Adaptation, Optimality, and Synthesis

UF UNIVERSITY of FLORIDA

Duke UNIVERSITY

TEXAS The University of Texas at Austin

UC SANTA CRUZ

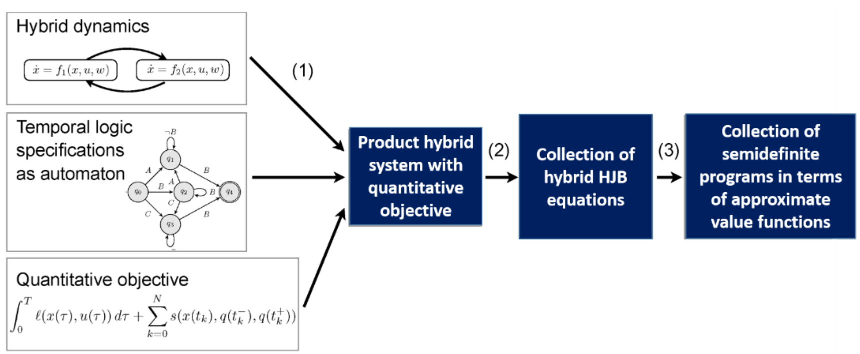
Adaptation, Optimality, and Synthesis

- Approximately optimal control methods for forward and inverse decision-making problems

- Real-time optimal control methods that can handle uncertainty, complex mission specifications, and rely on sophisticated approximation, learning, and sampling techniques to enhance scalability (avoid explicit discretization of continuous dynamics)

RT2 will establish new strategies for the development of approximately optimal control methods for continuous and stochastic hybrid systems for forward and inverse decision-making problems under complex mission specifications

- Tractable optimal control methods under complex mission specifications captured by temporal logic (TL) formulas, and extend them to systems with unknown uncertainties and run-time computational limitations



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

Adaptation, Optimality, and Synthesis

- Reinforcement learning (RL) for approximate dynamic programming (ADP)
  - Uncertainty in the dynamics
  - Unknown cost-to-go
- Exploration while Exploitation
  - Simulation of experience
  - Bellman Error extrapolation
- Computational Considerations
  - State following (StaF) methods
  - Sparsity

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

- ADP differential games
- Game structures

Adaptation, Optimality, and Synthesis

- Temporal-Logic (TL)-Constrained Synthesis and Verification without Discretization
- Controls with formal methods
  - Hierarchal structure
- Specifying behavior with TL
- Automaton representation for TL
- ADP approach

- Integration of high level tasks (via temporal logic) with motion planning
- Synthesizing motion plans
- Automatons to graphs

Reinforcement Learning Based ADP:

Computational reductions, faster learning, and more complex problems

Optimal Control Problem

Problem

Design a controller u that minimizes the cost

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

$$r\left(x^o,u^o\right)=Q(x^o)+u^{oT}Ru^o$$

Subject to dynamic constraints

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

Exact Solution

Optimal value function

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

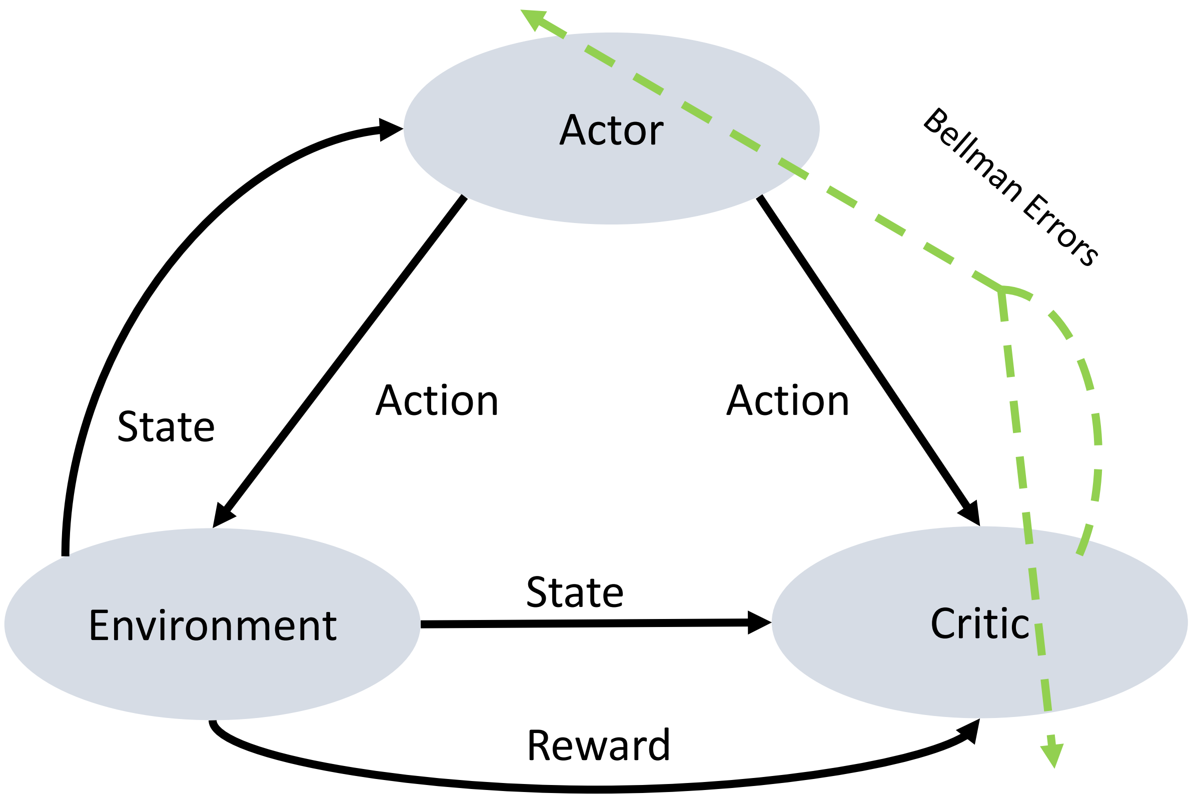
HJB equation

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

Optimal policy

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

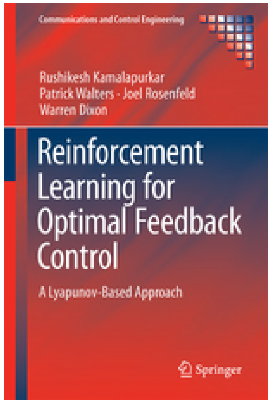
Typical RL-based ADP Approach



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



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

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

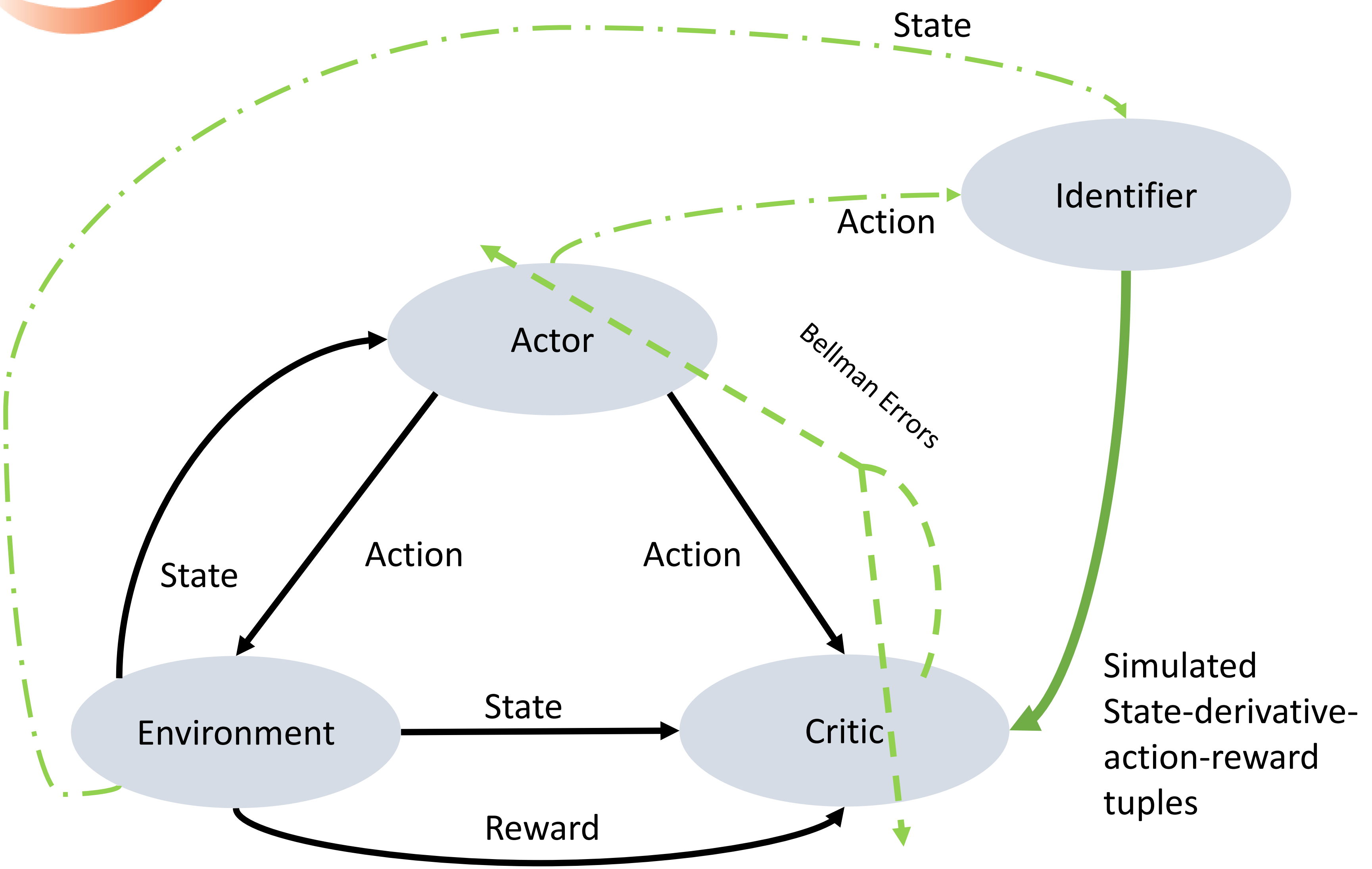
Update law

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

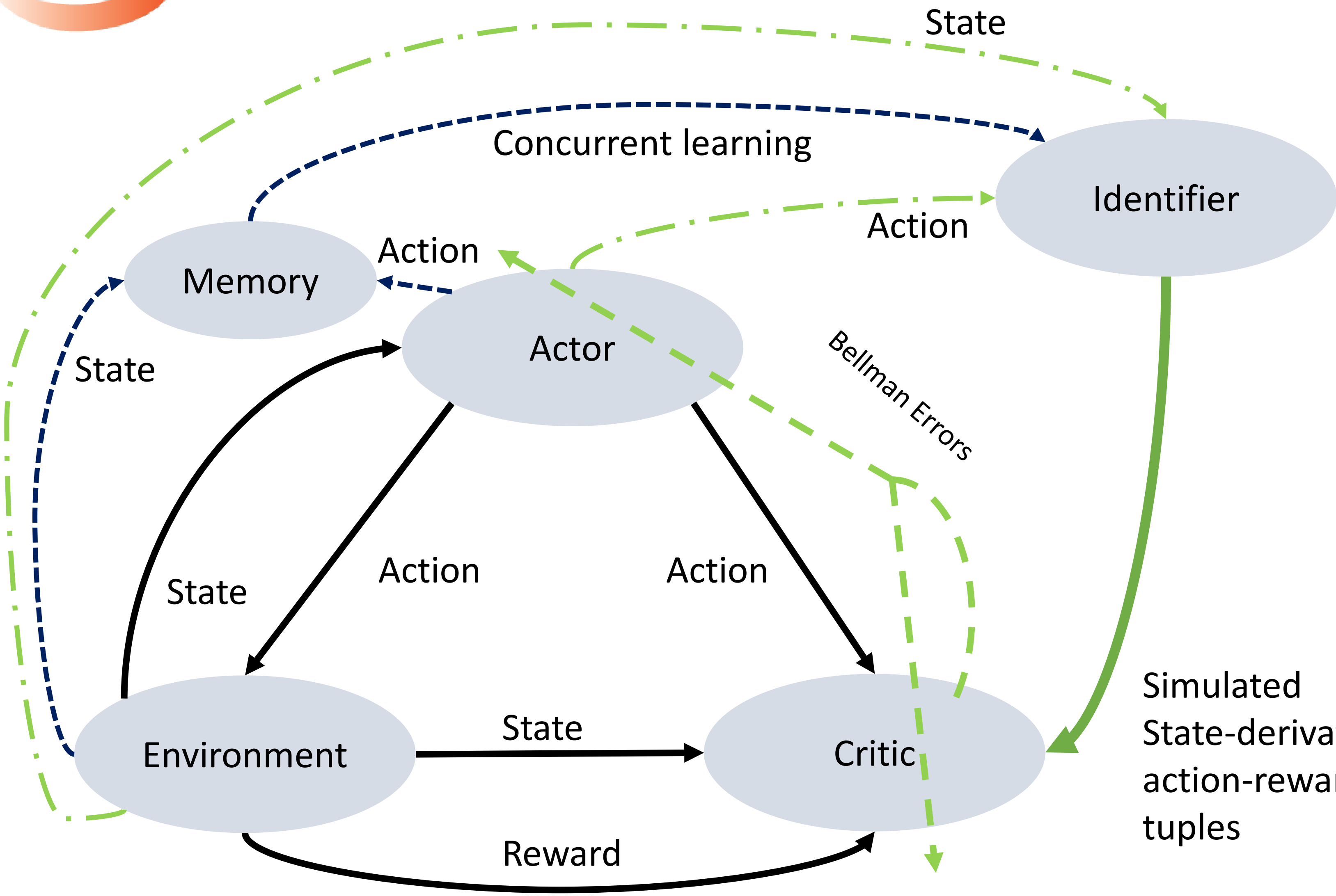
Weight estimation error dynamics

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

Simulated Experience



Simulated Experience



Simulated State-derivative-action-reward tuples

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

Simulation: Known Optimal Solution

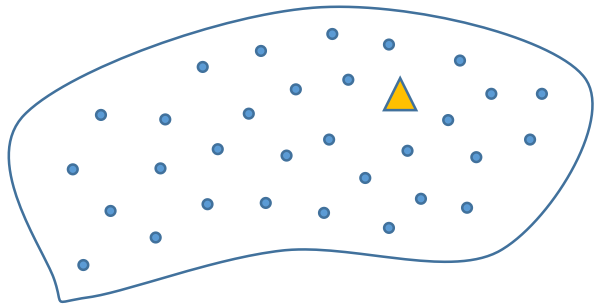
Computational Cost

Value function and policy approximation

Traditional Model Based RL (SGMBRL)approximation:

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

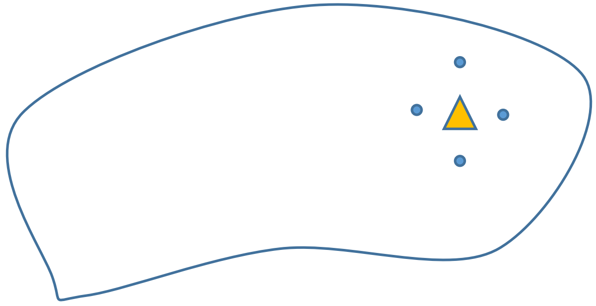
$$\hat{u}(x^o, \hat{W}_a) = \frac{1}{2} R^{-1} g^T(x^o) \nabla \sigma^T(x^o) \hat{W}_a$$



State Following (StaF) kernel function approximation:

$$\hat{V}(x^o, y^o, \hat{W}_c) = \hat{W}_c^T \phi(x^o, c(y^o))$$

$$\hat{u}(x^o, y^o, \hat{W}_a) = \frac{1}{2} R^{-1} g^T(x^o) \nabla \sigma^T(x^o, c(y^o)) \hat{W}_a$$



Evaluated at x^o using StaF kernels centered at $y^o \in \overline{B_r(x^o)}$

$\overline{B_r(x^o)}$: a compact set around the state x^o

$\hat{W}_a$: Actor weight apprximation

$\hat{W}_c$: Critic weight apprximation

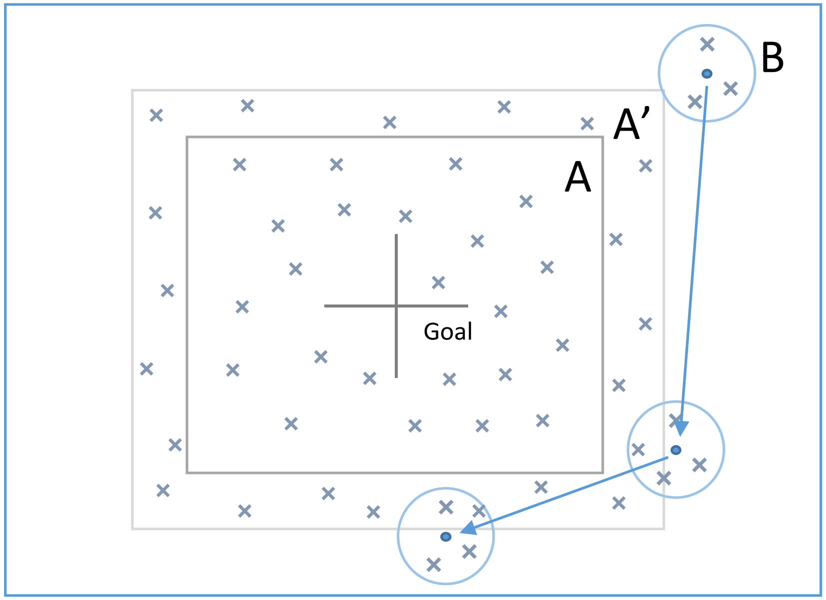
$\sigma(x^o)$: SGMBRL basis functions

$\phi(x^o, c(y^o))$: StaF basis functions

Combined Approach

Optimal regulation problem is to drive the state to the origin

- Divide the global space into regions
- Concurrently approximate the value function
  - Near state using StaF
  - Near the origin using SGMBRL
- Also include a transition region to marry the approximation regions



Value function and control policy approximations:

$$\hat{V}(x^o, y^o, \hat{W}_{1c}, \hat{W}_{2c}) = \lambda(x^o)\hat{W}_{1c}^T\sigma(x^o) + (1-\lambda(x^o))\hat{W}_{2c}^T\phi(x^o, c(y^o))$$

$$\hat{u}(x^o, y^o, \hat{W}_{1a}, \hat{W}_{2a}) = -\frac{1}{2}R^{-1}g^T(x^o)\left(\nabla\hat{V}(x^o, y^o, \hat{W}_{1c}, \hat{W}_{2c})\right)^T$$

StaF

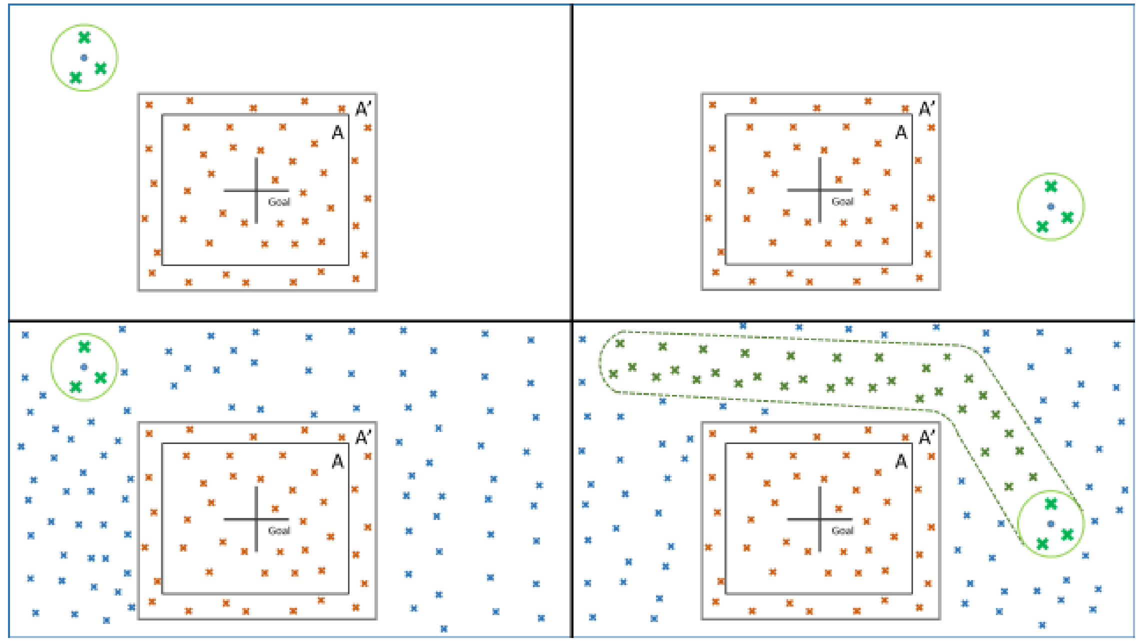
SGMBRL

Transition Function

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

Learning with Sparse Kernels

- Domain is partitioned into segments
- Each segment has a history stack
- Regional data switching
- Characterizes regions with varying dynamics or uncertainties
- A further step towards including memory (cognition)



On-going work with Scott Nivison, RW

Update Laws

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

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

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

Switching History Stack Term

Example:
Optimal Obstacle Avoid

Obstacle Avoidance

Consider an autonomous agent and dynamic avoidance regions

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

$$\dot{z}_i(t) = h_i(z_i(t))$$
Avoidance Region

HJB requires knowledge of the avoidance region dynamics for all time (i.e., for the entire operating domain)

Alleviate the need for knowledge of the avoidance region dynamics outside of the detection region

$$\dot{z}_i(t) = \mathscr{F}_i(x(t), z_i(t))\,h_i(z_i(t))$$

$$\mathscr{F}_i(x, z_i) = 0 \text{ for } \|x - z_i\| > r_d$$
$$\mathscr{F}_i(x, z_i) = 1 \text{ for } \|x - z_i\| \leq \bar{r}$$

Combined to form the following vehicle-avoidance-region system

$$\dot{\zeta}(t) = F(\zeta(t)) + G(\zeta(t))\,u,$$

$$F = \begin{bmatrix} f(x) \\ \mathscr{F}_1(x, z_1)\,h_1(z_1) \\ \vdots \\ \mathscr{F}_M(x, z_M)\,h_M(z_M) \end{bmatrix}, \quad G = \begin{bmatrix} g(x) \\ \mathbf{0}_{Mn \times m} \end{bmatrix}$$

Problem Formulation

Control Objective

Design a controller u which minimizes

$$J(\zeta, u) \triangleq \int_{t_0}^{\infty} r(\zeta(\tau), u(\tau)) d\tau$$

$$r(\zeta, u) \triangleq Q_x(x) + \sum_{i=1}^{M} s_i(x, z_i) Q_z(z_i) + \Psi(u) + P(\zeta)$$

Constraints

Dynamics

$$\dot{\zeta}(t) = F(\zeta(t)) + G(\zeta(t)) u(t)$$

Input saturations

$$\Psi(u) \triangleq 2 \sum_{i=1}^{m} \left[\int_0^{u_i} \left(\mu_{sat} r_i \tanh^{-1}\left(\frac{\xi_{u_i}}{\mu_{sat}} \right) \right) d\xi_{u_i} \right]$$

$$\sup_t (u_i) \leq \mu_{sat} \; \forall i = 1, \ldots, m$$

Exact Solution

Optimal value function

$$V^*(\zeta) = \min_{u(\tau) \in U | \tau \in \mathbb{R}_{\geq t}} \int_t^{\infty} r(\zeta(\tau), u(\tau)) d\tau$$

Hamilton Jacobi Bellman equation

$$0 = \frac{\partial V^*(\zeta)}{\partial x} (f(x) + g(x) u^*(\zeta)) + r(\zeta, u^*(\zeta)) + \sum_{i=1}^{M} \frac{\partial V^*(\zeta)}{\partial z_i} (\mathscr{F}_i(x, z_i) h_i(z_i))$$

Optimal control policy

$$u^*(\zeta) = -\mu_{sat} \operatorname{Tanh}\left(\frac{R^{-1} G(\zeta)^T}{2\mu_{sat}} (\nabla V^*(\zeta))^T \right)$$

Prevents collision with avoidance regions

$$P(\zeta) \triangleq \sum_{i=1}^{M} \left(\min\left\{ 0, \frac{\|x - z_i\|^2 - r_d^2}{(\|x - z_i\|^2 - r_a^2)^2} \right\} \right)^2$$

Approximate Solution

StaF kernel Optimal Value function representation:

$$V^*(y) = P_a(y) + V^{\#}(y)$$

$$V^{\#}(y) = W(\zeta)^T \sigma(y, c(\zeta)) + \epsilon(\zeta, y)$$

StaF kernels centered at $y \in \overline{B_r(\zeta)}$, evaluated at ζ

P_a: Bounded avoidance function

W: Ideal StaF weight

ϵ: Function reconstruction error

$\widehat{W}_c$: Critic weight estimate

$\widehat{W}_a$: Actor weight estimate

Concatenated vector of StaF basis functions

$$\sigma(\zeta, c(\zeta)) = \begin{bmatrix} \sigma_0(x, c(x)) \\ s_1(x, z_1)\sigma_1(z_1, c(z_1)) \\ \vdots \\ s_M(x, z_M)\sigma_M(z_M, c(z_M)) \end{bmatrix}$$

Only active in detection region

Value function approximation

$$\hat{V}\left(y, \zeta, \hat{W}_c\right) \triangleq P_a(y) + \hat{W}_c^T \sigma(y, c(\zeta))$$

$$\hat{\hat{D}}\left(y, \zeta, \hat{W}_a\right) \triangleq G^T(y) \nabla \sigma^T(y, c(\zeta)) \hat{W}_a + G^T(y) \nabla P_a^T(y)$$

Control policy approximation

$$\hat{u}\left(y, \zeta, \hat{W}_a\right) \triangleq -\mu_{sat} \mathrm{Tanh}\left(\frac{R^{-1}}{2\mu_{sat}} \hat{\hat{D}}\left(y, \zeta, \hat{W}_a\right)\right)$$

BE Implementation

Instantaneous Bellman error

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

Exploitation

Simulation of experience via Bellman error extrapolation

Select off-policy trajectories $\{x_k : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n\}_{i=1}^{N}$ such that each x_k maps the current state to a trajectory $x_k(x(t), t) \in \overline{B_r(x(t))}$, where $\zeta_k = [x_k^T, Z^T]^T$, and $Z = [z_1^T, \ldots, z_M^T]^T$. Then, evaluate δ at $y = x_k$.

$$\delta_k(t) = \delta\left(x_k(x(t), t), x(t), \hat{W}_c(t), \hat{W}_a(t), \hat{\theta}(t)\right)$$

Exploration

Value function update laws

$$\dot{\hat{W}}_c(t) = -\Gamma_c(t)\left(k_{c1}\frac{\omega(t)}{\rho^2(t)}\delta_t(t) + \frac{k_{c2}}{N}\sum_{k=1}^{N}\frac{\omega_k(t)}{\rho_k^2(t)}\delta_k(t)\right)$$

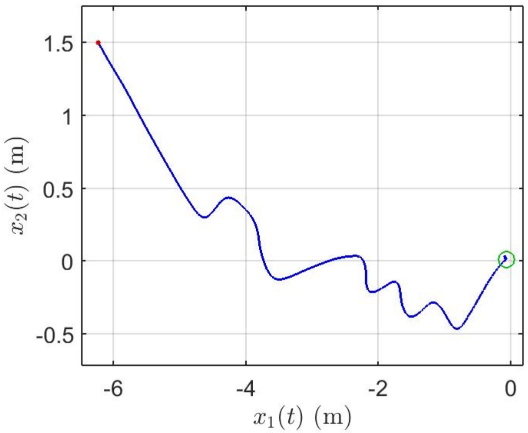
$$\dot{\Gamma}_c(t) = \beta_c\Gamma_c(t) - \Gamma_c(t)k_{c1}\frac{\omega(t)\omega^T(t)}{\rho^2(t)}\Gamma_c(t) - \Gamma_c(t)\frac{k_{c2}}{N}\sum_{k=1}^{N}\frac{\omega_k(t)\omega_k^T(t)}{\rho_k^2(t)}\Gamma_c(t)$$

$$\dot{\hat{W}}_a(t) = -\Gamma_a k_{a1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right) + \text{Possible Cross Terms}$$

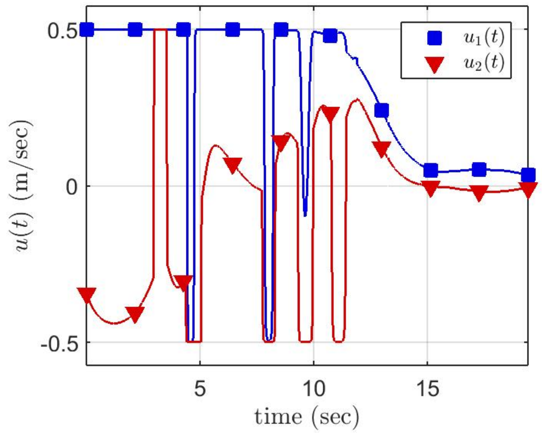
Actual system exploitation

Extrapolated system

Results – Experiment 1

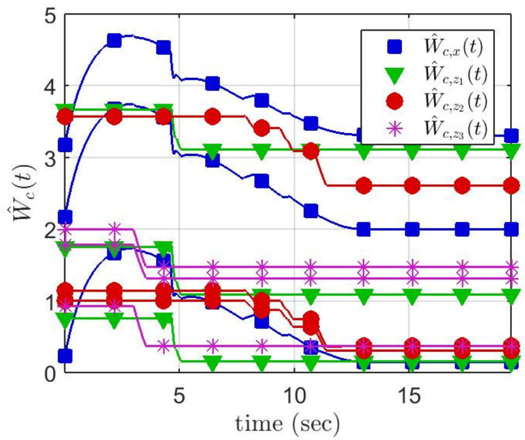


Phase-space portrait

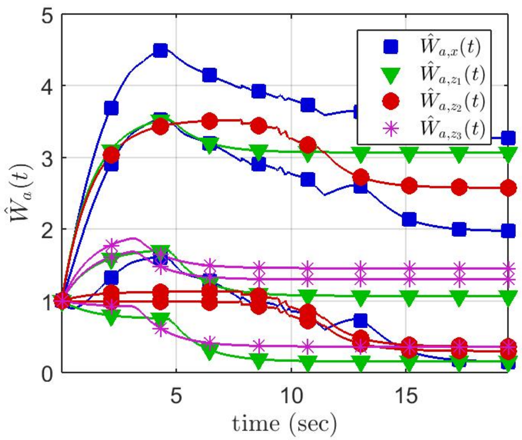


Approximate optimal input

- Modifies trajectory when avoidance regions are sensed
- Collision avoidance
- Control remains in saturation region



Critic weight estimate



Actor weight estimate

- Need to know the number of obstacles in advance
- Discriminate between the obstacles
- Large number of function approximations

Uncertain Avoidance Regions

Value function representation

$$V^{*}\left(x\left(t\right),Z\left(t\right)\right)=P_{a}\left(x\left(t\right),Z\left(t\right)\right)+V^{\#}\left(x\left(t\right),Z\left(t\right)\right)$$

$$V^{\#}\left(x\left(t\right),Z\left(t\right)\right)$$

Depends on possibly uncertain number of avoidance regions

Interpret $V^{\#}$ as a time-varying map

$$V_{t}^{\#}\left(x\left(t\right),t\right)=V_{t}^{\#}\left(x\left(t\right),\phi^{-1}\left(\kappa\right)\right)=V_{\kappa}^{\#}\left(x\left(t\right),\kappa\right)$$

Since $\kappa\in[0,\alpha]$ for $\alpha\in\mathbb{R}_{>0}$, we can approximate $V_{\kappa}^{\#}$ using StaF approximation

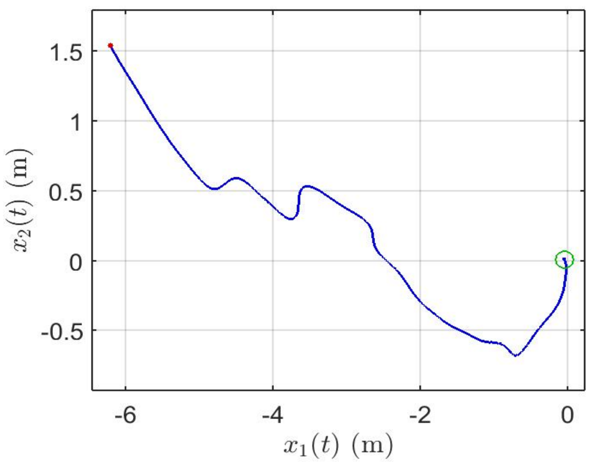
$$V_{\kappa}^{\#}\left(y\left(t\right)\right)=W^{T}\left(\zeta^{\#}\left(t\right)\right)\sigma\left(y\left(t\right),c\left(\zeta^{\#}\left(t\right)\right)\right)+\varepsilon\left(y\left(t\right),\zeta^{\#}\left(t\right)\right)\qquad y\in\overline{B_{r}\left(\zeta^{\#}\right)}$$

Redefined HJB contains uncertainties: $V_{\kappa}^{\#}, f, \mathcal{F}_i h_i$

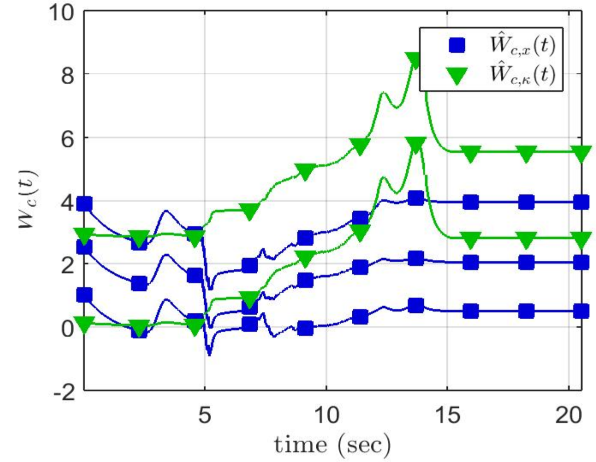
$$0=r\left(x,Z,u\right)+\frac{\partial V_{\kappa}^{\#}\left(\zeta^{\#}\right)}{\partial\zeta^{\#}}\left(F^{\#}\left(\zeta^{\#}\right)+G^{\#}\left(\zeta^{\#}\right)u\right)+\dot{P}_{a}$$

$$\dot{P}_{a}=\sum_{i=1}^{M}\frac{\partial P_{a,i}}{\partial x}\left(f\left(x\right)+g\left(x\right)u-\mathscr{F}_{i}\left(x,z_{i}\right)h_{i}\left(z_{i}\right)\right)$$

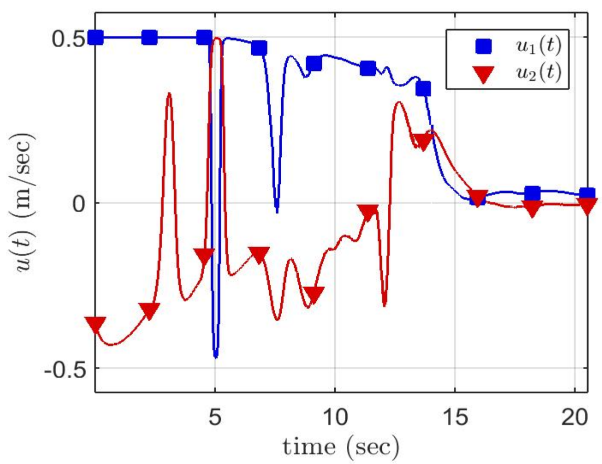
Results – Experiment 2



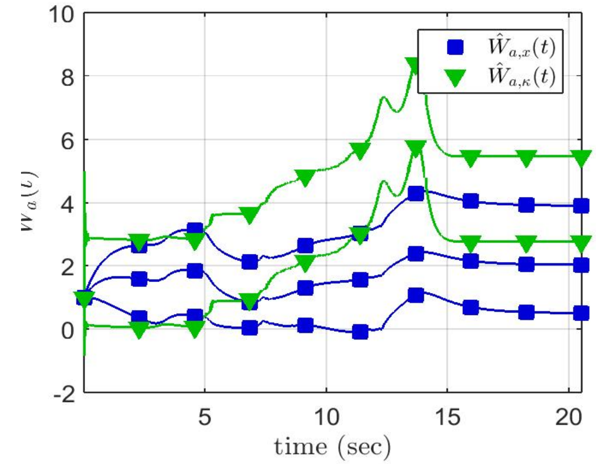
Phase-space portrait



Critic estimates



Control policy



Actor estimates

- Less than half the number of functions to approximate
- Estimates do not grow with number of obstacles
- No need to discriminate between obstacles
- Nearly identical costs

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

Example: Herding

Roaming agent

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

Influencing agent

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

Goal: Regulated roaming agent to desired goal location z_g

$$e_z(t) \triangleq z(t) - z_g$$

Influencing agent does not have direct control over roaming agent, and the influencing agent's state may be non-affine in roaming agent dynamics.

$$e_d(t) \triangleq \underbrace{\eta_d(t)}_{\text{Virtual state}} - z_g - k_d e_z(t) \qquad \dot{\eta}_d(t) \triangleq \underbrace{\mu_d(t)}_{\text{Virtual input}}$$

The pursuer tracks the virtual state using the auxiliary error and desired input

$$e_\eta(t) \triangleq \eta(t) - \eta_d(t)$$

Desired influencing agent input

$$u_d(t) \triangleq g(\eta_d(t))^{+} \mu_d(t) - g(\eta_d(t))^{+} h(z(t), \eta_d(t))$$

Input mismatch error

$$\mu_\eta(t) \triangleq u(t) - u_d(t)$$

The input mismatch $\mu_\eta(t)$ and virtual input $\mu_d(t)$ are designed to regulate the total state x(t)

$$\mu(t) \triangleq \begin{bmatrix} \mu_\eta^T(t) & \mu_d^T(t) \end{bmatrix}^T \qquad x(t) \triangleq \left[e_z^T(t), e_d^T(t), e_\eta^T(t)\right]^T$$

Optimal Control Formulation

Control Objective

Design a controller μ which minimizes

$$J(x,\mu) \triangleq \int_{t_0}^{\infty} r(x(\tau),\mu(\tau))d\tau$$

$$r\left(x,\mu\right) \triangleq Q\left(x\right) + P\left(x\right) + \Psi\left(\mu\right)$$

Dynamic constraints

$$\dot{x}\left(t\right) = F\left(x\left(t\right),\theta\right) + G\left(x\left(t\right)\right)\mu\left(t\right)$$

Hamilton Jacobi Bellman equation

$$0 = \nabla V^{*}\left(x\right)\left(F\left(x,\theta\right) + G\left(x\right)\mu^{*}\left(x\right)\right) + r\left(x,\mu^{*}\left(x\right)\right)$$

Optimal control policy

$$\mu^{*}\left(x\right) = -\frac{1}{2}R^{-1}G\left(x\right)^{T}\left(\nabla V^{*}\left(x\right)\right)^{T}$$

Replace uncertainties in dynamics and optimal HJB with estimates

$$V^{*},\ \nabla V^{*},\ \mu^{*},\ \theta \longrightarrow \hat{V},\ \nabla\hat{V},\ \hat{\mu},\ \hat{\theta}$$

Use actor-critic with StaF kernel method for value function approximation.
Use ICL for system ID.

Bellman error

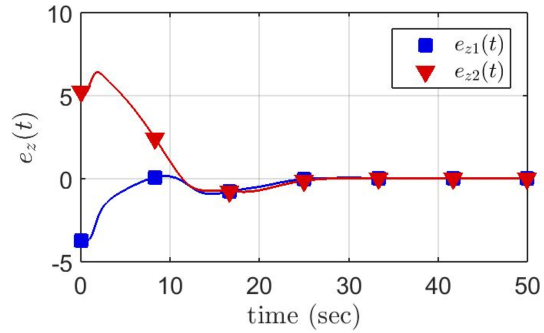
$$\delta\left(y,x,\hat{\theta},\hat{W}_c,\hat{W}_a\right) = r\left(y,\hat{\mu}\left(y,x,\hat{W}_a\right)\right) + \nabla\hat{V}\left(y,x,\hat{W}_c\right)\left(F\left(y,\hat{\theta}\right) + G\left(y\right)\hat{\mu}\left(y,x,\hat{W}_a\right)\right)$$

$$\hat{\mu}\left(y,x,\hat{W}_a\right) = -\frac{1}{2}R^{-1}G\left(x\right)^{T}\nabla\sigma\left(y,c\left(x\right)\right)^{T}\hat{W}_a$$

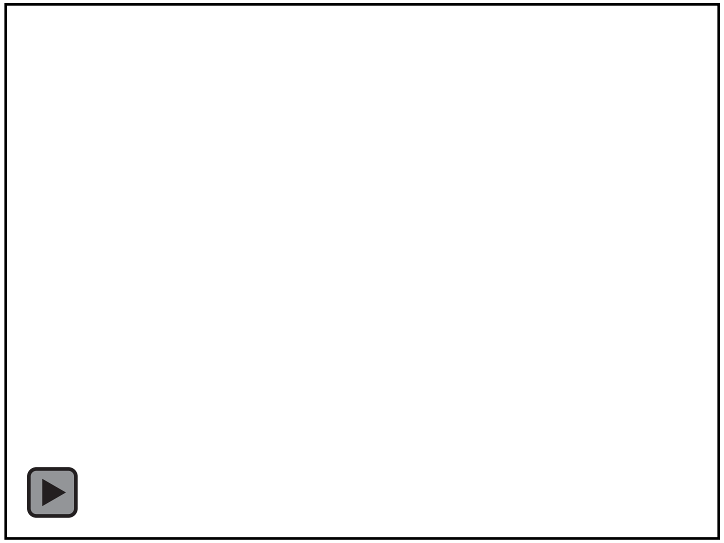
UF UNIVERSITY of FLORIDA
Duke UNIVERSITY
TEXAS The University of Texas at Austin
UC SANTA CRUZ

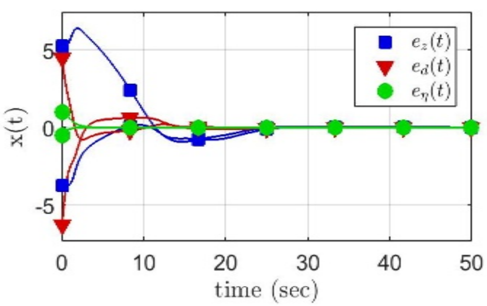
Simulation Results

Goal:
Regulate roaming agent (red) to $z_g = [0, -0.5]^T$.

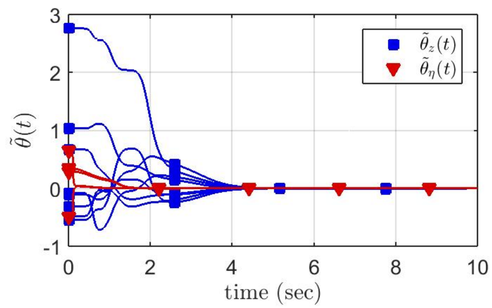


Target error

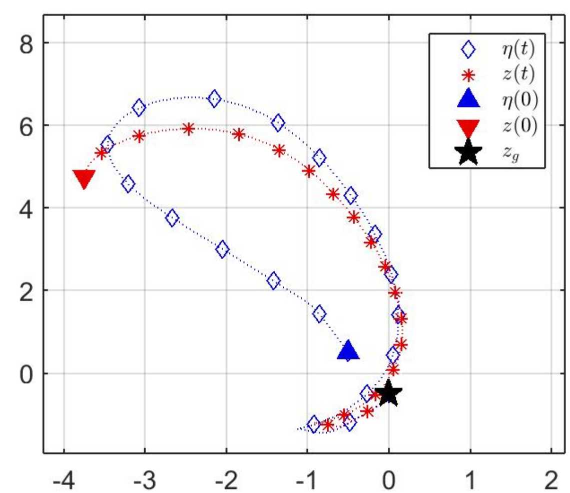




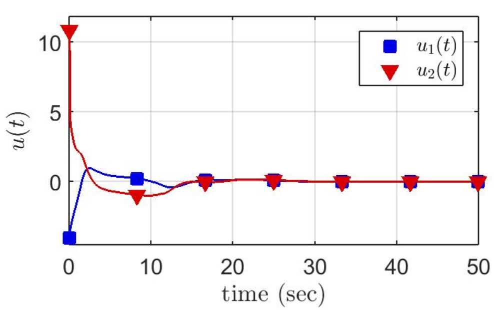
Total state



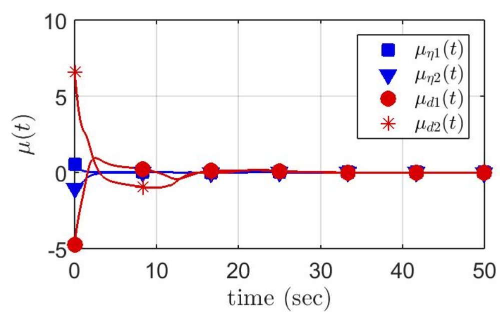
System identification error



Phase-space portrait



Herder input



Approximate optimal input

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

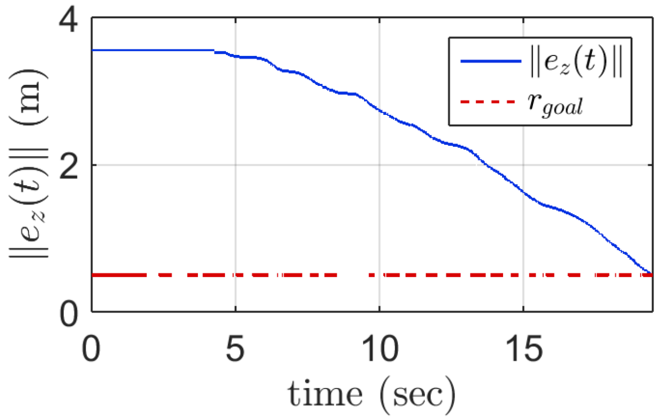
Experiment

- Experiment:
  - Parrot Bebop 2.0 quadcopter
  - Unactuated paper platform

- Goal:
  - Regulate roaming agent to a neighborhood ($r_{goal} = 0.5\ m$) about the desired location $z_g = [-2,0]^T\ m$

Influencing Agent (Parrot Bebop 2.0)

Roaming Agent (Paper platform)



Target error norm

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

Experiment