# Reinforcement Learning Based ADP

# Optimal Control Problem

**Problem**

Design a controller $u$ that minimizes the cost

$$J(x,u) = \int_0^\infty r\left(x(\tau), u(x(\tau))\right) d\tau$$

$$r(x^o, u^o) = Q(x^o) + u^{oT} R u^o$$

Subject to dynamic constraints

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

**Exact Solution**

Optimal value function

$$V^*(x^o) \triangleq \min_{u(\tau)\in U | \tau \in \mathbb{R}_{\geq t}} \int_t^\infty r\left(\phi^u(\tau; t, x^o), u(\tau)\right) d\tau$$

HJB equation

$$0 = \min_{u^o \in U} \left(\nabla V^*(x^o)\left(f(x^o) + g(x^o)\, u^o\right) + r(x^o, u^o)\right)$$

Optimal policy

$$u^*(x^o) = -\tfrac{1}{2} R^{-1} g^T(x^o) \left(\nabla V^*(x^o)\right)^T$$

# Challenges

## Uncertainty

- How to make the best possible decision in the presence of uncertainty, right now
- How to solve the exploration vs. exploitation problem, while also performing system identification
  - Bellman Error extrapolation (simulation of experience) = simultaneous exploration and exploitation
  - Concurrent learning = on-line data-based system identification
  - **Sparse Neural Networks (CDC 2019)**

## Expensive

- Curse of dimensionality – large computational cost
  - StaF approximation
  - Sparse NN approximation

## More complex problems

- How to include constraints, embed logic-based decision making, intermittency, ….
  - Formal methods, hybrid/switched systems ADP, scalability
  - **Application: Orbital transfer differential game (SciTech 2020)**

# Approximate BE Extrapolation

## Uncertainty

**Bellman error**

$$\delta\left(x,\hat{W}_c,\hat{W}_a\right)=r\left(x,\hat{u}\left(x,\hat{W}_a\right)\right)+\nabla\hat{V}\left(x,\hat{W}_c\right)\left(f\left(x\right)+g\left(x\right)\hat{u}\left(x,\hat{W}_a\right)\right)$$

**Parametric approximation** $\hat{f}\left(x,\hat{\theta}\right)$

**Approximate Bellman error**

$$\hat{\delta}\left(x,\hat{W}_c,\hat{W}_a,\hat{\theta}\right)=r\left(x,\hat{u}\left(x,\hat{W}_a\right)\right)+\nabla\hat{V}\left(x,\hat{W}_c\right)\left(Y\left(x\right)\hat{\theta}+g\left(x\right)\hat{u}\left(x,\hat{W}_a\right)\right)$$

$$\hat{\delta}_t\left(t\right)=\hat{\delta}\left(x\left(t\right),\hat{W}_c\left(t\right),\hat{W}_a\left(t\right),\hat{\theta}\left(t\right)\right)\qquad\hat{\delta}_{ti}\left(t\right)=\hat{\delta}\left(x_i,\hat{W}_c\left(t\right),\hat{W}_a\left(t\right),\hat{\theta}\left(t\right)\right)$$

If $\hat{\theta}(t)\to B_r(\theta)$ exponentially as $t\to\infty$,
$\hat{\delta}_t(t)\to B_{r1}(\delta_t(t))$ exponentially as $t\to\infty$.

Use $\hat{\delta}_t(t)$ and $\hat{\delta}_{ti}(t)$ for learning

**Update law**

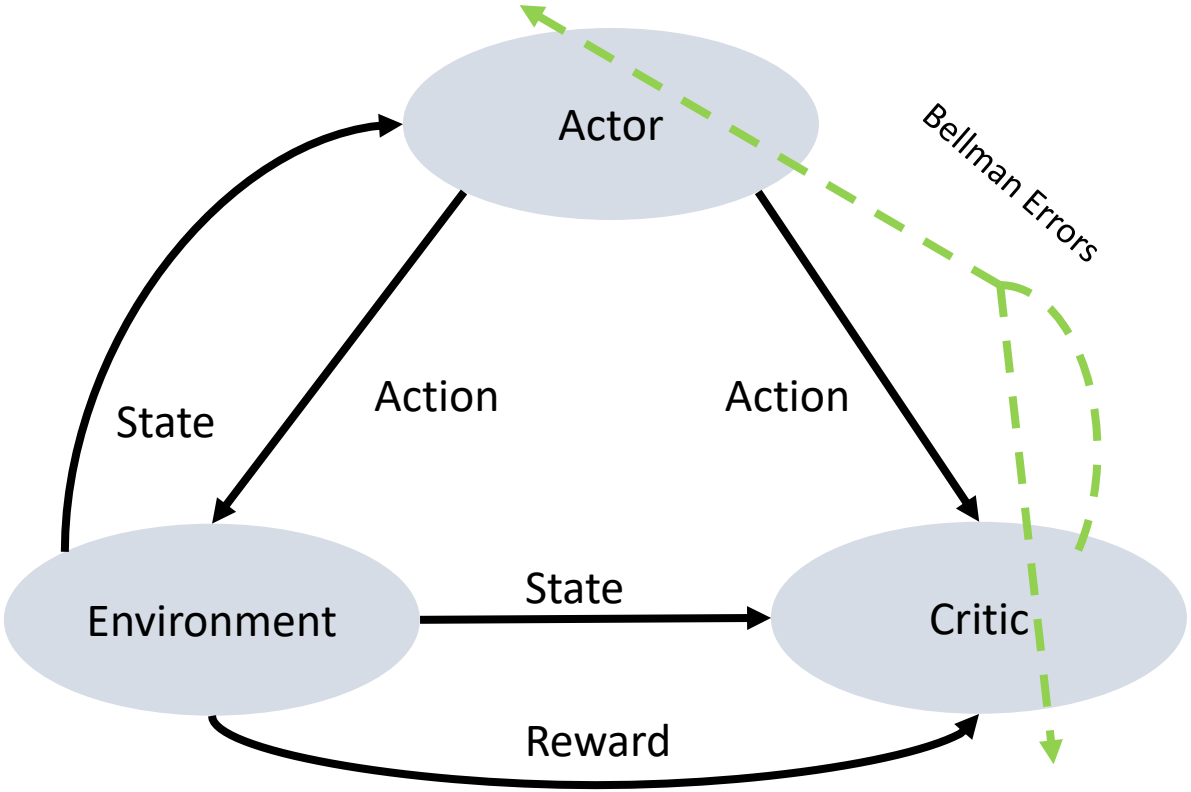
$$\dot{\hat{W}}_c=-\frac{\eta_{c1}\Gamma}{1+\nu\omega^T\Gamma\omega}\omega\hat{\delta}_t-\eta_{c2}\Gamma\sum_{i=1}^{N}\frac{1}{1+\nu\omega_i^T\Gamma\omega_i}\omega_i\hat{\delta}_{ti}$$

**Weight estimation error dynamics**

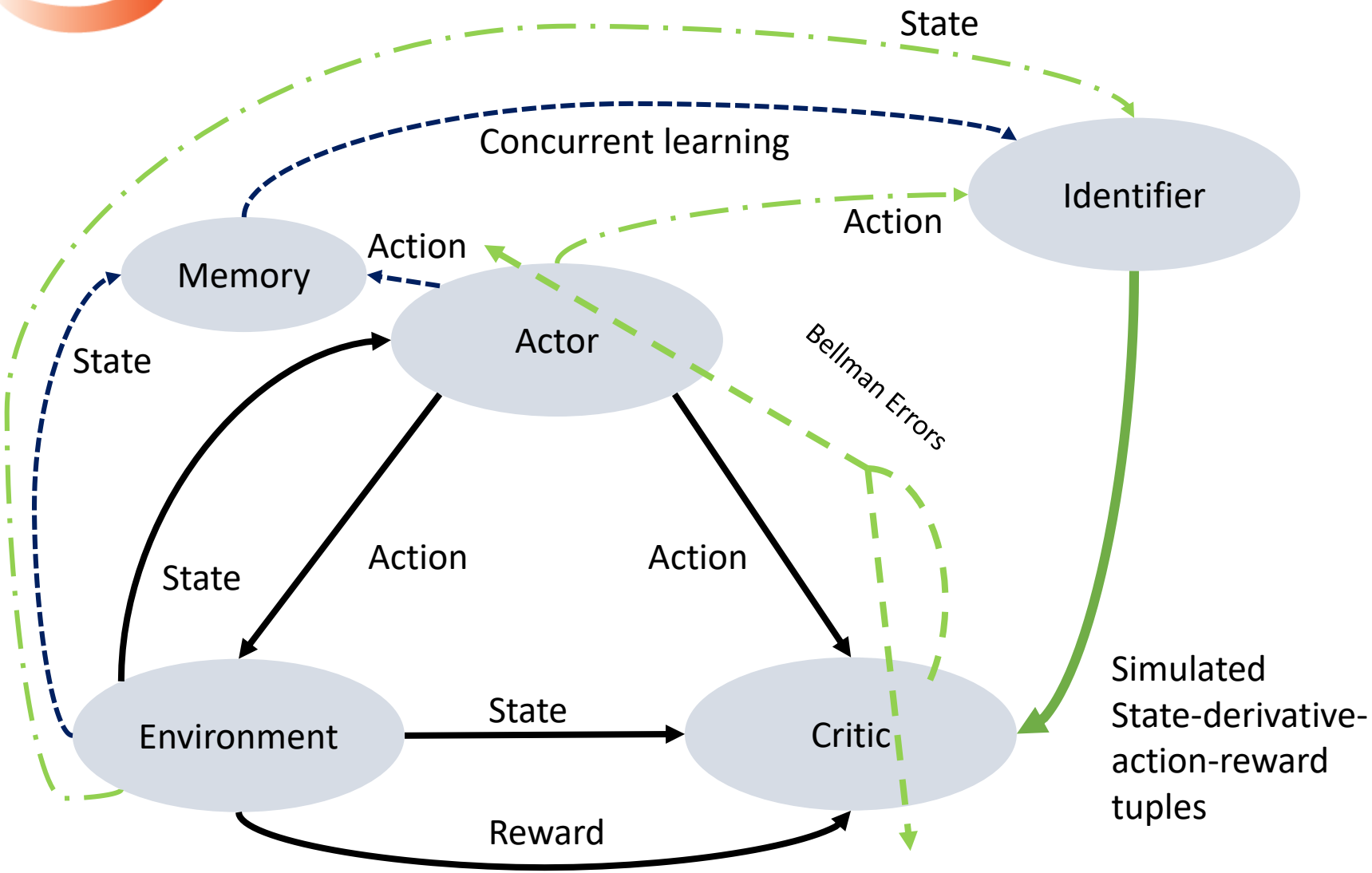
$$\dot{\tilde{W}}_c=-\Gamma\left(\eta_{c1}\frac{\omega\omega^T}{\rho}+\eta_{c2}\sum_{i=1}^{N}\frac{\omega_i\omega_i^T}{\rho_i}\right)\tilde{W}+\Delta$$

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

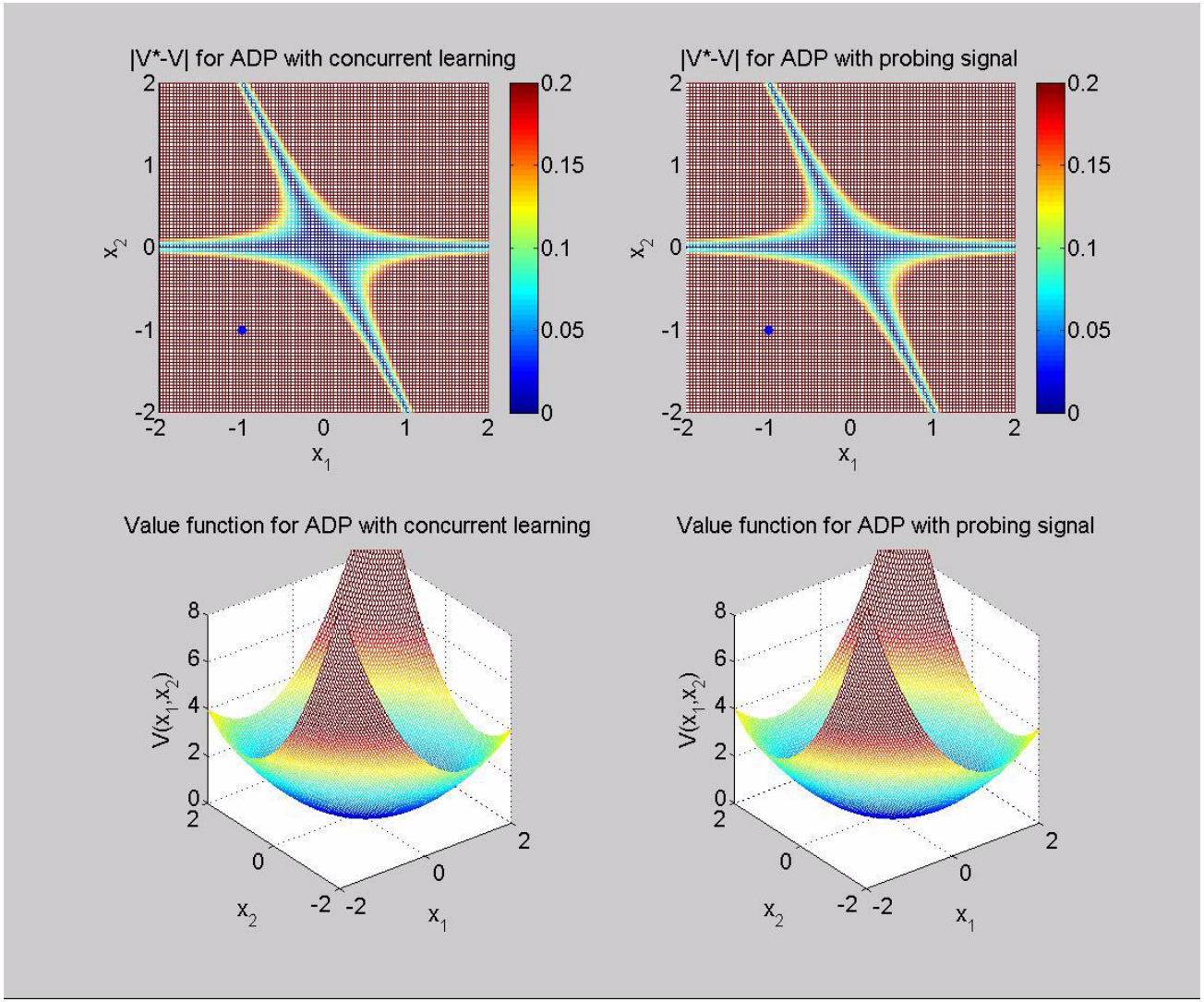
# Typical RL-based ADP Approach



# Simulated Experience



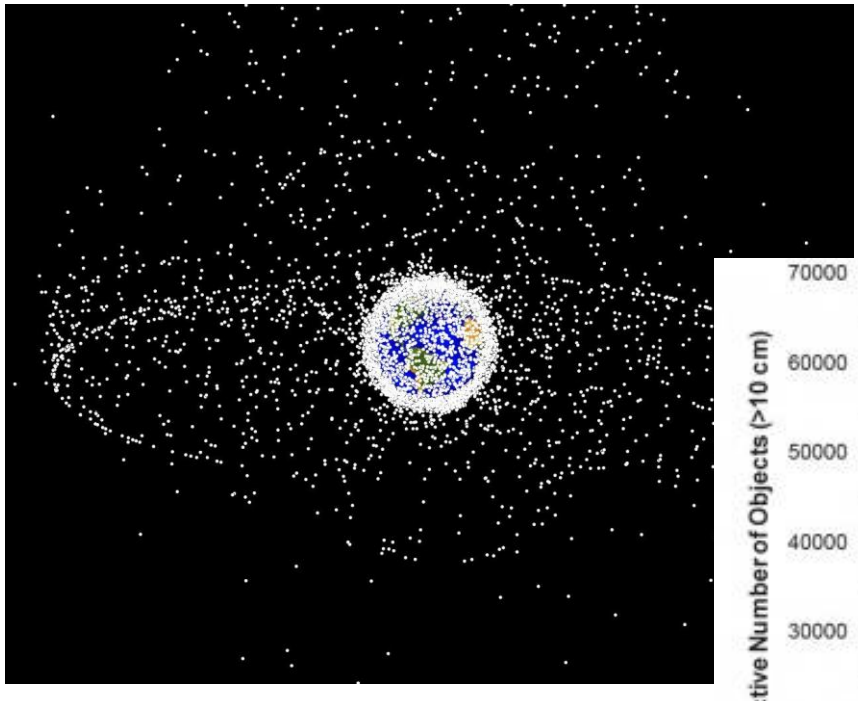
# Simulation: Known Optimal Solution



# Approximate Optimal Orbit Transfer of Non-cooperative Debris

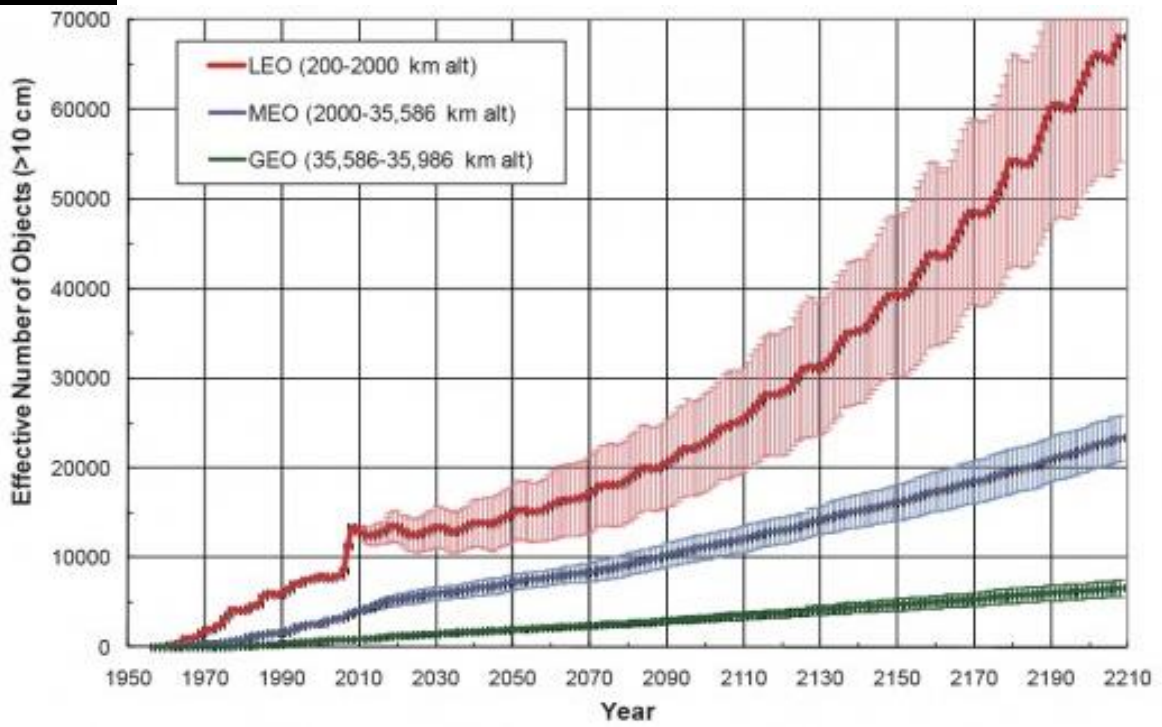
UF UNIVERSITY of FLORIDA
Duke UNIVERSITY
TEXAS The University of Texas at Austin
UC SANTA CRUZ

# Space Debris



[NASA]

## Approximate Optimal Orbit Transfer of Non-cooperative Debris



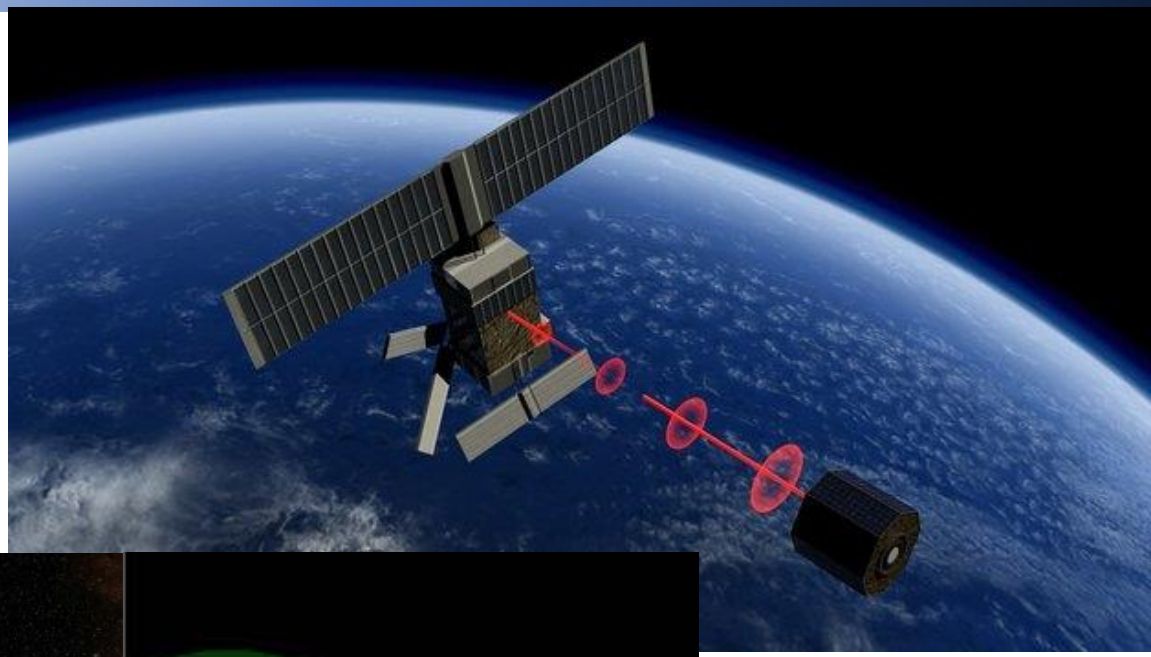

[NASA]

UF UNIVERSITY of FLORIDA  Duke UNIVERSITY  TEXAS The University of Texas at Austin  UC SANTA CRUZ

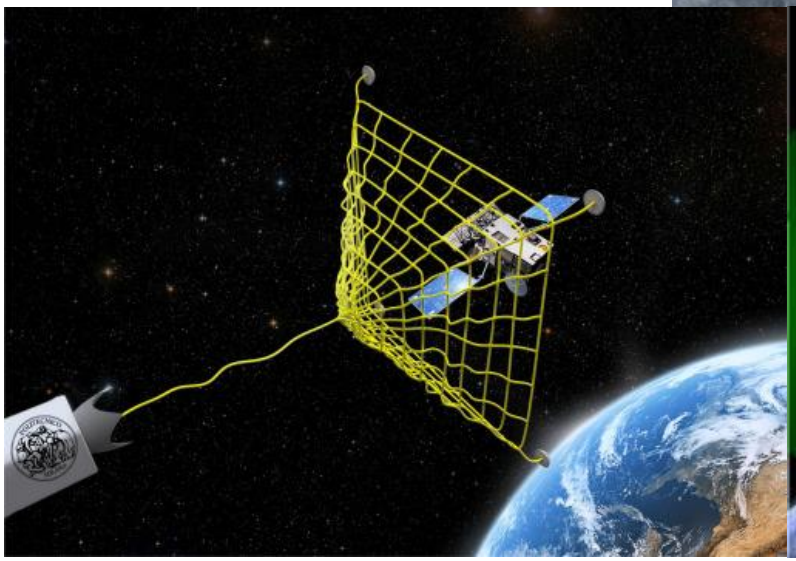
# Debris Removal

Removal Methods

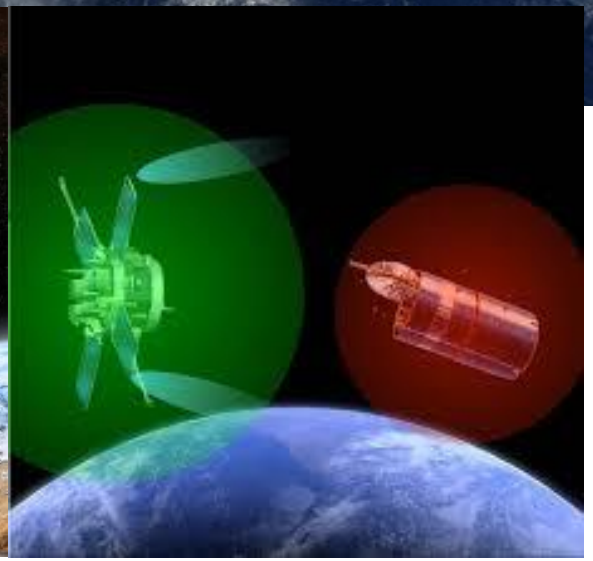
- Laser-Based
- Ion-Beam
- Tether-Based
- Sail-Based
- Satellite-Based
- Unconventional



[ESA]



[Benvenuto, Salvi, Lavagna]



[Hughes, Schaub]

# Problem Formulation

## Agent Dynamics

$$\dot{\eta}(t) = h\big(\eta(t)\big) + g(z(t), \eta(t), t)$$

$$\dot{z}(t) = d(t) \triangleq f(z(t), \eta(t), t)$$

## Error System

Servicing Satellite (SS), $\eta(t)$, intercept Disabled Satellite (DS), $z(t)$, at a distance, $r_d$

$$e_1 \triangleq z(t) - (\eta(t) - r_d)$$

Servicing Satellite (SS), $\eta(t)$, regulate the Disabled Satellite (DS), $z(t)$, to $z_g$

$$e_2 \triangleq (\eta(t) - r_d) - z_g - k_1\big(z(t) - z_g\big)$$

## Compact Dynamics

$$\dot{x}(t) = F\big(x(t)\big) + G\big(x(t)\big)u(t) + Kd(t)$$

# Problem Formulation

## Control Objective (2 player zero sum game)

Design a controller, $u$, which minimizes a cost function:

$$J(x,u,d,t_0) = \int_{t_0}^{\infty} \left(x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau) - \gamma^2 d^T(t) d(t)\right) d\tau$$

## Cost-to-Go

Optimal value function:

$$V^*\left(x(t)\right) = \min_{u(\tau)} \max_{d(\tau)} \int_{t}^{\infty} \left(x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau) - \gamma^2 d^T(\tau) d(\tau)\right) d\tau$$

## HJI Equation

Design a controller, $u$, which minimizes a cost function:

$$0 = x(t)^T Q x(t) + u(t)^T R u(t) - \gamma^2 d^T(t) d(t)$$
$$+\nabla V^*\left(x(t)\right)\left(F\left(x(t)\right) + G\left(x(t)\right) u^*\left(x(t)\right) + K d^*\left(x(t)\right)\right)$$

UF UNIVERSITY of FLORIDA  Duke UNIVERSITY  TEXAS The University of Texas at Austin  UC SANTA CRUZ

# Problem Formulation

## Universal Function Approximation Property

$$V^*(x) = W(x)^T \sigma\big(c(x)\big) + \epsilon_W(x)$$

$$u^*(x) = -\frac{R^{-1}G(x)^T}{2}\Big(\nabla\sigma\big(c(x)\big)^T W(x) + \epsilon_W(x)^T\Big)$$

$$d^*(x) = \frac{K^T}{2\gamma^2}\Big(\nabla\sigma\big(c(x)\big)^T W(x) + \epsilon_W(x)^T\Big)$$

## Stone-Weierstrauss Approximation Theorem

$$\hat{V}\big(x, \widehat{W}_c\big) = \widehat{W}_c{}^T \sigma\big(c(x)\big)$$

$$\hat{u}\big(x, \widehat{W}_a\big) = -\frac{R^{-1}G(x)^T}{2}\nabla\sigma\big(c(x)\big)^T \widehat{W}_a$$

$$\hat{d}\big(x, \widehat{W}_d\big) = \frac{K^T}{2\gamma^2}\nabla\sigma\big(c(x)\big)^T \widehat{W}_d$$

# Problem Formulation

## Bellman Error

$$\delta_t\left(x,\widehat{W}_c,\widehat{W}_a,\widehat{W}_d\right)=x(t)^TQx(t)+\hat{u}\left(x,\widehat{W}_a\right)^TR\hat{u}\left(x,\widehat{W}_a\right)-\gamma^2\hat{d}\left(x,\widehat{W}_d\right)^T\hat{d}\left(x,\widehat{W}_d\right)$$
$$+\nabla\hat{V}\left(x,\widehat{W}_c\right)\left(F(x)+G(x)\hat{u}\left(x,\widehat{W}_a\right)+K\hat{d}\left(x,\widehat{W}_d\right)\right)$$

## Update Laws

$$\dot{\widehat{W}}_c(t)=-\Gamma_c(t)\frac{k_{c1}}{N+1}\frac{\omega(t)}{\rho(t)^2}\delta_t(t)-\Gamma_c(t)\frac{k_{c2}}{N+1}\sum_{i=1}^{N}\frac{\omega_i(t)}{\rho_i(t)^2}\delta_{ti}(t)$$

$$\dot{\widehat{W}}_a(t)=\text{proj}\left(-K_ak_{a1}\left(\widehat{W}_a(t)-\widehat{W}_c(t)\right)\right)$$

$$\dot{\widehat{W}}_d(t)=\text{proj}\left(-K_dk_{d1}\left(\widehat{W}_d(t)-\widehat{W}_c(t)\right)\right)$$

$$\dot{\Gamma}_c(t)=\beta_c\Gamma_c(t)-\Gamma_c(t)\frac{k_{c1}}{N+1}\frac{\omega(t)\omega^T(t)}{\rho(t)^2}\Gamma_c(t)-\Gamma_c(t)\frac{k_{c2}}{N+1}\sum_{i=1}^{N}\frac{\omega_i(t)\omega_i^T(t)}{\rho_i(t)^2}\Gamma_c(t)$$

# Stability Analysis

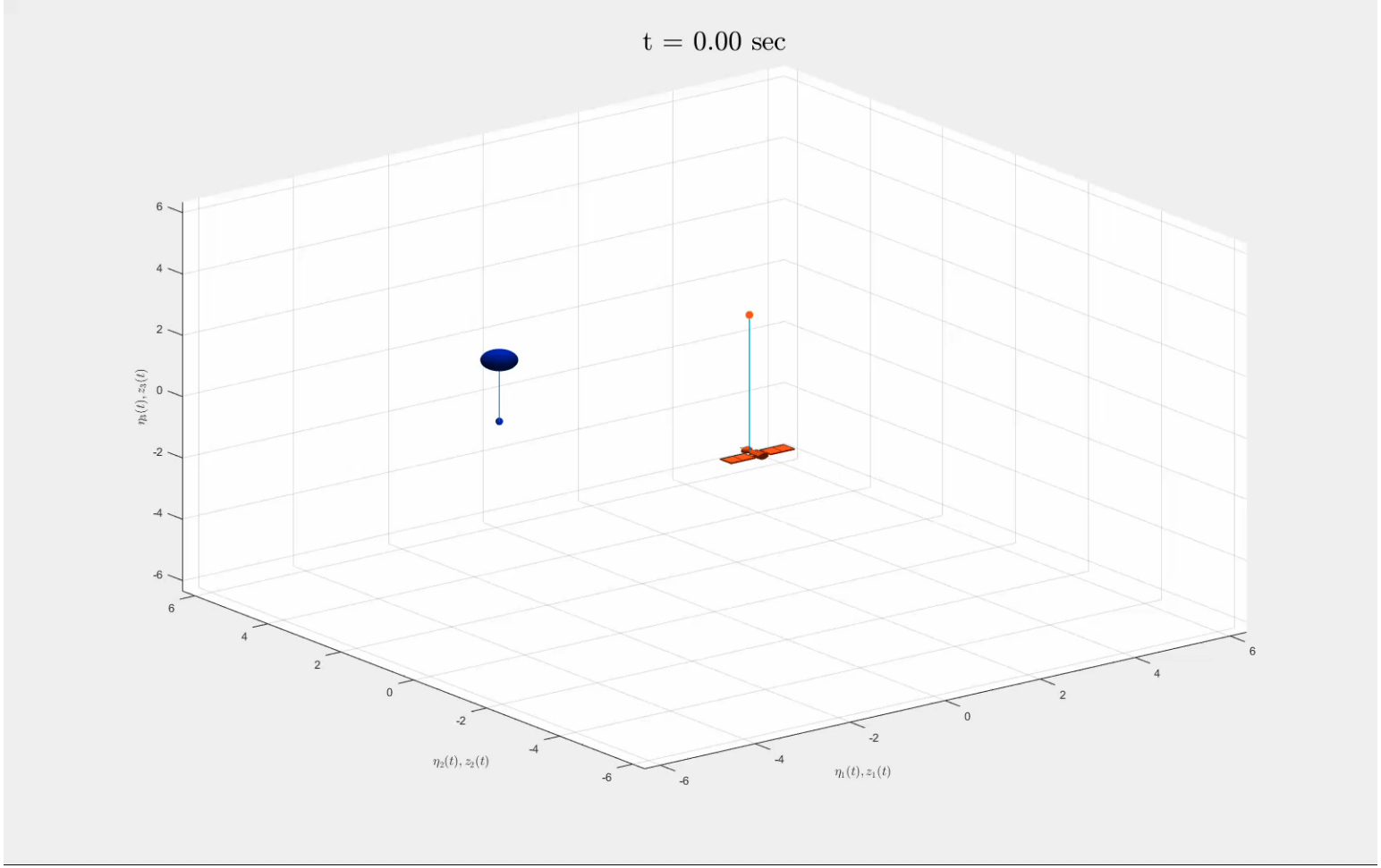
## Lyapunov Stability Analysis

$$V_L(Z_L,t) = V^*(x) + \frac{1}{2}\widetilde{W}_c{}^T\Gamma_c^{-1}(t)\widetilde{W}_c + \frac{1}{2}\widetilde{W}_a{}^T K_a^{-1}\widetilde{W}_a + \frac{1}{2}\widetilde{W}_a{}^T K_d^{-1}\widetilde{W}_d$$

$$\begin{aligned}\dot{V}_L(Z_L,t) \quad &= \nabla V^*\left(x,\widehat{W}_c\right)\left(F + G\hat{u} + K\hat{d}\right) + \widetilde{W}_c{}^T\Gamma_c^{-1}\dot{\widetilde{W}}_c + \widetilde{W}_a{}^T K_a^{-1}\dot{\widetilde{W}}_a \\ &\quad + \widetilde{W}_d{}^T K_d^{-1}\dot{\widetilde{W}}_d - \frac{1}{2}\widetilde{W}_c{}^T\left(\Gamma_c^{-1}\dot{\Gamma}_c\Gamma_c^{-1}\right)\end{aligned}$$

- System state $(x)$, weight estimation errors $\left(\widetilde{W}_c, \widetilde{W}_a, \widetilde{W}_d\right)$, and control policy $u(t)$ are **<u>Uniformly Ultimately Bounded</u>**
- By using a State Following (StaF) neural network architecture a local result is achieved

# Simulation Results

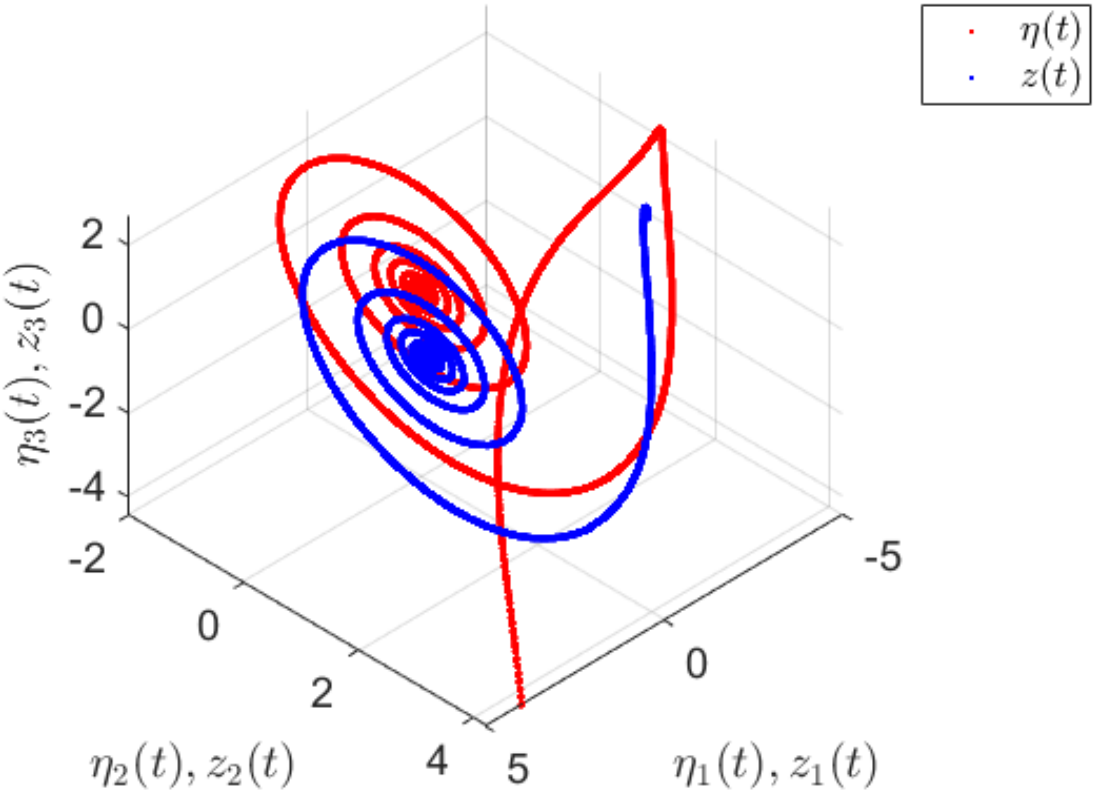



UF UNIVERSITY of FLORIDA
Duke UNIVERSITY
TEXAS The University of Texas at Austin
UC SANTA CRUZ

# Simulation Results

## Future Work

- Consider Orbital Mechanics
- Higher Fidelity Model of Electrostatic Dynamics



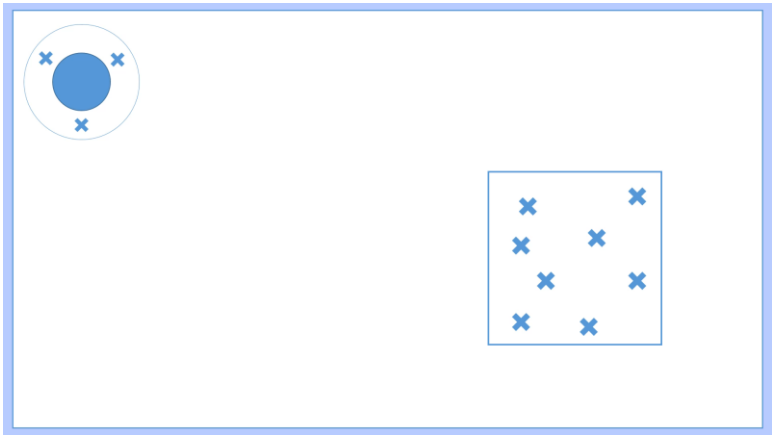
# Reinforcement Learning With Sparse Bellman Error Extrapolation For Infinite-horizon Approximate Optimal Regulation

UF UNIVERSITY of FLORIDA
Duke UNIVERSITY
TEXAS The University of Texas at Austin
UC SANTA CRUZ

# StaF and MBRL

**Needs?**

- Computational Efficiency
- Fewer active NN basis functions
- BE extrapolation in operational subset
- Global Approximation



**StaF and MBRL Approximation**

**State Following (StaF) Approximation**

- Local Value Function Approximation
- Explores Around the Current State
- Reduced Number of Basis Functions
- Trades Global Optimality for Computational Efficiency

- Combination of Techniques
- StaF Outside of Boxed Region
- MBRL Inside of Boxed Region

**Sparse NN**

- Decrease active neurons
- Switch between smaller MBRL segments
- StaF/MBRL Method

UF UNIVERSITY of FLORIDA  Duke UNIVERSITY  TEXAS The University of Texas at Austin  UC SANTA CRUZ

# BE Extrapolation

## Instantaneous BE: Residual from HJB

$$\hat{\delta}_i(x,t) \triangleq \hat{\delta}\left(x_i, \hat{W}_c(t), \hat{W}_a(t)\right)$$

## Weight Update Laws using MBRL

$$\dot{\hat{W}}_c(t) = -\eta_{c1}\Gamma\frac{\omega(t)}{\rho(t)}\hat{\delta} + \eta_{c2}\frac{1}{N_j}\sum_{i=1}^{N_j}\frac{\omega_i(t)}{\rho_i(t)}\hat{\delta}_i(t)$$

Actual system

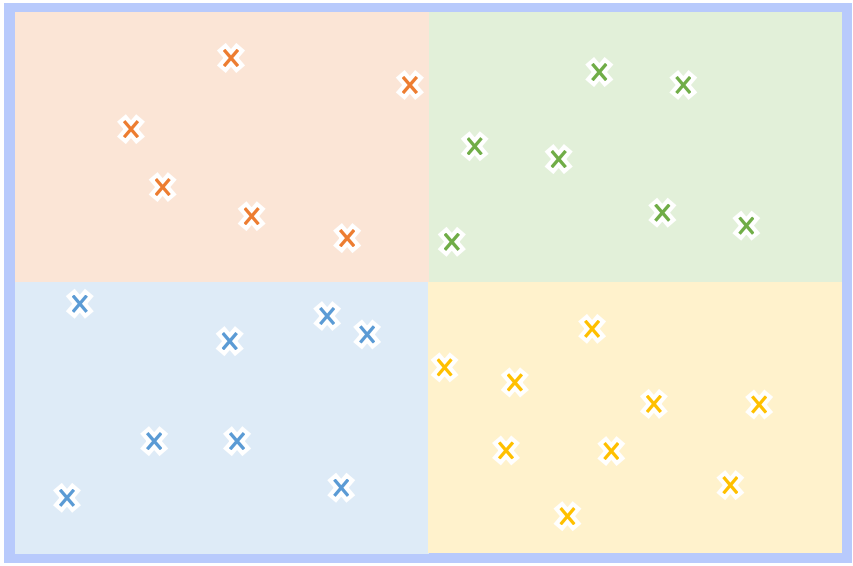
Extrapolated system

$$\dot{\Gamma}(t) = \left(\lambda\Gamma(t) - \frac{\eta_{c1}\Gamma(t)\omega(t)\omega(t)^T\Gamma(t)}{\rho(t)} - \Gamma(t)\eta_{c2}\left(\frac{1}{N_j}\sum_{i=1}^{N_j}\frac{\omega_i(t)\omega_i^T(t)}{\rho_i(t)}\hat{\delta}_i(t)\right)\Gamma(t)\right)\mathbf{1}_{\{\underline{\Gamma}\le\|\Gamma\|\le\bar{\Gamma}\}}$$

$$\dot{\hat{W}}_a(t) = -\eta_{c1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right) - \eta_{a2}\hat{W}_a(t) + \frac{\eta_{c1}G_\sigma^T(t)\hat{W}_a(t)\omega(t)^T}{4\rho(t)}\hat{W}_c(t) + \left(\frac{\eta_{c2}}{4N_j}\sum_{i=1}^{N_j}\frac{G_{i\sigma}^T\hat{W}_a(t)\omega_i(t)}{\rho_i(t)}\hat{\delta}_i(t)\right)\hat{W}_c(t)$$

# Segmentation

- Separate Operating Domain
- BE Extrapolation Contained to Segment
- Smaller History Stack (one for each segment)
- Switches Depending on Region
- Less Active History
- A further step towards including memory (cognition)
- Lyapunov analysis shows SG-UUB



**Update Laws**

$$\dot{\hat{W}}_c(t) = -\eta_{c1}\Gamma\frac{\omega(t)}{\rho(t)}\hat{\delta}(t) - \eta_{c2}\sum_c^j(t)$$

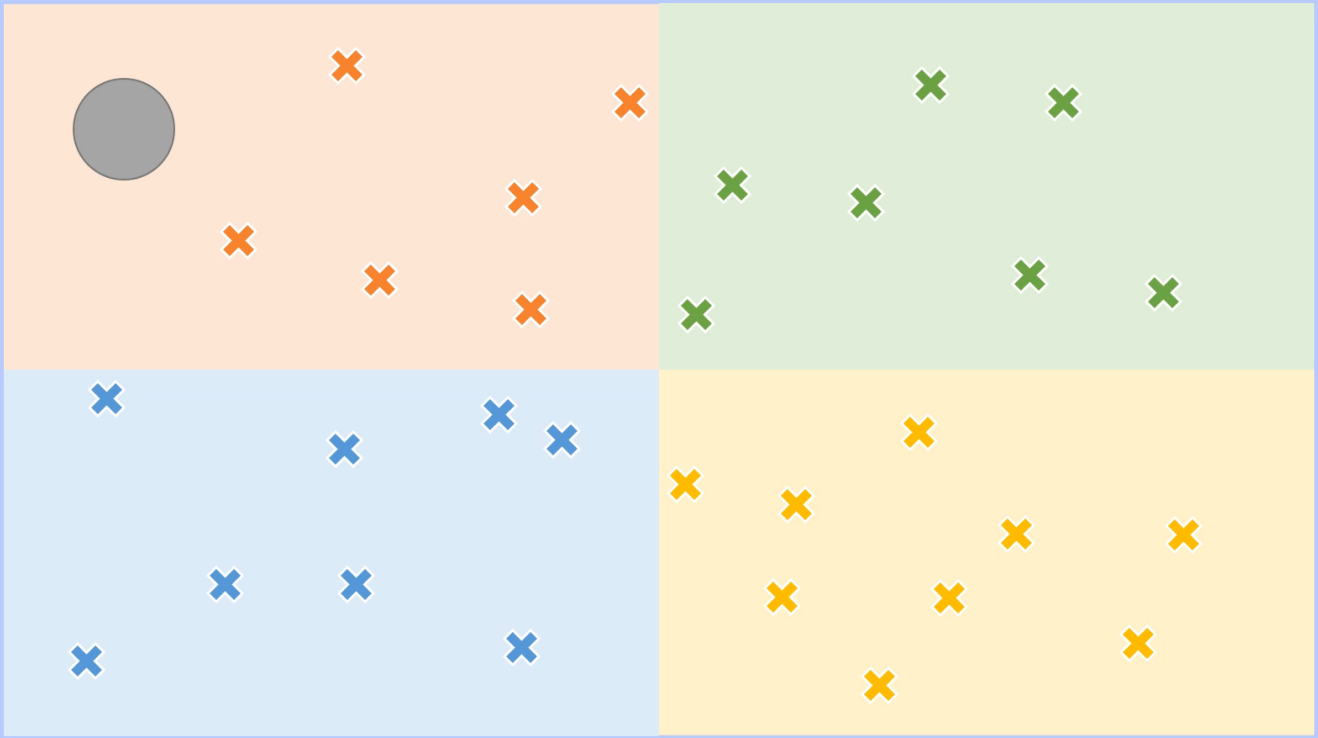
$$\dot{\Gamma}(t) = \left(\lambda\Gamma(t) - \eta_{c1}\frac{\Gamma(t)\omega(t)\omega(t)^T\Gamma(t)}{\rho(t)} - \Gamma(t)\eta_{c2}\left(\sum_\Gamma^j(t)\right)\Gamma(t)\right)\mathbf{1}_{\{\underline{\Gamma}\le\|\Gamma\|\le\bar{\Gamma}\}}$$

Switching History Stack Term

$$\dot{\hat{W}}_a(t) = -\eta_{a1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right) - \eta_{a2}\hat{W}_a(t) + \frac{\eta_{c1}G_\sigma(t)^T\hat{W}_a(t)\omega(t)^T}{4\rho(t)}\hat{W}_c(t) + \left(\eta_{c2}\sum_a^j(t)\right)\hat{W}_c(t)$$

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

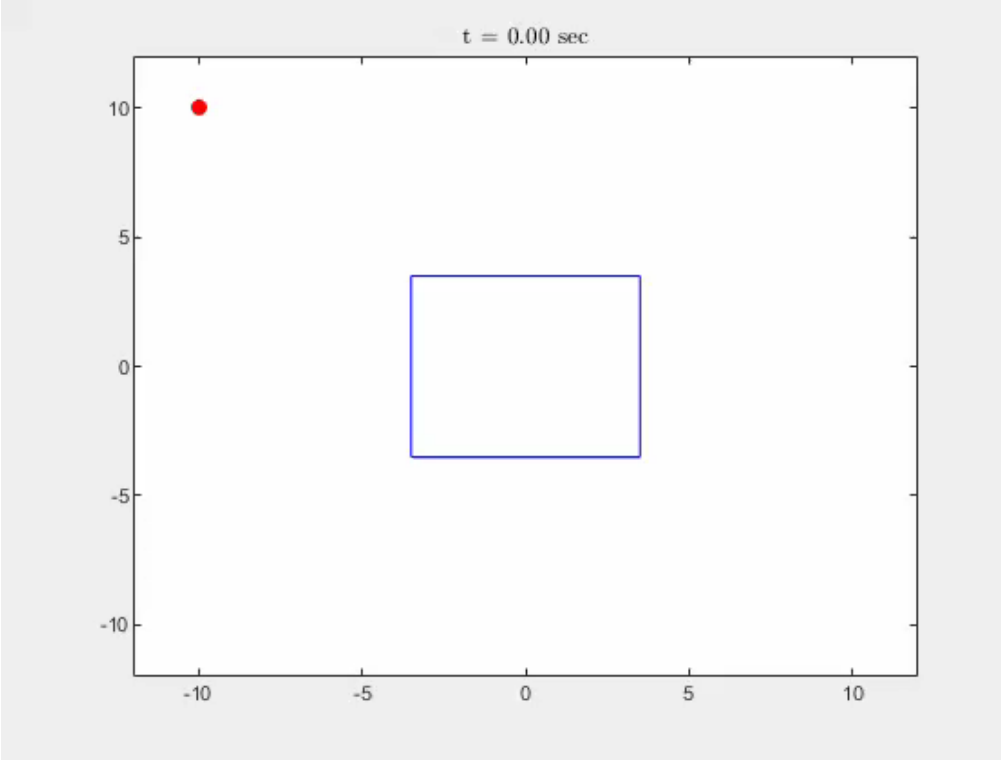
# Switching Segments



$$\dot{\hat{W}}_c(t) = -\eta_{c1}\Gamma\frac{\omega(t)}{\rho(t)}\hat{\delta} + \eta_{c2}\sum_c^j(t)$$

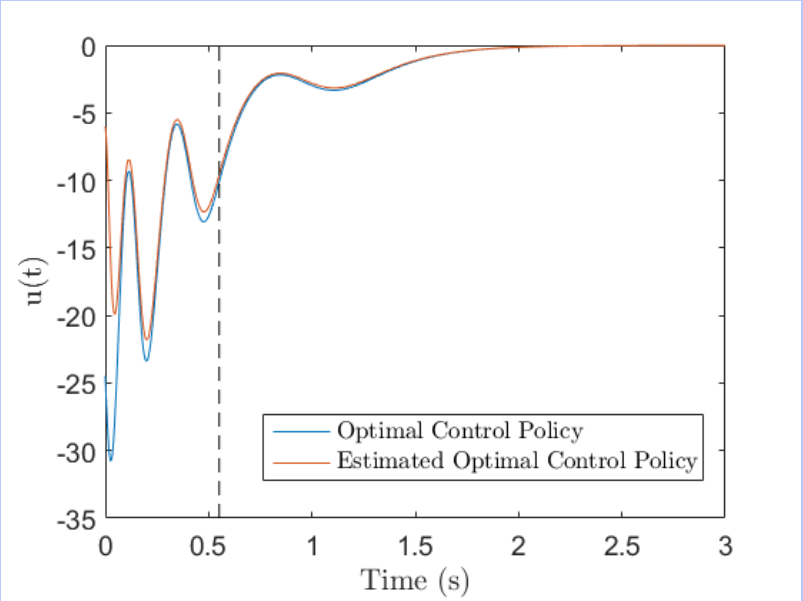
# Simulation Results

- Dynamics
  - $\dot{x} = f(x) + g(x)u(t)$
  - $f(x) = \begin{bmatrix} -x_1 + x_2 \\ -\frac{1}{2}x_1 - \frac{1}{2}x_2\left(1 - (\cos(2x_1) + 2)^2\right) \end{bmatrix}$
  - $g(x) = \begin{bmatrix} 0 \\ \cos(2x_1) + 2 \end{bmatrix}$
- Initial Conditions
  - $x(0) = [-10, 10]^T$
  - $\widehat{W}_a = \widehat{W}_c = \left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right]^T$
- Define Segmentation
  - $\Omega_1 \triangleq \{x \in \mathbb{R}^2 : 3.5 \leq |x_1|, |x_2| \leq 10\}$
    - Outer Segment
  - $\Omega_2 \triangleq \{x \in \mathbb{R}^2 : |x_1|, |x_2| < 3.5\}$
    - Inner Segment
- Define BE Bases
  - For $x_i \in \Omega_1$ (Outer Segment)
    - $\sigma_i(x_i) = \begin{bmatrix} x_{1,i}^2 & x_{1,i}x_{2,i} & x_{2,i}^2 \end{bmatrix}^T$
  - For $x_i \in \Omega_2$ (Inner Segment)
    - $\sigma_i(x_i) = \begin{bmatrix} x_{1,i}^2 & 0 & x_{2,i}^2 \end{bmatrix}^T$

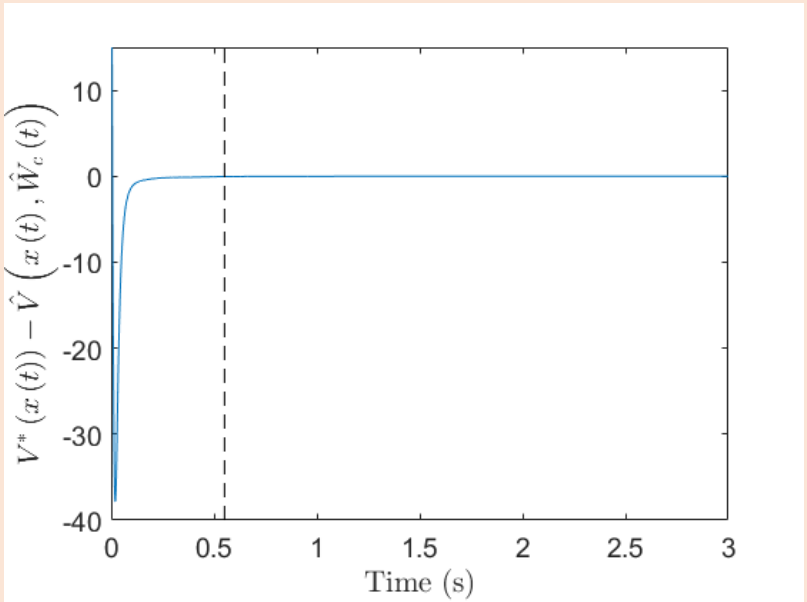


# Simulation Results

## Control Policy




## Value Function



# Future Work

## Benefits of Sparsity and Segmentation

- Decrease Active Memory Used
- Decrease Number of Computations in NN
- Establish Sparse, Switching Framework

## Future Work (on-going collaboration with Scott Nivison (RW))

- Quantify Sparsity Benefit
  - Number of Computations
  - Compare to Existing ADP Methods
  - Computation Time
- Utilize Barrier Functions
  - Guarantee System Safety
- Application to Multi-Agent Systems