Distributed Estimation, Objective Identification, and Intelligent Control of Multiagent Systems

Bosen Lian, Ph.D.

Postdoctoral Research Associate, University of Texas, Arlington



CONTENT

- ❖ Motivation
- ❖ Distributed Estimation with Distributed Kalman Filters
- ❖ Objective Identification with Inverse Reinforcement Learning
- ❖ Multiagent Control with Reinforcement Learning
- ❖ Future Work

CONTENT

- ❖ **Motivation**
- ❖ Distributed Estimation with Distributed Kalman Filters
- ❖ Objective Identification with Inverse Reinforcement Learning
- ❖ Multiagent Control with Reinforcement Learning
- ❖ Future Work

Estimation Theory

Machine Learning

Control Theory

Estimation Theory

Machine Learning

Control Theory

Distributed Kalman Filters

Inverse Reinforcement Learning

Reinforcement Learning

Distributed Estimation

Objective/Reward Identification

Multiagent Control

Estimation → Learning → Action



Optimize Performance Objective Function

Optimal Behaviors

CONTENT

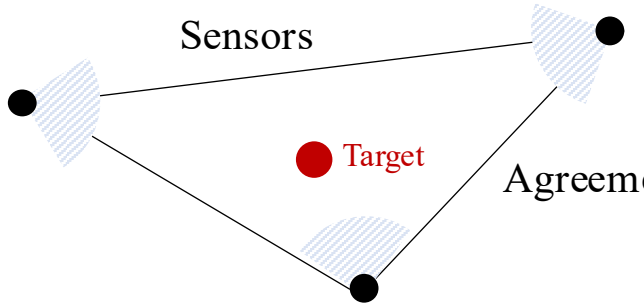
- Motivation
- **Distributed Estimation with Distributed Kalman Filters**
- Objective Identification with Inverse Reinforcement Learning
- Multiagent Control with Reinforcement Learning
- Future Work

Challenges in Distributed Kalman Filters

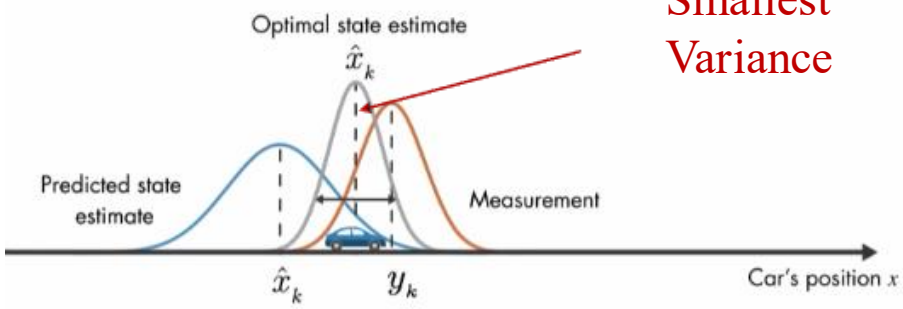
Estimation Theory

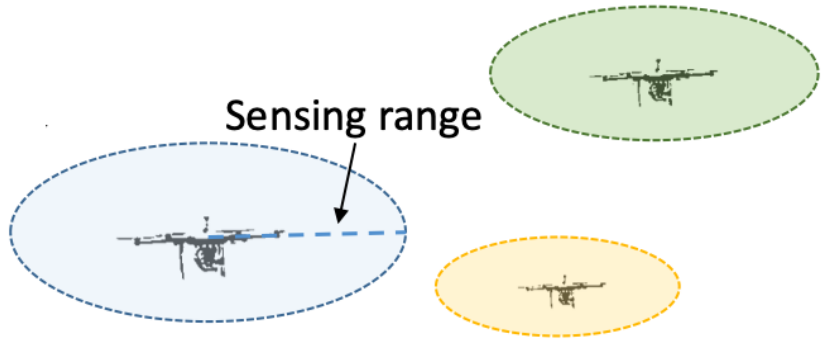
Distributed Kalman Filters for Estimation

1. **Consensus**
2. **Optimality**
3. **Limited Sensing Range**
4. **Robustness**



Smallest Variance





Perturbation

Uncertainty

→ Kalman Filter Estimation

To Address the Challenges

Developed **Distributed Kalman Consensus Filter** for Estimation with Moving Targets



1. Consensus
2. Near Optimality
3. Limited Sensing Range



Analyzed **Robustness Margins**

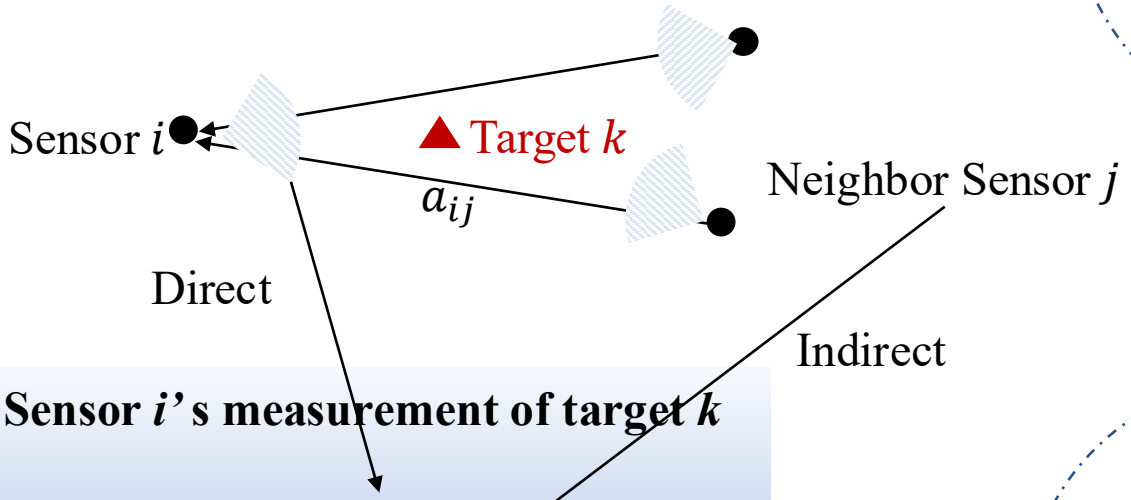


4. Guaranteed Robustness

Distributed Kalman Consensus Filter for Estimation with Moving Targets

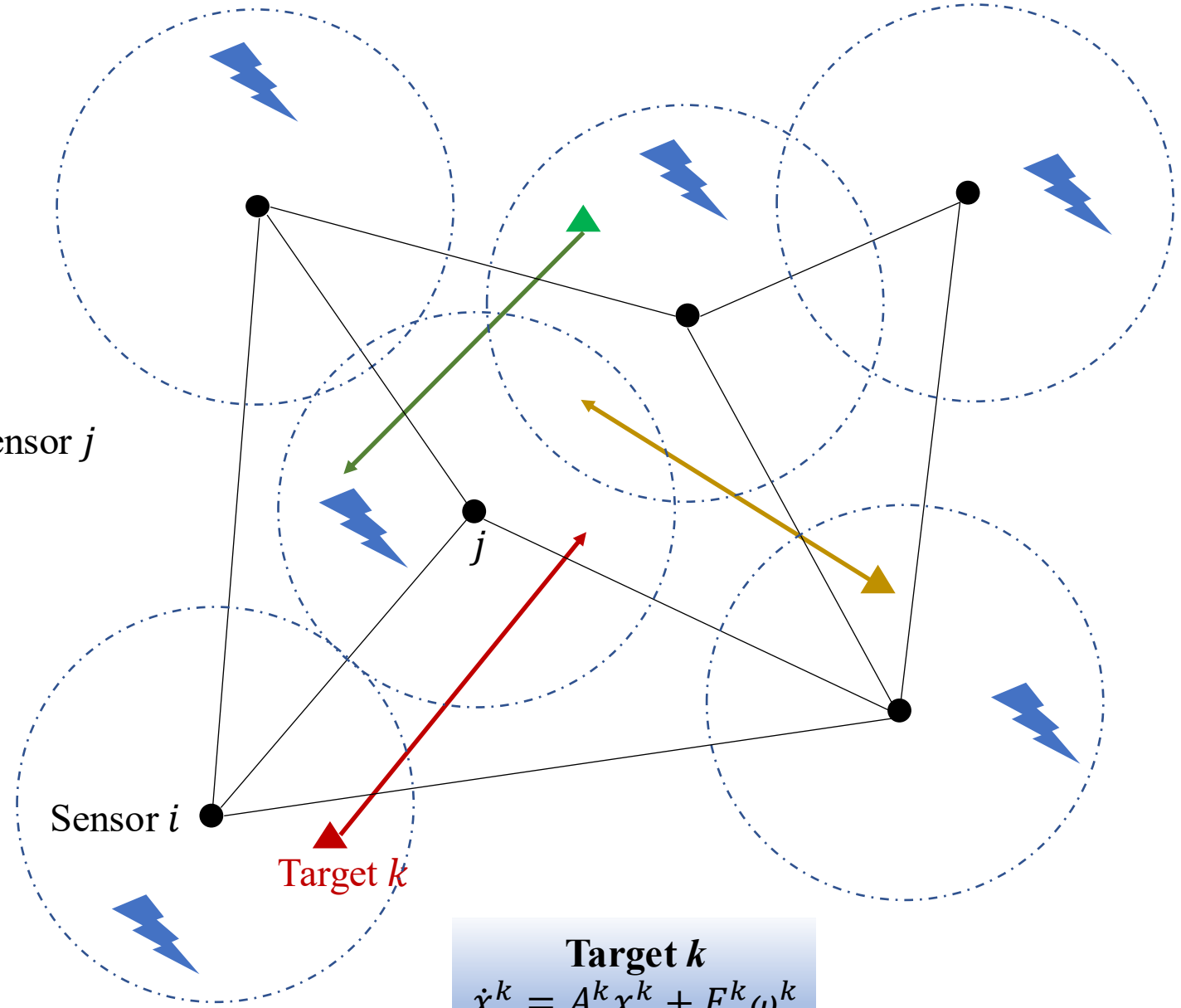
Tech Transfer to Industry

"Heterogeneous Autonomous Networks for Sensor Optimizing Locomotion," $150,000 contract from Lockheed Martin Advanced Technology Labs



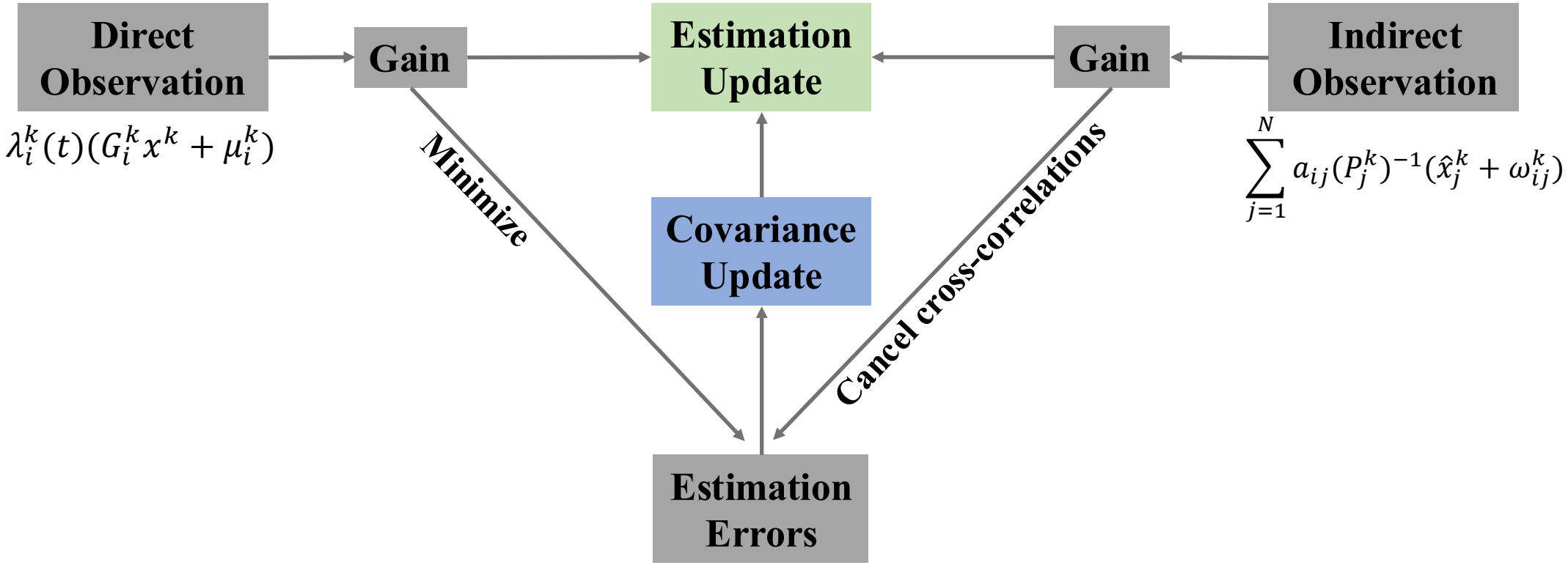
Sensor i's measurement of target k

$$z_i^k = \begin{bmatrix} \lambda_i^k(d_i^k, t)(G_i^k x^k + \mu_i^k) \\ \sum_{j=1}^{N} a_{ij}(P_j^k)^{-1}(\hat{x}_j^k + \omega_{ij}^k) \end{bmatrix}$$



Target k

$$\dot{x}^k = A^k x^k + F^k \omega^k$$



Distributed Estimation Update

$$\dot{\hat{x}}_i^k = A^k \hat{x}_i^k - \lambda_i^k P_i^k (G_i^k)^T (R_i^k)^{-1} (G_i^k (x^k - \hat{x}_i^k) + \mu_i^k)$$
$$+ 2d_i^k P_i^k [\sum_{j=1}^{N} a_{ij}^k (P_j^k)^{-1} (P_i^k + P_j^k)]^{-1} \sum_{j=1}^{N} a_{ij}^k (P_j^k)^{-1} (\hat{x}_j^k - \hat{x}_i^k)$$

Distributed Covariance Update

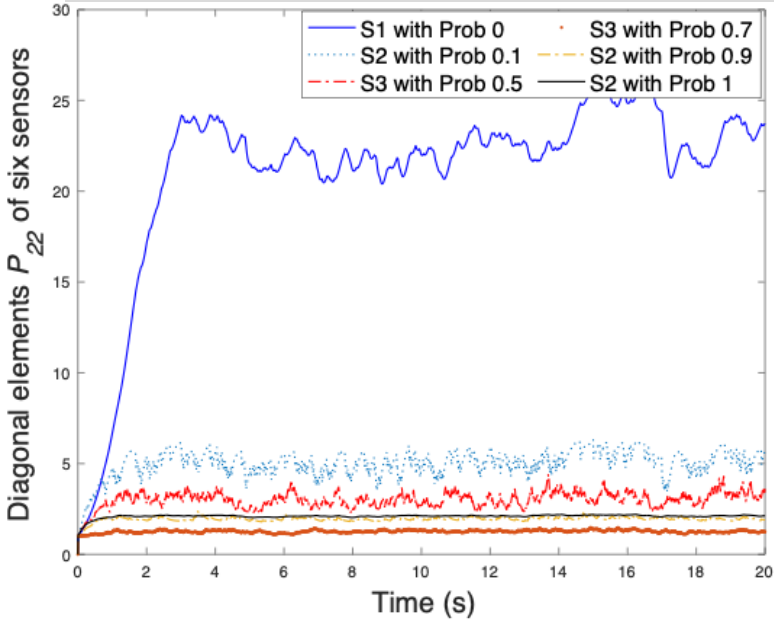
$$\dot{P}_i^k = A_i^k P_i^k + P_i^k (A_i^k)^T - \lambda_i^k P_i^k (G_i^k)^T (R_i^k)^{-1} G_i^k P_i^k + F^k W^k (F^k)^T$$
$$- 4 \sum_{j=1}^{N} a_{ij}^k P_i^k [\sum_{j=1}^{N} a_{ij}^k (P_j^k)^{-1} (P_i^k + P_j^k)]^{-1} \sum_{j=1}^{N} a_{ij}^k (P_j^k)^{-1} P_i^k$$

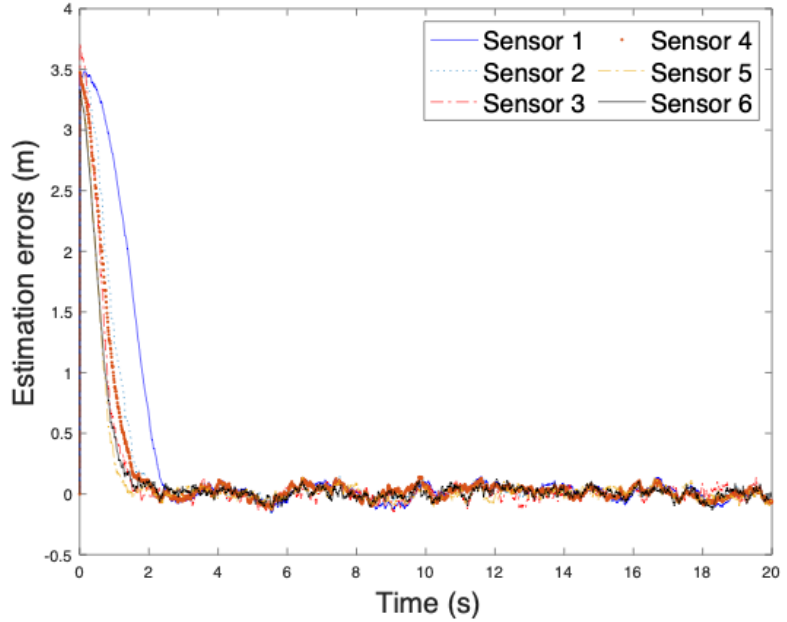
Effectiveness



Fig. 1 Ford Taurus

Average Covariance Value P_{22}

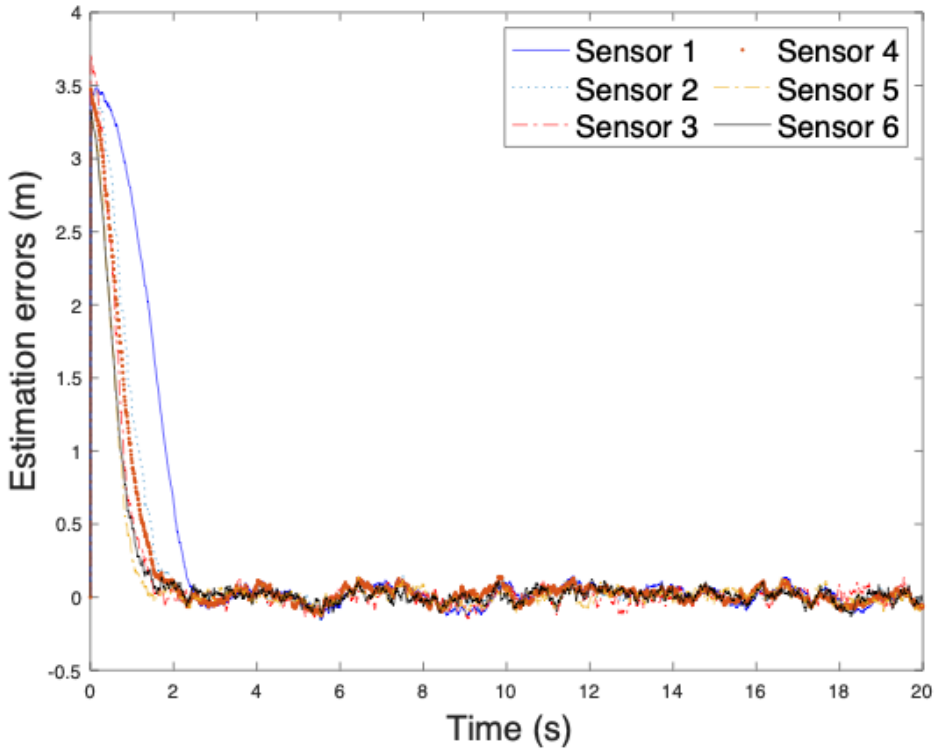




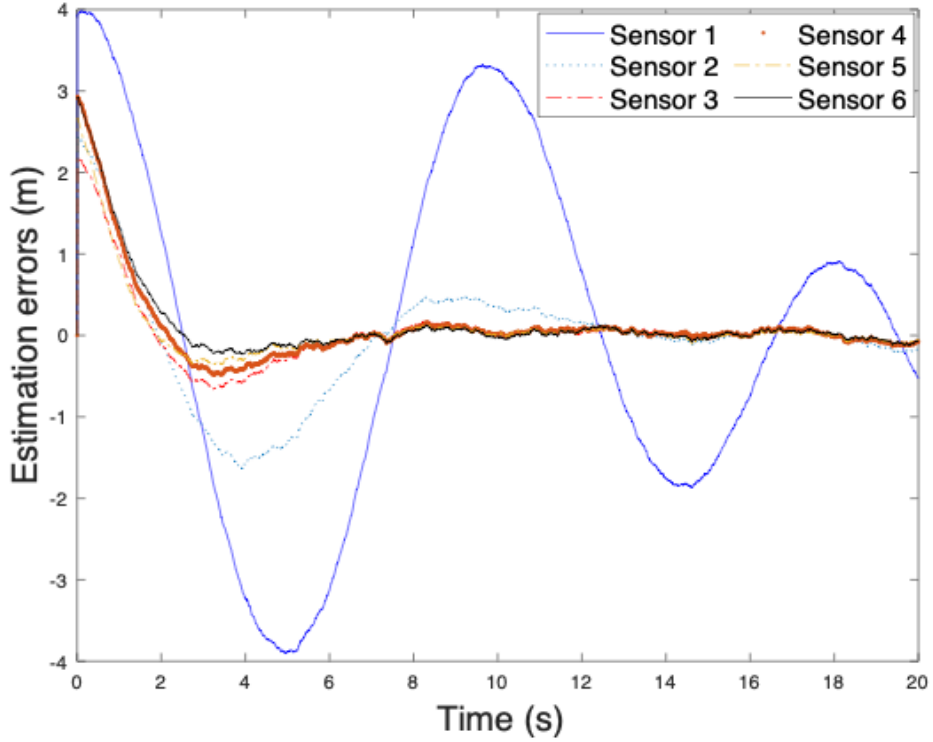
Lian, B., et al. (2019) "Distributed consensus-based Kalman filtering for estimation with multiple moving targets." *In 2019 IEEE 58th Conference on Decision and Control*.
Lian, B., et al. (2020). Distributed Kalman consensus filter for estimation with moving targets. *IEEE Transactions on Cybernetics*.

Faster Convergence

The Convergence of Average Estimation Errors of Each Sensor



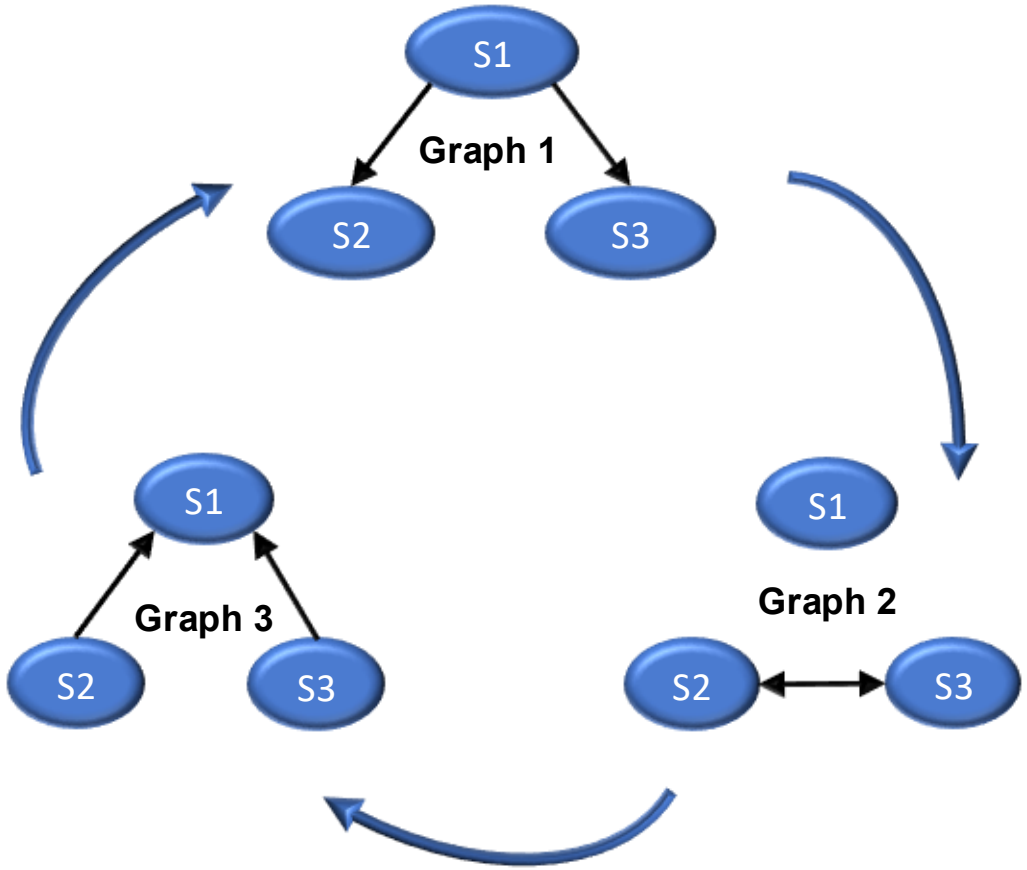
By DKCF

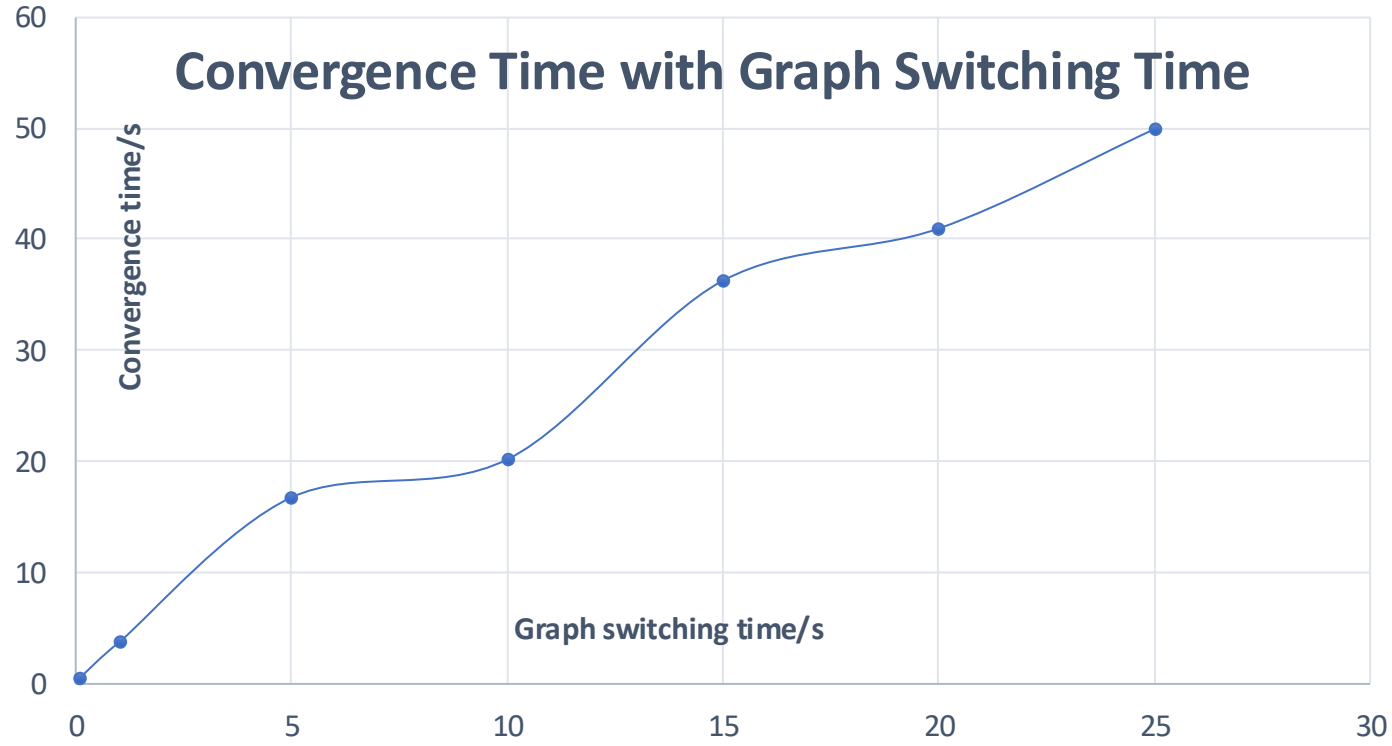


By KCF[5]

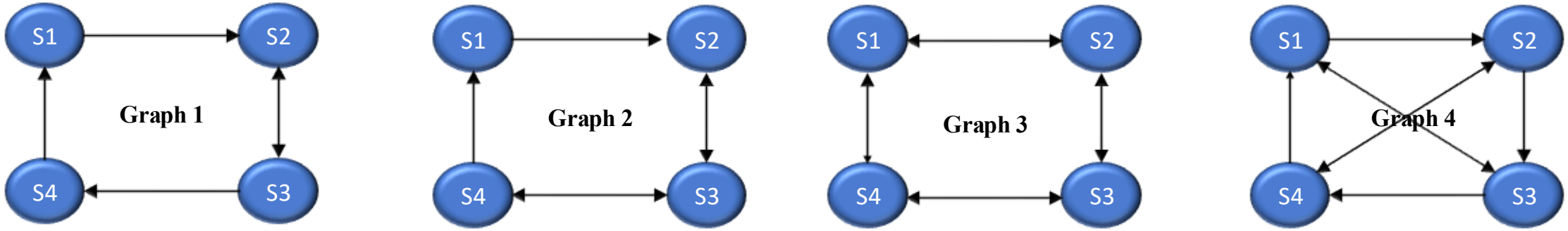
[5] Olfati-Saber, Reza. "Distributed Kalman filtering for sensor networks." *2007 46th IEEE Conference on Decision and Control*.

Convergence with Graph Switching

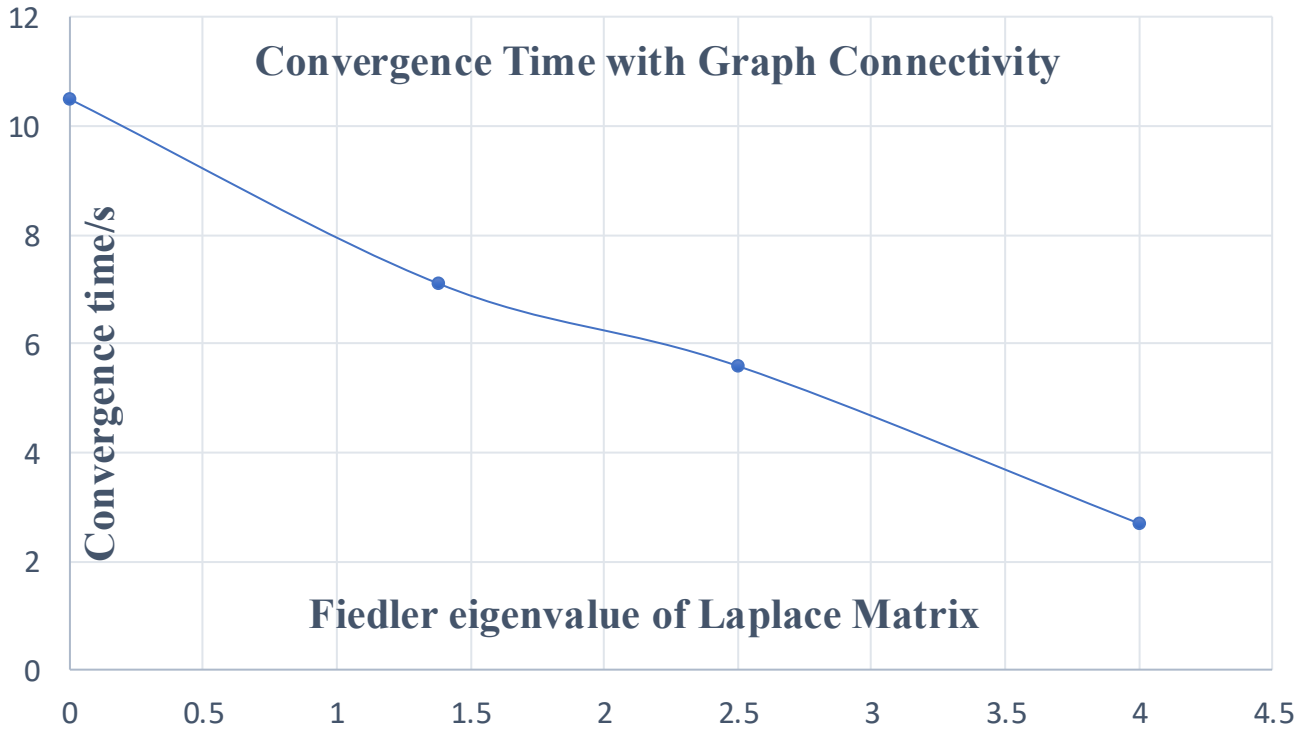




Convergence with Graph Connectivity



More Connected



Robustness Margin Analysis

Funded by ONR grant "Optimal Design for Assured Performance of Interactive Multibody Systems: Guaranteed Controls for Multi-pursuers, Estimation, Optimal Learning, Scalable Uncertainty Sampling, and Time-critical Communication", $815,019.00.

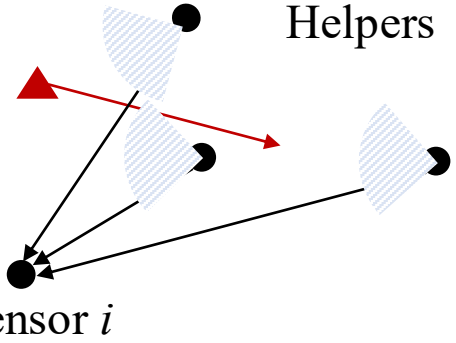
	Standard Kalman filter	Distributed Kalman filters
• Phase margin:	$\|PM\| < 60^0$	$\|PM\| < \arccos(\frac{1}{2} - \frac{(\alpha_i^*+2)\alpha_i^*}{2})$
• Gain margin:	$\frac{1}{2} < \|GM\| < \infty$	$\frac{1}{2+\alpha_i^*} < \|GM\| < \infty$

Improved Robustness

α_i^* is related to the neighbor numbers of sensor *i*.



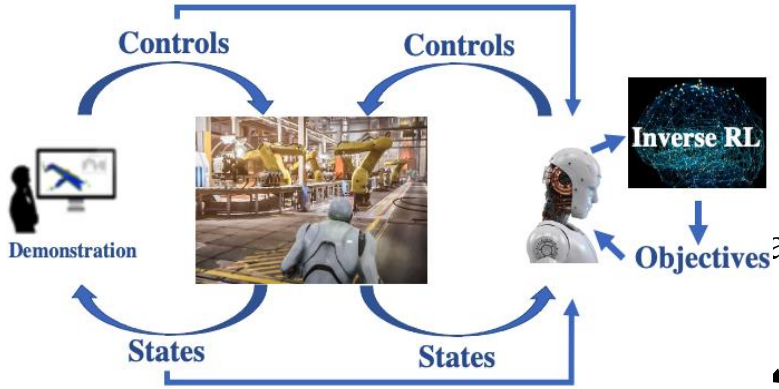
No Helper



Lian, B., Lewis, F. L., Hewer, G. A., Estabridis, K., & Chai, T. (2021). Robustness analysis of distributed Kalman filter for estimation in sensor networks. *IEEE Transactions on Cybernetics*.

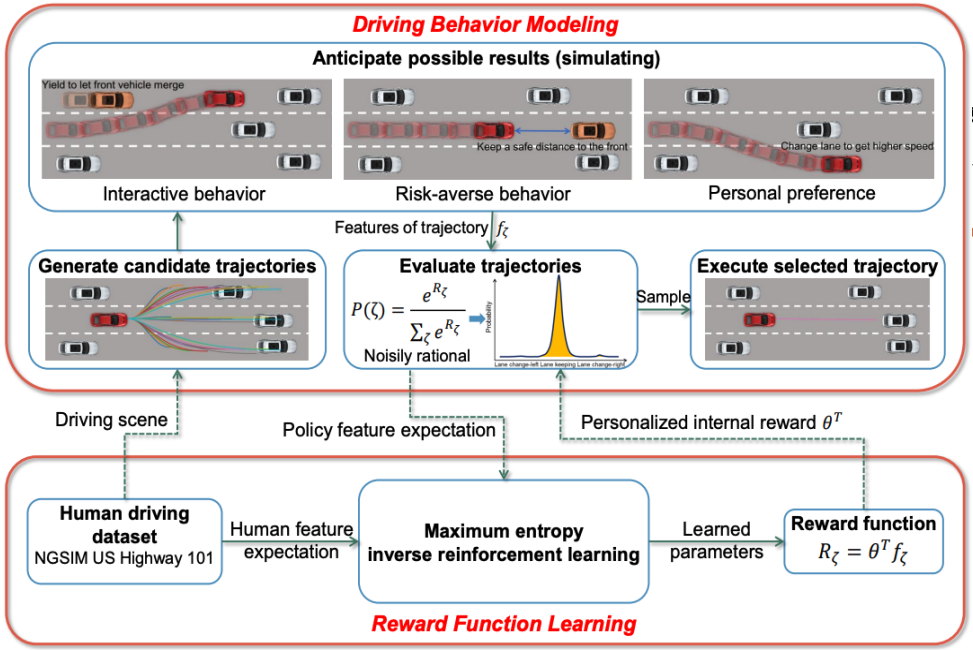
CONTENT

- Motivation
- Distributed Estimation with Distributed Kalman Filters
- **Objective Identification with Inverse Reinforcement Learning**
- Multiagent Control with Reinforcement Learning
- Future Work



ate/Track

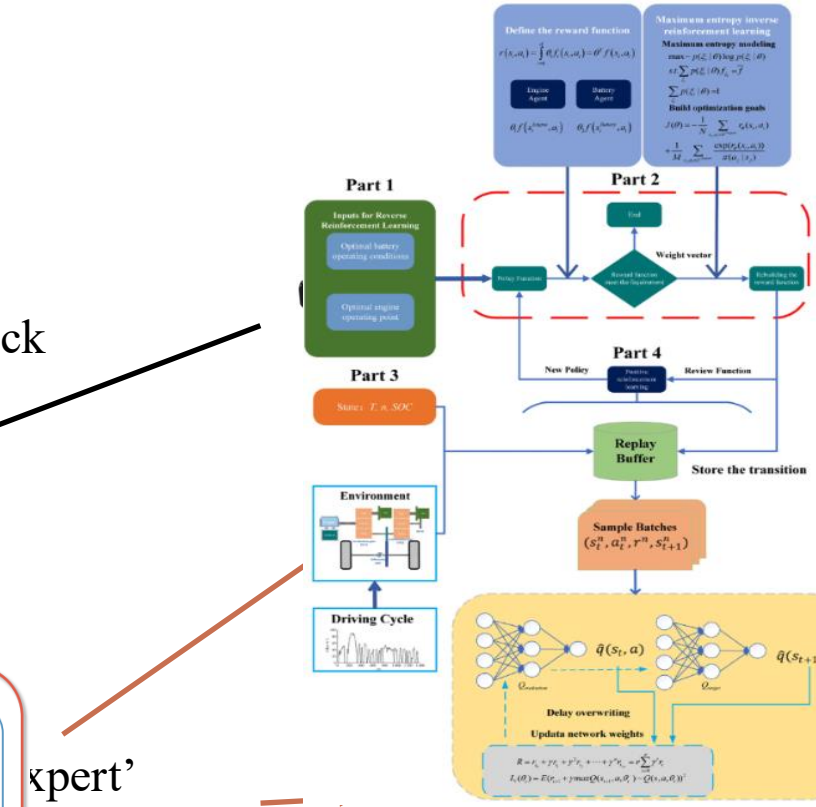
Human Machine Interactions



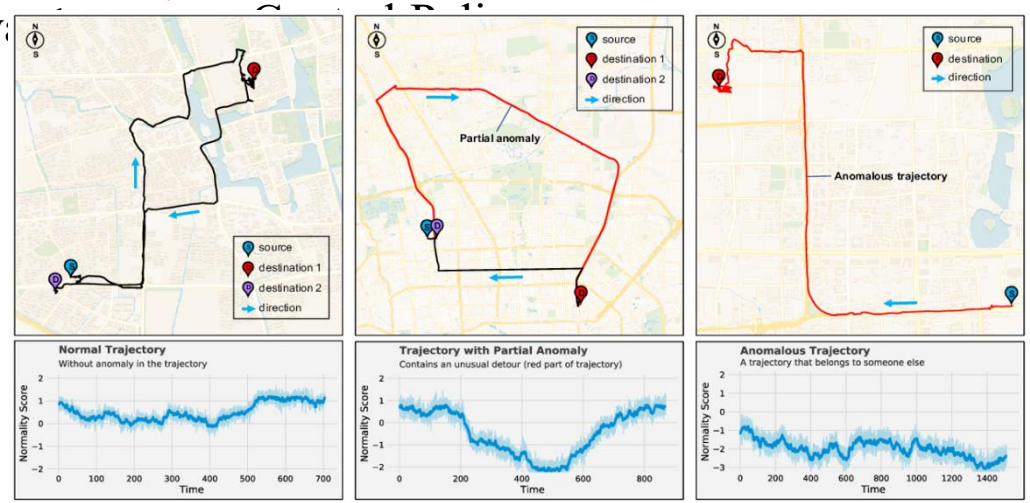
Autonomous Driving

xpert'

ve/Rew



Energy Management



Anomaly Detection

Challenges in Inverse Reinforcement Learning

Machine Learning

Control Theory

Objective Identification with Inverse Reinforcement Learning

1. Dynamic Systems

2. Unknown System Dynamics

3. Stability and Robustness



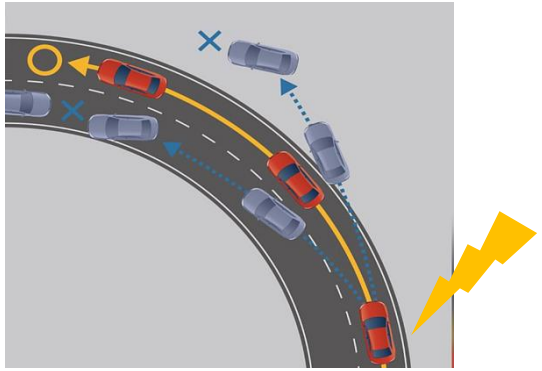
$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau$$

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

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

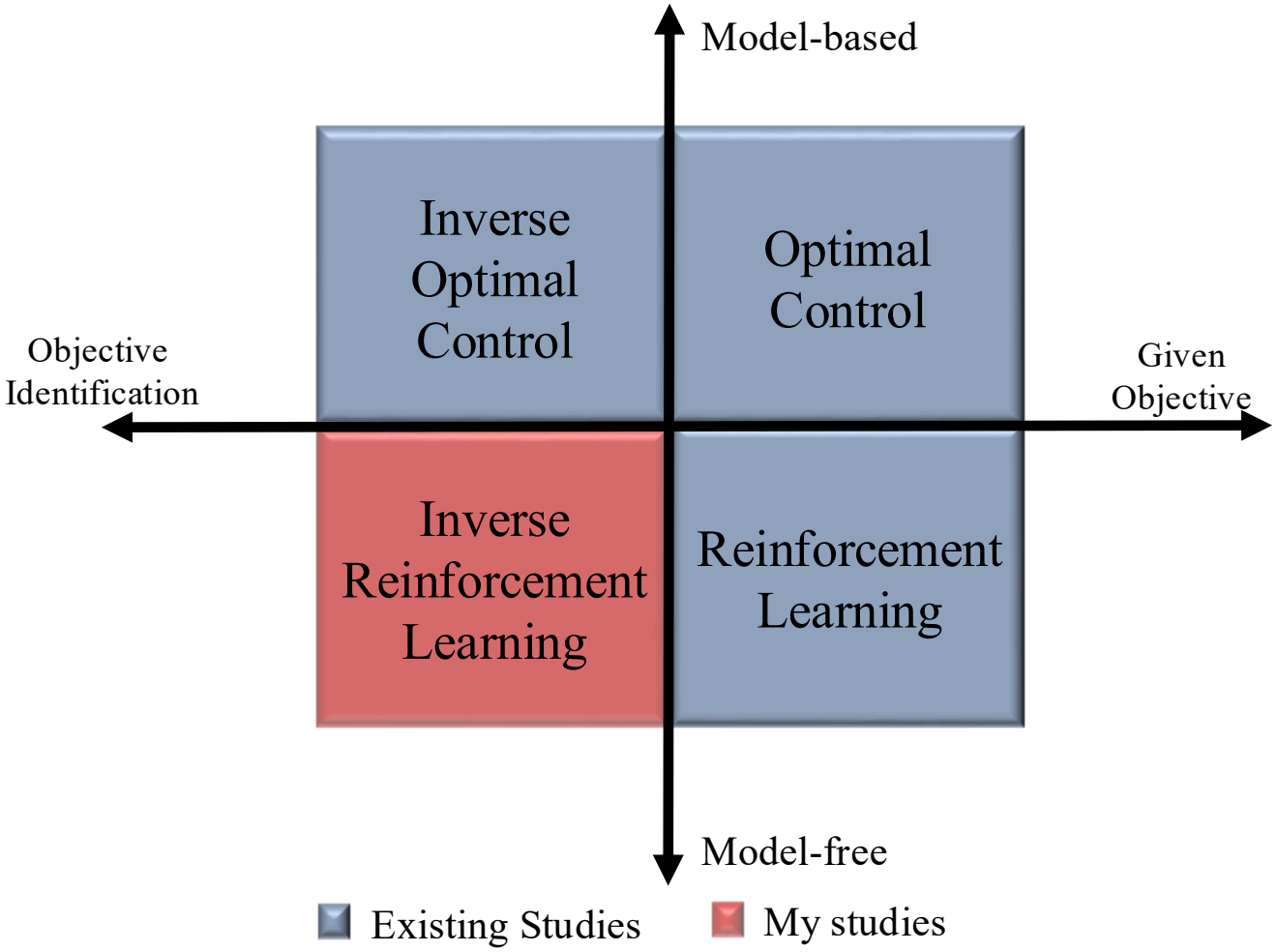
$$\dot{x} = f(x) + g(x)u$$

Unknown A, B, $f(x)$, and $g(x)$

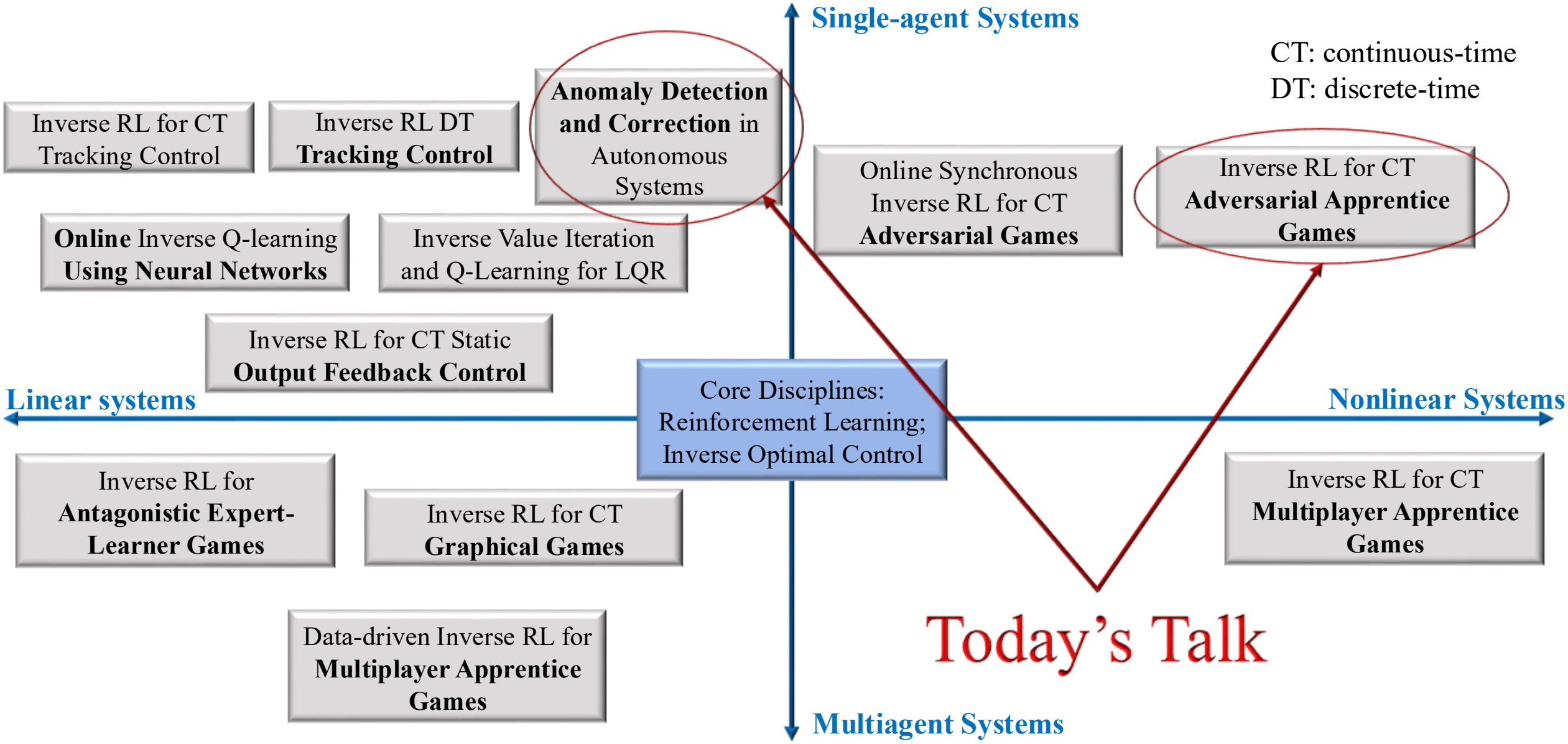


Inverse Reinforcement Learning for Optimal Control Systems and Games

- ✓ A New Discipline in Automatic Control
- ✓ A **Hybrid** of Reinforcement Learning and Inverse Optimal Control
- ✓ Lyapunov Stability



Our Results of Inverse Reinforcement Learning



Inverse Reinforcement Learning for Adversarial Apprentice Games



Expert

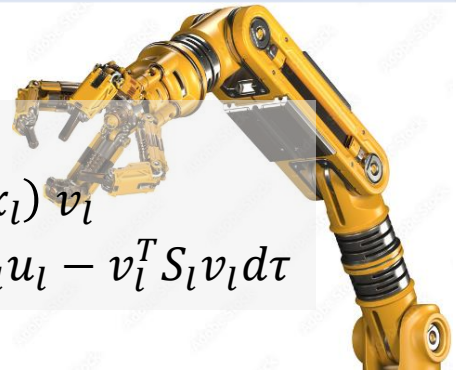
Dynamics: $\dot{x}_e = f(x_e) + g(x_e)\,u_e + h(x_e)\,v_e$

Objective: $V_e = \int_0^\infty x_e^T Q_e x_e + u_e^T R_e u_e - v_e^T S_e v_e d\tau$

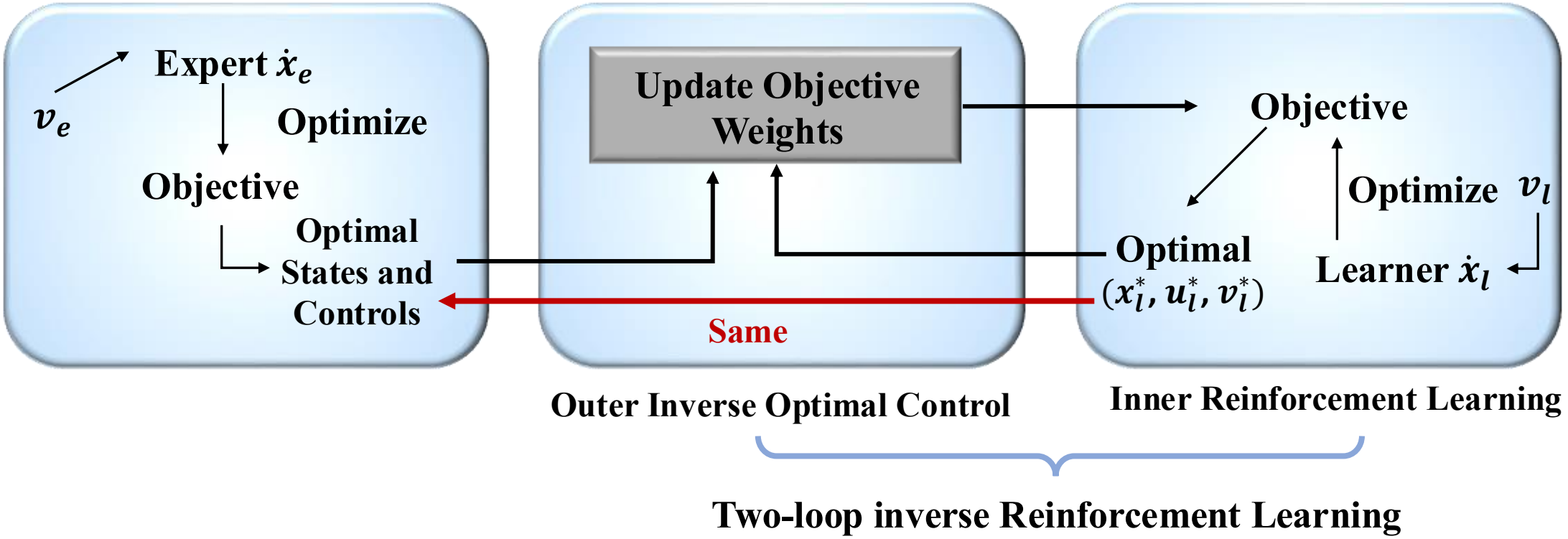
Learner

Dynamics: $\dot{x}_l = f(x_l) + g(x_l)\,u_l + h(x_l)\,v_l$

Trail Objective: $V_l = \int_0^\infty x_l^T Q_l x_l + u_l^T R_l u_l - v_l^T S_l v_l d\tau$



UNKNOWN



➢ Model-Free Data-Driven Inverse Reinforcement Learning Algorithm

Algorithm 1 Model-free Integral Inverse RL Algorithm for Adversarial Apprentice Game

1: **Initialization**: select $Q_l^0 \geq 0$, $R_l > 0$, $S_l > 0$, initial stabilizing u_l^{00}, and small thresholds ε_1, ε_2. Set $j = 0$. Apply stabilizing u_l to the learner dynamics (27).

2: **Outer j iteration loop based on inverse optimal control**

3: **Inner i iteration loop using optimal control:** given j, set $i = 0$, and use initial stabilizing u_l^{j0}.

4: Off-policy Integral RL for solving V_l^j, u_l^j and v_l^j

$$V_l^{ji}(x_l(t+T)) - V_l^{ji}(x_l(t)) - \int_t^{t+T} 2\Big((u_l^{j(i+1)})^T R_l (u_l - u_l^{ji}) - (v_l^{j(i+1)})^T S_l (v_l - v_l^{ji})\Big) d\tau = \int_t^{t+T} \Big(-q^T(x_l) Q_l^j q(x_l) - (u_l^{ji})^T R_l u_l^{ji} + (v_l^{ji})^T S_l v_l^{ji}\Big) d\tau.$$

5: Stop if $\|V_l^{ji} - V_l^{j(i-1)}\| \leq \varepsilon_1$, then set $V_l^j = V_l^{ji}$, $u_l^j = u_l^{ji}$ and $v_l^j = v_l^{ji}$. Otherwise, set $i \leftarrow i+1$ and go to Step 4.

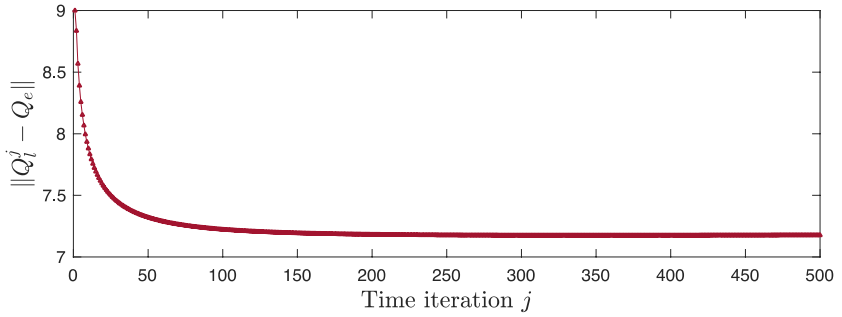
6: **State-penalty weight Q_l^{j+1} update using the expert's demonstration u_e^***

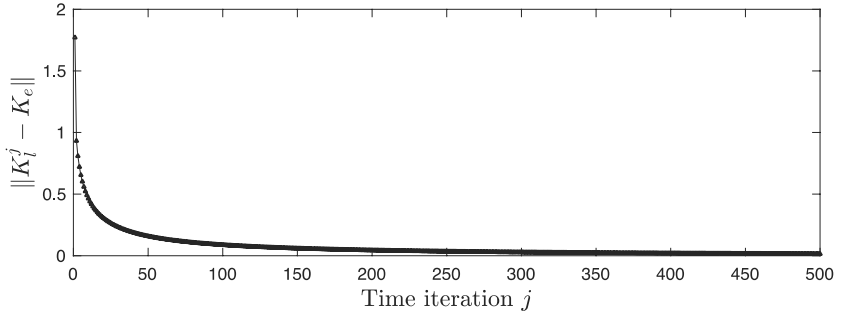
$$\int_t^{t+T} q^T(x_l) Q_l^{j+1} q(x_l) d\tau = \int_t^{t+T} \Big(u_e^{*T} R_l u_e^* - 2u_e^{*T} R_l u_l^j + (v_l^j)^T S_l v_l^j\Big) d\tau - V_l^j(x_l(t+T)) + V_l^j(x_l(t)).$$

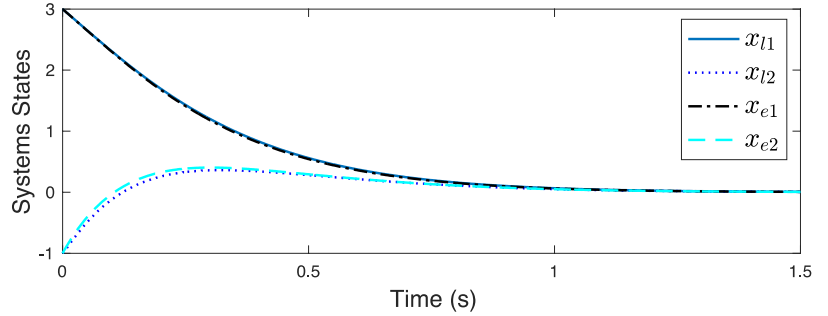
7: **Stop if** $\|Q_l^{j+1} - Q_l^j\| \leq \varepsilon_2$. Otherwise, set $u_l^{(j+1)0} = u_l^j$ and $j \leftarrow j+1$, then go to Step 3.

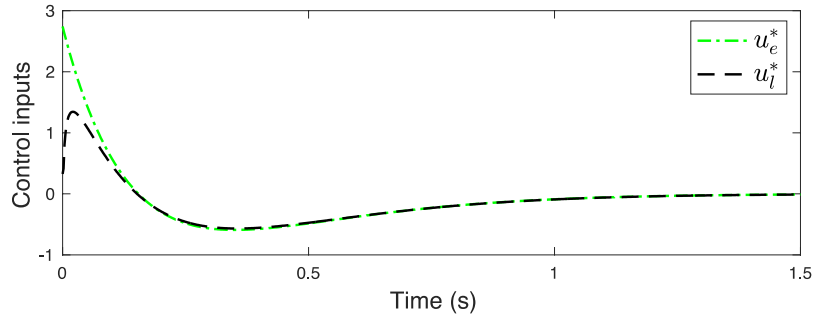
Reinforcement Learning (Off-Policy)

Inverse Optimal Control



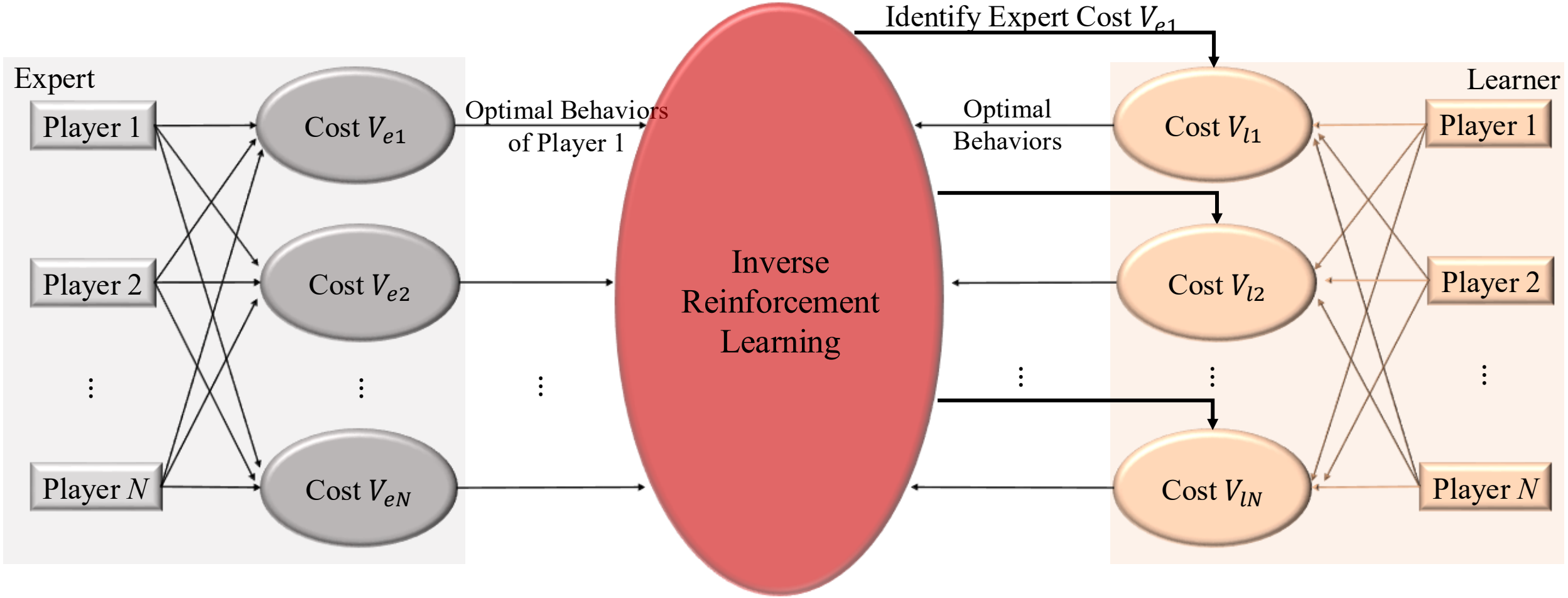






Lian, B., et al. (2021). Inverse reinforcement learning for adversarial apprentice games. *IEEE Transactions on Neural Networks and Learning Systems*.

Extended to Multi-player Apprentice Games



Lian, B., et al. (2022). Inverse reinforcement learning for multi-player noncooperative apprentice games. *Automatica*.

Inverse Reinforcement Learning for Anomaly Detection and Correction

Funded by ARO grant "Graphical Games and Distributed Reinforcement Learning Control in Human- networked Multi-group Societies", $391,083.00.

Unknown autonomous system with normal behavior



$$\dot{x}_n(t) = Ax_n(t) + Bu_n(t)$$

$$V_n(x_n(t_0), u_n) = \int_{t_0}^{\infty} (x_n^T Q_n x_n + u_n^T R_n u_n) d\tau$$

Objective

$$u_n^*(t) = -K_n x_n(t)$$

Intention

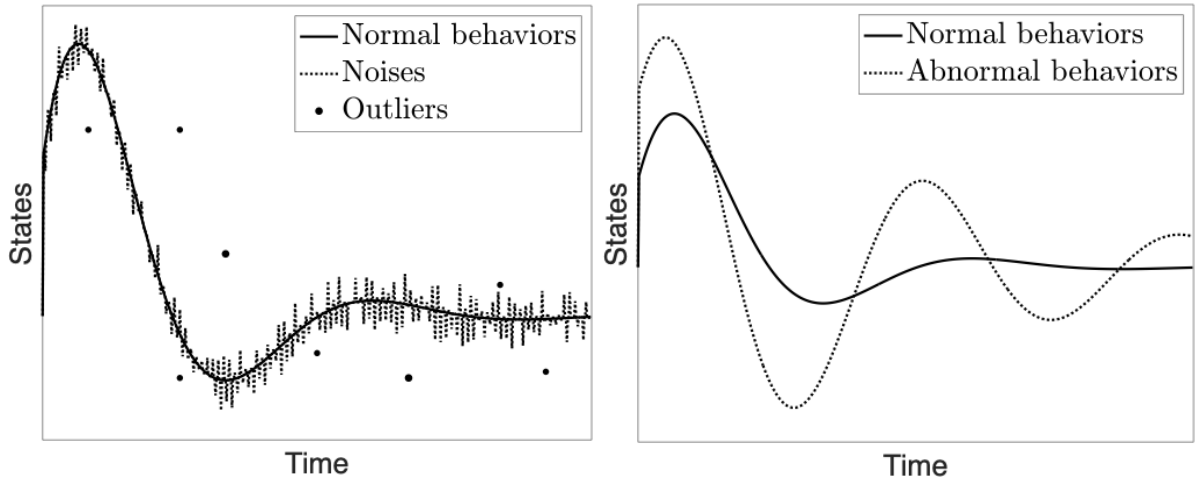
Model unknown test behavior

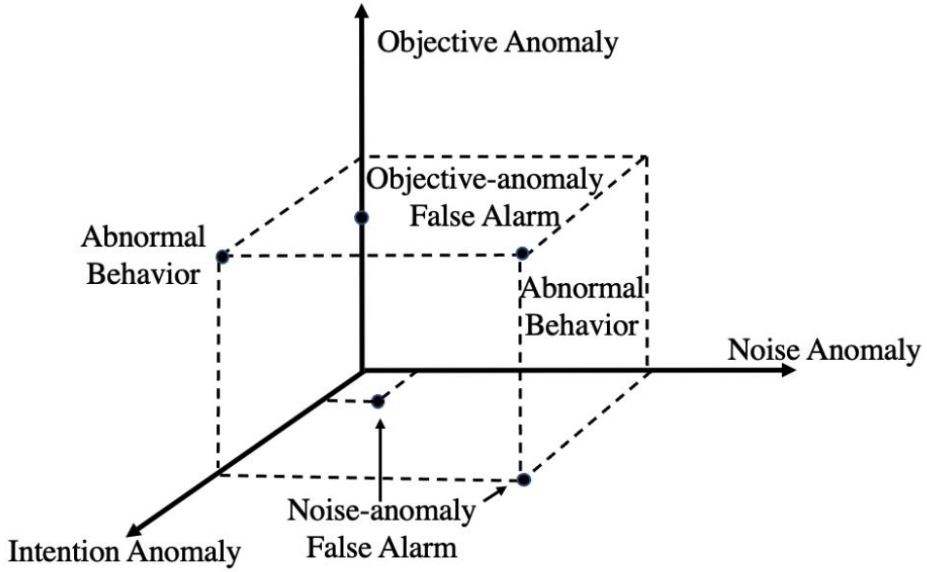
$$\dot{\hat{x}}(t) = A\hat{x}(t) + B\hat{u}(t)$$

$$\hat{V}(\hat{x}(t_0), \hat{u}) = \int_{t_0}^{\infty} (\hat{x}^T \hat{Q} \hat{x} + \hat{u}^T \hat{R} \hat{u}) d\tau$$

$$\hat{u}^*(t) = -\hat{K}\hat{x}(t)$$

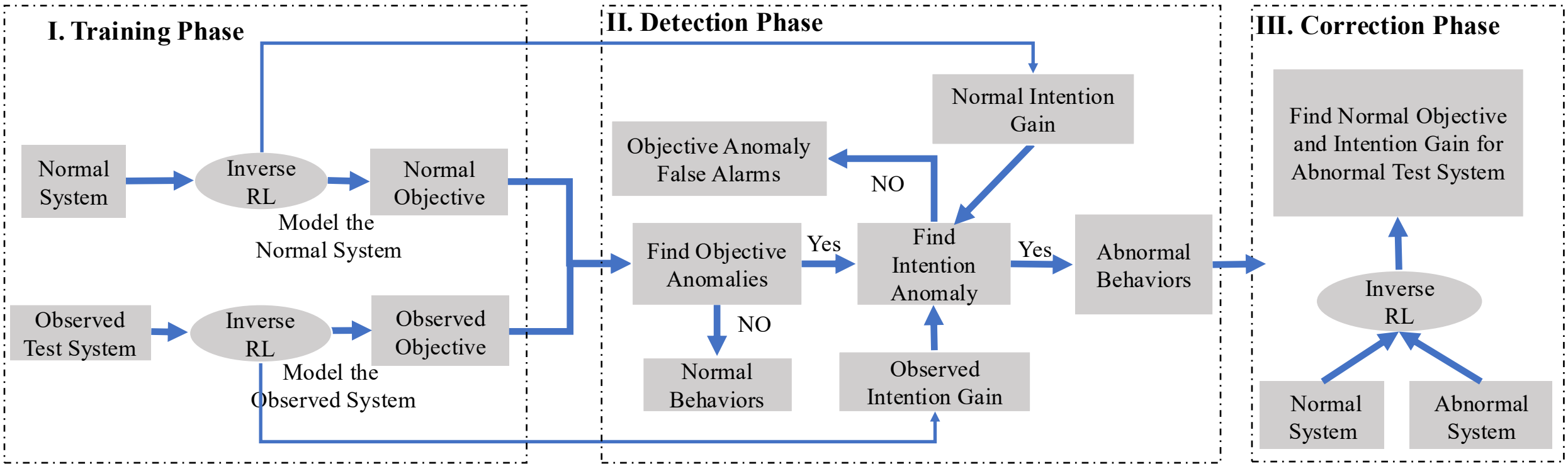






Bosen Lian, Yusuf Kartal, Frank L. Lewis, Dariusz G. Mikulski, Gregory R. Hudas, Yan Wan, and Ali Davoudi, (2022). "Anomaly Detection and Correction of Optimizing Autonomous Systems with Inverse Reinforcement Learning," *IEEE Transactions on Cybernetics*.

Diagram


I. Training Phase
Normal System
Inverse RL
Normal Objective
Model the Normal System
Observed Test System
Inverse RL
Observed Objective
Model the Observed System
II. Detection Phase
Normal Intention Gain
Objective Anomaly False Alarms
NO
Find Objective Anomalies
Yes
Find Intention Anomaly
Yes
Abnormal Behaviors
NO
Normal Behaviors
Observed Intention Gain
III. Correction Phase
Find Normal Objective and Intention Gain for Abnormal Test System
Inverse RL
Normal System
Abnormal System

Algorithm 1 Data-driven Inverse RL for the Normal Behaviors

1: Select $Q_0 = 0 \in \mathbb{R}^{k\times k}$, any $R = R^T \in \mathbb{R}^{m\times m} > 0$, threshold e_1, and stabilizing K_0. Set $i = 0$. Add probing noise $\bar{e}$ to u_n^*.

2: **Observe** the normal behavior trajectories, $(x_n(t), u_n^*(t))$ at each small interval T.

3: **Compute** P_i **and** K_{i+1} **from**

$$x_n^T(t+T)P_i x_n(t+T) - x_n^T(t)P_i x_n(t) - \int_t^{t+T} 2(u_n^* - u_{n_i})^T R K_{i+1} x_n d\tau = -\int_t^{t+T} \left((u_n^* - u_{n_i})^T R(u_n^* - u_{n_i}) + x_n^T Q_i x_n + u_{n_i}^T R u_{n_i}\right) d\tau,$$

Intention Reconstruct

4: **Inverse Optimal Control to Improve** Q_{i+1} **by**

$$\int_t^{t+T} x_n^T Q_{i+1} x_n d\tau = x_n^T(t)P_i x_n(t) - x_n^T(t+T)P_i x_n(t+T) - \int_t^{t+T} \left(u_{n_i}^T R u_{n_i} - 2(u_n^* - u_{n_i})^T K_{i+1} x_n\right) d\tau.$$

Objective Reconstruct

5: **Stop if** $\|K_{i+1} - K_i\| \le e_1$, otherwise, set $i \to i+1$, and go to Step 3.

Algorithm 2 Inverse RL Algorithm for Anomaly Detection

1: **Run** Algorithm 1 to obtained $(\hat{Q}_\infty, R)$ and $\hat{K}_\infty$ for online $(\hat{x}, \hat{u}^*)$.
2: **Compute** $\|Q_\infty - \hat{Q}_\infty\|$
3: **If** $\|Q_\infty - \hat{Q}_\infty\| > \varepsilon_1$, **then**
4: **Return** objective anomaly,
5: **Compute** $\|\hat{K}_\infty - K_\infty\|$;
6: If $\|\hat{K}_\infty - K_\infty\| > \varepsilon_2$, **then**
7: **Return** intention anomaly;
8: **else if** $\|\hat{K}_\infty - K_\infty\| \le \varepsilon_2$,
9: **Return** objective anomaly in Step 4 is false alarm and the observed behaviors are normal behaviors;
10: **end if**
11: **else if** $\|Q_\infty - \hat{Q}_\infty\| \le \varepsilon_1$, **then**
12: **Return** normal behaviors;
13: **end if**

where ε_1 and ε_2 are selected as 0.001 of $\|Q_\infty\|$ and $\|K_\infty\|$, respectively.

Abnormal

Algorithm 3 Inverse RL for Anomaly Detection and Correction Algorithm

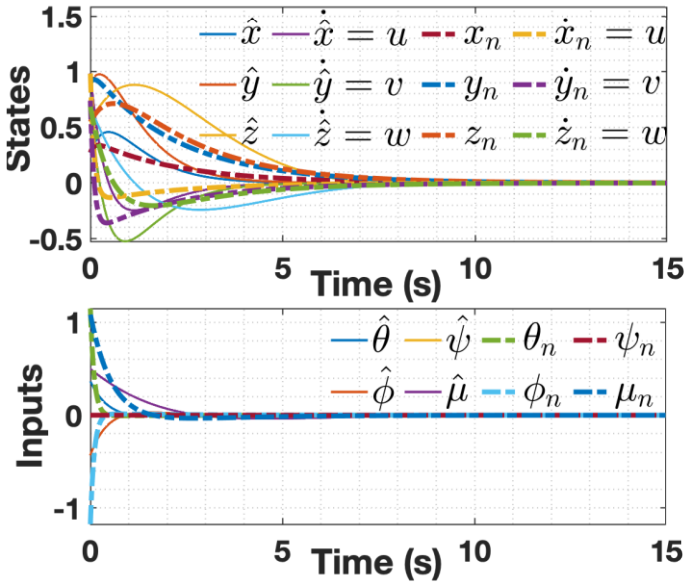
1: ***Run Algorithm 2 in Detection Phase to identify abnormal behaviors.***

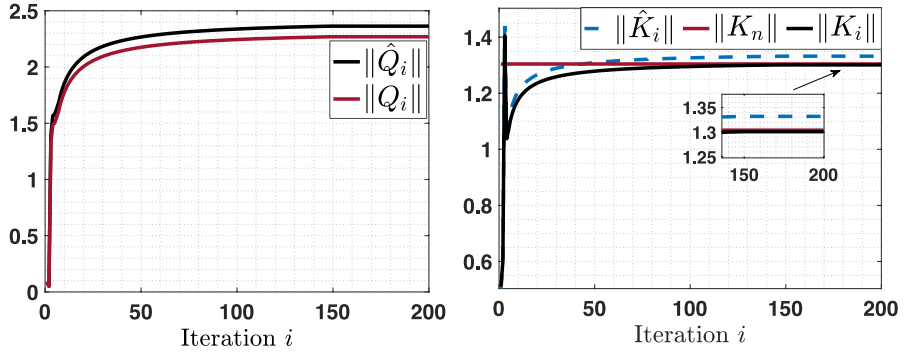
2: ***Correction Phase.***
Select initial $\tilde{Q}_0 = 0 \in \mathbb{R}^{k\times k}$, any $\tilde{R} = \tilde{R}^T \in \mathbb{R}^{m\times m} > 0$, and a stabilizing $\tilde{K}_0$. Set $j = 0$ and a threshold e_3.

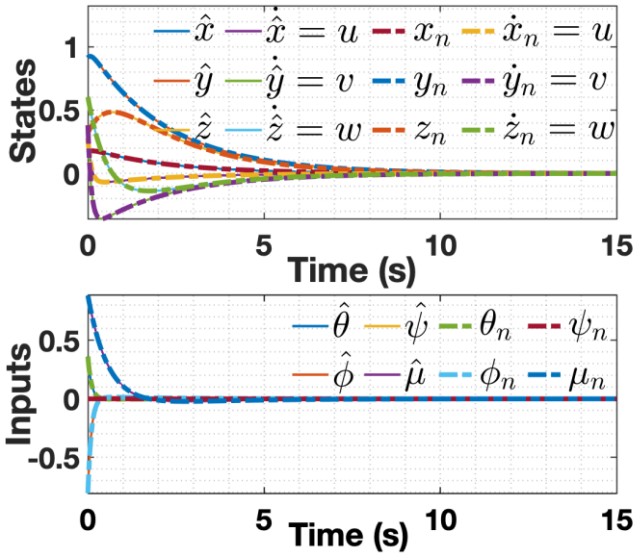
3: **Solve for** $\tilde{P}_j$, $\tilde{K}_{j+1}$ **and** $\tilde{Q}_{j+1}$ **from**

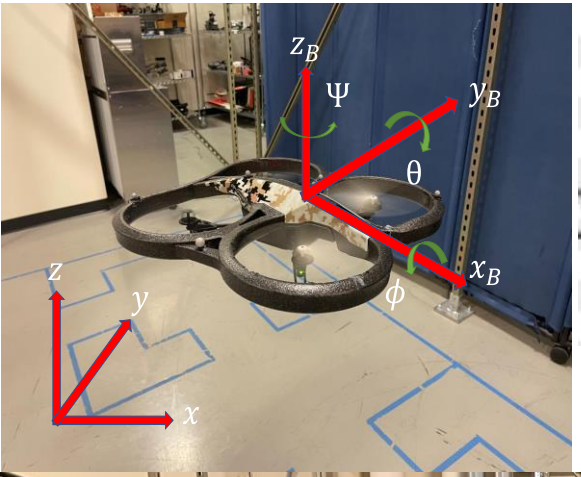
$$x_n^T(t+T)\tilde{P}_j x_n(t+T) - x_n^T(t)\tilde{P}_j x_n(t) + \hat{x}^T(t+T)\tilde{P}_j\hat{x}(t+T) - \hat{x}^T(t)\tilde{P}_j\hat{x}(t) - \int_t^{t+T} 2(u_n^* - \tilde{u}_{n_j})^T \tilde{R}\tilde{K}_{j+1} x_n d\tau - \int_t^{t+T} 2(\hat{u} - \tilde{u}_j)^T \tilde{R}\tilde{K}_{j+1}\hat{x} d\tau + \int_t^{t+T} \hat{x}^T \tilde{Q}_{j+1}\hat{x} d\tau = -\int_t^{t+T} \left((u_n^* - \tilde{u}_{n_j})^T \tilde{R}(u_n^* - \tilde{u}_{n_j}) + x_n^T \tilde{Q}_j x_n\right) d\tau - \int_t^{t+T} \left(\tilde{u}_j^T \tilde{R}\tilde{u}_j + \tilde{u}_{n_j}^T \tilde{R}\tilde{u}_{n_j}\right) d\tau.$$

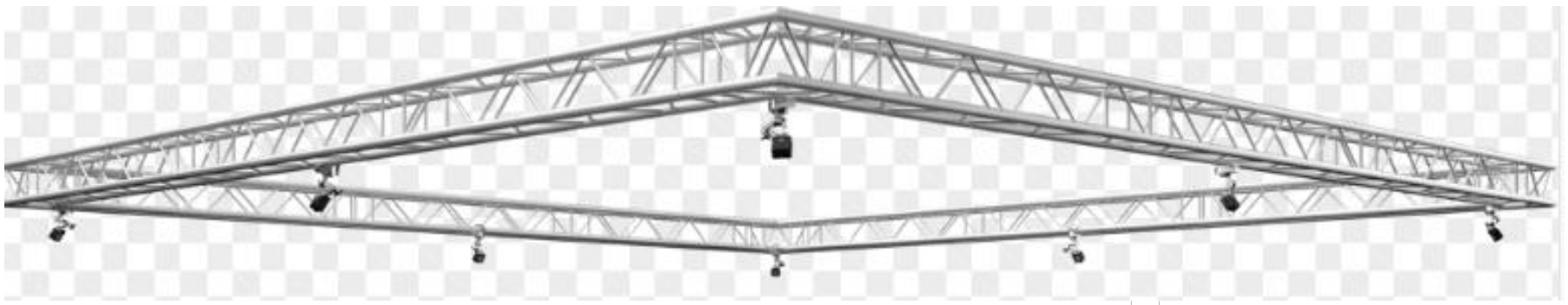
4: **Stop if** $\|\tilde{K}_j - \tilde{K}_{j-1}\| \le e_3$, otherwise, set $j \to j+1$.

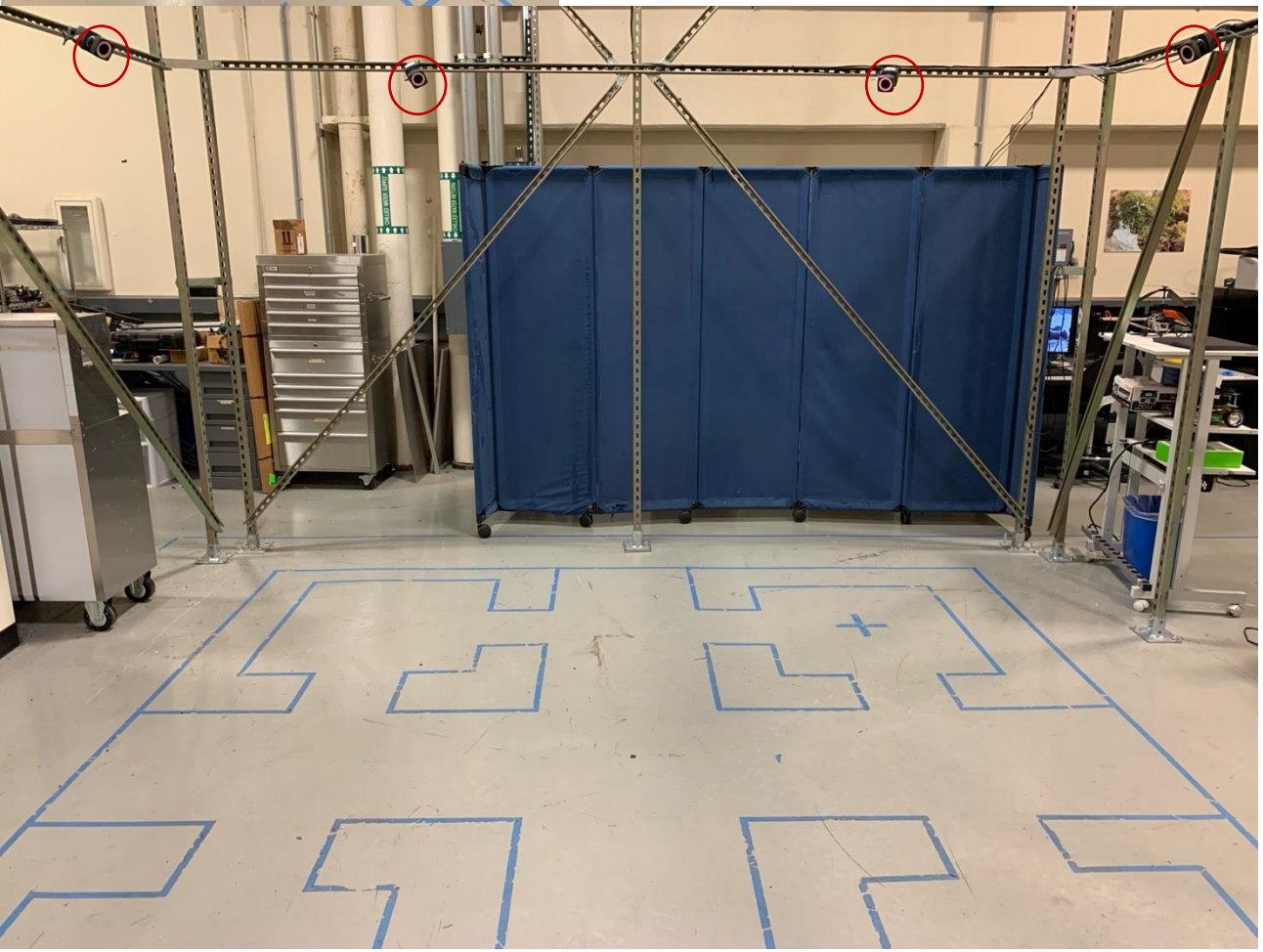


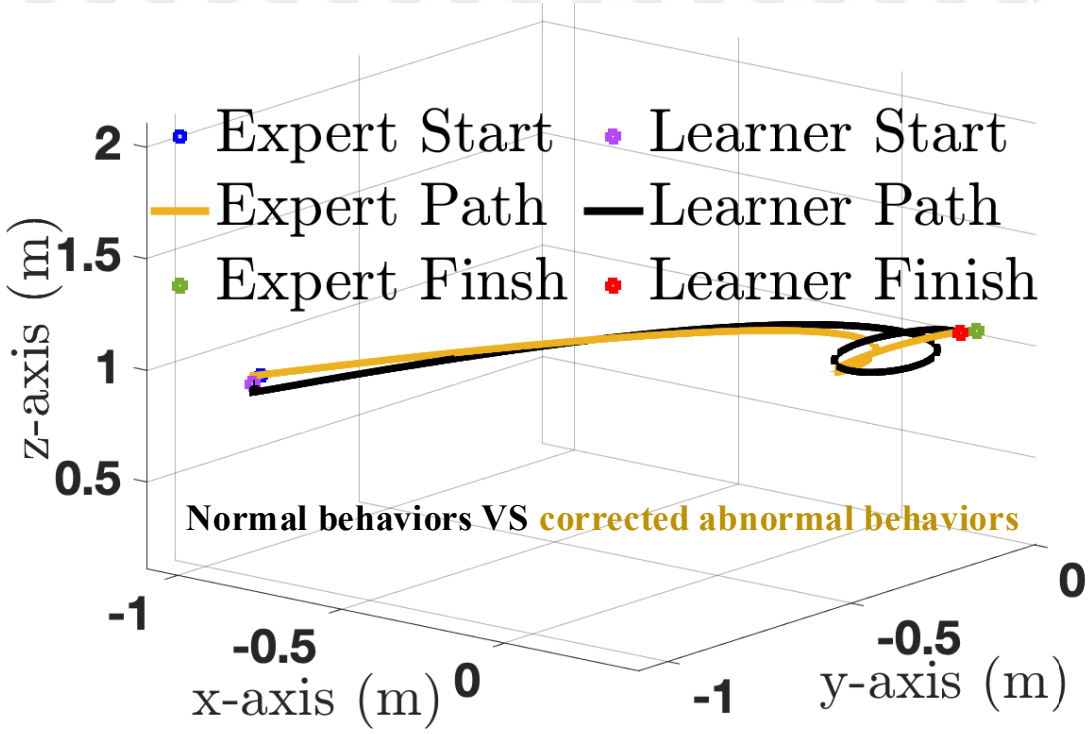


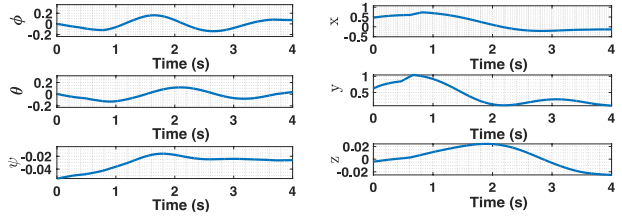






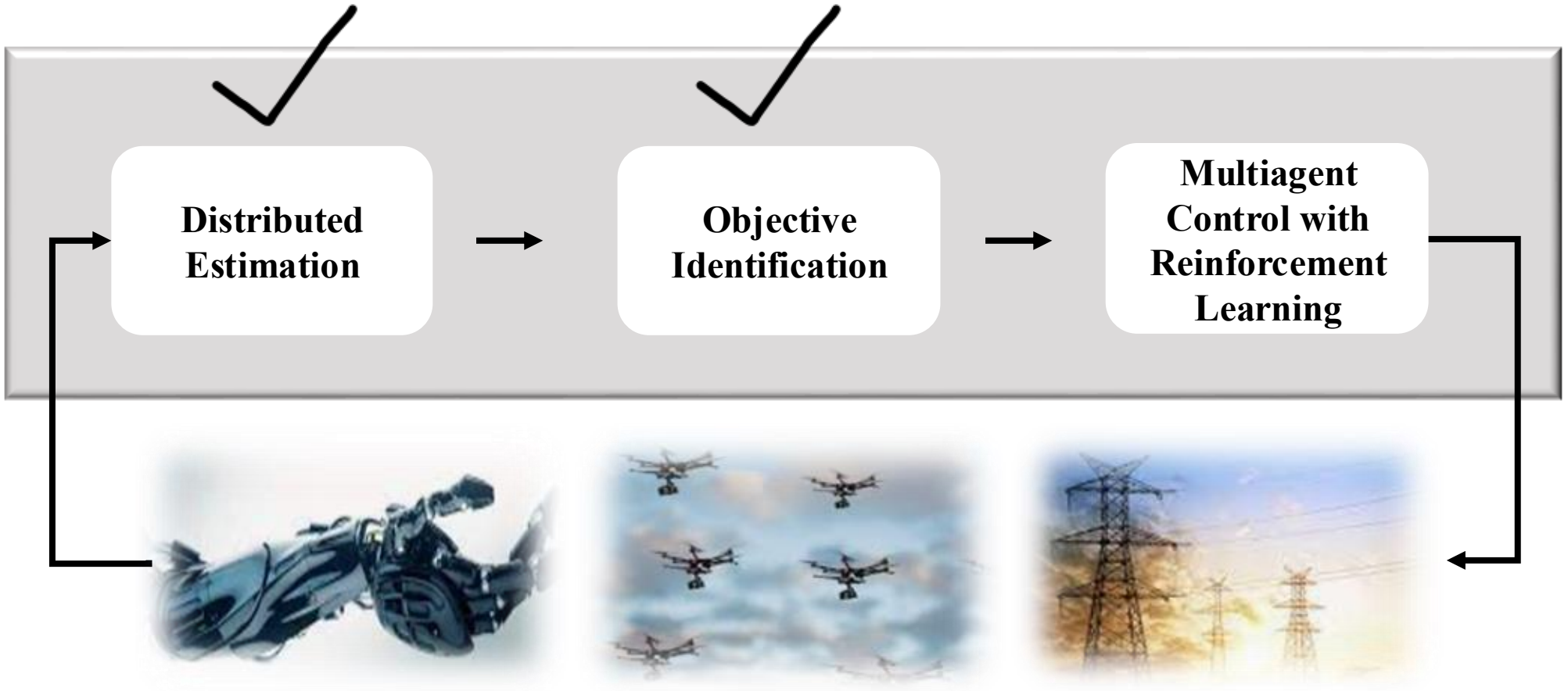






Euler angles

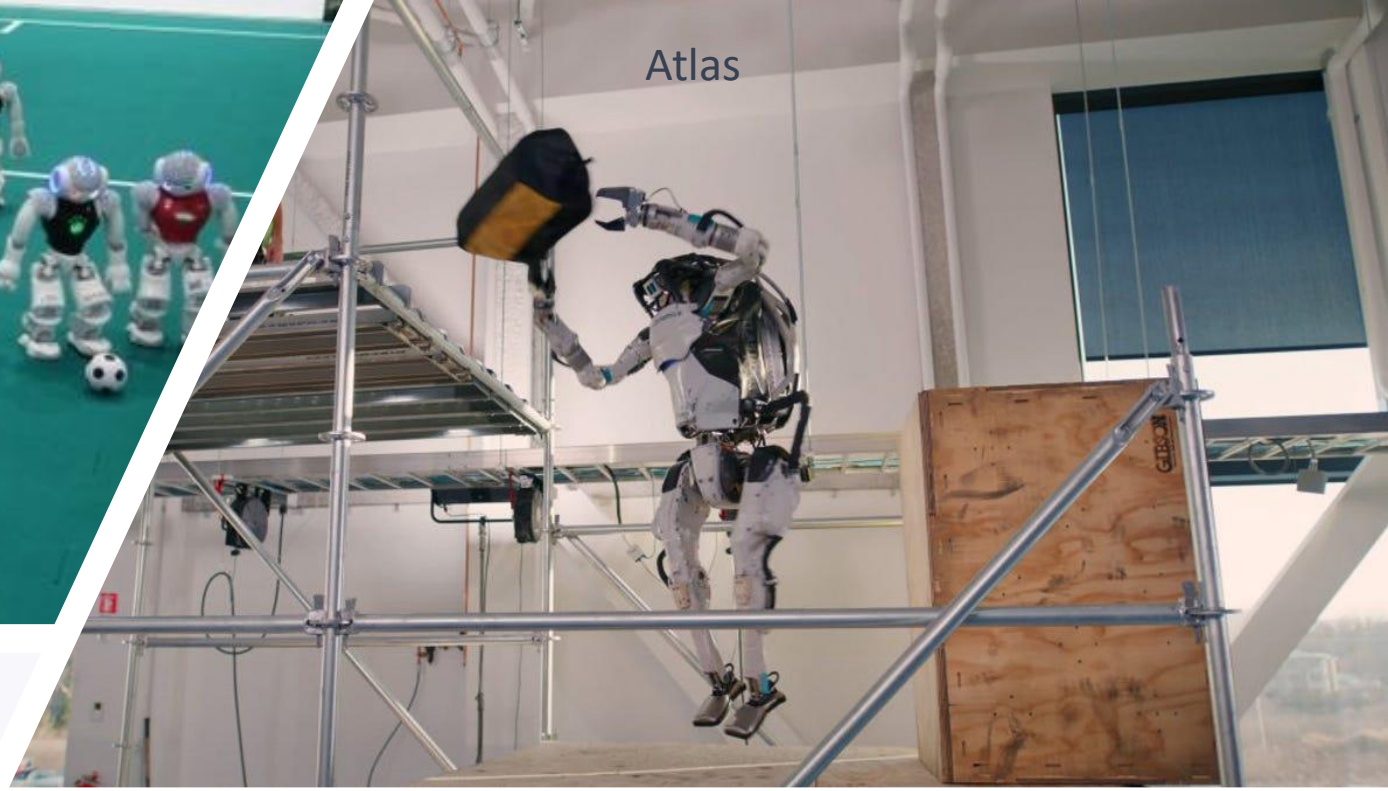
Cartesian Position

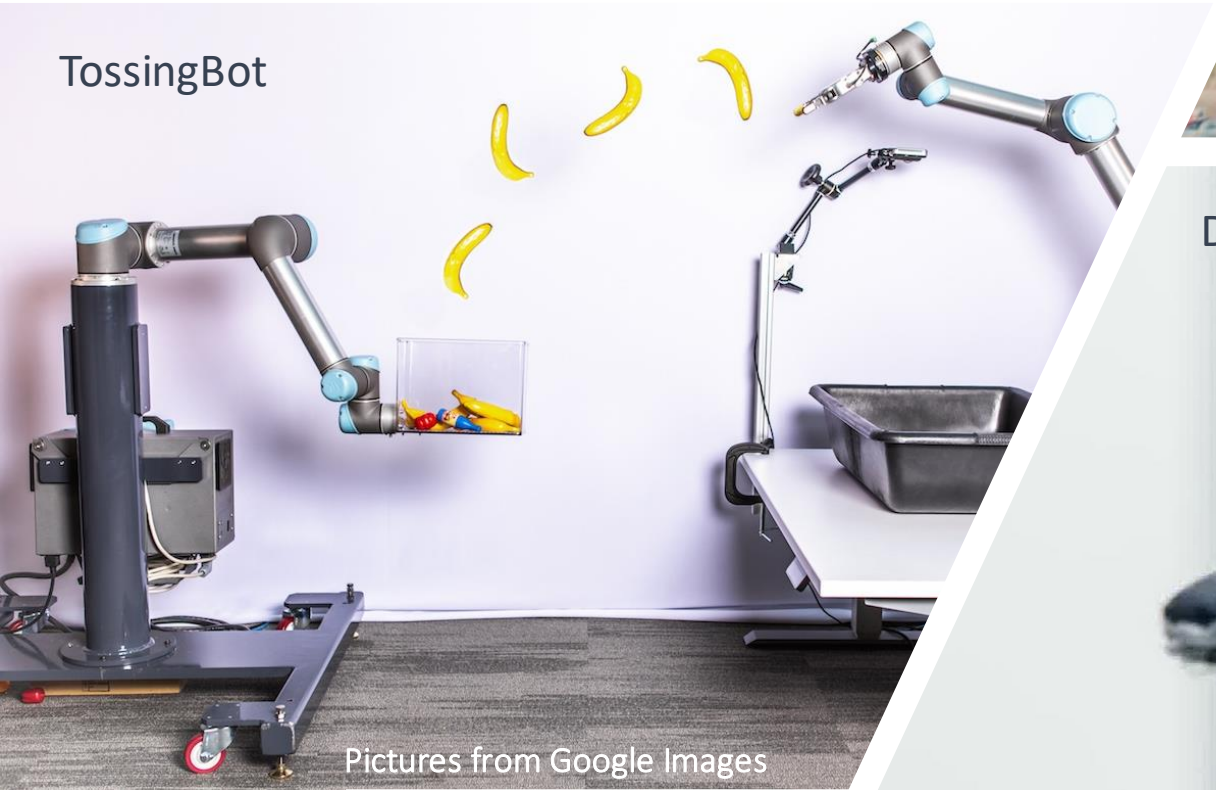

Distributed Estimation
Objective Identification
Multiagent Control with Reinforcement Learning

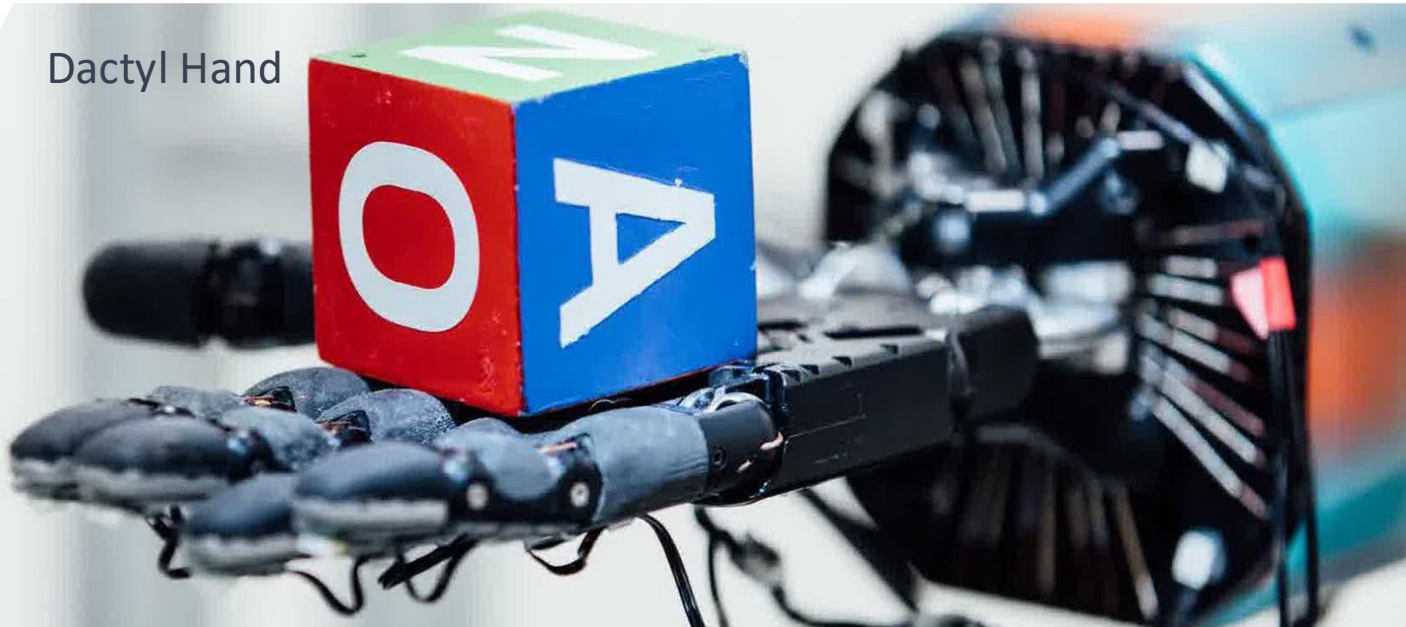
CONTENT

- Motivation
- Distributed Estimation with Distributed Kalman Filters
- Intention Identification with Inverse Reinforcement Learning
- **Multiagent Control with Reinforcement Learning**
- Future Work



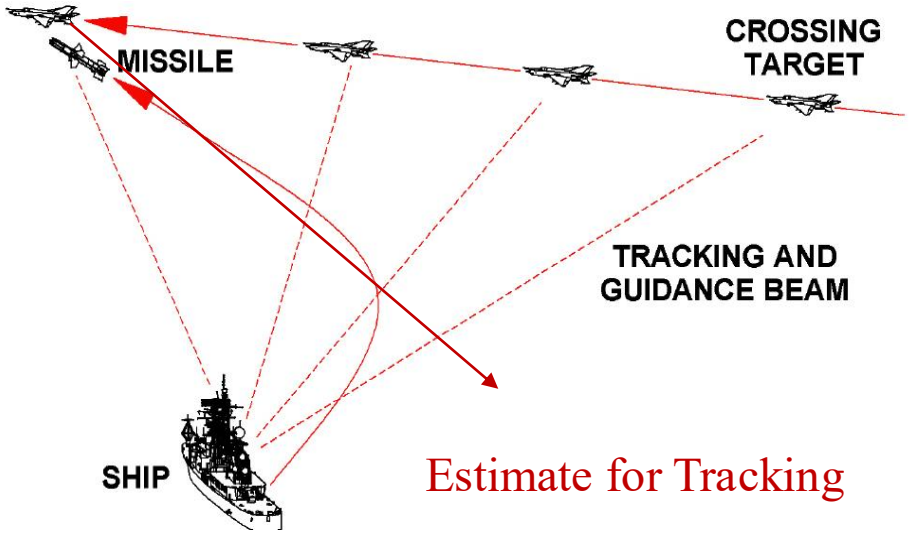




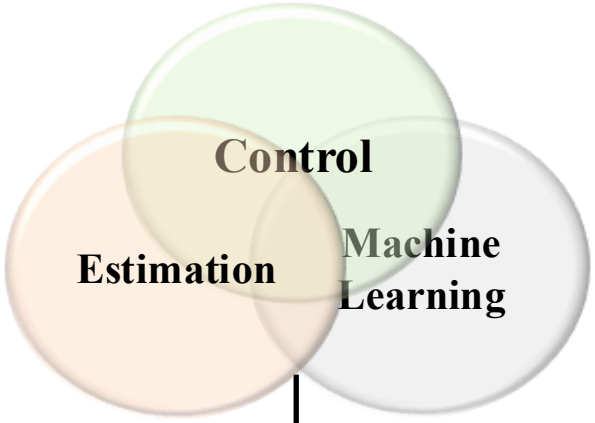


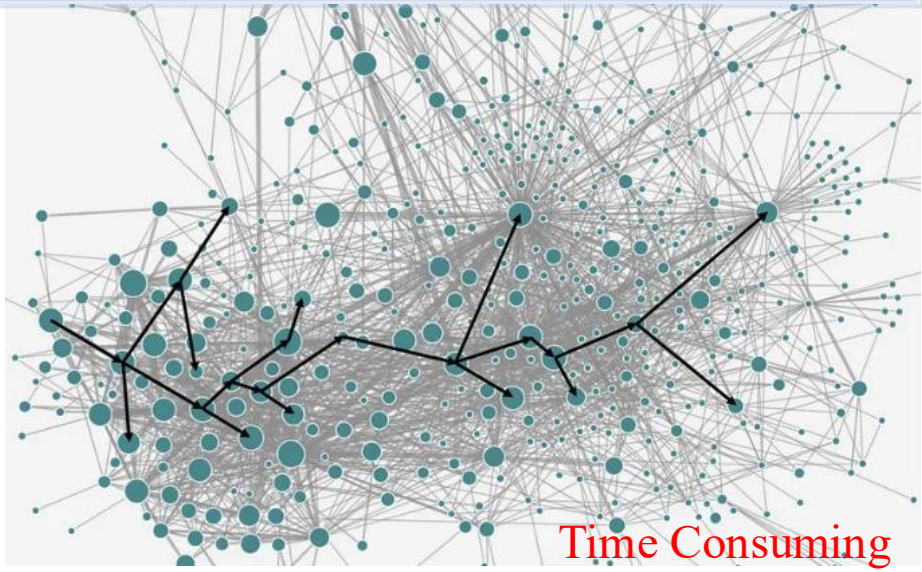
Pictures from Google Images

Challenges in Multiagent Control with Reinforcement Learning



Estimate for Tracking





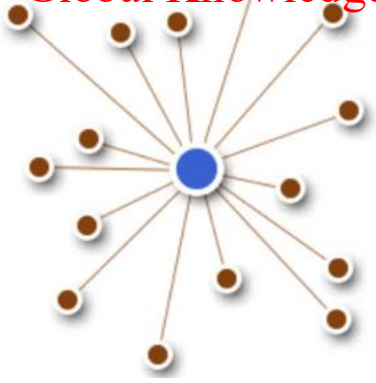
Time Consuming
Hard to Implement

1. Unavailable State Information

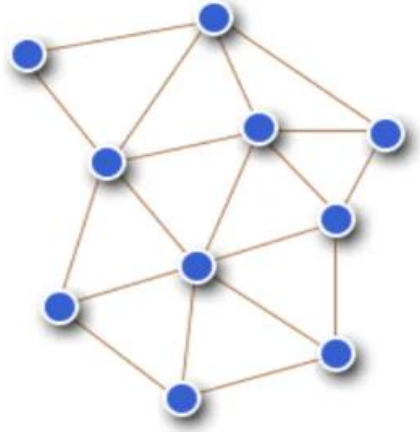
Multiagent Control with Reinforcement Learning

3. Large-scale Systems

Time Consuming
Global Knowledge



2. Non-distributed Solutions



To Address the Challenges

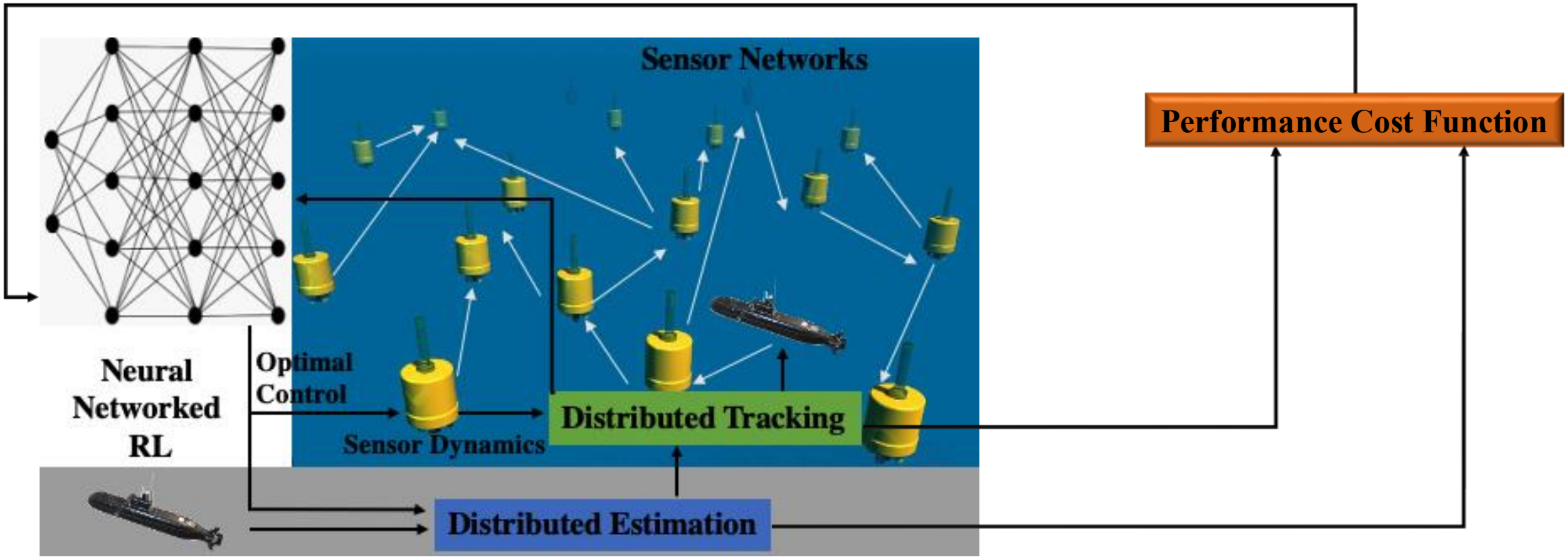
Distributed Estimation-based Tracking Control with Reinforcement Learning → **1. Unavailable State Information**

Distributed Control for Multiplayer Games with Reinforcement Learning → **2. Distributed Solutions**

Accelerated Reinforcement Learning for Large-scale Multiagent Systems → **3. Scale-free Systems**

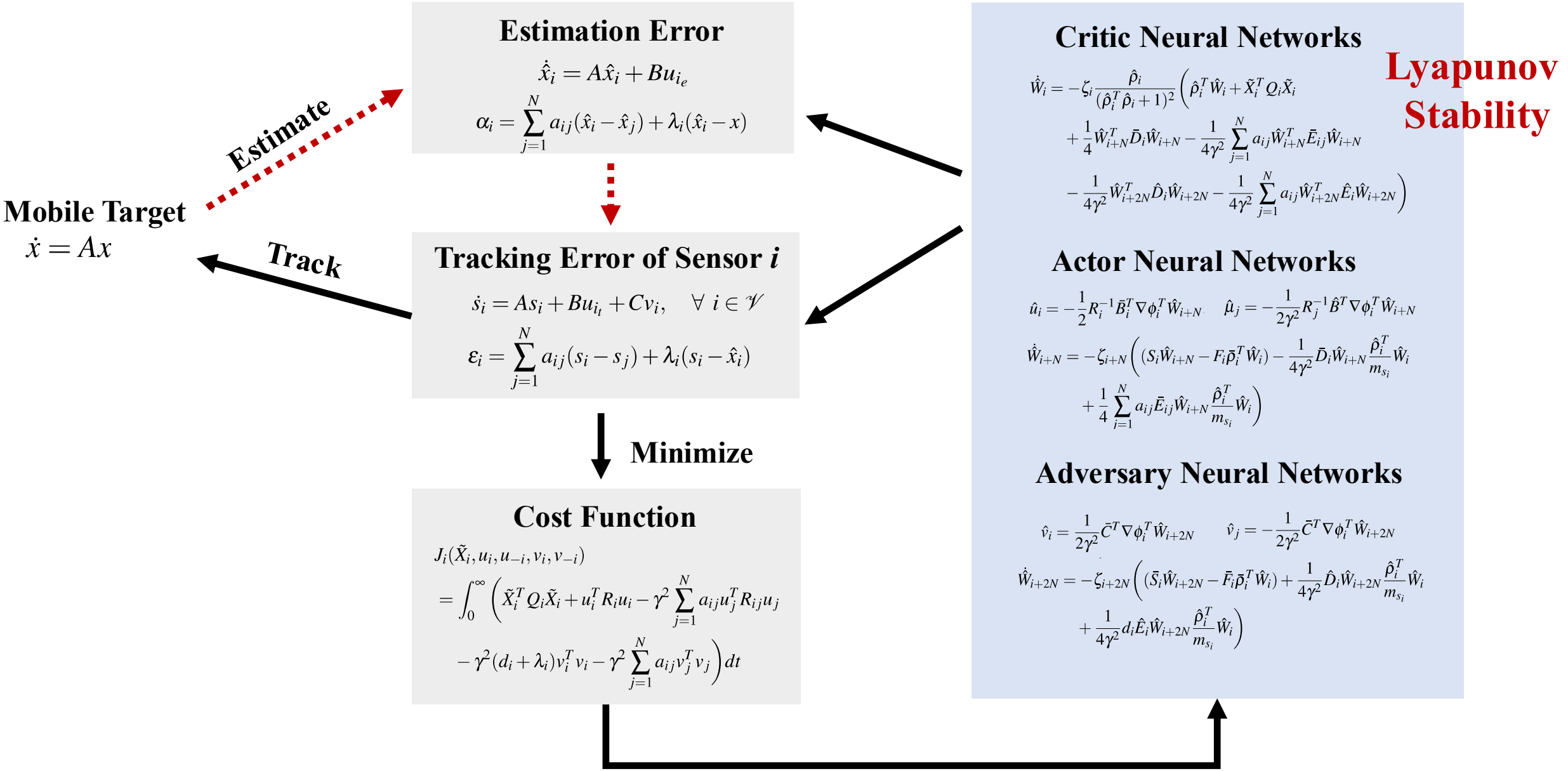
Distributed Estimation Based Tracking Control of Multiagent Sensor Networks with Reinforcement Learning

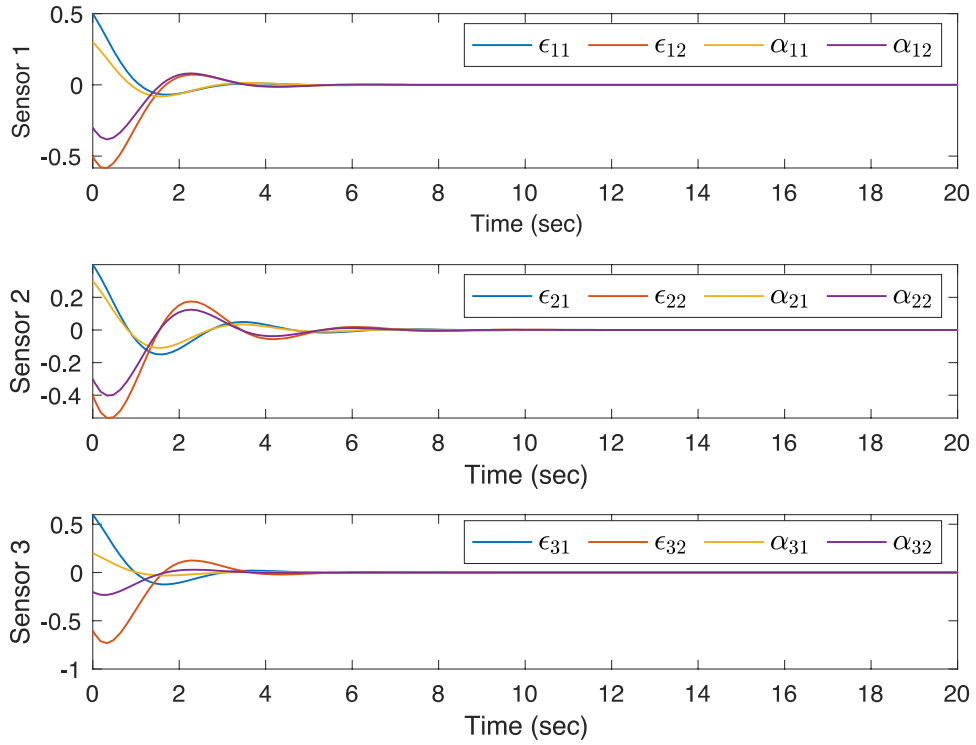
Funded by ONR grant "Optimal Design for Assured Performance of Interactive Multibody Systems: Guaranteed Controls for Multi-pursuers, Estimation, Optimal Learning, Scalable Uncertainty Sampling, and Time-critical Communication", $815,019.00.



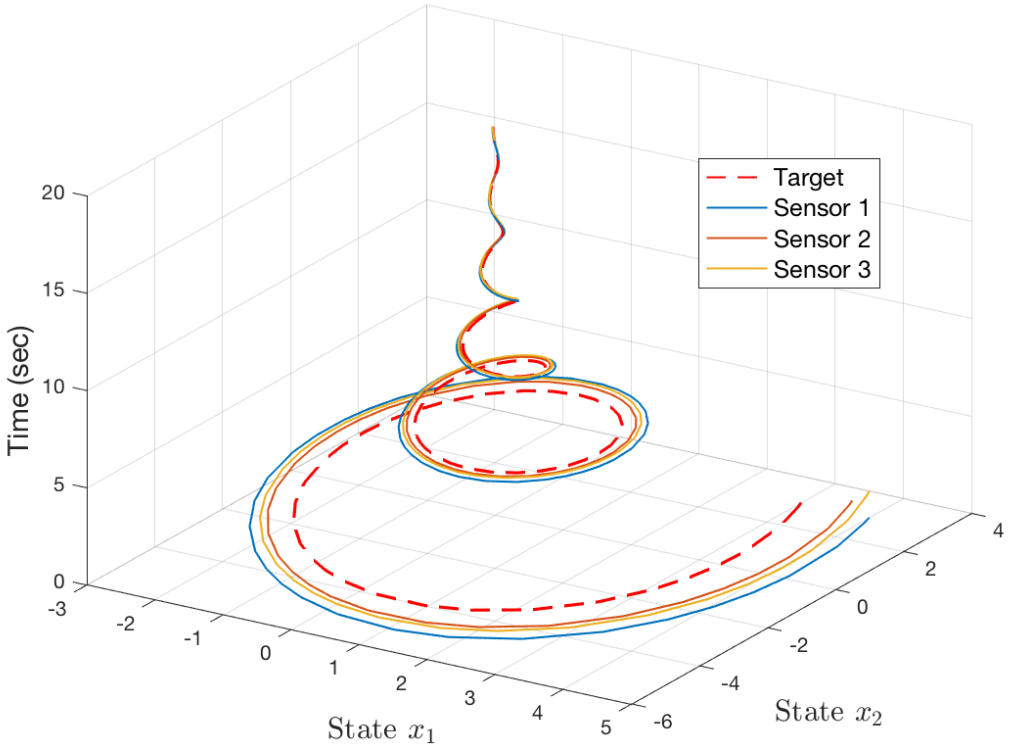
Lian, B., et al (2022). Online Learning of Minmax Solutions for Distributed Estimation and Tracking Control of Sensor Networks in Graphical Games. *IEEE Transactions on Control of Network Systems*.

Realtime Distributed Adversary-Actor-Critic Neural Networks



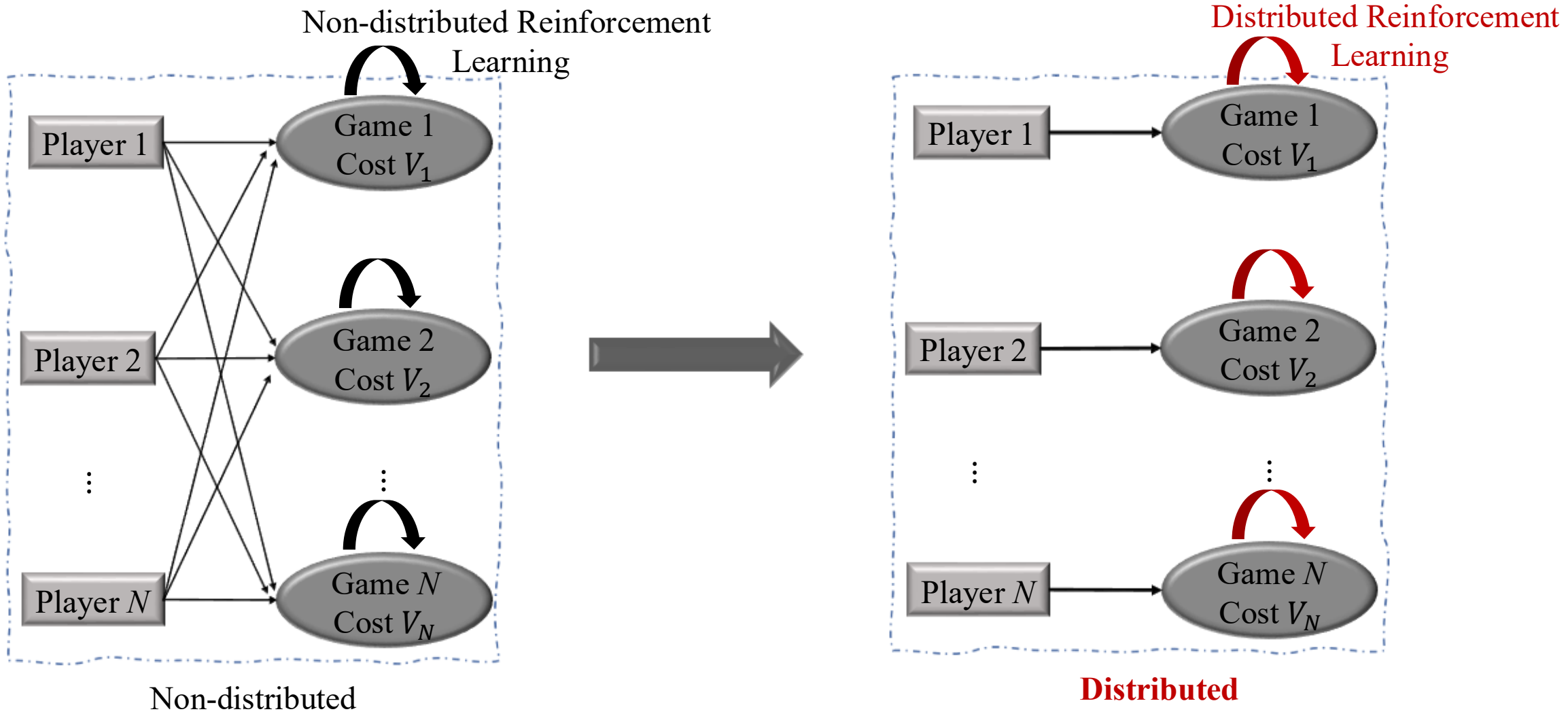


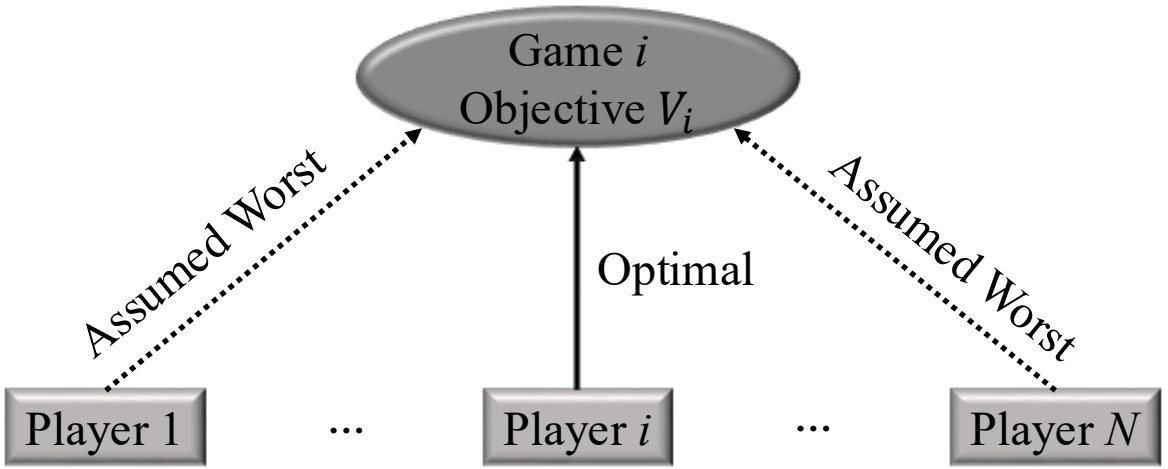
Estimation Errors and Tracking Errors Go to 0



3 Sensors Track the Target

Distributed Control for Noncooperative Multiplayer Games with Reinforcement Learning





Distributed Minimax Control Strategy

Distributed Model-free Reinforcement Learning

Algorithm 2 Data-driven Off-policy RL Algorithm to solve the Distributed Minmax Strategies in Multiplayer Games

1) Select initial stabilizing control policies K_i^0 and $K_{ij}^0 = 0$ for $i \in \mathcal{N}$, $j \in -i$. Set small thresholds e_i and $h = 0$. Apply $u_i = -K_i x + \varepsilon_i$ to (51), where K_i is a fixed stabilizing policy and ε_i is a small probing noise.
2) **Solve control feedback gains K_i^{h+1} and K_{ij}^{h+1} by**

$$
\begin{aligned}
&x^T(t+\Delta t)P_i^h x(t+\Delta t) - x^T(t)P_i^h x(t) \\
&\quad - \int_t^{t+\Delta t} 2(u_i - u_i^h)^T R_{ii} K_i^{h+1} x d\tau \\
&\quad + \int_t^{t+\Delta t} 2\gamma_i^2 \sum_{j\in -i} (u_j - v_{ij}^h)^T R_{ij} K_{ij}^{h+1} x d\tau \\
&= -\int_t^{t+\Delta t} \left(x^T Q_i x + (u_i^h)^T R_{ii} u_i^h - \gamma_i^2 \sum_{j\in -i} (v_{ij}^h)^T R_{ij} v_{ij}^h \right) d\tau.
\end{aligned}
$$

3) **Update control inputs u_i^{h+1} and v_{ij}^{h+1} by**

$$
\begin{aligned}
u_i^{h+1} &= -K_i^{h+1} x, \\
v_{ij}^{h+1} &= -K_{ij}^{h+1} x.
\end{aligned}
$$

4) **Stop if** $\|P_i^{h+1} - P_i^h\| \le e_i$, $\forall\, i \in \mathcal{N}$. Otherwise, set $h \to h+1$, and go to Step 2.

Asymptotic Stability

Distributed minimax strategy

$$u_i^* \triangleq \arg\min_{u_i} \max_{u_{-i}} V_i(x, u_i, u_{-i})$$

Lyapunov Theory

Multiplayer dynamic systems

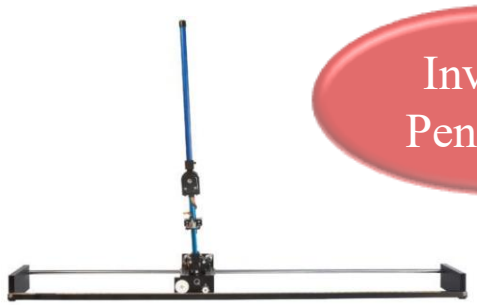
$$\dot{x} = \left(A - \sum_{j=1}^{N} B_j R_{jj}^{-1} B_j^T P_j \right) x$$

$$\lim_{t \to \infty} x(t) = 0$$

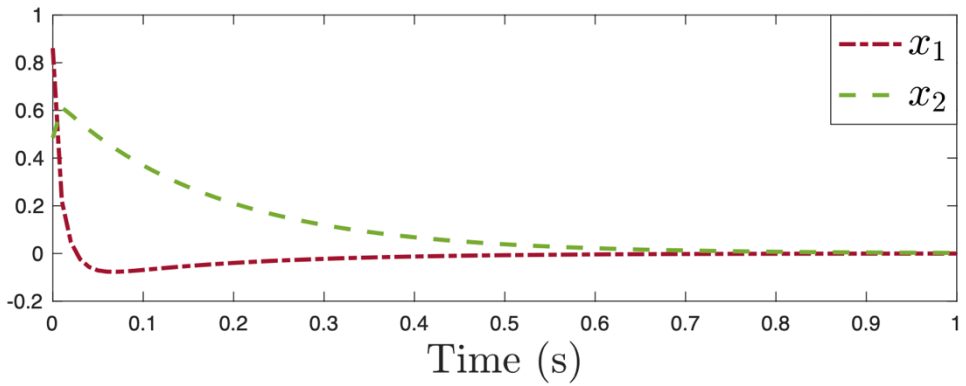
Improved Robustness

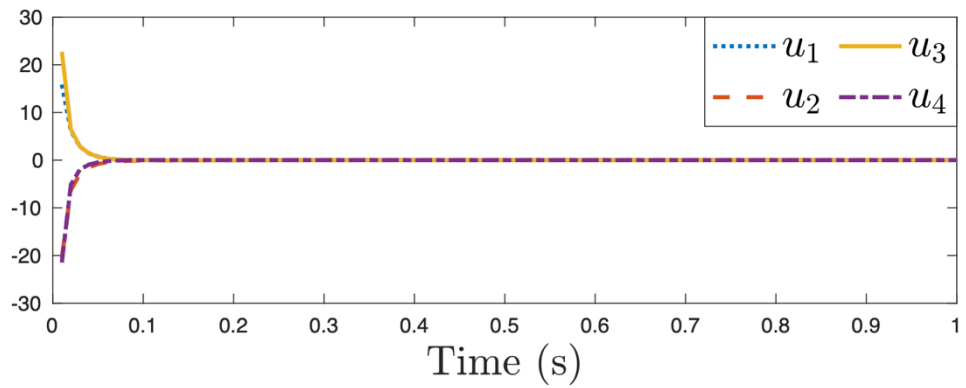
- Phase margin: $|PM| < 60^\circ + \omega_i$
- Gain margin: $\frac{a}{2} < GM < \infty$

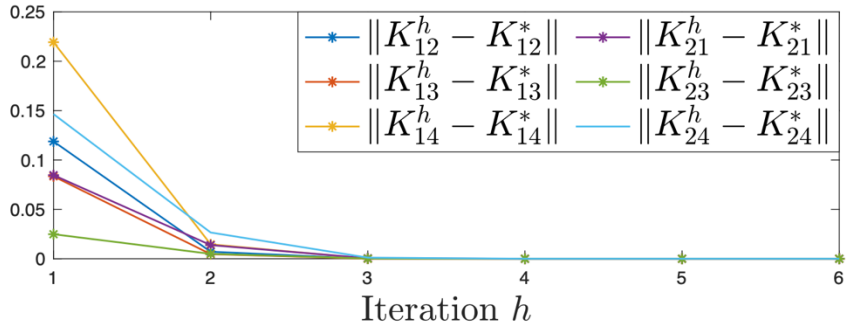
where $\omega_i > 0$ and $0 < a < 1$.

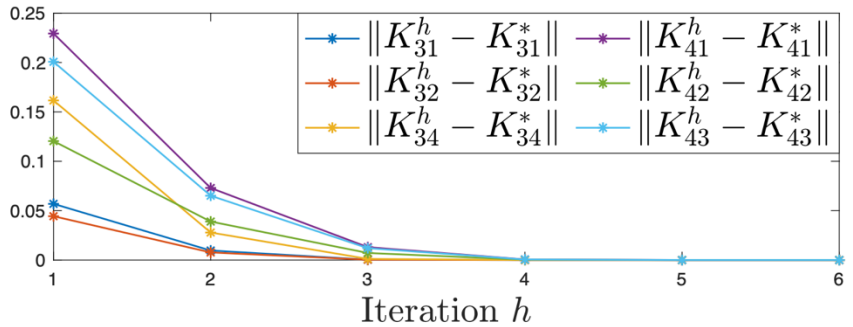


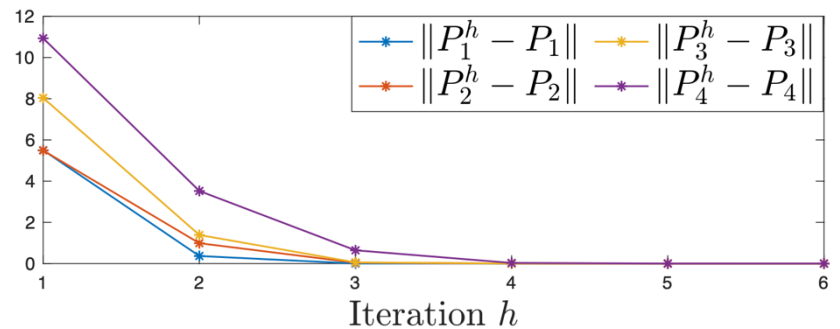
Inverted Pendulum

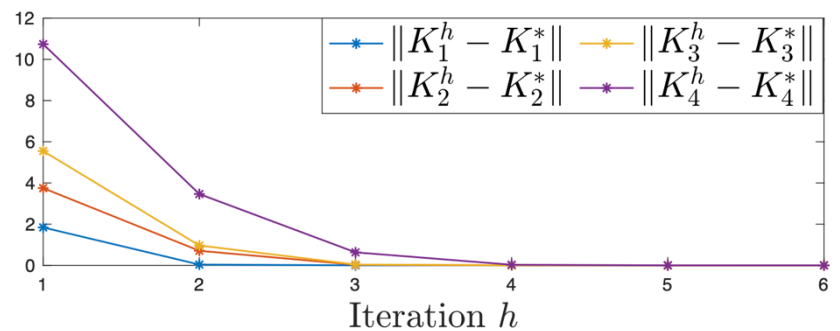




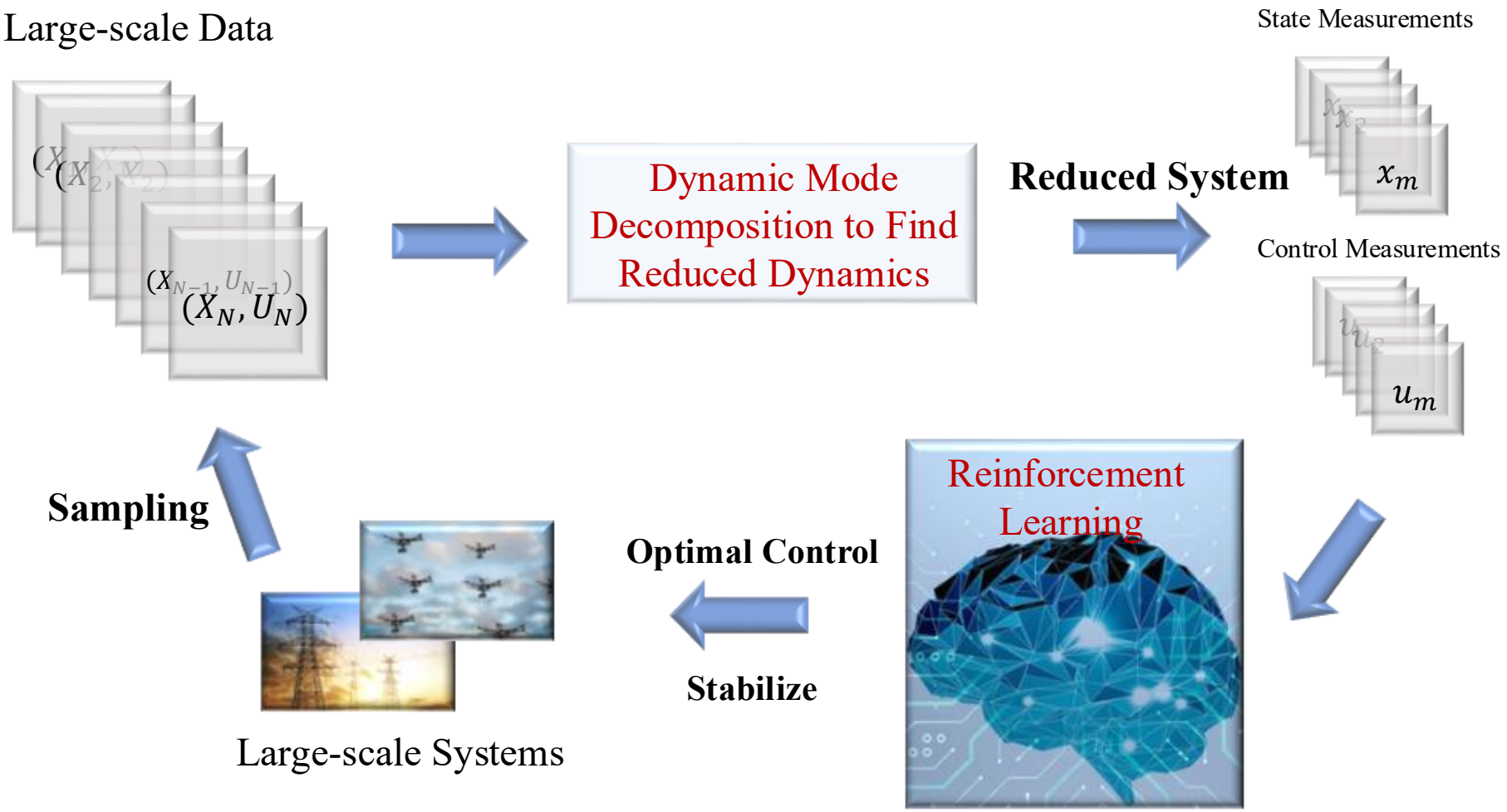








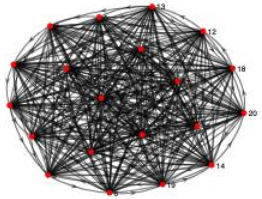
Accelerated Reinforcement Learning for Large-Scale Multiagent Systems



Donge, V. S., **Lian, B.**, Lewis, F. L., & Davoudi, A. (2023). Accelerated Reinforcement Learning via Dynamic Mode Decomposition. *IEEE Transactions on Control of Network Systems*.

Low-Scale Subsystems
$x_{i_k} \in \mathbb{R}^n, u_{i_k} \in \mathbb{R}^m$

$$x_{i_{k+1}} = A_{ii}x_{i_k} + A_{ij}\sum_{j\neq i,j=1}^{M} e_{ij}(x_{i_k} - x_{j_k}) + B_i u_{i_k}, \quad i \in \mathcal{M}$$



***M* agents**

Large-Scale Global Systems
$\tilde{x}_k \in \mathbb{R}^{nM}, \tilde{u}_k \in \mathbb{R}^{mM}$

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}\tilde{u}_k$$

$$J(\tilde{u}_k, \tilde{x}_k) = \sum_{k=0}^{\infty} [\tilde{x}_k^{\mathrm{T}}\tilde{Q}\tilde{x}_k + \tilde{u}_k^{\mathrm{T}}\tilde{R}\tilde{u}_k]$$

$r << nM$

Reduced Scale Global Systems
$\tilde{\xi}_k \in \mathbb{R}^r$

$$\xi_{k+1} = Y\tilde{A}Y^{\dagger}\xi_k + Y\tilde{B}\tilde{u}_k$$

$$\bar{J}(\tilde{u}_k, \xi_k) = \sum_{k=0}^{\infty} [\xi_k^{\mathrm{T}}\tilde{Q}_r\xi_k + \tilde{u}_k^{\mathrm{T}}R\tilde{u}_k]$$

Dynamic Mode Decomposition

$$\xi_k \approx Y\tilde{x}_k$$

$$\theta = \frac{1}{\lambda}X_2\tilde{V}\tilde{S}^{-1}v$$

$$\tilde{T}v = X_2\tilde{V}\tilde{S}^{-1}\tilde{U}^{\mathrm{t}}v = YX_2\tilde{V}\tilde{S}^{-1}v = \bar{T}v = \lambda v.$$

$$X_1 = \begin{bmatrix} | & | & \cdots & | \\ x_1 & x_2 & \cdots & x_{p-1} \\ | & | & \cdots & | \end{bmatrix}, \quad X_2 = \begin{bmatrix} | & | & \cdots & | \\ x_2 & x_3 & \cdots & x_p \\ | & | & \cdots & | \end{bmatrix}$$

Sampling

Algorithm 2 Dynamic Mode Decomposition-based Model-free Reinforcement Learning for Large-scale Systems

1: **Initialization**: Given N, set $j = 0$ and small threshold e. Select stabilizing $\tilde{K}_r^0$, $\tilde{K}^0$, and control input $\tilde{u}_k = -\tilde{K}\tilde{x}_k + \varepsilon_k$, where ε_k is a probing noise.

2: **Data Collection**: Same as step 2 in Algorithm 1 and compute (20).

3: **DMD Preconditioning**: Set strict threshold μ. Compute reduced-order data vectors $(\hat{\rho}_{xx}, \hat{\rho}_{xu}, \hat{\rho}_{uu})$ in (41) and operators (φ_r^j, ψ_r^j) given by

$$\varphi_r^j = \begin{bmatrix} \xi_k^{\mathrm{T}}\tilde{Q}_r\xi_k + \xi_k^{\mathrm{T}}(\tilde{K}_r^j)^{\mathrm{T}}\tilde{R}\tilde{K}_r^j\xi_k \\ \vdots \\ \xi_{k+\zeta-1}^{\mathrm{T}}\tilde{Q}_r\xi_{k+\zeta-1} + \xi_{k+\zeta-1}^{\mathrm{T}}(\tilde{K}_r^j)^{\mathrm{T}}\tilde{R}\tilde{K}_r^j\xi_{k+\zeta-1} \end{bmatrix},$$

$$\psi_r^j = [\hat{\rho}_{xx} \quad \hat{\rho}_{xu} \quad \hat{\rho}_{uu}],$$

$$\hat{\rho}_{xx} = \rho_{xx_s}(\theta \otimes \theta) \in \mathbb{R}^{1\times r^2},$$
$$\hat{\rho}_{xu} = \rho_{xu_s}(I_{mM} \otimes \theta) \in \mathbb{R}^{1\times r(mM)},$$
$$\hat{\rho}_{uu} = \rho_{uu_s}(I_{mM} \otimes I_{mM}) \in \mathbb{R}^{1\times (mM)^2}.$$

4: **Policy Evaluation**: Compute $\tilde{P}_r^{j+1}$ by

$$((\psi_r^j)^{\mathrm{T}}\psi_r^j)^{-1}(\psi_r^j)^{\mathrm{T}}\varphi_r^j = \begin{bmatrix} \mathrm{vec}(\tilde{P}_r^{j+1})^{\mathrm{T}} \\ \mathrm{vec}((Y\tilde{B})^{\mathrm{T}}\tilde{P}_r^{j+1}Y\tilde{A}Y^{\dagger})^{\mathrm{T}} \\ \mathrm{vec}((Y\tilde{B})^{\mathrm{T}}\tilde{P}_r^{j+1}Y\tilde{B})^{\mathrm{T}} \end{bmatrix}.$$

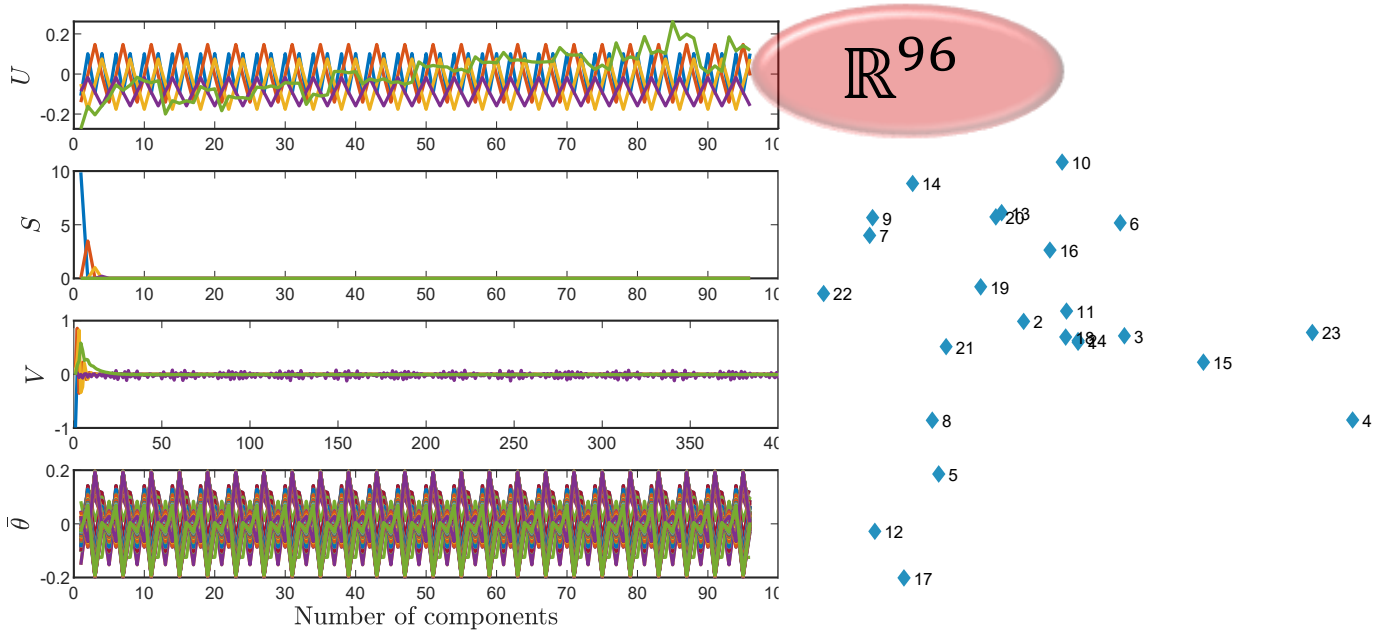
5: **Policy Improvement**: Compute $\tilde{K}_r^{j+1}$ by

$$\tilde{K}_r^{j+1} = (R + (Y\tilde{B})^{\mathrm{T}}\tilde{P}_r^{j+1}Y\tilde{B})^{-1}(Y\tilde{B})^{\mathrm{T}}\tilde{P}_r^{j+1}A.$$

6: **Stop if** $\|\tilde{K}_r^{j+1} - \tilde{K}_r^j\| \leq e$; Otherwise, set $j = j+1$, and go to step 2.

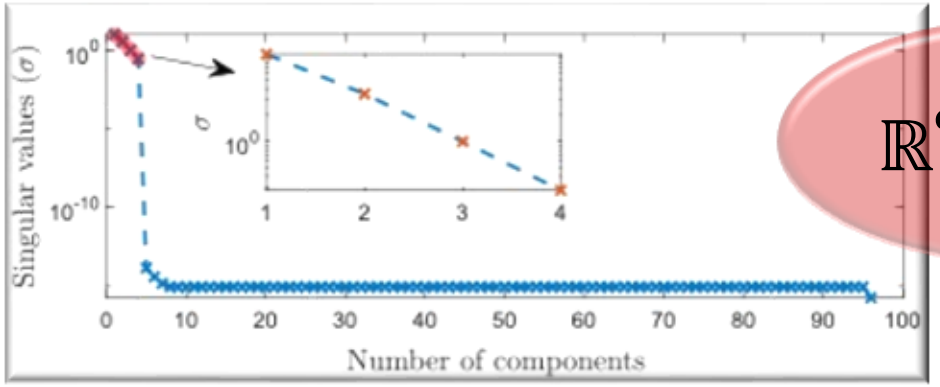
7: **Collect**:

$$\tilde{K}^{j+1} = Y\tilde{K}_r^{j+1}.$$



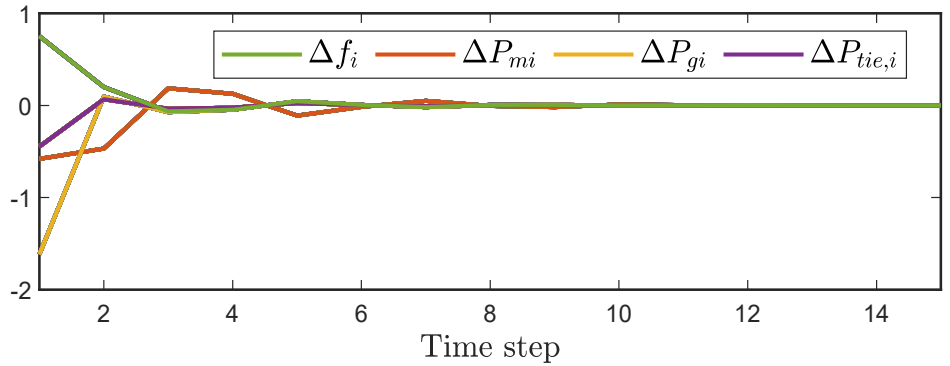
Total dynamic modes of **24**-area power system

Dynamic Mode Decomposition

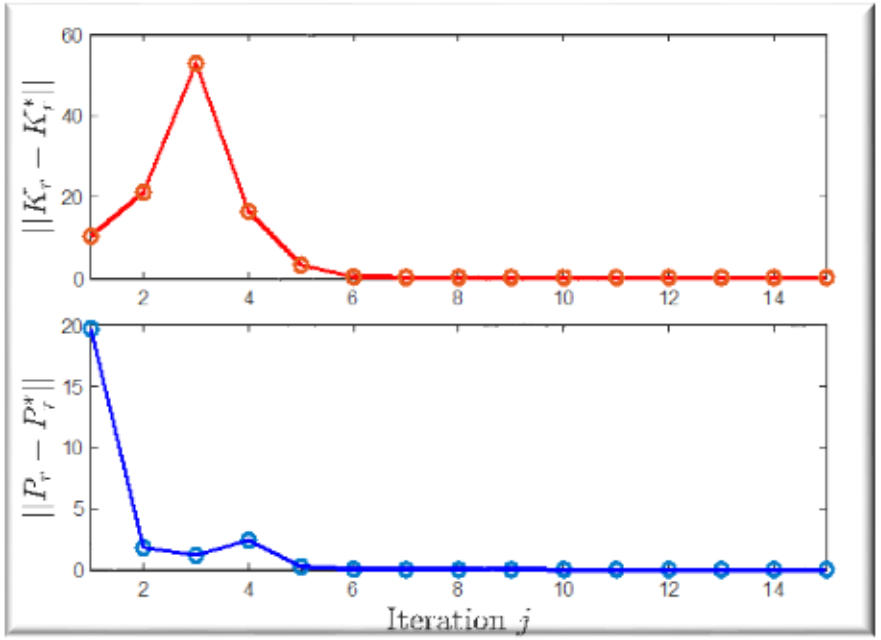


4 dominant singular values

Reinforcement Learning



Trajectories of frequency, generator output, governor valve, and tie-line power of *i*-th area system



Convergence of trained parameters to the desired ones

Selected Publications

➢ Distributed Estimation

[1] **Lian, B**., Wan, Y., Lewis, F. L., & Chai, T. (2020). Distributed Kalman consensus filter for estimation with moving targets. *IEEE Transactions on Cybernetics*.
[2] **Lian, B**., Lewis, F. L., Hewer, G. A., Estabridis, K., & Chai, T. (2021). Robustness analysis of distributed Kalman filter for estimation in sensor networks. *IEEE Transactions on Cybernetics*.

➢ Inverse Reinforcement Learning for Optimal Control Systems and Games

[1] **Lian, B**., Xue, W., Lewis, F. L., & Chai, T. (2021). Robust inverse Q-learning for continuous-time linear systems in adversarial environments. *IEEE Transactions on Cybernetics.*
[2] Xue, W., **Lian, B**., Fan, J., Kolaric, P., Chai, T., & Lewis, F. L. (2021). Inverse reinforcement Q-learning through expert imitation for discrete-time systems. *IEEE Transactions on Neural Networks and Learning Systems*.
[3] **Lian, B**., Xue, W., Lewis, F. L., & Chai, T. (2022). Inverse reinforcement learning for multi-player noncooperative apprentice games. *Automatica*.
[4] **Lian, B**., Donge, V. S., Lewis, F. L., Chai, T., & Davoudi, A. (2022). Data-driven inverse reinforcement learning control for linear multiplayer games. *IEEE Transactions on Neural Networks and Learning Systems*.
[5] Donge, V. S., **Lian, B**., Lewis, F. L., & Davoudi, A. (2022). Multi-agent graphical games with inverse reinforcement learning. *IEEE Transactions on Control of Network Systems*.
[6] **Lian, B**., Kartal, Y., Lewis, F. L., Mikulski, D. G., Hudas, G. R., & Davoudi, A. (2022). Anomaly detection and correction of optimizing autonomous systems with inverse reinforcement learning. *IEEE Transactions on Cybernetics*.
[7] Xue, W., **Lian, B**., Fan, J., Chai, T., & Lewis, F. L. (2023). Inverse reinforcement learning for trajectory imitation using static output feedback control. *IEEE Transactions on Cybernetics*.

➢ Reinforcement Learning based Multiagent Control

[1] **Lian B**., Lewis F. L., Hewer G. A., Estabridis K., Chai, T. (2022). Online learning of minmax solutions for distributed estimation and tracking control of sensor networks in graphical games. *IEEE Transactions on Control of Network Systems*.
[2] **Lian B**., Donge V. S., Xue W., Lewis F. L., Davoudi A. (2022). Distributed minmax strategy for multiplayer games: stability, robustness, and algorithms. *IEEE Transactions on Neural Networks and Learning Systems*.
[3] Donge V. S., **Lian B**., Lewis F. L., Davoudi A. (2023). Accelerated reinforcement learning via dynamic mode decomposition. *IEEE Transactions on Control of Network Systems*.

➢ Networked Control Systems & Clusters

[1] **Lian B.**, Zhang Q., Li J. (2018). Integrated sliding mode control and neural networks-based packet disordering prediction for nonlinear networked control systems. *IEEE Transactions on Neural Networks and Learning Systems*.

[2] Zhang X., **Lian B.**, Lewis F. L., Wan Y., Cheng, D. (2021). Directed graph clustering algorithms, topology, and weak links. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

➢ Books

[1] **Lian B.**, Xue W., Lewis F.L., Modares H., Kiumarsi B. *Integral And Inverse Reinforcement Learning For Optimal Control Systems And Games*, Springer, accepted to appear in October 2023.

➢ Under Review

[1] **Lian B.**, Xue W., Kolaric P., Lewis F.L., Davoudi A. (2021). Inverse value iteration and Q-learning: algorithms and stability, *IEEE Transactions on Automatic Control*.

[2] **Lian B.**, Xue W., Xie Y., Lewis F.L., Davoudi A. (2022). Off-policy inverse Q-learning for discrete-time antagonistic unknown systems, *Automatica*.

[3] **Lian B.,** Xue W., Lewis F.L., Davoudi A. (2022). Nash-minmax strategy for multiplayer multiagent graphical games with reinforcement learning, *IEEE Transactions on Control of Network Systems*.

[4] **Lian B.**, Xue W., Lewis F.L., Davoudi A. (2022). Inverse static output-feedback control learning for discrete-time systems, *IEEE Transactions on Cybernetics.*

[5] **Lian B.**, Koru A.T., Xue W., Lewis F.L., Davoudi A. (2023). Distributed dynamic cluster formation for consensus of multiagent systems, *IEEE Transactions on Automatic Control*.

[6] **Lian B.**, Jiang Y., Xue W., Lewis F.L. (2023). Stochastic optimal control for networked control systems with packet disordering using reinforcement learning, *Automatica*.

[7] Xue W., **Lian B.**, Fan J., Chai T., Lewis F.L. (2023). Model-free H∞ inverse reinforcement learning for expert-learner zero-sum games, *SIAM Journal on Control and Optimization*.

[8] Donge V., **Lian B.**, Lewis F.L., Davoudi A. (2023). Reinforcement learning for complex nonlinear systems, *IEEE Transactions on Cybernetics*.

Acknowledgment

- 2020/6-2023/6 **Army Research Office** Grant W911NF-20-1-013, Graphical Games and Distributed Reinforcement Learning Control in Human- networked Multi-group Societies, PI: Frank L. Lewis, Yan Wan, and Ali Davoudi. $391,083.00.

- 2018/6-2022/5 **Office of Naval Research** Grant N00014-18-1-2221, Optimal Design for Assured Performance of Interactive Multibody Systems: Guaranteed Controls for Multi-pursuers, Estimation, Optimal Learning, Scalable Uncertainty Sampling, and Time-critical Communication, PI: Frank L. Lewis and Yan Wan. $815,019.00.

- 2018/9-2021/8 **National Science Foundation** Grant 1839804, EAGER: Real-Time: Collaborative Research: Unified Theory of Model-based and Data-driven Real-time Optimization and Control for Uncertain Networked Systems, PI: Frank L. Lewis, Yan Wan, and Ali Davoudi. $220,010.00.

- 2019/2-2019/12. **Lockheed Martin** Advanced Technology Labs, Heterogeneous Autonomous Networks for Sensor Optimizing Locomotion, Research Contract, PI: Frank L. Lewis and Yan Wan. $150,000.

CONTENT

- Motivation
- Distributed Estimation with Distributed Kalman Filters
- Intention Identification with Inverse Reinforcement Learning
- Multiagent Control with Reinforcement Learning
- **Future Work**

Future Research Interests

Large-Scale Connectivity

?



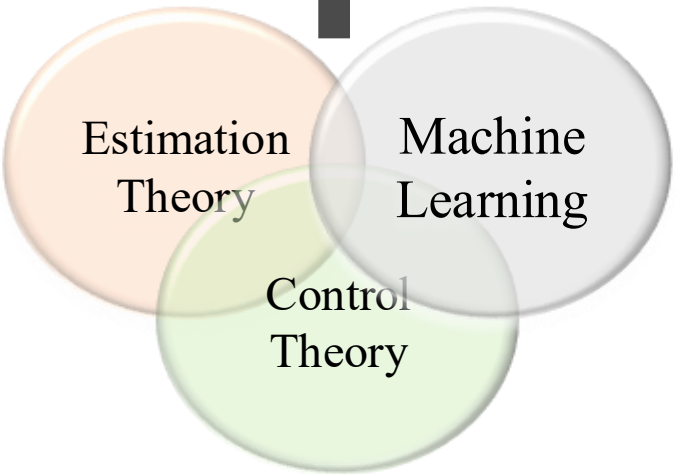




Edge Computing

?

Next Generation of Multiagent Dynamic Systems



Real-Time Optimal Control

?

Safety and Resilience

?

Thank you!

Q & A