Verification for Machine Learning, Autonomy, and Neural Networks Survey

by Weiming Xiang, Patrick Musau, Ayana A. Wild, Diego Manzanas Lopez, Nathaniel Hamilton, Xiaodong Yang, Joel Rosenfeld, Taylor T. Johnson

Eric Langlois

University of Toronto

Feb. 11, 2019

Table of Contents

1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software

Table of Contents

1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software

Control Theory

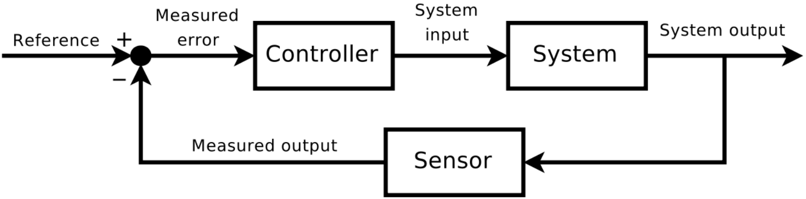


Figure by Wikimedia user Orzetto | CC BY-SA 4.0

Stability in Control Theory

If the system is near an equilibrium, will it stay there?

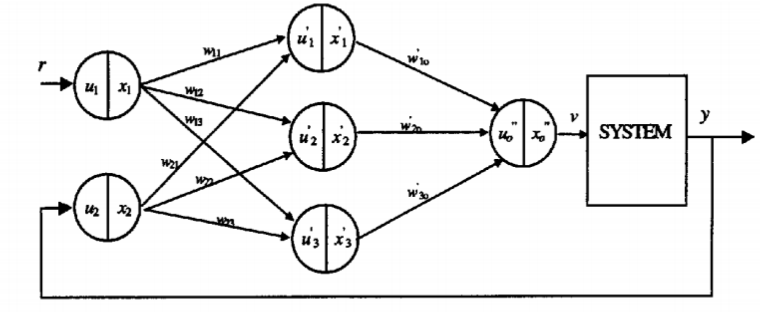
Lyapunov stable: For any radius ϵ, if we start close enough ($< \delta$) to the equilibrium, then the system will stay close ($< \epsilon$) forever.

Asymptotically stable: If we start close enough ($< \delta$) to the equilibrium then the system will converge to the equilibrium.

Table of Contents

1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software

- Tiny Neural Network Proportional-Integral-Derivative (PID) Controller
- Learn weights online



Huailin Shu and Youguo Pi. "PID neural networks for time-delay systems". In: *Computers & Chemical Engineering* 24.2-7 (2000), pp. 859–862.

Cited Work on Intelligent Control

- Neural Network Controllers
    - Multilayer Perceptron
    - Diagonal Recurrent
    - Radial Basis Function
- Model Predictive Control
- Online learning methods
- Often guarantee Lyapunov stability (under assumptions)
- Semi-globally uniformly ultimately bounded
- Stability while learning the optimal policy

Model Predictive Control

Controller performs optimization over predicted future state

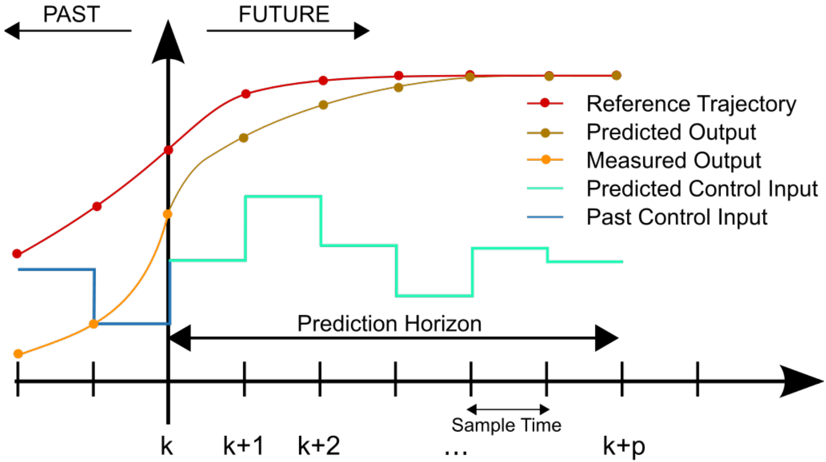


Figure by Martin Behrendt | CC BY-SA 3.0

Introductory Papers on Neural Network Control

- Foundational paper: Kumpati S. Narendra and Kannan Parthasarathy. "Identification and control of dynamical systems using neural networks". In: *IEEE Trans. Neural Networks* 1.1 (1990), pp. 4–27
- Survey of NN control systems: Toshio Fukuda and Takanori Shibata. "Theory and applications of neural networks for industrial control systems". In: *IEEE Trans. Industrial Electronics* 39.6 (1992), pp. 472–489
- Behnam Bavarian. "Introduction to neural networks for intelligent control". In: *IEEE Control Systems Magazine* 8.2 (1988), pp. 3–7

Table of Contents

1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software

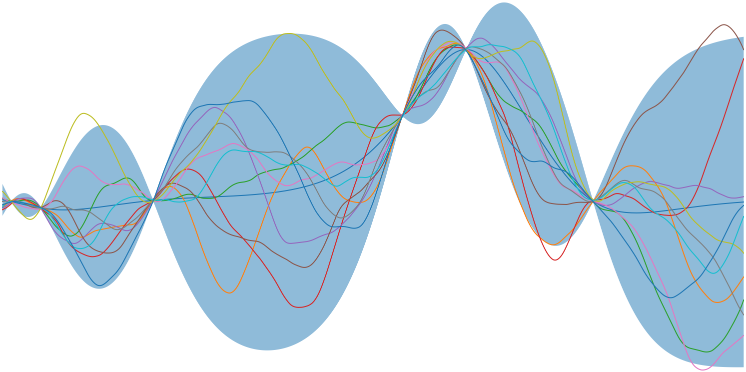
Specification Learning

Referenced results:

- Learn Signal Temporal Logic (STL) formulas from positive examples traces only
- Scalable framework to mine STL specifications from a closed-loop model of a system's behaviour
- Boolean formula learning
- Modelling system uncertainty with Gaussian Processes (GP)
  - Apply reachability analysis: verify unsafe states avoided

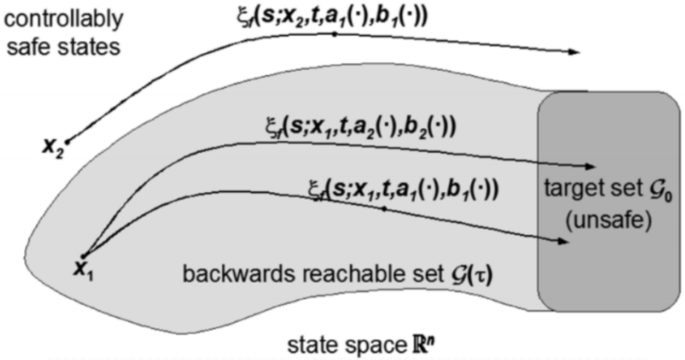
Gaussian Process

Represents an uncertainty distribution over functions.



Hamilton-Jacobi Reachability

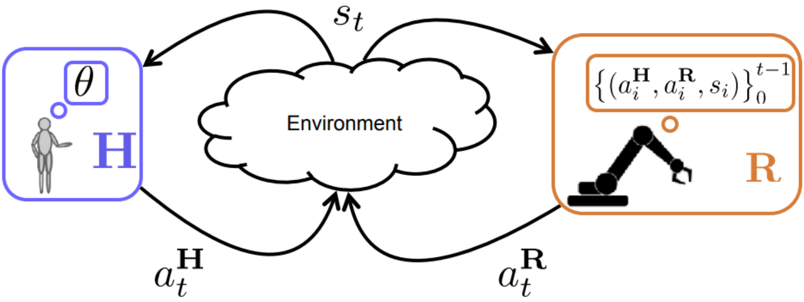
Can worse-case dynamics force the system into undesirable states?



Somil Bansal et al. "Hamilton-Jacobi reachability: A brief overview and recent advances". In: *CDC*. IEEE, 2017, pp. 2242–2253.

Value Alignment

Ensure objectives of AI systems match human values.



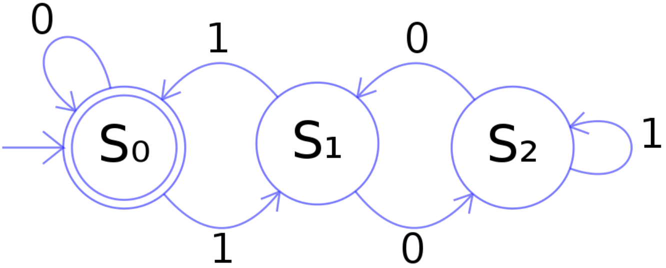
Dylan Hadfield-Menell et al. "Cooperative Inverse Reinforcement Learning". In: *NIPS*. 2016, pp. 3909–3917.

Subsection 1

Automata Learning

Automata

- **discrete** A discrete set of states.
- **continuous** Continuous states.
- **hybrid** Finite discrete states with continuous variables
- **timed** Finite automaton with a set of real-valued clocks. Sub-class of hybrid automata.



Automata Learning

Learn the structure of an automaton by observing state traces (passive) or posing queries (active).
Referenced results:

- Learn formal languages via queries: Dana Angluin. "Queries and Concept Learning". In: *Machine Learning* 2.4 (1987), pp. 319–342
- Limited learning of hybrid automata
- Efficient learning of 1-clock timed automata
- Theory of learning timed automata
- Deterministic real-time automaton learning

Application:

- Driving pattern discovery: Yihuan Zhang et al. "Car-following Behavior Model Learning Using Timed Automata". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 2353–2358

Automata Learning Software

learnlib.de

Table of Contents

1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software

Section 4

Safe Reinforcement Learning

Markov Decision Process

A tuple $(S, A, T, R, p_0, \gamma)$ where

- S is a set of states
- A is a set of actions
- $T : S \times A \to S$ is a probabilistic transition function
- $S \times A \times S \to \mathbb{R}$ is a reward function
- p_0 specifies the initial state distribution
- $\gamma \in [0, 1]$ is a discount factor

Objective: Find a policy $\pi : S \times A \to \mathbb{R}$ that maximizes the cumulative discounted reward

$$\sum_{t=0}^{\infty} \gamma^t R(s_t, r_t, s_{t+1})$$

A Comprehensive Survey of Safe Reinforcement Learning

"Safe Reinforcement Learning can be defined as the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes."

Fundamental approaches:

1. Alter the optimization criteria
2. Modify the exploration process

Javier Garcia and Fernando Fernandez. "A comprehensive survey on safe reinforcement learning". In: *Journal of Machine Learning Research* 16 (2015), pp. 1437–1480.

Safe RL Research

- Safe RL developed based on Lyapunov stability verification
- Felix Berkenkamp et al. "Safe Model-based Reinforcement Learning with Stability Guarantees". In: *NIPS*. 2017, pp. 908–919
- Reachability-base approach
  - Vulnerable to model inaccuracies
- Model environment with Gaussian Processes
- Jeremy H. Gillula and Claire J. Tomlin. "Guaranteed Safe Online Learning via Reachability: tracking a ground target using a quadrotor". In: *ICRA*. IEEE, 2012, pp. 2723–2730

Safe RL Research (continued)

- Validate actions with temporal logic.
  Mohammed Alshiekh et al. "Safe Reinforcement Learning via Shielding". In: *AAAI*. AAAI Press, 2018, pp. 2669–2678
- Probabilistic model checking to verify and repair learned policies.
  Shashank Pathak, Luca Pulina, and Armando Tacchella. "Verification and repair of control policies for safe reinforcement learning". In: *Appl. Intell.* 48.4 (2018), pp. 886–908
- Combination of formal verification and runtime monitoring
  Nathan Fulton and André Platzer. "Safe Reinforcement Learning via Formal Methods: Toward Safe Control Through Proof and Learning". In: *AAAI*. AAAI Press, 2018, pp. 6485–6492
- Training an intervention model from human oversight
  William Saunders et al. "Trial without Error: Towards Safe Reinforcement Learning via Human Intervention". In: *AAMAS*. 2018, pp. 2067–2069

Table of Contents

1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software

Neural Network Verification Approaches

Validating even simple properties about their behaviour is NP-complete

Reachability Estimate the possible set of network outputs over some input distribution.

- Software: AI^2 `safeai.ethz.ch`
- Practical for large networks

Sensitivity Measure maximal output deviation for bounded input deviation.

Falsification Find adversarial examples.

Adversarial Robustness Be resilient against adversarial examples

- An attack-independent robustness metric.
- Conditions under which no adversarial examples exist

Table of Contents

1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software

NN Verification and Testing Software

Verification Tools

Reluplex, Sherlock, AnalyzeNN, AI^2, PLNN, Planet, NeVer, VeriDeep, DeepGo, L_0-TRE, SafeCV, Certified ReLU Robustness, NNAF, Convex Adversarial

Testing Tools

DeepCover, DeepXplore, DeepConcolic

Questions?