Machine Learning 4 Controls: A Perspective from Space Controls Systems

Samir Bennani
Benedicte Girouart,
Massimo Casasco

21/10/2020

ADCSS2020

"This really is an innovative approach, but I'm afraid we can't consider it. It's never been done before."

Vision 4 Change: Evolving Landscape

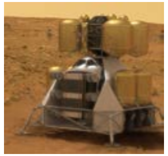
SPACE 1.0

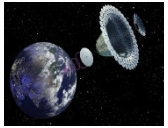
- Expandable Space Transportation Systems
- ISS
- Earth Observation / Communication
- Science/Astronomy/Exploration

NEW SPACE 4.0

LEO / GEO/ Beyond GEO LUNAR & Beyond

- BFR Generalised Space Transportation System (SpaceX)
- Space Assembly Large Structures /Repair Station Satellite Life extension / Debris Mitigation (ADRIOSS)
- Space Mining, Rare Earth Elements, Oxygen/Water, HE3 Manufacturing, ISRO, Tourism Repair Station, Manufacturing In Space
- Refueling and Propellant Depots
- Planetary Defense, Impacters,
- Deep Space Gateway / Lunar Observatory Human Outpost Distributed Space Solar Power (2Earth), Energy Harvesting
- Humans 2 Mars
- Deep Space Network, Space Internet, Communication Link Way Station





EMERGING SPACE

- Commercial Access and Exploitation of the Space Station
- Commercial Flight (Blue Origin – SpaceX)
- Commercial Micro Launchers
- Space Tourism
- Large Scale Constellations (5G)
- Cube Sats / New Services / Distributed / FF RS
- Re-usable Rockets – Precise Landing
- Space Control
- InOrbit Servicing / Space Tugs
- Commercial Space Transportation (Space X)
- Commercial Exploration Systems (Lunar X)
- Complex Cube SaT mission (RACE FF / RDV / Science etc)

Generalised Autonomy 4 Access 2 Space

Re-entry Technologies

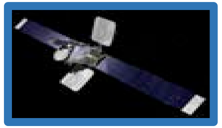
Technologies

- Vision and Hybrid Navigation Sensors and Algorithms / Precision Real Time Guidance
- Advanced Avionics / Autonomous GNC
- Health Management Systems / Mission Autonomy Manager
- Autonomous Precision Landing
- Multi-boost and Deep throttling Engine
- Clustered Multi-Engine
- Supersonic Retro-Propulsion
- Slosh Control / Active Loads Management
- Pneumatic Separation System
- Advanced Ballute / Parafoil / Inflatable
- Grid Fins / Novel EDL controls / Legs

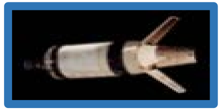
Capabilities

- 1st 2nd Stage Recovery / reuse
- Upper Stage Recovery / reuse
- Any Time Fairing and PL recovery / reuse
- Short time Inspection and Refueling
- Uppers Stage Long Duration Versatile Operation (RDV – Docking - Robotics)
- Safe Abort System
- Orbital Maneuvering
- Frequent & Rapid Access to Space
- Cheap Access to Space

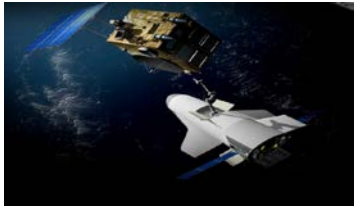
Enabling Missions



Reusable STS



Orbit Debris Mitigation

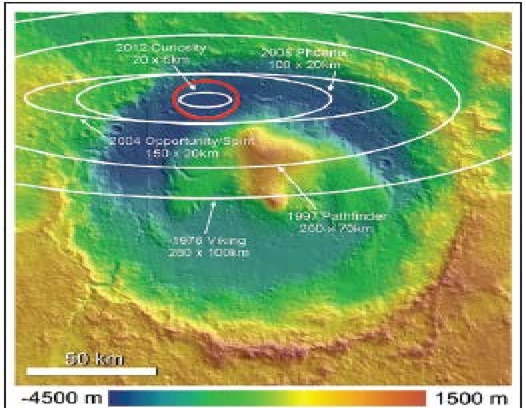


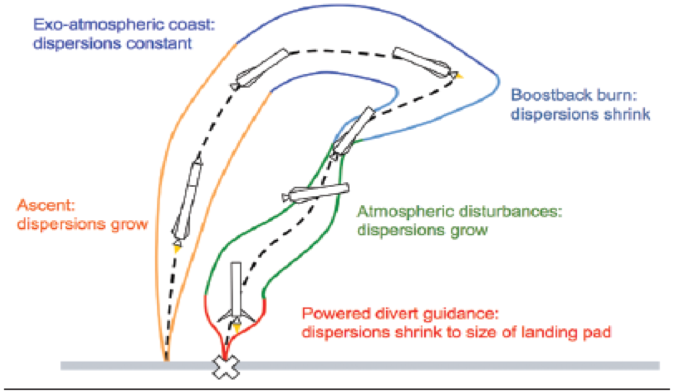
Space Rider Reusable Servicer

Docking to DSG

Technology Acceleration Platforms

Autonomous Re-entry & Entry Descent Landing





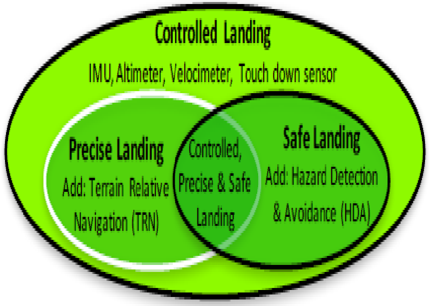
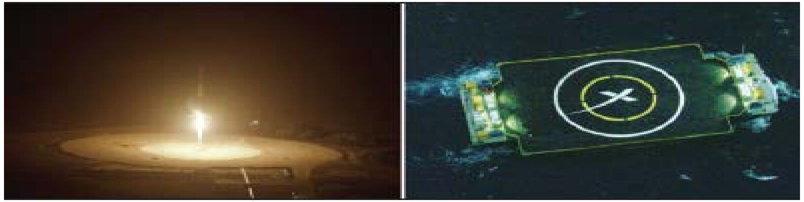
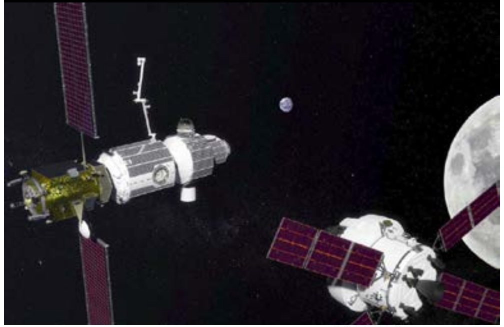


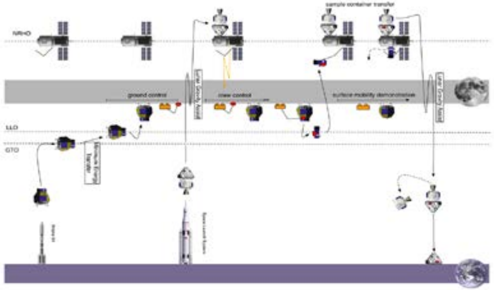
Figure 1. GN&C landing system capabilities enabled with PL&HA technologies.



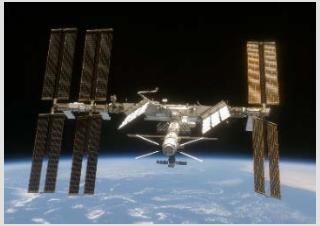
In Orbit Servicing & Assembly

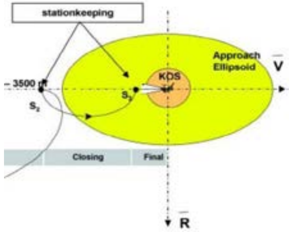
HERACLES LAE-GNC Phase A
+
NROGNC: RvD in NRO Orbits



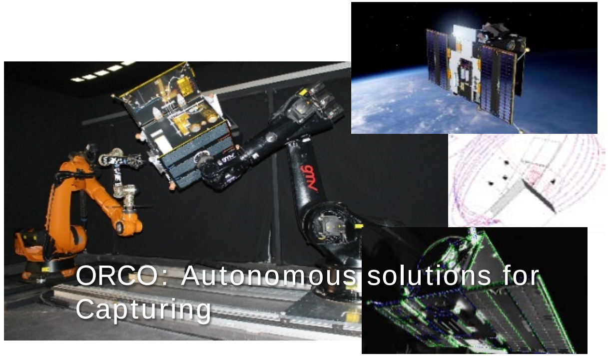


IOA-GNC: ADVANCED GNC FOR IN-ORBIT ASSEMBLY OF FLEXIBLE VEHICLES





ATV to ISS: RdV & Docking



ORCO: Autonomous solutions for Capturing

Orbital Robotic GNC Technologies

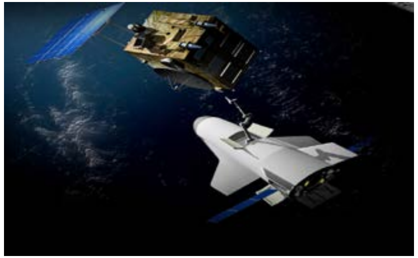
Technologies

- Relative Navigation Vision / Sensors and Algorithms & Advanced Real-time Guidance AOCS / GNC
- Advanced Integrated Avionics
- Integrated AOCS & Servicing Robotics
- Servicing Tools 4 Assembly
- Long duration Cryogenics
- Propellant Transfer – Re-fuel
- Mission Autonomy Manager & Control
- Berthing System
- Integrated MV Robotic Control
- Propellant Fuel Transfer - Cryogen Transfer
- Cooperative Control Servicing
- Integrated Modular Systems

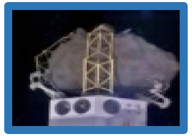
Capabilities

- Inspection- Servicing
- A. Rendezvous- Docking Berthing
- Capture - Assembly
- Refueling
- Satellite Relocation/Deorbiting
- Repair, Replenish
- In space manufacturing / assembly
- Mining
- Science

Enabling Missions

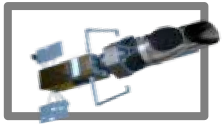


Space Rider

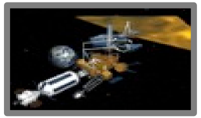


Asteroid Intercept

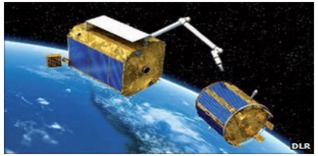
DSG



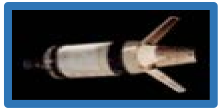
Cooperative Servicing



On-Orbit Assembly

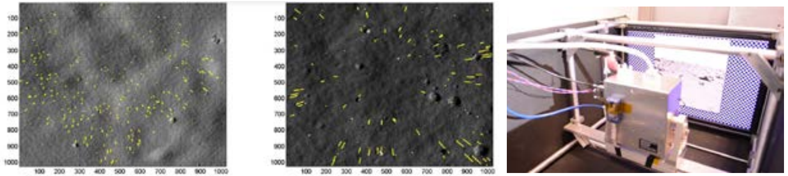


Servicing Space TUG

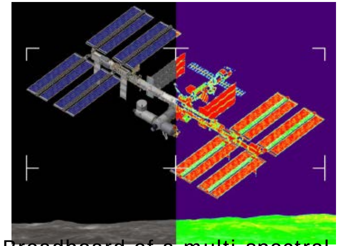


Debris Mitigation

Vision based navigation

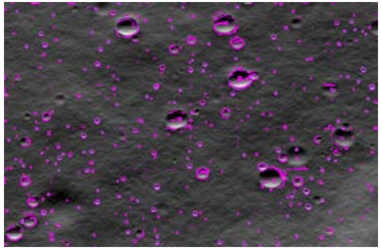


VISNAV-EM1 (Feature Detection and Real-Time Test Bench)

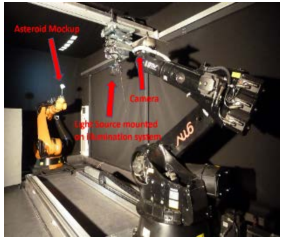


Breadboard of a multi-spectral camera for rendezvous in Lagrangian Orbits (HERACLES)

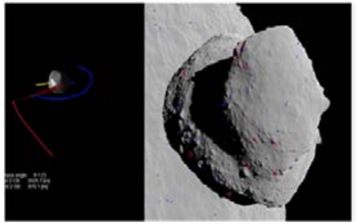
AVERT Flying Test Bench



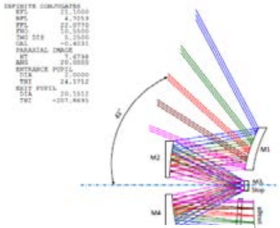
Neural Networks for Visual Navigation

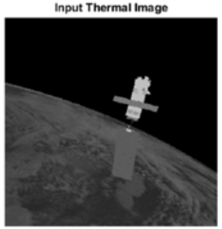


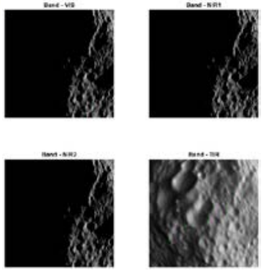
HILROB setup for CAMPHORVNAV



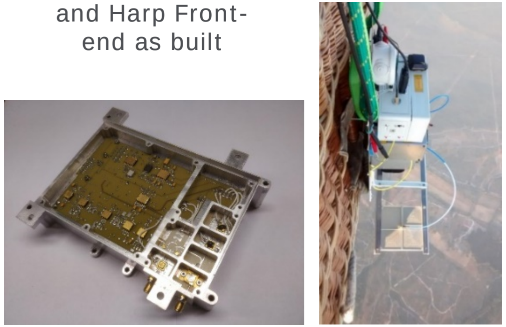
HERA: Landmark tracking navigation



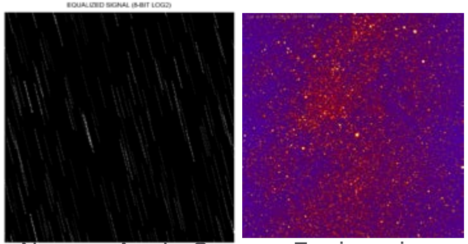




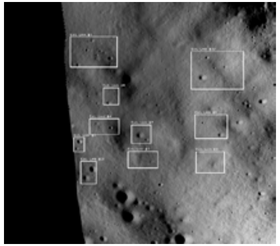
MSRN: Optical design, Thermal IR image Envisat and of asteroid in different bands (visible, Near IR, Thermal IR)



Radar Planetary Altimeter EM: Balloon flight tests and Harp Front-end as built



Narrow Angle Camera Engineering Model for MSR Rendezvous

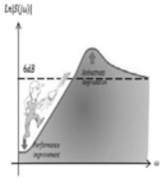


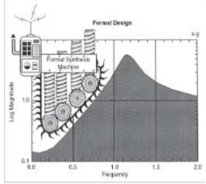
GENEVIS: Landmarks matched on the input image

Future Trends in Control

- Emergence of Machine Learning in Control
- Driven by Autonomous Decision making and Perception
- Machine Learning applications in Control
- How to Structure and Integrate the technologies over Disciplines?
  - CS and SW Machine learning
  - Physical Modelling
  - Dynamical Systems and Optimization

Controls an Evolving Field

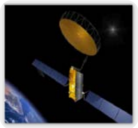


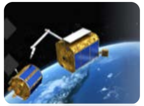


Classical View on Control 1.0

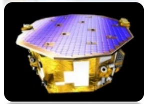
- Feedforward & Feedback tool to design of desired Dynamics via sensing, actuation and computation
- Feedback to stabilize, achieve desired performance (adaption,tracking), disturbance rejection, minimizes effort, achieves robustness

Evolving & Changing View 2.0





- **Formal mathematical framework** of tools, techniques, **to model, (control) design and analyse** complex interconnected systems in the presence of uncertainty and complexity
- **Theory that combines and optimises dynamics, interconnections, communication, computing & software**
  - Control = Glue(dynamics, uncertainty, feedback connection) = Desired System Behavior = **Augmented Systems**



Key Principle: Optimisation based on Conservation Laws

Feedback is everywhere

When conservation law is not respected feedback harms

- Feedback enables to manage optimal subsystem inter-operation and component uncertainty propagation at system level >>> Complexity management at System Level

Trends in Control 4.0

- **Embedded Optimisation, Layering architectures, Networked & Distributed Control, Formal Methods Cyber-physical Systems, Integrated AI –SW –Control- Complex Logistics Design & Management ...**

(inspired Murray CDS / Bode Lecture) **A NEW THEORY is on its Way**

Machine Learning vs Controls

Machine Learning

uses data to control system
reduce uncertainty

more data
→ better models/predictions

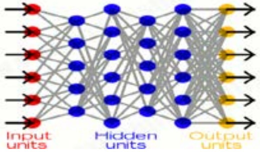
probabilistic guarantees

Control

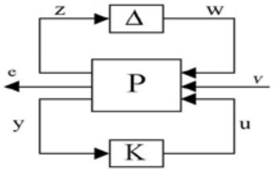
uses feedback to
mitigate uncertainty

better models/predictions
→ better performance

worst-case guarantees



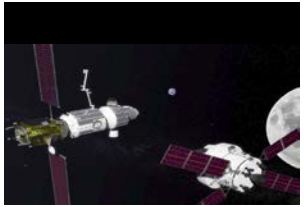
Can ML and Controls be combined so that we **safely achieve** better performance?
Where can ML help in the Process?

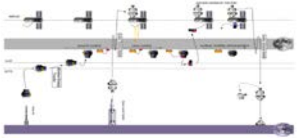


Why/When use ML & Data Based Techniques?

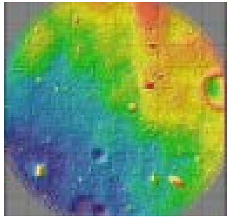
using past **data** to **learn** about
and/or
act upon the world
ONLINE

Environments
too complex

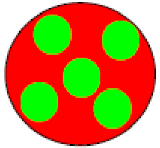


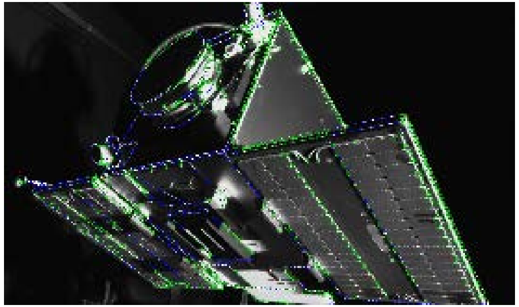


Sensing
too complex

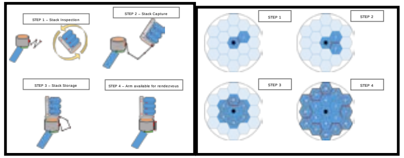


Hazard Detection
Algorithms





Dynamics & Models
too complex



Guarantees are (usually) **probabilistic**

Mapping Needs Machine Learning Techniques

Applications

Requirements

Complex Modelling & Simulation

Vision

Perception Filtering

Sensor Fusion

Navigation

Health Monitoring

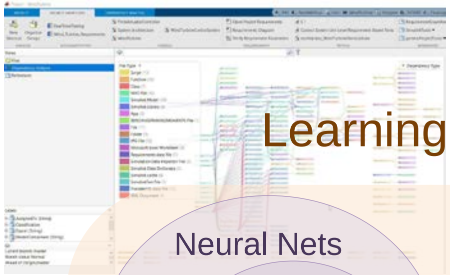
Decision Making

Planning Guidance

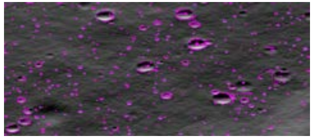
Controls & FDIR

Complexity Management

V&V .....



Learning Techniques



Neural Nets

Deep Learning

Statistical Models

AOGs

SVMs

Graphical Models

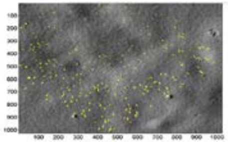
Bayesian Belief Nets

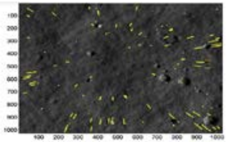
MLNs

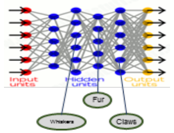
Markov Models

Ensemble Methods

Decision Trees







Specs.

Accuracy

Explainability

Robustness

Efficiency

....

Two Disjoint & Complementary Worlds

Aerospace / Mechanical / Electrical Eng.

Theory of Controls & Dynamical Systems

Model based
Physics
Theory
Optimisation
Stability
Structured @ Low Level Architectures

Computer Science / SW

Machine Learning Reinforcement Learning

Data based
Heuristics
Data Analysis
Linear Algebra
Stochastic Optimization
Structured @ High Level
Complex Systems

Integration

Aerospace / Mechanical / Electrical Eng.

Computer Science SW

Theory of Controls & Dynamical Systems

- Model based
- Physics
- Theory
- Optimisation
- Stability
- Structured @ Low Level
- Architectures

Optimization

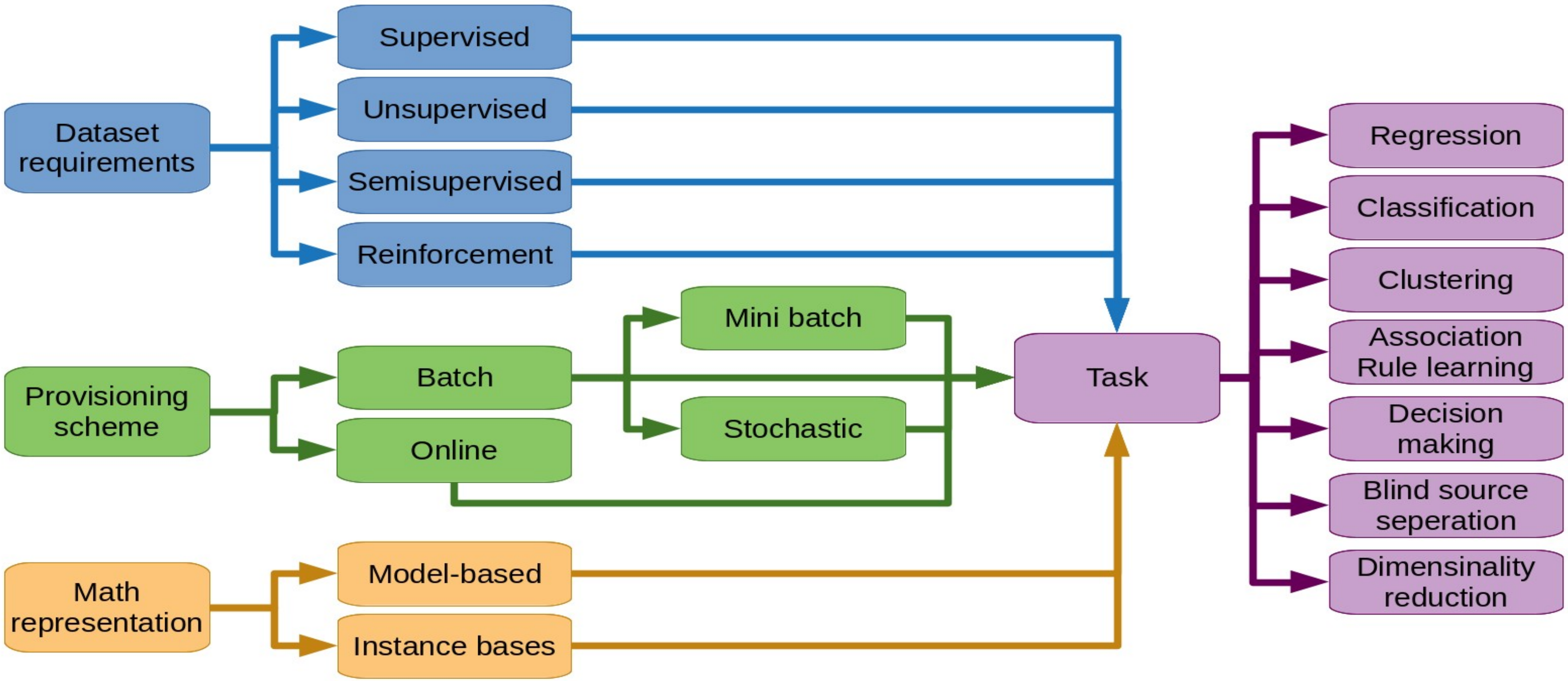
New Theory

New Tools

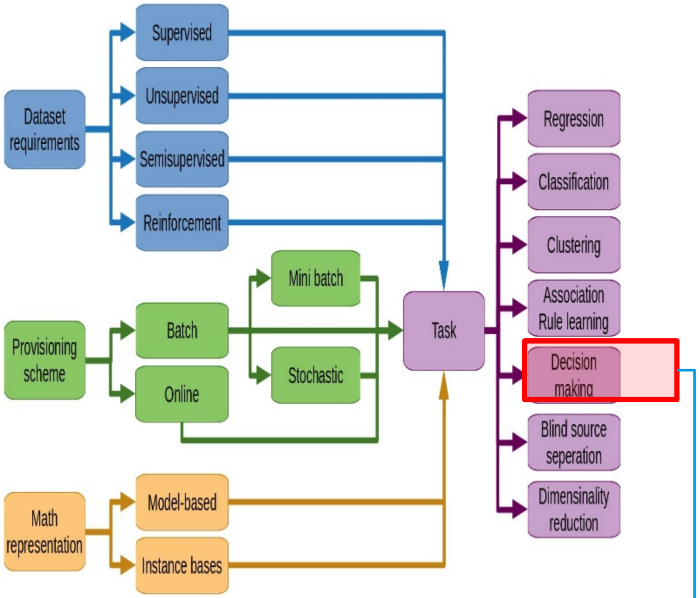
Machine Learning Reinforcement Learning

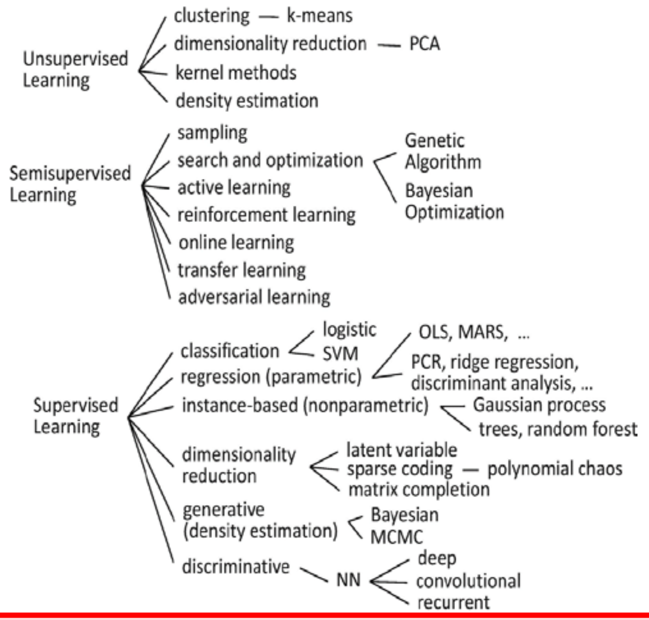
- Data based
- Heuristics
- Data Analysis
- Linear Algebra
- Stochastic Optimization
- Structured @ High Level
- Complex Systems

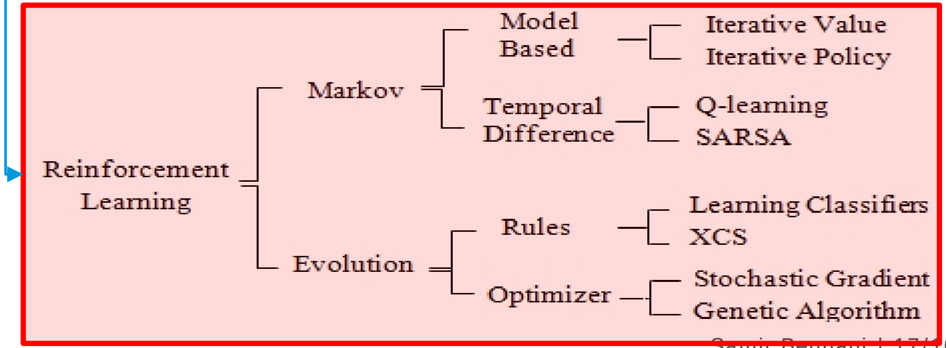
Possible ML Taxonomy



Possible Taxonomies







Facts……Extraordinary Claims Require Extraordinary Evidence

blog.openai.com/openai-baselines-dqn/

arxiv:1709.06560

Test and Benchmark various ML papers….

Results we find that implementation differences which are often not reflected in publications can have dramatic impacts on performance.

"Reinforcement learning results are tricky to reproduce: performance is very noisy, algorithms have many moving parts which allow for subtle bugs, and many papers don't report all the required tricks."
"RL algorithms are challenging to implement correctly; good results typically only come after fixing many seemingly-trivial bugs."

Deep Reinforcement Learning that Matters

Peter Henderson[1]*, Riashat Islam[1,2]*, Philip Bachman[2]
Joelle Pineau[1], Doina Precup[1], David Meger[1]
[1] McGill University, Montreal, Canada
[2] Microsoft Maluuba, Montreal, Canada

Outline

- Is robustness an issue in RL?
- Examples from optimal control (LQR/LQG)
- Proposed method to recover robustness
- Conclusions

Logic?

Use linear optimal control problems to understand performance of RL techniques

RL provides most benefits for problems that can't be addressed by standard system ID + linear optimal control
Obtaining insight in RL methods (explainability)
Use known LTI problems as "test" cases

Can one develop a Model-Free Real-Time Control Approach Robust to Uncertainties ?

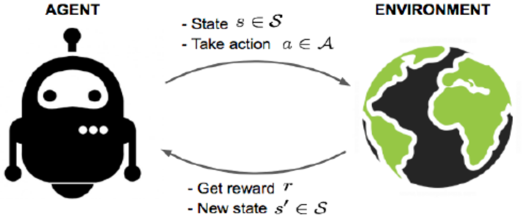
- Uncertainties from un-modeled system dynamics
- We shall comment on process noise wrt robustness....
- Connection with robustness recovery strategies,
    - Loop Transfer Recovery (LQG/LTR) (model-based)
    - Domain randomization (model-free)

“Model-Free” Reinforcement Learning

Goal: Train a control policy from data to maximize a cumulative reward

- Training data obtained from a simulator or the real system
- Often assume state feedback
- Many algorithms (Q-learning, value iteration, policy iteration, policy search) [1,2,3]
- Algorithms have close connections to dynamic programming and optimal control.

$$\underbrace{V^{\pi}(s)}_{\text{Value func.}} = \underbrace{r(s,\pi(s))}_{\text{Reward}} + \gamma \sum_{s'} \underbrace{p(s'|s,a)}_{\text{Dynamics}} V^{\pi}(s')$$

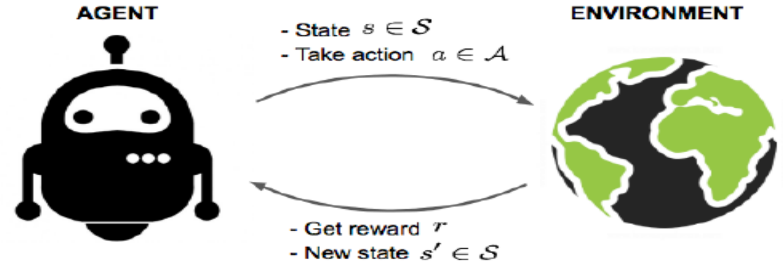


The “agent” is the controller and the “environment” includes the plant, uncertainty, disturbances, noise, etc.

$$V^{*}(s) = \max_{\pi} V^{\pi}(s)$$

[1] D.P. Bertsekas, “Reinforcement Learning and Optimal Control,” 2019.
[2] R.S. Sutton and A.G. Barto, “Reinforcement Learning: An Introduction,” 2018.
[3] C. Szepesvári, “Algorithms for Reinforcement Learning,” 2010.

RL General Setup



At each time step, agen [illegible] s a reward.

Goal for the agent: choose actions to maximize total discounted reward.

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

$$\underbrace{V^{\pi}(s)}_{\text{Value func.}} = \underbrace{r(s, \pi(s))}_{\text{Reward}} + \gamma \sum_{s'} \underbrace{p(s'|s, a)}_{\text{Dynamics}} V^{\pi}(s')$$

Optimal Action: **a**

Agent provides a Policy: $\boldsymbol{\pi}$ **,** is a control law (explicit or implicit as in MPC)

Can the Agent learn the optimal policy by suitable use of state and reward data?

RL: A general machine learning paradigm to solve problems and attain goals.

Logic?

Use linear optimal control problems to understand performance of RL techniques

RL provides most benefits for problems that can’t be addressed by standard approach

- Classical Modelling and System Identification Techniques

+

- Linear Optimal Control

Obtaining insight in RL methods (explainability) use known LTI problems as “test” cases

What About Robustness?

Δ represents un-modeled dynamics.

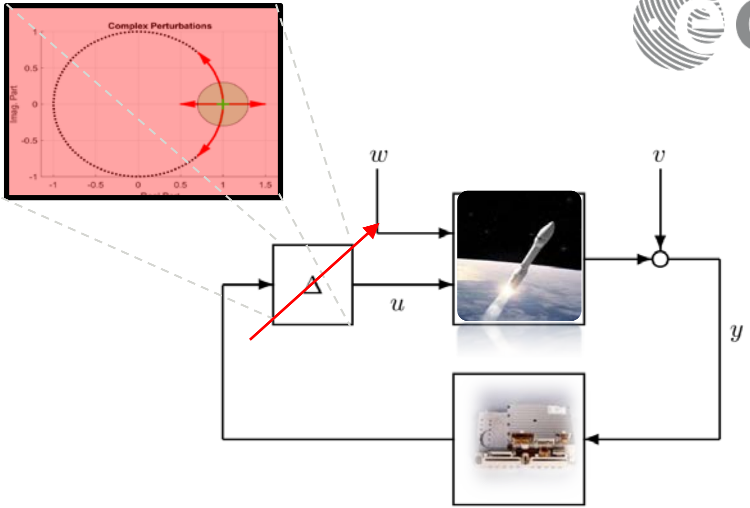
Combined Gain-Phase Margin given by complex function $\Delta(\Delta g^{\,j\,\Delta\theta})$

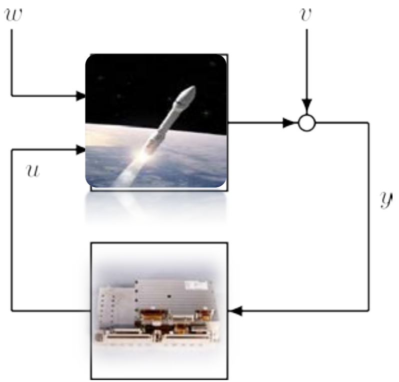
Classical cases:

Gain Margin : change in system gain (Δg),

Phase Margin: change in system phase ($e^{j\,\Delta\theta}$)

Disc Margin: simultaneous changes gain&phase ($\Delta g e^{j\,\Delta\theta}$)





[1] Venkataraman & Seiler, Recovering Robustness in Model Free Reinforcement Learning, '18 arXiv and ACC 2019

What About Robustness?

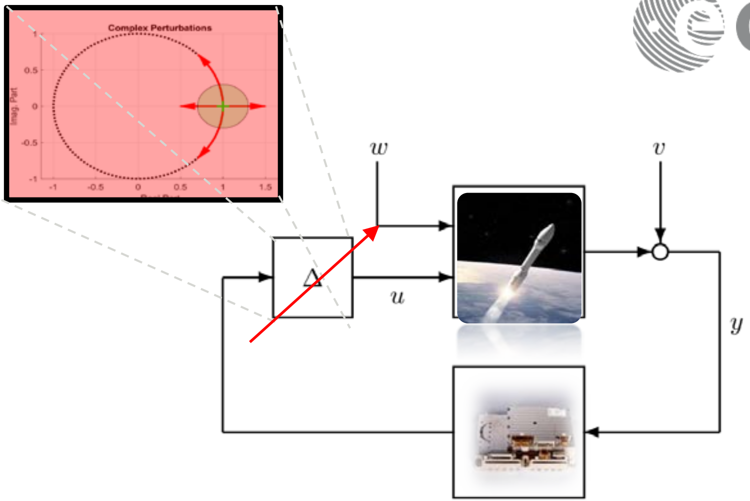
Δ represents un-modeled dynamics.

Combined Gain-Phase Margin given by complex function $\Delta(\Delta g^{\,j\,\Delta\theta})$

Classical cases:

gain (Δg), phase $^{\Delta\theta}$ ($e^{j\,\Delta\theta}$)

Simultaneous changes / complex ($\Delta g e^{j\,\Delta\theta}$)



[1] J. Doyle. Guaranteed margins for LQG regulators, IEEE TAC, 1978.
[2] Venkataraman & Seiler, Recovering Robustness in Model Free Reinforcement Learning, '18 arXiv and ACC 2019

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC-23, NO. 4, AUGUST 1978

Guaranteed Margins for LQG Regulators

JOHN C. DOYLE

Abstract—There are none.

Strategies?

Training on Simulation / Virtual Sim. vs Model

- Training can exploit modelling flaws [1].
- Loss of performance transitioning from simulator to real system (sim2real) [2].

Training on Real system / Virtual vs Real

- Changes in system dynamics (complicated Aero, Slosh, Thermal)
- Change in environment (gusts, ground effects, etc....)
- Product & Component Variations

[1] Recht, “A Tour of Reinforcement Learning,” arXiv, 2018.
[2] Peng, et al., “Sim-to-Real Transfer of Robotic..,” arXiv, 2018.
[3] Alonso etal: Virtual vs Real: arXiv:1703.01250

Linear Quadratic Regulator

Minimize
$$J_{LQ}(u) := \lim_{N\to\infty} \frac{1}{N} E\left[\sum_{t=0}^{N} x_t^T Q x_t + u_t^T R u_t\right]$$

Subject To:
$$x_{t+1} = Ax_t + Bu_t + B_w w_t$$

The optimal controller is a state-feedback: $u_t = -Kx_t$

Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator

Maryam Fazel[1], Rong Ge[2], Sham M. Kakade[1], and Mehran Mesbahi[1]

Gain *K* computed by solving a Riccati equation.
One can use gradient descent to solve LQR online [1]
This solution is model-based, i.e. it uses data *A*,*B*, etc
LQR regulators have provably good margins (±6dB, 60º).

[1] Fazel et al. , "Global convergence… LQR," 2018 arXiv:1801.05039

Linear Quadratic Gaussian Regulators LQG

Minimize
$$J_{LQ}(u) := \lim_{N\to\infty} \frac{1}{N} E\left[\sum_{t=0}^{N} x_t^T Q x_t + u_t^T R u_t\right]$$

Subject To:
$$x_{t+1} = Ax_t + Bu_t + B_w w_t$$
$$y_t = Cx_t + v_t$$

The optimal controller is an observer/state-feedback:
$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t + L(y_t - C\hat{x}_t)$$
$$u_t = -K\hat{x}_t$$

Gains *(K,L)* computed by solving two Riccati equations.
This solution is model-based, i.e. it uses data *A,B,C,* etc

Stability Margins for LQG Regulators

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC-23, NO. 4, AUGUST 1978

Guaranteed Margins for LQG Regulators

JOHN C. DOYLE

Abstract—There are none.

LQG regulators can have arbitrarily small margins [1].

[1] Doyle, “Guaranteed Margins for LQG Regulators,” TAC, 1978.

Stability Margins for RL With Output Feedback

Doyle's example [1] is a second-order system.
It can also be solved within RL framework using policy gradient search:

$$z_{t+1} = A_K(\theta)z_t + B_K(\theta)y_t$$
$$u_t = C_K(\theta)z_t$$

where

$$A_K(\theta) := \begin{bmatrix} 0 & \theta_1 \\ 1 & \theta_2 \end{bmatrix}, B_K(\theta) := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C_K^T(\theta) := \begin{bmatrix} \theta_3 \\ \theta_4 \end{bmatrix}$$

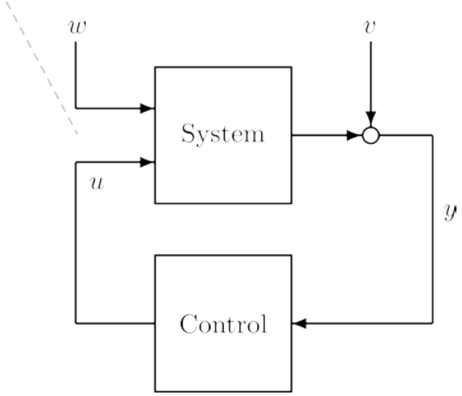
RL converges to the Optimal LQG control as data collection tends to infinity
RL has the same poor margins as theoretically derived….

[1] Doyle, "Guaranteed Margins for LQG Regulators," TAC, 1978.

Markov Decision Process

Markov Decision Processes (MDPs)

- Sets of states, *S*, and actions, *A*
- Reward function, $r: S \times A \rightarrow \mathbb{R}$
- State transition probability, *T*



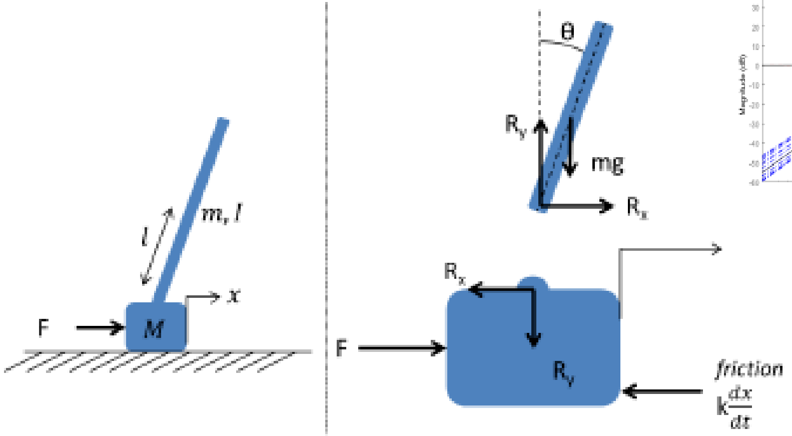
Synthesize a control policy from input/output data to maximize the cumulative reward:

$$J_{RL}(a) := E\left[\sum_{t=0}^{N} r(s_t, a_t)\right]$$

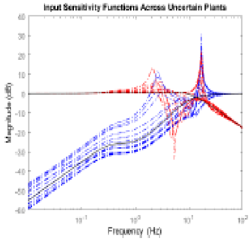
- o LQR is a special case of this RL formulation using MDPs [1].
- o LQG is a special case using Partially Observed MDPs (POMDPs).

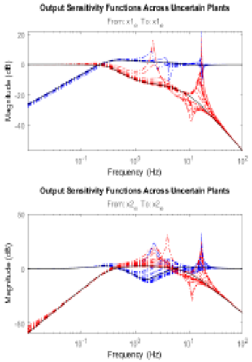
[1] Recht, "A Tour of Reinforcement Learning," arXiv, 2018.

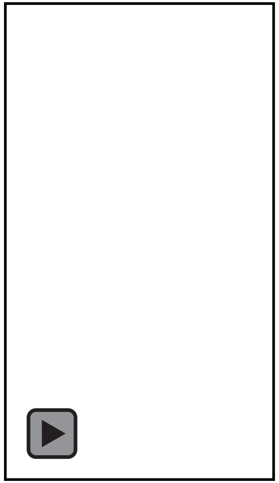
Inverted Pendulum



[ht]







$$\begin{bmatrix} \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{-mgl}{D_0} & \frac{-4lk}{3D_0} & 0 \\ \frac{(M+m)g}{D_0} & \frac{-k}{D_0} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{4l}{3D_0} \\ \frac{-1}{D0} \end{bmatrix} u$$

Unmodeled dynamics can also de-stabilize LQR/LQG controllers >> All state-feedback RL controllers.

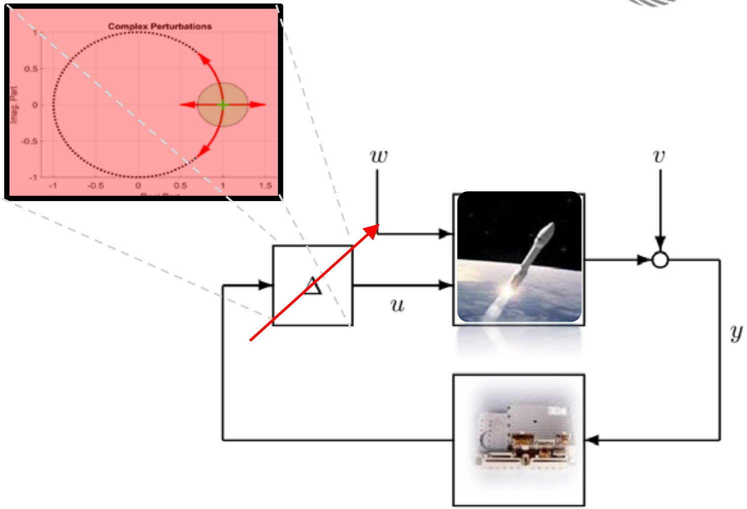
Margin Recovery

- Is robustness an issue in RL?
- Examples from optimal control (LQR/LQG)
- **Proposed method to recover robustness**
- Conclusions

Robust LQG Design

Inject synthetic gain/phase variations during the training.

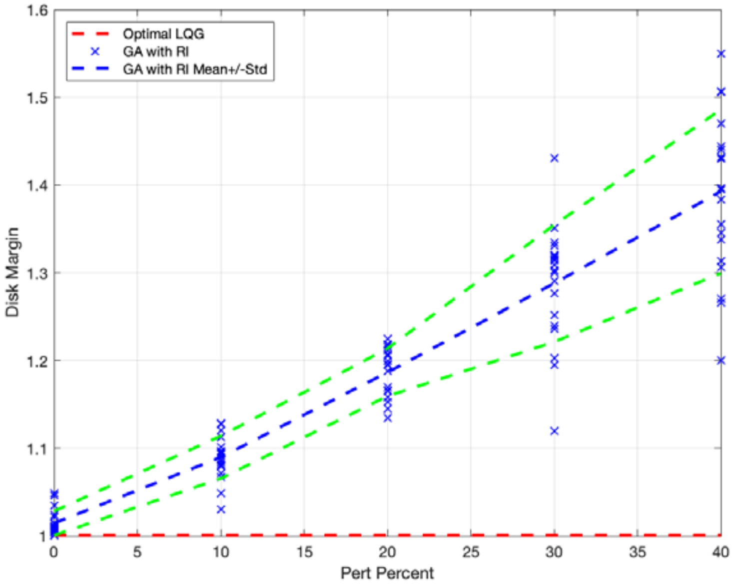
<u>For initial tests:</u>
$\Delta=1+\delta$ where δ
is uniformly sampled in *[-b,b]*



Method tested using policy gradient with random restarts.

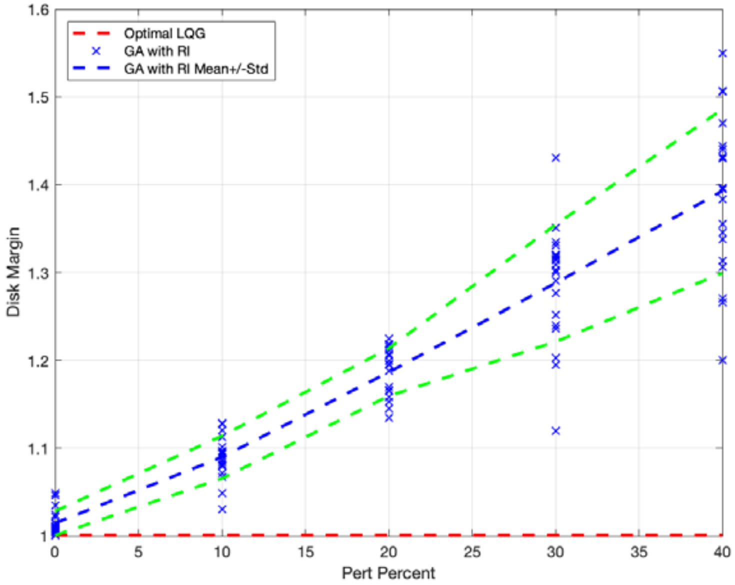
Policy cost and gradient typically computed by averaging over simulated trajectories.
For efficiency, these computations were done analytically using the model data.

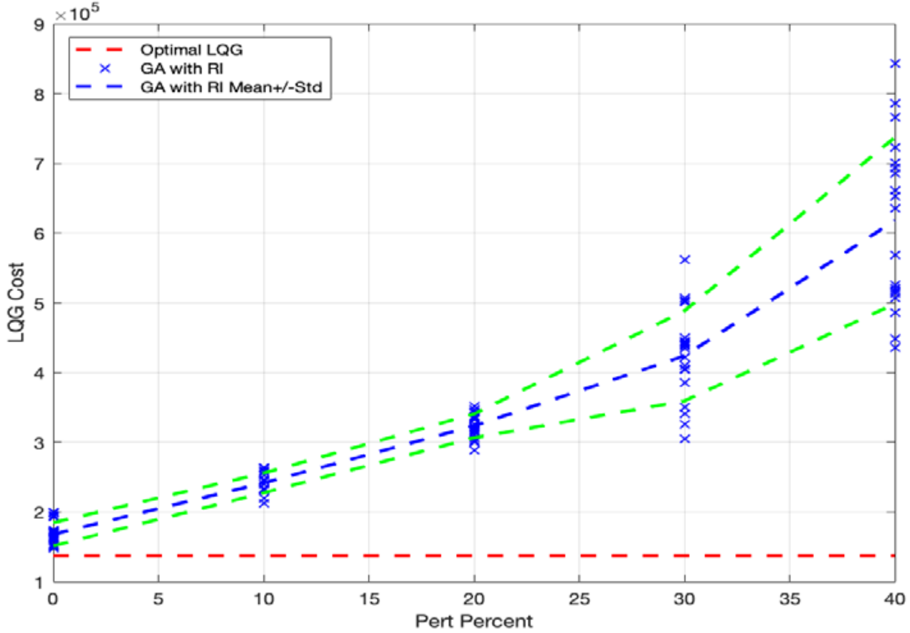
Robustness ↑ with Perturbation parameter b↑



Results: Performance

Performance ↓ with Perturbation parameter b↑





Conclusion set 1

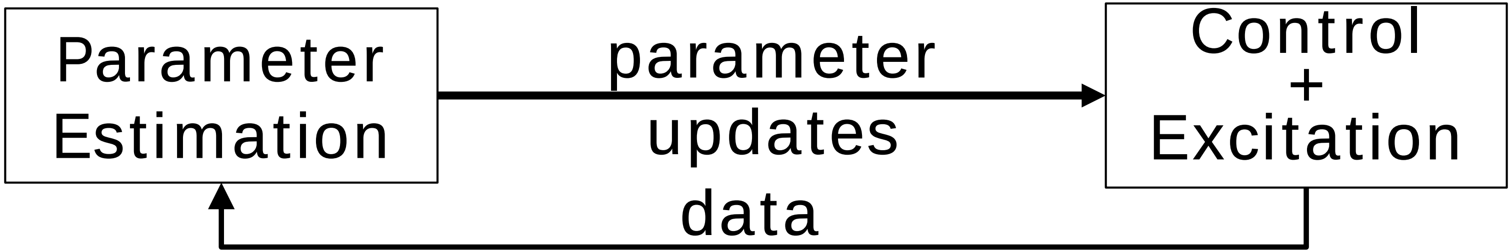
For systems we know to model and manage

- ML is only as good as what the designer specifies and models
- ML is understandable via adoption of design principles from controls: toward Model Based – Machine Learning – or Physics Based Machine Learning (Model Predictive Control….)
- ML robustness can be recovered via proper uncertainty modelling
- ML can recover any design methods…Not efficient …for systems we know

Data based – Unknown System: From Off-line 2 On-line

- MB - ML control for Embedded Systems >>> **SYSTEM LEVEL SYNTHESIS**
  - new control theory for complex embedded systems
  - new approaches for the control of Large Scale Networks/Distributed Systems/Partial Differential Equations
  - new efficient real-time optimisation technologies for on-line anticipation, modelling, estimation, filtering, perception and control (MPC, Adaptive, LPV, SLS)
  - real-time robustness, performance guarantees at System level (Avionics/SW/System/Physics)

From Off-Line to Online



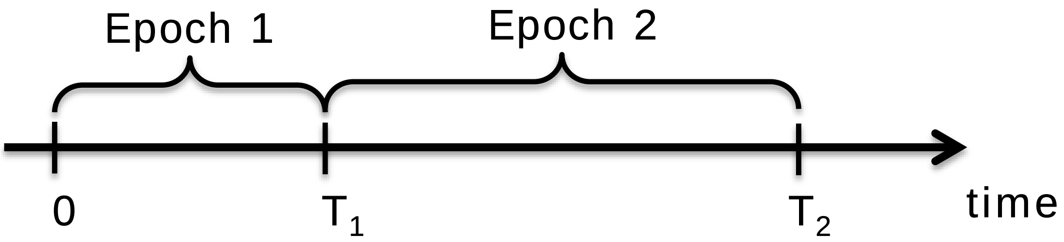
Self-tuning control:
fixed unknown parameters

Adaptive control:
varying unknown parameters

LTI system: OLS + MV Control = Optimal Policy!

On-line Robust System Level Synthesis Control

At every T_i do:



- $$(\hat{A}^{(i)}, \hat{B}^{(i)}) = \underset{(A,B)}{\operatorname{argmin}} \sum_{t \in E_i} \|x_{t+1} - Ax_t - Bu_t\|_2^2$$
- $$\mathbf{K}^{(i)} = \mathrm{RobustSLS}(\hat{A}^{(i)}, \hat{B}^{(i)}, \underline{\epsilon^{(i)}})$$ **sharp bounds from time-series data?**
- $$\mathbf{u}^{(i)} = \underline{\mathbf{K}^{(i)}\mathbf{x}} + \underline{\eta^{(i)}}$$ **explore vs. exploit?**

Self-Tuning Control RL Loop

Case studies:

- *Self tuning* control of unknown system
- Learning to control unknown discrete systems
- Optimization based approach to exploration and exploitation

Emphasis:

- Finite data guarantees
- Provable stability, robustness, performance
- Quantitative comparison of algorithms

Key technical tools:

- Markov Decision Processes
- Concentration inequalities
- Robust & optimal control /Koopman Theory/ Real-time System Identification (Subspace like) / System Level Synthesis / LPV control / Model Predictive Control / Adaptive Control / Estimation Filtering Perception / Integral Quadratic Constrained Control etc..

Further Reading

- S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, On the sample complexity of the linear quadratic regulator, Journal of Foundations of Computational Math, 2019.

- S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, Regret bounds for robust adaptive control of the linear quadratic regulator, NeurIPS, 2018.

- S. Dean, S. Tu, N. Matni, B. Recht, Safely learning to control the constrained linear quadratic regulator, IEEE American Control Conference, 2019.

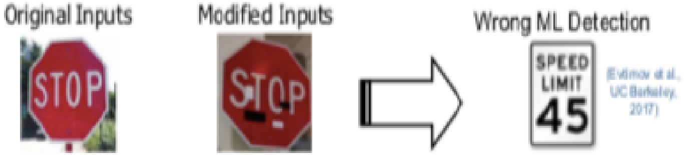
- Y.-S. Wang, N. Matni, and J. C. Doyle, A system level approach to controller synthesis, IEEE Transactions on Automatic Control, 2019.

- H. Mania, S. Tu, B. Recht, Certainty Equivalent Control for LQR is Optimal, NeurIPS, 2019.

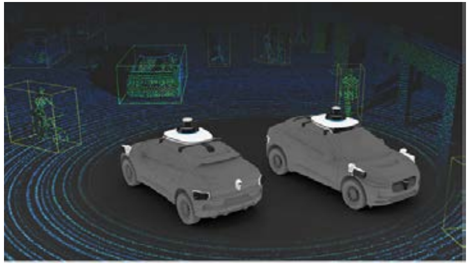
Challenges in Robust Deep Learning Systems

Change in lighting
(DeepXplore SOSP’17)

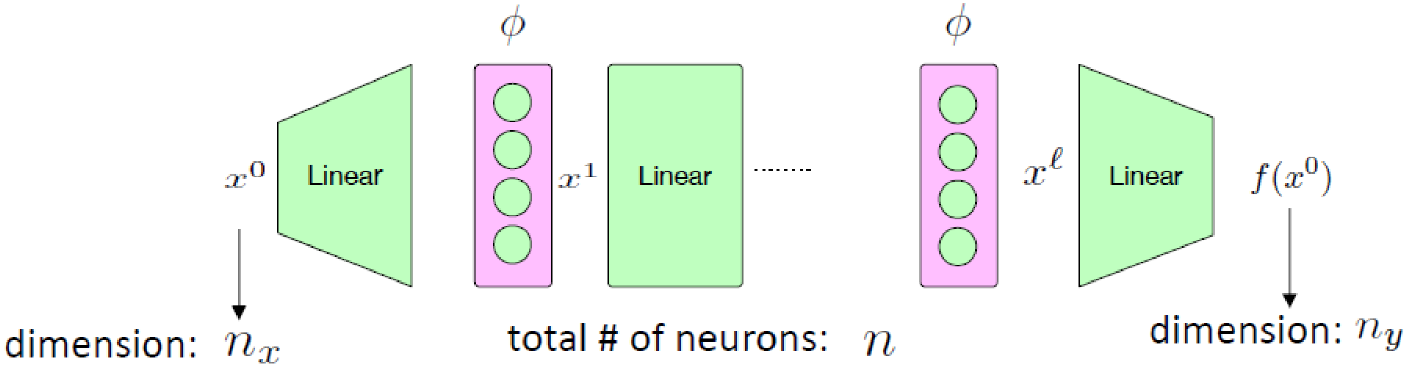
IEEE SPECTRUM



Physical attacks
(EEF+’15)



Neural Net as a Dynamical System

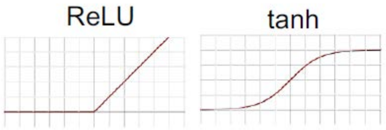


$$x^{k+1} = \phi(\underbrace{W^k x^k + b^k}_{\text{linear layer}}) \quad k = 0, \cdots, \ell - 1$$

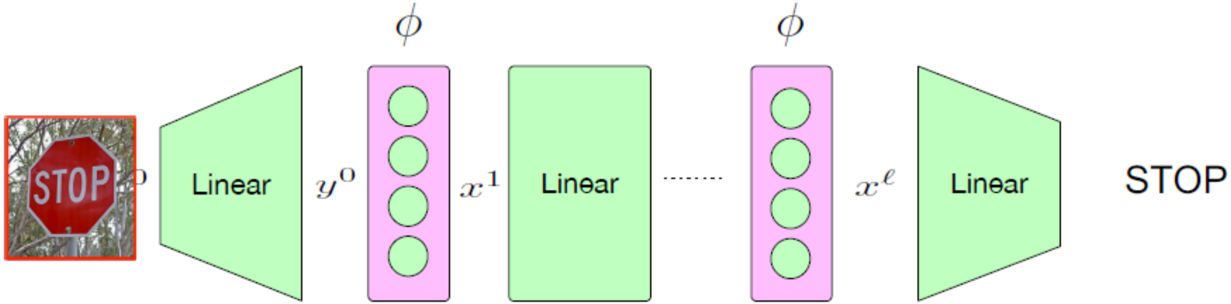
activation layer

linear layer

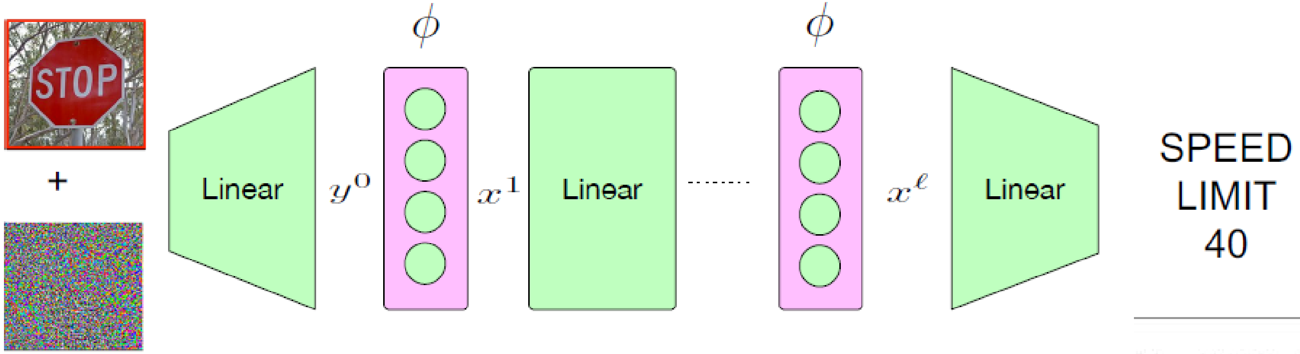
# of layers

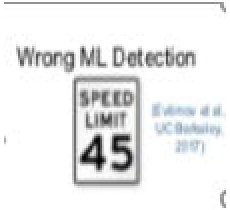


Fragility of Deep Networks   Nominal

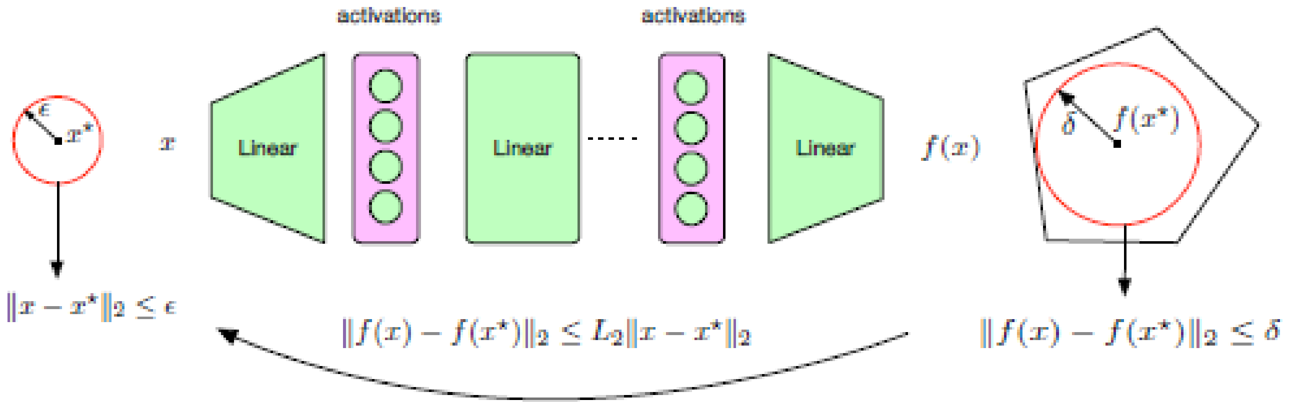


Fragility of Deep Networks Perturbed





Lipschitz Constants for Deep Neural Networks



$$L = \sup_{x,y \in \mathcal{X}} \frac{\|f(x) - f(y)\|}{\|x - y\|}.$$

A tight upper bound on L generally useful to:

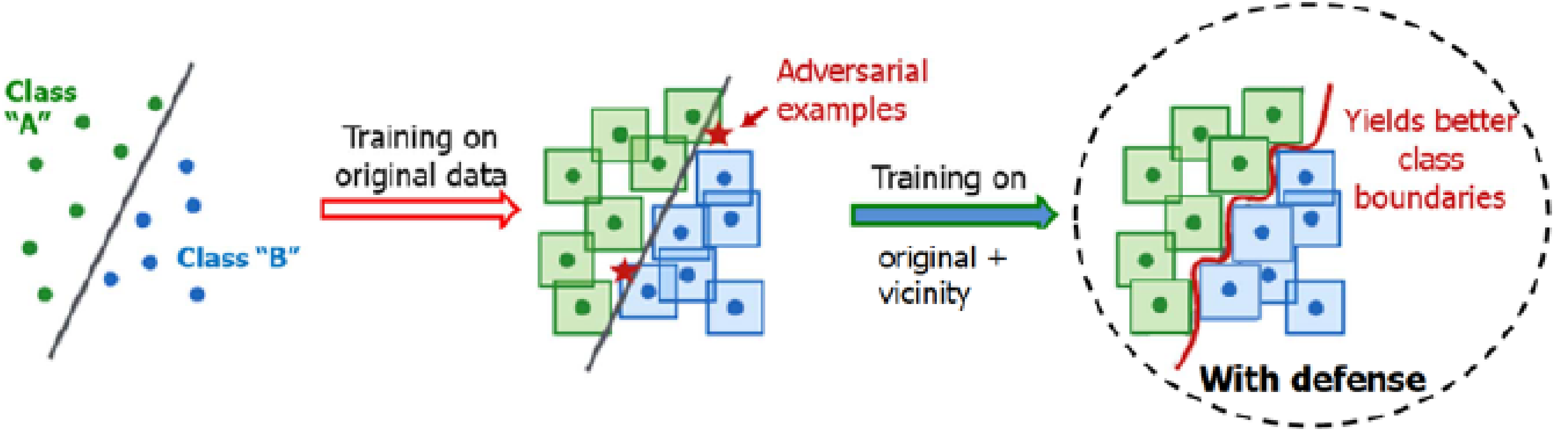
– Robustness certification of classifiers
– Closed-loop stability analysis of neural network controllers
– Robust training
– Generalization bounds

Finding L_2 is NP hard

Robustness of Deep Networks

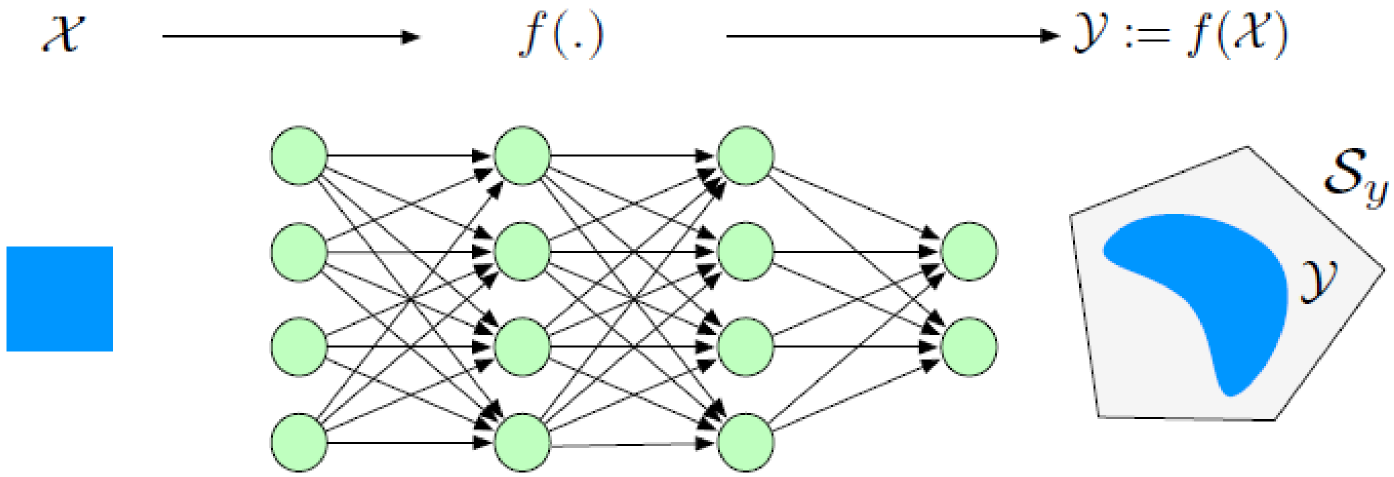
Robust & Adversarial Training

$$\text{minimize}_\theta \ \mathbb{E}_{(x^\star, y^\star)}[\max_{\delta \in \Delta} \ \text{loss}(f_\theta(x^\star + \delta), y^\star)]$$

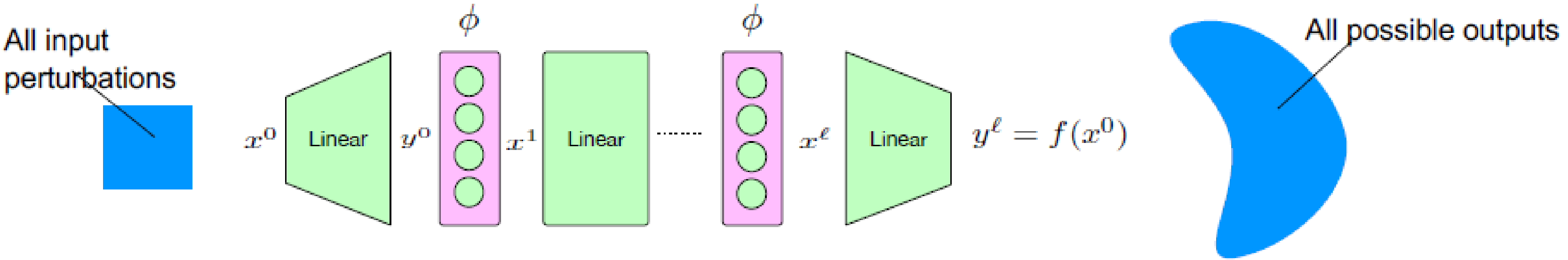


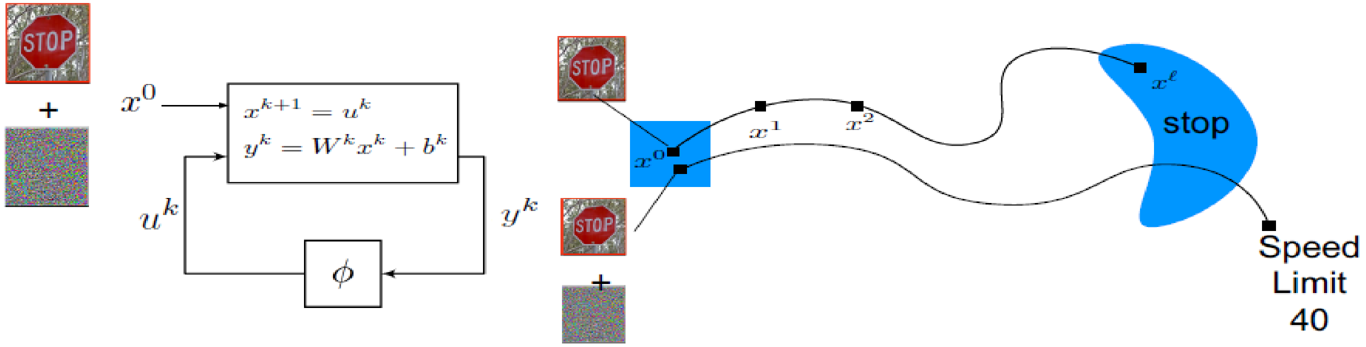
Robustness of Deep Networks

Post training verification of deep networks



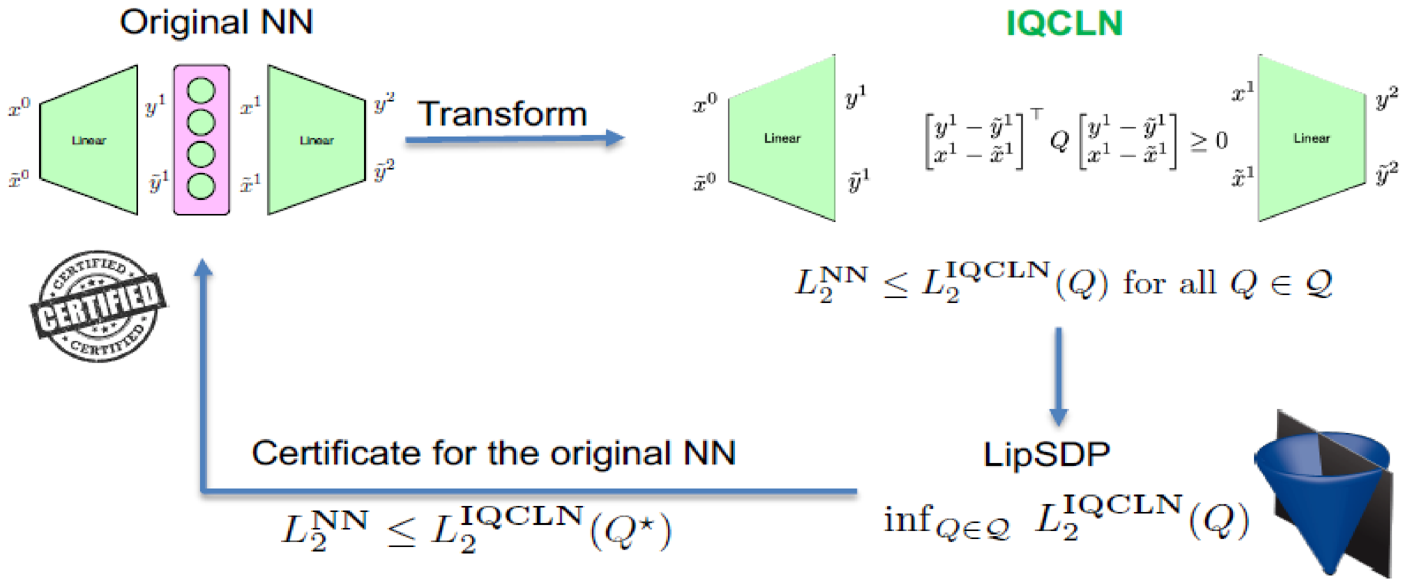
Neural Net as a Dynamical System





Incrementally Quadratically Constrained Linear Network

Any property proven for the IQCLN hold for the Original Network



https://github.com/arobey1/LipSDP

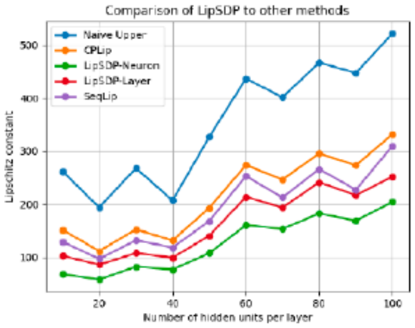
Comparative Analysis

Platform: MATLAB, CVX toolbox, and MOSEK on a 9-core CPU with 16GB of RAM

Variants of **LipSDP**: **LipSDP-Network, LipSDP-Neuron, LipSDP-Layer**

CPLip: Combettes, Patrick L., and Jean-Christophe Pesquet. "Lipschitz Certificates for Neural Network Structures Driven by Averaged Activation Operators." arXiv preprint arXiv:1903.01014(2019).

SeqLip: Virmaux, Aladin, and Kevin Scaman. "Lipschitz regularity of deep neural networks: analysis and efficient estimation." Advances in Neural Information Processing Systems. 2018.

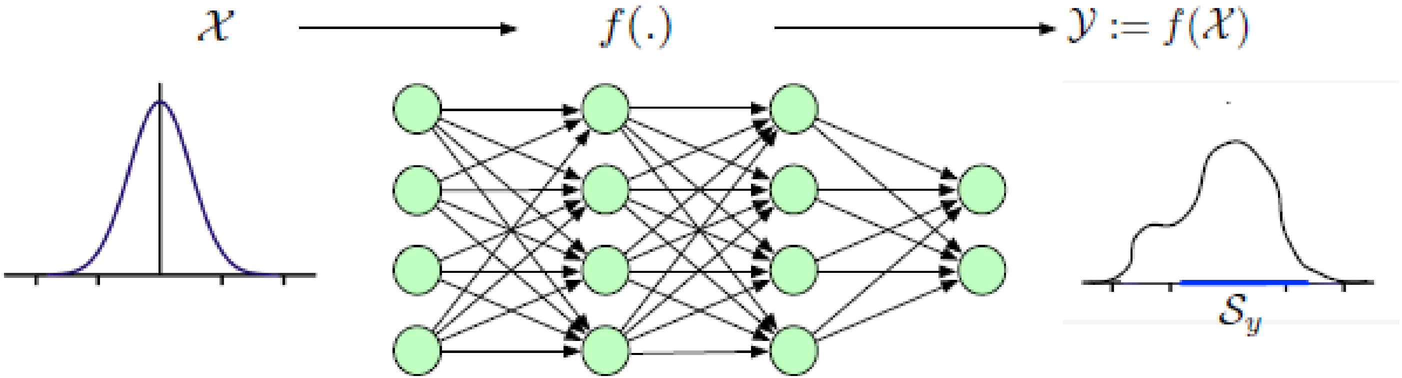


Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, George J. Pappas
Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks
arXiv:1906.04893 [cs.LG]

https://github.com/arobey1/LipSDP

Probabilistic Approach

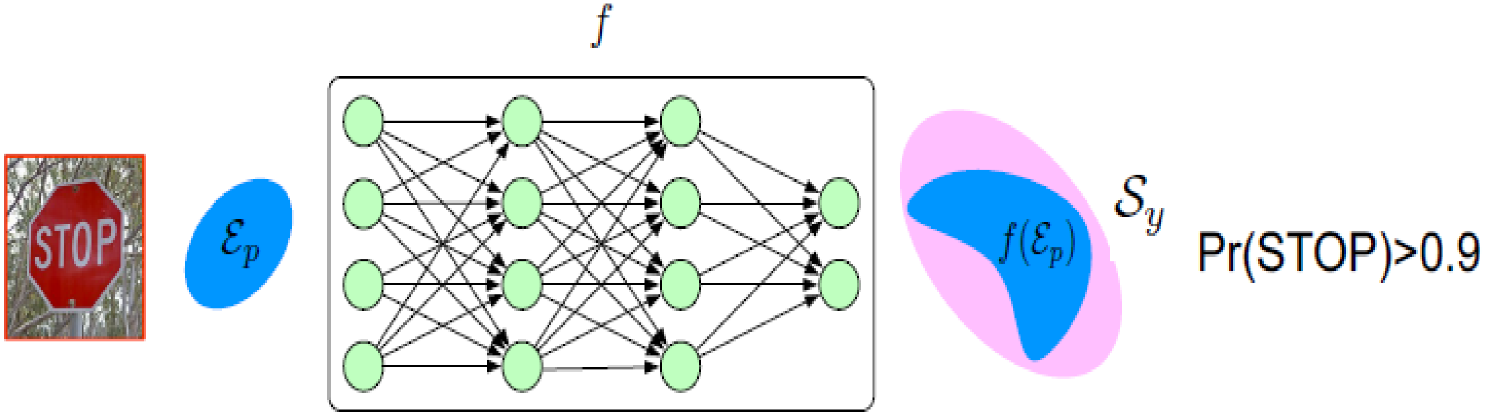
Input uncertainty can be random / no bounded constraints



$$\mathbf{Pr}(f(x) \in \mathcal{S}_y) = ?$$

Probabilistic Approach

SDP to propagate confidence ellipsoids [Fazlyab et al., CDC, 2019]



$$\text{minimize Volume}(\mathcal{S}_y) \quad \text{subject to } f(\mathcal{E}_p) \subseteq \mathcal{S}_y.$$

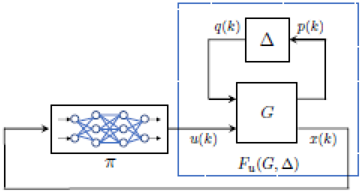
$$\mathbf{Pr}(f(x) \in \mathcal{S}_y) \geq \mathbf{Pr}(f^{QCLN}(x; Q) \in \mathcal{S}_y).$$

Further Reading

arXiv:2004.07876
Haimin Hu, Mahyar Fazlyab, Manfred Morari, George J. Pappas
Reach-SDP: Reachability Analysis of Closed-Loop Systems with Neural Network Controllers via Semidefinite Programming

arXiv:1903.01287
Mahyar Fazlyab, Manfred Morari, George J. Pappas
Safety Verification and Robustness Analysis of Neural Networks via Quadratic Constraints and Semidefinite Programming

arXiv:2006.07579
He Yin, Peter Seiler, Murat Arcak
Stability Analysis using Quadratic Constraints for Systems with Neural Network Cont



Conclusion

- Can we make ML learning understandable? yes...
- Can we make ML learning robust? yes...
- Can we make ML learning verifiable? Yes...
- Can we make ML Efficiently Real-Time Implementable? ... Yes
  - This shall all be based on novel strong results from Controls...
    - Towards a robust embedded model based machine learning theory..

ELECTRIC LIGHT DID NOT COME FROM THE CONTINUOUS IMPROVEMENT OF CANDLES

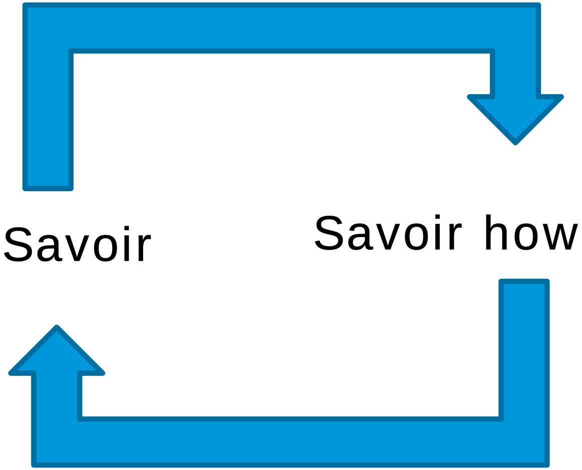
Oren Harari....

If you want 2

You need 2

Learn 2 Savoir

Savoir how 2 Learn



We are in Control

Credits 2

This presentation was inspired by

John Doyle, Andy Packard,

Richard Murray, Nikolai Matni, Benjamin Recht

Stephen Boyd etal..

Pete Seiler & Laurent Lessard

Georges Papas, Manfred Morari, Mayer Fayzlap

Brunton & Mesic

Carsten Scherer, Megretski, etc...

Lenhart Ljung, Karl Amstrong

... and a new generation of brilliant minds....

.....and of course ESA and Colleagues...