x})^\mathsf{T} \hat{f}(x) > 0, \\ \hat{f}(x), & \text{otherwise.} \end{cases}$$

Review: Dynamics model with stable equilibrium

[Manek & Kolter 2019]

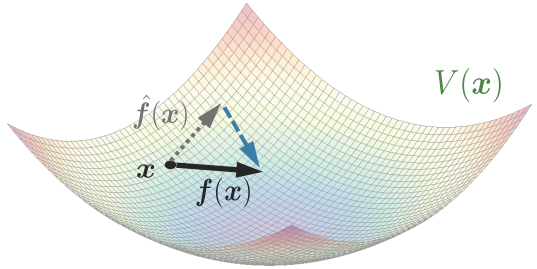
- Let $\dot{\boldsymbol{x}} = \hat{f}(x)$ be a base dynamics model (e.g., $\hat{f}$: neural net)
- Modify $\hat{f}$ so that equilibrium $\boldsymbol{x} = 0$ becomes asymptotically stable:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{cases} \hat{f}(x) - \frac{\nabla V(\boldsymbol{x})^\mathsf{T} \hat{f}(x)}{\|V(\boldsymbol{x})\|_2^2} \nabla V(\boldsymbol{x}), & \text{if } \nabla V(\boldsymbol{x})^\mathsf{T} \hat{f}(x) > 0, \\ \hat{f}(x), & \text{otherwise.} \end{cases}$$

- $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is *learnable* Lyapunov (candidate) function

$$V(\boldsymbol{x}) = \sigma\left(q(\boldsymbol{x}) - q(\boldsymbol{x}_\mathrm{e})\right) + \varepsilon \|\boldsymbol{x} - \boldsymbol{x}_\mathrm{e}\|_2^2$$

  - q : input-convex neural net [Amos+ 2017]
  - σ : convex non-decreasing function
  - designed such that $V(\boldsymbol{x} = \mathbf{0}) = 0$, $V(\boldsymbol{x} \neq \mathbf{0}) > 0$



Review: Dynamics model with stable equilibrium

[Manek & Kolter 2019]

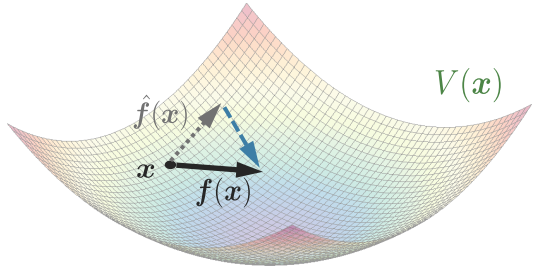
- Let $\dot{\boldsymbol{x}} = \hat{f}(x)$ be a base dynamics model (e.g., $\hat{f}$: neural net)
- Modify $\hat{f}$ so that equilibrium $\boldsymbol{x} = 0$ becomes asymptotically stable:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{cases} \hat{f}(x) - \frac{\nabla V(\boldsymbol{x})^\mathsf{T} \hat{f}(x)}{\|\nabla V(\boldsymbol{x})\|_2^2} \nabla V(\boldsymbol{x}), & \text{if } \nabla V(\boldsymbol{x})^\mathsf{T} \hat{f}(x) > 0, \\ \hat{f}(x), & \text{otherwise.} \end{cases}$$

- $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is *learnable* Lyapunov (candidate) function

$$V(\boldsymbol{x}) = \sigma\left(q(\boldsymbol{x}) - q(\boldsymbol{x}_\mathrm{e})\right) + \varepsilon \|\boldsymbol{x} - \boldsymbol{x}_\mathrm{e}\|_2^2$$

  - q : input-convex neural net [Amos+ 2017]
  - σ : convex non-decreasing function
  - designed such that $V(\boldsymbol{x} = \mathbf{0}) = 0$, $V(\boldsymbol{x} \neq \mathbf{0}) > 0$



- ☹ We may want to handle more general stable invariant sets

Proposed method

- Problem: Learn dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ that has a stable invariant set $S \subset \mathcal{X}$
  - We need to learn S at the same time

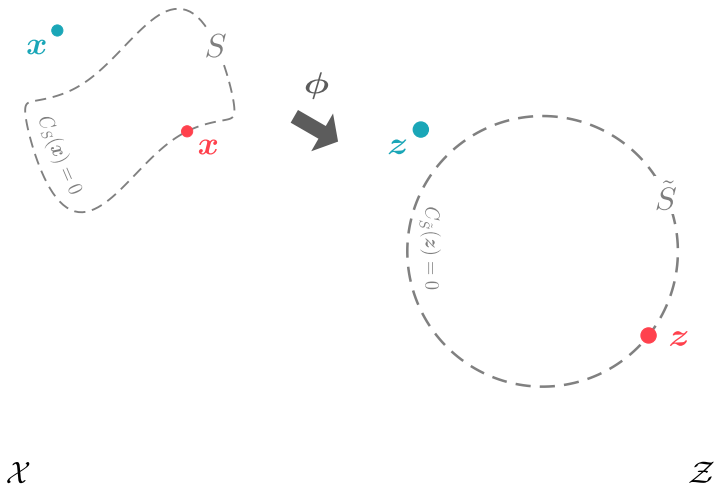
- Proposed dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\phi}^{-1}\left(\tilde{\boldsymbol{f}}(\boldsymbol{\phi}(\boldsymbol{x}))\right)$

Proposed method

- Problem: Learn dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ that has a stable invariant set $S \subset \mathcal{X}$
  - ➢ We need to learn S at the same time

- Proposed dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\phi}^{-1}\left(\tilde{\boldsymbol{f}}\left(\boldsymbol{\phi}(\boldsymbol{x})\right)\right)$

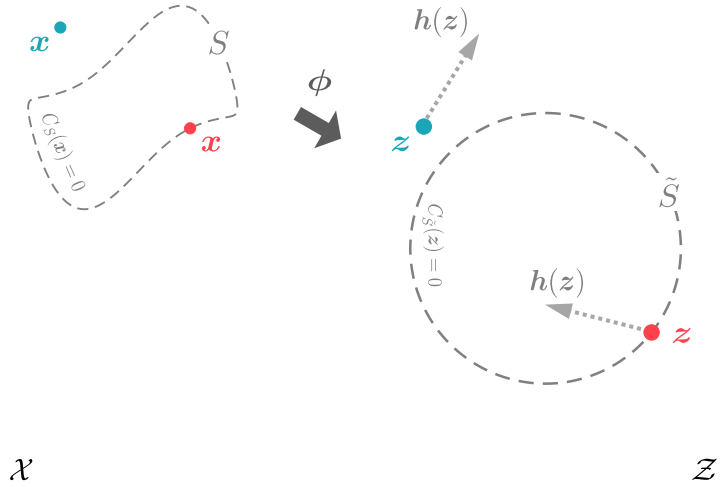
➡ 1. Transform $\boldsymbol{x}$ to $\boldsymbol{z} \in \mathcal{Z}$ using invertible map $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{Z}$



Proposed method

- Problem: Learn dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ that has a stable invariant set $S \subset \mathcal{X}$
  - ➢ We need to learn S at the same time

- Proposed dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\phi}^{-1}\left(\tilde{\boldsymbol{f}}(\boldsymbol{\phi}(\boldsymbol{x}))\right)$
  1. Transform $\boldsymbol{x}$ to $\boldsymbol{z} \in \mathcal{Z}$ using invertible map $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{Z}$
  2. Base dynamics model $\dot{\boldsymbol{z}} = h(z)$



Proposed method

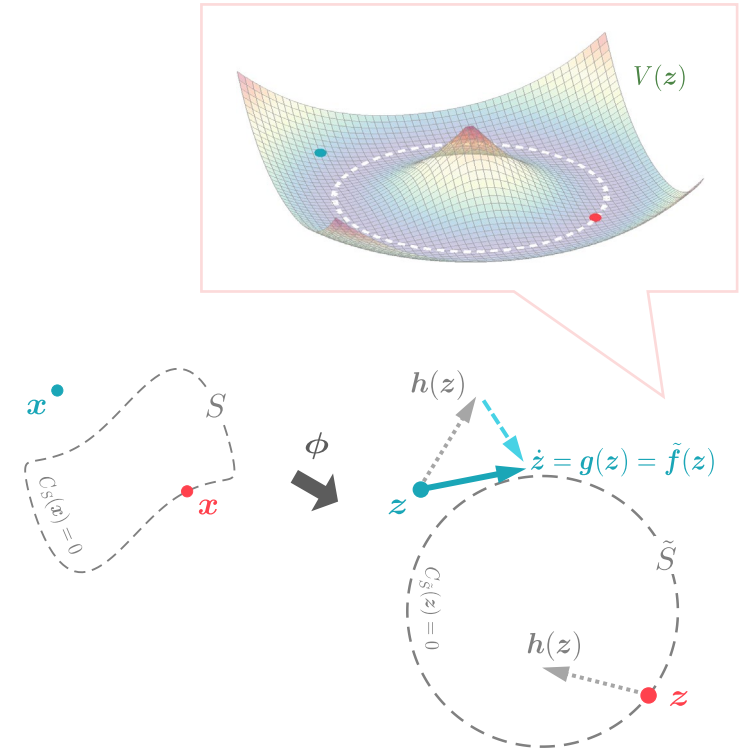
- Problem: Learn dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ that has a stable invariant set $S \subset \mathcal{X}$
  - ➢ We need to learn S at the same time

- Proposed dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\phi}^{-1}\left(\tilde{\boldsymbol{f}}(\boldsymbol{\phi}(\boldsymbol{x}))\right)$
  1. Transform $\boldsymbol{x}$ to $\boldsymbol{z} \in \mathcal{Z}$ using invertible map $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{Z}$
  2. Base dynamics model $\dot{\boldsymbol{z}} = h(\boldsymbol{z})$
  3. ➡ Modify $h(\boldsymbol{z})$ as $g(\boldsymbol{z})$ for ensuring stability of $\tilde{S}$

$$\boldsymbol{g}(\boldsymbol{z}) = \begin{cases} \boldsymbol{h}(\boldsymbol{z}), & \boldsymbol{z} \in \tilde{S} \\ \boldsymbol{h}(\boldsymbol{z}) - \text{step}\left(\nabla V(\boldsymbol{z})^{\mathsf{T}}\boldsymbol{h}(\boldsymbol{z}) + \alpha V(\boldsymbol{z})\right)\frac{\nabla V(\boldsymbol{z})^{\mathsf{T}}\boldsymbol{h}(\boldsymbol{z}) + \alpha V(\boldsymbol{z}) + \eta(\boldsymbol{z})}{\|\nabla V(\boldsymbol{z})\|_2^2}\nabla V(\boldsymbol{z}), & \boldsymbol{z} \notin \tilde{S} \end{cases}$$

where $V(\boldsymbol{z}) = \sigma\left(q(\boldsymbol{z}) - q\left(\mathbb{P}_{\tilde{S}}\boldsymbol{z}\right)\right) + \varepsilon\|\boldsymbol{z} - \mathbb{P}_{\tilde{S}}\boldsymbol{z}\|_2^2$

projection of $\boldsymbol{z}$ to $\tilde{S}$



Proposed method

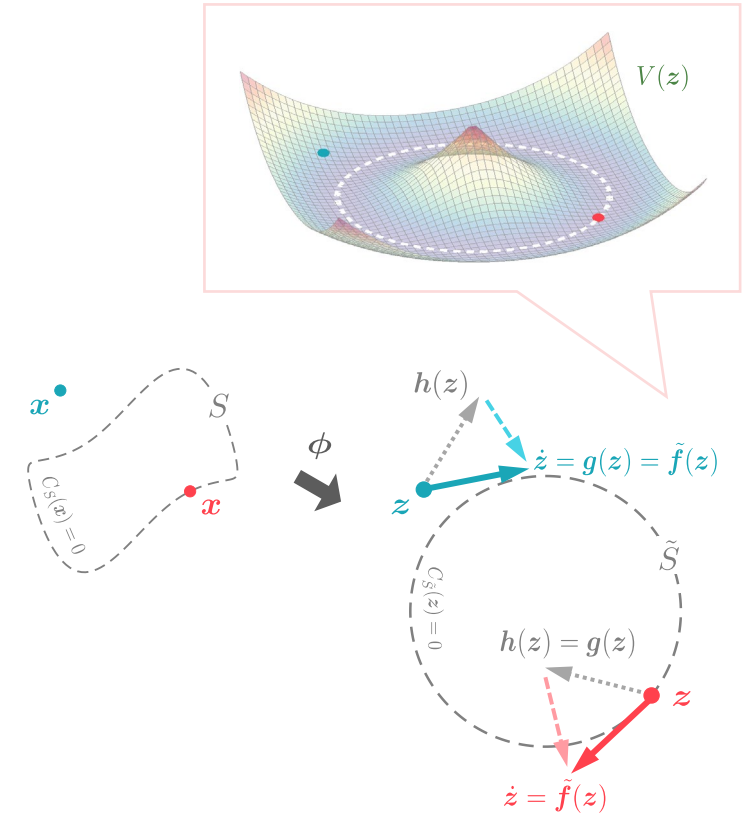
- Problem: Learn dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ that has a stable invariant set $S \subset \mathcal{X}$
  - ➢ We need to learn S at the same time

- Proposed dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\phi}^{-1}\left(\tilde{\boldsymbol{f}}(\boldsymbol{\phi}(\boldsymbol{x}))\right)$
  1. Transform $\boldsymbol{x}$ to $\boldsymbol{z} \in \mathcal{Z}$ using invertible map $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{Z}$
  2. Base dynamics model $\dot{\boldsymbol{z}} = h(\boldsymbol{z})$
  3. Modify $h(\boldsymbol{z})$ as $g(\boldsymbol{z})$ for ensuring stability of $\tilde{S}$

$$\boldsymbol{g}(\boldsymbol{z}) = \begin{cases} \boldsymbol{h}(\boldsymbol{z}), & \boldsymbol{z} \in \tilde{S} \\ \boldsymbol{h}(\boldsymbol{z}) - \text{step}\left(\nabla V(\boldsymbol{z})^{\mathsf{T}}\boldsymbol{h}(\boldsymbol{z}) + \alpha V(\boldsymbol{z})\right)\frac{\nabla V(\boldsymbol{z})^{\mathsf{T}}\boldsymbol{h}(\boldsymbol{z}) + \alpha V(\boldsymbol{z}) + \eta(\boldsymbol{z})}{\|\nabla V(\boldsymbol{z})\|_2^2}\nabla V(\boldsymbol{z}), & \boldsymbol{z} \notin \tilde{S} \end{cases}$$

➡ 4. Modify $g(\boldsymbol{z})$ as $\tilde{f}(\boldsymbol{z})$ for ensuring invariance of $\tilde{S}$

$$\tilde{\boldsymbol{f}}(\boldsymbol{z}) = \begin{cases} \boldsymbol{g}(\boldsymbol{z}), & C_{\tilde{S}}(\boldsymbol{z}) \neq 0 \\ \boldsymbol{g}(\boldsymbol{z}) - \frac{\nabla C_{\tilde{S}}(\boldsymbol{z})^{\mathsf{T}}\boldsymbol{g}(\boldsymbol{z}) - \xi(\boldsymbol{z})}{\|C_{\tilde{S}}(\boldsymbol{z})\|_2^2}\nabla C_{\tilde{S}}(\boldsymbol{z}), & C_{\tilde{S}}(\boldsymbol{z}) = 0 \end{cases}$$



Proposed method

- Problem: Learn dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ that has a stable invariant set $S \subset \mathcal{X}$
  - ➢ We need to learn S at the same time

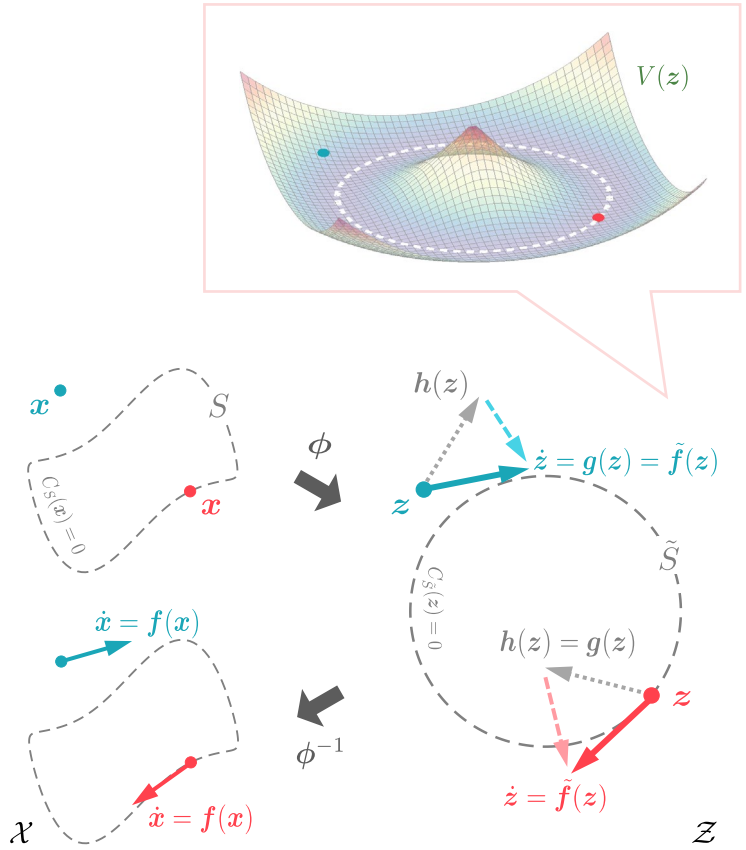
- Proposed dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\phi}^{-1}\left(\tilde{\boldsymbol{f}}(\boldsymbol{\phi}(\boldsymbol{x}))\right)$
  1. Transform $\boldsymbol{x}$ to $\boldsymbol{z} \in \mathcal{Z}$ using invertible map $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{Z}$
  2. Base dynamics model $\dot{\boldsymbol{z}} = h(\boldsymbol{z})$
  3. Modify $h(\boldsymbol{z})$ as $g(\boldsymbol{z})$ for ensuring stability of $\tilde{S}$

$$\boldsymbol{g}(\boldsymbol{z}) = \begin{cases} \boldsymbol{h}(\boldsymbol{z}), & \boldsymbol{z} \in \tilde{S} \\ \boldsymbol{h}(\boldsymbol{z}) - \text{step}\left(\nabla V(\boldsymbol{z})^{\mathsf{T}}\boldsymbol{h}(\boldsymbol{z}) + \alpha V(\boldsymbol{z})\right)\frac{\nabla V(\boldsymbol{z})^{\mathsf{T}}\boldsymbol{h}(\boldsymbol{z}) + \alpha V(\boldsymbol{z}) + \eta(\boldsymbol{z})}{\|\nabla V(\boldsymbol{z})\|_2^2}\nabla V(\boldsymbol{z}), & \boldsymbol{z} \notin \tilde{S} \end{cases}$$

  4. Modify $g(\boldsymbol{z})$ as $\tilde{f}(\boldsymbol{z})$ for ensuring invariance of $\tilde{S}$

$$\tilde{\boldsymbol{f}}(\boldsymbol{z}) = \begin{cases} \boldsymbol{g}(\boldsymbol{z}), & C_{\tilde{S}}(\boldsymbol{z}) \neq 0 \\ \boldsymbol{g}(\boldsymbol{z}) - \frac{\nabla C_{\tilde{S}}(\boldsymbol{z})^{\mathsf{T}}\boldsymbol{g}(\boldsymbol{z}) - \xi(\boldsymbol{z})}{\|C_{\tilde{S}}(\boldsymbol{z})\|_2^2}\nabla C_{\tilde{S}}(\boldsymbol{z}), & C_{\tilde{S}}(\boldsymbol{z}) = 0 \end{cases}$$

  ➡ 5. Project back from $\mathcal{Z}$ to $\mathcal{X}$ using $\phi^{-1} : \mathcal{Z} \to \mathcal{X}$

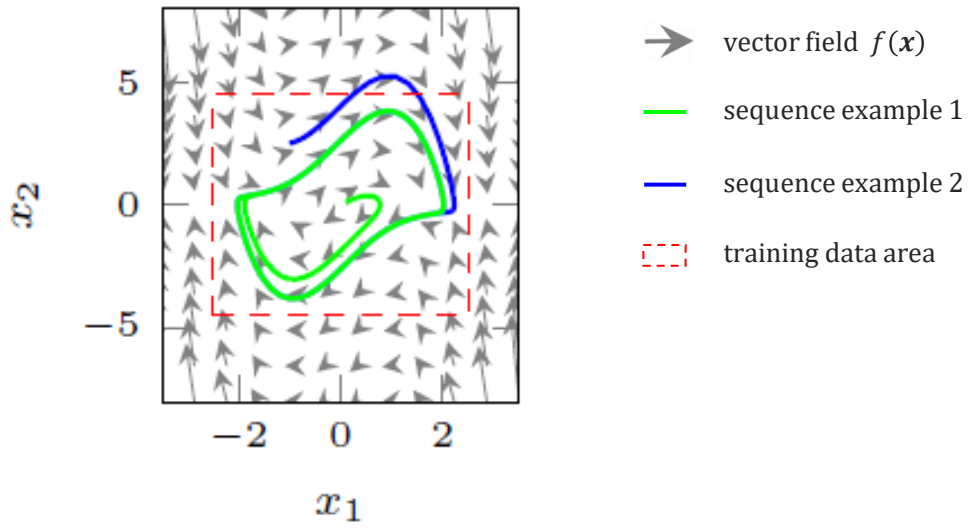


Experiment 1

Data

Van der Pol oscillator

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \mu(1 - x_1^2)x_2 - x_1 \end{bmatrix}$$

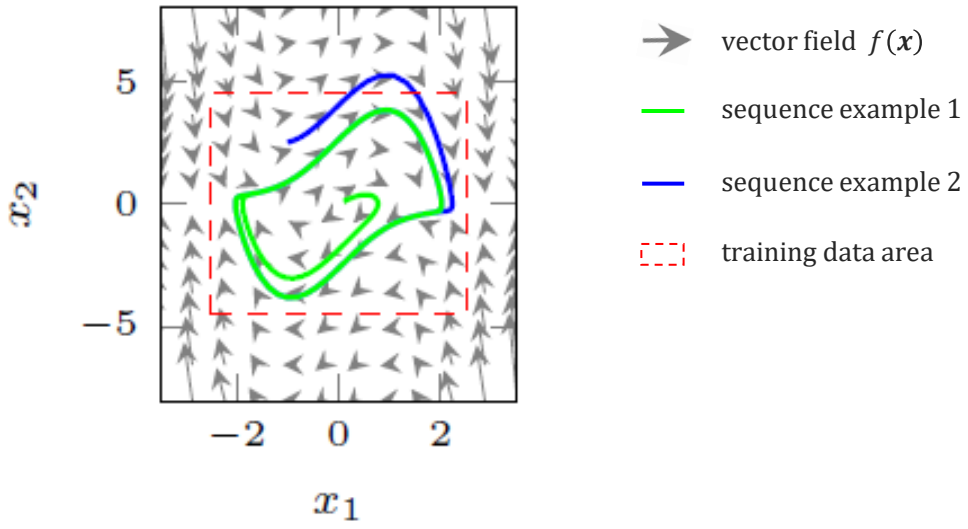


Experiment 1

Data

Van der Pol oscillator

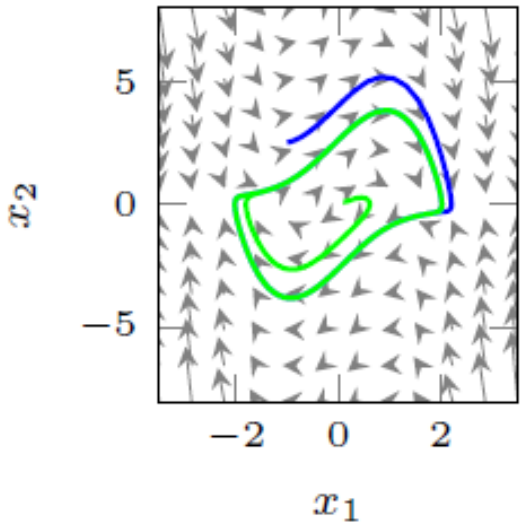
$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \mu(1 - x_1^2)x_2 - x_1 \end{bmatrix}$$



Result (1/3)

Output from learned proposed model with:

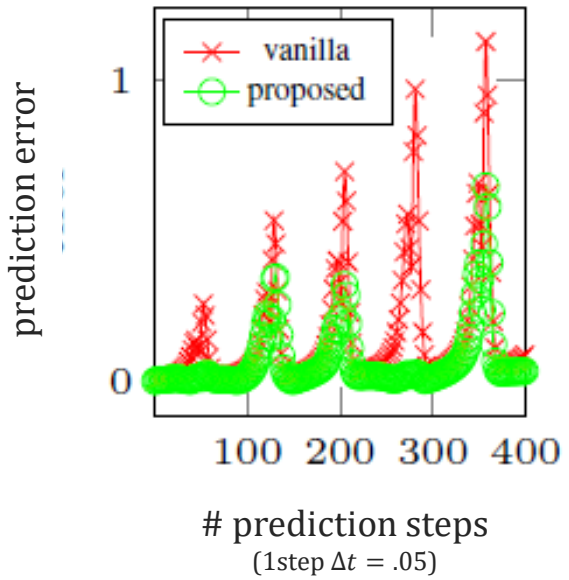
- unit circle as $\tilde{S}$
- fully-connected NNs as $\boldsymbol{h}$ and V



Experiment 1

Result (2/3)

Long-term prediction

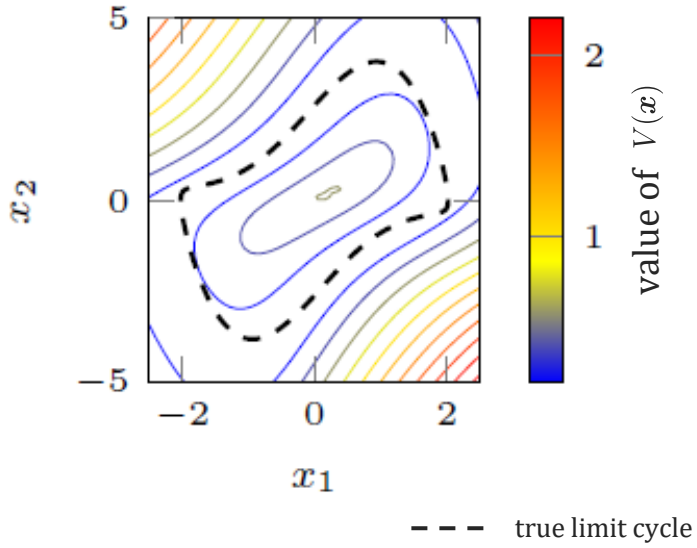


neural ODE
(without stability guarantee)

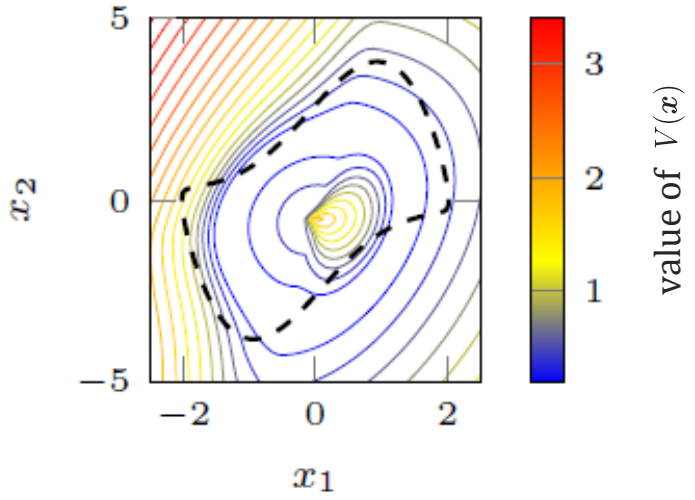
proposed model
(with stability guarantee)

Experiment 1

Result (3/3)



learned $V(x)$
in proposed method



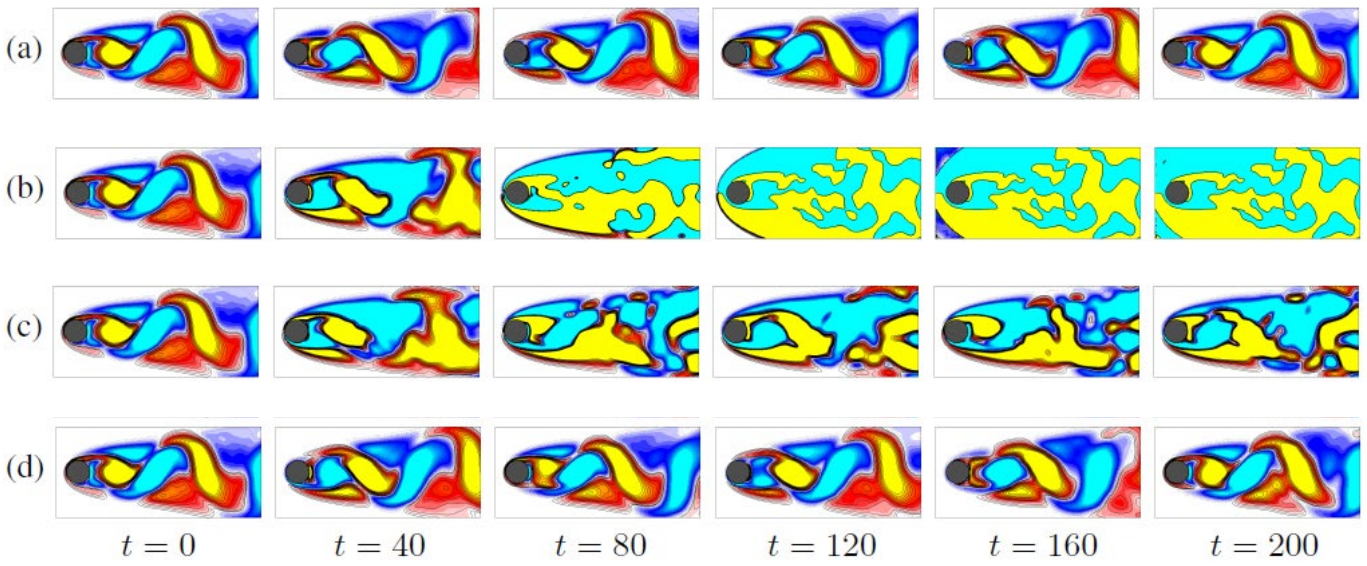
learned $V(x)$
without ϕ

Experiment 2

Data 2-D fluid flow around cylinder-like object

- reaches limit cycle known as Kármán's vortex
- training data taken *before* the limit cycle; test data taken *after* the limit cycle

Result Long-term prediction



Ground truth

Without any stability guarantee

Stable equilibrium [Manek&Kolter 19]

Proposed method
(with stable limit cycle)

Summary

- Learning dynamics model with general stable invariant set
- Realized by transforming state vector to latent state by invertible neural net
  - ➢ and ensuring stability and invariance in the latent space
- Useful when prior knowledge of target dynamics is available