Data-Driven Stabilization of Nonlinear Systems with Formal Guarantees*

Work in progress with Giorgos Mamakoukas

Pros and Cons of Data-Driven Control

Pros:

- Often can’t model a system directly from first principles
- Data-driven methods are becoming more accurate, verifiable, general

Cons:

- Can’t know exact error between data-driven model and true system
  - Best we can hope for is a bound on this error
- As a result, control design on data-driven models often lack formal guarantees
  - Example: data-driven stabilization

Stability of Linear Systems

Definitions:

1) A continuous-time linear system $\dot{x} = Ax$ is said to be stable if all eigenvalues of A lie strictly in the left half plane
   - i.e. $Re(\lambda) < 0$ for all λ of A

2) A discrete-time linear system $x_{k+1} = Ax_k$ is said to be stable if all eigenvalues of A lie strictly in the unit circle
   - i.e. $\rho(A) = \max_{\lambda} |\lambda| < 1$

In both cases, if the system is stable, $x \to 0$ exponentially

Stabilization via Control (continuous-time)

Given a linear control system

$$\dot{x} = Ax + Bu,$$

where A is unstable, find gains K such that $A - BK$ is stable.

Define $u = -Kx$, then

$$\dot{x} = Ax + Bu = Ax - BKx = (A - BK)x$$

Q: *What if $\dot{x} = Ax + Bu$ is a data-driven model of an unknown unstable system? Does stabilizing the model stabilize the true system?*

Problem with Data-Driven Stabilization

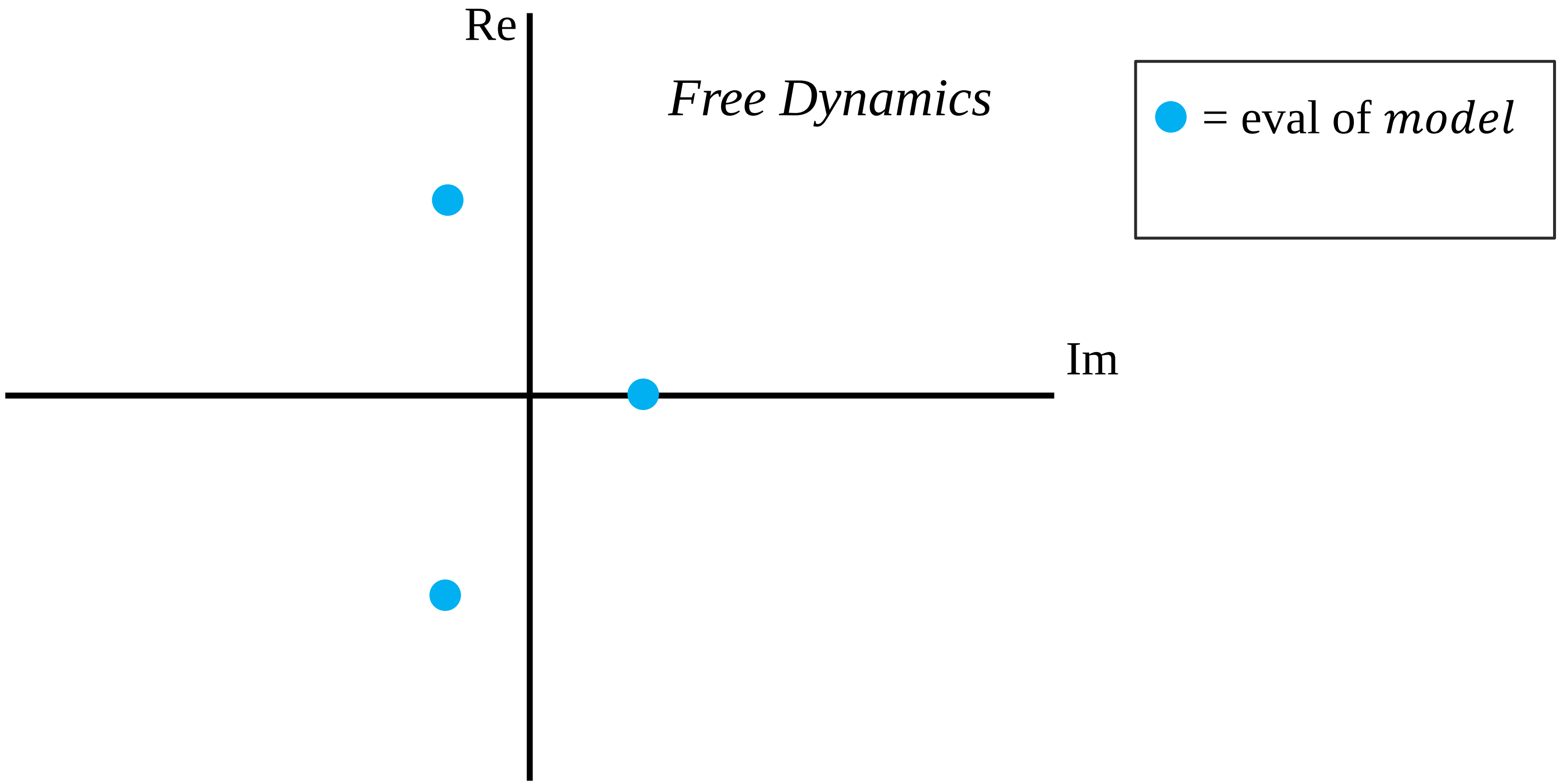
Say you have an unstable nonlinear system: $\dot{x} = f(x, u)$

You use a data-driven method to derive a model: $\dot{x} = Ax + Bu$

Problem with Data-Driven Stabilization

Say you have an unstable nonlinear system: $\dot{x} = f(x, u)$

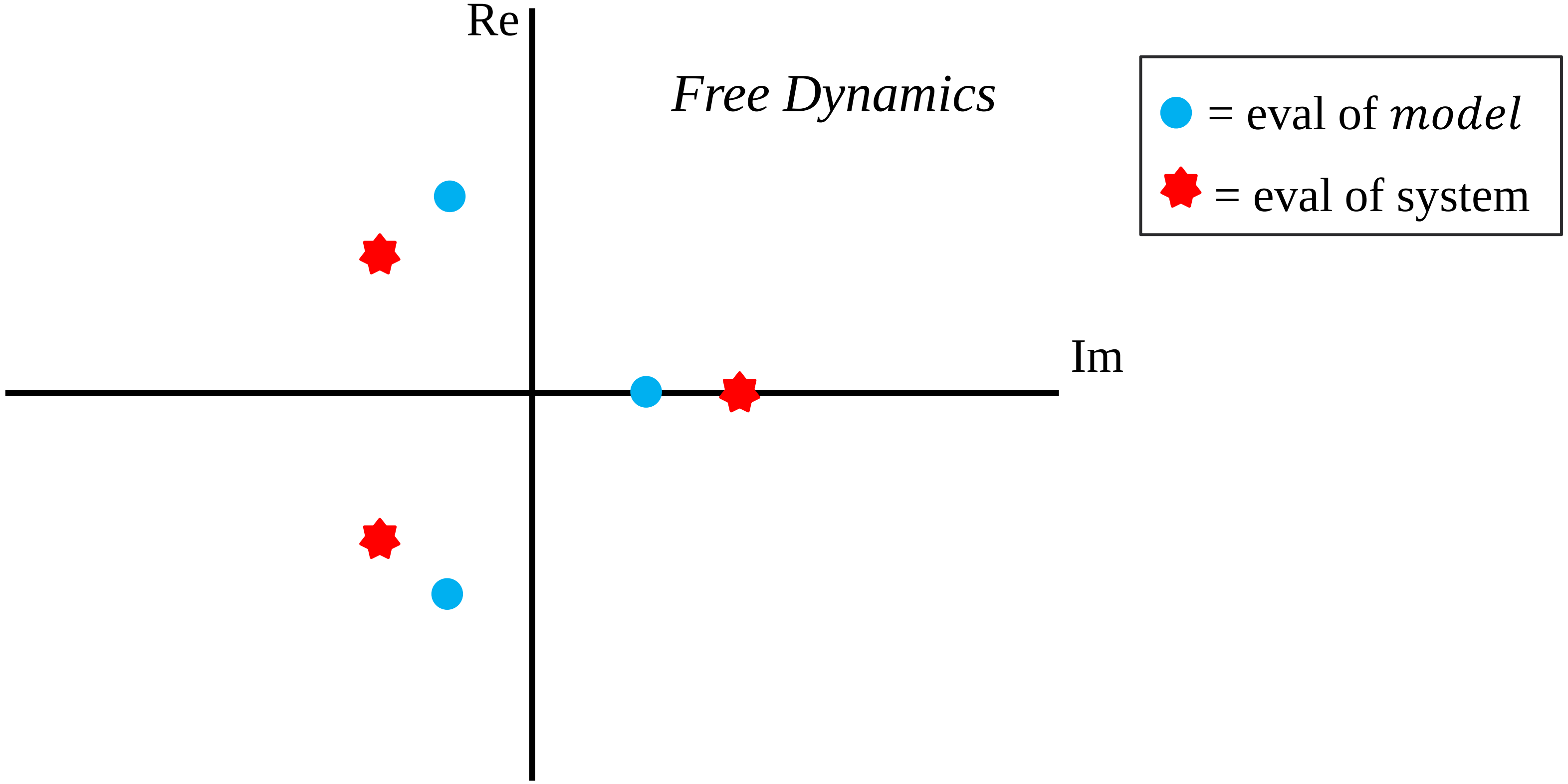
You use a data-driven method to derive a model: $\dot{x} = Ax + Bu$



Problem with Data-Driven Stabilization

Say you have an unstable nonlinear system: $\dot{x} = f(x, u)$

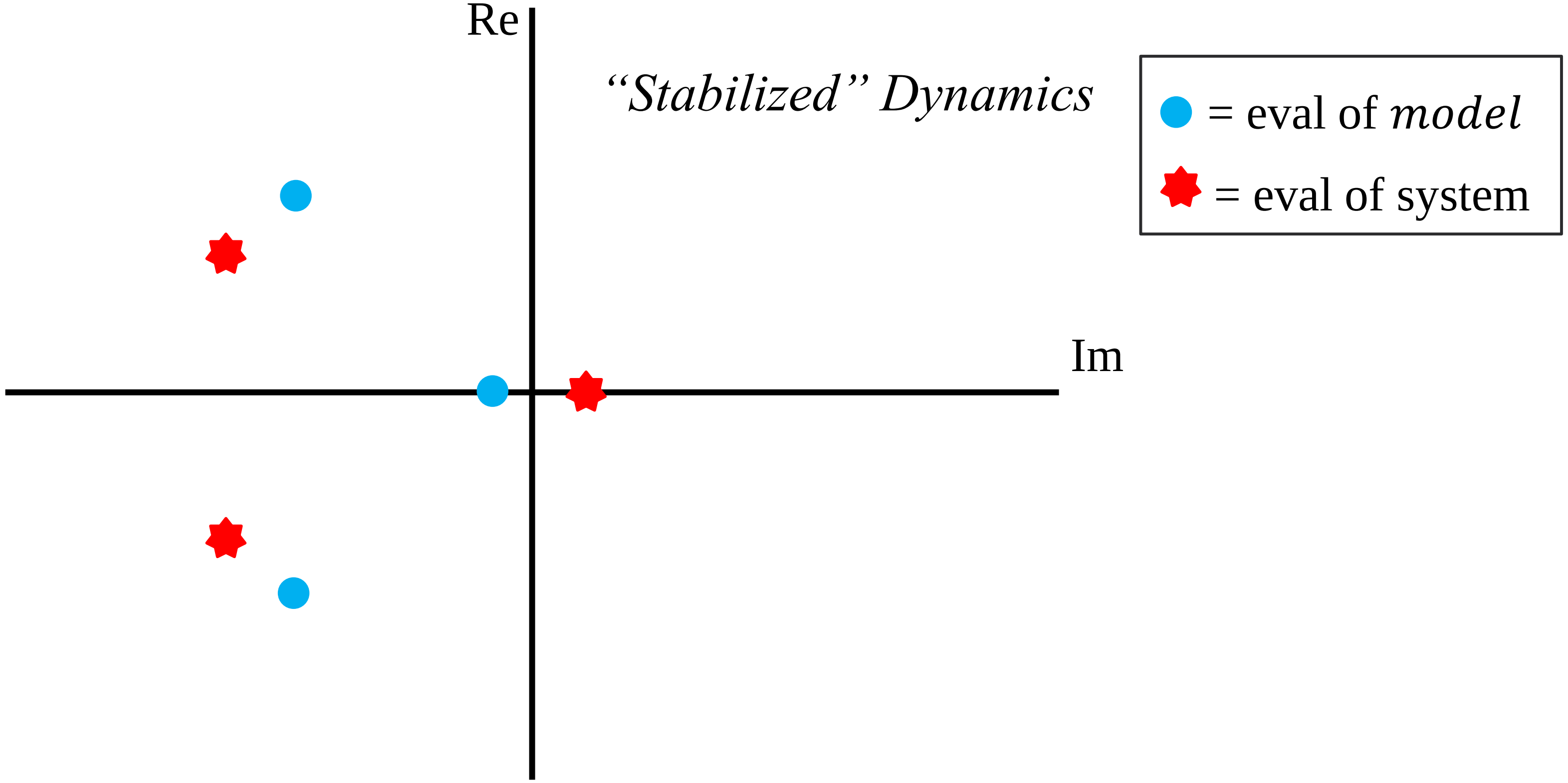
You use a data-driven method to derive a model: $\dot{x} = Ax + Bu$



Problem with Data-Driven Stabilization

Say you have an unstable nonlinear system: $\dot{x} = f(x, u)$

You use a data-driven method to derive a model: $\dot{x} = Ax + Bu$

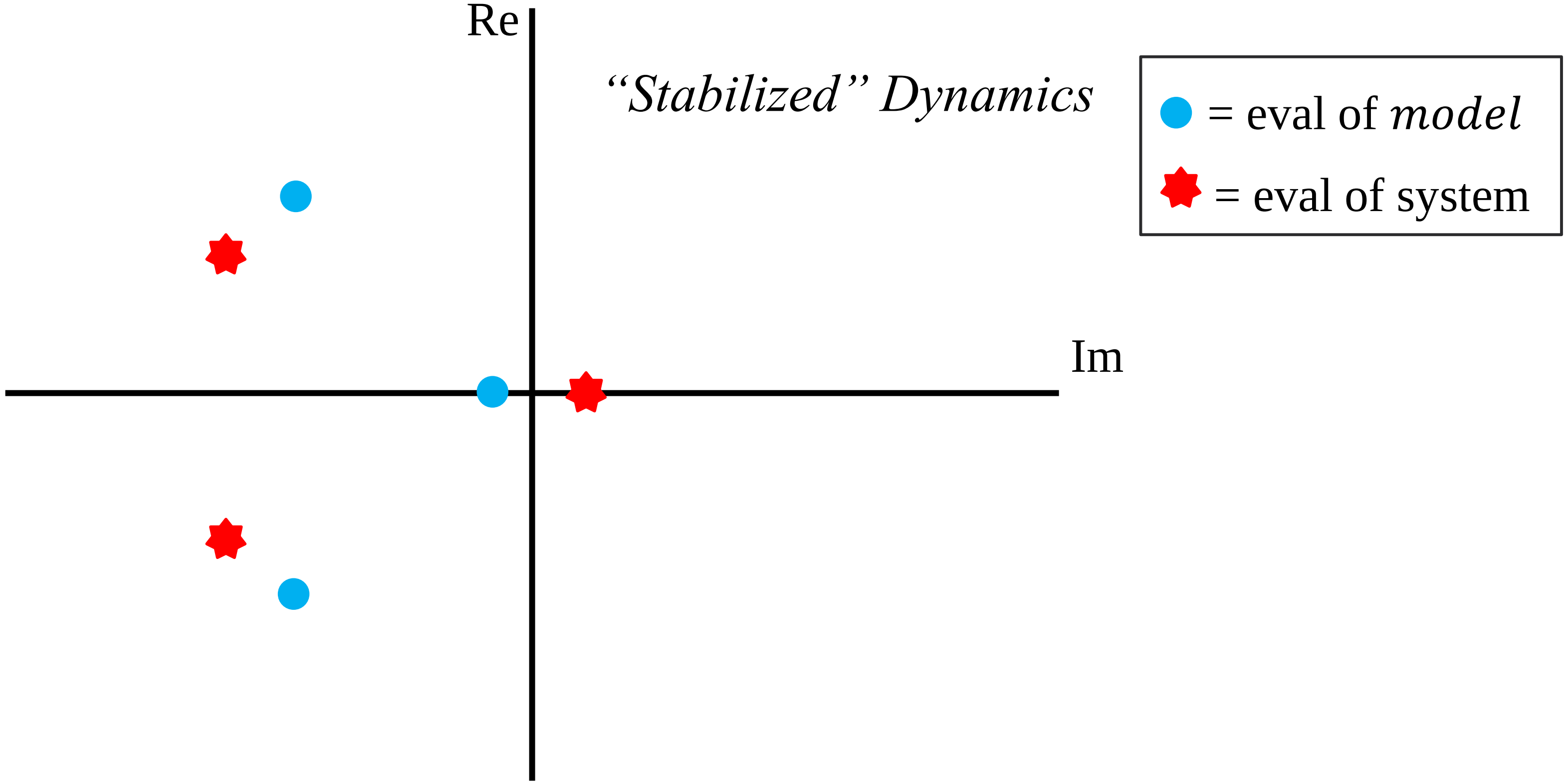


Problem with Data-Driven Stabilization

Say you have an unstable nonlinear system: $\dot{x} = f(x, u)$

You use a data-driven method to derive a model: $\dot{x} = Ax + Bu$

Takeaway:
Naively stabilizing the data-driven model doesn't necessarily stabilize the system



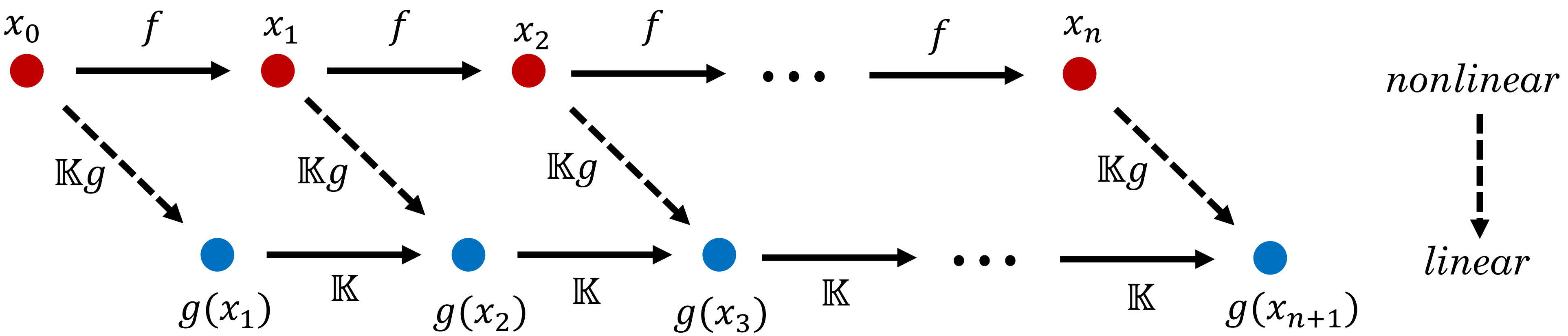
The Koopman Operator

Consider a discrete-time, unforced, nonlinear dynamical system

$$x_{k+1} = f(x_k), \quad x_k \in \Omega \subset \mathbb{R}^n$$

The Koopman operator $\mathbb{K}$ is an infinite-dimensional linear operator that acts on functions of the state $g : \Omega \to \mathbb{C}$

$$[\mathbb{K}g](x_n) = g\big(f(x_n)\big) = g(x_{n+1}).$$



Data-Driven Approximation of $\mathbb{K}$

Take M snapshots of your data: $\{x_m, y_m = x_{m+1}\}_{m=1}^{M}$

Choose dictionary of observables: $\mathcal{D} = \{\psi_1(x), \psi_2(x), \dots, \psi_N(x)\}$

"Lifted" data: $\Psi(x_m) = [\psi_1(x_m), \psi_2(x_m), \dots, \psi_N(x_m)] \in \mathbb{R}^{1\times N}$

Then the approximate Koopman operator is: $\boxed{K = G^+ A} \rightarrow \Psi_{\mathrm{k+1}} = K\Psi_{\mathrm{k}}$

where $+$ is the pseudoinverse and

$$G = \frac{1}{M}\sum_{m=1}^{M} \Psi(x_m)^*\Psi(x_m)\,, \qquad \mathrm{A} = \frac{1}{M}\sum_{m=1}^{M} \Psi(x_m)^*\Psi(y_m)$$

K is a finite dimensional approximation of $\mathbb{K}$ ($\mathbb{K}$ projected onto $\mathcal{D}$)

Eigenvalue-eigenfunction pairs of $\mathbb{K} \approx$ eigenvalue-eigenvector pairs of K

Problems with EDMD

1. “Too much”:
   - Eigenvalues of K that aren’t closed to an eigenvalue of $\mathbb{K}$
   - Call these “spurious” or “unphysical” eigenvalues
2. “Too little”:
   - K misses parts of the spectrum of $\mathbb{K}$
3. Verification:
   - Can’t quantify error between K and $\mathbb{K}$

Enter Residual DMD (ResDMD)

- Work done by Matthew Colbrook
  - Colbrook, Matthew J., Lorna J. Ayton, and Máté Szőke. "Residual dynamic mode decomposition: robust and verified Koopmanism." *Journal of Fluid Mechanics* 955 (2023): A21.
  - Colbrook, Matthew J., and Alex Townsend. "Rigorous data-driven computation of spectral properties of Koopman operators for dynamical systems." *Communications on Pure and Applied Mathematics* 77.1 (2024): 221-283.
- "A very easy wat to get error bounds for your DMD computations!"
- That's right, ResDMD gives error bounds on the eigenvalues of the approximate Koopman model with respect to the spectrum of the true nonlinear system!
- And much more…

ResDMD Computation

There error bounds require no additional data, computational load!

Compute one more matrix: $\mathrm{L} = \frac{1}{M}\sum_{m=1}^{M} \Psi(y_m)^*\Psi(y_m)$

Then, the error of an eigenpair (λ, g) of K relative to the true spectrum of $\mathbb{K}$ is

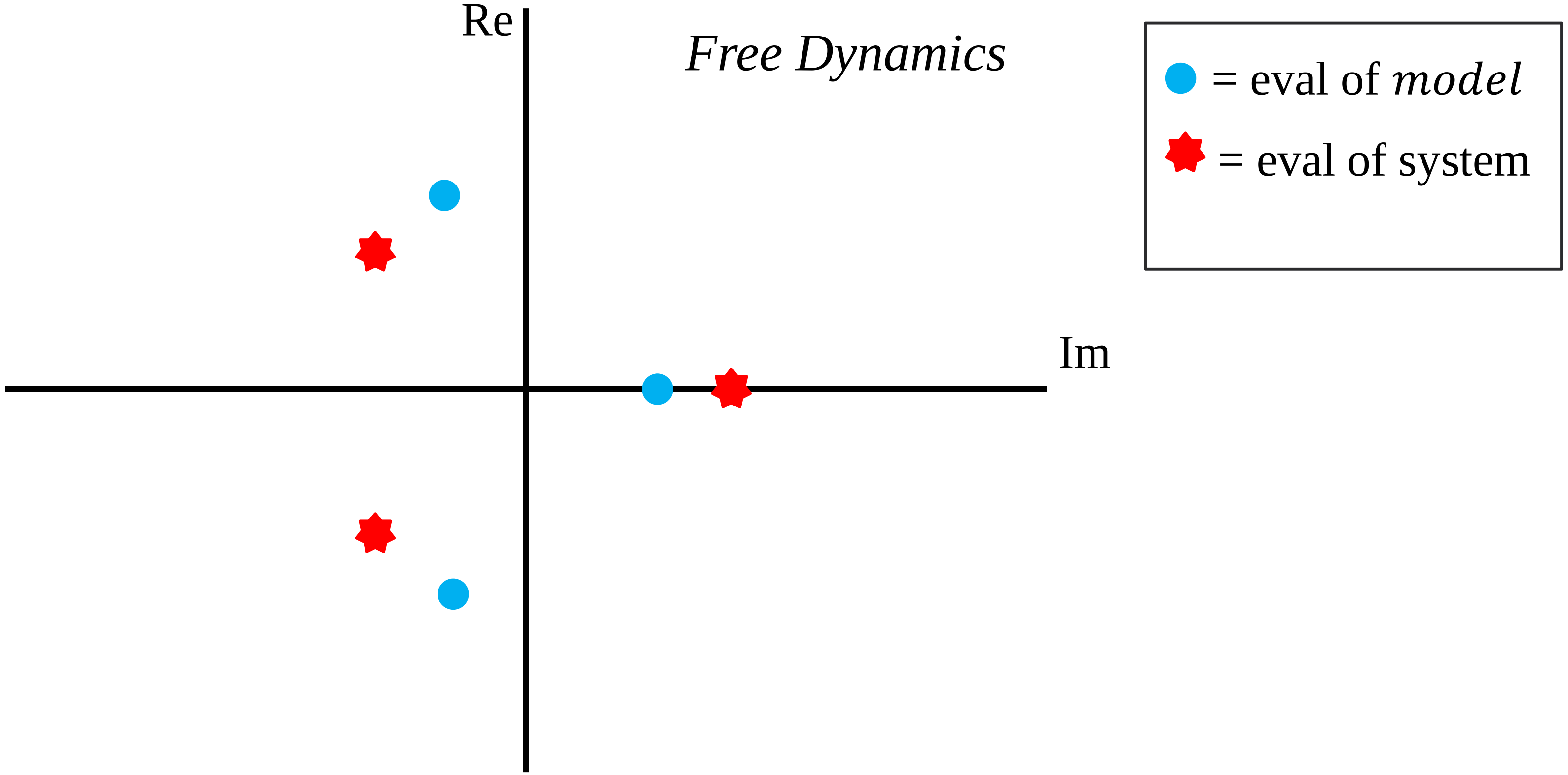
$$\frac{\int_\Omega |[\mathbb{K}g](x) - \lambda g(x)|^2 d\omega(x)}{\int_\Omega |g(x)|^2 d\omega(x)} \approx \frac{g^*\big(L - \lambda A^* - \bar{\lambda} A + |\lambda|^2 G\big)g}{g^* G g} = r(\lambda, g)$$

Further, the error of λ with respect to the eigenvalues of $\mathbb{K}$ is $\leq r(\lambda, g)$

That is, $dist(\lambda, \lambda_{true}) \leq r(\lambda, g)$

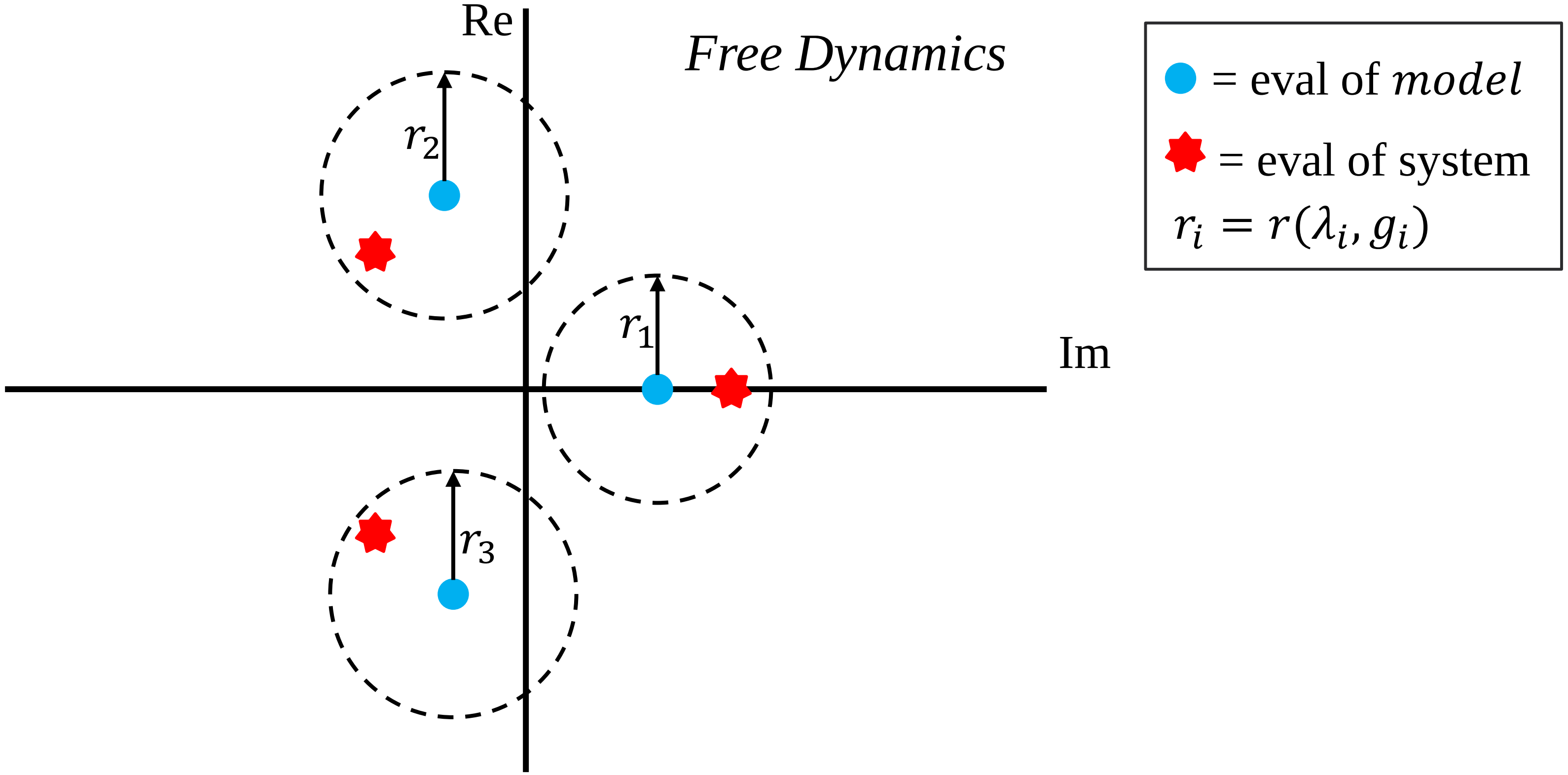
ResDMD Picture

ResDMD gives “error balls” about each eigenvalue of K



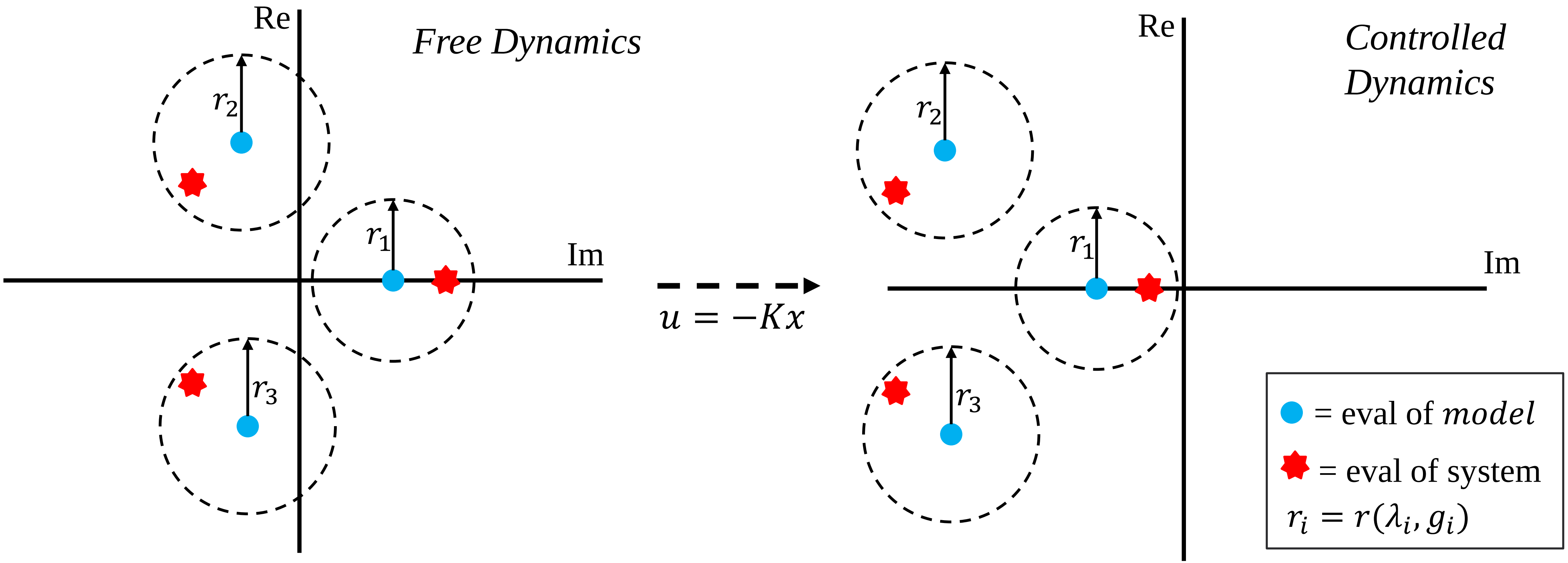
ResDMD Picture

ResDMD gives “error balls” about each eigenvalue of K



My idea: stabilize the error balls!

Synthesize K s.t. $u = -Kx$ pushes the *error balls* into a stable region



EDMD with Control (EDMDc)

Given an unstable control-affine system: $x_{k+1} = f_0(x_k) + \sum_i f_i(x)u_i$

Take M snapshots of state and control: $\{x_m, y_m, u_m\}_{m=1}^{M}$

Add control to dictionary: $\mathcal{D} = \{\psi_1(x), \psi_2(x), \dots, \psi_N(x)\} \cup \{u\}$

Lift data: $\Psi(x_m, u_m) = [\Psi_x(x_m) \mid u_m] = [\psi_1(x_m), \dots, \psi_N(x_m), u_m]$

EDMDc Koopman: $K = [K_x, K_u] = G^{+}A \ \rightarrow \ \Psi_{\mathrm{k+1}} = K_x(\Psi_x)_k + K_u u,$

where

$$G = \frac{1}{M}\sum_{m=1}^{M} \Psi(x_m, u_m)^{*}\Psi(x_m, u_m)\,, \mathrm{A} = \frac{1}{M}\sum_{m=1}^{M} \Psi(x_m, u_m)^{*}\Psi(y_m, u_{m+1})$$

ResDMD with Control (ResDMDc)

Compute A_x, G_x, L_x with only Ψ_x, the state-dependent observables

$$G_x = \frac{1}{M}\sum_{m=1}^{M} \Psi_x(x_m)^* \Psi_x(x_m)\, , \mathrm{A_x} = \frac{1}{M}\sum_{m=1}^{M} \Psi_\mathrm{x}(x_m)^* \Psi_x(y_m)$$

and

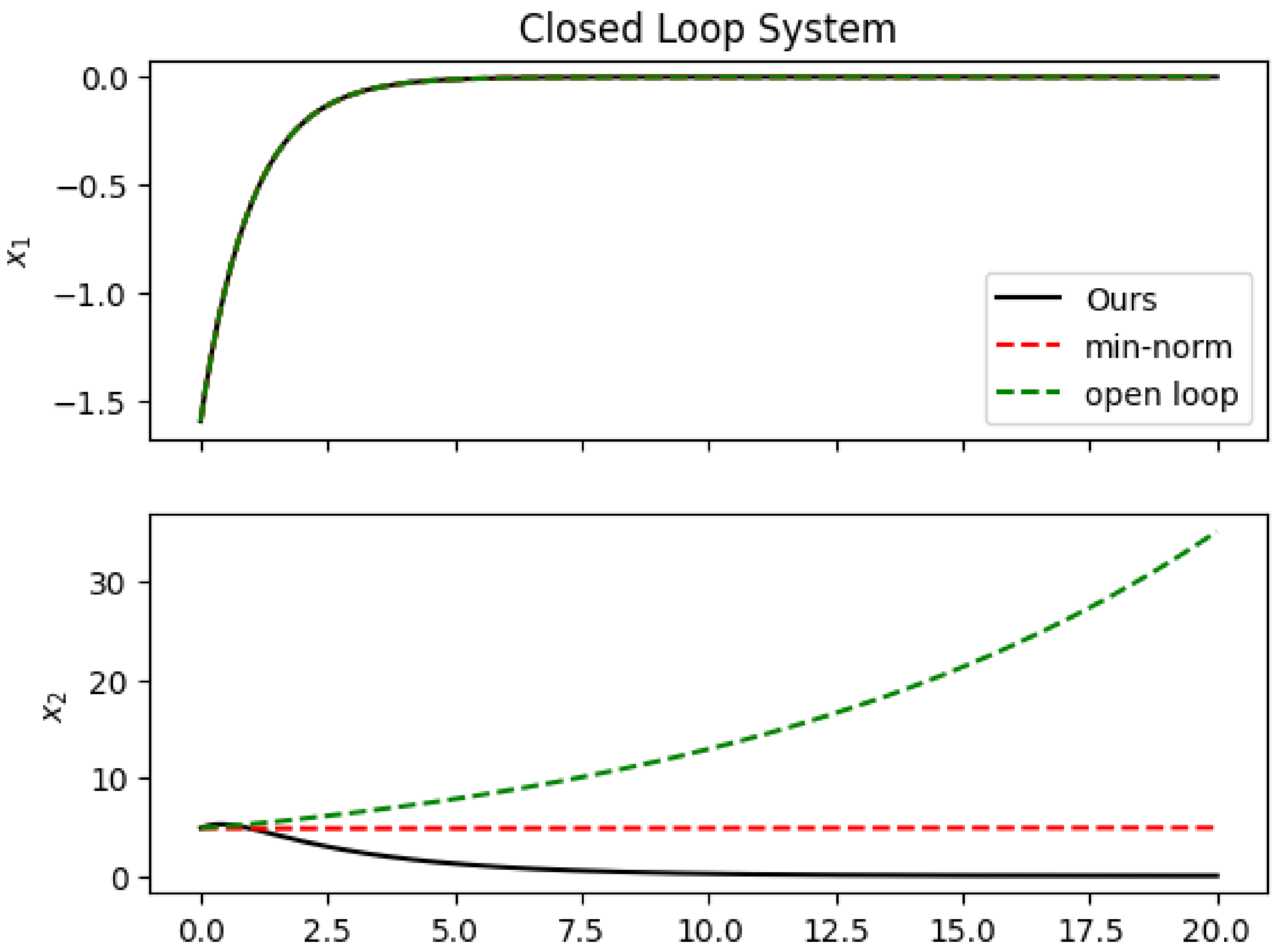
$$\mathrm{L_x} = \frac{1}{M}\sum_{m=1}^{M} \Psi_x(y_m)^* \Psi_\mathrm{x}(y_m)$$

Then, the error of an eigenpair (λ, g) of K_x relative to the true spectrum of $\mathbb{K}_x$ (i.e. the free dynamics $f_0(x)$) is

$$\frac{\int_\Omega |[\mathbb{K}g](x) - \lambda g(x)|^2 d\omega(x)}{\int_\Omega |g(x)|^2 d\omega(x)} \approx \frac{g^*\left(L_x - \lambda A_x^* - \bar{\lambda} A_x + |\lambda|^2 G_x\right) g}{g^* G_x g} = r(\lambda, g)$$

Some Preliminary Results

Consider a 2D diagonal linear system with λ_1 stable and λ_2 only slightly unstable



Simulation Challenges

- Learning strongly unstable systems is really hard
  - Values blow up real fast
- Learning weekly unstable systems typically doesn’t result in large ResDMD errors
- From my experience, EDMDc is more resilient to bad dictionaries than you would expect
  - ResDMD errors stay relatively small
- To synthesize the feedback gains K, I’m using an algorithm that solves an optimization problem
  - It often runs forever or terminates due to “infeasibility”

Obstacle to Guaranteeing Stability