Workshop for the 2021 5th IEEE Conference on Control Technology and Applications (CCTA)

San Diego, California, August 8-11 , 2021

Multi-Vehicle and Assured Autonomous Control for Aerospace Applications

- Organized by the IEEE CSS Technical Committee on Aerospace Controls -

The intersection between Machine Learning & Guidance, Navigation, and Controls (GNC)

Sunday, August 8, 2021

Dr. Heather Hussain, BR&T GNC&A *(Presenter)*
Dr. Joseph Gaudio, AFS Autonomy Research Division *(Presenter)*
Dr. James Paduano, AFS Autonomy Tech Area Lead
Anubhav Guha, MIT PhD Candidate, prior AFS Autonomy Research Division *(Presenter)*
John Piotti, AFS Autonomy Research Division
Bradley Lan, Morse Corp. GNC, prior AFS Autonomy Research Division

Abstract: "The Intersection between Machine Learning & GNC"

CCTA TCAC Workshop: Multi-Vehicle and Assured Autonomous Control for Aerospace Applications

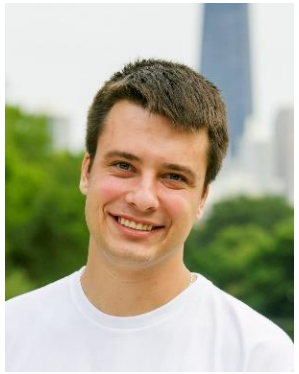
- *As machine learning methods become more prevalent in guidance, navigation, controls, and autonomy (GNC&A), problems of a dynamical nature will increasingly need to be considered. The dynamical nature of these problems may include regressors that are time-varying, necessitating new algorithms in machine learning approaches as well as real-time decision making in the presence of uncertainties using adaptive control approaches. Problems of stability, fast learning with analytical guarantees, and constrained nonlinear systems have to be simultaneously addressed. Some of these problems have to be addressed from a machine learning perspective, while others have to be dealt with using adaptive control approaches. Throughout, analytical guarantees must be considered in order to apply machine learning for decision making in real-time. These theoretical guarantees are necessary to address advanced guidance and control challenges for next generation aerial vehicles. Recently, under the DARPA AlphaDog fight Trials (ADT) program for within-visual-range dogfighting, Aurora Flight Sciences developed competitive AI agents that are robust, trustable and capable. The approach combined reinforcement and machine learning, expert-systems and rule-based methods, and principles from guidance, navigation and control (GNC). Holistically fusing these domains enabled the AI to incorporate domain knowledge and results from traditional disciplines while still leveraging the latest tools from machine/reinforcement learning.*

- *Dr. Heather Hussain, BR&T GNC&A (Presenter)*
- *Dr. Joseph Gaudio, AFS Autonomy Research Division (Presenter)*
- *Dr. James Paduano, AFS Autonomy Tech Area Lead*
- *Anubhav Guha, MIT PhD Candidate, prior AFS Autonomy Research Division (Presenter)*
- *John Piotti, AFS Autonomy Research Division*
- *Bradley Lan, Morse Corp. GNC, prior AFS Autonomy Research Division*

Biographies (Presenters)

CCTA TCAC Workshop: Multi-Vehicle and Assured Autonomous Control for Aerospace Applications

- ***Dr. Heather Hussain*** *received the B.S. degree and M.S. degree in mechanical engineering from the Rochester Institute of Technology, Rochester, NY, USA, in 2012, and the Sc.D. degree in mechanical engineering at the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA in 2017. Her work experience comprises several internships spanning the aerospace and consumer electronics industries– namely, in Product Design at Apple Inc., as a research Scholar at the Munitions Directorate of the Air Force Research Laboratory, and her work in the design and development of verifiable adaptive flight control systems at The Boeing Company. Ms. Hussain's doctoral research at MIT was sponsored by the Boeing Strategic University Initiative under the direction of Dr. Eugene Lavretsky and Dr. Anuradha Annaswamy. Ms. Hussain joined BR&T's Guidance, Navigation, Control, and Autonomy (GNC&A) group in September 2017. Her research interests lie in adaptive control theory, particularly with applications in aerospace. Ms. Hussain is a member of AIAA and IEEE.*



- ***Dr. Joseph E. Gaudio*** *received the B.S. degree in mechanical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2016, and the Ph.D. degree in mechanical engineering at the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA in 2020. His work experience comprises several internships at Boeing Research and Technology (BR&T) and The Boeing Phantom Works. His doctoral research at MIT was sponsored by the Air Force Research Laboratory and the Boeing Strategic University Initiative under the direction of Dr. Anuradha Annaswamy. Gaudio joined the Autonomy Research Division at Aurora Flight Sciences, a Boeing Company in June 2020. His research interests lie in nonlinear, adaptive, and machine learning-based guidance, control, and online optimization. He is a member of IEEE.*

- ***Anubhav Guha*** *received a B.S in Mechanical Engineering & a B.S in Physics from the Massachusetts Institute of Technology in 2018. From 2018-2020 he worked as an autonomy engineer at Aurora Flight Sciences (AFS), where he served as the program manager and principal investigator for the AFS DARPA AlphaDogfight program. In 2020 Mr. Guha joined MIT's Active Adaptive Control Laboratory as a PhD candidate under the supervision of Dr. Anuradha Annaswamy. His research interests lie in adaptive control theory, reinforcement learning, and optimal planning & control.*

Outline

CCTA TCAC Workshop: Multi-Vehicle and Assured Autonomous Control for Aerospace Applications

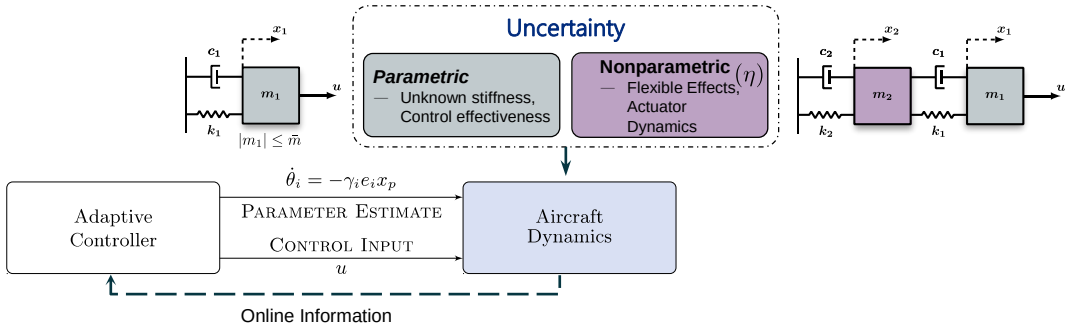
- **Introduction**
  - Control Challenges for Future Air Vehicle Platforms
- ~~**Robust Adaptive Control for High Speed Platforms (Dr. Heather Hussain)**~~
- **The intersection between Machine Learning & Adaptation (Dr. Joseph Gaudio)**
  - Common Parameter Update Laws and Error Models
  - Common Modifications to Parameter Update Laws
  - Common Concepts and Tools
- **Deep Reinforcement Learning Methods for Air-to-Air Combat & Control (Anubhav Guha)**
  - Timeline of Artificial Intelligence in Games
  - DARPA Alpha Dogfight Trials
  - Key Components of Reinforcement Learning
  - Curriculum for Reinforcement Learning
  - Final Results from AlphaDogfight Trials
  - Key Takeaways
- **Summary & Questions**

Introduction

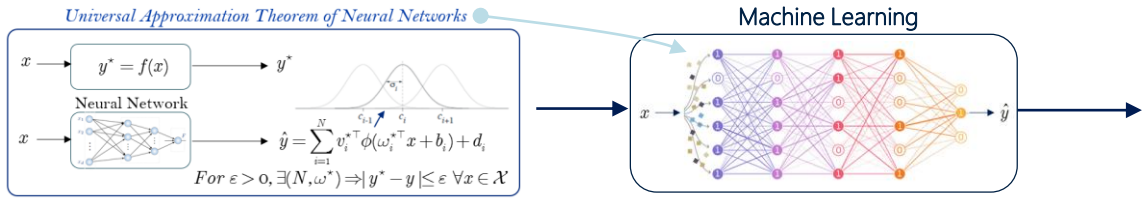
Motivation for Adaptive and Learning Based Systems

- *With the advent of each next generation technology, demands for a rapidly reconfigurable control system yielding invariant performance under increasingly unknown or widely varying operating conditions becomes crucial.*
  - *e.g. High Speed platform challenges [1] "HIFiRE 6: Overview and Status Update 2015," 20th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, 2015. [2] Robust and Adaptive Control for High Speed Vehicles, ACGSC Meeting 119, Dayton, OH, 2017*

- *Adaptive control theory is a mature control discipline that allows for real-time compensation of parametric uncertainties and changes in system dynamics*
  - *Premise: Adapt system parameters to provide a vehicle response that more closely follows the reference model*



- *Machine Learning, a largely data driven process where learning typically occurs offline, is also key in addressing the control challenges of future autonomous systems.*



- *Objective: Obtaining efficient, stable, & robust learning systems for GNC-enabled autonomy.*

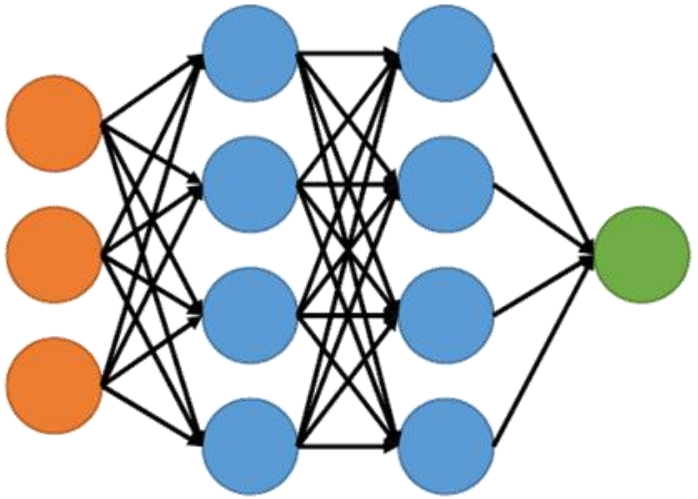
Intersection between Machine Learning & Adaptation

Dr. Joseph Gaudio, AFS Autonomy Research Division
(presentation based on PhD thesis, prior to joining AFS)

Online/Adaptive Learning in Machine Learning Problems

- ▷ Most machine learning algorithms assume features are constant
- ▷ Need for algorithms which explicitly account for feature time variation (step changes, continuously varying)
- ▷ Extend learning algorithms to online sequential decision making systems with limited computation
- ▷ Increased need for continual/lifelong learning systems
- ▷ Standard machine learning systems are increasingly becoming feedback systems
- ▷ Requirement for provably stable algorithms

Employ techniques from adaptive control theory to enable safe and fast learning for systems with time-varying regressors



Training Neural Networks

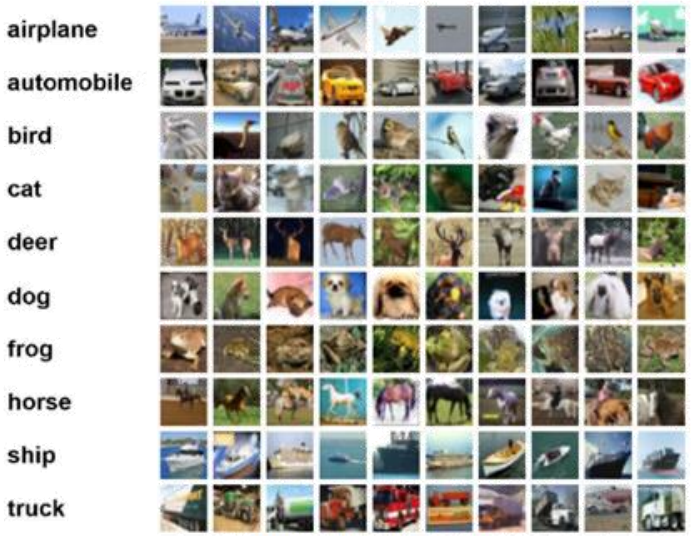


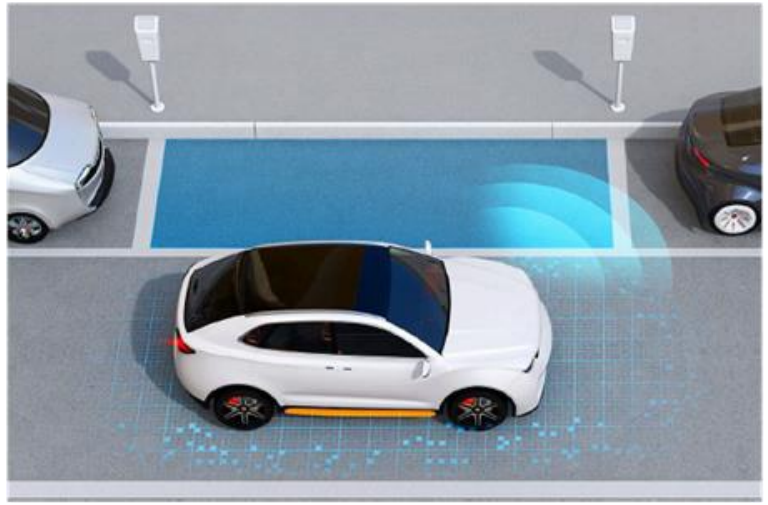
Image Classification
https://www.cs.toronto.edu/~kriz/cifar.html

Online/Adaptive Learning and Control in Dynamical Systems

- ▷ Adaptive identification and control is becoming increasingly prevalent in society (autonomous vehicles, robotics)
- ▷ Dynamics of the learning process need to be analyzed alongside plant dynamics
- ▷ Physical systems have input constraints which need to be accounted for during the learning process
- ▷ Long-term learning enabled by parameter convergence
- ▷ Requirement for provably stable algorithms

Employ techniques inspired by machine learning within an adaptive control framework to enable safe, fast learning and control in dynamical systems

http://aaclab.mit.edu/previous-autonomous-flight-systems.php



Autonomous Vehicles
https://spectrum.ieee.org/image/MzIzMjg2NA.jpeg

Intersection between Machine Learning & Adaptation:
Common Parameter Update Laws and Error Models

Linear Regression Models

Known	Unknown
y : output	θ^* : parameter
ϕ : feature, regressor	

$$\text{Model} \quad y = \phi^T \theta^*$$

$$\text{Prediction} \quad \hat{y} = \phi^T \theta$$

Goal: Learn θ

Squared error (loss):

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

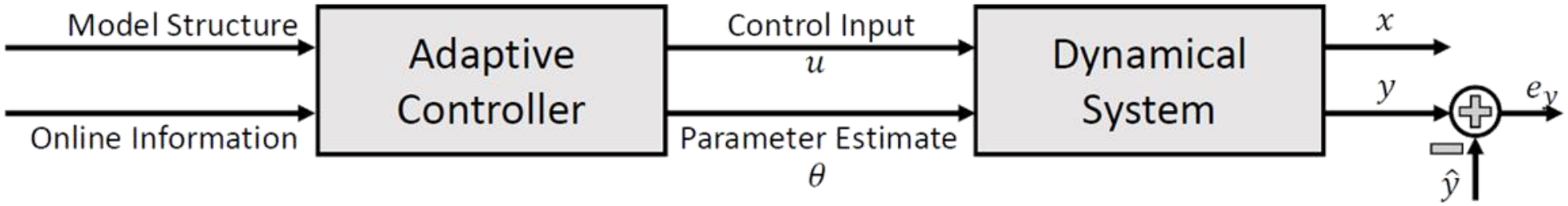
For large systems we can employ an iterative method: step size is γ

$$\theta_{k+1} \leftarrow\!\shortmid \theta_k - \gamma \nabla_\theta L(\theta_k)$$

Control

Outputs can be related to features/data through a dynamical system

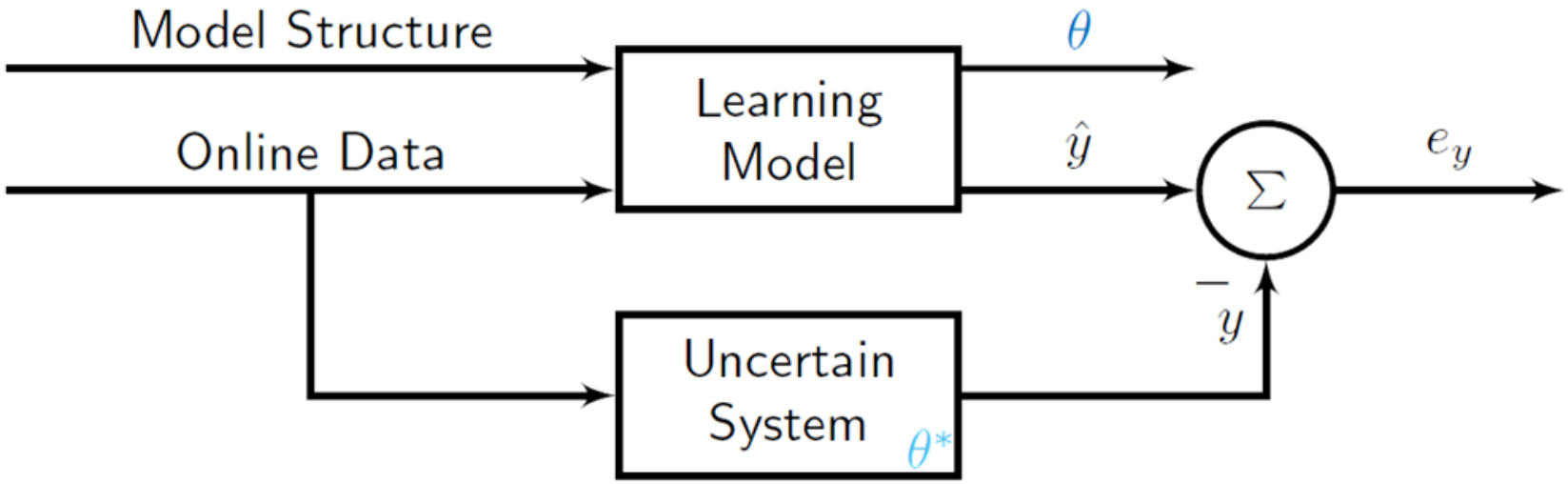
$$\dot{\theta} = C_1(\phi, \theta, e_y) \qquad \dot{x} = f_1(x, \theta^*, u)$$
$$u = C_2(\phi, \theta, e_y) \qquad y = f_2(x, \theta^*, u)$$



e_y, ϕ: Tracking Error, Real-Time Data

u, x, y: Input, State, Output

The Two Errors in Learning and Adaptation Based Systems



Parameter Estimation Error : $\tilde{\theta} = \theta - \theta^*$ ← Directly adjustable, not measurable

Output Prediction Error : $e_y = \hat{y} - y$ ← Measurable, not directly adjustable

Current Research: Fast Convergence of Both Errors

A Comparison of Different Learning Examples

Linear Regression

$$y = \phi^T \theta^*$$
$$\hat{y} = \phi^T \theta$$
$$e_y = \hat{y} - y$$
$$L = \frac{1}{2}\|\phi^T \theta - y\|_2^2$$
$$\dot{\theta} = -\gamma \nabla L(\theta) = -\gamma \phi e_y$$

Stability

$$V = \frac{1}{2}\|\theta - \theta^*\|_2^2$$

Neural Networks

$$y = f(\theta^*, \phi)$$
$$\hat{y} = f(\theta, \phi)$$
$$e_y = \hat{y} - y$$
$$L = \frac{1}{2}\|f(\theta, \phi) - y\|_2^2$$
$$\dot{\theta} = -\gamma \nabla L(\theta)$$

Stability

?

Parameter Estimation

$$\dot{x} = Ax + b(u + \phi^T \theta^*)$$
$$\dot{\hat{x}} = A\hat{x} + b(u + \phi^T \theta)$$
$$e = \hat{x} - x$$
$$\dot{\theta} = -\gamma \phi e^T P b$$

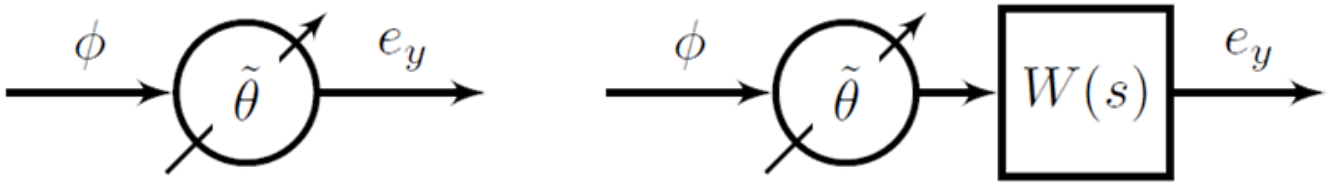
Stability

$$V = \frac{1}{2}\|\theta - \theta^*\|_2^2 + \frac{1}{2} e^T P e$$

Similar Parameter Update Laws

Adaptive Control: Error Models

Two classes of error models



Output error: $e_y = \hat{y} - y$, Parameter error: $\tilde{\theta} = \theta - \theta^*$ (θ^*: True parameter)

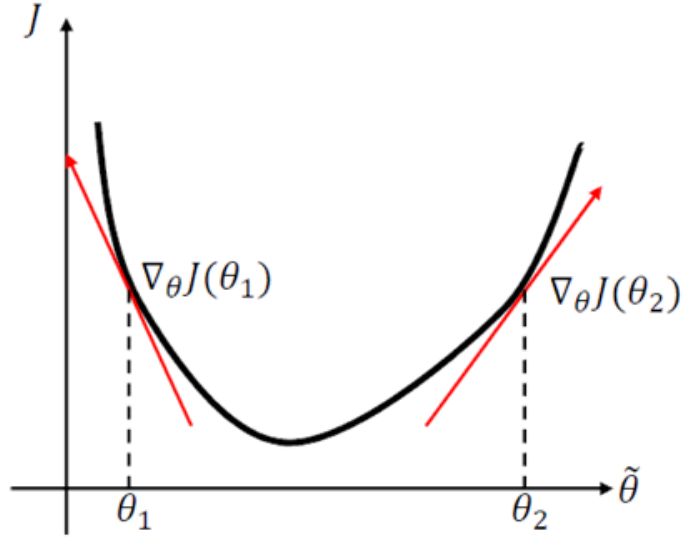
Adjust $\tilde{\theta}$ so that $e_y(t) \to 0$

Choose a cost function $J(\theta(t)) = (1/2)e_y^2(t)$:

$$\dot{\theta}(t) = -\gamma \nabla_\theta J(\theta(t)) = -\gamma \phi(t) e_y(t)$$

Same adaptive law for stricly positive real $W(s)$[1][2][3][4]

Many common update law modifications and concepts between fields[5]



[1] K. S. Narendra and A. M. Annaswamy (1989). *Stable Adaptive Systems*. (out of print). NJ: Prentice-Hall, Inc.

[2] S. Sastry and M. Bodson (1989). *Adaptive Control: Stability, Convergence and Robustness*. Prentice-Hall.

[3] K. J. Åström and B. Wittenmark (1995). *Adaptive Control: Second Edition*. Addison-Wesley Publishing Company.

[4] P. A. Ioannou and J. Sun (1996). *Robust Adaptive Control*. PTR Prentice-Hall.

[5] J. E. Gaudio, T. E. Gibson, A. M. Annaswamy, M. A. Bolender, and E. Lavretsky (2019). "Connections Between Adaptive Control and Optimization in Machine Learning". *58th IEEE Conference on Decision and Control (CDC), International Conference on Machine Learning (ICML), Workshop on Adaptive & Multitask Learning.*

Machine Learning: Error Models

Underlying model: $y_k = f(\phi_k, \theta^*)$, Predicted output: $\hat{y}_k = f(\phi_k, \theta_k)$

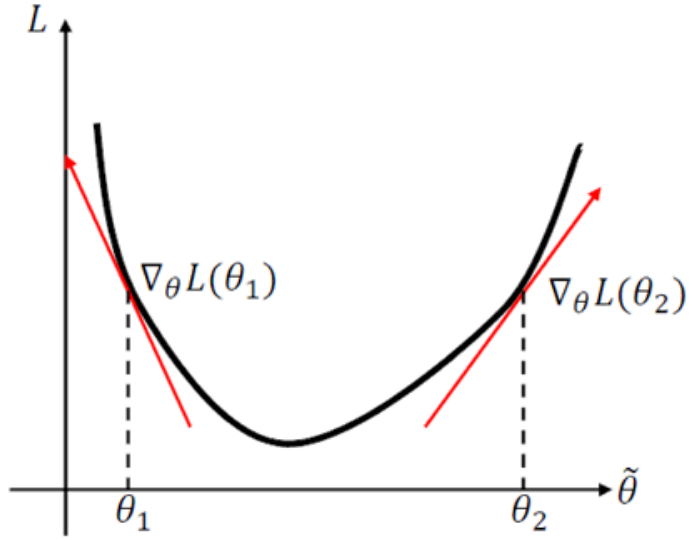
Determine parameter optimization rule:

Output error $e_{y,k} = \hat{y}_k - y_k$

Construct loss function $L(\theta_k)$

Parameter optimization rule[1][2][3][4][5]:

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



[1] R. O. Duda, P. E. Hart, and D. G. Stork (2001). *Pattern Classification, 2nd Edition*. John Wiley & Sons.

[2] C. M. Bishop (2006). *Pattern Recognition and Machine Learning*. Springer.

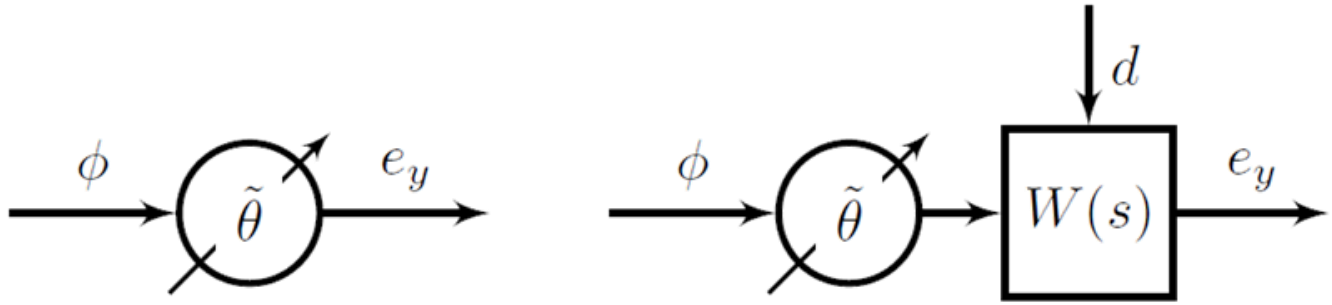
[3] T. Hastie, R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

[4] B. Efron and T. Hastie (2016). *Computer Age Statistical Inference: Algorithms, Evidence and Data Science*. Cambridge University Press.

[5] I. Goodfellow, Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press.

Intersection between Machine Learning & Adaptation:
Common Modifications to Parameter Update Laws

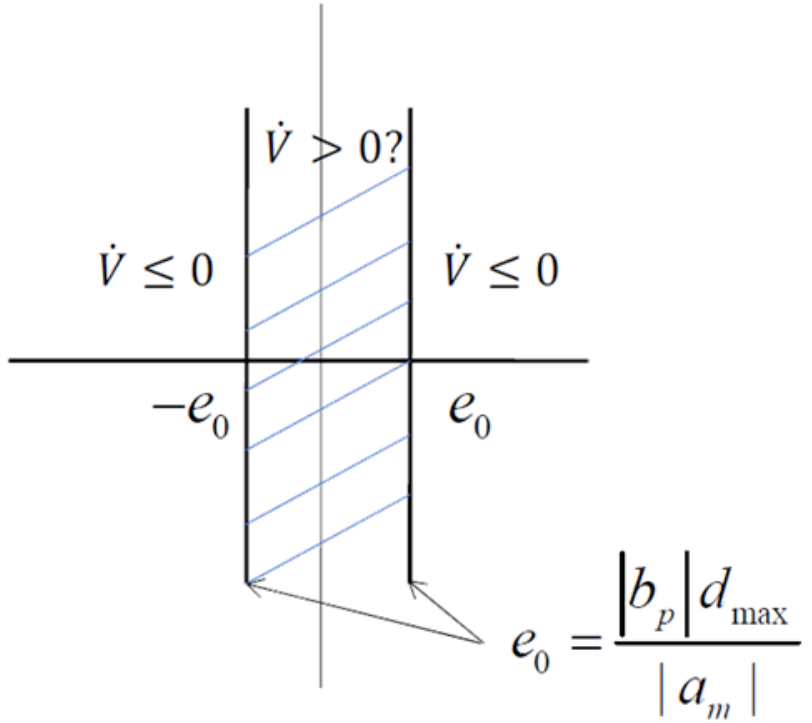
Robust Adaptive Control



Modeling error, time-variations, latencies are always present

Update law based on stability (Lyapunov function V)

$$V = e^2 + |b_p|\tilde{\theta}^2$$
$$\dot{V} = 2a_m e^2 + 2b_p e d$$
$$\leq -|e|(2|a_m||e| - 2|b_p|d_{max})$$



Modifications are needed

Adaptive Control: Modification to Account for Disturbances

Modify update law to incorporate the effect of d:

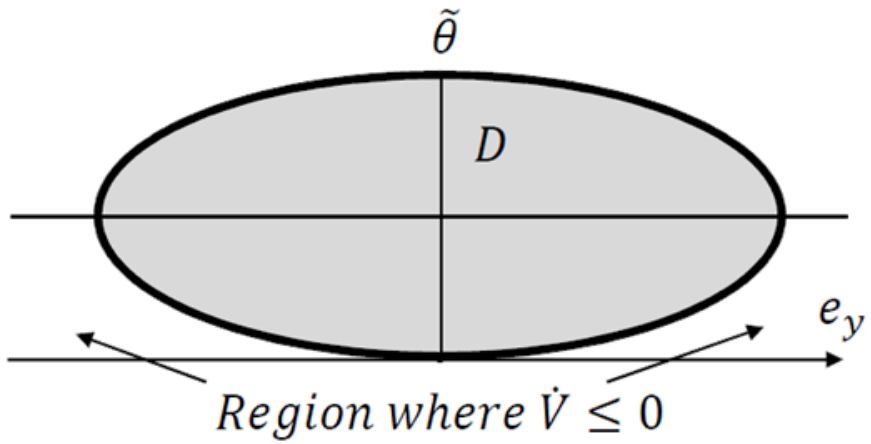
$$\dot{\theta}(t) = -\gamma \left[\nabla_\theta L(\theta(t)) + \sigma \mathcal{G}(\theta(t), e_y(t))\right]$$

$\sigma > 0$: scaling parameter

Different choices of $\mathcal{G}$ suggested in[1][2]:

- σ-modification: $\mathcal{G} = \theta$
- e-modification: $\mathcal{G} = \|e_y\|\theta$

Leads to $\dot{V} \leq 0$ outside of D, a compact set



[1] P. A. Ioannou and P. V. Kokotovic (1984). "Robust Redesign of Adaptive Control". *IEEE Transactions on Automatic Control* 29.3, pp. 202–211.

[2] K. S. Narendra and A. M. Annaswamy (1987a). "A New Adaptive Law for Robust Adaptation Without Persistent Excitation". *IEEE Transactions on Automatic Control* 32.2, pp. 134–145.

Machine Learning: Regularization

Change loss function to avoid overfitting to noise:

$$\bar{L}(\theta) = L(\theta) + \sigma\mathcal{R}(\theta)$$

$\sigma > 0$: Lagrange multiplier

Leads to "regularized follow the leader" algorithm[1]:

$$\theta_{k+1} = \theta_k - \gamma_k \left[\nabla_\theta L(\theta_k) + \sigma\nabla_\theta \mathcal{R}(\theta_k)\right]$$

Choice of $\mathcal{R}$:

- ℓ_2 regularization (ridge regression): $\mathcal{R} = (1/2)\|\theta\|_2^2$ (coincides with the σ-modification[2])
- ℓ_1 regularization (lasso): with $\mathcal{R} = \|\theta\|_1$ (induces sparsity[3])

---

[1] E. Hazan (2016). "Introduction to Online Convex Optimization". *Foundations and Trends in Optimization* 2.3-4, pp. 157–325.

[2] P. A. Ioannou and P. V. Kokotovic (1984). "Robust Redesign of Adaptive Control". *IEEE Transactions on Automatic Control* 29.3, pp. 202–211.

[3] I. Goodfellow, Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press.

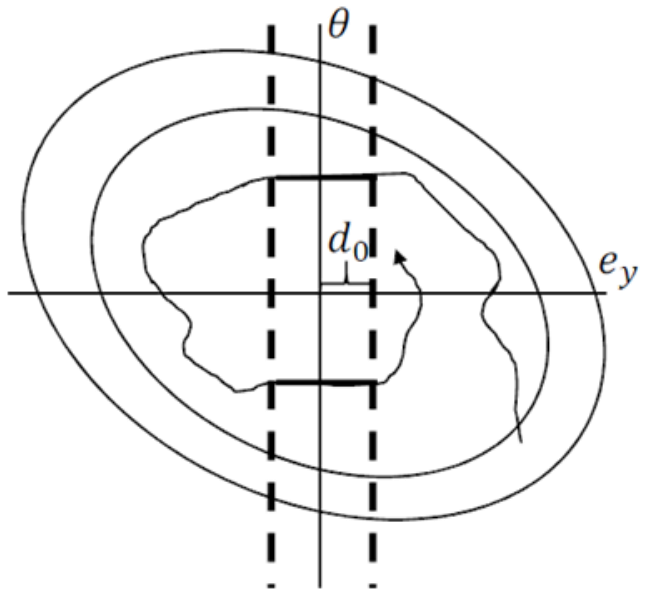
Adaptive Control: Deadzone Modification

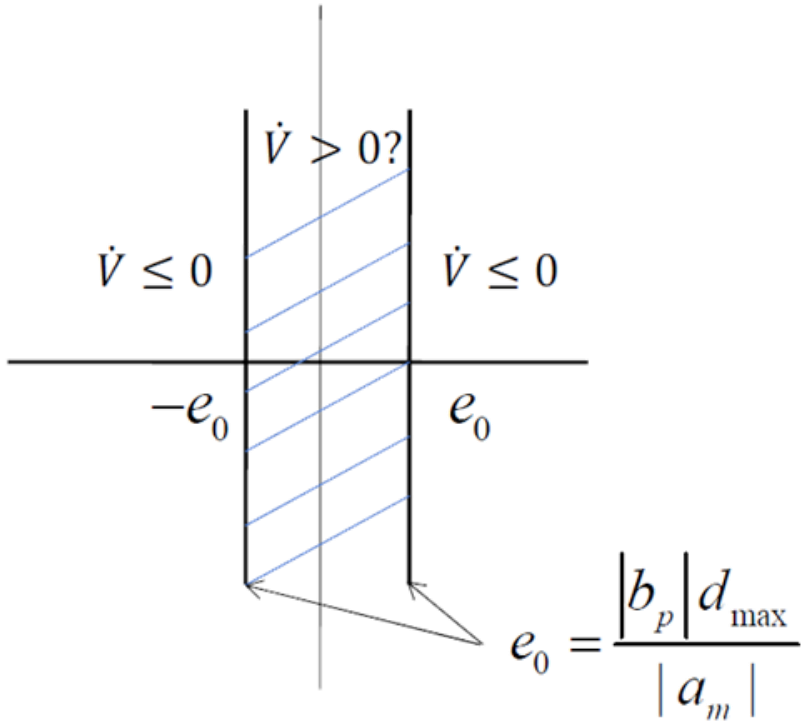
Introduce "dead zone"[1]:

$$\dot{\theta}(t) = \begin{cases} -\gamma\phi e_y, & \mathcal{D}(e_y) > d_0 + \epsilon \\ 0, & \mathcal{D}(e_y) \leq d_0 + \epsilon \end{cases}$$

Common choice: $\mathcal{D} = \|e_y\|$

$\dot{V} \leq$ in D^c





[1] B. B. Peterson and K. S. Narendra (1982). "Bounded Error Adaptive Control". *IEEE Transactions on Automatic Control* 27.6, pp. 1161–1168.

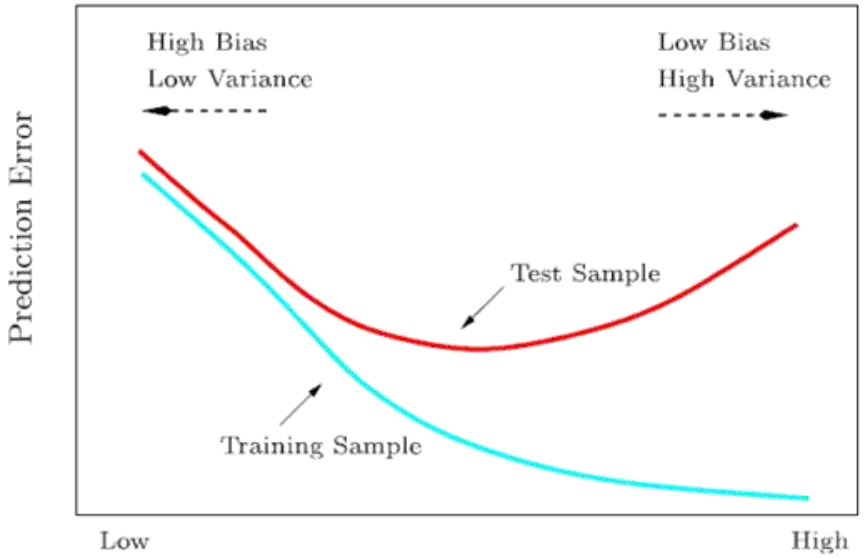
Machine Learning: Early Stopping

Training often stopped early to counter overfitting

Use multiple data sets and stop the parameter update process when loss begins to increase[1]

Needed for training neural networks due to their large number of parameters[2]

Can act as regularization[3]



[2]

[1] L. Prechelt (1998). "Automatic early stopping using cross validation: quantifying the criteria". *Neural Networks* 11.4, pp. 761–767.

[2] T. Hastie, R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer; C. M. Bishop (2006). *Pattern Recognition and Machine Learning*. Springer; B. Efron and T. Hastie (2016). *Computer Age Statistical Inference: Algorithms, Evidence and Data Science*. Cambridge University Press; I. Goodfellow, Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press.

[3] J. Sjöberg and L. Ljung (1995). "Overtraining, regularization and searching for a minimum, with application to neural networks". *International Journal of Control* 62.6, pp. 1391–1407.

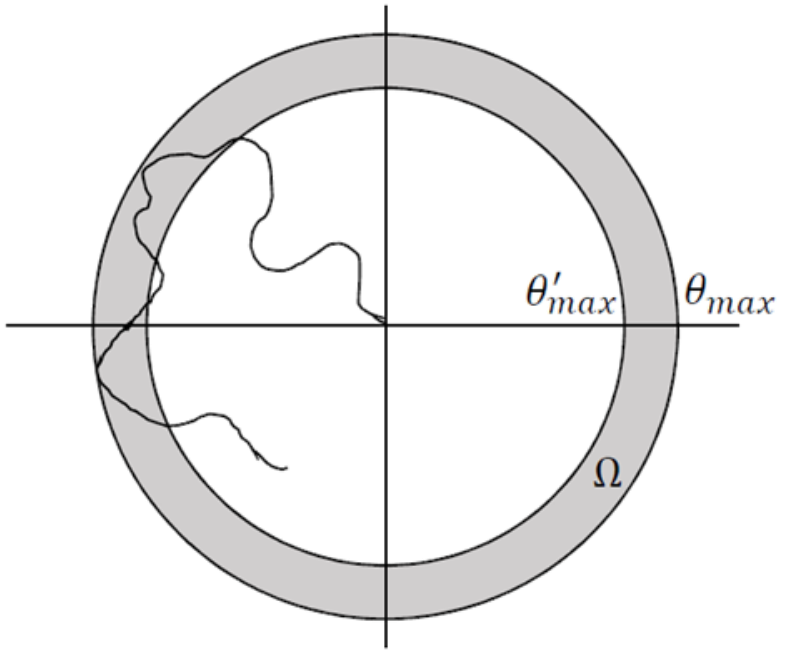
Adaptive Control: Projection

Compact region for the parameters θ

During the learning process the parameters are not allowed to leave the region

Robustness to disturbances and unmodeled dynamics[1][2][3]

$$\text{Proj}(\theta_i, \zeta_i) = \begin{cases} \frac{\theta_{i,\max}^2 - \theta_i^2}{\theta_{i,\max}^2 - \theta_{i,\max}'^2}\zeta_i, & \theta_i \in \Omega_i \wedge \theta_i\zeta_i > 0 \\ \zeta_i, & \text{otherwise} \end{cases}$$



Modify update law:

$$\dot{\theta}(t) = -\gamma \text{Proj}\left[\theta(t), \nabla_\theta L(\theta(t))\right].$$

[1] G. Kreisselmeier and K. S. Narendra (1982). "Stable Model Reference Adaptive Control in the Presence of Bounded Disturbances". *IEEE Transactions on Automatic Control* 27.6, pp. 1169–1175.

[2] E. Lavretsky, T. E. Gibson, and A. M. Annaswamy (2012). "Projection Operator in Adaptive Systems". *arXiv preprint arXiv:1112.4232*.

[3] H. S. Hussain (2017). "Robust Adaptive Control in the Presence of Unmodeled Dynamics". PhD thesis. MIT.

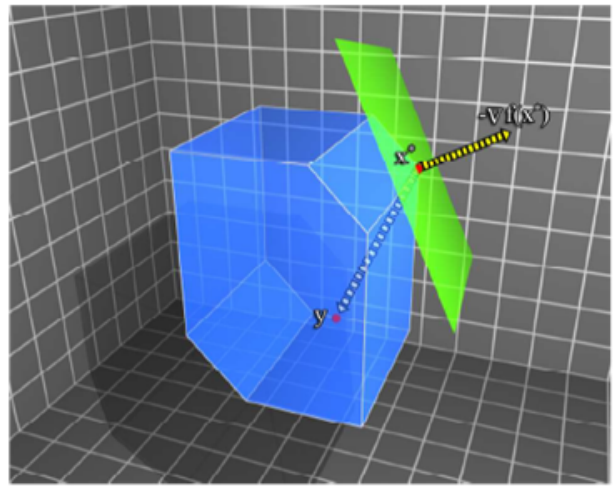
Machine Learning: Projection

Projection operation which finds the point in a convex set which is closest to a specified point:

$$\Pi_{\Theta}(\bar{\theta}) \triangleq \underset{\theta \in \Theta}{\arg\min} \|\theta - \bar{\theta}\|$$

Employed in the update sequence[1][2][3][4][5]

$$\bar{\theta}_{k+1} = \theta_k - \gamma_k \nabla_\theta L(\theta_k), \qquad \theta_{k+1} = \Pi_{\Theta}(\bar{\theta}_{k+1})$$



[4]

[1] M. Zinkevich (2003). "Online Convex Programming and Generalized Infinitesimal Gradient Ascent". In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936.

[2] E. Hazan, A. Agarwal, and S. Kale (2007). "Logarithmic regret algorithms for online convex optimization". *Machine Learning* 69.2-3, pp. 169–192.

[3] E. Hazan, A. Rakhlin, and P. L. Bartlett (2008). "Adaptive Online Gradient Descent". In: *Advances in Neural Information Processing Systems 20*. Curran Associates, Inc., pp. 65–72.

[4] E. Hazan (2016). "Introduction to Online Convex Optimization". *Foundations and Trends in Optimization* 2.3-4, pp. 157–325.

[5] S. Bubeck (2015). "Convex Optimization: Algorithms and Complexity". *Foundations and Trends in Machine Learning* 8.3-4, pp. 231–357.

Adaptive Control: Adaptive Gains and Stepsizes

$e_y(t,\tau) = \tilde{\theta}^T(t)\phi(\tau),\ \tau \leq t$ adjust $\tilde{\theta}$ so $\int_{t_0}^{t} e_y^2(t,\tau)d\tau$ is minimized:

$$\dot{\tilde{\theta}} = -\gamma \int_{t_0}^{t} e_y(t,\tau)\phi(\tau)d\tau$$

Uses all available data, need to use forgetting factor: weight with $exp\{-\lambda_\Gamma(t-\tau)\}$

Leads to a RLS type of update law[1][2][3]:

$$\dot{\theta}(t) = -\Gamma(t)\phi(t)e_y(t)$$

$$\dot{\Gamma}(t) = \begin{cases} \lambda_\Gamma\Gamma(t) - \frac{\Gamma(t)\phi(t)\phi^T(t)\Gamma(t)}{\mathcal{N}(t)}, & \|\Gamma(t)\| \leq \Gamma_{\max} \\ 0, & \text{otherwise} \end{cases}$$

$\lambda_\Gamma \geq 0$ is a *forgetting factor* and $\mathcal{N}(t) = (1 + \mu\phi^T(t)\phi(t))$ is a *normalizing signal*

---

[1] G. Kreisselmeier (1977). "Adaptive Observers with Exponential Rate of Convergence". *IEEE Transactions on Automatic Control* 22.1, pp. 2–8.

[2] K. S. Narendra and A. M. Annaswamy (1989). *Stable Adaptive Systems*. (out of print). NJ: Prentice-Hall, Inc.

[3] P. A. Ioannou and J. Sun (1996). *Robust Adaptive Control*. PTR Prentice-Hall.

Machine Learning: Adaptive Gains and Stepsizes

Adaptive step size methods[1][2][3][4] have seen widespread use in machine learning

Helps handle sparse and small gradients by adjusting the step size as a function of gradients

A common update law:

$$\bar{\theta}_{k+1} = \theta_k - \gamma_k m_k / V_k^{1/2}, \qquad \theta_{k+1} = \Pi_\Theta(\bar{\theta}_{k+1})$$

Normalization by gradient as compared to normalization by regressor

	$m_k(g_1, \ldots, g_k)$	$V_k(g_1, \ldots, g_k)$
Projected gradient descent	g_k	I
AdaGrad	g_k	$\epsilon I + \mathsf{diag}(\sum_{i=1}^k g_i \odot g_i)$
Adam	$(1-\beta_1)\sum_{i=1}^k \beta_1^{k-i} g_i$	$(1-\beta_2)\mathsf{diag}(\sum_{i=1}^k \beta_2^{k-i} g_i \odot g_i)$

[1] J. Duchi, E. Hazan, and Y. Singer (2011). "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". *Journal of Machine Learning Research* 12, pp. 2121–2159.

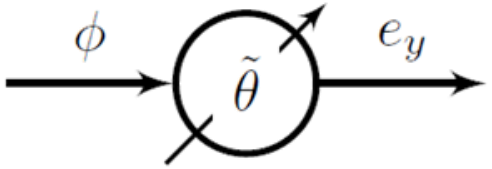
[2] M. D. Zeiler (2012). "ADADELTA: an adaptive learning rate method". *arXiv preprint arXiv:1212.5701*.

[3] D. P. Kingma and J. L. Ba (2017). "Adam: A Method for Stochastic Optimization". *arXiv preprint arXiv:1412.6980*.

[4] S. J. Reddi, S. Kale, and S. Kumar (2018). "On the Convergence of Adam and Beyond". In: *International Conference on Learning Representations*.

Intersection between Machine Learning & Adaptation: Common Concepts and Tools

Adaptive Control: Persistent Excitation



Persistent Excitation of ϕ: $R(\phi) = \int_t^{t+T} \phi(\tau)\phi^T(\tau)d\tau \geq \alpha I$

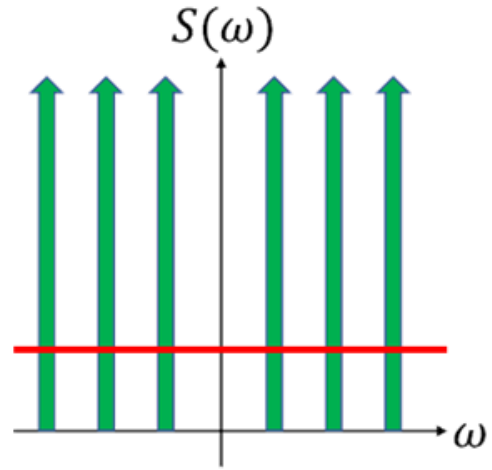
ϕ can be periodic[1], e.g. $\phi(t) = \sum_i a_i \sin \omega_i t$

The PE condition corresponds to certain spectral conditions being satisfied by the regressor[2]

Deterministic and stochastic cases have been considered[3]

PE condition $\Rightarrow$ perfect learning

$\triangleright$ Exponential convergence rates



[1] A. P. Morgan and K. S. Narendra (1977). "On the Uniform Asymptotic Stability of Certain Linear Nonautonomous Differential Equations". *SIAM Journal on Control and Optimization* 15.1, pp. 5–24.

[2] S. Boyd and S. Sastry (1983). "On parameter convergence in adaptive control". *Systems & Control Letters* 3.6, pp. 311–319; S. Boyd and S. S. Sastry (1986). "Necessary and Sufficient Conditions for Parameter Convergence in Adaptive Control". *Automatica* 22.6, pp. 629–639.

[3] G. C. Goodwin and K. S. Sin (1984). *Adaptive Filtering Prediction and Control*. Prentice Hall; B. D. Anderson and C. Johnson (1982). "Exponential Convergence of Adaptive Identification and Control Algorithms". *Automatica* 18.1, pp. 1–13; K. S. Narendra and A. M. Annaswamy (1986). "Robust Adaptive Control in the Presence of Bounded Disturbances". *IEEE Transactions on Automatic Control* 31.4, pp. 306–315; K. S. Narendra and A. M. Annaswamy (1987b). "Persistent excitation in adaptive systems". *International Journal of Control* 45.1, pp. 127–160; L. Ljung (1987). *System Identification: Theory for the User*. Prentice-Hall.

Machine Learning: Persistent Excitation

Often consider white noise/Gaussian $\mathcal{N}(0, \sigma^2)$ inputs for regressor ϕ[1]

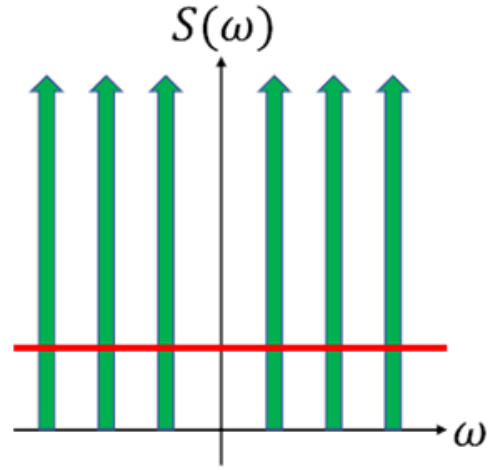
In the limit of infinite samples: $S(\omega) = c > 0$, a constant, sufficient for $R(\phi)$ positive definite

Realistic case of finite samples:

With probability $1 - p_f$, and $T \geq T_{min}(p_f)$: $\quad \|\text{parameter error}\| \leq \tilde{\mathcal{O}}\left(poly\left(\frac{1}{T}\right), polylog\left(\frac{1}{p_f}\right)\right)$

Probability of failure p_f allows for error due to the presence of finite samples

▷ Polynomial convergence rates



[1] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu (2018). "Regret Bounds for Robust Adaptive Control of the Linear Quadratic Regulator". In: *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., pp. 4192–4201.

Adaptive Control: Stability Framework

Update laws chosen by constructing Lyapunov functions V[1]:

$$V = \gamma^{-1}\tilde{\theta}^T\tilde{\theta} + e^T P e + \alpha\tilde{\phi}^T\bar{P}\tilde{\phi}$$

$\dot{\tilde{\phi}} = F\tilde{\phi}$ F: Hurwitz matrix

Leads to:

$$\dot{V} = -e^T Q e - \alpha\tilde{\phi}^T\bar{Q}\tilde{\phi} + 2e^T P b\theta^{*T}\tilde{\phi} \leq 0 \qquad \text{for } \alpha > \alpha_0$$

Define $\delta(t) = 2e^T P b\theta^{*T}\tilde{\phi}$

We also obtain:

$$\int_{t_0}^{T} e^T Q e dt - \int_{t_0}^{T}\delta(t)dt \leq -\int_{t_0}^{T}\dot{V}dt = V(t_0) - V(T) < \infty$$

Therefore $e \in \mathcal{L}_2 \cap \mathcal{L}_\infty$

[1] K. S. Narendra and A. M. Annaswamy (1989). *Stable Adaptive Systems*. (out of print). NJ: Prentice-Hall, Inc.

Machine Learning: Regret

In online learning, efficiency of an algorithm is often analyzed using the notion of "regret"

Choose convex cost $\mathcal{C}_k = e_k^T Q e_k$, $Q = Q^T > 0$ Best cost: $\min_{\theta \in \Theta} \sum_{k=1}^{T} \mathcal{C}_k(\theta)$

Represents cost associated with the best parameter estimate in hindsight[1][2][3][4]

Define regret as follows:

$$\text{regret}_T = \sum_{k=1}^{T} \mathcal{C}_k(\theta_k) - \min_{\theta \in \Theta} \sum_{k=1}^{T} \mathcal{C}_k(\theta)$$

This is equivalent to:

$$\text{continuous regret}_T = \int_{t_0}^{T} e^T Q e dt - \int_{t_0}^{T} \bar{\delta}(t) dt$$

$\bar{\delta}(t)$: Exponentially decaying signal

---

[1] M. Zinkevich (2003). "Online Convex Programming and Generalized Infinitesimal Gradient Ascent". In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936.

[2] E. Hazan, A. Agarwal, and S. Kale (2007). "Logarithmic regret algorithms for online convex optimization". *Machine Learning* 69.2-3, pp. 169–192.

[3] E. Hazan, A. Rakhlin, and P. L. Bartlett (2008). "Adaptive Online Gradient Descent". In: *Advances in Neural Information Processing Systems 20*. Curran Associates, Inc., pp. 65–72.

[4] E. Hazan (2016). "Introduction to Online Convex Optimization". *Foundations and Trends in Optimization* 2.3-4, pp. 157–325.

Lyapunov Functions and Regret

$$\text{Machine Learning : regret}_T = \sum_{k=1}^{T} \mathcal{C}_k(\theta_k) - \min_{\theta \in \Theta} \sum_{k=1}^{T} \mathcal{C}_k(\theta)$$

$$\text{Quadratic Cost, Continuous : regret}_T = \int_{t_0}^{T} e^T Q e dt - \int_{t_0}^{T} \bar{\delta}(t) dt$$

$$\text{Adaptive Control : } \int_{t_0}^{T} e^T Q e dt - \int_{t_0}^{T} \delta(t) dt \leq - \int_{t_0}^{T} \dot{V} dt = V(t_0) - V(T)$$

Machine learning: Show that average regret grows sublinearly[1],i.e. $(1/T)\text{regret}_T \to 0$ with T

Adaptive control: Convergence of $e(t)$ to zero established by updating V to include effect of initial conditions. $\text{regret}_T = \mathcal{O}(1)$, a constant

[1] E. Hazan (2016). "Introduction to Online Convex Optimization". *Foundations and Trends in Optimization* 2.3-4, pp. 157–325.

Improved Algorithm Performance Bounds

Loss Function : $L = \frac{1}{2}\|\phi^T\theta - y\|_2^2$, Prediction Error : $e_y = \hat{y} - y$, Parameter Error : $\tilde{\theta} = \theta - \theta^*$

$$\text{regret}_T = \int_{t_0}^{T} e_y^2 dt = \int_{t_0}^{T} (\phi^T\tilde{\theta})^2 dt$$

Adaptive Control

Algorithm : $\dot{\theta} = -\gamma\nabla_\theta L(\theta) = \gamma\phi e_y$

Lyapunov Function : $V = \gamma^{-1}\tilde{\theta}^T\tilde{\theta}$

Time Derivative : $\dot{V} = -e_y^2$

$$\int_{t_0}^{T} e_y^2 dt = -\int_{t_0}^{T} \dot{V} dt = V(t_0) - V(T) < \infty$$

Machine Learning

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

$\forall\theta_1, \theta_2 \in \Theta,\ \|\theta_1 - \theta_2\| \leq D$

$|f(\theta_1) - f(\theta_2)| \leq G\|\theta_1 - \theta_2\|$

Step Size : $\gamma_k = \frac{D}{G\sqrt{k}}$

$\text{regret}_T \leq V(t_0)$ | $\mathcal{O}(1)$ vs. $\mathcal{O}(\sqrt{T})$ | $\text{regret}_T \leq \frac{3}{2}GD\sqrt{T}$

Deep Reinforment Learning Methods for Air-to-Air Combat & Control

- *Anubhav Guha, MIT PhD Candidate, prior AFS Autonomy Research Division* ***(Presenter)***
- *Dr. James Paduano, AFS Autonomy Tech Area Lead*
- *John Piotti, AFS Autonomy Research Division*
- *Bradley Lan, Morse Corp. GNC, prior AFS Autonomy Research Division*

Deep Reinforcement Learning Methods for Air-to-Air Combat & Control

This material, namely slides 6, 9, and 10, is based on research sponsored by the Air Force Research Laboratory under Subaward number FA8650-19-2-6983. Other than media embedded herein from Johns Hopkins Applied Physics Laboratory (APL), the views and conclusions contained in this presentation are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

Aurora FLIGHT SCIENCES
A Boeing Company

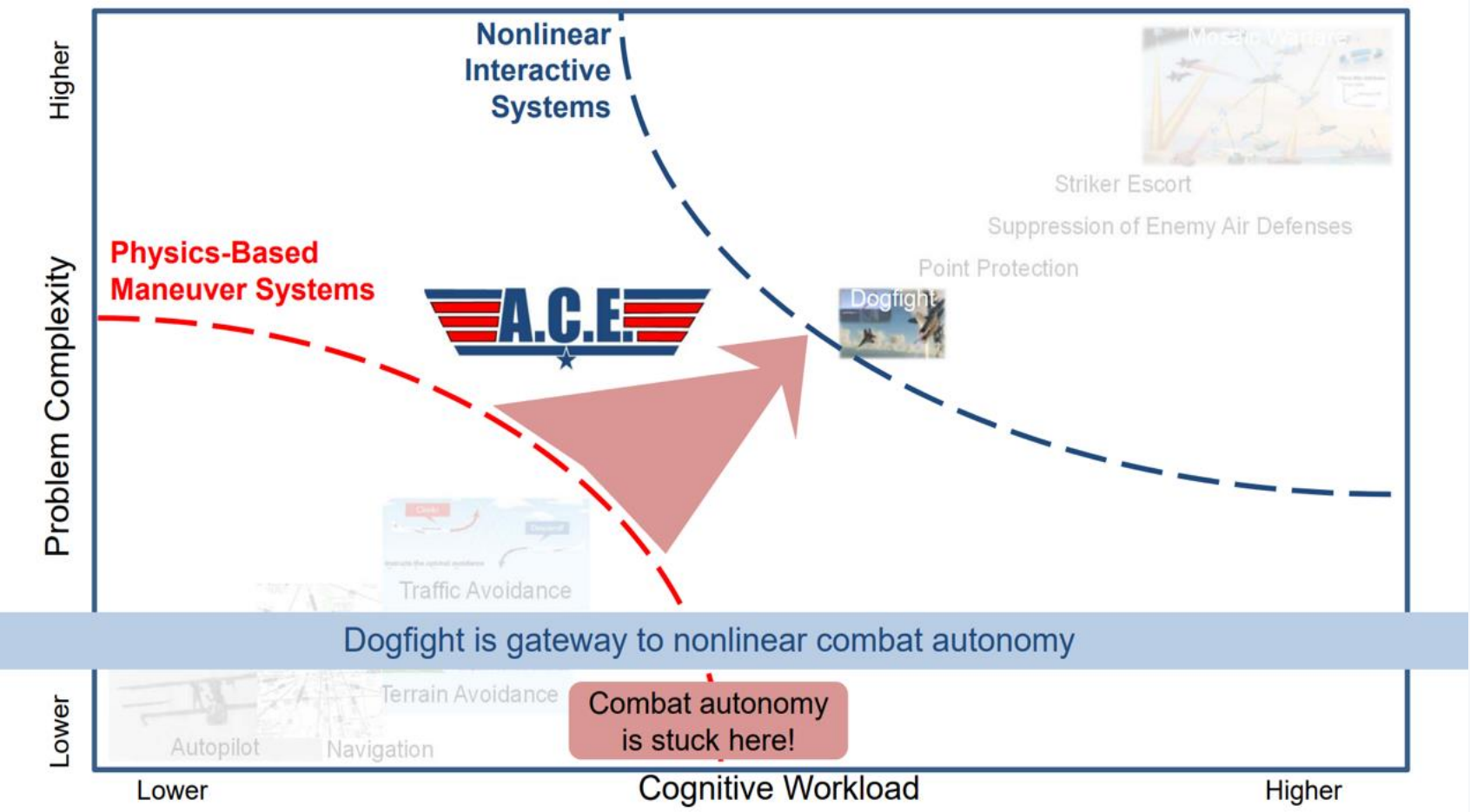
Overview

- Timeline of Artificial Intelligence in Games
- DARPA Alpha Dogfight Trials
- Key Components of Reinforcement Learning
- Hierarchical Reinforcement Learning and Curriculum Training
- Final Results from AlphaDogfight Trials
- Key Takeaways
- Questions

Timeline of Artificial Intelligence in Games

Year	Summary	Description
1952	Machines Playing Checkers	Arthur Sameul joins IBM Poughkeepsie Laboratory and begins working on some of the very first machine learning programs, first creating programs that play checkers
1963	Machines Playing Tic-Tac-Toe	Donald Michie creates a ‘machine’ consisting of 304 match boxes and beads, which uses reinforcement learning to play Tic-Tac-Toe
1992	Machines Playing Backgammon	Gerald Tesauro develops TD-Gammon, a computer backgammon program that uses an artificial neural network trained using temporal difference learning. TD-Gammon can rival, but not consistently surpass, the abilities of top human backgammon players
1997	IBM Deep Blue Beats Kasparov	IBM’s Deep Blue beats the world champion at chess
2011	Beating Humans in Jeopardy	Using a combination of machine learning, natural language processing and information retrieval techniques, IBM’s Watson beats two human champions in a Jeopardy! competition
2016	Beating Humans in Go	Google’s AlphaGo program becomes the first Computer Go program to beat an unhandicapped professional human player using a combination of machine learning and tree search techniques. Later improved as AlphaGo Zero and then in 2017 generalized to Chess and more two player games with Alpha Zero
2019	Beating Humans in Dota 2	In April 2019, OpenAI won a Dota 2 best-of-three series against The international 2018 champions OG at a live event in San Francisco. A four-day online event to play against the bots, open to the public, occurred the same month where OpenAI Five won all but 4075 out of 42,729 games.
2020+	Real-world combat games?	

Program Overview



Source: DARPA Air Combat Evolution

- The ability to successfully apply full-scale AI to combat problems is the first step in applying autonomy to highly complex systems of value
- Fully autonomous agents enable novel missions and task directives

DARPA Air Combat Evolution (ACE) – ADT follow-on

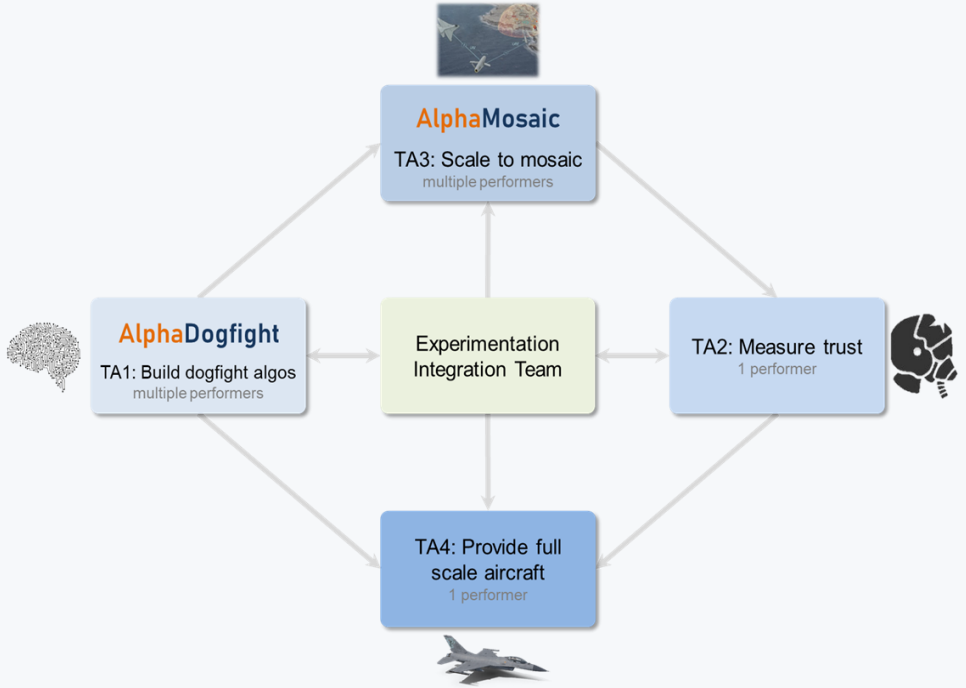
Motivation: Human pilots & battle managers will need to rely on behaviors of autonomous systems to increase ratio of humans to unmanned assets

TA1 **Increase performance** of automated tactical decision making

TA2 Build pilot **trust** in combat automation

TA3 **Scale** performance and maintain trust up the stack

TA4 Demonstrate performance on increasingly **realistic** platforms



Source: DARPA Air Combat Evolution

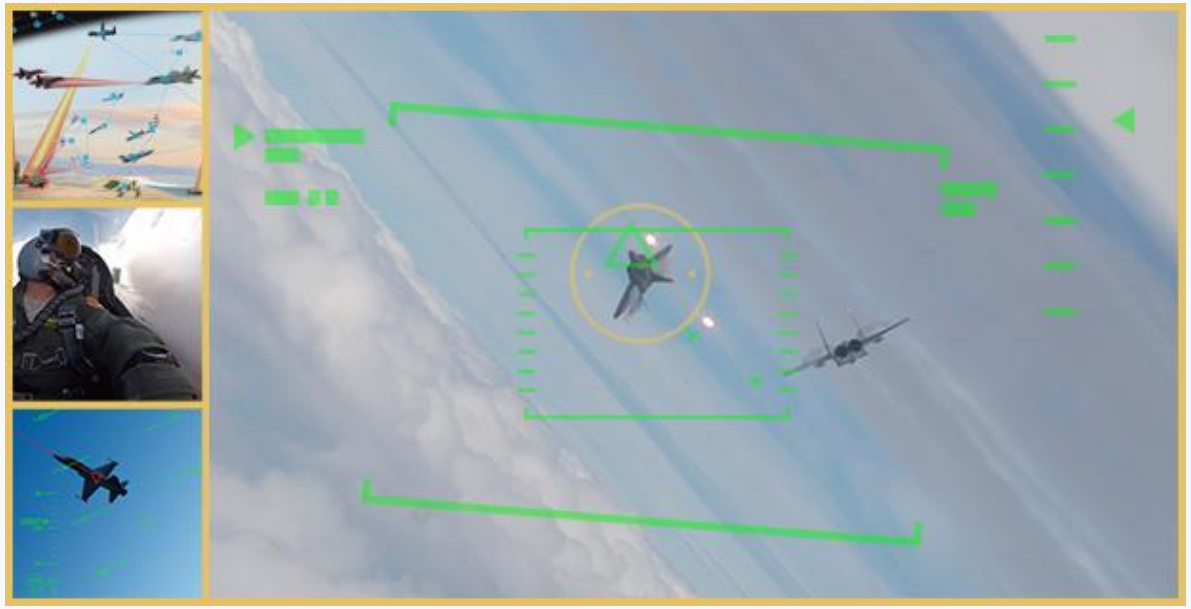
Alpha Dogfight Trials

- A guns only, 1v1 dogfight
- Modern fighter jet dynamics

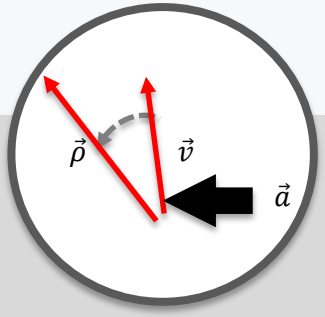
- Explainability and trust are important factors!
- Our approach – a structured and disciplined application of modern reinforcement learning and training techniques

Problem Specifics

- 1v1 Dogfight (adversarial)
- No Noise
- Mixed observability
- $\boldsymbol{INPUTS}$: $[p, Nz, \beta, Fx]$
- $\boldsymbol{OUTPUTS}$: $[u, v, w, \phi, \theta, \psi \ \ldots] \epsilon \ \mathbb{R}^{\sim 50}$
- $\boldsymbol{OBJECTIVE}$: $Enemy\ Health = 0$
- Well posed as a reinforcement learning/sequential decision making task

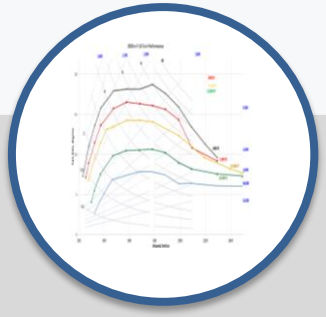


Aurora's AlphaDogfight Approach



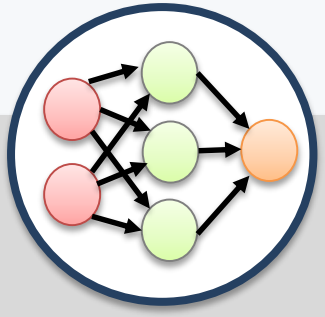
Pro-Nav Targeting

- Leverage strong controller design
- Emphasis on perfecting basic techniques



Expert System

- Heavily utilize pilot-designed objective functions and state machines
- Rule-based decision making
- Utilize EM theory and analysis



End-to-End AI

- Reinforcement Learning trained agent
- Bootstrapped training via imitation learning
- Trained vs canned agents and prior developed agents (pro-nav and expert systems)

Self-Play AI

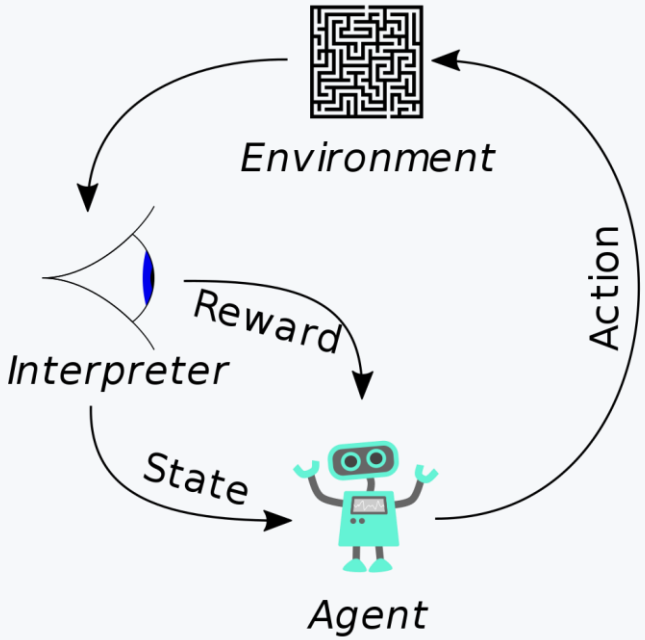
- Built off End-to-End AI
- Trained via population based self-play
- Able to generate novel strategies and tactics

Hierarchical RL

- Hierarchical Reinforcement Learning Structure
- Leverages the power of RL/AI
- Maintains understandability and modularity through explicit structure and design

Key Concepts in Reinforcement Learning

- Reward shaping
- State space design
- Different types of reinforcement learning for different use cases:



Algorithm	Description	Action Space	State Space
Monte Carlo	Every visit to Monte Carlo	Discrete	Discrete
Q-learning	State–action–reward–state	Discrete	Discrete
SARSA	State–action–reward–state–action	Discrete	Discrete
Q-learning - Lambda	State–action–reward–state with eligibility traces	Discrete	Discrete
SARSA - Lambda	State–action–reward–state–action with eligibility traces	Discrete	Discrete
DQN	Deep Q Network	Discrete	Continuous
DDPG	Deep Deterministic Policy Gradient	Continuous	Continuous
A3C	Asynchronous Advantage Actor-Critic Algorithm	Continuous	Continuous
NAF	Q-Learning with Normalized Advantage Functions	Continuous	Continuous
TRPO	Trust Region Policy Optimization	Continuous	Continuous
PPO	Proximal Policy Optimization	Continuous	Continuous
TD3	Twin Delayed Deep Deterministic Policy Gradient	Continuous	Continuous
SAC	Soft Actor-Critic	Continuous	Continuous

RL Example: Q-Learning

- Q value and the Q-learning update:

New Q value

Temporal Difference

$$Q_{i+1}(s_t, a_t) \leftarrow Q_i(s_t, a_t) + \alpha \cdot [r(s,a) + \gamma \cdot \max_a Q_i(s_t + 1, \alpha) - Q(s_t, a_t)]$$

Old Q value

- Bellman equation & optimal value function:

$$V(s) = \max_{a \in A}\{Q^*(s,a)\}$$
$$V(s) = \max_{a \in A}\{r(s,a) + \gamma V\big(f(s,a)\big)\}$$

- Optimality of Q-learning:

$$\lim_{i \to \infty} \max_{a \in A}\{Q_i(s,a)\} = V(s)$$

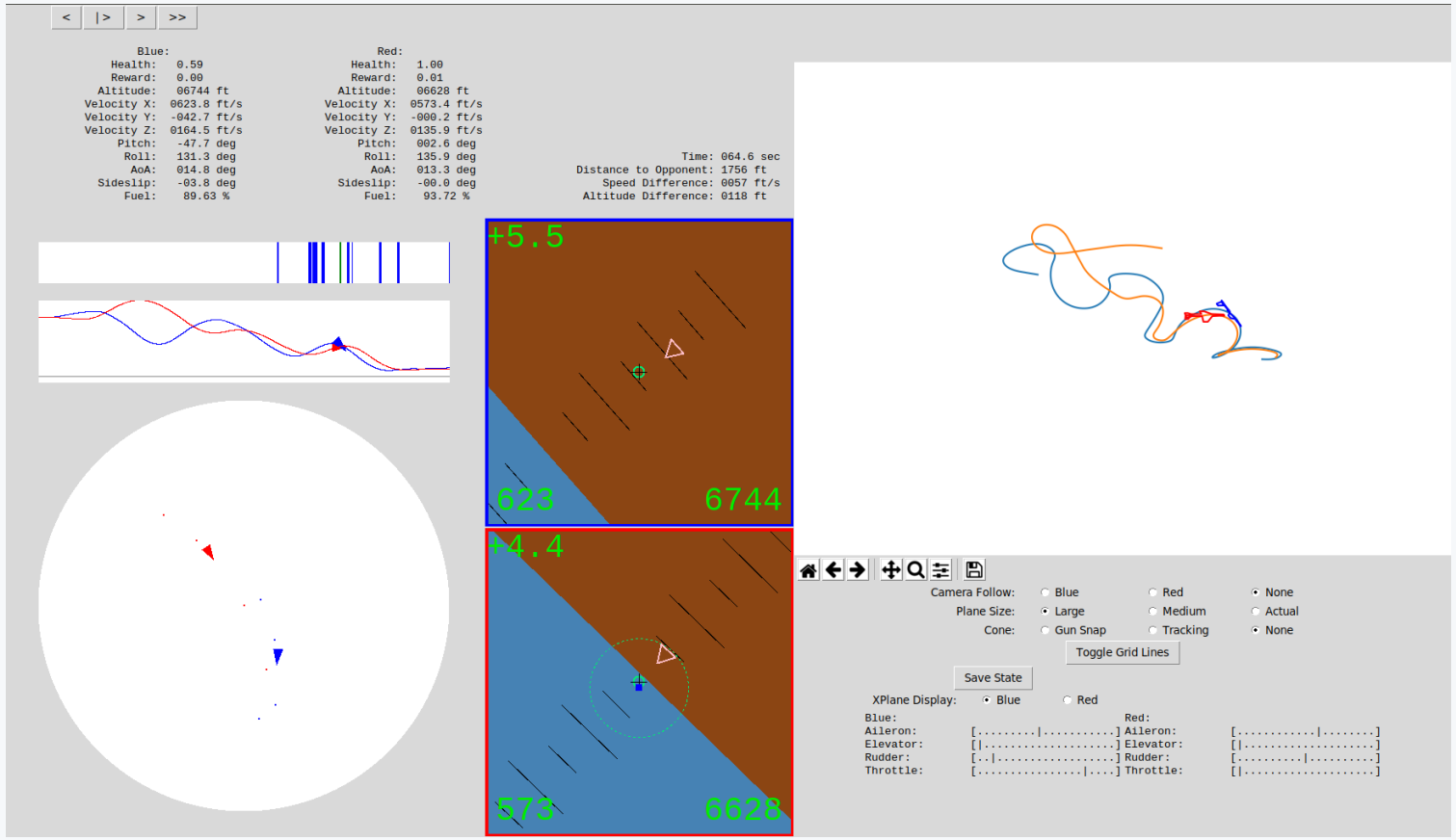
Reinforcement Learning Difficulties

- Credit Assignment Problem
  - "Reward Engineering"

$$\pi^* = argmax_\pi \, E\left[\sum_{t=0}^{\infty} \gamma^t r_t | \pi\right]$$

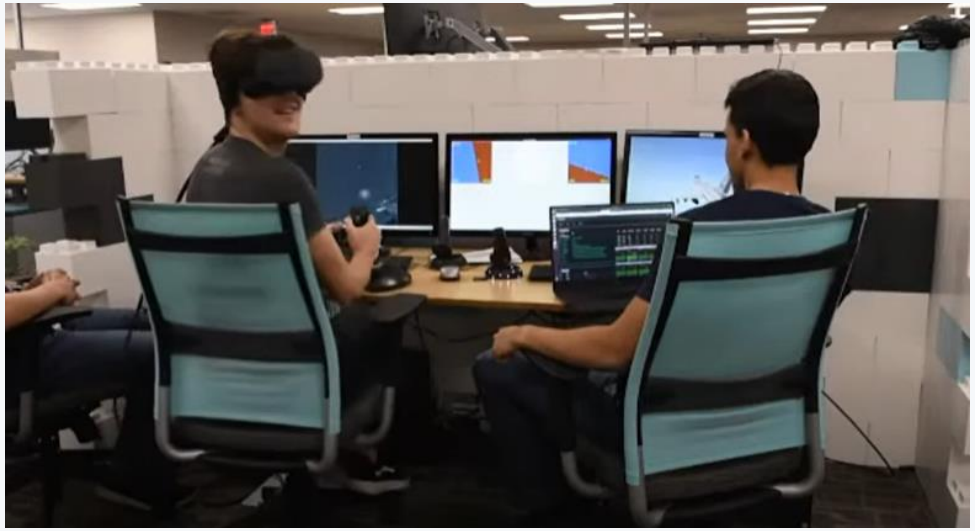
- Exploration vs. Exploitation
  - Heuristic solutions (process noise, epsilon-greedy, UCB)
- Curse of Dimensionality
- Transfer learning, non-stationary environments, etc.

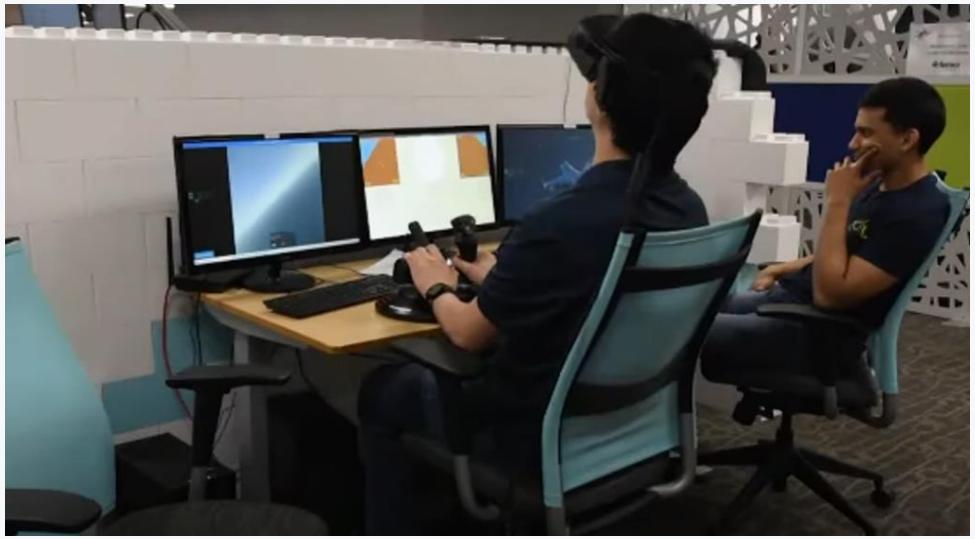
Tools to Drive Training



- Visualization tools help experts debug and accelerate training

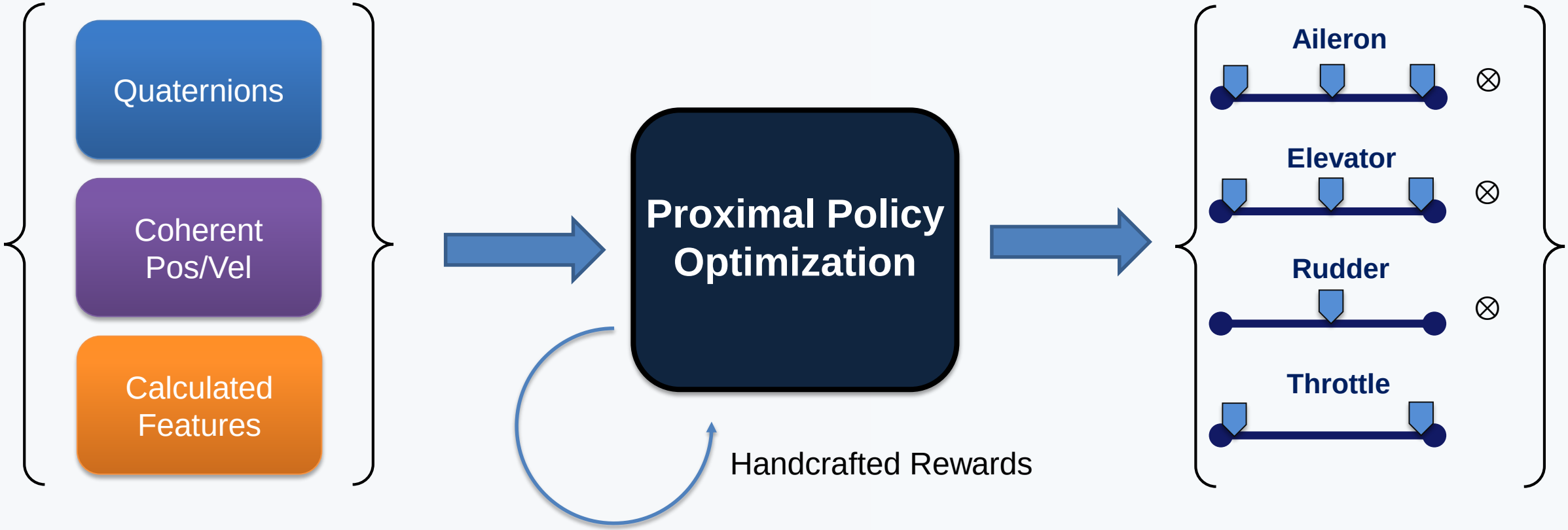
Tools to Drive Training





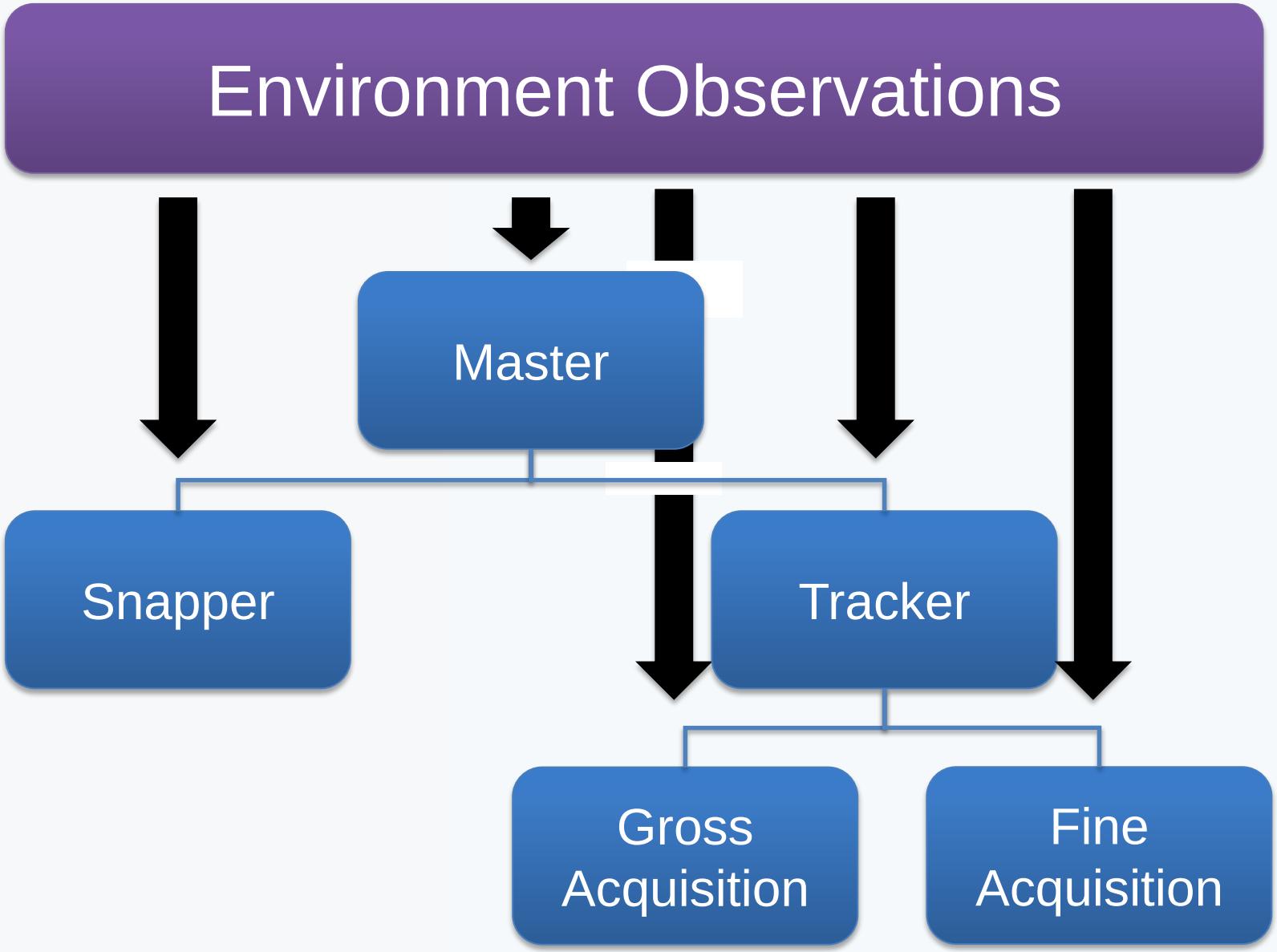
- AI verification and validation techniques through VR
- Know your system

Base Single-Level Agent



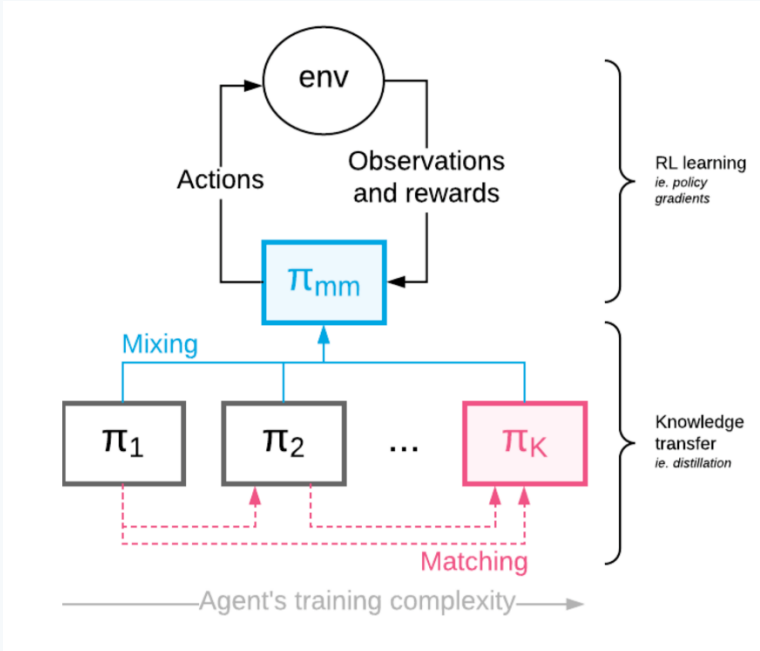
Hierarchical RL

- Hierarchical RL design enables greater non-linearity in learned behaviors
- Solve credit-assignment and tackle reward sparsity more efficiently
- Engender higher levels of “explainability” of agent behavior
- Takes full advantage of Population Based Self Play



Curriculum for Reinforcement Learning

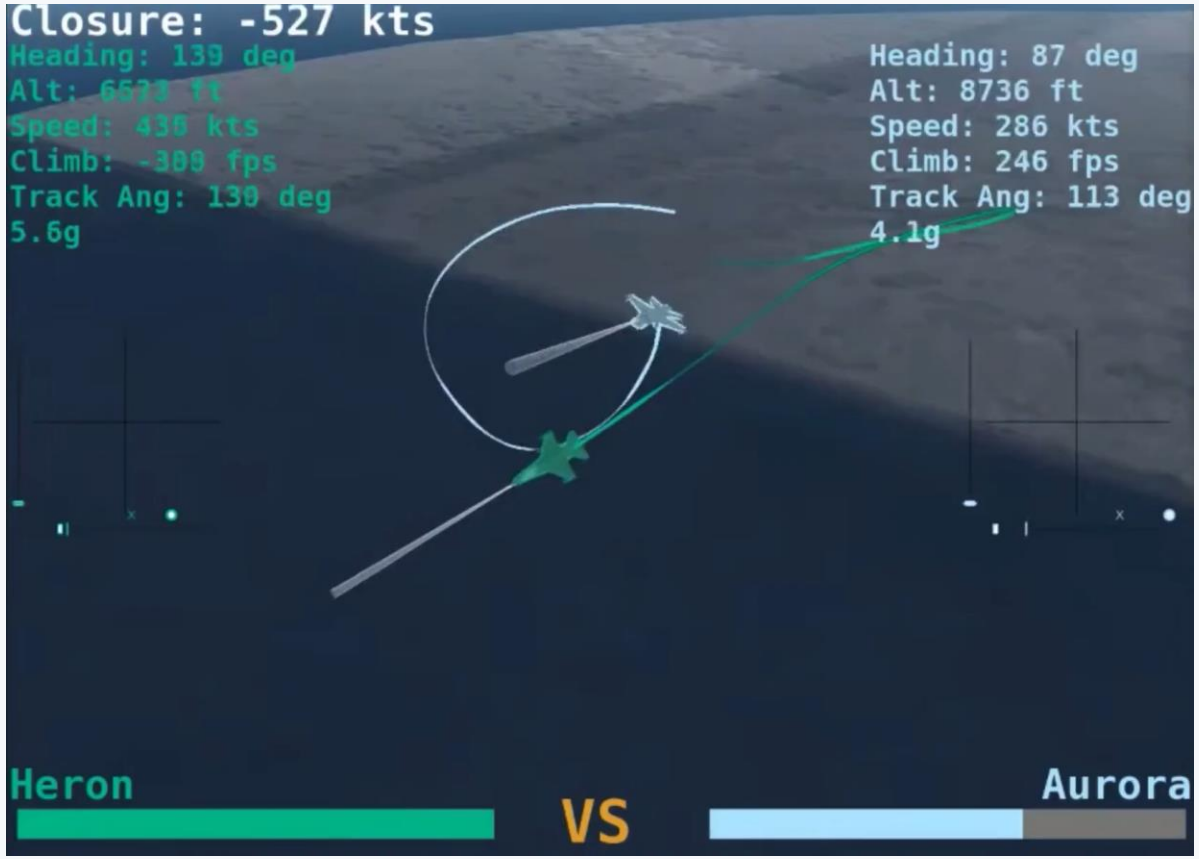
- Train a single policy on progressively more difficult tasks
  - Contrasts with training a policy from scratch on the desired task/data distribution
- Requires more reward tuning/design and environment design
- An aerial combat example:
  - Very Easy: Train agent starting directly behind opponent
  - Easy: Train agent from offensive initial conditions
  - Medium: Train agent starting from neutral conditions
  - Hard: Train agent starting from defensive initial conditions
  - Very Hard: Train agent starting directly in front of opponent



Czarnecki, Wojciech, et al. "Mix & match agent curricula for reinforcement learning." *International Conference on Machine Learning*. PMLR, 2018.

The AlphaDogfight Final Trials

- A multi-day publicized event held in Summer 2020
- Eight competitors faced off in a round-robin style tournament
- Aurora Flight Sciences placed 3rd



Key Takeaways

- Flexible software architecture is key
- Lean on domain experts when trust and explainability are desired
- Tools that enable scalable training and data-collection are key
- Performance and trust/interpretability are hard to balance!
- AI tools and algorithms can play a leading and effective role in the future of combat
  - …but much more work still needs to be done

Aurora FLIGHT SCIENCES
A Boeing Company

Workshop for the 2021 5th IEEE Conference on Control Technology and Applications (CCTA)

San Diego, California, August 8-11 , 2021

Multi-Vehicle and Assured Autonomous Control for Aerospace Applications

- Organized by the IEEE CSS Technical Committee on Aerospace Controls -

Questions?