Master Thesis Defense

Knowledge Production and Control of a Black Box Using Machine Learning

Student: Christopher W. Blake

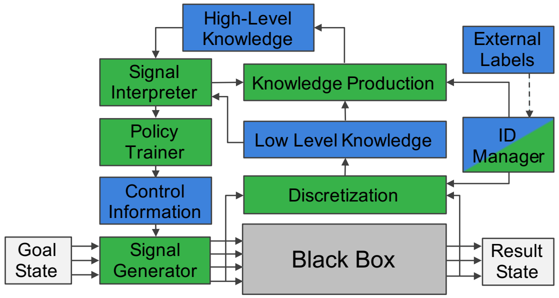
Supervisor: Prof. Eirini Ntoutsi

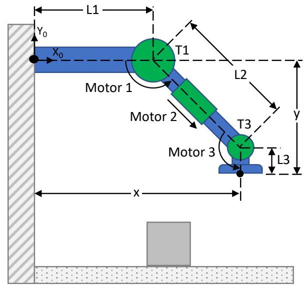
November 2018

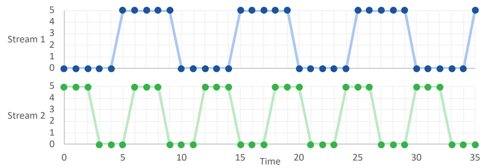
Leibniz Universität Hannover

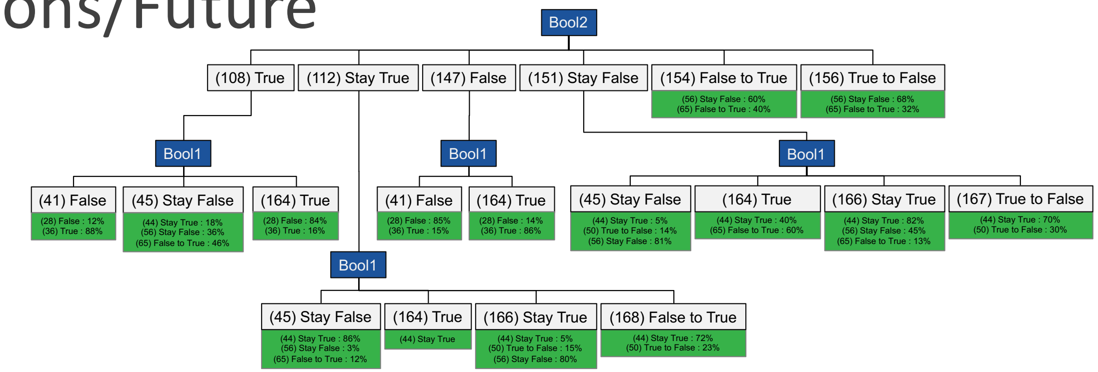
Overview

- Background + Motivation
- Problem Statement
- Related Work
- Mathematical Formulation
- Learning Theory
- Experimental Setup
- Results
- Conclusion
- Recommendations/Future









Background

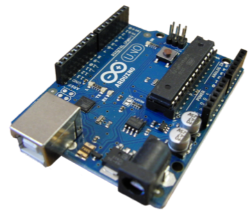
- 7 Years Industry, Product Development
- B.S. Mechanical Engineering
- Ex-Dancer
- Ex-Gymnast

Motivation and Inspiration

- Development Times
- Testing Times
- Learning Dance (as an Adult)
- Observing Children (and Adults)

Problem Statement

- Complex products/systems are regularly developed.
- Requires a costly-to-develop control process.
- Requires extensive analysis and domain knowledge.



Objectives

- ✓ Creation of an automatic or semi-automatic method for development of control systems.
- ✓ Identification of the unique data experienced by a model.
- ✓ Identification of the repeating structures within the unique data.
- ✓ Identification of the primitive functions of the model, providing the primitive control mechanism.

Enables

- ✓ Shorter time-to-market
- ✓ Extended analysis
- ✓ More capable products

Related Work

Iterative Learning Control

- Optimal Control
- Adaptive Control
- Robust Control
- Intelligent Control

Mixed Approach

- Neural Networks
- Genetic Algorithms
- Traditional Feedback

*Language Learning

Disadvantages

- < 5 Parameter Optimization
- Require Prior Knowledge
- Require Domain Knowledge

Most Similar Work

Kevin L Moore. Iterative Learning Control. *Iterative Learning Control for Deterministic Systems*, pages 425–488, 1993.

Process

1. Neural network (NN) trained to emulate the plant.
2. Controller trained on emulated plant NN.
3. NN backpropagation provides error for controller training.

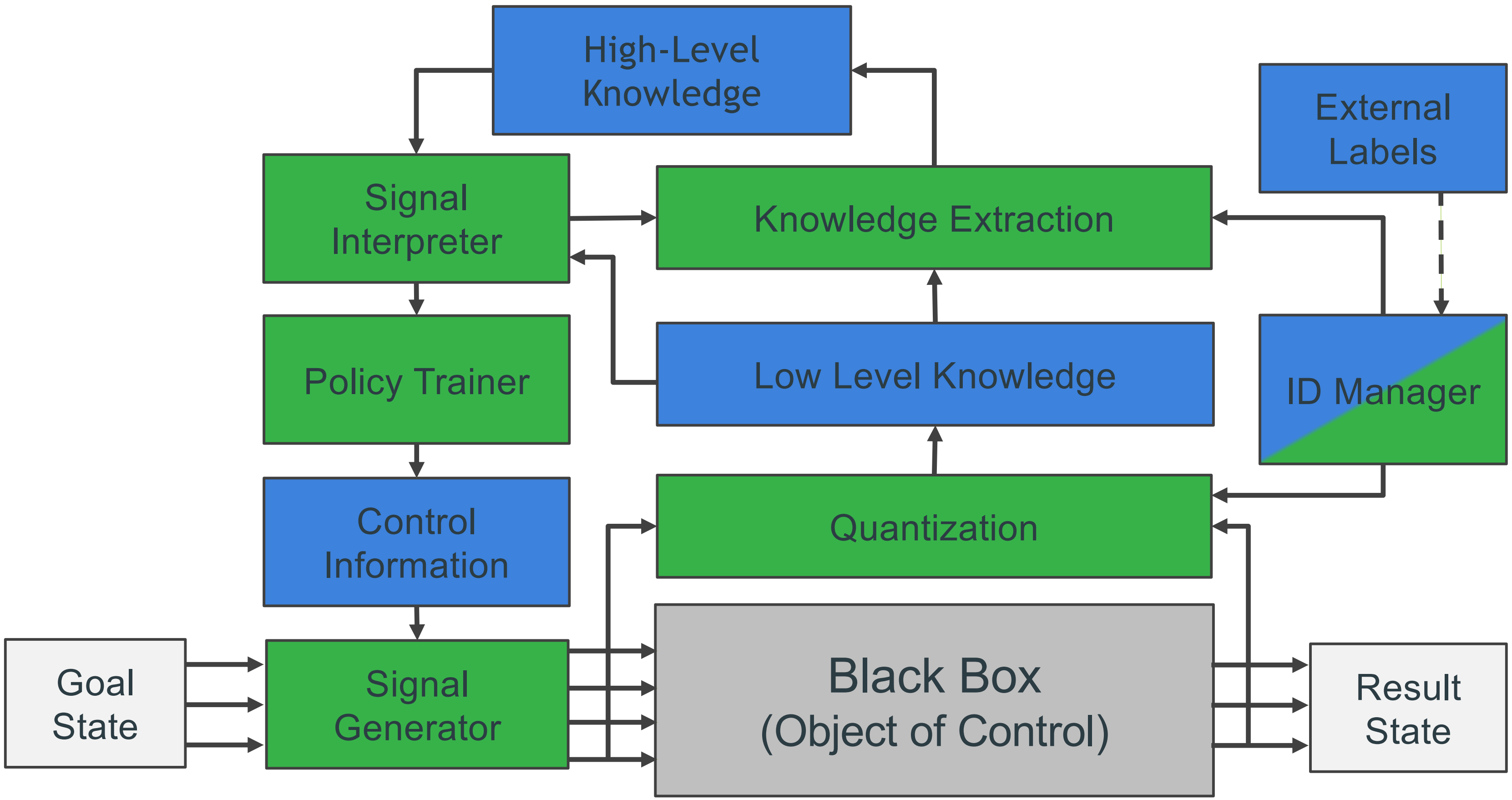
Advantages

- Handles more variables
- Automatic method
- Intelligent process

Disadvantages

- Only deterministic systems
- Prior info of plant
  - Feedback type
  - Degrees of freedom
- Error info not real
- Domain knowledge

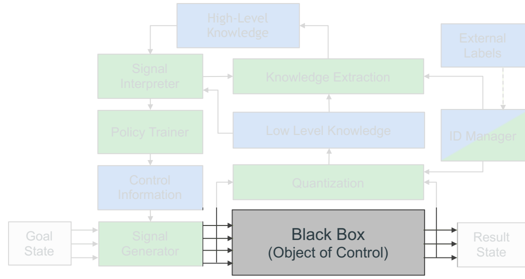
This Work



Advantages

- Easy-To-Read Results
- Less Prior Information
- Stream-Based

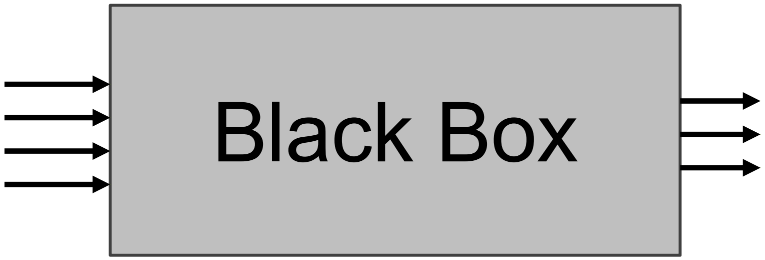
- Non-Deterministic Systems
- Extensible Architecture
- Layered Knowledge



Formulation – Uncertain Model

Input Space

$X \in I^d$



Output Space

$Y \in O^k$

Function Space

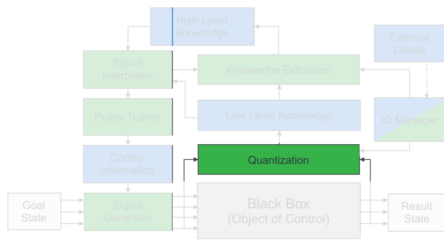
$$F = \{f_1(X), f_2(X), \ldots f_l(X)\}$$

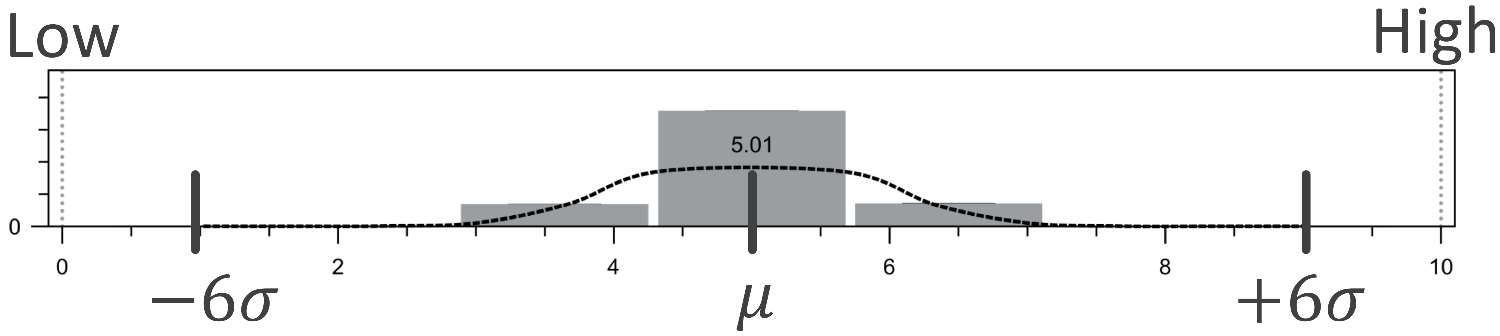
(with internal memory)

Movement Sound Math

$d = number\ of\ inputs$
$k = number\ of\ outputs$
$l = number\ of\ functions$

Formulation – Quantization





Bin Statistics

$$N = \sum 1_t$$

$$Sum = \sum v_t$$

$$SqSum = \sum v_t^2$$

$$\mu = Sum/N$$

$$\sigma = SqSum - 2N\mu + N\sigma^2$$

Action Space

$$S = Split(inner\_value)$$

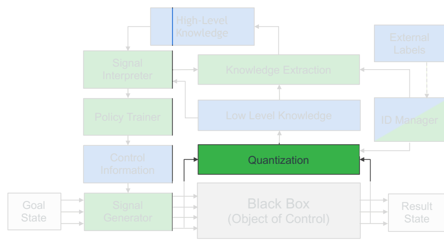
$$M = Merge(neighbor)$$

Nomenclature

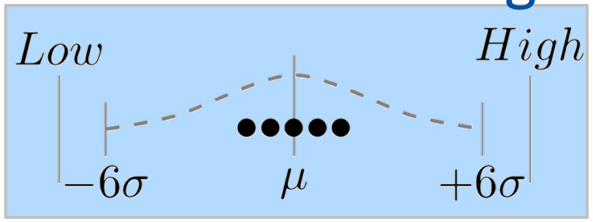
[0.00 (0.97) |5.01| (9.02) 10.00] [47]

Low | -6σ | μ | +6σ | High | Data Count

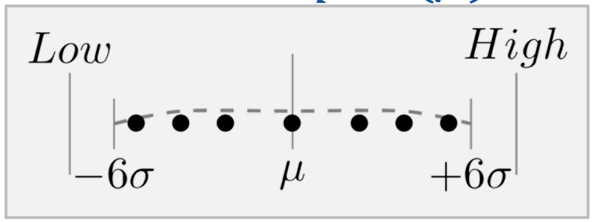
Theory - Quantization



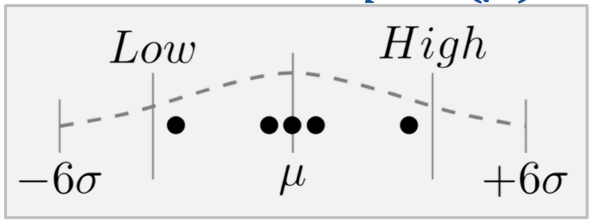
Well-Formed Range



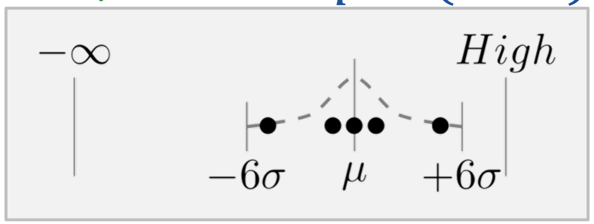
Flat → $Split(\mu)$



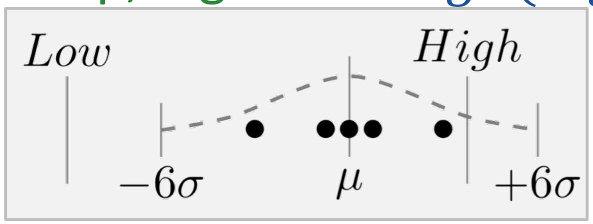
Outside → $Split(\mu)$



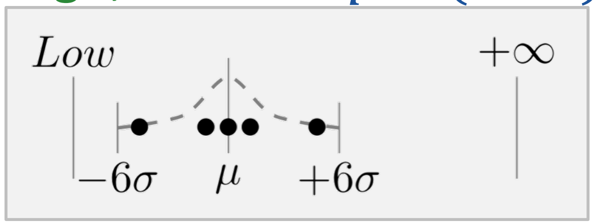
Low, $-\infty$ → $Split(-6\sigma)$



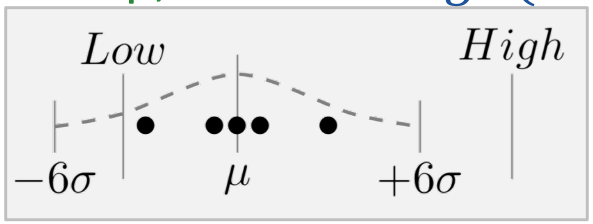
Overlap, high → $Merge(high)$



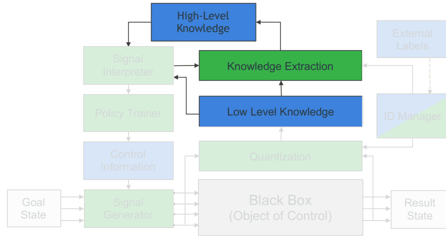
High, $+\infty$ → $Split(+6\sigma)$



Overlap, low → $Merge(\text{low})$



Formulation – Knowledge Model



Knowledge Space

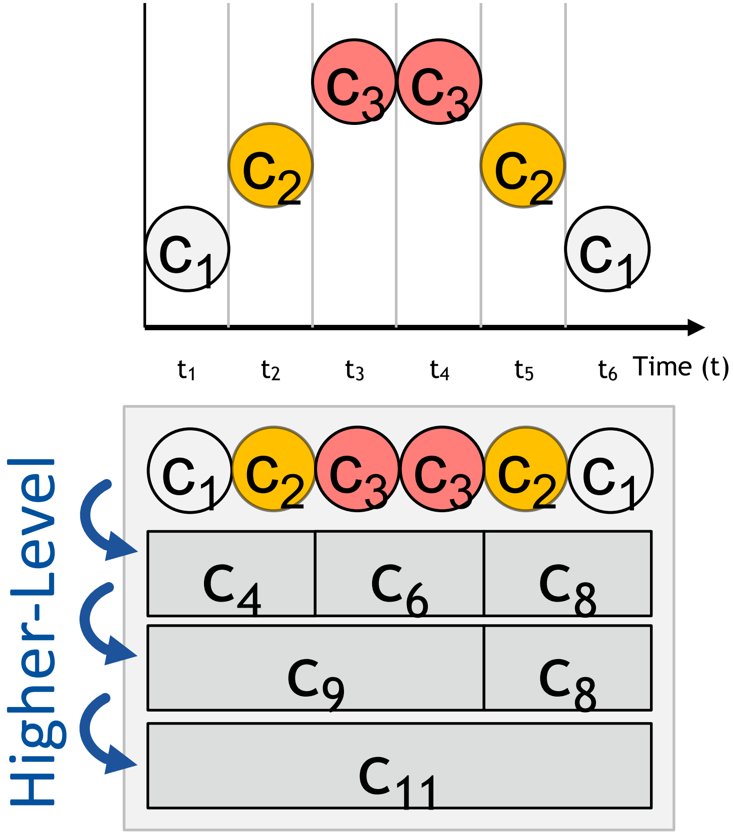
$$c \in C$$

Sequentiality

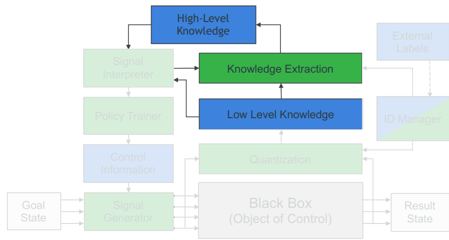
$$c_i = (c_1, c_3, c_4)$$

Simultaneity

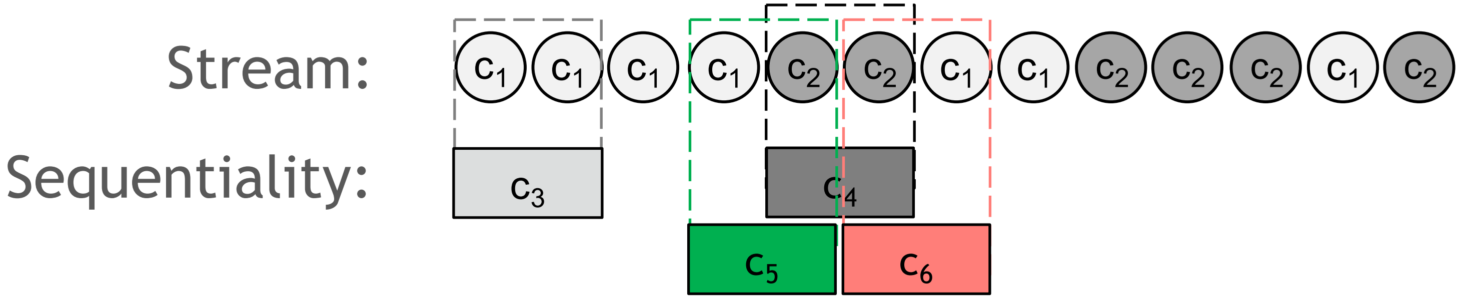
$$c_i = (c_1; c_2)$$

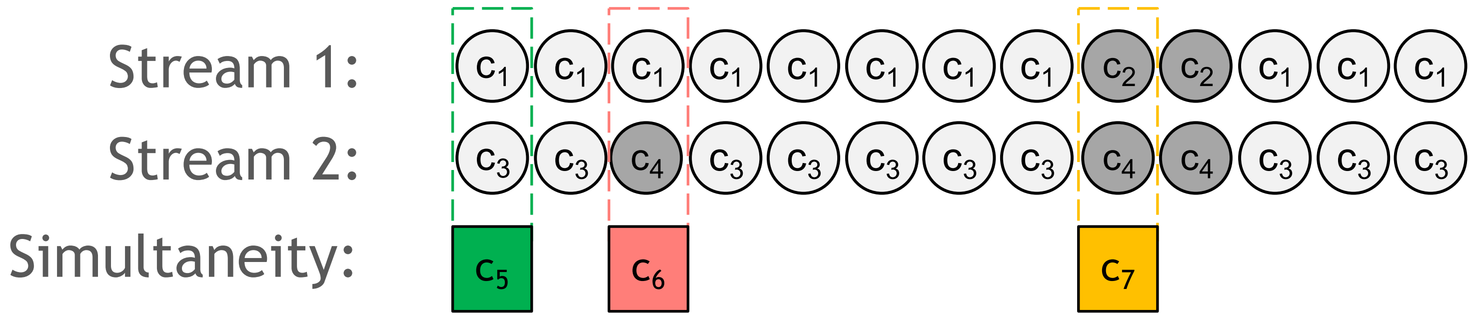


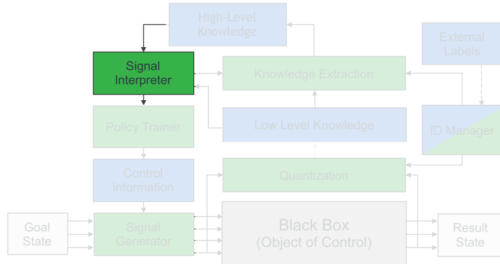
Theory – Knowledge Extraction



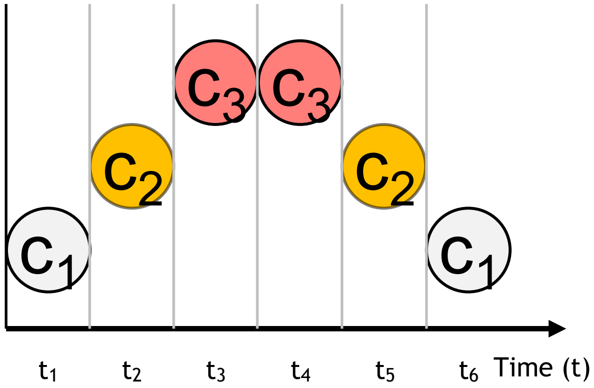
Raw Value:	0.1	-0.1	0.0	0.2	0.3	0.1	-0.1	0.3	5.0	5.1	0.2	0.1	-0.2	4.9	4.8	5.2	5.1	5.0	0.1	0.1	0.0
Quantized:	0	0	0	0	0	0	0	0	5	5	0	0	0	5	5	5	5	5	0	0	0
Interpreted:	c_1	c_1	c_1	c_1	c_1	c_1	c_1	c_1	c_2	c_2	c_1	c_1	c_1	c_2	c_2	c_2	c_2	c_2	c_1	c_1	c_1



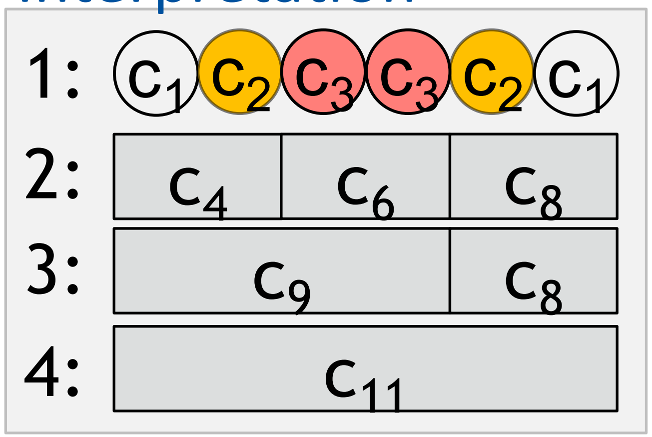




Theory – Interpretation

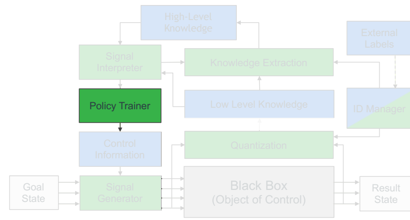


Interpretation



Pass	Interpretation	Generated Knowledge
1	c1; c2; c3; c3; c2; c1	c4 =(c1;c2) c5 =(c2;c3) c6 =(c3;c3) c7 =(c3;c2) c8 =(c2;c1)
2	c4; c6; c8	c9 =(c4;c6) c10 =(c6;c8)
3	c9; c8	c11 = (c9; c8)
4	c11	(done)

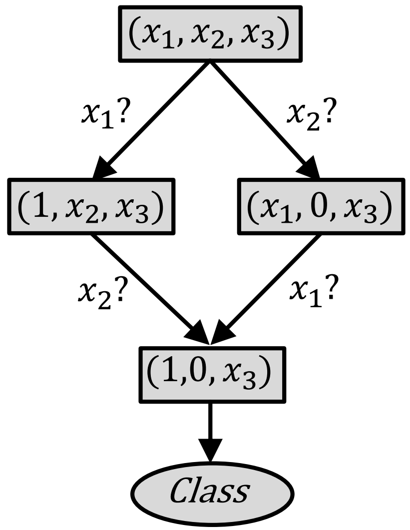
Formulation – RLDT



Feature	Values
Color	white, brown,
Bruise	yes, no
Oder	choc, fruity, none

State $(s \in S)$

Set of feature-value pairs.



Actions $(F, R \in A)$

Query – for another feature's value
Report - pick a classification label

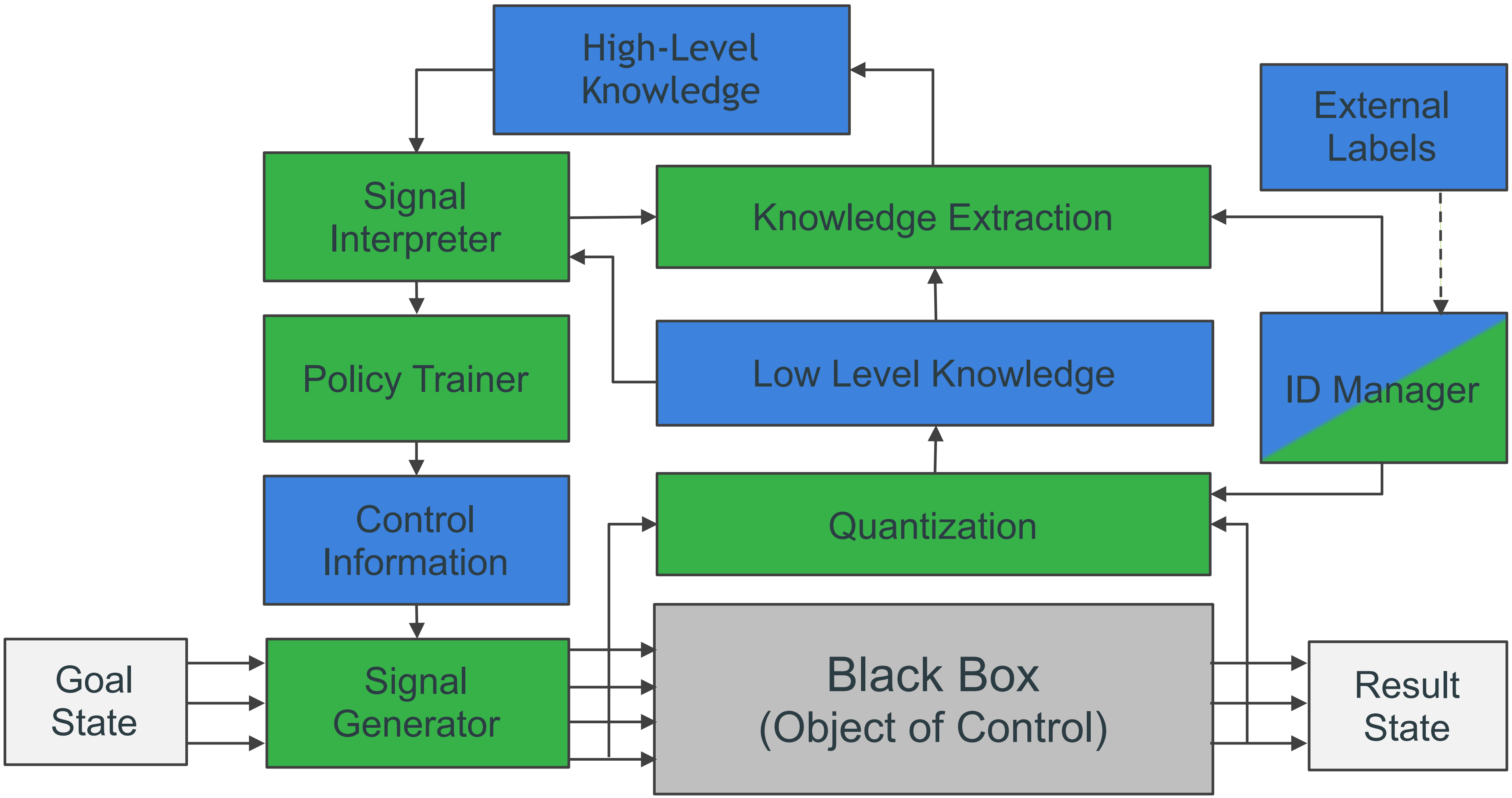
Value Function (Q)

(-) for each query
(+) for each correct classification
(-) for each incorrect classification

Policy (Decision Tree)

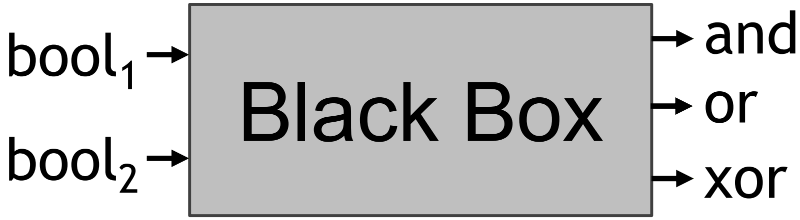
1. Good classification results
2. Fewer queries

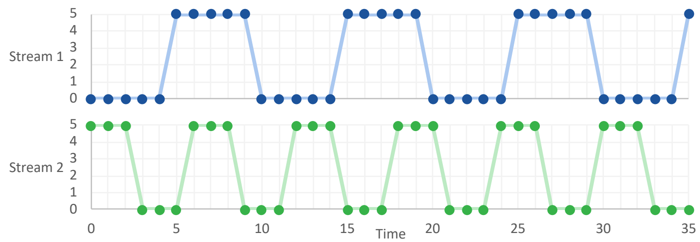
Combined Processes



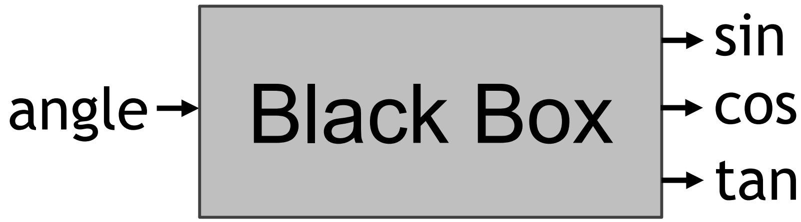
Experimentation

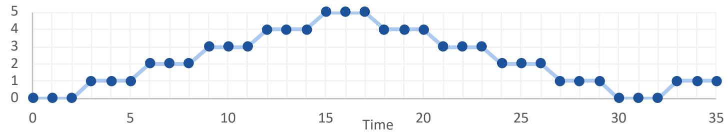
Logic Operators



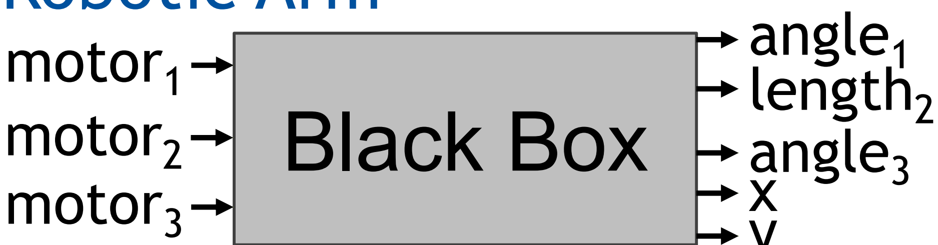


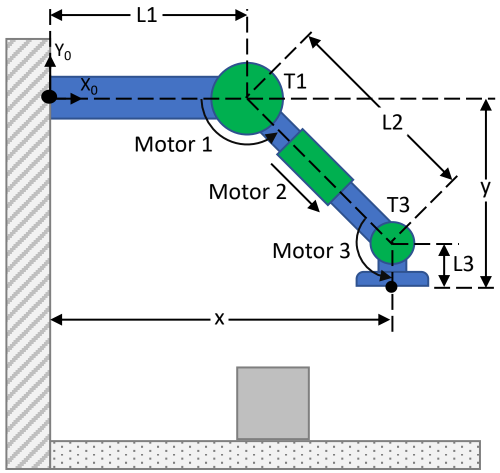
Trigonometric Functions





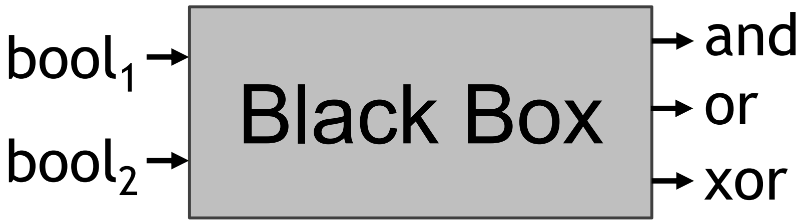
Robotic Arm





Results

Logic Operations

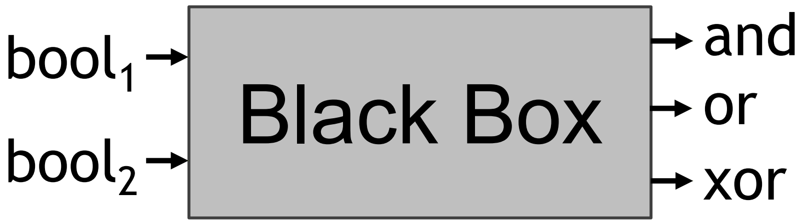


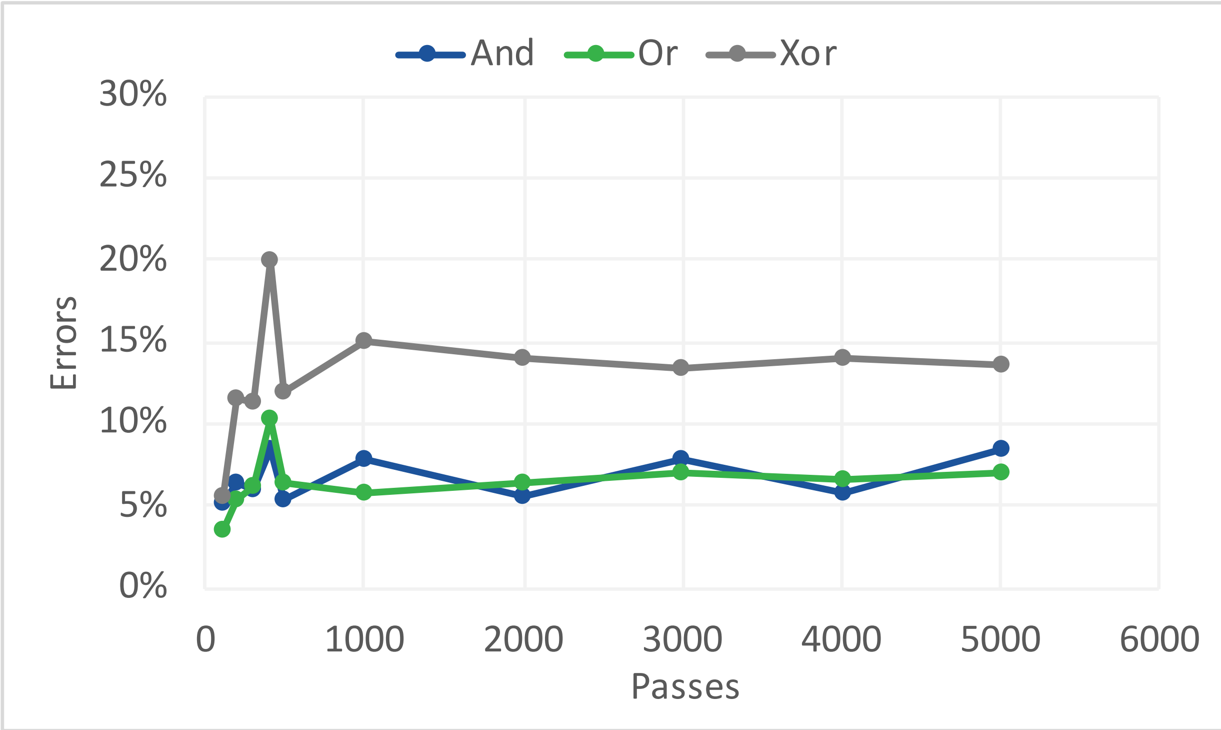
Bool1		
ID	**Name**	**Content**
8		[-∞ |∞| 0.00]
41	False	[0.00 (0.00) |0.00| (0.00) 0.00]
45	Stay False	(41; 41)
164	True	[0.00 (5.00) |5.00| (5.00) 5.00]
165		[5.00 |∞| ∞]
166	Stay True	(164; 164)
167	True to False	(164; 41)
168	False to True	(41; 164)

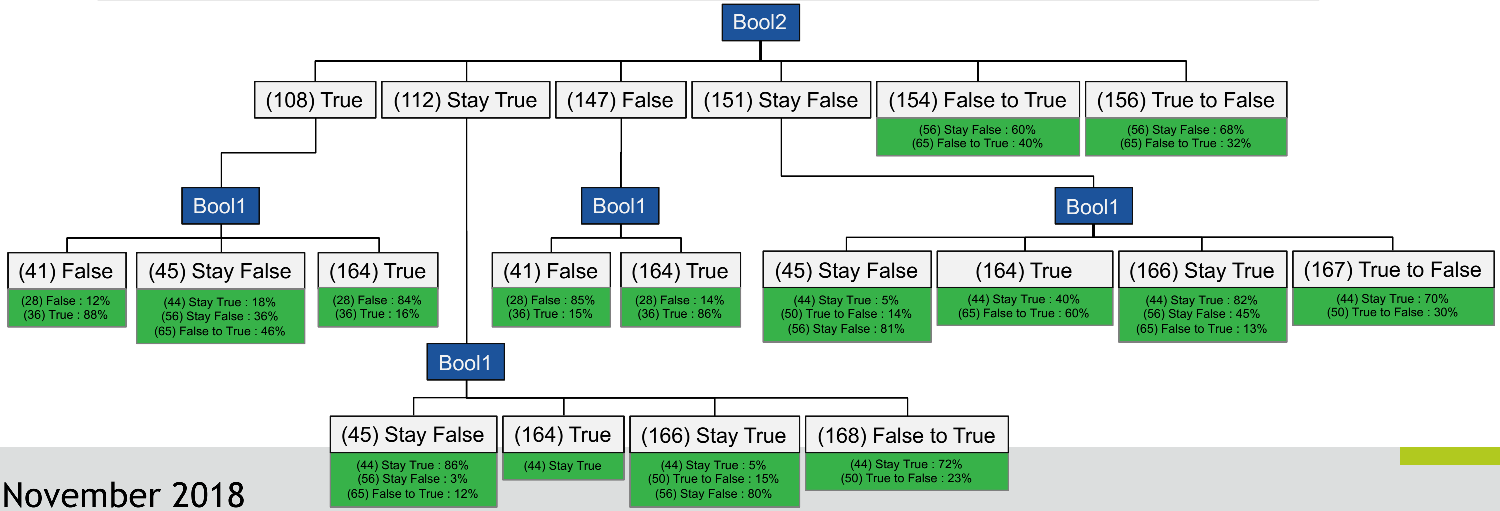
Exclusive Or		
ID	**Name**	**Content**
14		[-∞ |∞| 0.00] (∞)
28	False	[0.00 (0.00) |0.00| (0.00) 0.56]
35		[0.56 |∞| 5.00] (∞)
36	True	[5.00 (5.00) |5.00| (5.00) ∞]
44	Stay True	(36; 36)
50	True to False	(36; 28)
56	Stay False	(28; 28)
65	False to True	(28; 36)

Results

Logic Operations

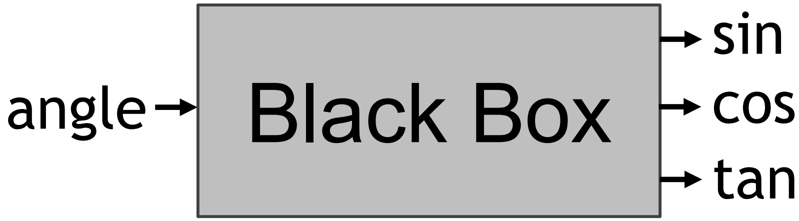


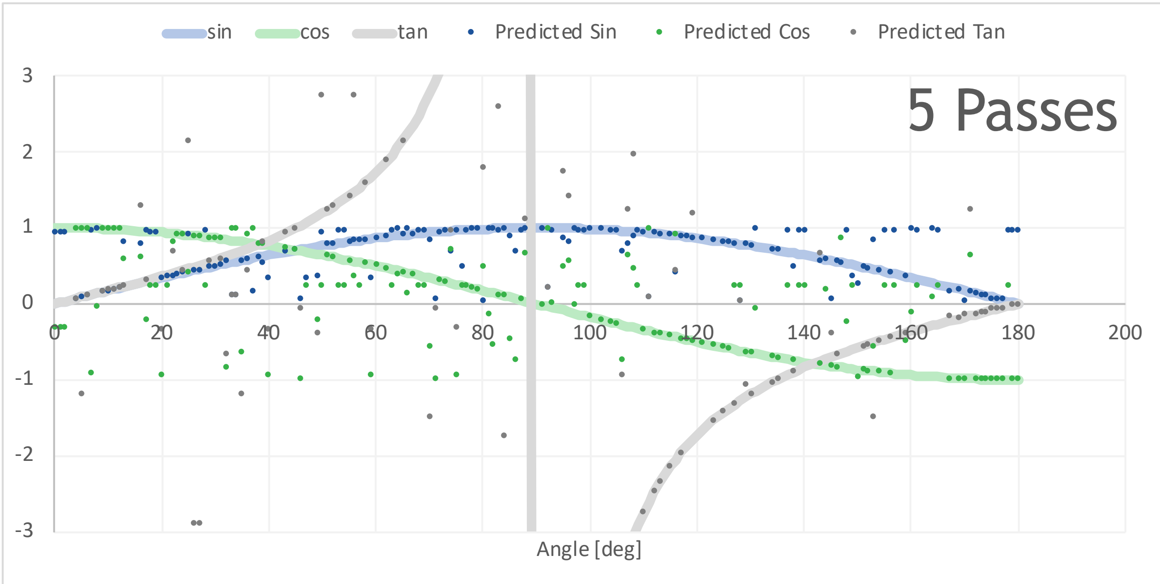


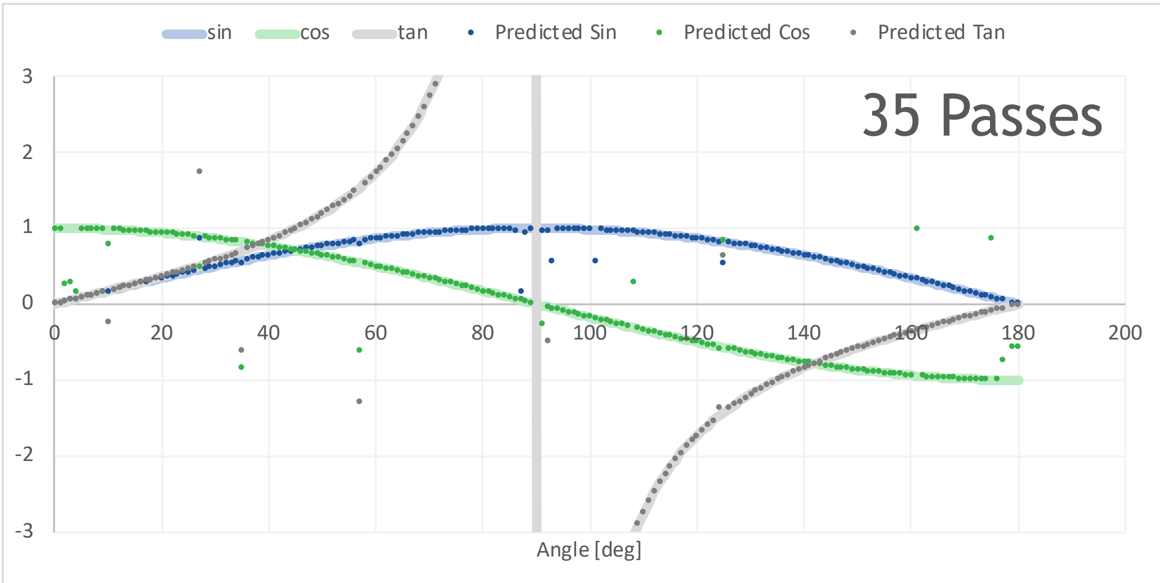


Results

Trig Functions

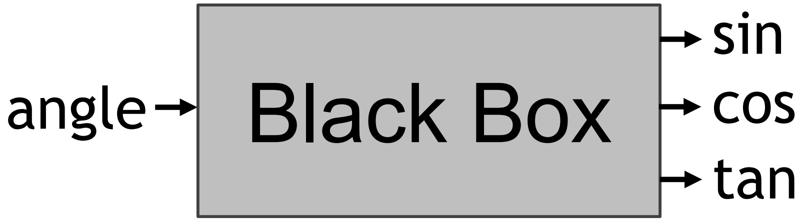


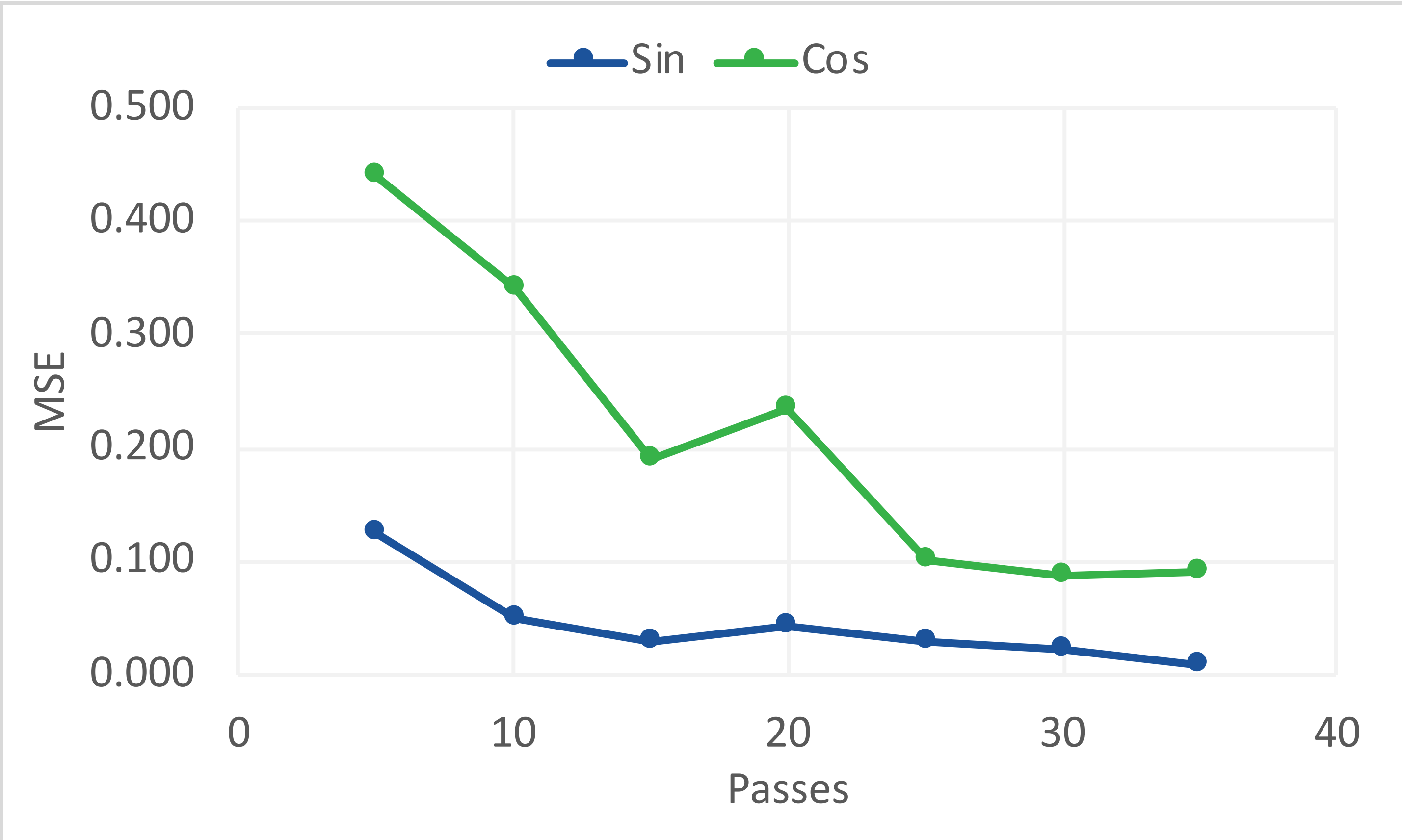




Results

Trig Functions

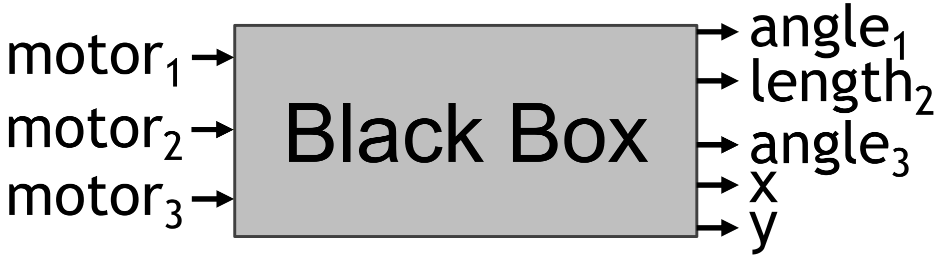




*Tan not shown

Results

Robotic Arm

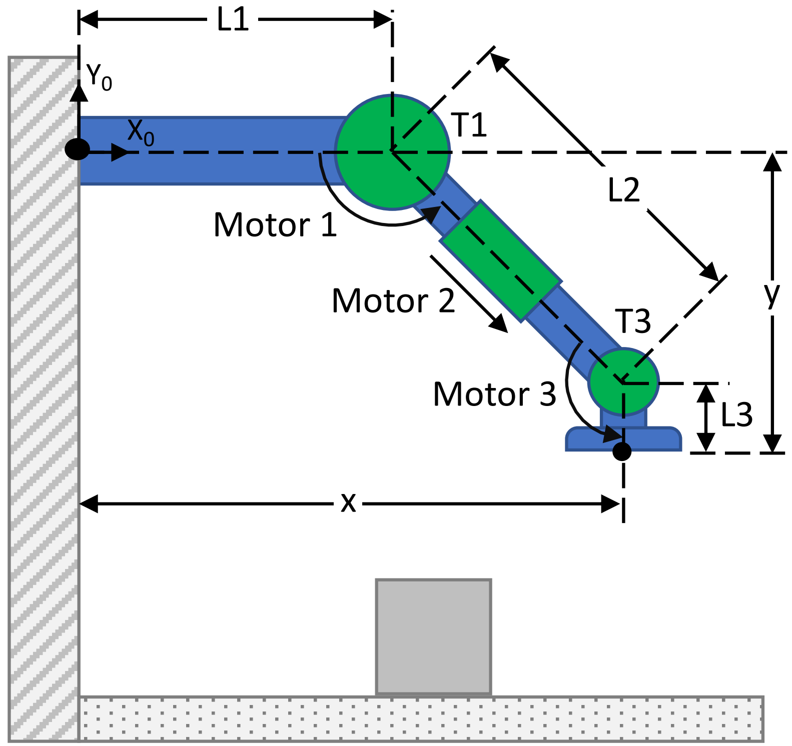


Learned

- Input Vocabulary

Failed

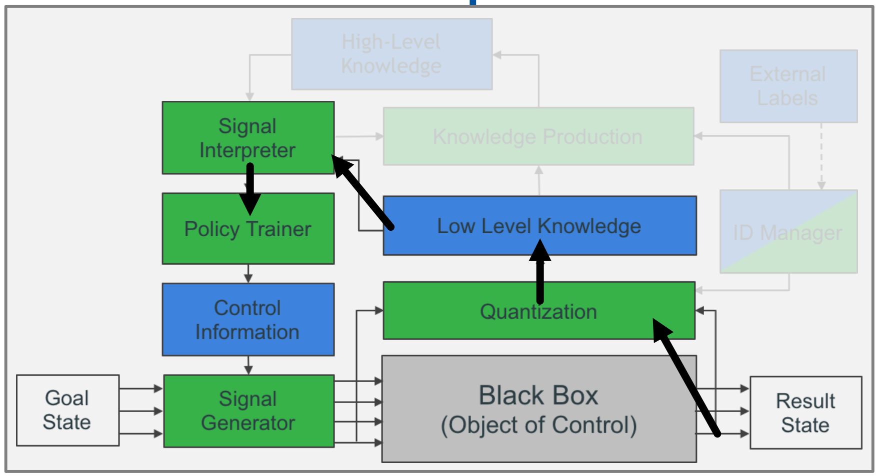
- Output Vocabulary
- Policy Generation



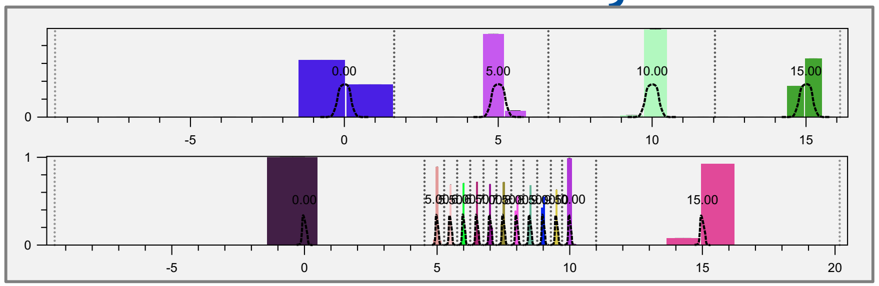
Expected Result

Future

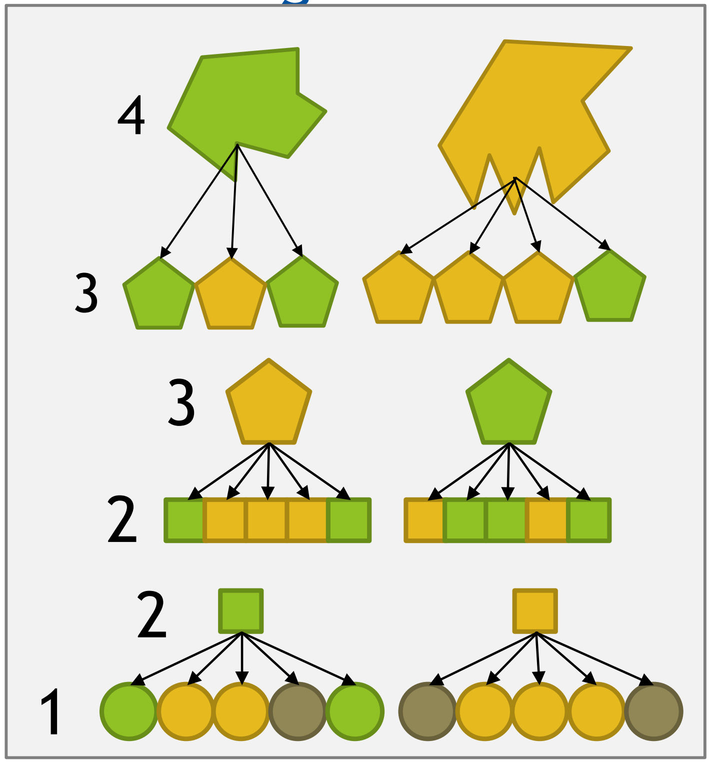
Closed-Loop Control



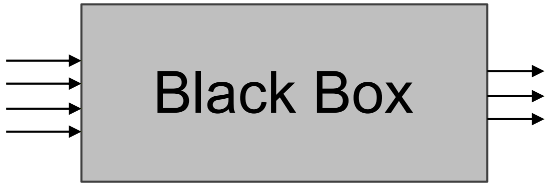
Device Safety



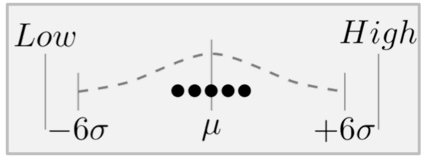
Expanded Knowledge Production



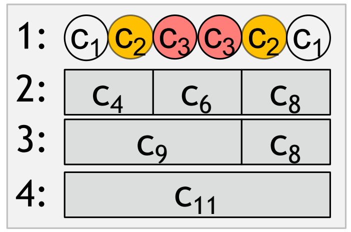
Conclusions (1/2)



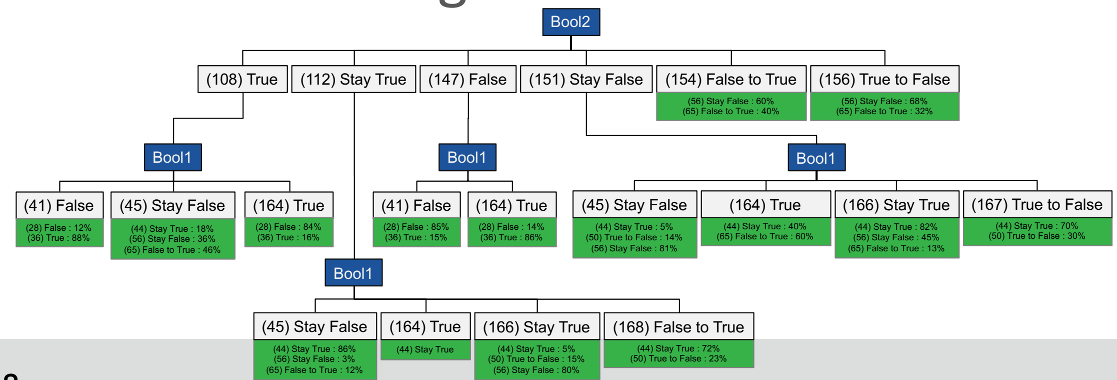
- The **unique values** of the inputs and outputs of a system can be identified and labeled while streaming, assuming a gaussian distribution.



- Complex knowledge can be developed out of the simple concepts of **simultaneity** and **sequentiality**.
- Simple (low-level) knowledge can be combined to form more complex **(higher-level) knowledge**, and be tracked.

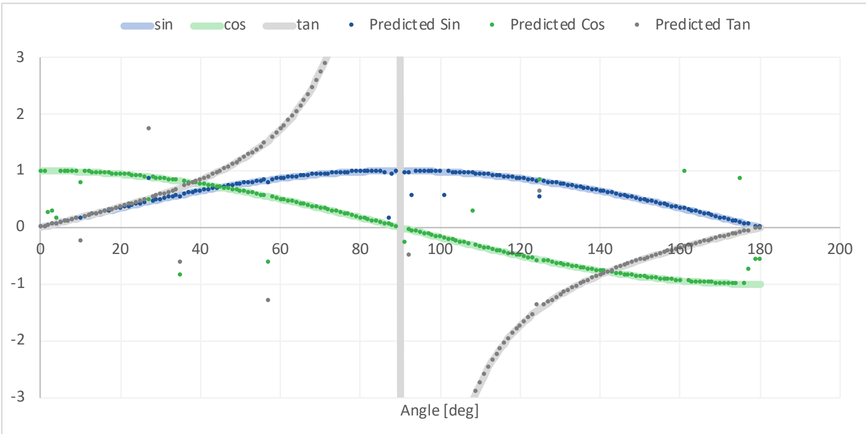


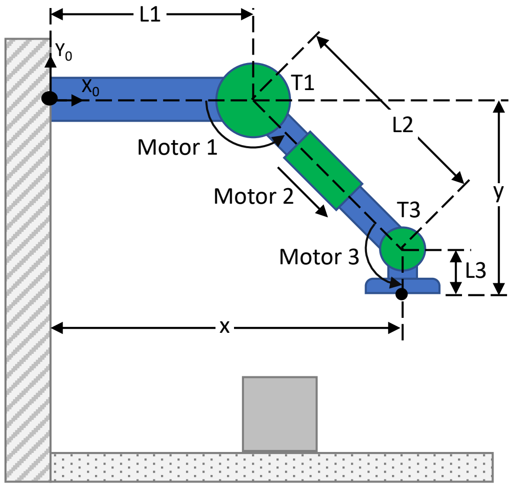
- A **knowledge-based decision tree** can be generated using reinforcement learning.



Conclusions (2/2)

- Testing shows that systems with **binary, categorical** and even **continuous** data can be learned.
- Testing shows that **only deterministic** systems are currently possible.
- **Dynamic systems** are theoretically **possible** with adaption of the learning process, but this is not yet tested.





References

- Abhinav Garlapati, Aditi Raghunathan, Vaishnavh Nagarajan, and Balaraman Ravindran. A Reinforcement Learning Approach to Online Learning of Decision Trees. Technical report, Department of Computer Science, Indian Institute of Technology, Madras, 2015.
- Simon Kirby. Edinburgh Occasional Papers in Linguistics Language evo- lution without natural selection: From vocabulary to syntax in a population of learners. Technical report, 1998.
- Kevin L Moore. Iterative Learning Control. *Iterative Learning Control for Deterministic Systems*, pages 425–488, 1993.
- Derrick H. Nguyen and Bernard Widrow. Neural networks for self-learning control systems, 1991.
- Youqing Wang, Furong Gao, and Francis J Doyle Iii. Survey on itera- tive learning control, repetitive control, and run-to-run control. *Journal of Process Control*, 19:1589–1600.

THANK YOU