Lessons from Adaptive Control: Towards Real-time Machine Learning

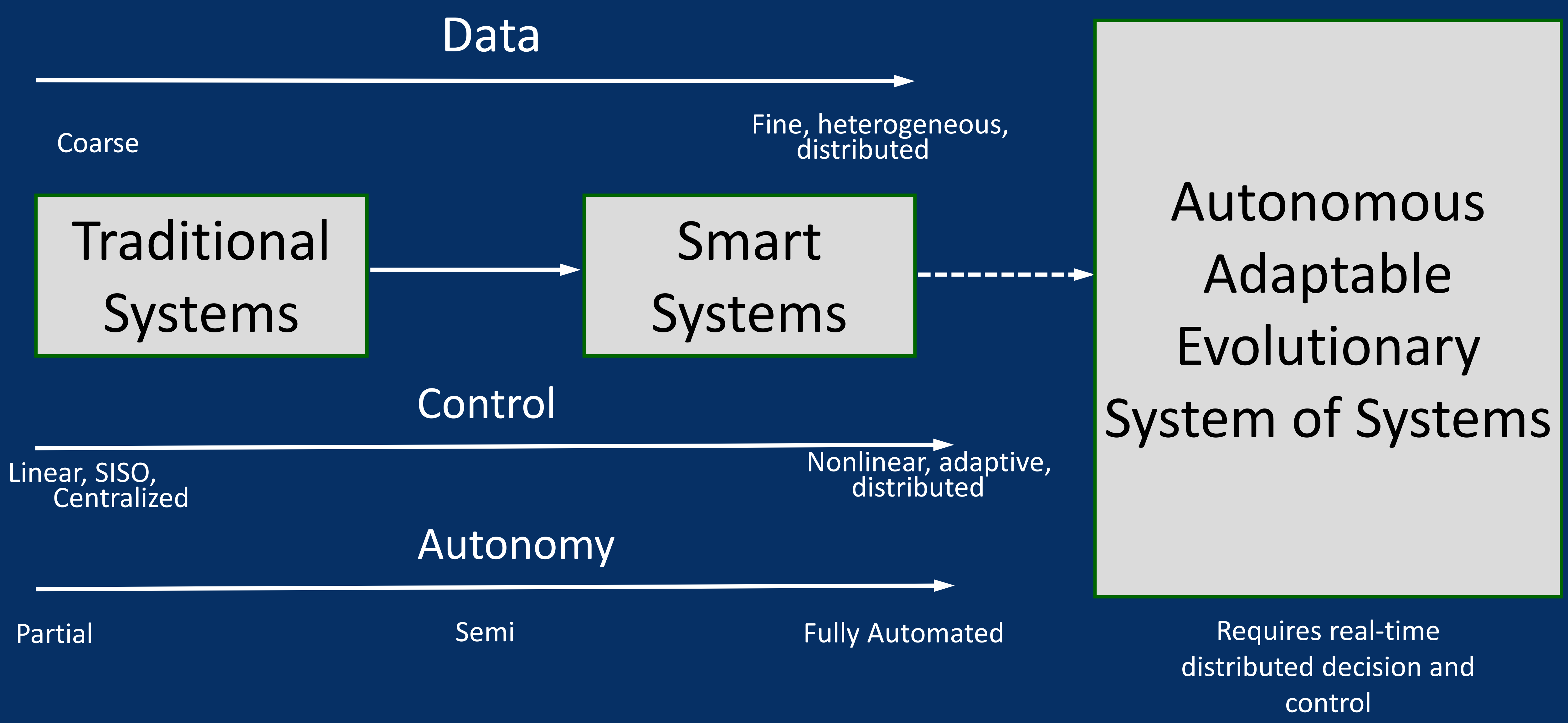
Anuradha Annaswamy*

Active-Adaptive Control Laboratory

Massachusetts Institute of Technology

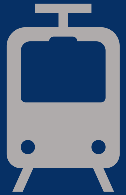
* In collaboration with Mike Bolender, Yingnan Cui, Peter Fisher, Joey Gaudio, Anubhav Guha, Eugene Lavretsky, Daniel Maldonado, Jose Moreu, Arnab Sarker, Sunbochen Tang

Evolution of Systems

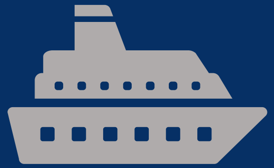


Every 10^9 miles current transportation systems have:

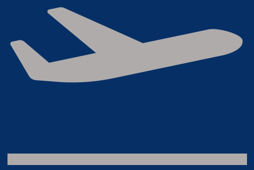
7 Deaths



0.4 Deaths



0.08 Deaths



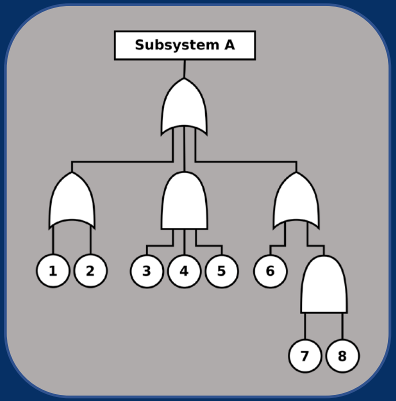
0.07 Deaths

Traditional Approaches to Safety Guarantees

System Analysis

Hazard Class	SW Level	Failure/Flight Hr
Catastrophic	A	10^{-9}
Hazardous	B	10^{-7}
Major	C	10^{-5}
Minor	D	-------
No Effect	E	-------

Subsystem Analysis

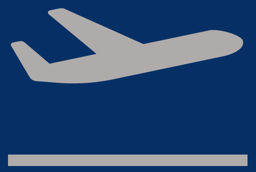


Component Analysis

- Empirical Testing
- Model Based
- Formal Methods

Every 10^9 miles autonomous transportation systems require:

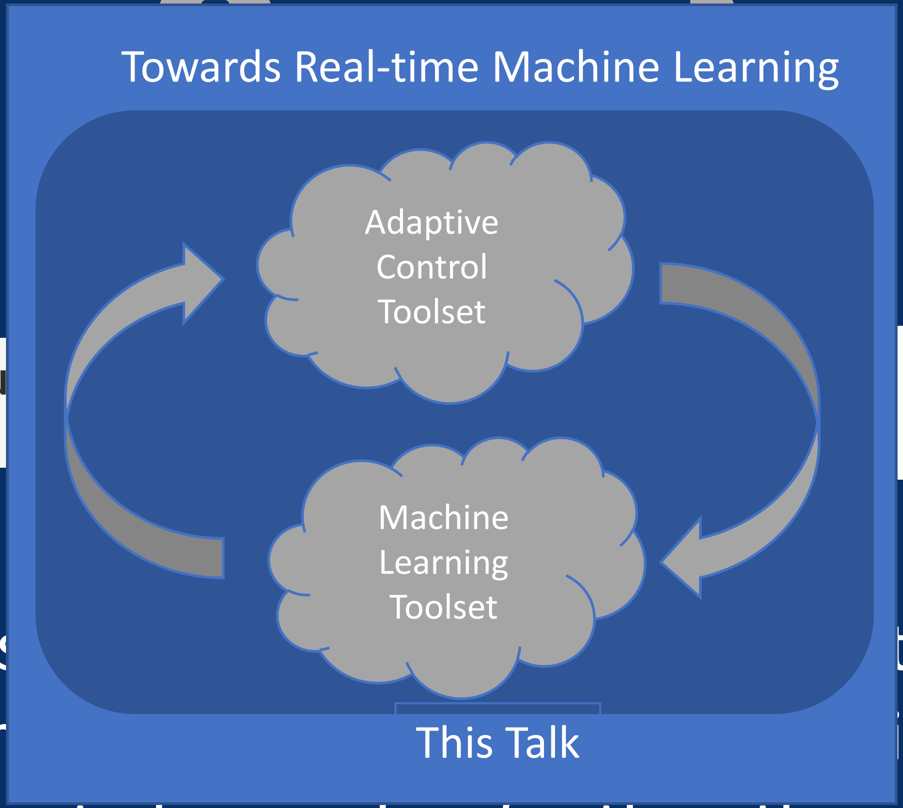
<<7 Deaths



<<0.07 Deaths


3 crashes, 3 deaths raise questions about Tesla's Au
TOM KRISHER January 3, 2020

APnews.com




Wed, Mar 20, 2019 page9

Boeing 737 crashes raise tough questions on aircraft automation

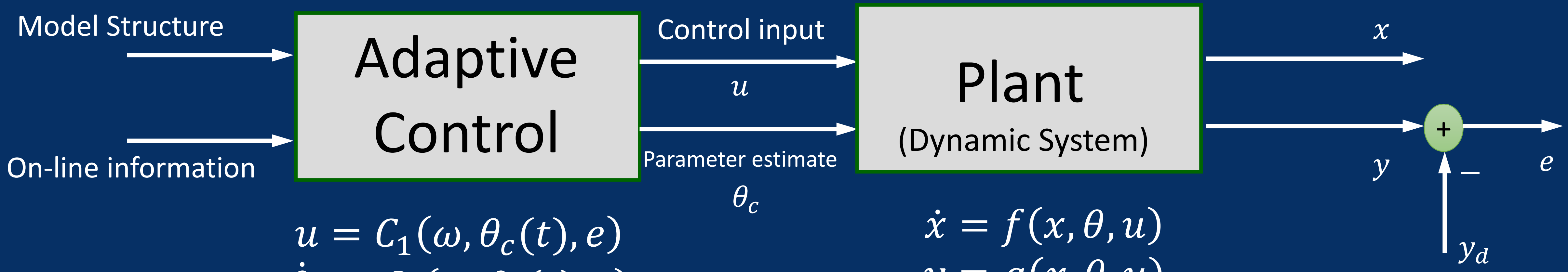
taipeitimes..com

- Being "just as … t it
- Public sentim … ire autonomous transportation to be much safer than the current systems

Outline

- Learning in Adaptive Systems
  - Adaptive Estimation and Adaptive Control
  - Error Models & Learning rules
  - Stability framework – Imperfect Learning
  - Persistent Excitation – Learning with guarantees
- Machine Learning
  - Neural Networks
  - Reinforcement Learning
- New Solutions
  - High-order Tuners – towards accelerated performance
  - Sub-Gaussian spectral lines – towards robust learning
  - Integration of RL and Adaptive Control – towards real-time machine learning
  - Safety and Stability – Adaptation with Calibrated CBF

Learning in Adaptive Systems



$$u = C_1(\omega, \theta_c(t), e)$$
$$\dot{\theta}_c = C_2(\omega, \theta_c(t), e)$$

$$\dot{x} = f(x, \theta, u)$$
$$y = g(x, \theta, u)$$

θ: parameter, unknown

x, y, u: state, output, input

ω: real-time data

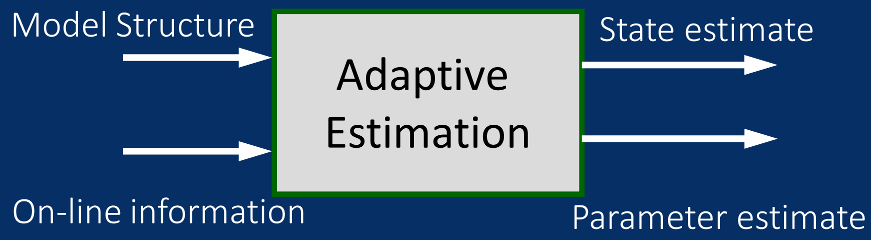
e: tracking error

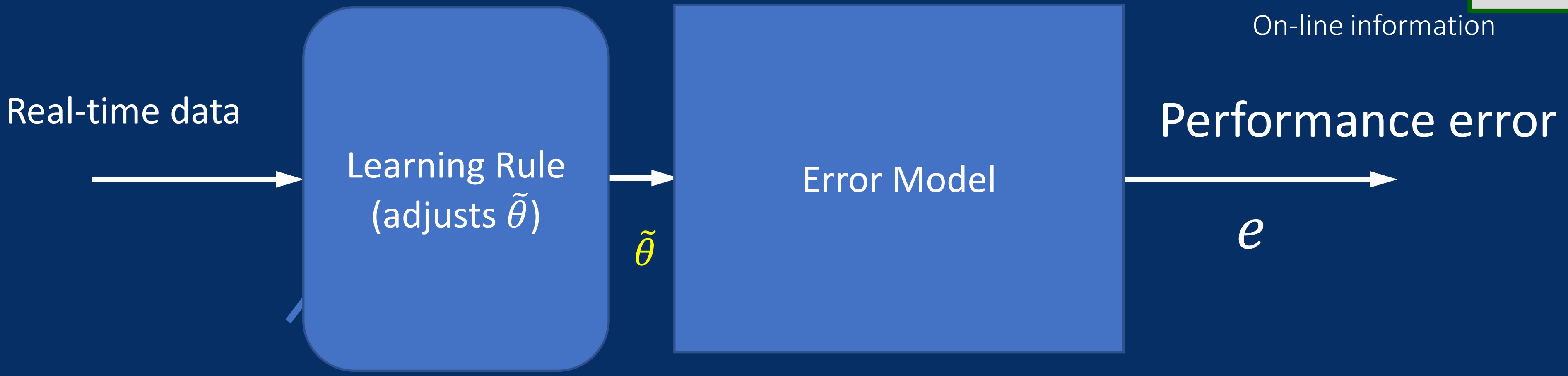
θ_c: control parameter estimate

Goal: Find u, C_1, C_2 so that regulation and tracking occur

Error Models – Two types of errors

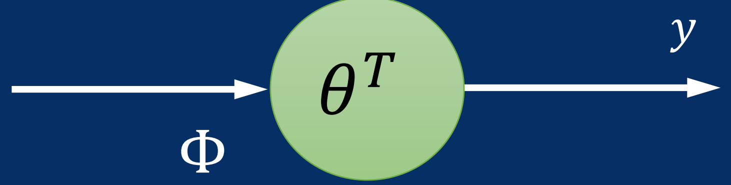
- e: Performance error (ex. $\hat{x} - x;\ x - x_m$)
  - can be measured, needs to be reduced
- $\tilde{\theta}$: Parameter error (ex. $\hat{\theta} - \theta$)
  - Unknown, can be adjusted – Learning Rule





Goal: Determine error models and learning rules

System Model:



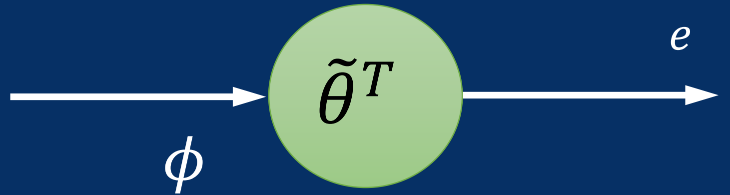
$$y = \phi^T \theta$$

Performance error: $e = \hat{y} - y$

Build an estimator: $\hat{y} = \phi^T \hat{\theta}$

Parameter error: $\tilde{\theta} = \hat{\theta} - \theta$

Error Model:



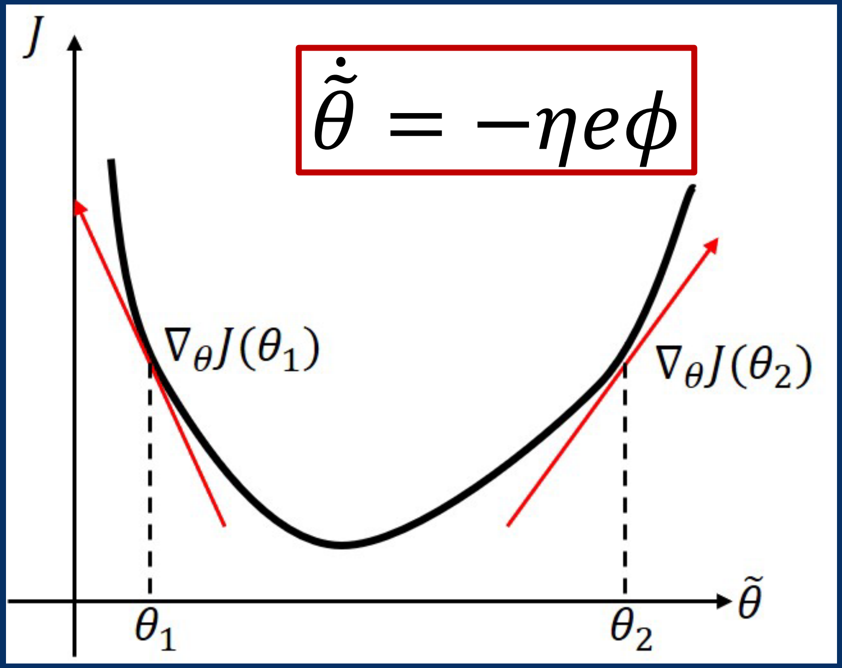
$$J = e^2$$

$$\dot{\tilde{\theta}} = -\eta_t \nabla J$$

Stability Framework:*

$$V = \left|\left|\tilde{\theta}\right|\right|^2$$

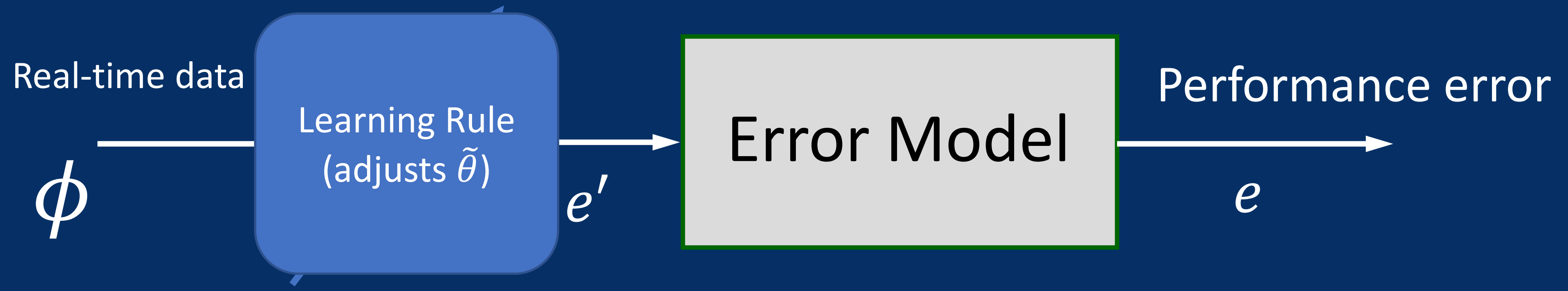
$$\dot{V} = 2\tilde{\theta}^T \dot{\tilde{\theta}} = -2\,\eta \tilde{\theta}^T e\phi = -2\eta e^2 \le 0$$



* Narendra and Annaswamy, 1989

Robbins and Munro, 1951

Dynamic Error Models



- e' cannot be measured, but only after some latencies as e.
- Performance and Learning are conflicting objectives
- With persistent excitation, $\tilde{\theta}(t) \rightarrow 0 \Rightarrow$ Learning!
- With imperfect learning, guaranteed performance can be ensured: $e(t) \rightarrow 0 \Rightarrow$ Control Performance
- Use a stability (Lyapunov) framework

Goal: Find *learning rules* for adjusting $\tilde{\theta}$ so that $\dot{V} \leq 0$

Adaptive Control: Milestones*

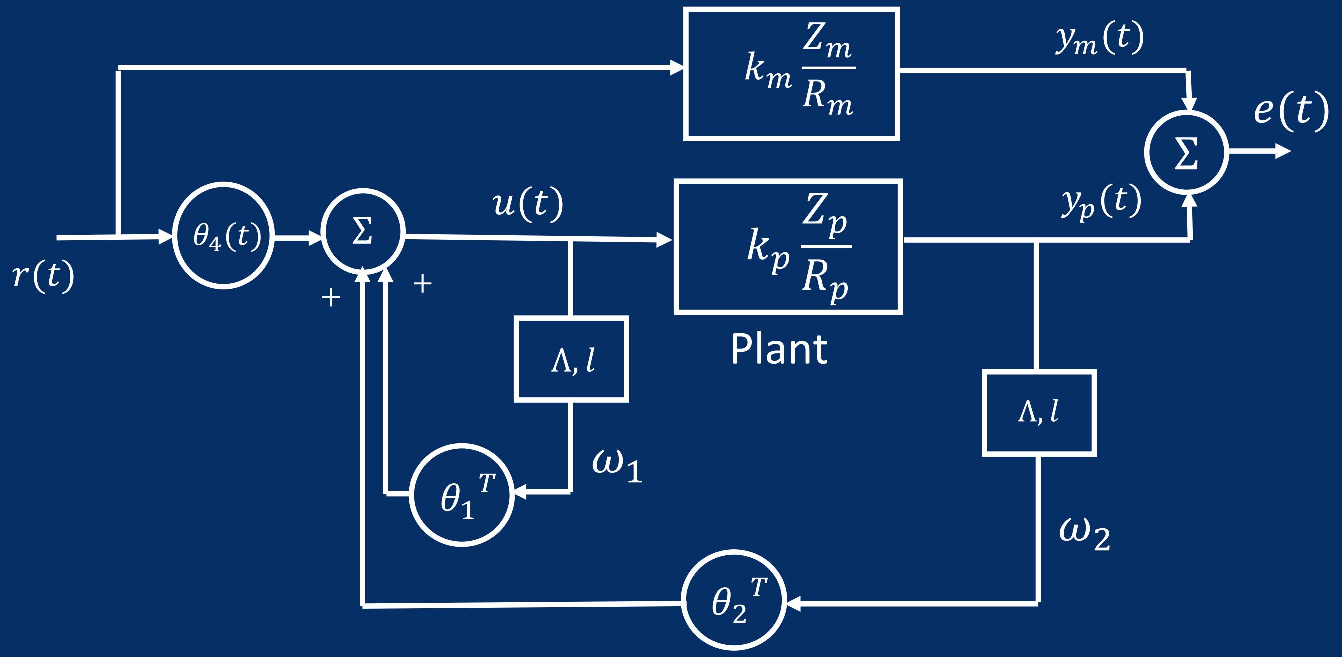
$$\dot{x} = f(x, \theta, u)$$
$$y = g(x, \theta, u)$$

1. f, g: linear. Stability established in 1980.
2. Robustness to disturbances and unmodeled dynamics in the ’90s.
3. f, g: nonlinear. Stability and robustness established in 1990-2000.

$$u = C_1(\omega, \theta_c(t), e)$$
$$\dot{\theta}_c = C_2(\omega, \theta_c(t), e)$$

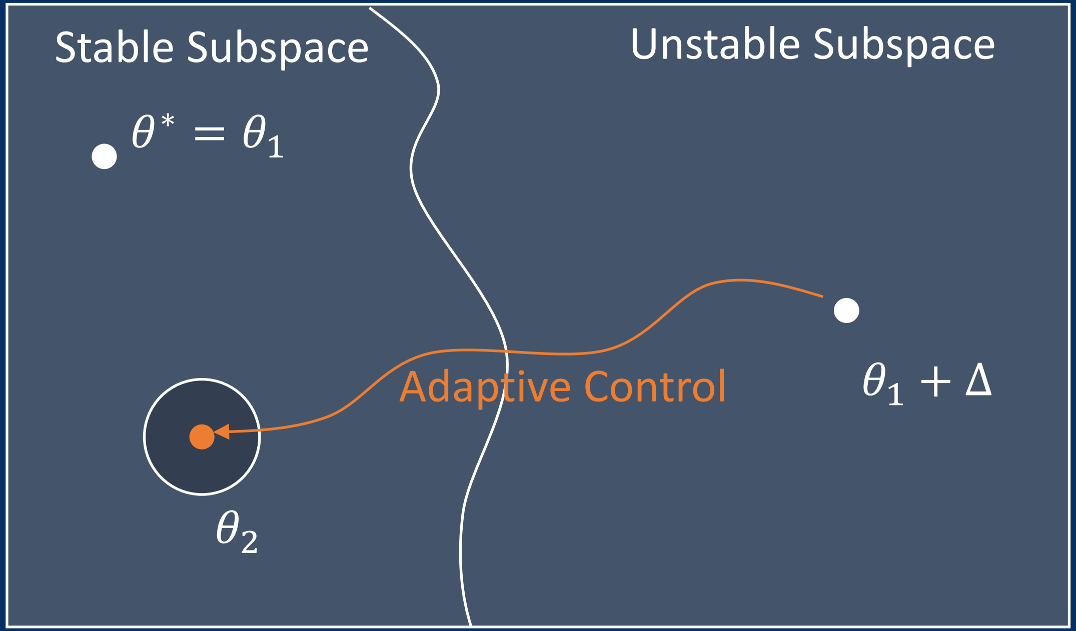
* A.M. Annaswamy and A.L. Fradkov, A historical perspective of adaptive control and learning, *Annual Reviews in Control*, 2021.

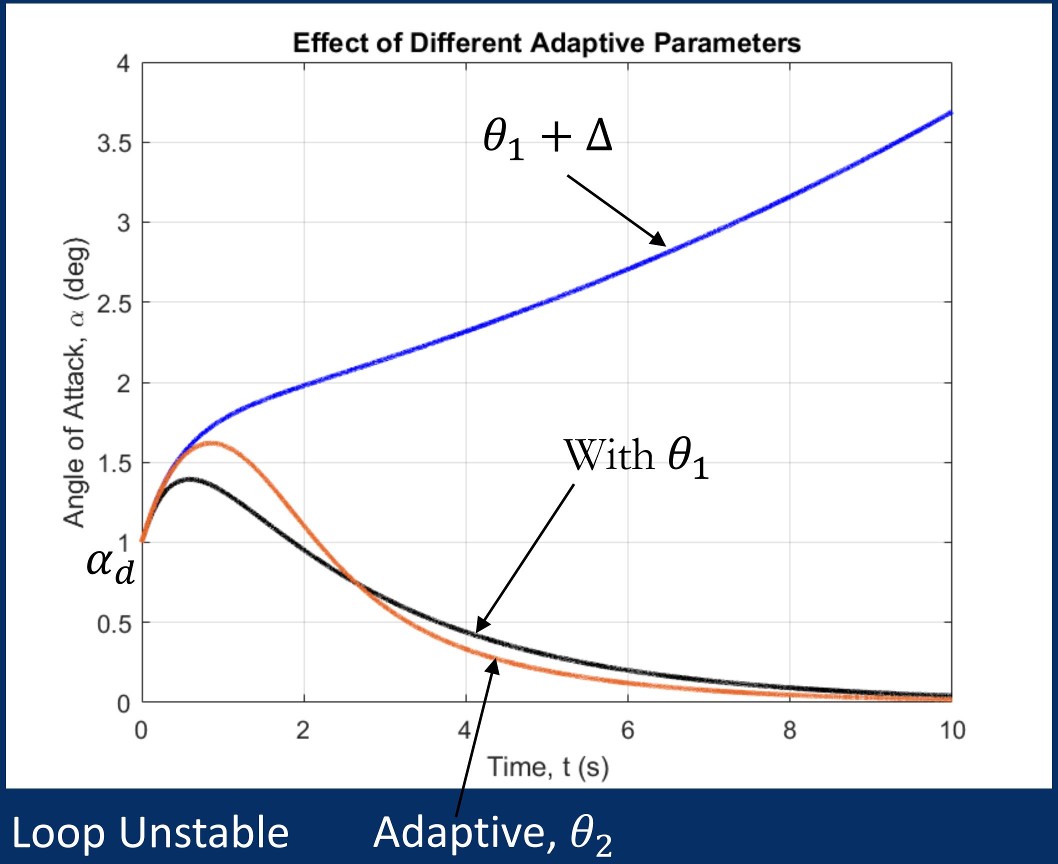
Adaptive Controller Structure

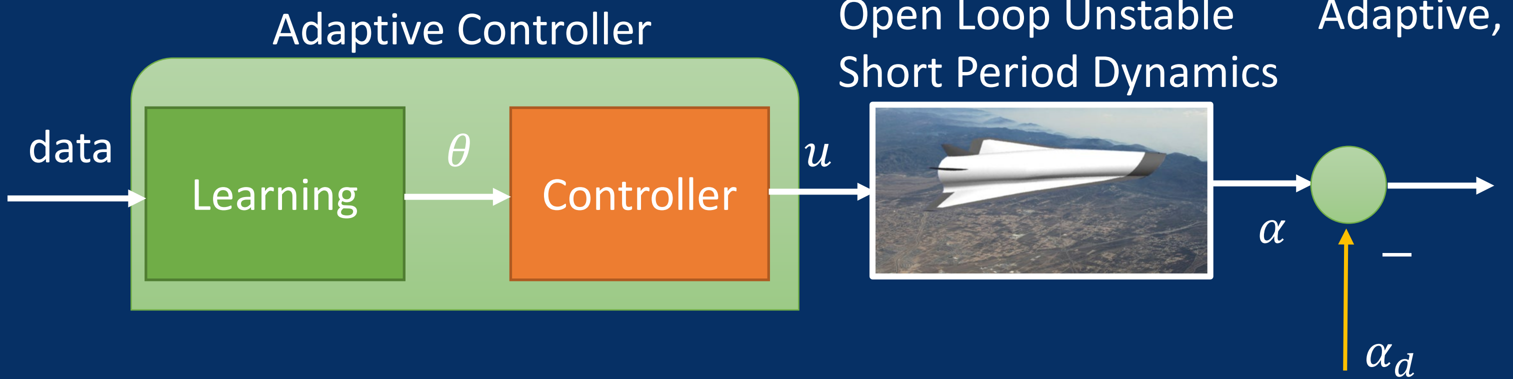


- $\exists \theta_1, \theta_2, \lambda, \theta_4$ s.t. Plant + Controller = Reference model
- Adaptive law: $\dot{\theta}(t) = -\Gamma e(t)\omega(t)$ $(n^* = 1)$; SPR model; Kalman-Yakubovich lemma
- For $n^* \geq 2$: augmented error, high-order tuner, back-stepping

Guarantees with Imperfect Learning





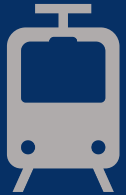


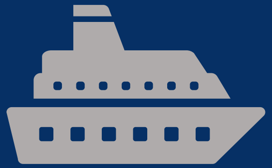
Performance goal: $Min(e)$

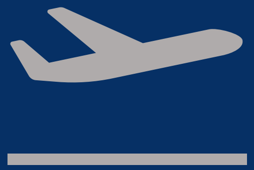
Imperfect learning,
yet improved performance

Every 10^9 miles current transportation systems have:


7 Deaths


0.4 Deaths


0.08 Deaths

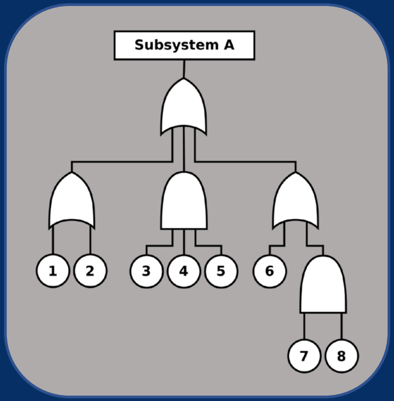

0.07 Deaths

Traditional Approaches to Safety Guarantees

System Analysis

Hazard Class	SW Level	Failure/Flight Hr
Catastrophic	A	10^{-9}
Hazardous	B	10^{-7}
Major	C	10^{-5}
Minor	D	-------
No Effect	E	-------

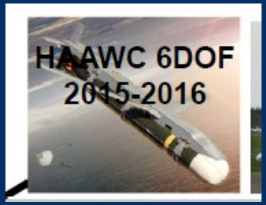
Subsystem Analysis

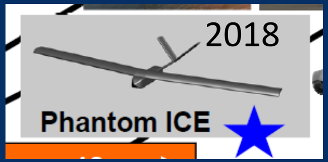


Adaptive Control

- Model-based
- On-line control
- Integration with cyber
- Integration with human decision-making

Adaptive Control with Learning: Applications





Improved performance and reliability

Unmanned Flight Control Applications

★ - Flight Test

Courtesy Boeing Company

Machine Learning

The ability of a computer to learn using on-line data

- Significant success in image & speech recognition, games

Motivation:

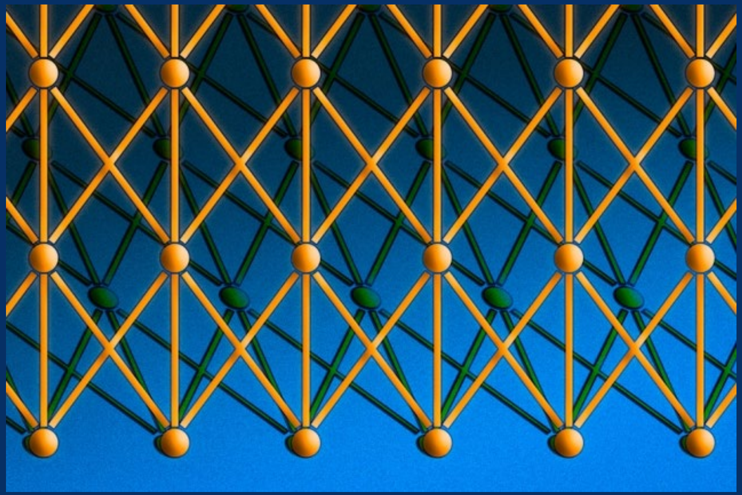
- Complex environment
- Hard to sense
- Difficult to model
- Big-data & Computational complexity

Typical approaches for learning:

1. Approximation of a Nonlinear Mapping
    - *Neural Networks*
2. Optimization of a Cost Function
    - *Reinforcement Learning*

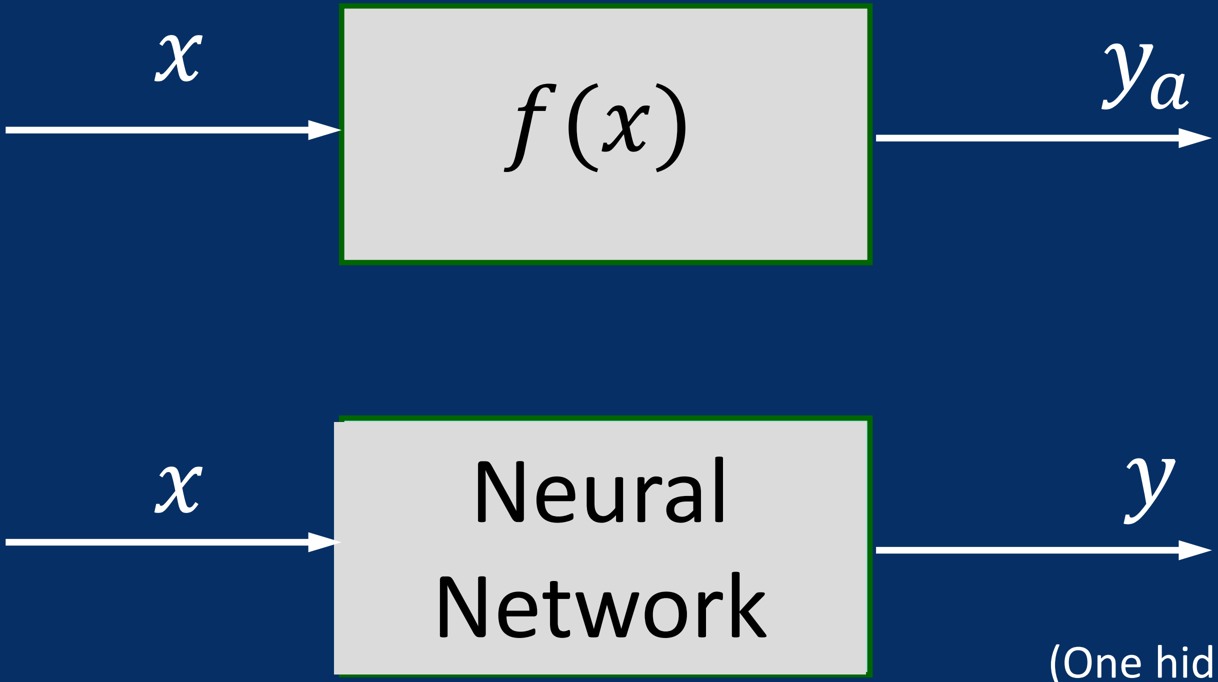
A bit of history

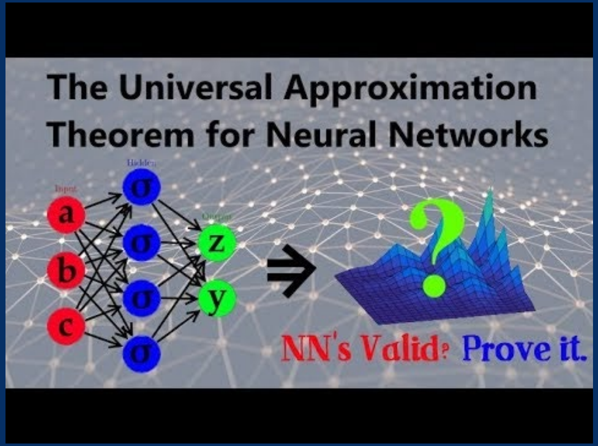
- Proposed in 1944 by McCullough and Pitts
- Controversy in the '70s: Multilayered Perceptrons, Minsky and Papert
- Resurgence in the 1980s
- A re-resurgence in the 21st century – fast processing power of graphics



(from MIT News)

Universal Approximation Theorem





(From YouTube, *Why do Nnets work?*)

$$y = \sum_{i=1}^{N} w_{2i}^{*T} \phi\left(w_{1i}^{*T} x + b_i\right)$$

(One hidden layer)

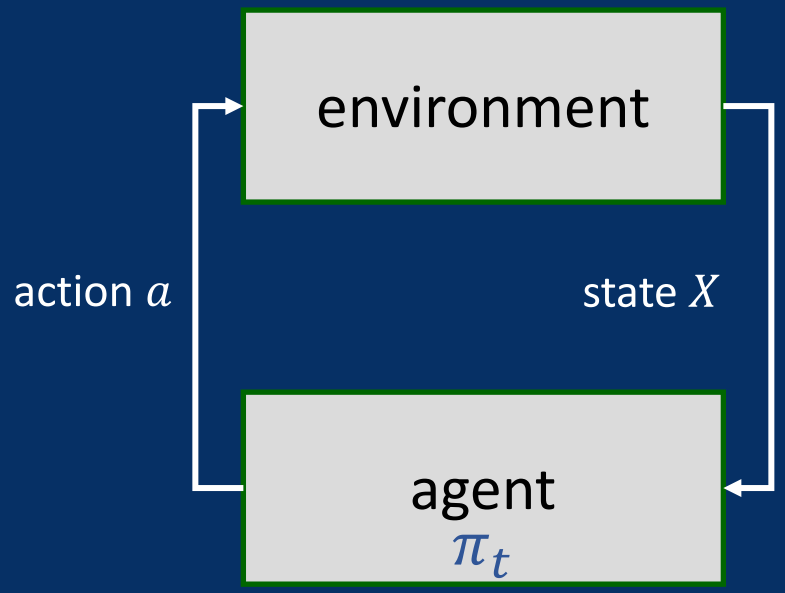
$\phi(.)$: activation function

$$\dot{\theta} = -\Gamma \nabla_\theta L_t(\theta)$$

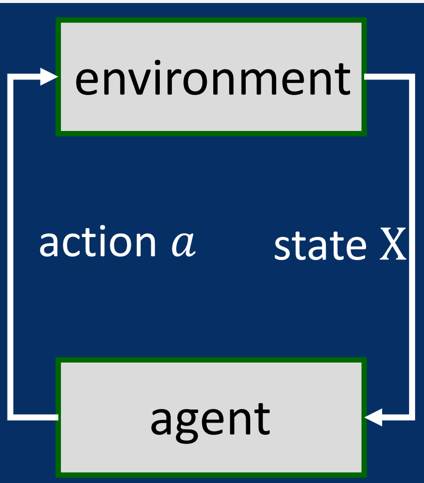
For $\epsilon > 0, \exists N, w^* \; s.t. \quad |y_a - y| \leq \epsilon \quad \forall x \in X$

Estimate weights w_j using back-propagation

- Model/Environment: Markov Decision Process
- Maximize a Value function $V(x_i) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(X_t) | X_0 = x_i]$; γ: Discount factor
- $R(X_i)$: Reward associated with X_i
- Control/Agent: Policy π_t
- Choose π_t such that $V(x_i)$ is optimized
- Express using a Q-function $Q^*(x_i, u) := R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u V^*(x_j)$



$[n]$: future instants



- Model/Environment: Markov Decision Process
- Maximize $Q(x_i, u)$; Optimal policy: $\pi^*(x_i) = \text{argmax}_{u \in \mathcal{U}} Q^*(x_i, u)$
- Approximate $Q(x_i, u)$ by $Q_\theta(x_i, u)$:
  - Linear regression: $Q_\theta(x_i, u) = \boldsymbol{\theta}^\top \boldsymbol{\psi}(x_i, u)$
  - Neural networks: $Q_\theta = \sum_{i=1}^{N} \theta_{2i}^{*T} \phi\left(\theta_{1i}^{*T} x + \theta_{1u}^{T} u + b_i\right)$

Estimate the unknown parameters through an iterative algorithm: $\theta_{k+1} = \theta_k - \gamma_k \nabla L_k(\theta_k)$

- Unknown nonlinear system: $\dot{x} = f(x, u)$
- Minimize infinite-horizon cost: $J(u; x_0) = \int_0^\infty r(x, u)dt$
- Optimal cost-to-go: $V^*(x_0) = \inf_u J(u; x_0)$
- Hamilton-Jacobi-Bellman Equation: $0 = \inf_a \{\partial_x V^*(x) f(x, a) + r(x, a)\}$
- Optimal control policy: $\mu^*(x) = \operatorname{arginf}_a \{\partial_x V^*(x) f(x, a) + r(x, a)\}$
- Reinforcement Learning: Policy/Value Iteration

Approximate Policy Iteration for Optimal Control*

- Use neural networks to approximate H_k, V_k and μ_k
  - $\widehat{H}(w_k, w, u) \approx H_k(x, u),\ \ \hat{V}(c_k, x) \approx V_k(x),\ \ \ \hat{\mu}(\theta_k, x) \approx \mu_k(x)$
- The iterative procedure of the resulting approximate policy iteration:
  - For a sampling period $[t_k, t_{k+1}]$, collect data (x, u) and cost $r(x, u)$
  - Policy evaluation: Given θ_k, solve for w_k and c_k from HJB equation

$$0 = \int_{t_k}^{t_{k+1}} \widehat{H}\left(w_k, x, \hat{\mu}(\theta_k, x)\right) dt, \qquad \hat{V}(c_k, x) = \int_{t_k}^{t_{k+1}} \left(\widehat{H}\left(w_k, x, u\right) - r(x, u)\right) dt$$

  - Policy improvement: Update $\theta_{k+1} = \inf_{\theta} \widehat{H}\left(w_k, x, \hat{\mu}(\theta, x)\right)$
- Special case: Single layer network reduces the problem to weight estimation
- Estimation accuracy depends on persistent excitation condition

Every 10^9 miles current transportation systems have:

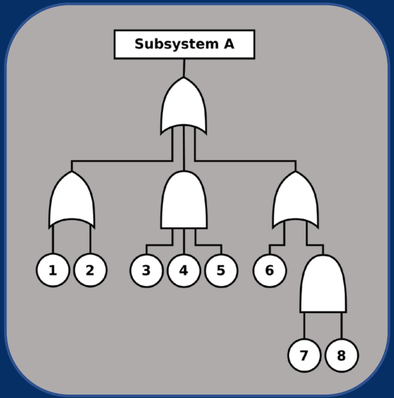
7 Deaths | 0.4 Deaths | 0.08 Deaths | 0.07 Deaths

Traditional Approaches to Safety Guarantees

System Analysis

Hazard Class	SW Level	Failure/Flight Hr
Catastrophic	A	10^{-9}
Hazardous	B	10^{-7}
Major	C	10^{-5}
Minor	D	-------
No Effect	E	-------

Subsystem Analysis

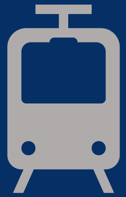


Component Analysis

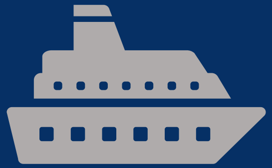
- Empirical Testing
- Model Based
- Formal Methods

Every 10^9 miles autonomous transportation systems require:

<<7 Deaths



<<0.4 Deaths



<<0.08 Deaths

<<0.07 Deaths

New
Approaches to
Safety Guarantees

Machine Learning

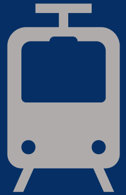
gluon.ai

Over 14 million photos with 21000 categories, best classification rate to date: 85.8%*

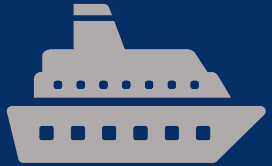
*L. Wei, “Circumventing Outliers of AutoAugment with Knowledge Distillation”, arXiv, 2020.

Every 10^9 miles autonomous transportation systems require:

<<7 Deaths



<<0.4 Deaths

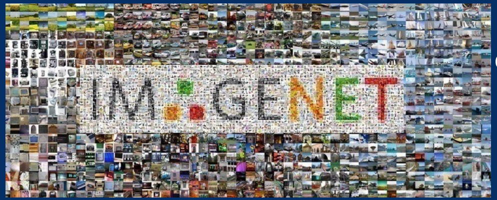


<<0.08 Deaths

<<0.07 Deaths

New
Approaches to
Safety Guarantees

Machine Learning



gluon.ai

Over 14 million photos with 21000 categories, best classification rate to date: 85.8%*

Not adequate for safety critical systems

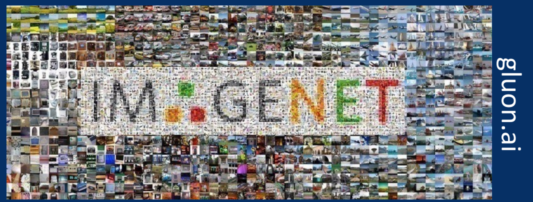
*L. Wei, "Circumventing Outliers of AutoAugment with Knowledge Distillation", arXiv, 2020.

Every 10^9 miles autonomous transportation systems require:

<<7 Deaths	<<0.4 Deaths	<<0.08 Deaths	<<0.07 Deaths

New
Approaches to
Safety Guarantees

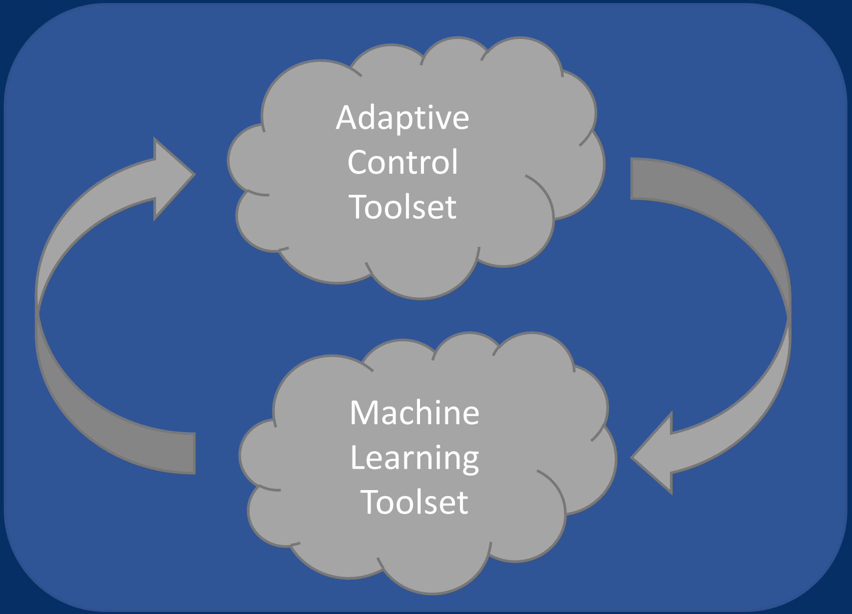
Machine Learning



Over 14 million photos with 21000 categories, best classification rate to date: 85.8%*

*L. Wei, "Circumventing Outliers of AutoAugment with Knowledge Distillation", arXiv, 2020.

Rest of this talk



NEW SOLUTIONS:

ACCELERATED PERFORMANCE

- High-order tuner

ROBUST LEARNING

- Sub-Gaussian spectral lines

REAL-TIME MACHINE LEARNING

- Integration with reinforcement learning

STABIITY AND SAFETY

- Adaptation and Calibrated Control Barrier Functions

NEW SOLUTIONS:

ACCELERATED PERFORMANCE

- High-order tuner

ROBUST LEARNING

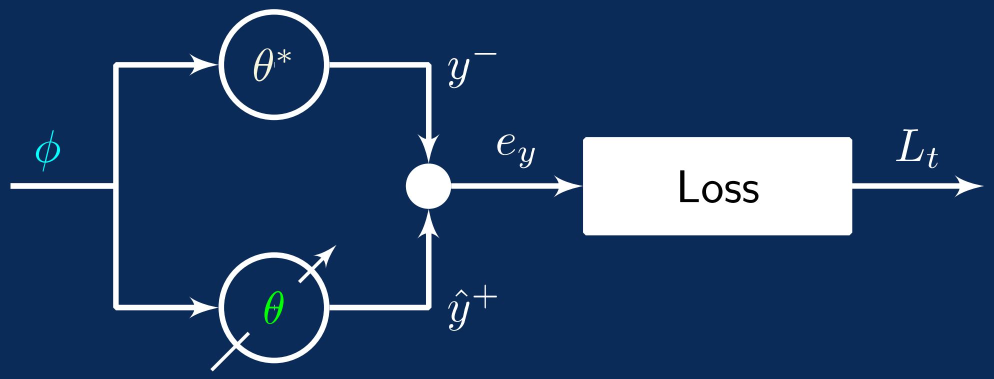
- Sub-Gaussian spectral lines

REAL-TIME MACHINE LEARNING

- Integration with reinforcement learning

STABIITY AND SAFETY

- Adaptation and Calibrated Control Barrier Functions



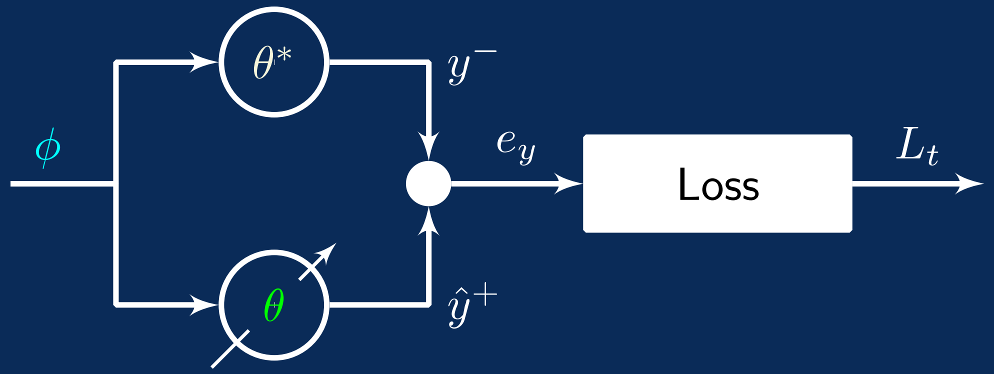
Plant: $y = \phi^T \theta^*$

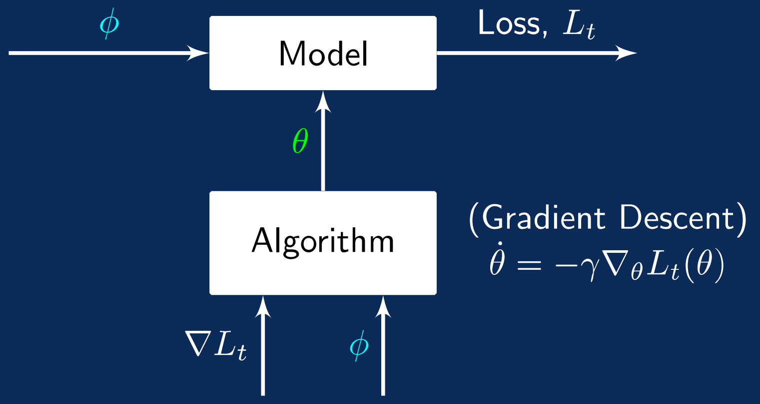
Estimator: $\hat{y} = \phi^T \theta$

Loss:

$$L_t(\theta) = \frac{1}{2}\|\phi^T \theta - y\|_2^2$$

(any convex function of θ)





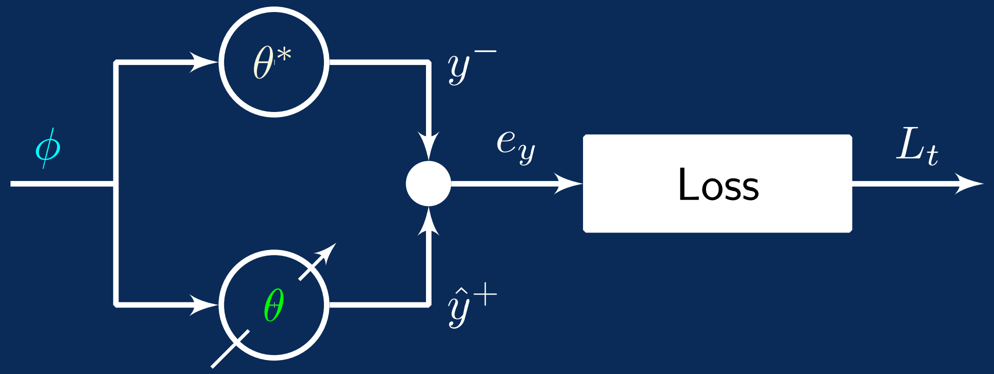
Plant: $y = \phi^T \theta^*$

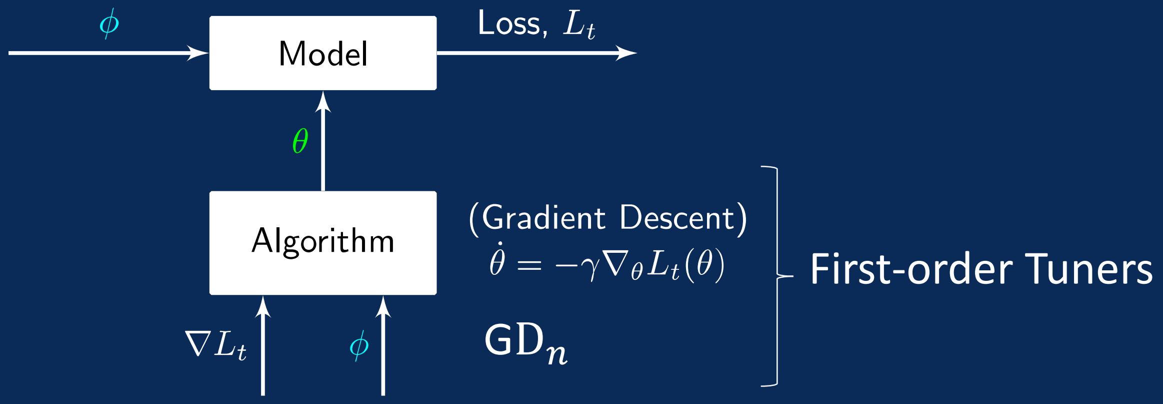
Estimator: $\hat{y} = \phi^T \theta$

Loss:

$$L_t(\theta) = \frac{1}{2} \|\phi^T \theta - y\|_2^2$$

(any convex function of θ)





Plant: $y = \phi^T \theta^*$

Estimator: $\hat{y} = \phi^T \theta$

Loss:

$$L_t(\theta) = \frac{1}{2} \|\phi^T \theta - y\|_2^2$$

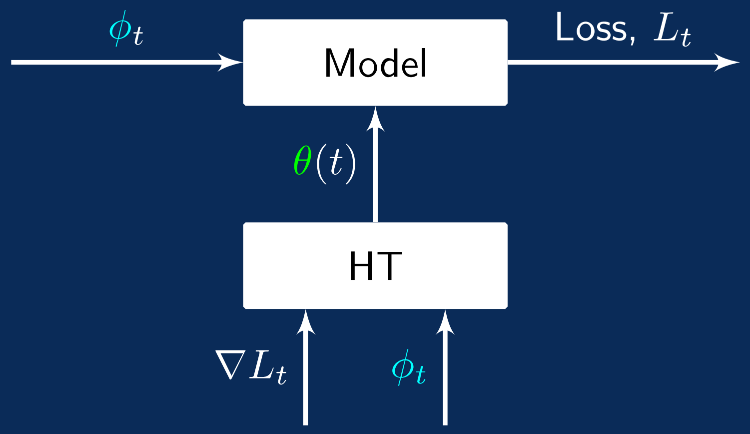
(any convex function of θ)

Gradient Descent, Normalized ($\mathrm{GD_n}$):

$$\dot{\theta}(t) = -\frac{\Gamma}{\mathcal{N}_t} \nabla_\theta L_t(\theta)$$

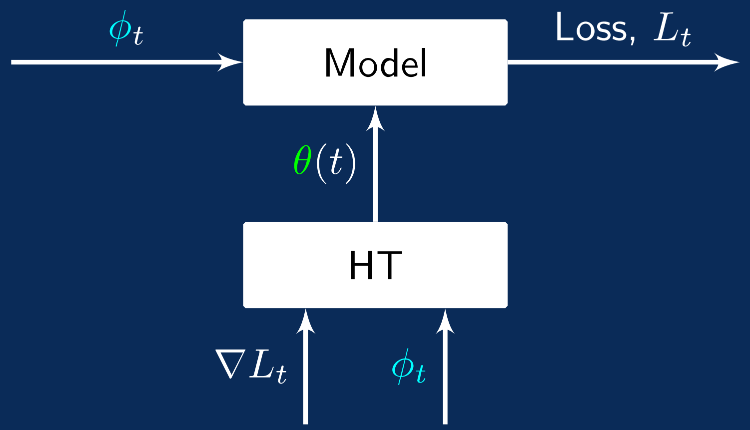
Γ : learning rate > 0; $\mathcal{N}_t = 1 + \|\phi\|_2^2$: Normalization

Accelerated Performance with a High-order Tuner*



* A. S. Morse. High-order parameter tuners for the adaptive control of linear and nonlinear systems, 1993.
** J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "A Class of High Order Tuners for Adaptive Systems," IEEE Control Systems Letters, 2021.

Accelerated Performance with a High-order Tuner*



High-Order Tuner (HT)[1]:

$$\dot{\vartheta}(t) = -\frac{\gamma}{\mathcal{N}_t}\nabla L_t(\theta(t)),$$

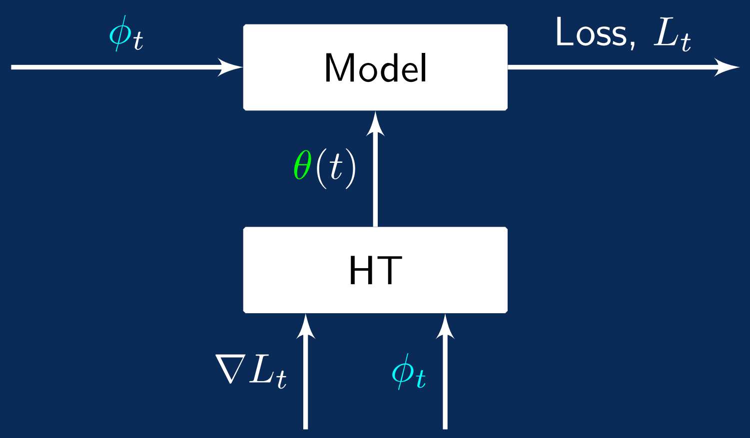
$$\dot{\theta}(t) = -\beta(\theta(t) - \vartheta(t)).$$

$$\mathcal{N}_t = 1 + \|\phi_t\|^2$$

* A. S. Morse. High-order parameter tuners for the adaptive control of linear and nonlinear systems, 1993.

** J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, “A Class of High Order Tuners for Adaptive Systems,” IEEE Control Systems Letters, 2021.

Accelerated Performance with a High-order Tuner*



High-Order Tuner (HT)[1]:

$$\dot{\vartheta}(t) = -\frac{\gamma}{\mathcal{N}_t}\nabla L_t(\theta(t)), \qquad \mathcal{N}_t = 1 + \|\phi_t\|^2$$

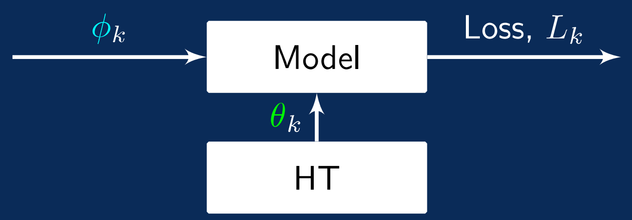
$$\dot{\theta}(t) = -\beta(\theta(t) - \vartheta(t)).$$

Theorem: All solutions are globally bounded, with a Lyapunov function

$$V = \frac{1}{\gamma}\|\vartheta - \theta^*\|^2 + \frac{1}{\gamma}\|\theta - \vartheta\|^2$$

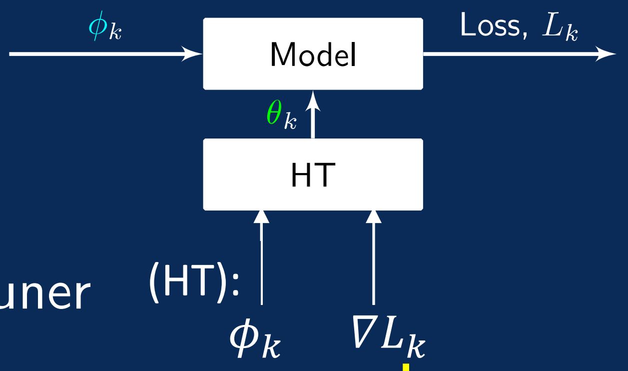
* A. S. Morse. High-order parameter tuners for the adaptive control of linear and nonlinear systems, 1993.
** J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, “A Class of High Order Tuners for Adaptive Systems,” IEEE Control Systems Letters, 2021.

Accelerated Performance (discrete-time)*



* J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

Accelerated Performance (discrete-time)*



Current Nnet algorithms:

$$\theta_{k+1} = \theta_k - \gamma_k \nabla_\theta L(\theta_k)$$

Discrete and continuous High-Order Tuner (HT):

Proposed Discrete HT

$$\bar{\theta}_k = \theta_k - \gamma\beta \frac{\nabla L_k(\theta_k)}{\mathcal{N}_k}, \qquad \mathcal{N}_k = 1 + \|\phi_k\|^2,$$

$$\theta_{k+1} = \bar{\theta}_k - \beta(\bar{\theta}_k - \vartheta_k),$$

$$\vartheta_{k+1} = \vartheta_k - \gamma \frac{\nabla L_k(\theta_{k+1})}{\mathcal{N}_k}$$

Proposed Continuous HT

$$\dot{\vartheta} = -\frac{\gamma}{\mathcal{N}_t} \nabla_\theta L_t(\theta),$$

$$\dot{\theta} = -\beta(\theta - \vartheta)$$

Theorem: All solutions are globally bounded, with a Lyapunov function

$$V_k = \frac{1}{\gamma}\|\vartheta_k - \theta^*\|^2 + \frac{1}{\gamma}\|\theta_k - \vartheta_k\|^2$$

* J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

Adaptive Control tools: Convergence of errors to zero.

- ▷ Asymptotic Tools: $f(\theta_k) - f(\theta^*) \to 0$ as $k \to \infty$

** Y. Nesterov (2018). *Lectures on Convex Optimization.* Springer.

Adaptive Control tools: Convergence of errors to zero.

- Asymptotic Tools: $f(\theta_k) - f(\theta^*) \to 0$ as $k \to \infty$
- Non-asymptotic tools:
  - GD: $f(x_k) - f(x^*) \leq \epsilon$ if $k \geq \mathcal{O}(1/\epsilon)$
  - Nesterov **: $f(x_k) - f(x^*) \leq \epsilon$ if $k \geq \mathcal{O}(1/\sqrt{\epsilon})$

Theorem 5: HT guarantees that *

$$L_k(\theta_k) - L_k(\theta^*) \leq \epsilon \text{ for } k \geq \mathcal{O}(1/\sqrt{\epsilon} \cdot \log(1/\epsilon))$$

$$f_k = \bar{L}\left(\frac{L_k}{N_k} + g_k\right) \quad (g_k \text{ small; ensures strong convexity})$$

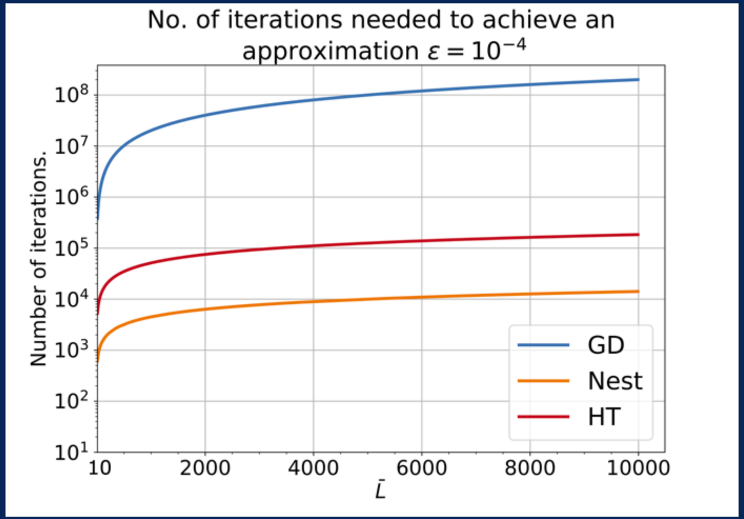
* J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.
** Y. Nesterov (2018). *Lectures on Convex Optimization.* Springer.

Adaptive Control tools: Convergence of errors to zero.

- ▷ Asymptotic Tools: $f(\theta_k) - f(\theta^*) \to 0$ as $k \to \infty$
- ▷ Non-asymptotic tools:
  - ▷ GD: $f(x_k) - f(x^*) \leq \epsilon$ if $k \geq \mathcal{O}(1/\epsilon)$
  - ▷ Nesterov **: $f(x_k) - f(x^*) \leq \epsilon$ if $k \geq \mathcal{O}(1/\sqrt{\epsilon})$

Theorem 6: HT guarantees that *

$$L_k(\theta_k) - L_k(\theta^*) \leq \epsilon \text{ for } k \geq \mathcal{O}(1/\sqrt{\epsilon} \cdot \log(1/\epsilon))$$

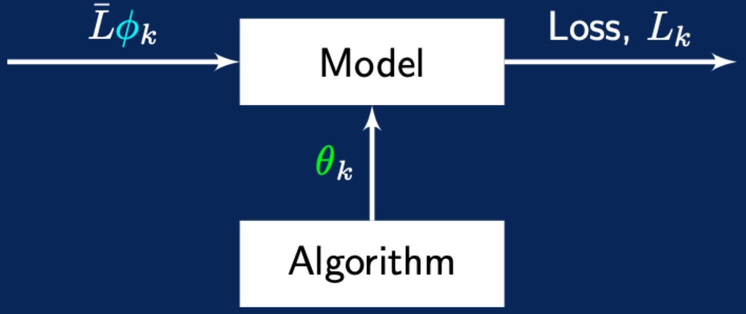


$\bar{L}$: Smoothness parameter.

$$f_k = \bar{L}\left(\frac{L_k}{N_k} + g_k\right)$$ (g_k small; ensures strong convexity)

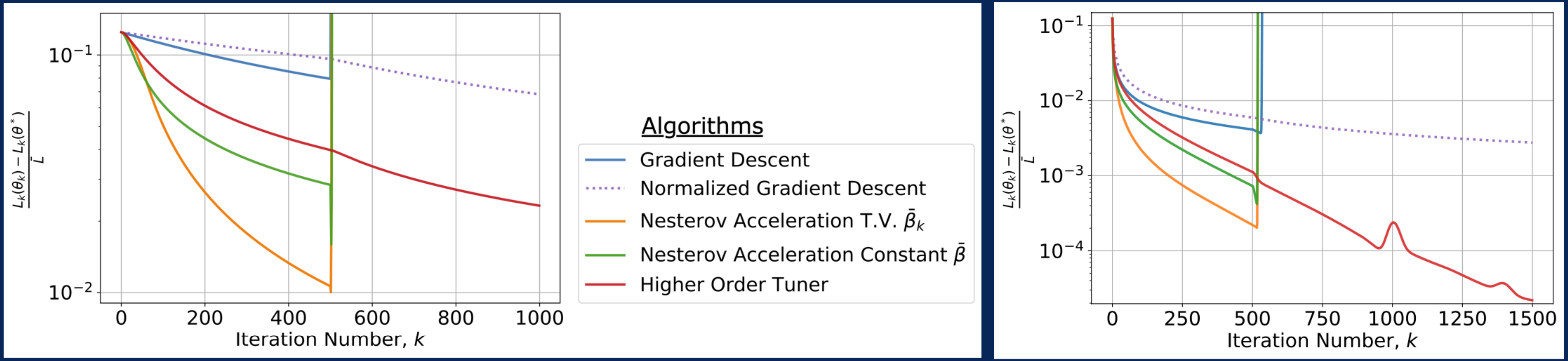
* J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.
** Y. Nesterov (2018). *Lectures on Convex Optimization.* Springer.

Modified Smooth-Hard Problem, with time-varying regressors *



$$L_k(\theta_k) = \|\phi_k^T \theta_k\|^2 + B_k^T \theta_k$$

(quadratic, non-homogeneous, convex)



(a) (b)

Figure: (a) At iteration $k = 500$, step change in $\bar{L}$ from 2 to 8000. (b) At iteration $k = 500$, step change in $\bar{L}$, from 2 to 8.

** Yurii Nesterov. Lectures on Convex Optimization. Springer, 2018 (p. 69).*

* J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

Blurring can be caused by many factors:

- Movement during the image capture process, by the camera or, when long exposure times are used, by the subject
- Out-of-focus optics, use of a wide-angle lens, atmospheric turbulence, or a short exposure time, which reduces the number of photons captured
- Scattered light distortion in confocal microscopy

- Model for blur*:

$$y = \phi^T \theta^* + n$$

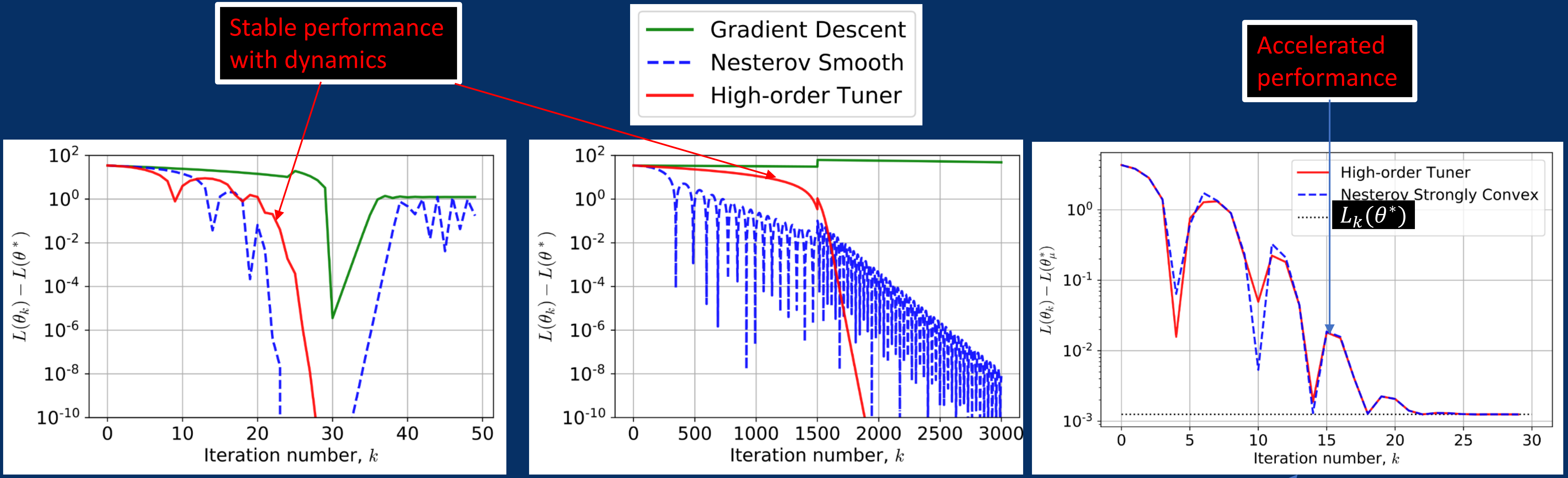
* https://www.mathworks.com/help/images/image-deblurring.html

De-Blurring an Image with a Time-Varying Blur*,**

* Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, *2*(1), 183-202.

** J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

High-order Tuner for Convex and Dynamic Loss Functions*



Step change in b_k from 7 to 14 at $k = 25$

Step change in b_k from 7 to 14 at $k = 1500$

No change in b_k

Loss: $L_k(\theta) = \log(a_k e^{b_k \theta} + a_k e^{-b_k \theta})$

Loss: $L_k(\theta) = \log(a_k e^{b_k \theta} + a_k e^{-b_k \theta}) + \frac{\mu}{2} \|\theta - \theta_0\|^2$

* Moreu, José M., and Anuradha M. Annaswamy. "A Stable High-order Tuner for General Convex Functions." *IEEE L-CSS, 2021.*

** J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, “Accelerated Learning with Robustness to Adversarial Regressors,” 3rd L4DC Conference, 2021.

*** Gaudio, Joseph E., et al. "A Class of High Order Tuners for Adaptive Systems." *IEEE L-CSS, 2020.*

- A new algorithm that utilizes a High-order Tuner (HT) has been proposed
- Leads to stability.
- Has no Hamiltonian; Lagrangian has similarities to that in Wibisono et al. PNAS, 2015.
- Has very nice accelerated learning properties.

Algorithm	Constant Regressor # Iterations	Time-Varying Regressor
Gradient Descent Normalized	$\mathcal{O}(1/\epsilon)$	Stable
Gradient Descent Fixed	$\mathcal{O}(1/\epsilon)$	Unstable
Nesterov Acceleration Varying	$\mathcal{O}(1/\sqrt{\epsilon})$	Unstable
Nesterov Acceleration Fixed	$\mathcal{O}(1/\sqrt{\epsilon} \cdot \log(1/\epsilon))$	Unstable
HT	$\mathcal{O}(1/\sqrt{\epsilon} \cdot \log(1/\epsilon))$	**Stable**

NEW SOLUTIONS:

ACCELERATED PERFORMANCE

- High-order tuner

ROBUST LEARNING

- Sub-Gaussian spectral lines

REAL-TIME MACHINE LEARNING

- Integration with reinforcement learning

STABIITY AND SAFETY

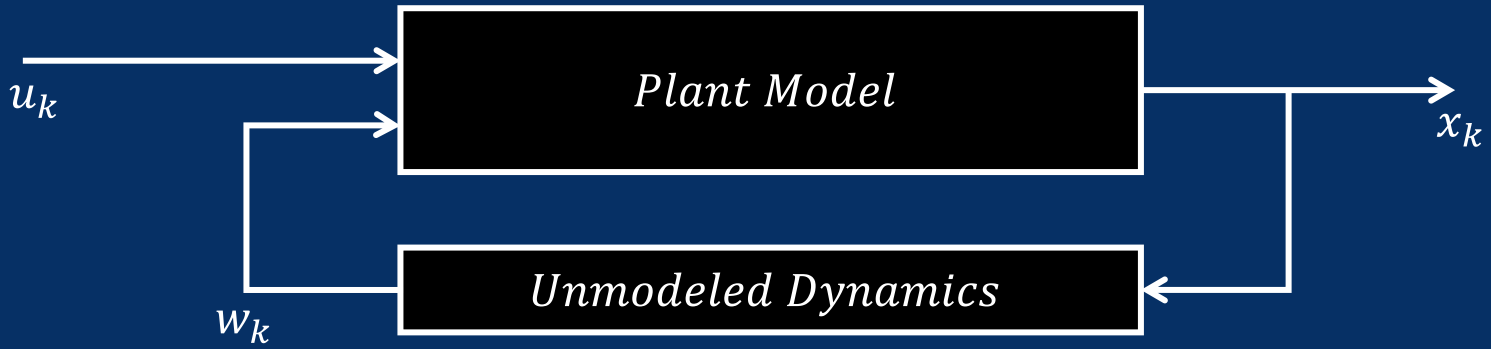
- Adaptation and Calibrated Control Barrier Functions

Consider a standard LQR problem in the presence of unmodeled dynamics:

$$x_{k+1} = A_* x_k + B_* u_k + w_k + \eta_k, \quad w_k = g(x_0, w_0, \dots, w_{k-1}, u_0, \dots, u_k)$$

w_k: unmodeled dynamics; η_k: measurement noise

Unknown A_* and B_*



Goal:

- Learn A_* and B_*
- Determine an LQR controller: $\min_{\mathrm{u}} \; J: \sum_k (x_k^T Q x_k + u_k^T R u_k)$
- Develop a non-asymptotic approach

* A. Sarker, P. Fisher, J.E. Gaudio, and A.M. Annaswamy, "Parameter Estimation Bounds Based on the Theory of Spectral Lines." J. Artificial Intelligence, vol. 316, March 2023.

Definition 1 (Sub-Gaussian Spectral Line).

A stochastic sequence $\{u_k\}_{k\geq k_0}$ is said to have a sub-Gaussian spectral line from i to $i+S$ at a frequency ω_0 of amplitude $\bar{u}(\omega_0)$ and radius R if

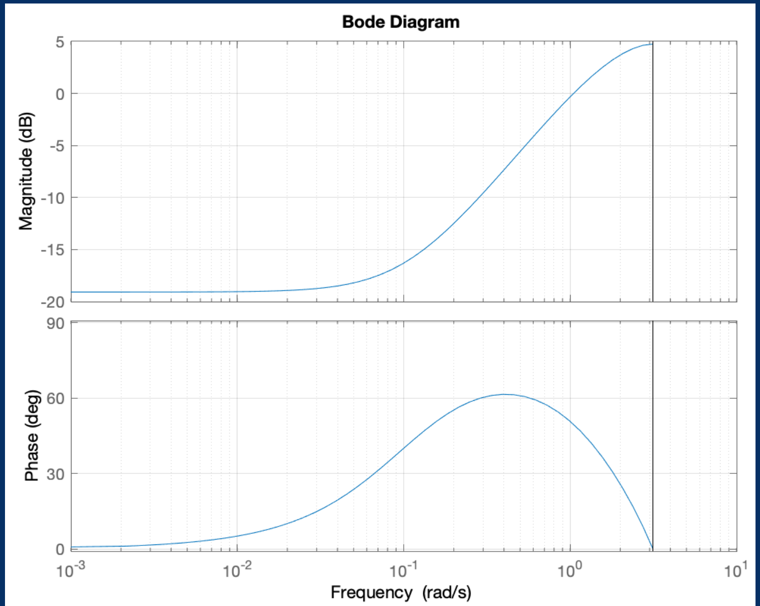
$$\frac{1}{S+1}\sum_{k=i}^{i+S} u_k e^{-j\omega_0 k} - \bar{u}(\omega_0) \sim \mathrm{subG}(R^2/(S+1))\,.$$

The definition above admits a natural decoupling by which we can use $\bar{u}(\omega_0)$ to apply tools from adaptive control, and the variance proxy of the sub-Gaussian noise to make claims with high probability.

* A. Sarker, P. Fisher, J.E. Gaudio, and A.M. Annaswamy, "Parameter Estimation Bounds Based on the Theory of Spectral Lines." arXiv preprint arXiv:2006.12687.

A Spectral Lines-Based Algorithm*

- Our approach: learn from a deterministic input with chosen frequency content
- Idea: choose frequency content to keep w_k small



Algorithm 2 Example of LQR Control with Sub-Gaussian Spectral Lines

1: **Require:** Frequency Constraint Set $\mathcal{F}$, Stabilizing Controller $K^{(0)}$, failure probability $\delta \in (0,1)$, Amplitude Constraint M, Cost Matrices Q, R

2: **for** $i = 0, 1, 2, \ldots$ **do**

3: **Set** Epoch Time $T_i = C(A_*, B_*, Q, R, \delta) \times 2^i$, where $C(A_*, B_*, Q, R, \delta)$ is a constant function of underlying system parameters, Amplitude Constraint $\overline{M}^{(i)} = MT_i^{1/4}$

4: **Choose** Distinct $f_1^{(i)}, \ldots, f_{\lceil d/2 \rceil}^{(i)} \in \mathcal{F}$, $M_1^{(i)}, \ldots, M_{\lceil d/2 \rceil}^{(i)}$, such that $\frac{1}{2}\overline{M}^{(i)} \leq M_j^{(i)} \leq \overline{M}^{(i)}$.

5: **for** $k = \sum_{\ell=0}^{i-1} T_\ell + 1, \ldots, \sum_{\ell=0}^{i-1} T_\ell + T_i$ **do**

6: **Define** $u_{spec,k} = \sum_{j=1}^{\lceil d/2 \rceil} M_j^{(i)} \cos(2\pi f_j^{(i)} k)$;

7: **Assign** $u_k = K^{(i)} x_k + u_{spec,k}$.

8: **Receive** $x_{k+1} = A_* x_k + B_* u_k + w_k + \eta_k$

9: **end for**

10: **Estimate**

$$(\widehat{A}, \widehat{B}) = \underset{A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}}{\operatorname{argmin}} \sum_{k=\sum_{\ell=0}^{i-1} T_i + 1}^{\sum_{\ell=0}^{i-1} T_\ell + T_i} \| x_{k+1} - A x_k - B u_k \|_2^2$$

11: **Set**

$$K^{(i+1)} = -(R + \widehat{B}^\top \widehat{P} \widehat{B})^{-1} \widehat{B}^\top \widehat{P} \widehat{A}$$

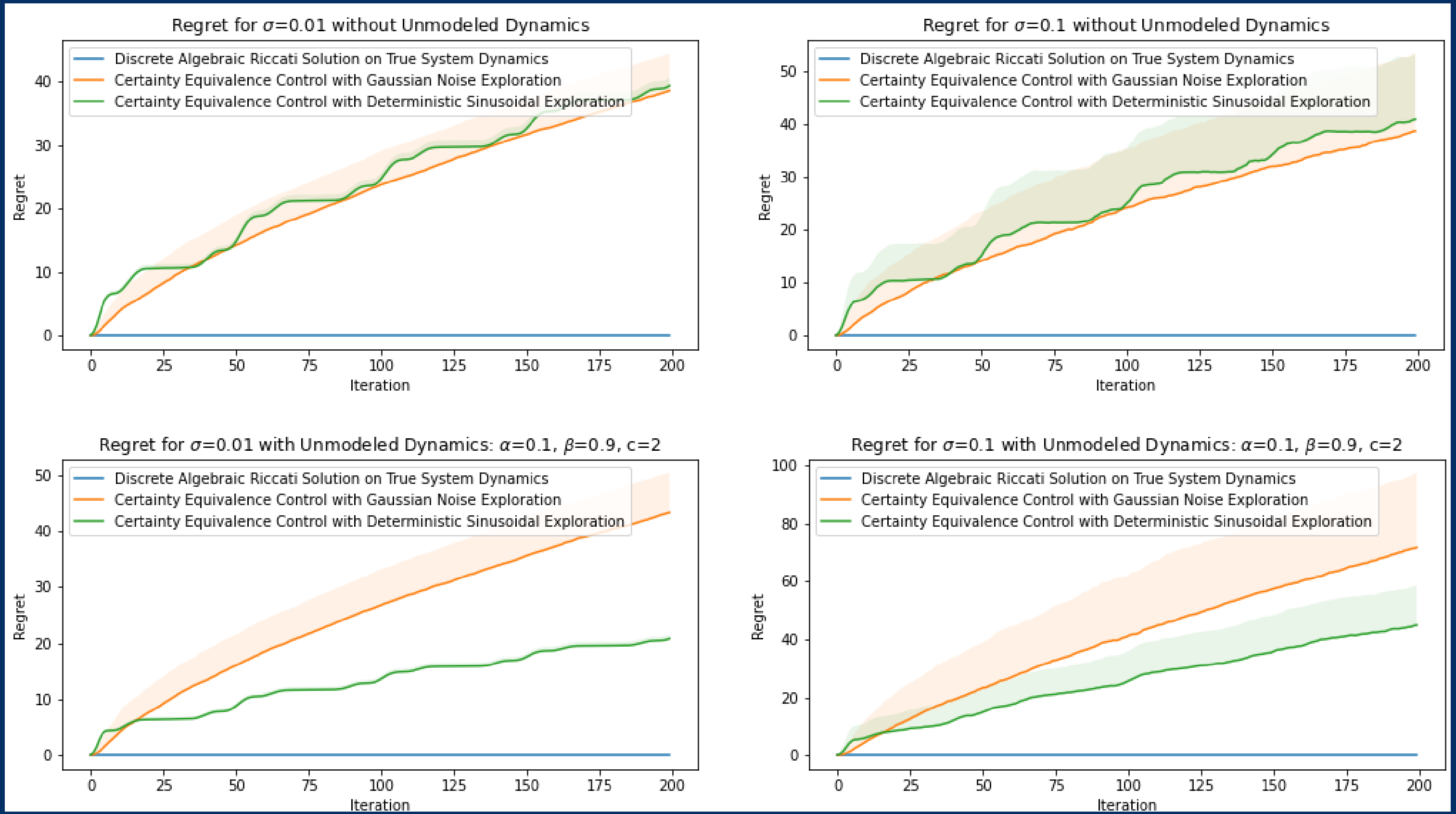
for $\widehat{P}$ which satisfies

$$\widehat{P} = \widehat{A}^\top \widehat{P} \widehat{A} - \widehat{A}^\top \widehat{P} \widehat{B} (R + \widehat{B}^\top \widehat{P} \widehat{B})^{-1} \widehat{B}^\top \widehat{P} \widehat{A} + Q$$

12: **end for**

* A. Sarker, P. Fisher, J.E. Gaudio, and A.M. Annaswamy, “Parameter Estimation Bounds Based on the Theory of Spectral Lines.” J. Artificial Intelligence, vol. 316, March 2023.

- 3^{rd}-order LTI system simulated with two noise-to-signal ratios (σ)
- Unmodeled dynamics $g(\cdot)$ were given by a 1^{st}-order nonlinear high-pass filter
- System was modeled with and without unmodeled dynamics
- Regrets of Algorithms 1 and 2 are comparable without unmodeled dynamics:
- With unmodeled dynamics, Algorithm 2 outperforms Algorithm 1:



* A. Sarker, P. Fisher, J.E. Gaudio, and A.M. Annaswamy, "Parameter Estimation Bounds Based on the Theory of Spectral Lines." arXiv preprint arXiv:2006.12687.

NEW SOLUTIONS:

ACCELERATED PERFORMANCE

- High-order tuner

ROBUST LEARNING

- Sub-Gaussian spectral lines

REAL-TIME MACHINE LEARNING

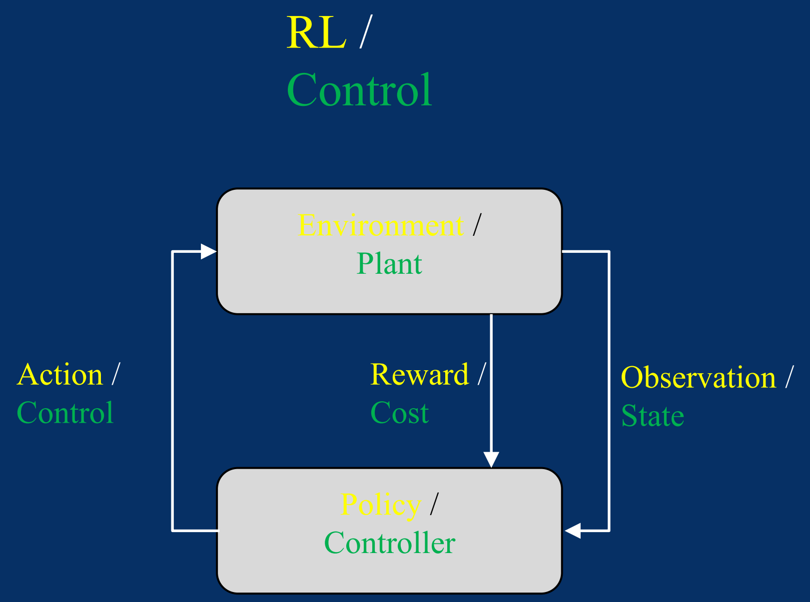
- Integration with reinforcement learning

STABIITY AND SAFETY

- Adaptation and Calibrated Control Barrier Functions

- Reinforcement Learning
  - Training in Simulation
  - Approximate solutions to difficult optimal control problems

- Adaptive control
  - Online learning
  - Solves constrained class of problems
  - Real time
  - Applicable in continuous and discrete-time



An online policy: AC-RL

- Idea: Modify the trained policy output $u_r \rightarrow u$ so that the true model tracks the reference model

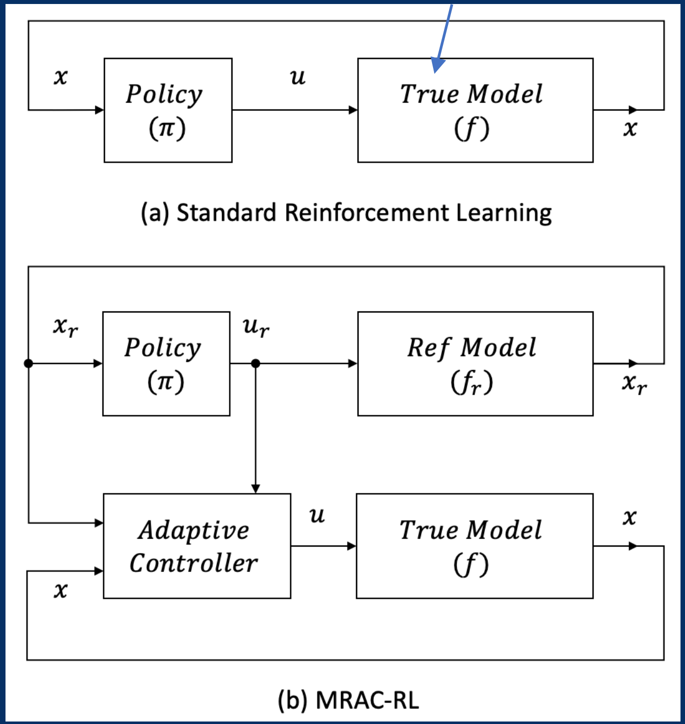
$$\dot{x}_r = f_r(x_r, u_r); \quad (u_r = \pi(x_r))$$
$$\dot{x} = f(x, u)$$

AC-RL:

$$u = u_r + g\left(e, \widehat{\Theta}\right) \qquad e = x - x_r$$
$$\dot{\widehat{\Theta}} = \Gamma_\zeta \nabla L(e, \dot{e})$$

- Globally stable for a class of $f(x, u)$*
- Leads to $\lim_{t \to \infty} ||e(t)|| = 0$
- Elements of $g\left(e, \widehat{\Theta}\right)$ come from the offline policy and the plant model $f(x, u)$

subjected to parametric changes



(a) Standard Reinforcement Learning

(b) MRAC-RL

Annaswamy *et al*. "Integration of adaptive control and reinforcement learning for real-time control and learning." *IEEE Transactions on Automatic Control* (2023).

Quadrotor: Hover Using Adaptive Control*

* Dydek, Zachary T., Anuradha M. Annaswamy, and Eugene Lavretsky. "Adaptive control of quadrotor UAVs: A design trade study with flight evaluations." *IEEE Trans. CST,* vol. 21 (2012)

- Autonomous landing of quadrotor on a moving platform
- Parameter uncertainties (25%)
- Loss of Effectiveness (50-75%)
- Success:
  - $|\Delta z| \leq 5cm$ **and**
  - $|\Delta xy| \leq 25cm$ **and**
  - $|\phi|, |\theta| \leq 10°$ **and**
  - $|v_{xy}| \leq 50cm/s$ **and**
  - $|v_z| \leq 10cm/s$
- Failure:
  - $\Delta z \leq 0$ **or**
  - Timeout
- Goal: Succeed ASAP
- Assumptions:
  - Full state feedback
  - Landing pos + vel measurable

Annaswamy *et al*. "Integration of adaptive control and reinforcement learning for real-time control and learning." *IEEE Transactions on Automatic Control* (2023).

Quadrotor: Land on a moving platform

With 50% Loss of Effectiveness mid-flight

Pure RL

AC-RL

RL Success Rate	AC-RL Success Rate	LOE
94%	– –	0%
71%	95%	10%
28%	81%	25%
4%	47%	50%
0%	11%	75%

Significant improvement with AC-RL over pure RL

Quadrotor crashes

Quadrotor: Land on a moving platform

With parametric uncertainties mid-flight, comparison with additional-training in RL through Domain Randomization (DR-RL)

DR-RL

AC-RL

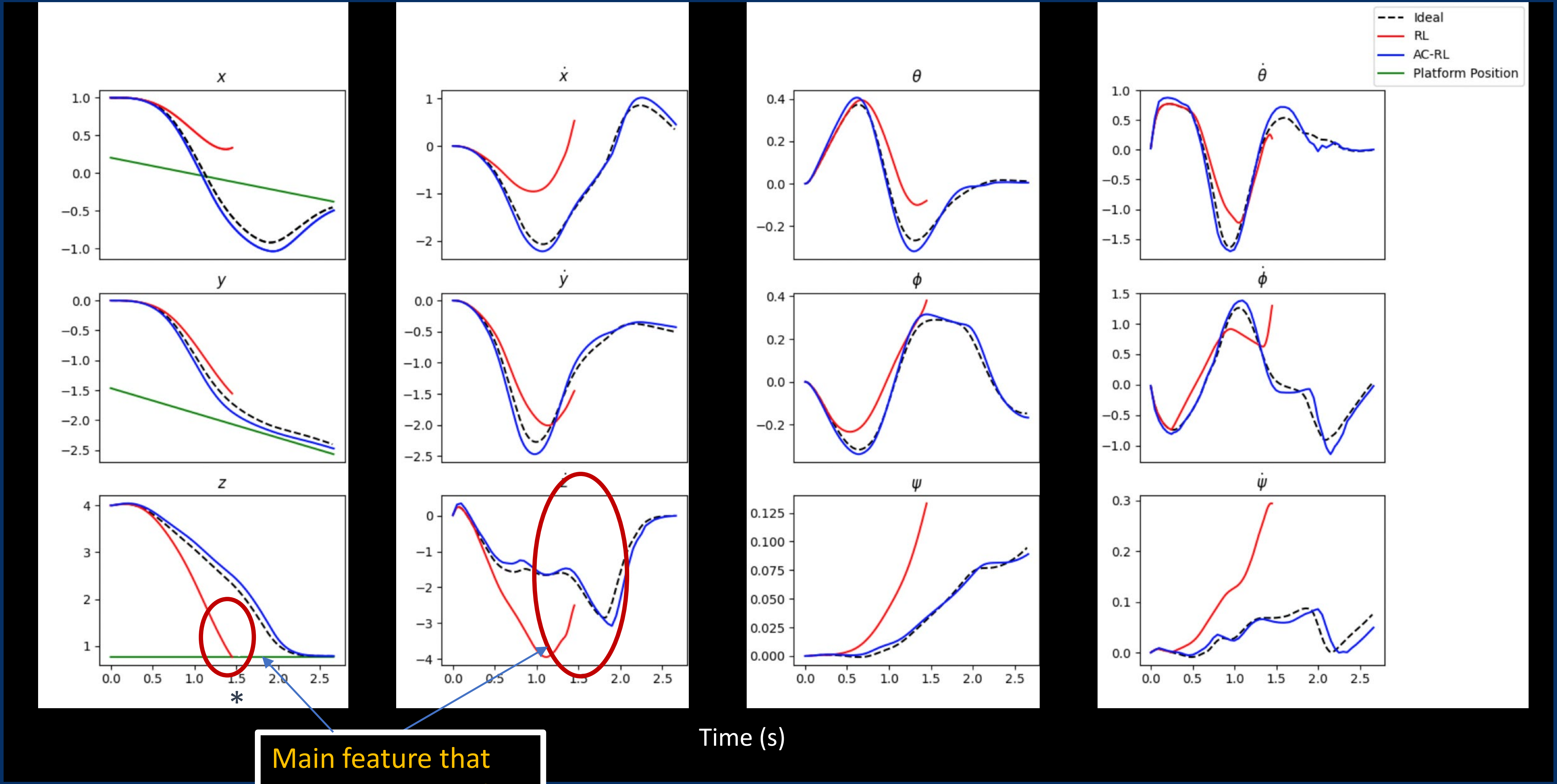
$\pm 25\%$ PARAMETRIC UNCERTAINTY RESULTS

Algorithm	Results	
	Success Rate	**Avg. Success Time**
RL	48%	7.5s
AC-RL	**82%**	**3.5s**
DR-RL	74%	8.9s

- Additional improvement with AC-RL over DR-RL
- Does not require either additional training or computational effort

RL Success Rate	AC-RL Success Rate	LOE
94%	−−	0%
71%	95%	10%
28%	81%	25%
4%	47%	50%
0%	11%	75%

Why is AC-RL successful?



Main feature that gives AC-RL an edge

* Quadrotor crashes

NEW SOLUTIONS:

ACCELERATED PERFORMANCE

- High-order tuner

ROBUST LEARNING

- Sub-Gaussian spectral lines

REAL-TIME MACHINE LEARNING

- Integration with reinforcement learning

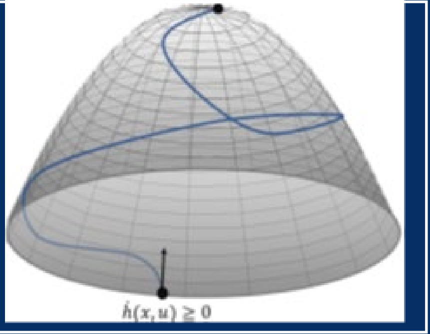
STABIITY AND SAFETY

- Adaptation and Calibrated Control Barrier Functions

Performance and Safety in Adaptive Systems

Safety:

Guarantee that $x(t)$ stays within a set C for any $t \geq 0$.



Real-time uncertainties, anomalies

x_{cmd}

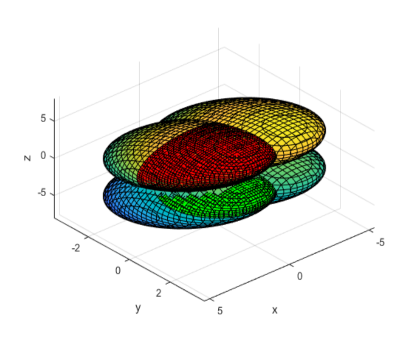
Robust Adaptive Controller

u

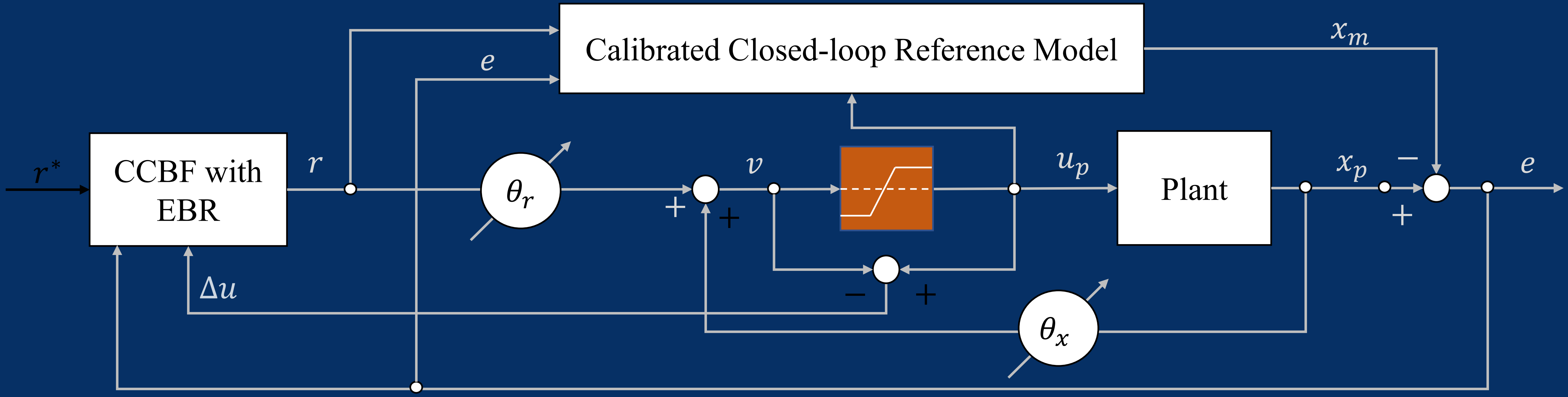
Control surfaces and thrust

Performance:

Guarantee that $x(t)$ follows a command signal $x_{cmd}(t)$.



- Adaptive controller accommodates uncertainties and magnitude limits.
- Constraints are met using a calibrated control barrier function (CCBF) for a reference model and an error-based relaxation (EBR).

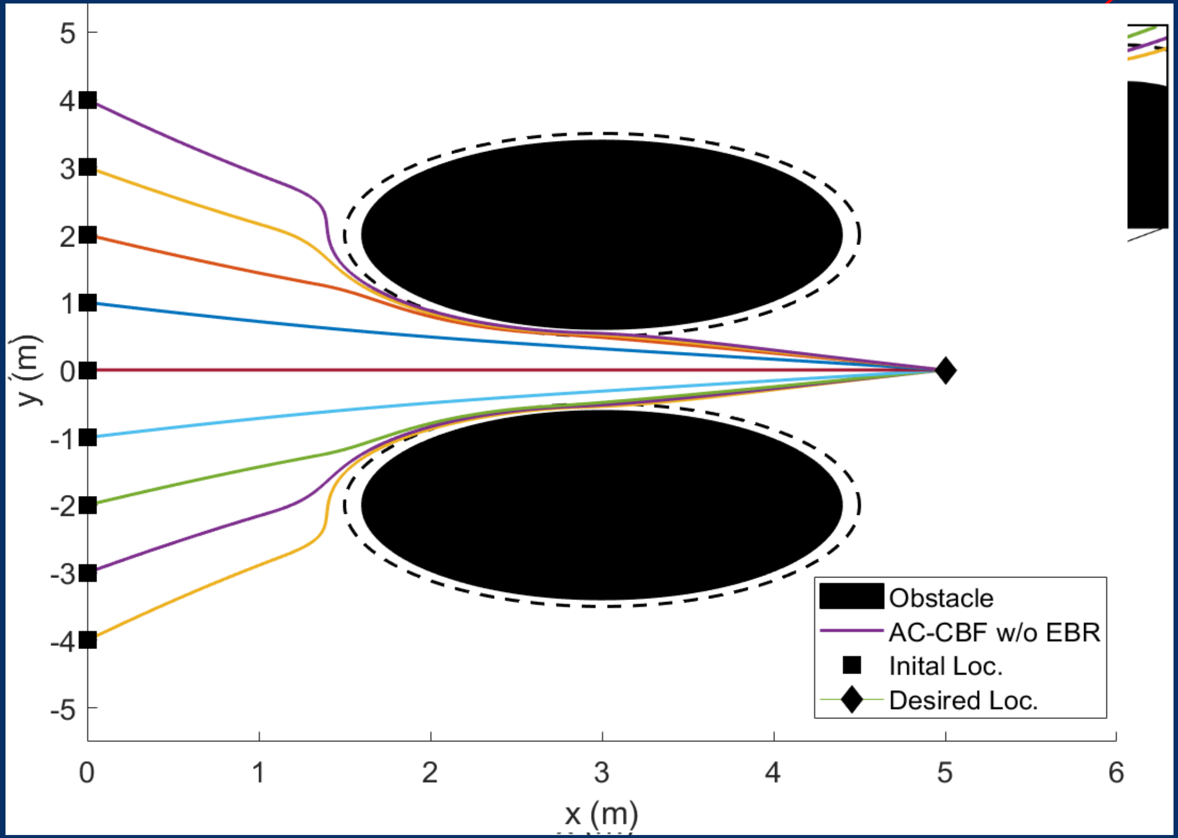


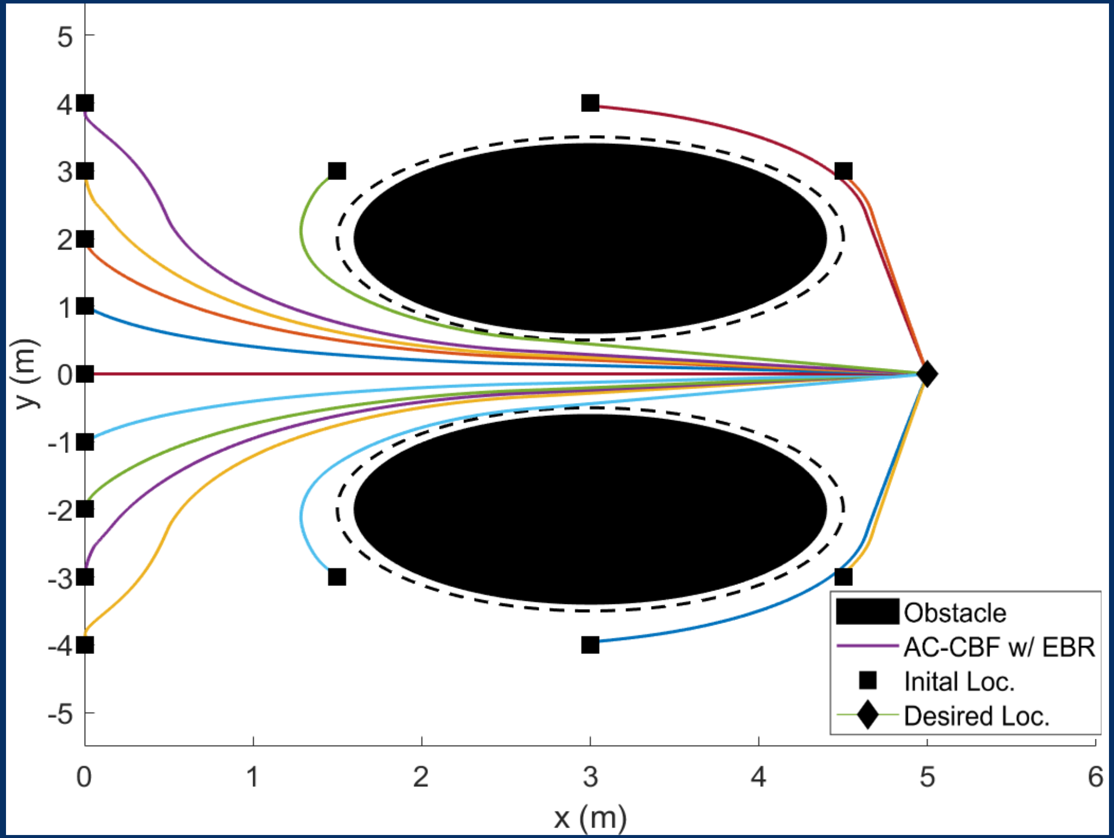
* J. Autenrieb and A.M. Annaswamy, "Safe and stable adaptive control with learning for a class of dynamic systems," CDC 2023.

A new adaptive algorithm

Example case 1: Obstacle avoidance

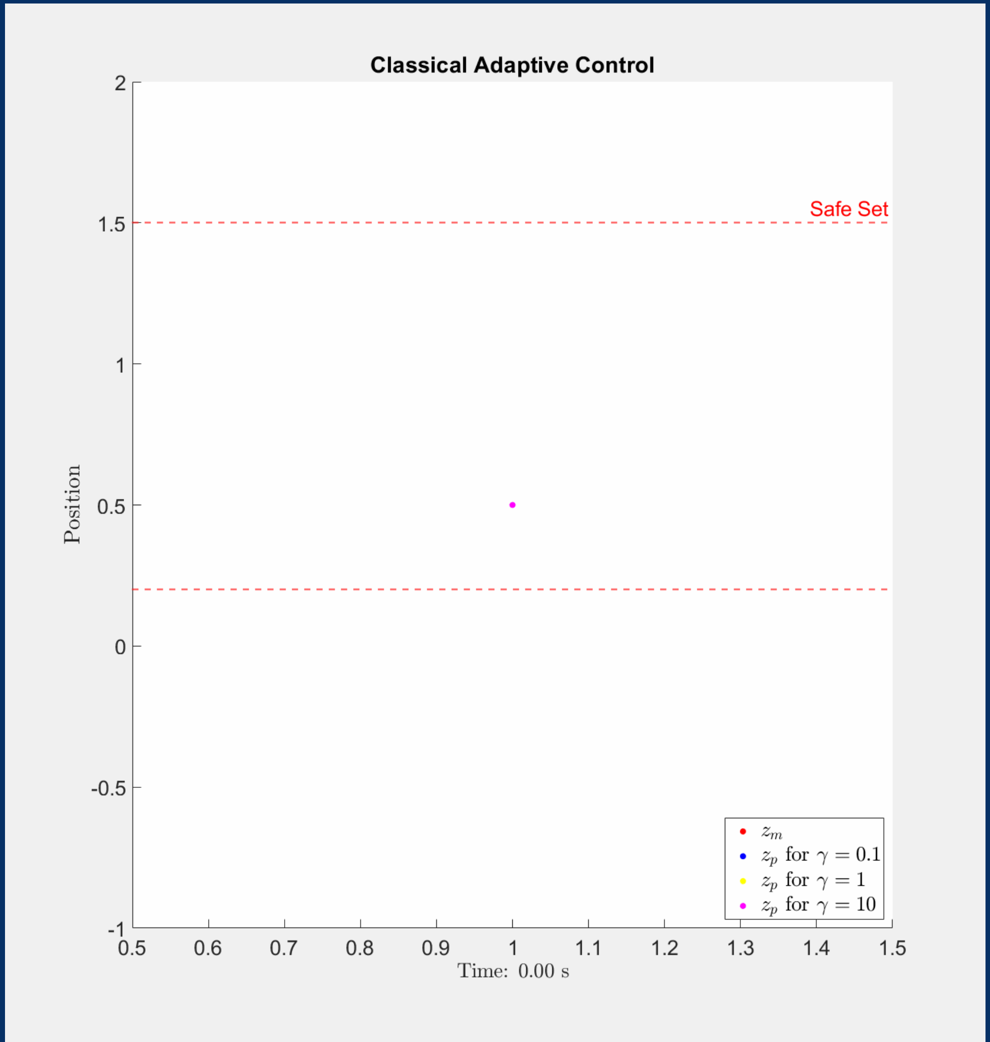
Constraint violation



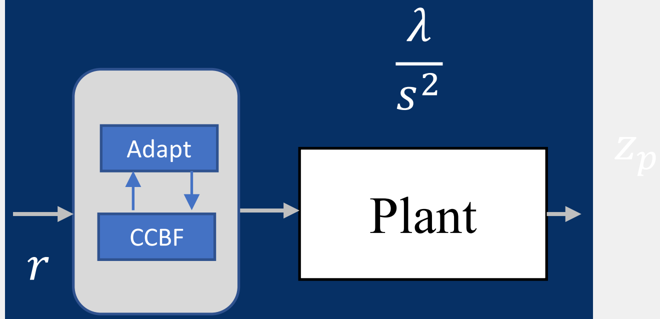


EBR: Error-based Relaxation

Example 2: A double integrator (using Simulink Desktop Real-time Emulator)



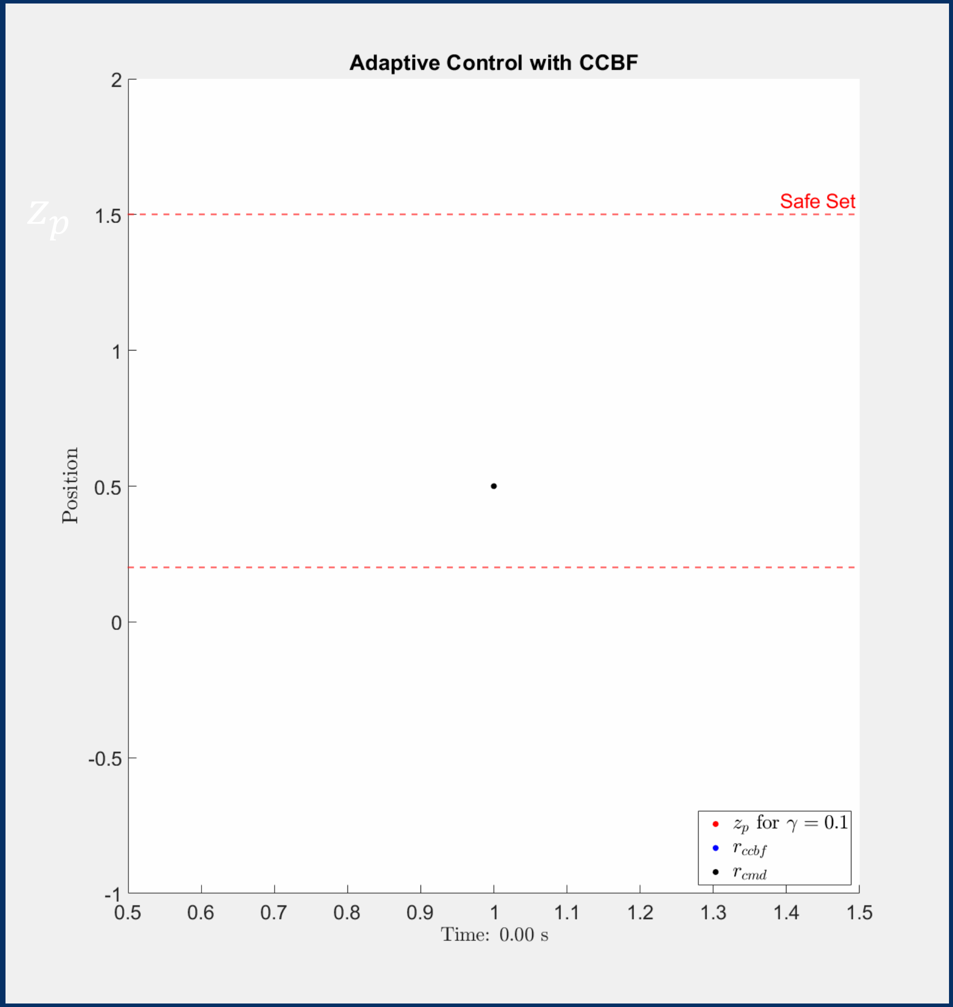
Stability



λ: loss of effectiveness

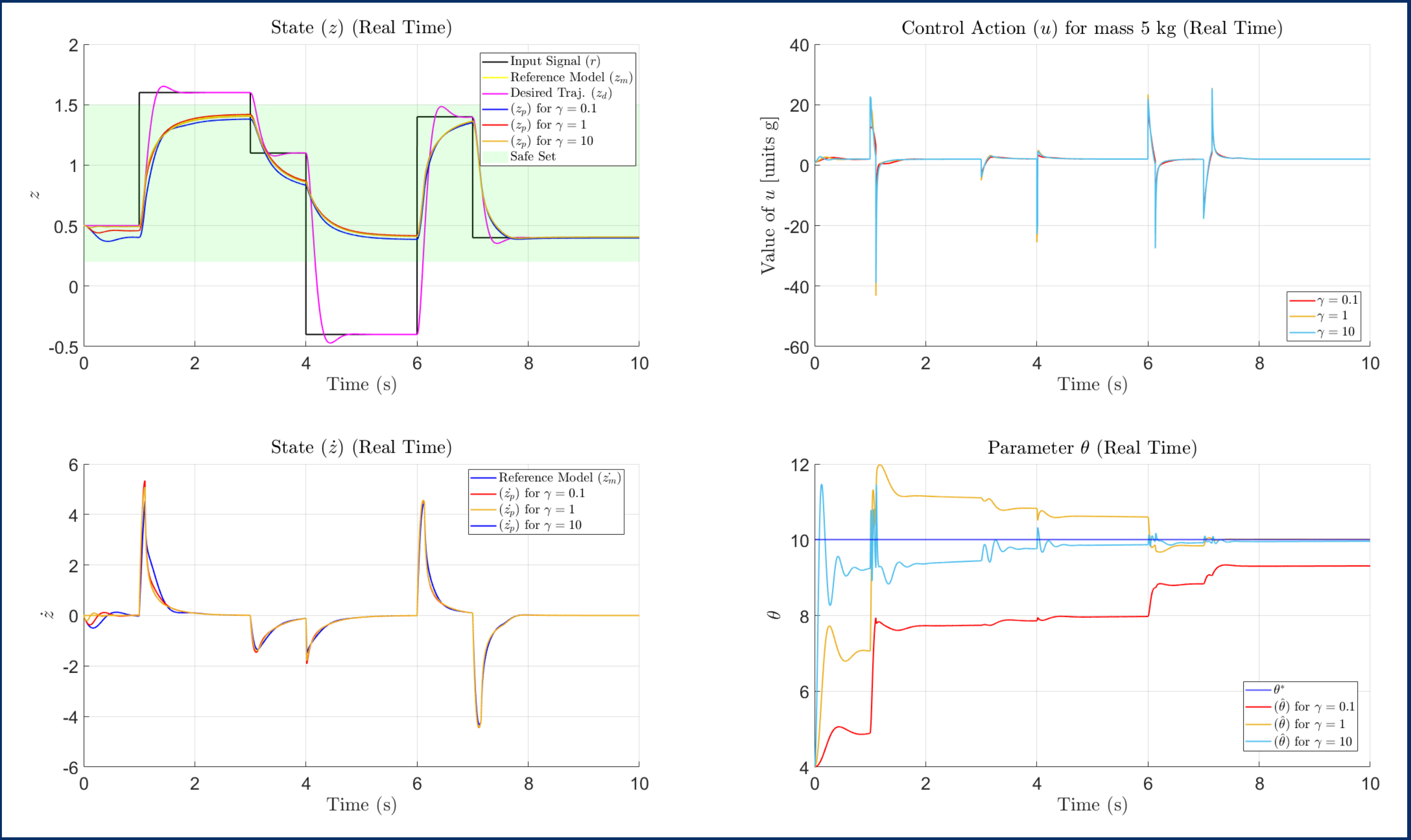
γ: learning gain

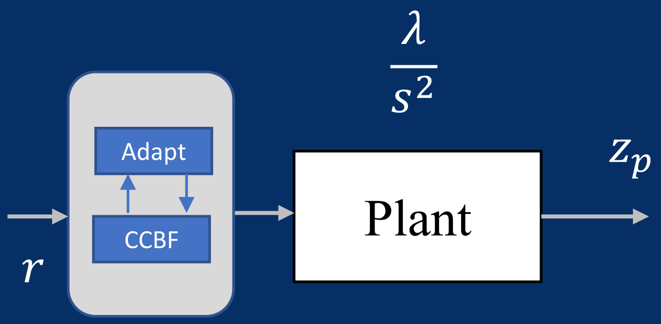
z_m: safe target



Safety and Stability

Example 2: A double integrator (using Simulink Desktop Real-time Emulator)



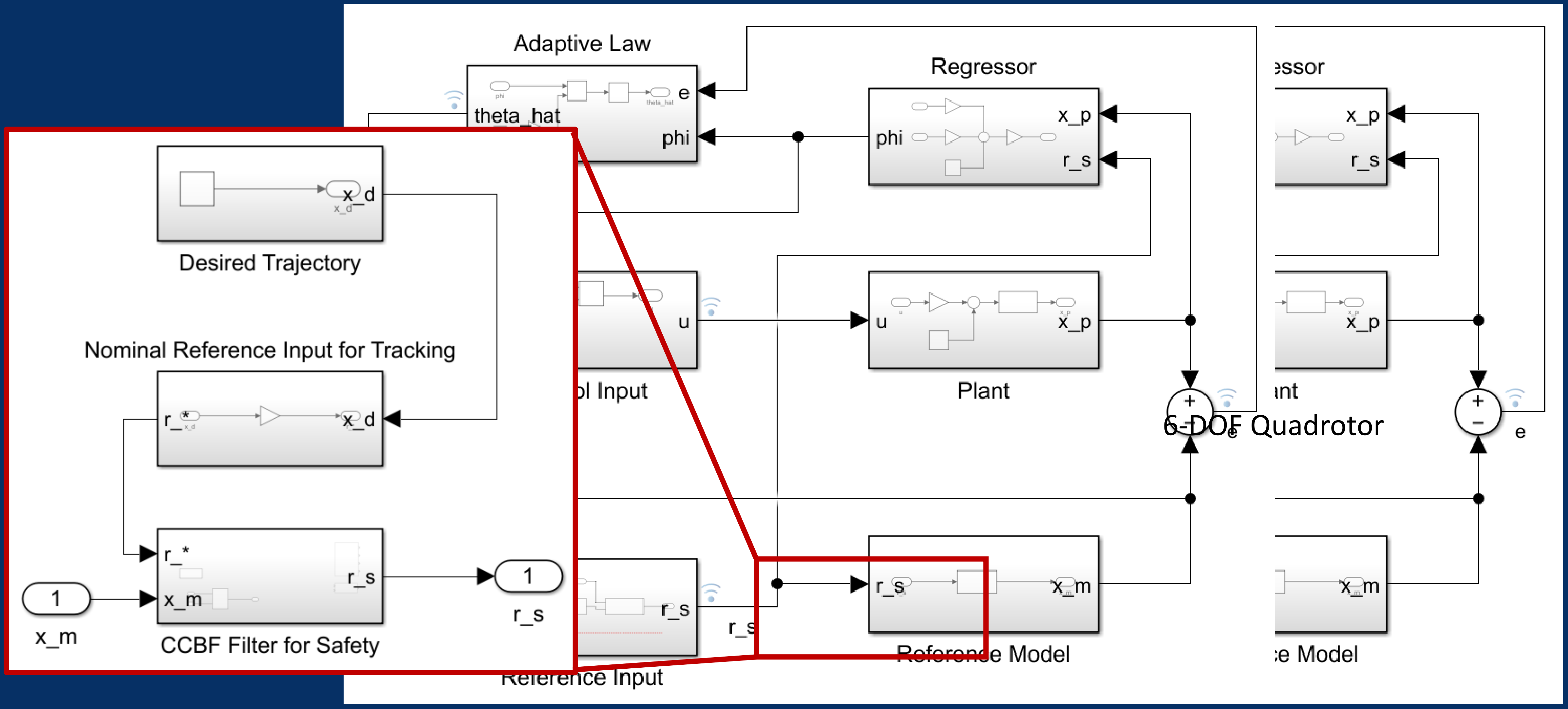


λ: loss of effectiveness

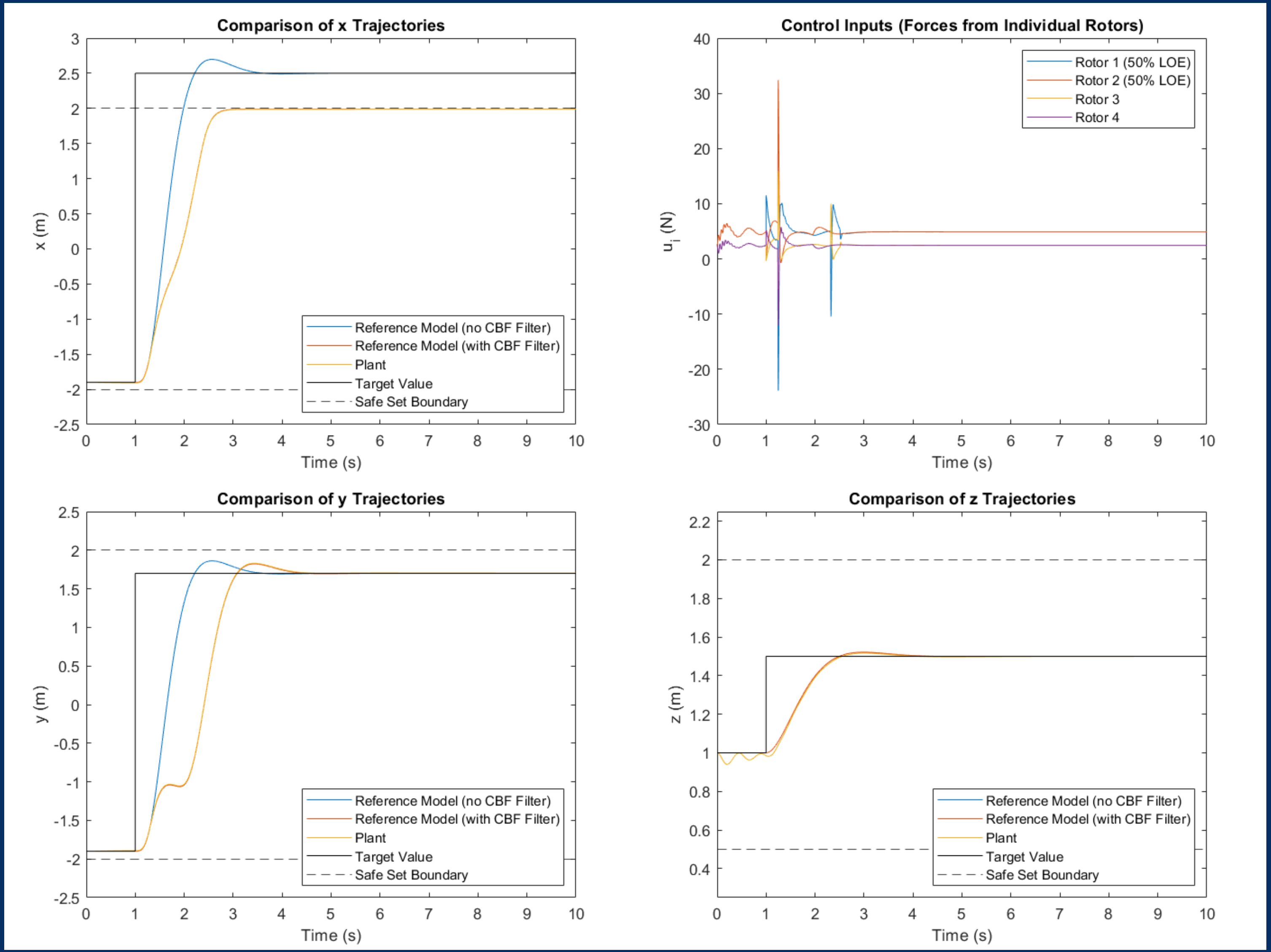
γ: learning gain

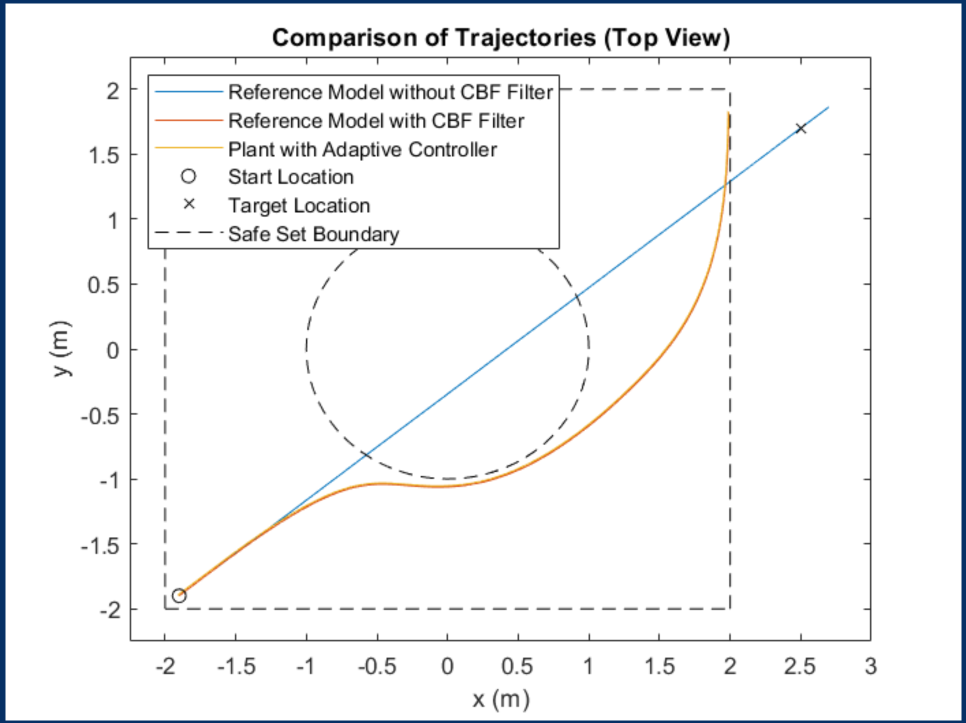
z_m: safe target

Example 3: A 6-DOF Quadrotor



Example 3: A 6-DOF Quadrotor





Summary

- Learning in Adaptive Systems
  - Adaptive Estimation and Adaptive Control
  - Error Models & Learning rules
  - Stability framework – Imperfect Learning
  - Persistent Excitation – Learning with guarantees
- Machine Learning
  - Neural Networks
  - Reinforcement Learning
- New Solutions
  - High-order Tuners – towards accelerated performance
  - Sub-Gaussian spectral lines – towards robust learning
  - Integration of RL and Adaptive Control – towards real-time machine learning
  - Safety and Stability – Adaptation with Calibrated CBF

Some Takeaways

- Learning
  - Occurs at multiple time-scales
- Safety-critical Systems
  - Adapt first – requires a stability+adaptive control framework
  - Guarantees with imperfect learning are essential
  - Learning comes with hindsight
- Towards fully autonomous systems
  - Real-time decision making tools – with guarantees
  - Combination of adaptive control and ML needed
- "Control for Learning" needs to be addressed
  - For decision-making under fast time-scales

Thank you!

aanna@mit.edu