Lecture 3

Learning Control Laws with Dynamical Systems

Outline

- Why learned dynamics with standard machine learning algorithm will be unstable
- Stable Estimator of Dynamical Systems (SEDS)
- Linear Parameter Varying Dynamical Systems

Straight out of the box machine learning tools will often not lead to a stable dynamical system

MODEL

$$\dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x}) \rightarrow \left\{ \lim_{t\rightarrow\infty} \|\boldsymbol{x} - \boldsymbol{x}^*\| = 0 \right.$$

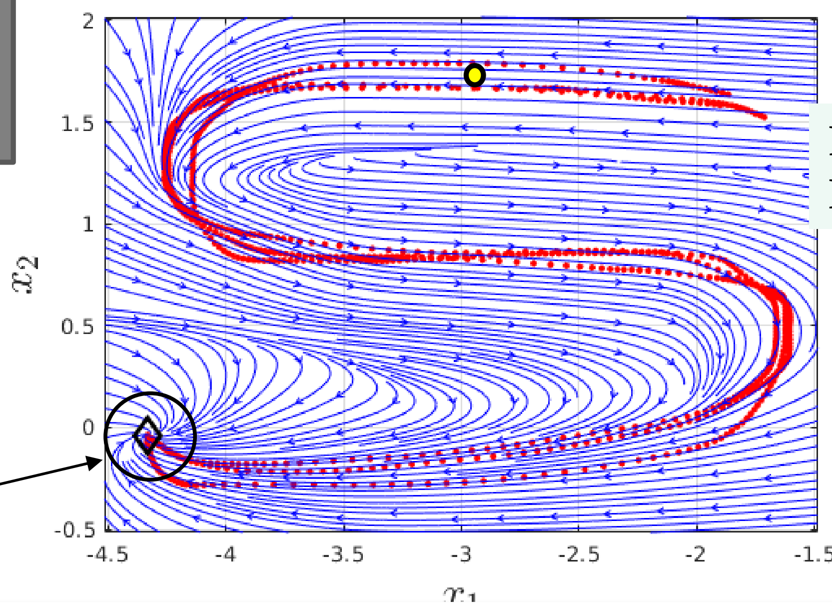
Velocity — Position — Target

$\boldsymbol{x}, \dot{\boldsymbol{x}} \in \mathbb{R}^N$
$\mathbf{f}(\boldsymbol{x}) : \mathbb{R}^N \rightarrow \mathbb{R}^N$

Demonstrations

Vector field $\mathbf{f}(\boldsymbol{x})$

Target



From a Machine Learning perspective: Learning $\mathbf{f}(\boldsymbol{x})$ is a regression problem!

Can we ensure convergence to the target with any algorithm?

We model the robot as a **point mass** moving according to a *time-invariant autonomous dynamical system* (DS)

Training data

DATA: set of M reference trajectories

$$\{X, \dot{X}\} = \left\{ \left\{ x_t^i, \dot{x}_t^i \right\}_{t=1}^{T_m} \right\}_{m=1}^{M}$$

T_m : Length of each trajectory

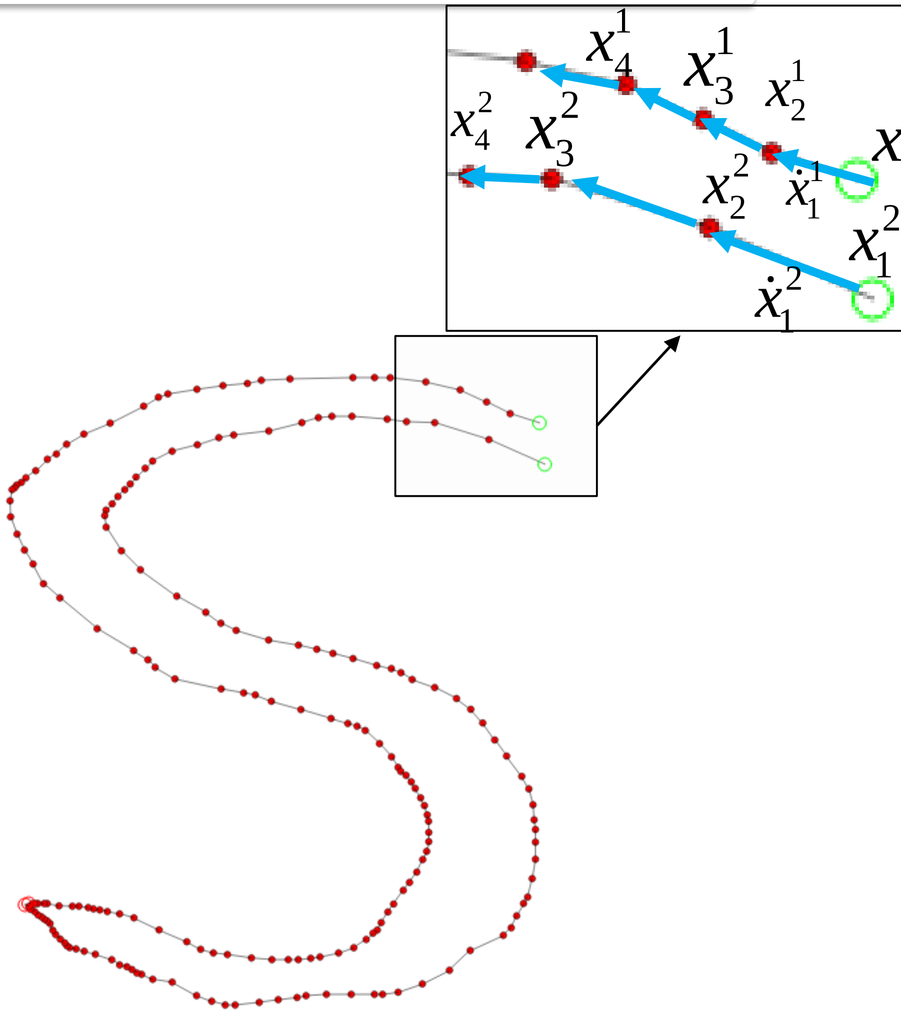
Choose a function f:

$$\dot{x} = f(x; \Theta)$$

Θ : Model's parameters

Search the parameters Θ that fit the data at best according to a loss: $L(X, f, \Theta)$

Any regression technique could be used.



Using support vector regression

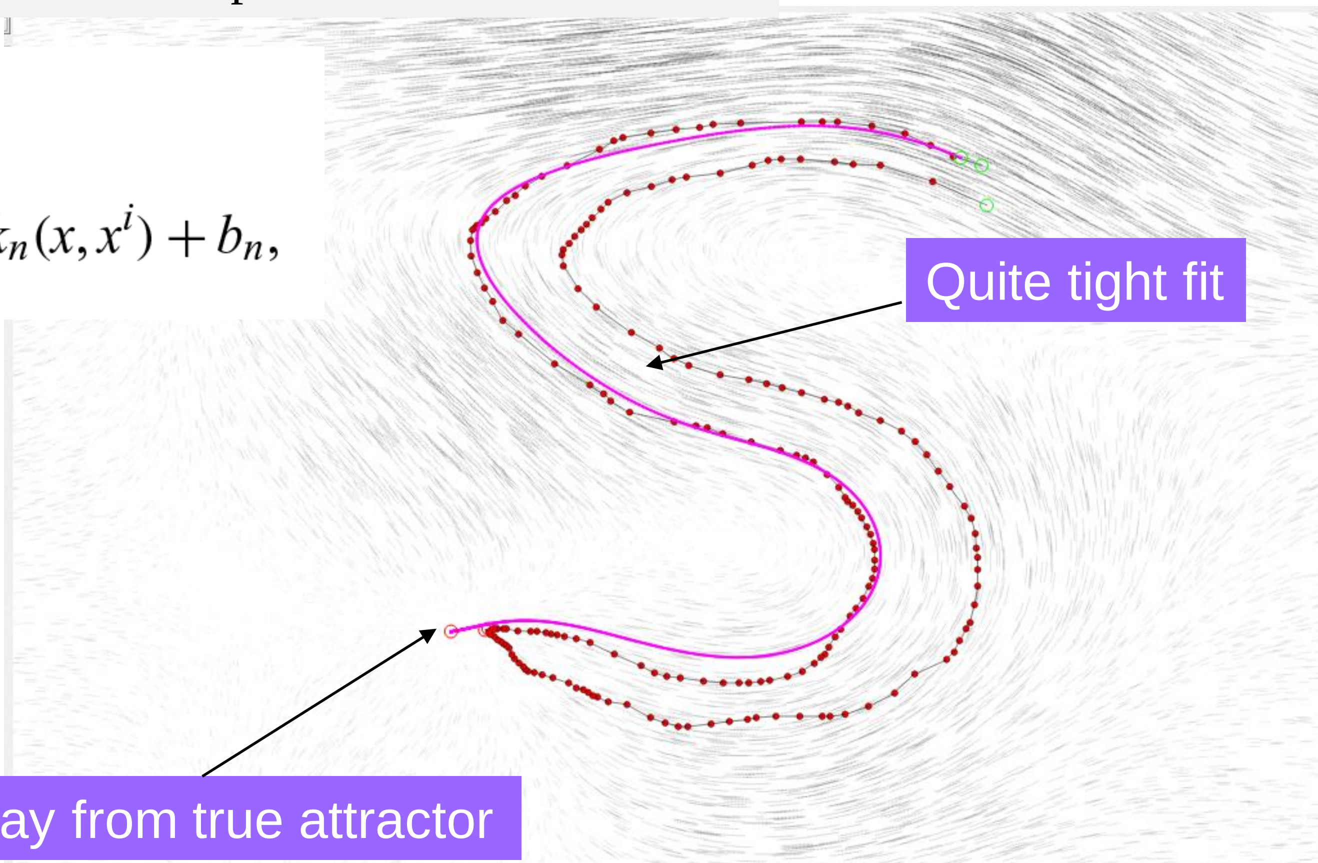
If you use Support Vector Regression,

you must fit each of the velocity dimension independently.

If $\dot{x} \in \mathbb{R}^N$, we need $n = 1 \ldots N$ independent SVR fits.

$$\dot{x}_n = f_n(x;\ \theta^n_{\mathrm{SVR}})$$

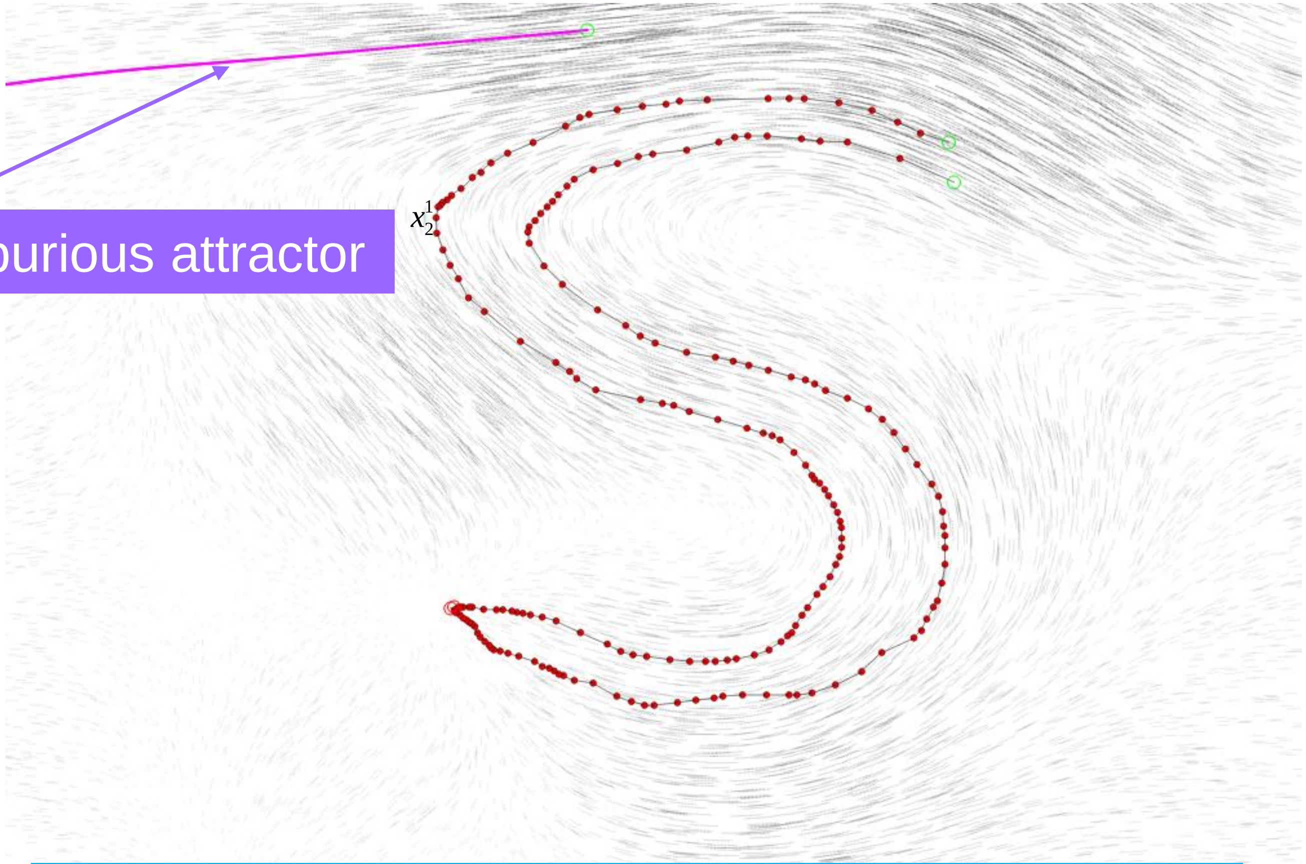
$$= \sum_{i=1}^{L} (\alpha_i - \alpha_i^*)_n k_n(x, x^i) + b_n,$$



Fit using RBF kernel with tiny ε loss and small kernel width.

Using support vector regression

Drifts to some other spurious attractor



Fit using RBF kernel with tiny ε loss and small kernel width.

Using Gaussian Mixture Regression

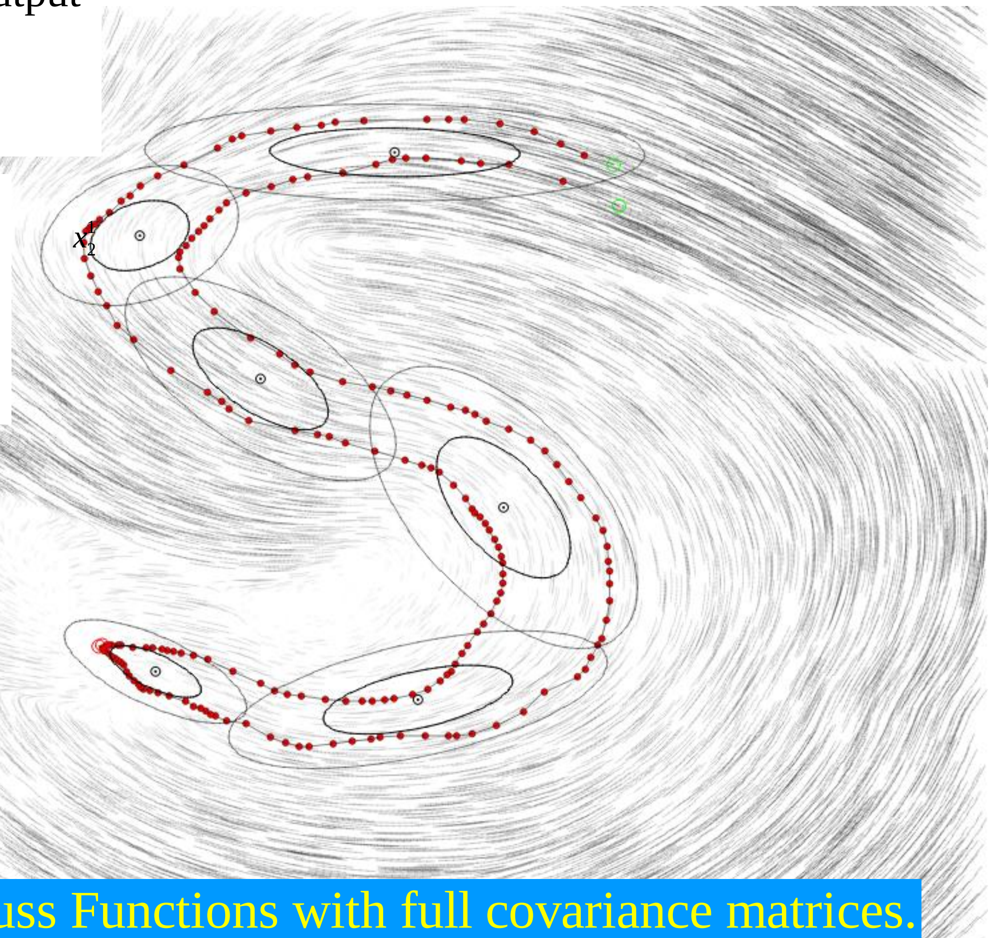
If you use Gaussian Mixture Regression,
you can fit all velocity dimensions at once,
as you learn a model of the joint distribution $p(x, \dot{x})$.

You query for the velocity by conditioning the output for each velocity.

This can be written as a compact form:

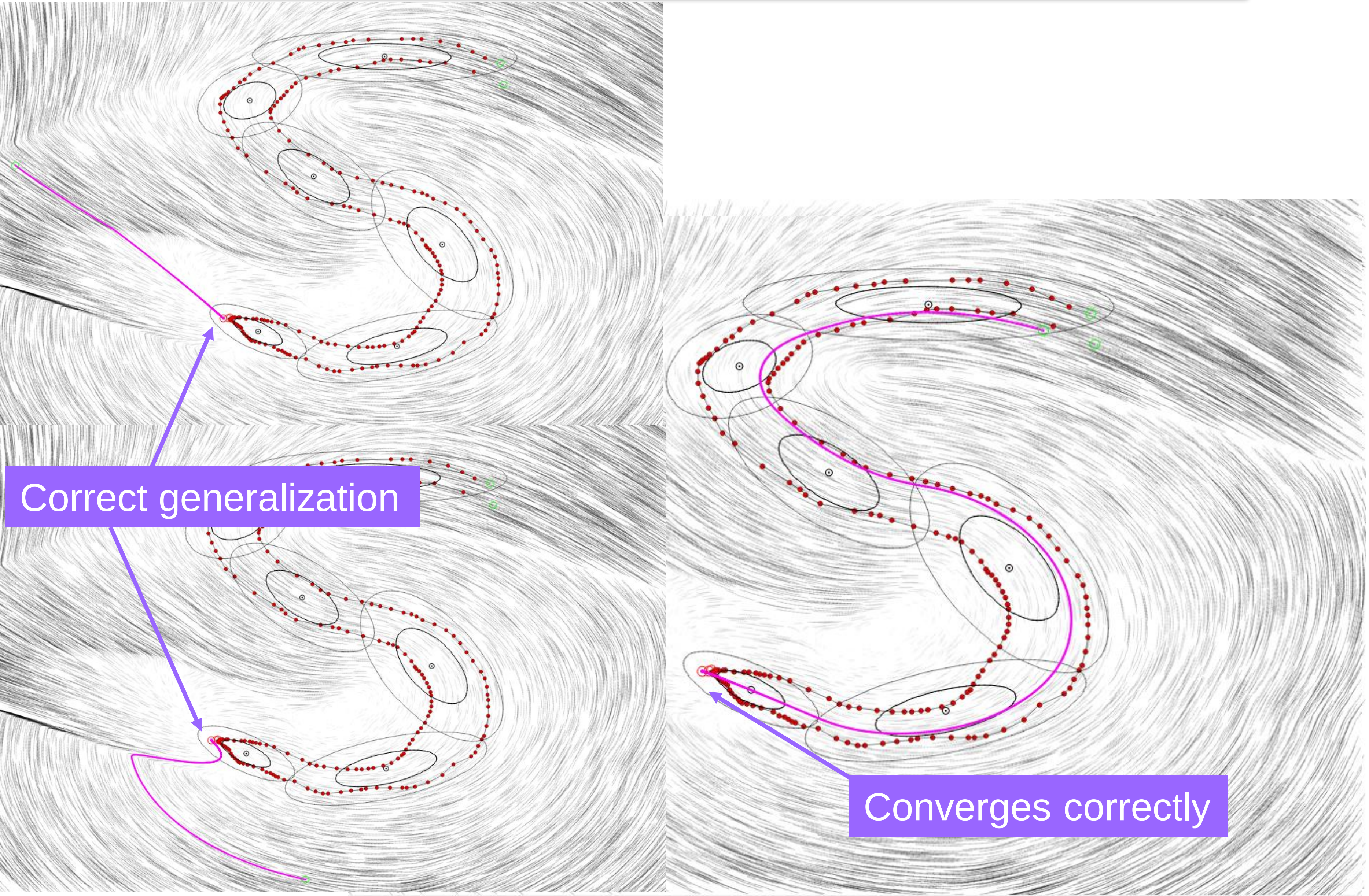
$$\dot{x} = f(x;\ \Theta_{\text{GMR}})$$

$$= \mathbb{E}\{p(\dot{x}|x)\} = \sum^{K} \gamma_k(x) \tilde{\mu}^k(x),$$



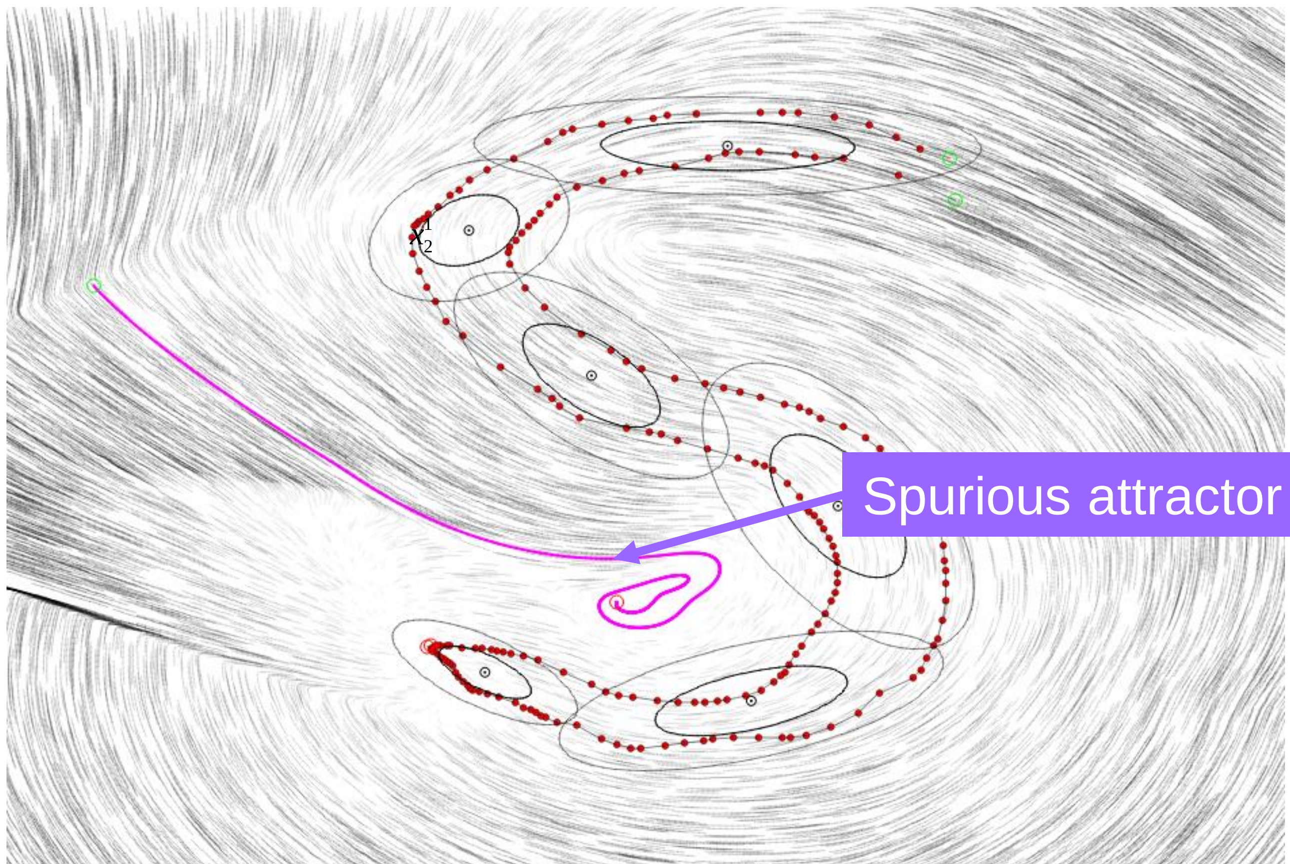
Fit using 6 Gauss Functions with full covariance matrices.

Using Gaussian Mixture Regression



Using Gaussian Mixture Regression

Spurious attractor



Fit using 6 Gauss Functions with full covariance matrices.

Summary

- Learning a control law composed of a dynamical system can be formulated as a regression problem.
  - We regress on the velocity given the state.

- Using out of the box machine learning techniques for regression is insufficient:
  - It may lead to imprecise estimate of the attractor.
  - Many trajectories may drift away from the attractor.
  - Spurious fixed point attractors with spurious local dynamics may arise.