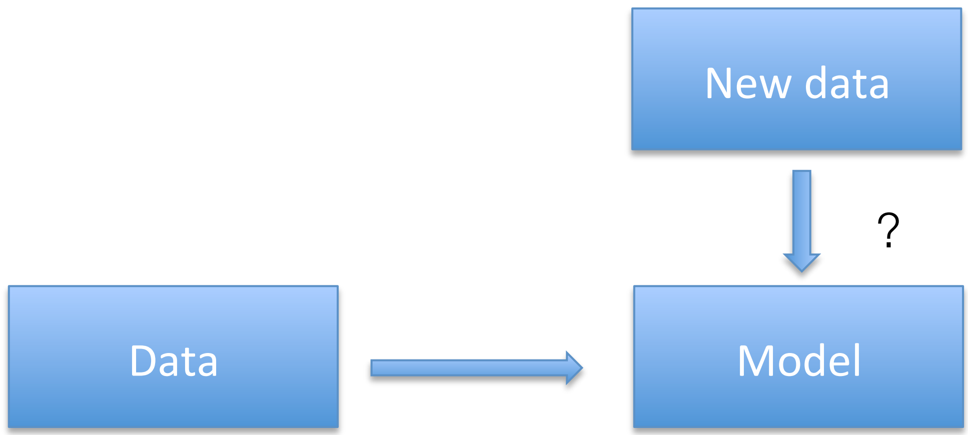
Tutorial on Learning from Demonstration

Part 3: Refinement in Dynamical Systems

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Incremental learning



Batch learning

Loss of data

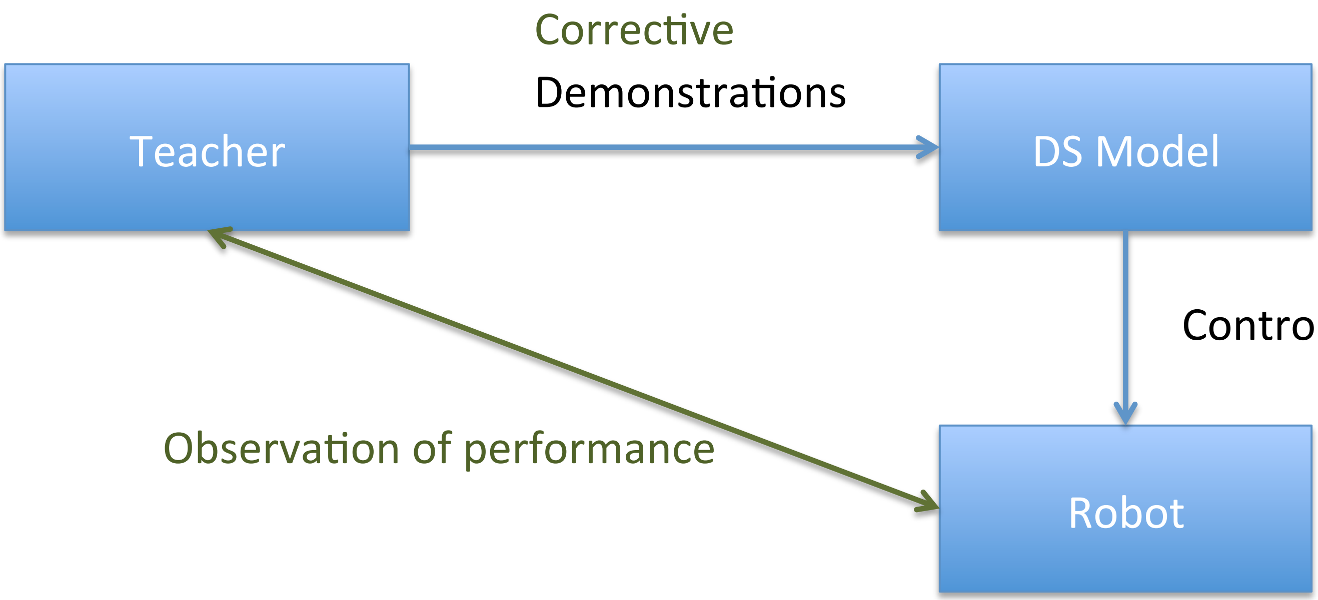
How can we consolidate new data with previous data?

Incremental learning

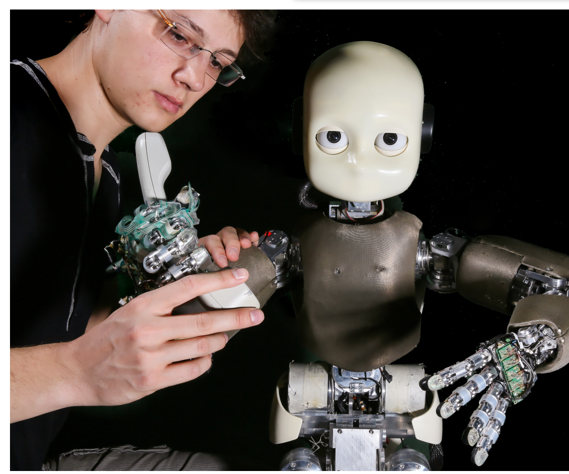
Gaussian Process Regression (GPR)

Locally Weighted Projection Regression (LWPR)

Incremental Learning is important for effective LfD



DS



bility properties!

Loss of stability properties, example

- Let's say we start with some dynamical system with a single equilibrium point at the origin.

$$\dot{x} = f(x) \qquad f(0) = 0 \qquad f(x) \neq 0, \quad \forall x \neq 0$$

- Now we add a term representing the contribution from incremental learning.

$$\dot{x} = f(x) + g(x)$$

- If at any point, $g(x) = -f(x)$ the system has introduced a *spurious attractor.*
- Can be much worse!

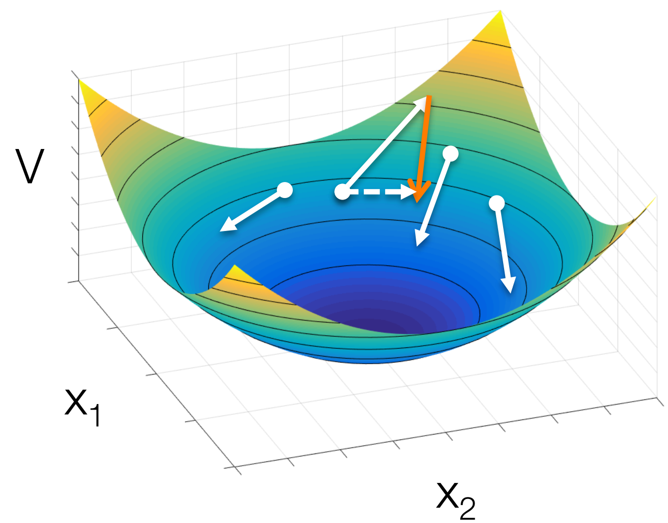
Stabilized Incremental Learning

- A family of algorithms that are based on “repairing” a destabilized system.

$$\dot{x} = \boxed{f(x) + g(x)}$$

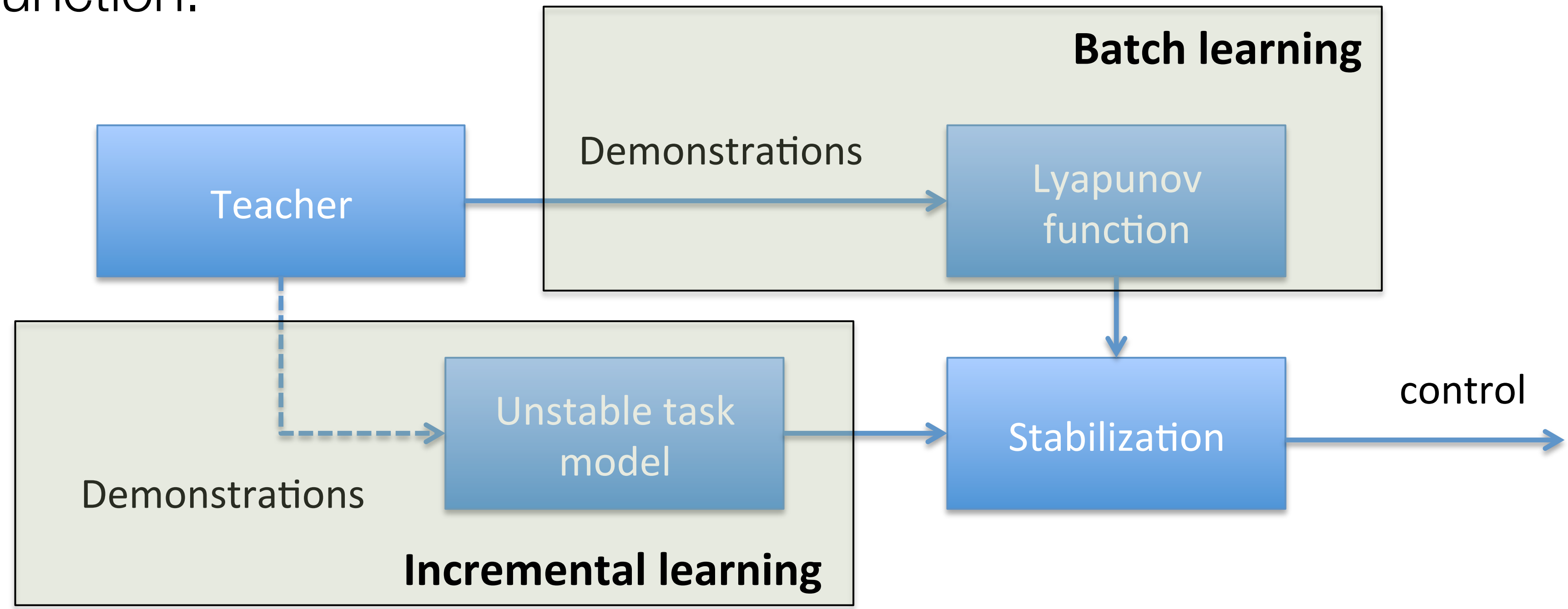
- Force system to descend into valley of Lyapunov function. Choose u to satisfy:

$$(f(x) + g(x) + u(x))^T \nabla V < 0$$



Learning the Energy Function

Stabilizing requires knowledge of a parameterized Lyapunov function.



Khansari-Zadeh and Billard, **Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions**, Robotics and Autonomous Systems 2013

Lemme et al., **Neural learning of vector fields for encoding stable dynamical systems**, Neurocomputing 2014

Modulating a Dynamical System

- So far we considered modifying a DS by addition:

$$\dot{x} = f(x) + g(x)$$

- But there are other ways to change the system:

$$\dot{x} = M(x)f(x)$$

$M(x)$ full rank => no introduction of spurious attractors

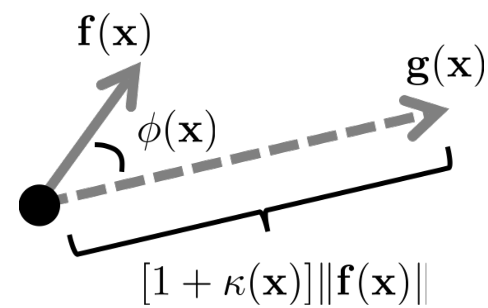
Rotation and scaling

- Can we modulate to make the DS align with incoming data?

$$\dot{x} = M(x) f(x)$$

$$M(x) = [1 + \kappa(x)] R(x)$$

$$\mathbf{R}(\mathbf{x}) = \begin{bmatrix} \cos(\phi(\mathbf{x})) & -\sin(\phi(\mathbf{x})) \\ \sin(\phi(\mathbf{x})) & \cos(\phi(\mathbf{x})) \end{bmatrix}$$

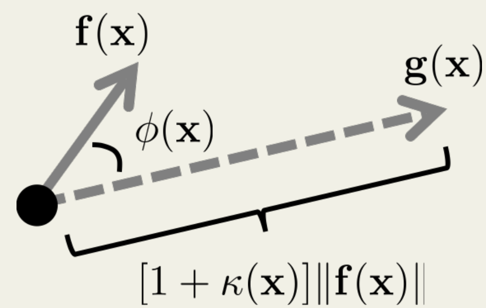


- Modulation parameterized by scaling and rotation angle:

$$\boldsymbol{\theta}(\mathbf{x}) = \begin{bmatrix} \kappa(\mathbf{x}) \\ \phi(\mathbf{x}) \end{bmatrix}$$

Learn from demonstrations.

From Trajectory Data to Reshaping Parameters



$$\boldsymbol{\theta}(\mathbf{x}) = \begin{bmatrix} \kappa(\mathbf{x}) \\ \phi(\mathbf{x}) \end{bmatrix}$$

Trajectory data

$$\{x_m, \dot{x}_m\}_{m=1}^{M}$$

$$x_m^o = f(x_m)$$

$$\{\dot{x}_m^o\}_{m=1}^{M}$$

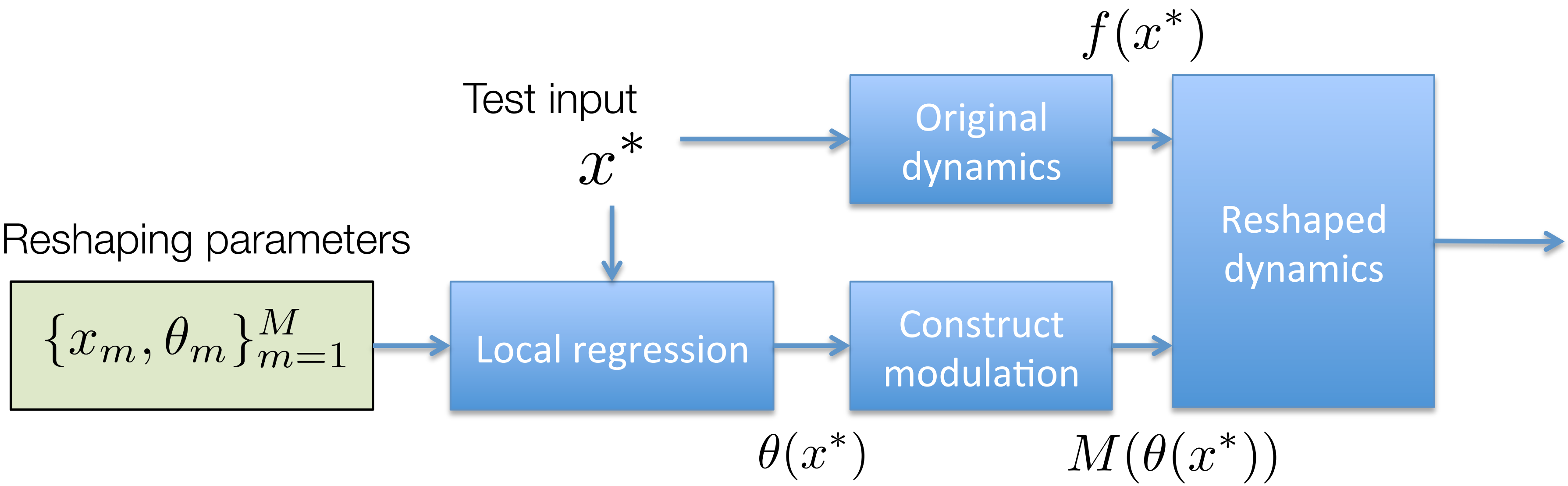
$$\kappa_m = \frac{\|\dot{x}_m\|}{\|\dot{x}_m^o\|} - 1$$

$$\phi_m = \frac{acos(\langle \dot{x}_m, \dot{x}_m^o \rangle)}{\|\dot{x}_m\|\|\dot{x}_m^o\|}$$

Reshaping parameters

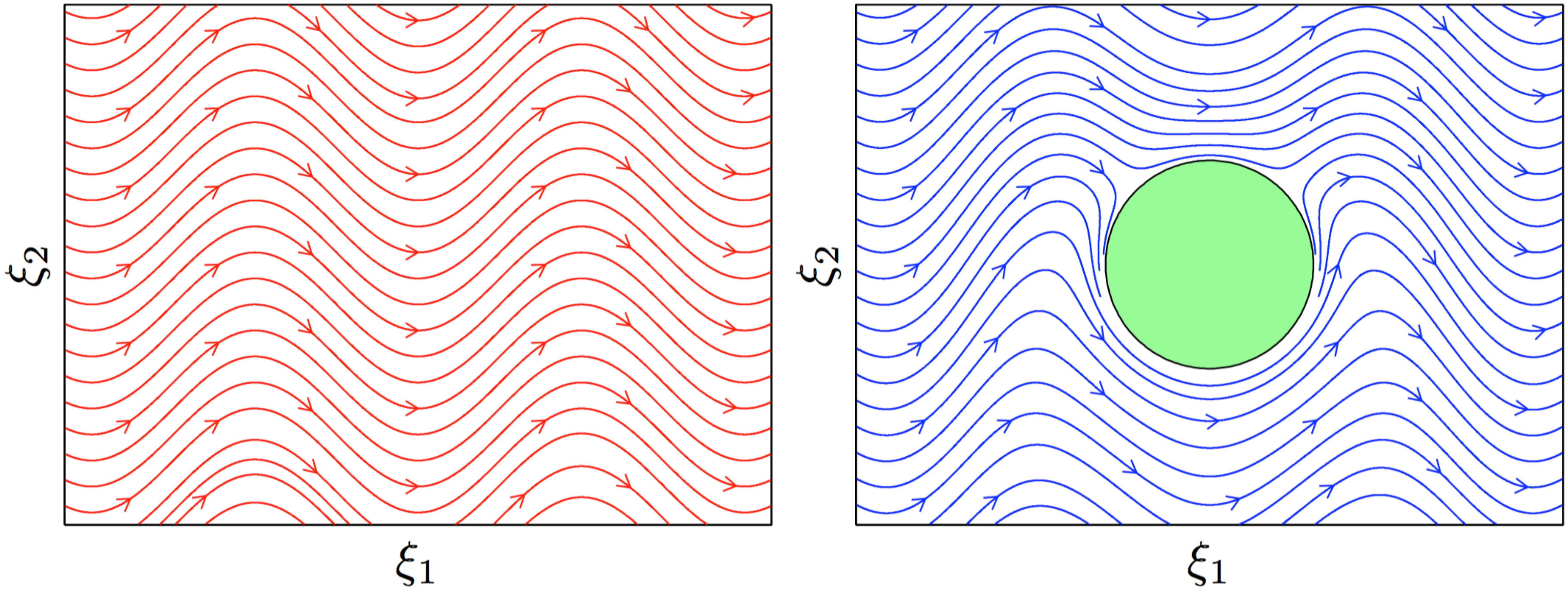
$$\{x_m, \theta_m\}_{m=1}^{M}$$

Learning and Using the Reshaped Dynamics



- Original dynamics must have the desired qualitative properties:
  - Point attractor
  - Limit cycle
  - Multiple attractors
- The detail comes from local modulations.

Modulating for Obstacle Avoidance



Khansari-Zadeh and Billard, **A Dynamical Systems Approach to Realtime Obstacle Avoidance**, Autonomous robots 2012