Almost Surely Stable Deep Dynamics

Nathan Lawrence, Philip Loewen, Michael Forbes, Johan Backstrom, Bhushan Gopaluni

UBC

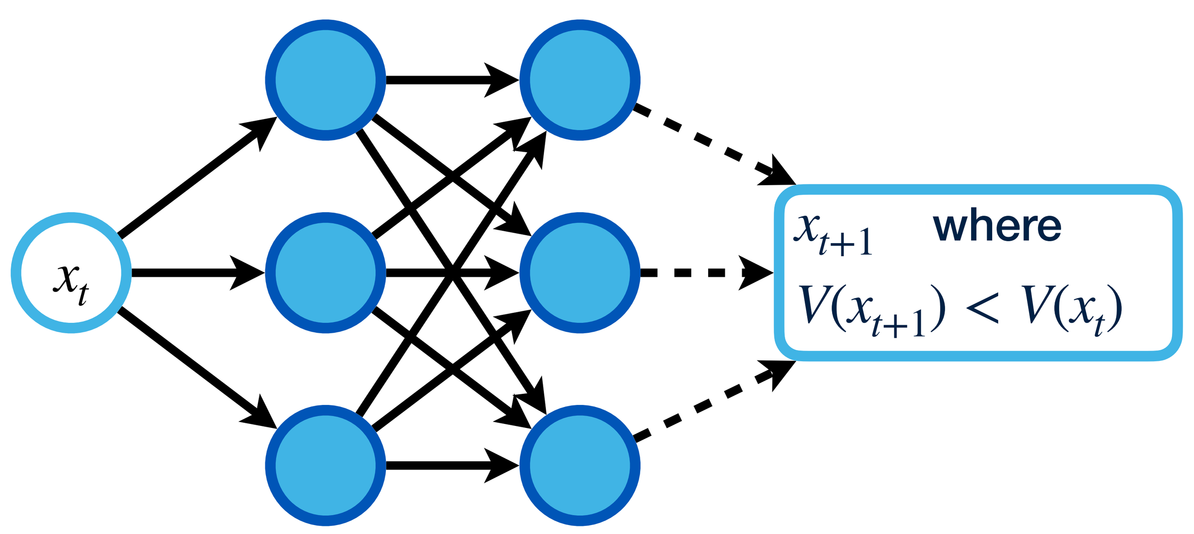
NEURAL INFORMATION PROCESSING SYSTEMS

We construct DNN-based dynamic models with *implicit* stability guarantees.

This paper is concerned with modeling *stable* discrete-time stochastic systems of the form:

$$x_{t+1} = f(x_t, \omega_{t+1})$$

We embed a Lyapunov neural network into a DNN-based dynamic model, making it inherently stable. [1] proposed this idea for continuous-time systems, however, we are concerned with training based on noisy, discrete state observations.

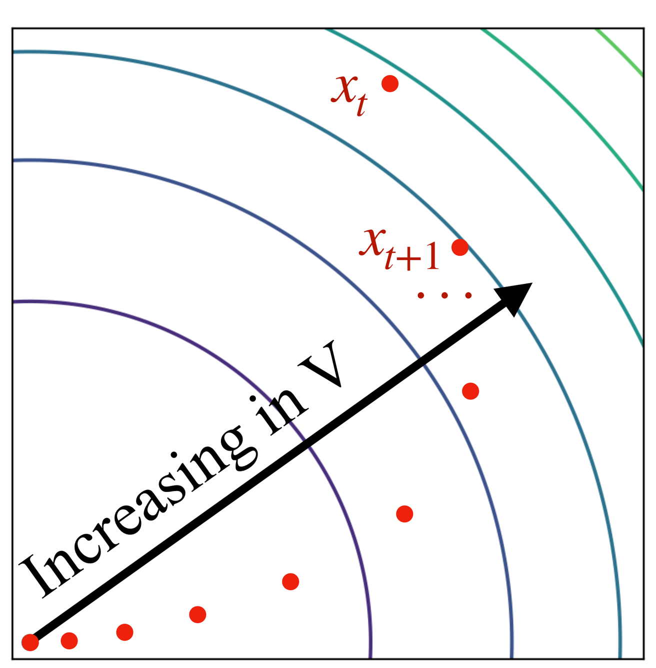


An implicit output layer enforces a stability condition on the dynamic model.

Lyapunov stability

Lyapunov theory is a powerful framework for proving stability. A (candidate) Lyapunov function is a 'bowl' shaped function that satisfies several conditions:

- $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is continuous with $V(0) = 0$
- $\varphi(\|x\|) \leq V(x) \quad \forall x \in \mathbb{R}^n$ where φ is a nonnegative, continuous, strictly increasing and unbounded function

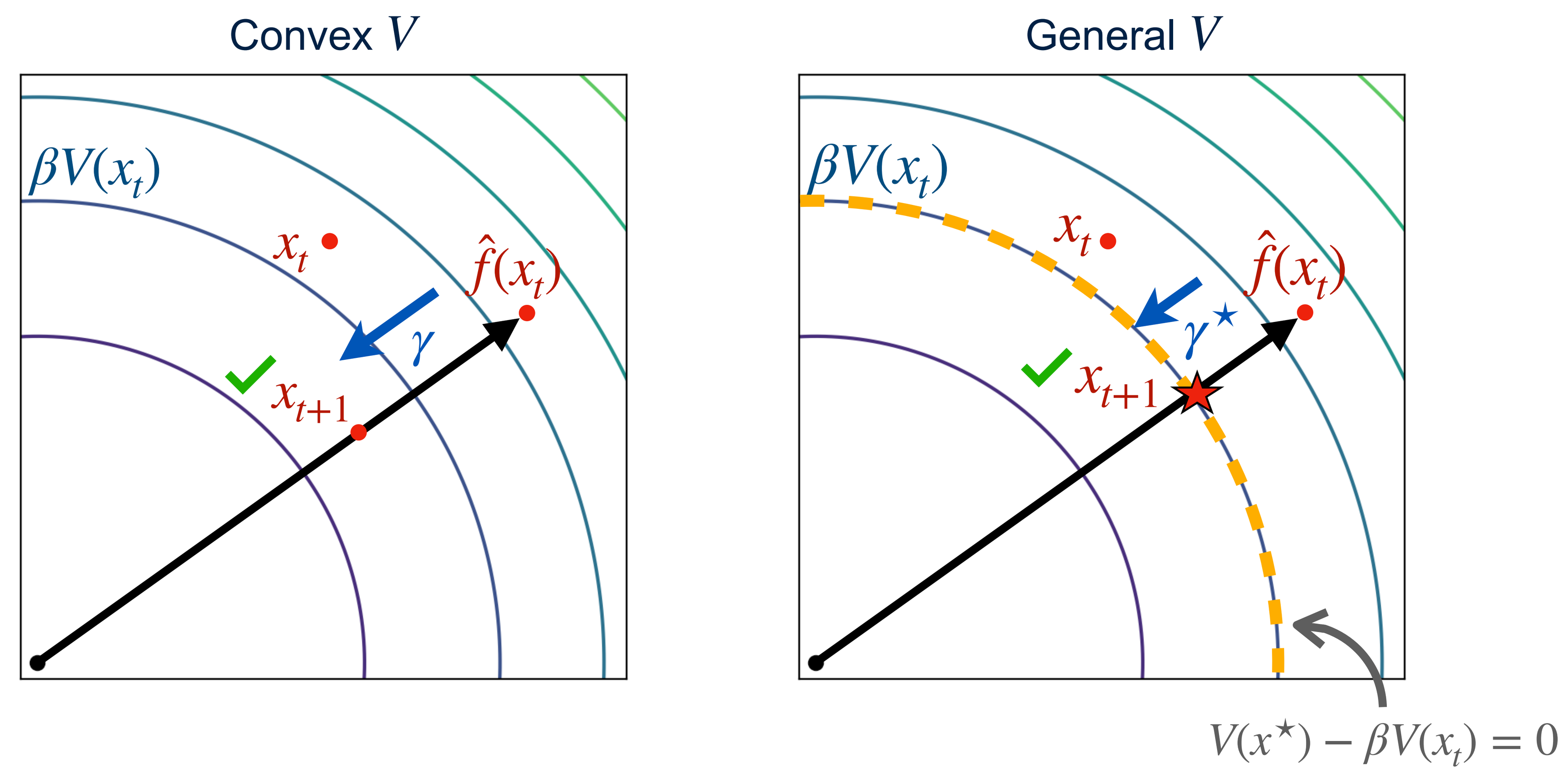


If there exists V and constant $0 < \beta \leq 1$ such that:

$$V(x_{t+1}) < \beta V(x_t) \quad \forall t$$

Then the system is asymptotically stable [2].

Key: A Lyapunov function is typically sought for particular dynamics, whereas our model has a Lyapunov function 'built-in' to the dynamics, inherently satisfying the stability condition.



The nominal dynamic model $\hat{f}$ is 'absorbed' back into a suitable level set of V.

Closed-form dynamics for convex Lyapunov NN

This column treats deterministic systems, then is extend to the stochastic setting. A Lyapunov neural network (NN) V 'guides' the nominal dynamic model $\hat{f}$ to the equilibrium point:

1. If $V(\hat{f}(x_t)) \leq \beta V(x_t)$, then no problem
2. If $V(\hat{f}(x_t)) > \beta V(x_t)$, then define:

$$x_{t+1} = \gamma(x_t)\hat{f}(x_t) = \frac{\beta V(x_t)}{V(\hat{f}(x_t))}\hat{f}(x_t)$$

➡ Easy to implement; $\hat{f}$ and V are trained simultaneously; However, the model may rely too heavily on convexity.

Implicitly stable dynamics for general Lyapunov NN

We generalize the above strategy through an implicit model:

1. If $V(\hat{f}(x_t)) \leq \beta V(x_t)$, then no problem
2. If $V(\hat{f}(x_t)) > \beta V(x_t)$, then solve the equation in $\gamma^\star$:

$$V(\gamma^\star \hat{f}(x_t)) - \beta V(x_t) = 0$$

➡ The implicit model is continuous in x; The one-dimensional root-finding problem can be evaluated via a Newton-bisection hybrid; However, training can be more difficult/slower.

Stable mixture density network parameters

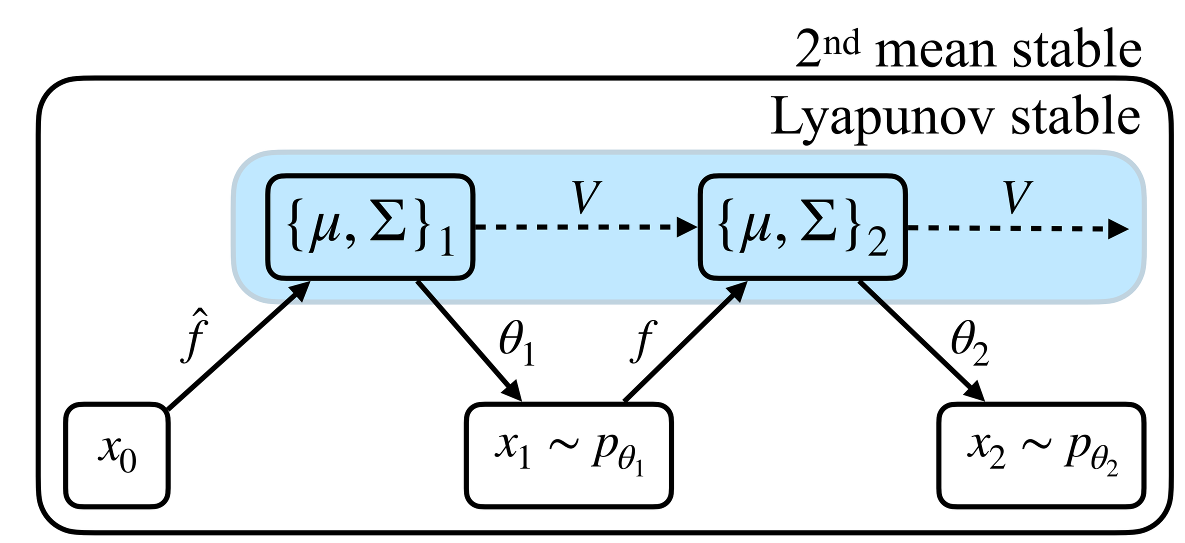
A mixture density network [3] is a convenient way of modeling conditional density functions:

$$p(x_{t+1} \mid x_t) = \sum_{i=1}^{k} \pi_i(x_t)\phi_i(x_{t+1} \mid x_t)$$

The means/covariances are given by the network, which we explicitly control in a stable fashion using the previous techniques:

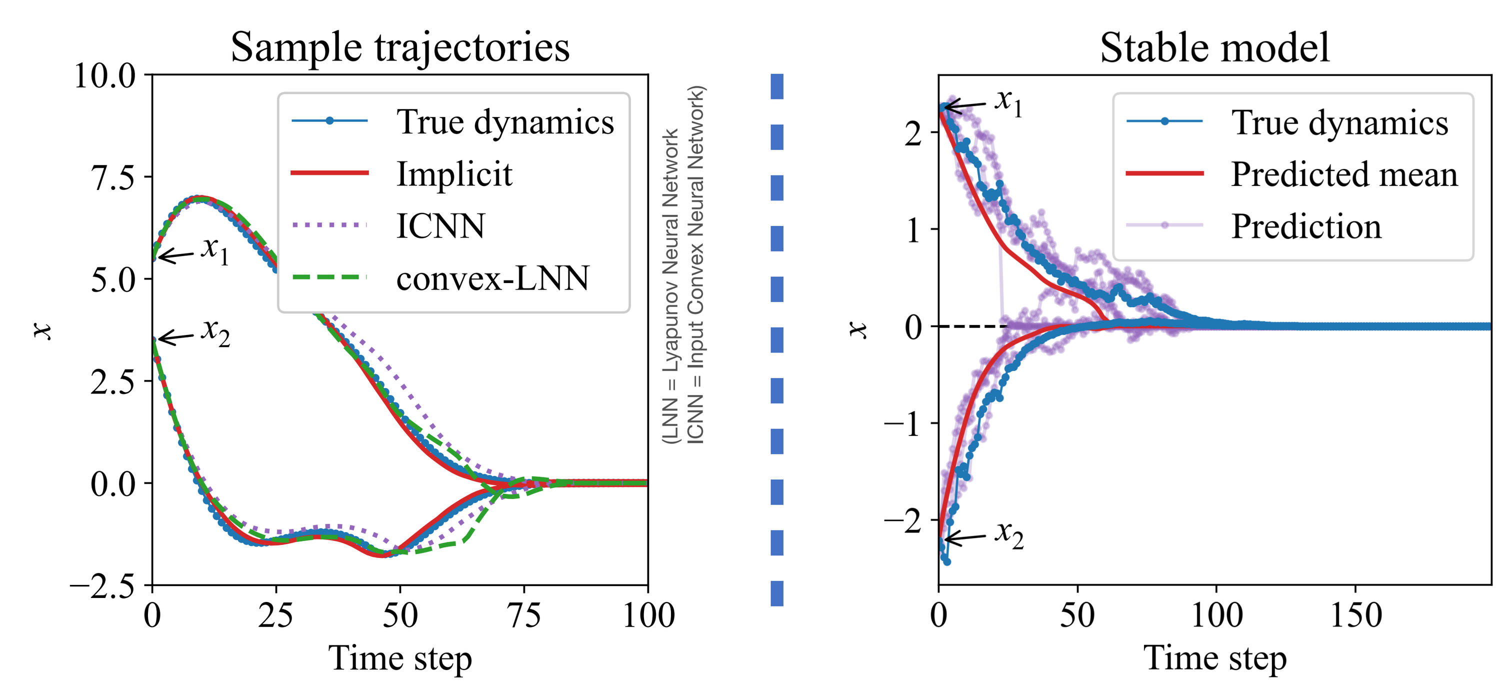
$$V(\mu_{t+1}) < V(\mu_t) \quad \forall t$$

$$\|\Sigma_t\| = \mathcal{O}\left(V(\mu_t)\right) \to 0$$



Examples

Here we show sample trajectories after training on a nonlinear deterministic system, and after training on a nonlinear stochastic ODE. Please see paper for experimental details.



These are separate examples. (Left) Comparing (non-) convex approaches for a deterministic system; (Right) Sample trajectories of a learned model for a stochastic system.

References

1. Manek G, Kolter JZ. Learning stable deep dynamics models. In Advances in Neural Information Processing Systems 2019 (pp. 11128-11136).
2. Khalil HK. Nonlinear systems. Prentice-Hall, 2002.
3. Bishop CM. Mixture density networks. 1994.

https://dais.chbe.ubc.ca/assets/preprints/2020C6_Lawrence_NeurIPS.pdf