Data-Driven Optimal Targeting Control of Chaotic Dynamical Systems

MAE 546 Optimal Control and Estimation

Samuel Otto

Chaotic Dynamical Systems

- Found throughout nature, science, and engineering
  - Biology: Atrial fibrillation and epilepsy
  - Lightly damped nonlinear structural vibrations
  - Multi-body orbits perhaps encountered in asteroid mining
  - Turbulence, passive and active flow control devices
- Chaotic systems are characterized by
  - Sensitivity to initial conditions
  - Mixing of trajectories in phase space
  - Behavior can appear random or exhibit intermittent quasi-periodicity
- Technically: presence of a chaotic attractor with
  - No stable embedded orbits
  - Topological transitivity (mixing) ➔ ergodic

Controlling Chaos

- Ott, Grebogi, and Yorke (OGY) seminal 1990 paper
  - Unstable periodic orbits embedded in the chaotic attractor can be stabilized with arbitrarily small control
  - Simply wait until trajectory enters sufficiently small neighborhood of desired orbit and activate stabilizing control system.
- Oftentimes, chaos is undesirable, but control over the system is limited
  - Low-power device to stabilize atrial fibrillation
  - Low thrust control of chaotic asteroid orbit for mining
- Chaos may actually be desirable if it can be controlled
  - Enables diverse behavior of the system
  - Low power covert communication

Targeting Control

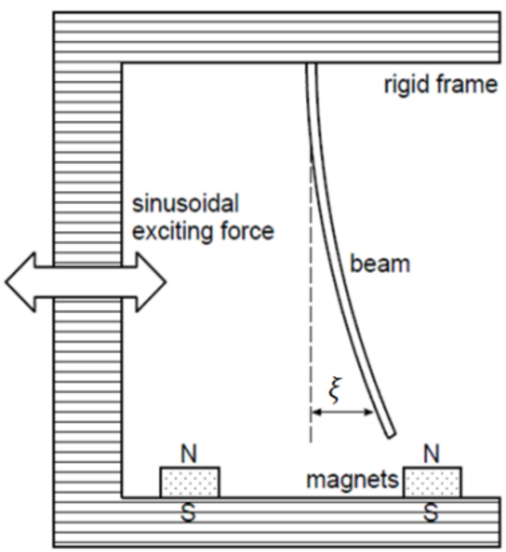
- Problem: We may have to wait an extremely long time before a chaotic trajectory enters a sufficiently small neighborhood of the desired orbit
- Solution: (Targeting) use small perturbations to guide trajectories toward the desired orbit -- reducing time to stabilize it
- Two methods considered here:
    1. Neighboring optimal control near a nominal targeting path or tree leading to the desired orbit
    2. Reinforcement learning to construct optimal targeting controller over the entire attractor

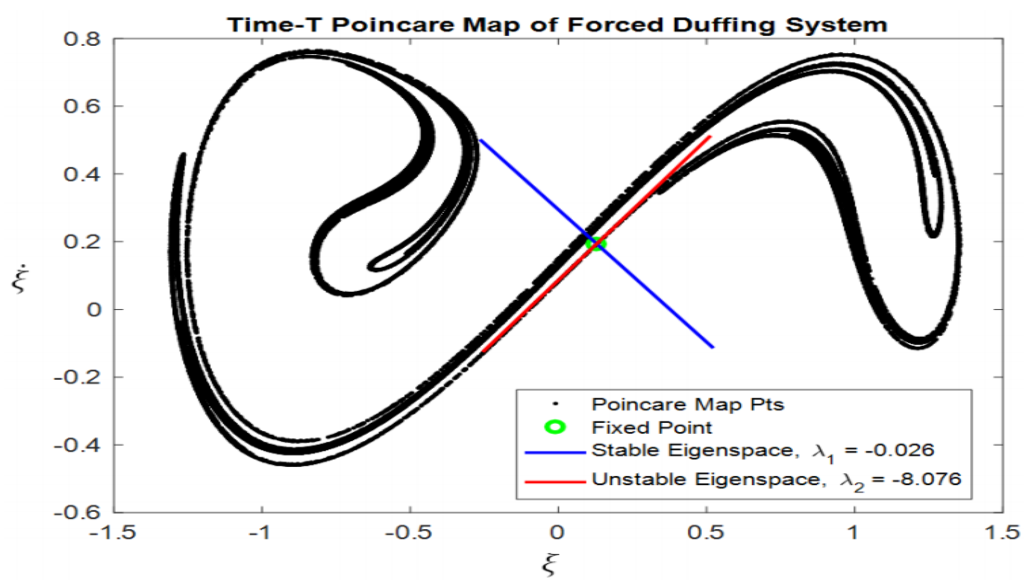
Example Problem: Forced Duffing Eqn.

$$\ddot{\xi} + \delta\dot{\xi} + \alpha\xi + \beta\xi^3 = \gamma\cos(\omega t) + p(t), \qquad \boldsymbol{x} = \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix}$$

- Forced oscillator with cubic nonlinear stiffness
- Exhibits transverse homoclinic tangle of stable and unstable manifolds in time $T = 2\pi/\omega$ Poincare map
- Discrete time control $\boldsymbol{u}_n = [u(1), \dots, u(5)]^T, t \in [nT, (n+1)T]$

$$p(t) = u(1) + u(2)\cos\omega t + u(3)\sin\omega t + u(4)\cos 2\omega t + u(5)\sin 2\omega t$$





Entirely Data-Driven Approach

- Oftentimes, we do not have access to a model of the system’s dynamics which can be evaluated quickly enough to design control systems
- We will infer accurate and efficient models of highly nonlinear chaotic dynamics using data alone.
  - **All controllers designed using the learned dynamics model only!**
- Challenges with Chaotic Poincare maps
  - Highly nonlinear discrete time dynamics
  - Filamented, fractal structure of data manifolds in Poincare map
- Global models with enough terms to capture nonlinearity are inefficient and tend to over-fit
- Local nonlinear modeling is preferable
  - Each model has fewer terms ➔ efficient predictions
  - Arbitrarily complex dynamics are represented by adding more models, not by increasing model complexity.

Local Nonlinear Models in Bayesian Framework

- Novel approach to system modeling and analysis
- Each simple model $i = 1,2,\ldots,N$ takes the form

$$\boldsymbol{X}_{n+1}^{i} = \hat{\boldsymbol{f}}^{i}(\boldsymbol{x}_n, \boldsymbol{u}_n) + \boldsymbol{V}^{i}, \qquad \boldsymbol{V}^{i} \sim p_{\boldsymbol{V}^{i}}(\boldsymbol{v}^{i}) = \mathcal{N}\left(\boldsymbol{v}^{i}, \boldsymbol{0}, R^{i}\right)$$

- Latent random variable $Z \in \{1,2,\ldots,N\}$ indicating the model. Categorical prior distribution

$$P(Z = i) = \phi^{i}, \qquad \phi^{1} + \phi^{2} + \cdots + \phi^{N} = 1$$

- Each model is associated with a region of validity defined by a Gaussian density

$$p_{\boldsymbol{X}|Z=i}(\boldsymbol{x}|Z = i) = \mathcal{N}\left(\boldsymbol{x}, \boldsymbol{\mu}_{x}^{i}, \Sigma^{i}\right)$$

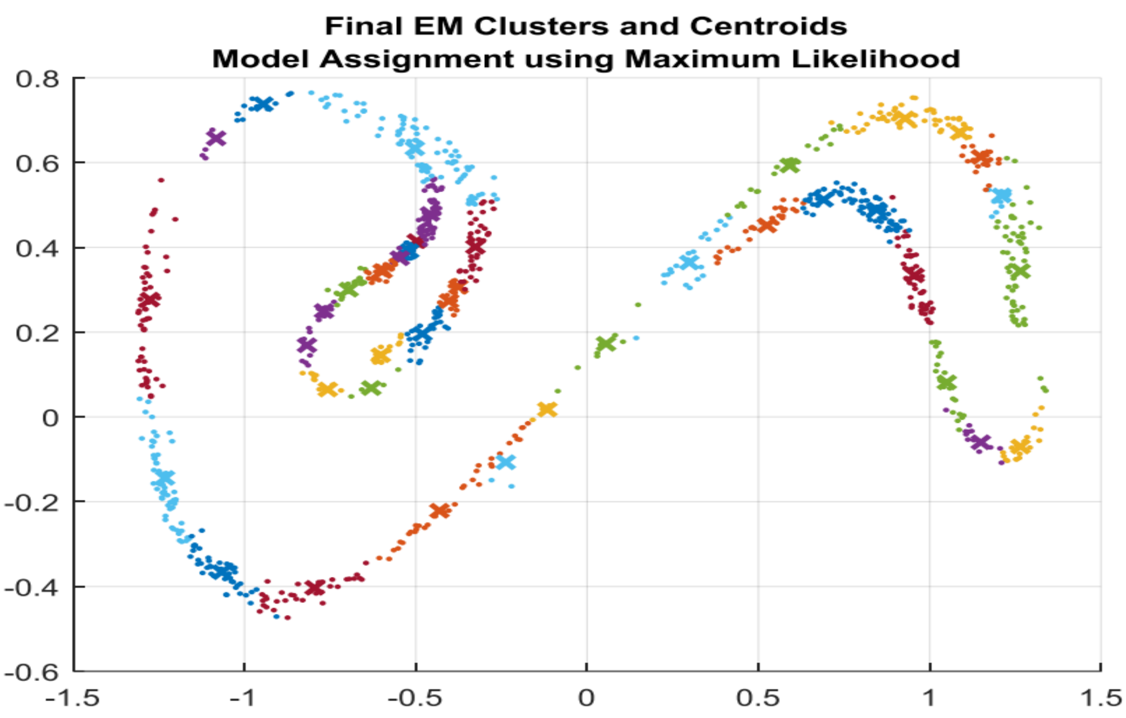
- Bayes rule to infer model probabilities at $\boldsymbol{x}$

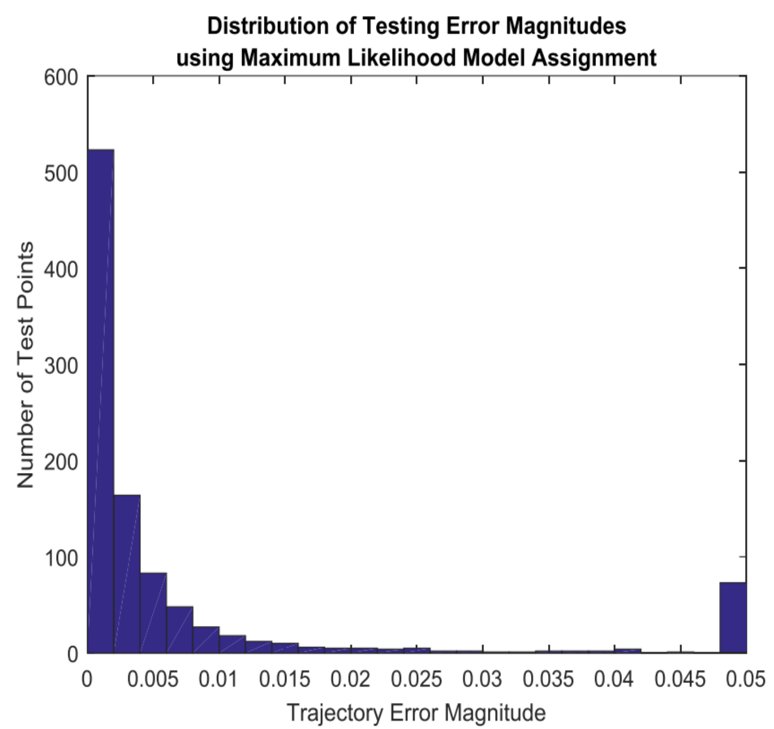
$$P(Z = i|\boldsymbol{X} = \boldsymbol{x}) = \left[\sum_{k=1}^{N} p_{\boldsymbol{X}|Z=k}(\boldsymbol{x}|Z = k)P(Z = k)\right]^{-1} p_{\boldsymbol{X}|Z=i}(\boldsymbol{x}|Z = i)P(Z = i)$$

Local Nonlinear Models in Bayesian Framework

- All parameters trained according to maximum likelihood criterion (min cross entropy) using the Expectation Maximization (EM) algorithm
- Nonlinear kernel regression used to build local models
- Maximum likelihood model assignment

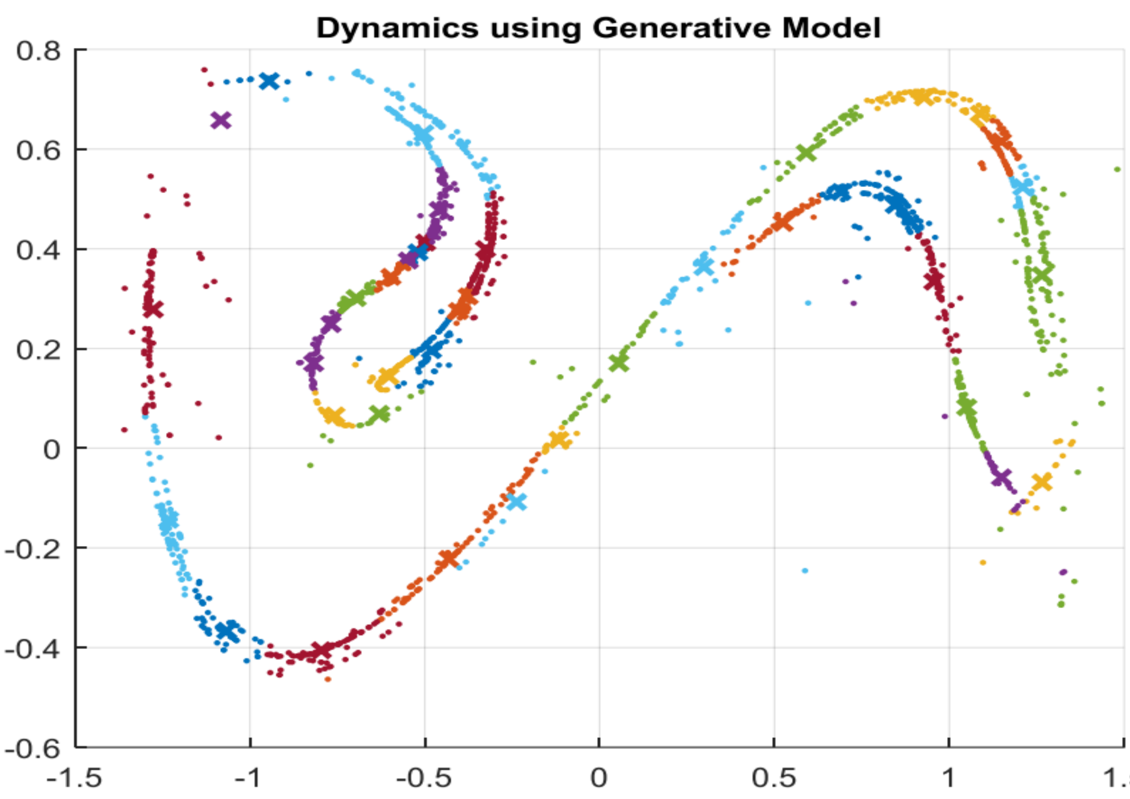
$$\hat{\boldsymbol{x}}_{n+1} = \hat{\boldsymbol{f}}(\boldsymbol{x}_n, \boldsymbol{u}_n) = \hat{\boldsymbol{f}}^{i^*}(\boldsymbol{x}_n, \boldsymbol{u}_n), \qquad i^* = \underset{i \in \{1,\ldots,N\}}{\operatorname{argmax}} P(Z = i | \boldsymbol{X} = \boldsymbol{x}_n)$$

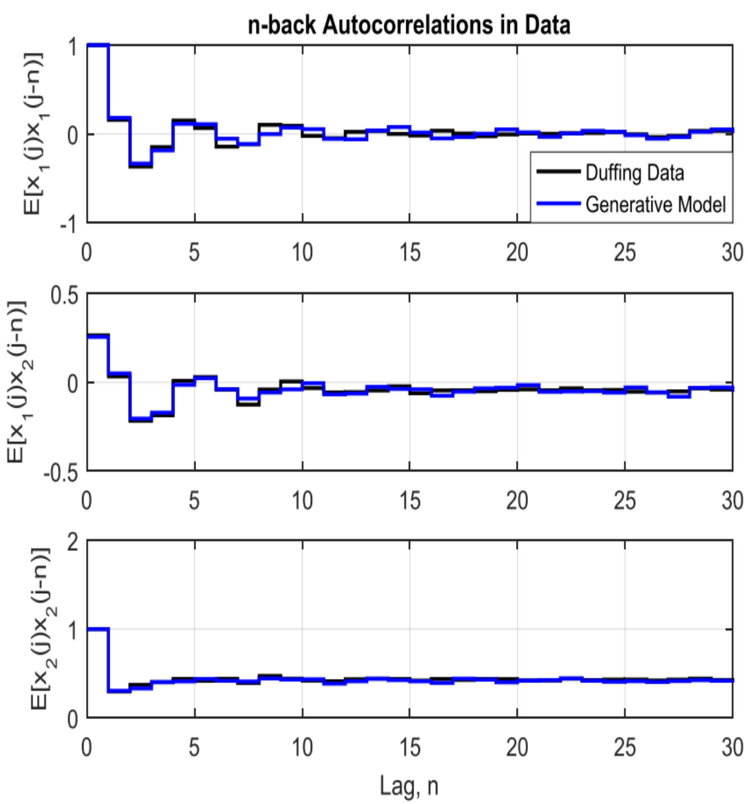




Performance as Generative Model

- Does the learned model approximate the statistical behavior of the real system?
  - Look at (unforced) Poincare map generated by the model
  - Look at autocorrelations for real and modeled system





OGY Control using LQR

- A fixed point of the Poincare map is located and linearized using the learned model

$$\widehat{\boldsymbol{x}}_{FP} = \widehat{\boldsymbol{f}}(\widehat{\boldsymbol{x}}_{FP}, \boldsymbol{0}), \qquad \Delta \boldsymbol{x}_n = \boldsymbol{x}_n - \widehat{\boldsymbol{x}}_{FP},$$
$$\widehat{\Phi}_{FP} = D_{\boldsymbol{x}} \widehat{\boldsymbol{f}}(\widehat{\boldsymbol{x}}_{FP}, \boldsymbol{0}), \qquad \widehat{\Gamma}_{FP} = D_{\boldsymbol{u}} \widehat{\boldsymbol{f}}(\widehat{\boldsymbol{x}}_{FP}, \boldsymbol{0})$$

- A Linear Quadratic Regulator was designed to stabilize the unstable periodic orbit and minimize the following cost

$$J^{FP} = \frac{1}{2} \sum_{n=1}^{\infty} \left[\Delta \boldsymbol{x}_n^T Q_{FP} \Delta \boldsymbol{x}_n + \boldsymbol{u}_n R_{FP} \boldsymbol{u}_n \right], \qquad \Delta \boldsymbol{x}_{n+1} = \widehat{\Phi}_{FP} \Delta \boldsymbol{x}_n + \widehat{\Gamma}_{FP} \boldsymbol{u}_n$$

- The optimal feedback gain C_{FP} is found and OGY control is implemented

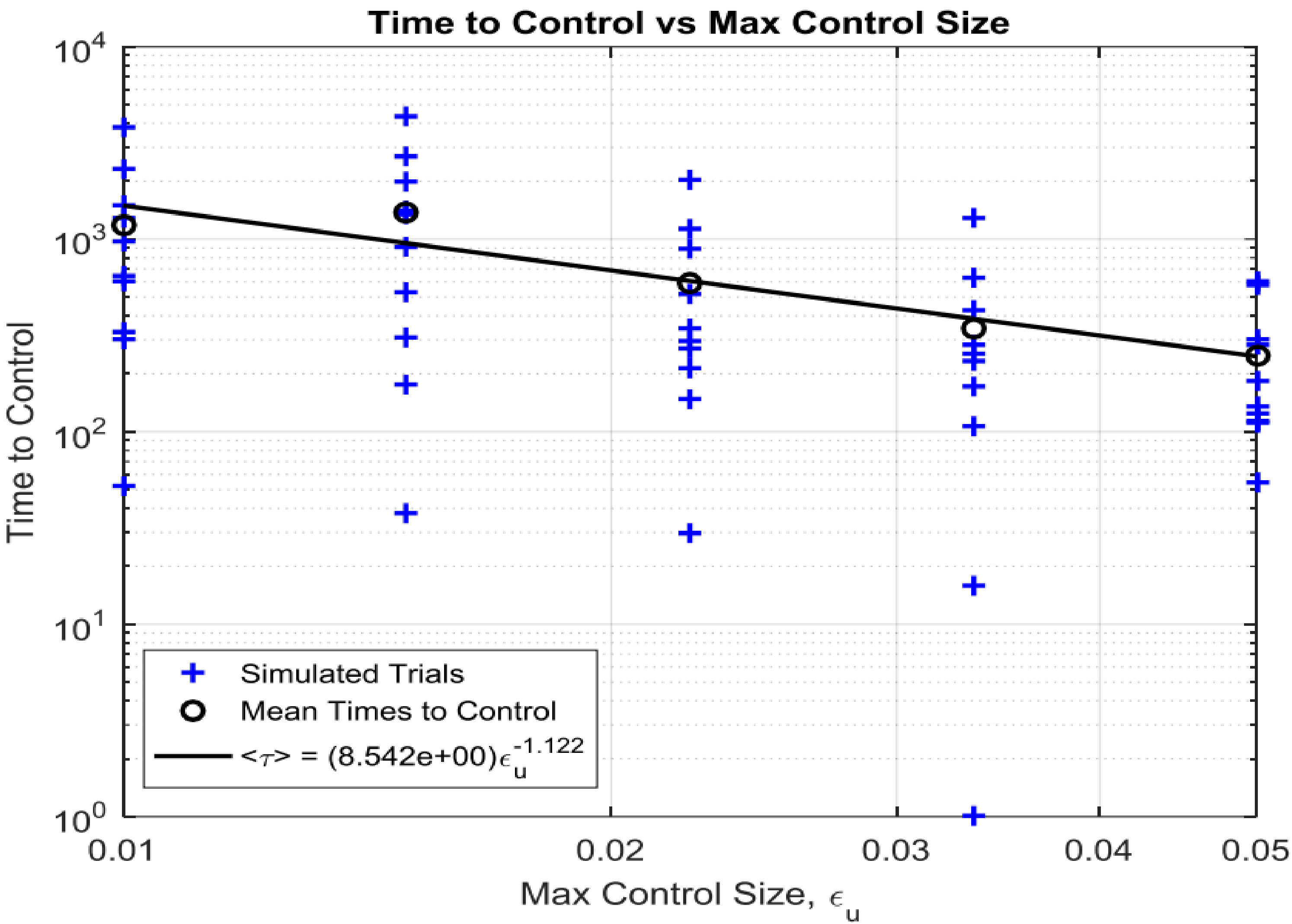
$$\boldsymbol{u}_n^{(OGY)} = \begin{cases} \widetilde{\boldsymbol{u}}_n & \widetilde{\boldsymbol{u}}_n^T G_W \widetilde{\boldsymbol{u}}_n \leq \epsilon_u^2 \\ \boldsymbol{0} & otherwise \end{cases}, \qquad \widetilde{\boldsymbol{u}}_n = -C_{FP} \Delta \boldsymbol{x}_n$$

- Design choices:

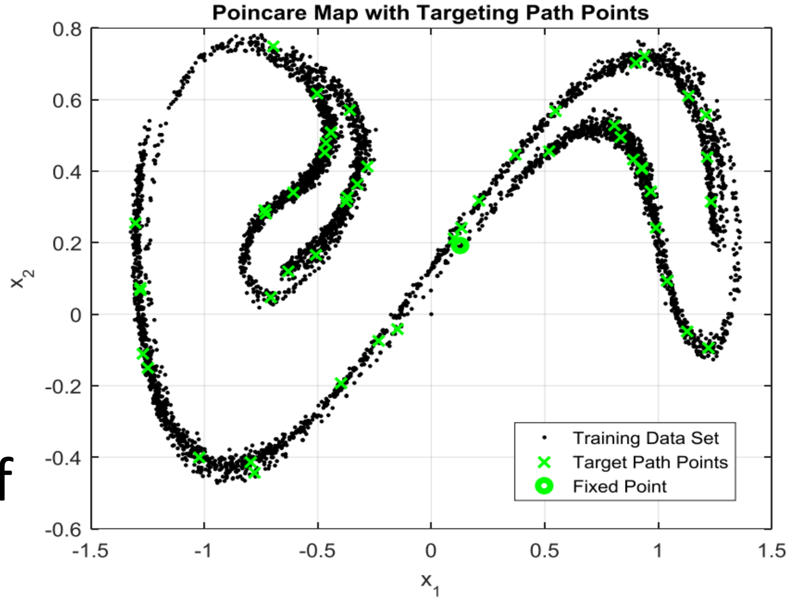
$$Q_{FP} = (0.1) I_2, \qquad R_{FP} = G_W = diag[2\pi, \pi, \pi, \pi, \pi]$$

Time to Stabilize Scaling with Control Size

- Numerical experiments performed to determine scaling of expected time to control using the OGY method and LQR at $\hat{\boldsymbol{x}}_{FP}$



Neighboring Optimal Control using Targeting Path



- A path consisting of $\mathrm{M} = 50$ unforced training points $\{\overline{\boldsymbol{x}}_n\}_{n=1}^{M}$ leading to the neighborhood of $\widehat{\boldsymbol{x}}_{FP}$ was selected as the nominal targeting path
  - Fast mixing time ➔ good coverage of attractor
- A neighboring optimal controller was designed to capture points near the targeting path $\Delta\boldsymbol{x}_n = \boldsymbol{x}_n - \overline{\boldsymbol{x}}_n$
- The following cost function with geometric weighting of the state was minimized with model-linearized dynamics

$$J = \frac{1}{2}\sum_{n=1}^{M}[\gamma^{n-M}\Delta\boldsymbol{x}_n^T Q \Delta\boldsymbol{x}_n + \boldsymbol{u}_n^T R \boldsymbol{u}_n] + \frac{1}{2}\Delta\boldsymbol{x}_M^T Q \Delta\boldsymbol{x}_M, \qquad \gamma \geq 1$$

$$\Delta\boldsymbol{x}_{n+1} = \widehat{\Phi}_n \Delta\boldsymbol{x}_n + \widehat{\Gamma}_n \boldsymbol{u}_n, \qquad \widehat{\Phi}_n = D_{\boldsymbol{x}}\widehat{\boldsymbol{f}}(\overline{\boldsymbol{x}}_n, \boldsymbol{0}), \qquad \widehat{\Gamma}_n = D_{\boldsymbol{u}}\widehat{\boldsymbol{f}}(\overline{\boldsymbol{x}}_n, \boldsymbol{0})$$

Neighboring Optimal Control using Targeting Path

- Bellman's equations were solved to perform the minimization

$$V_n^*(\Delta \boldsymbol{x}_n) = \min_{\boldsymbol{u}_n} \left[\frac{1}{2}\gamma^{n-m}\Delta \boldsymbol{x}_n^T Q \Delta \boldsymbol{x}_n + \frac{1}{2}\boldsymbol{u}_n^T R \boldsymbol{u}_n + V_{n+1}^*\big(\widehat{\Phi}_n \Delta \boldsymbol{x}_n + \widehat{\Gamma}_n \boldsymbol{u}_n\big)\right]$$

- The optimal value function takes the form $V_n^*(\Delta \boldsymbol{x}_n) = \frac{1}{2}\Delta \boldsymbol{x}_n^T P_n \Delta \boldsymbol{x}_n$ with $P_M = Q$. This gives optimal control

$$\widetilde{\boldsymbol{u}}_n^* = -\big(R + \widehat{\Gamma}_n^T P_{n+1} \widehat{\Gamma}_n\big)^{-1} \widehat{\Gamma}_n^T P_{n+1} \widehat{\Phi}_n \Delta \boldsymbol{x}_n = -C_n \Delta \boldsymbol{x}_n$$

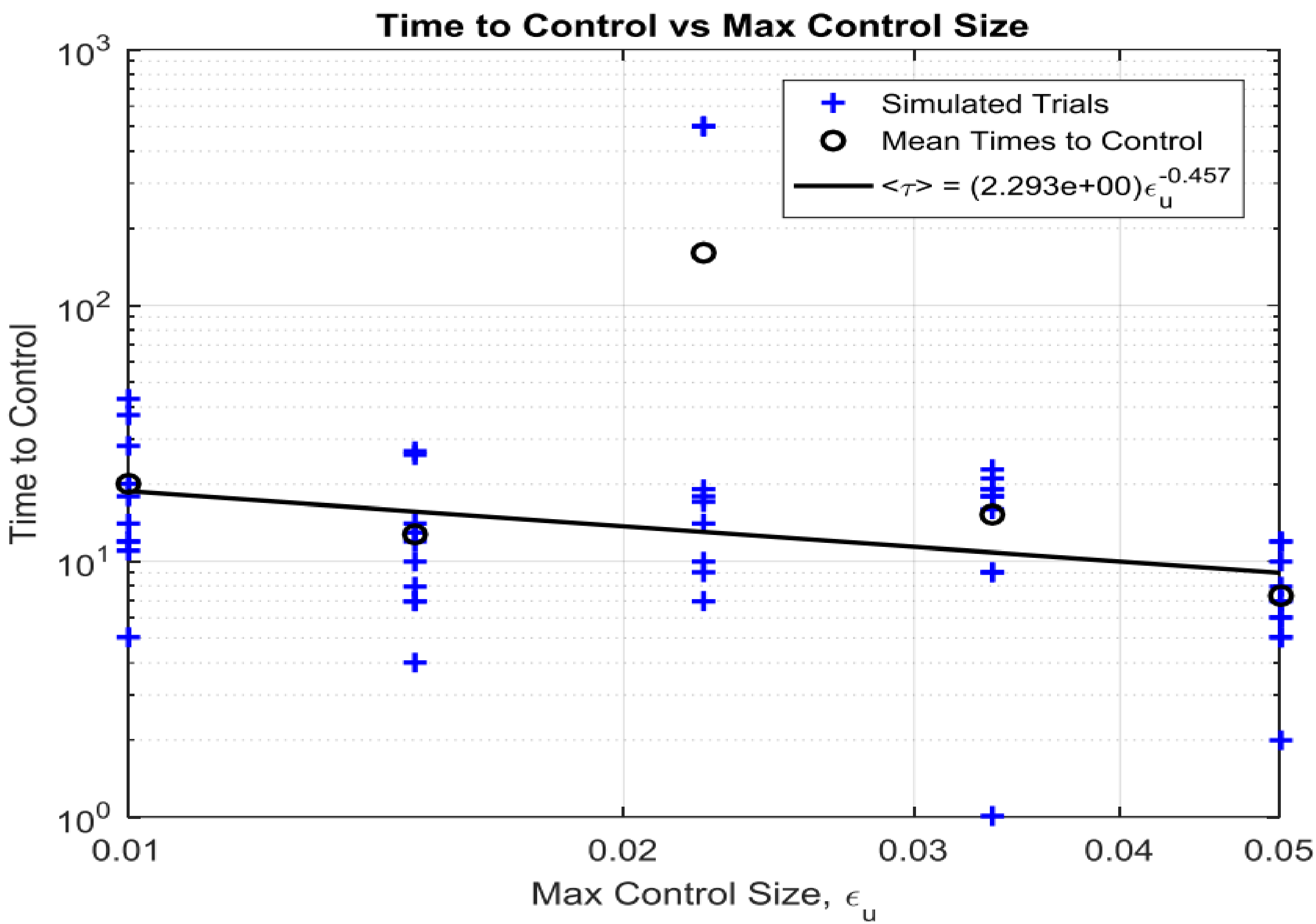
- And the discrete time algebraic Riccati equation

$$P_n = \gamma^{n-M} Q + \widehat{\Phi}_n^T P_{n+1} \widehat{\Phi}_n - \widehat{\Phi}_n^T P_{n+1} \widehat{\Gamma}_n \big(R + \widehat{\Gamma}_n^T P_{n+1} \widehat{\Gamma}_n\big)^{-1} \widehat{\Gamma}_n^T P_{n+1} \widehat{\Phi}_n$$

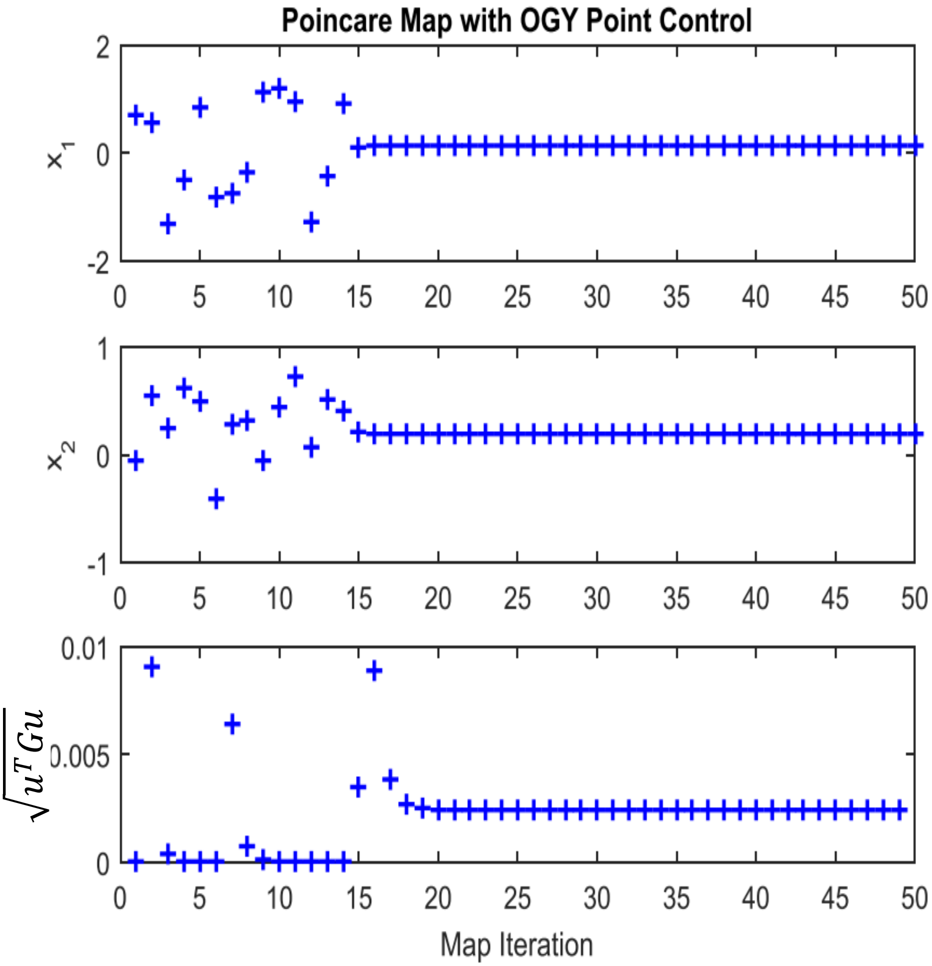
Time to Stabilize Scaling with Control Size

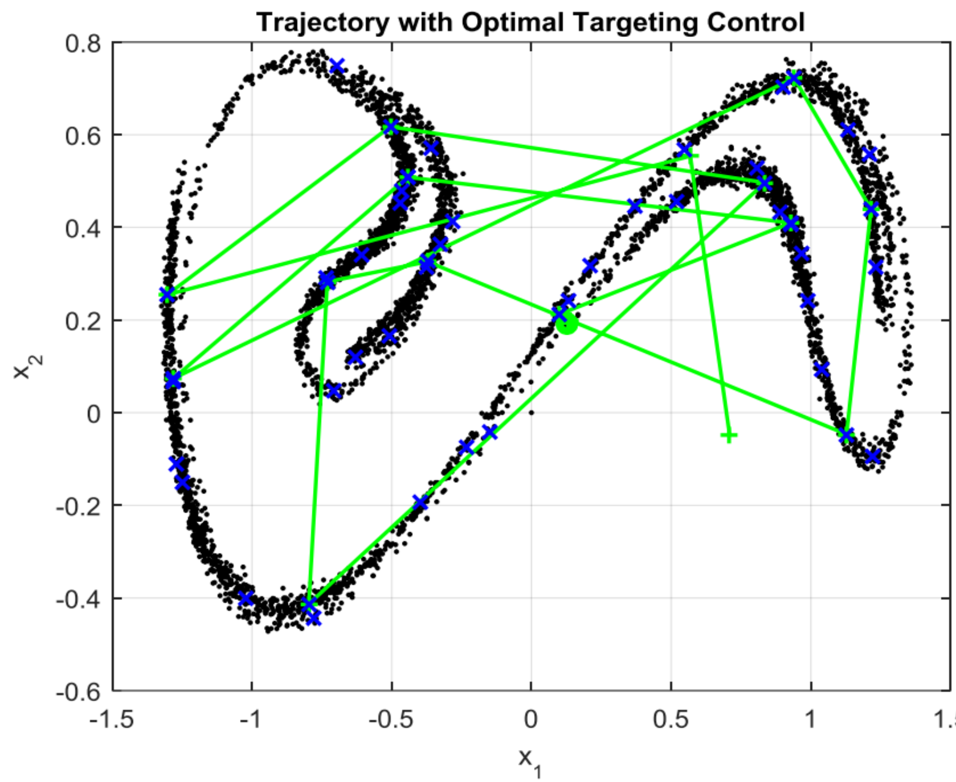
- The following design choices were made:

$$\gamma = 1.1, \qquad Q = C_{FP}^T G_W C_{FP} + (0.7) I_2, \qquad R = (10^{-3}) I_5$$



An Example Trajectory with $\epsilon_u = 0.01$





Optimal Control using Reinforcement Learning

- The following constrained optimization problem was posed over the entire attractor with $\Delta \boldsymbol{x}_n = \boldsymbol{x}_n - \widehat{\boldsymbol{x}}_{FP}$

$$\text{minimize } J = \frac{1}{2}\sum_{n=1}^{\infty} \beta^{n-1}[\Delta \boldsymbol{x}_n^T Q \Delta \boldsymbol{x}_n + \boldsymbol{u}_n^T R \boldsymbol{u}_n] \quad s.t. \quad \boldsymbol{u}_n^T G_W \boldsymbol{u}_n \leq \epsilon_u^2$$

- Subject to the nonlinear modeled dynamics $\boldsymbol{x}_{n+1} = \widehat{\boldsymbol{f}}(\boldsymbol{x}_n, \boldsymbol{u}_n)$ and geometric decay $0 < \beta \leq 1$
- The optimal value function over the attractor is introduced and Bellman's equations are formulated

$$V^*(\boldsymbol{x}) = \min_{\boldsymbol{u}^T G_W \boldsymbol{u} \leq \epsilon_u^2} \frac{1}{2}[\Delta \boldsymbol{x}^T Q \Delta \boldsymbol{x} + \boldsymbol{u}^T R \boldsymbol{u}] + \beta V^*\left(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{u})\right)$$

Fitted Value Iteration

- A subset of the training points $\{\overline{\boldsymbol{x}}_j\}_{j=1}^{M}$ are chosen and used to defined the value function globally by interpolation (or regression).
  - Natural neighbor interpolation with 500 points was used
- The value function at each point $V_j^* = V^*(\overline{\boldsymbol{x}}_j)$ is updated by performing the following minimization

$$V_j^* \leftarrow \min_{\boldsymbol{u}^T G_W \boldsymbol{u} \le \epsilon_u^2} \frac{1}{2}\left[\left(\overline{\boldsymbol{x}}_j - \widehat{\boldsymbol{x}}_{FP}\right)^T Q\left(\overline{\boldsymbol{x}}_j - \widehat{\boldsymbol{x}}_{FP}\right) + \boldsymbol{u}^T R \boldsymbol{u}\right] + \beta V^*\left(\widehat{\boldsymbol{f}}\left(\overline{\boldsymbol{x}}_j, \boldsymbol{u}\right)\right)$$

  - Matlab’s fmincon() was used for minimization
- Iterate until the value function converges

Optimal Value Function and Control

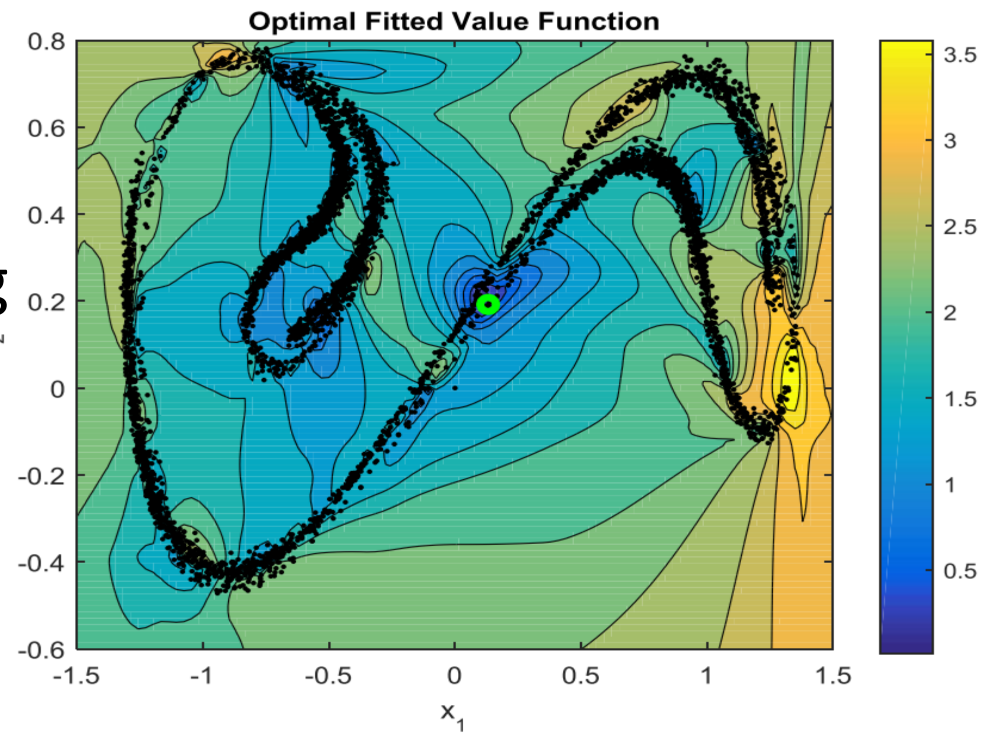
- The following design choices were made

$$\beta = 1, \qquad Q = C_{FP}^{T} G_W C_{FP} + (0.1) I_2, \qquad R = (0.1) I_5, \qquad \epsilon_u = 0.01$$

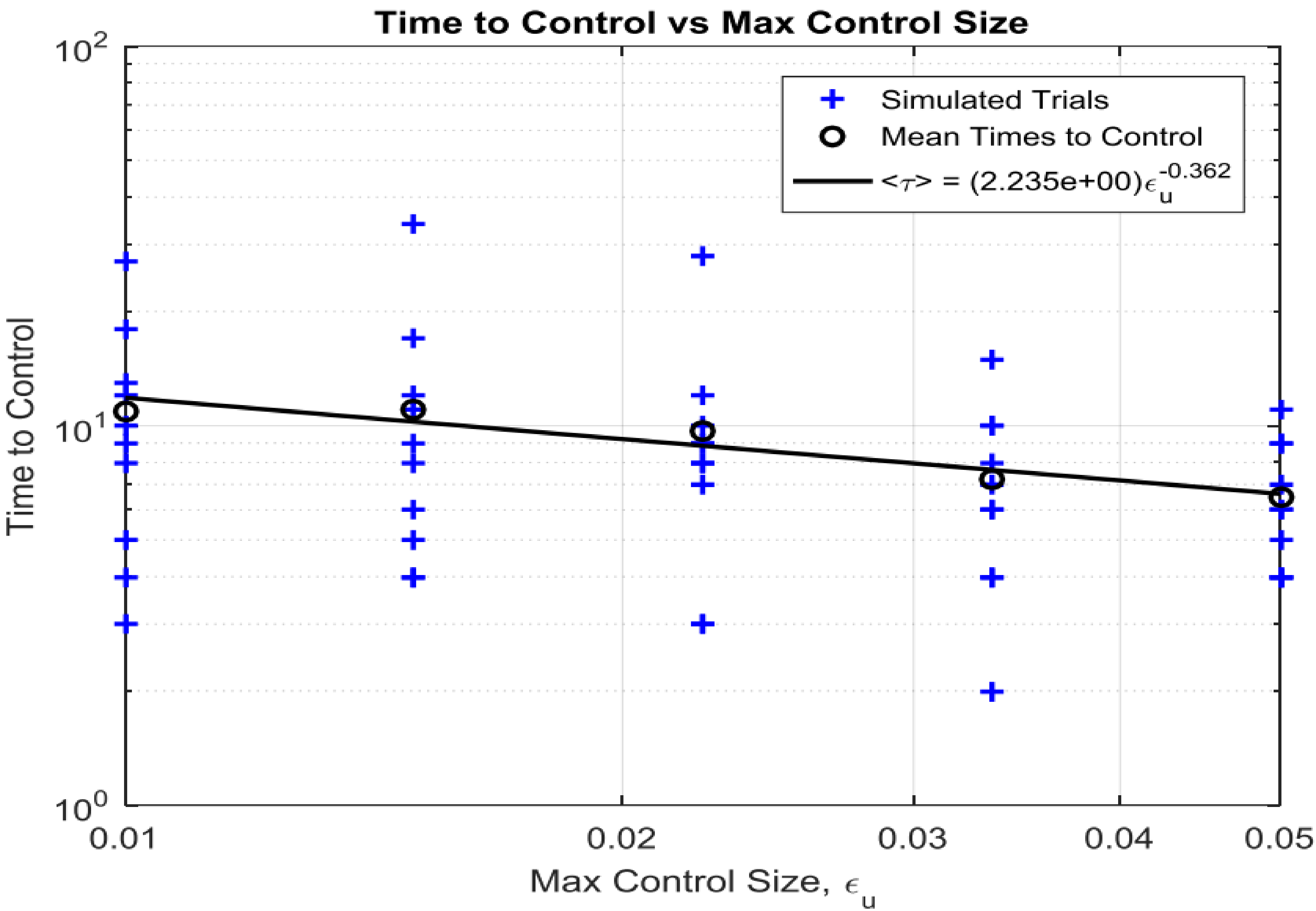
- The optimal control at each step was determined by performing a minimization

$$\boldsymbol{u}^* = \underset{\boldsymbol{u}^T G_W \boldsymbol{u} \le \epsilon_u^2}{\operatorname{argmin}} \frac{1}{2}\left[(\boldsymbol{x} - \widehat{\boldsymbol{x}}_{FP})^T Q (\boldsymbol{x} - \widehat{\boldsymbol{x}}_{FP}) + \boldsymbol{u}^T R \boldsymbol{u}\right] + \beta V^*\left(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{u})\right)$$

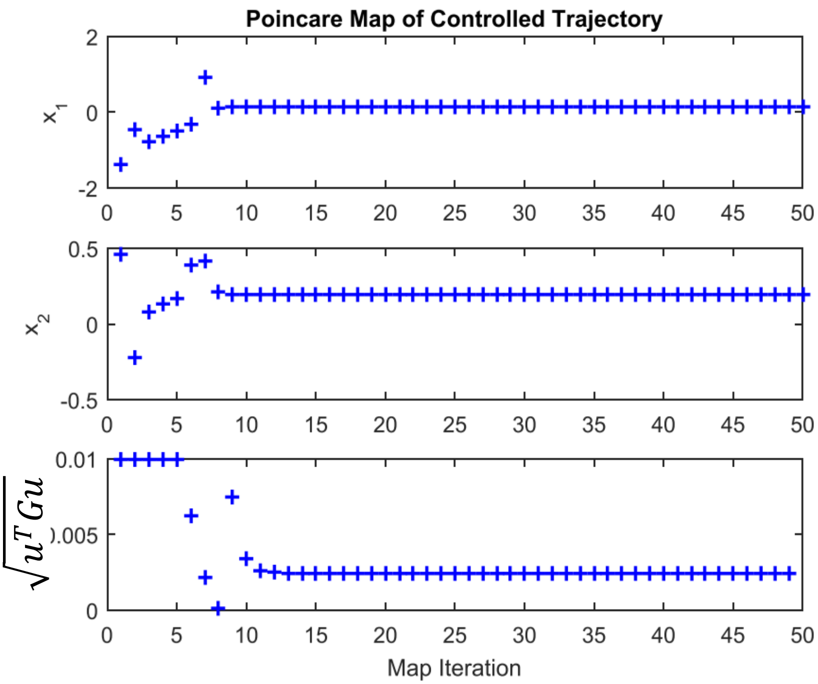
- (<u>Note</u>: It is possible to find the control at many points and *build a model for the optimal control directly* using a collection of nonlinear models and EM as with the dynamics)

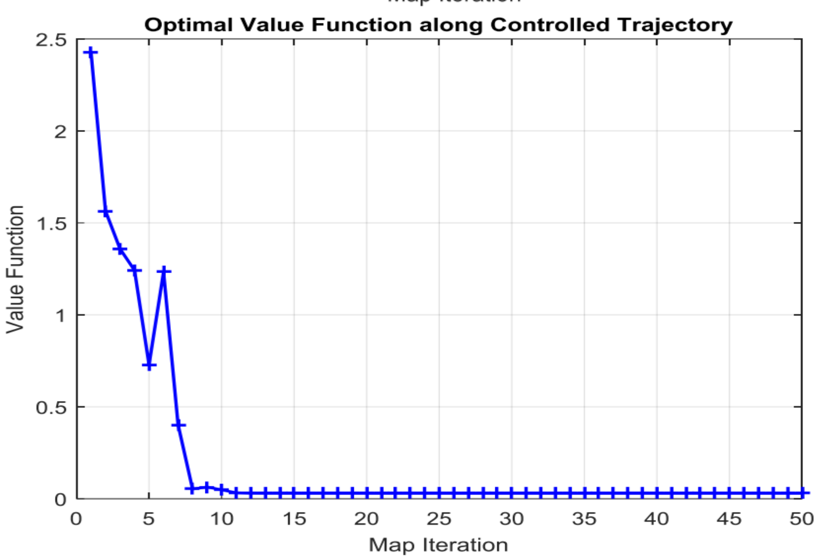


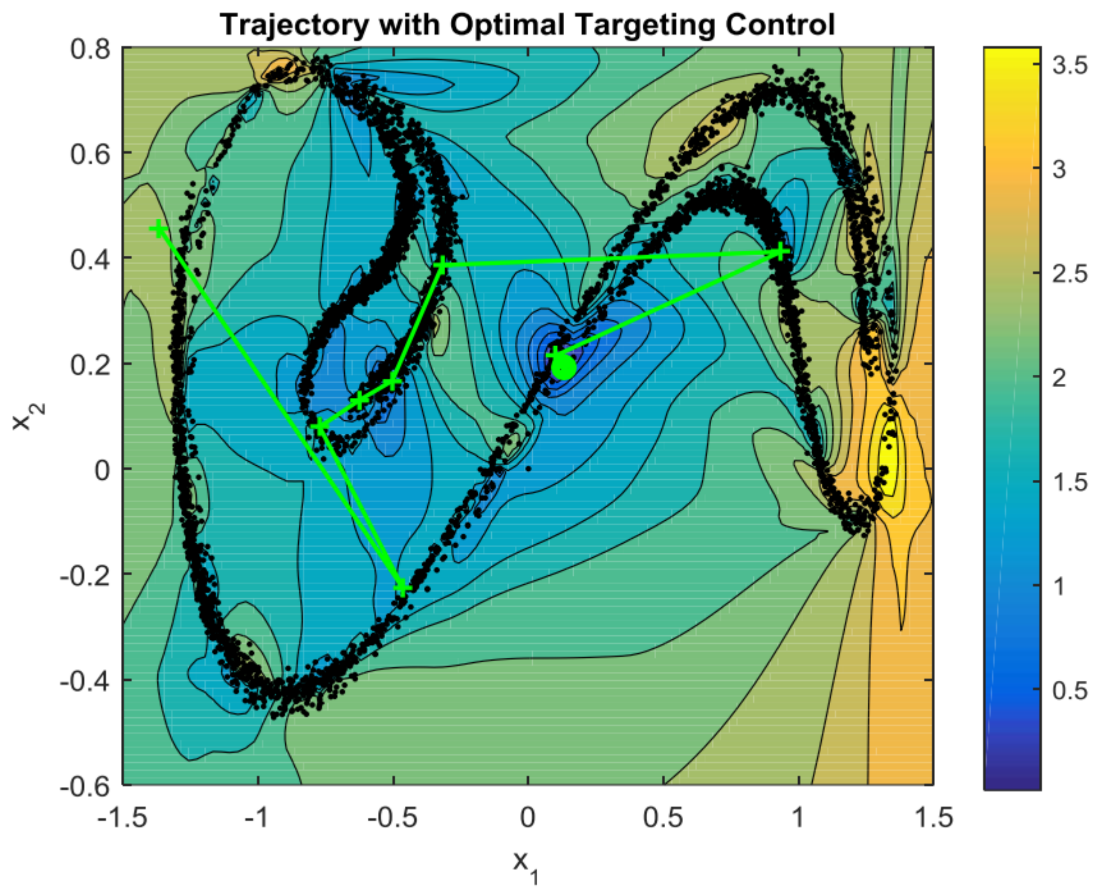
Time to Stabilize Scaling with Control Size



An Example Trajectory with $\epsilon_u = 0.01$







Conclusion

- Novel data-driven modeling technique was introduced
- Accurate and efficient representation of nonlinear dynamics enabled the design of optimal targeting controllers
- Both targeting controllers show almost two orders of magnitude reduction in time to control over OGY only.
- In its current implementation, the reinforcement learning approach is expensive
  - this cost can be reduced by learning a model for the optimal control
- Future work includes using the local nonlinear modeling technique for state estimation
  - Multiple model estimation with extended or quasilinear Kalman filter

Questions?