



# Evaluation of ALC Toolchain on Neural Network models and Architectures

Hunter Wang

Mentor: Shreyas Ramakrishna  
P.I. Dr. Gabor Karsai



# Background – ALC Toolchain

<< Modeling >>  
1. Modeling

<< Construction >>  
2. Construction

<< V&V&A >>  
3. V&V&A

<< Workflows >>  
4. Workflows

<< DataSets >>  
5. DataSets

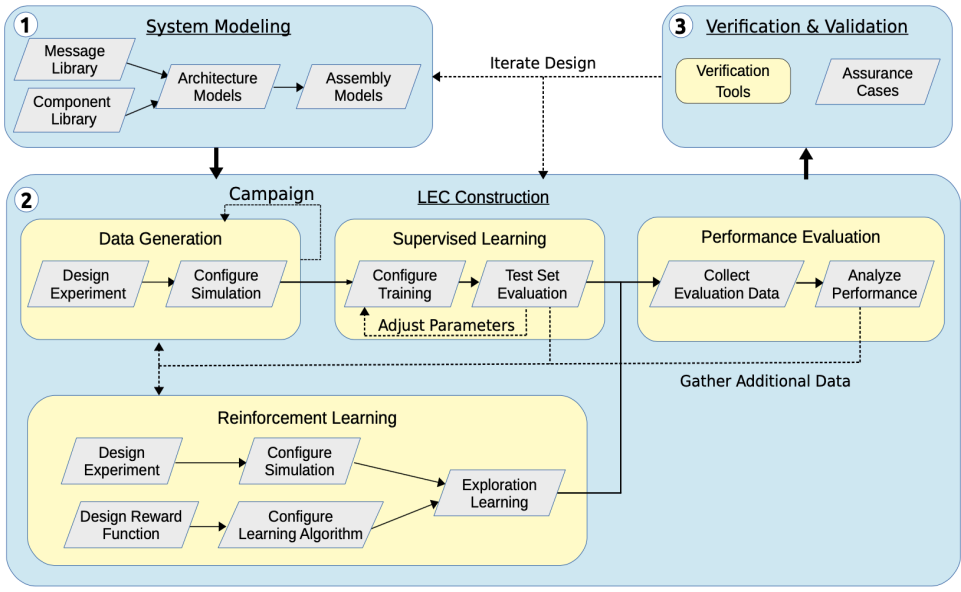
<< MessageLibrary >>  
MessageLibrary

<< BlockLibrary >>  
BlockLibrary

<< Systems >>  
Systems

<< Assemblys >>  
Assemblys

<< WorldModels >>  
WorldModels

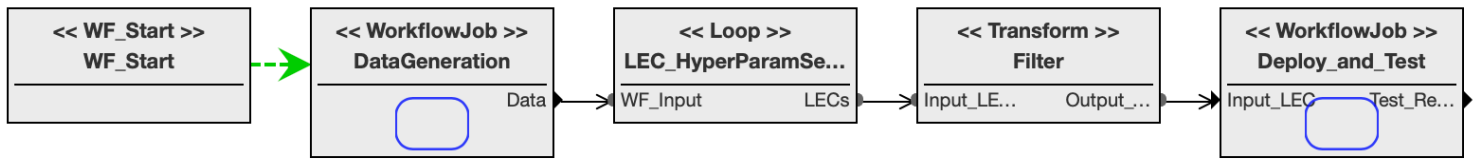


<< Verification >>  
Verification

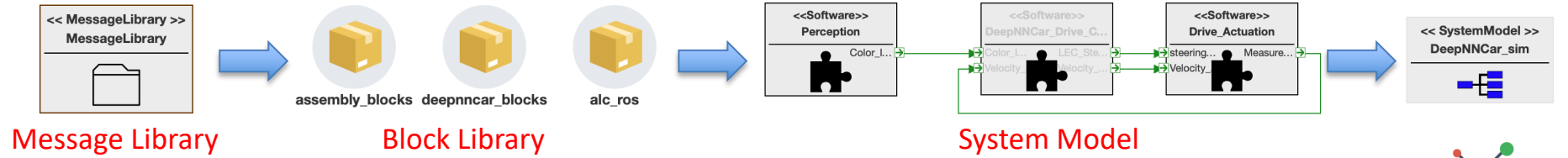
<< Validation >>  
Validation

<< Assurance >>  
Assurance

Source: Charles Hartsell, Nagabhushan Mahadevan, Shreyas Ramakrishna, Abhishek Dubey, Theodore Bapty, Taylor Johnson, Xenofon Koutsoukos, Janos Sztipanovits, and Gabor Karsai. 2019. CPS Design with Learning-Enabled Components: A Case Study. In *Proceedings of the 30th International Workshop on Rapid System Prototyping (RSP'19)* (RSP '19). ACM, New York, NY, USA, 57-63.



# ALC Toolchain for Autonomous Driving



**Dataset Collection**

**LEC Training**

**LEC Testing**

**LEC construction elements**

**Construction**

This includes the activities that have been defined in this project for data collection, training and testing of LECs.

The activities are grouped under `Data Collection`, `Training` and `Testing`.

Deploys ROS packages with ROS nodes onto the server running the ALC toolchain

DeploymentModel

Assembly Model

**Training Data Selection**

**LEC model design**

**Analysis of trained model**

**Model evaluation and loss plotting**

Model ID	LEC	Model	Loss	Time
Model-1	LEC	Model	0.0112003	13:30:50.00
Model-2	LEC	Model	0.0012003	16:30:50.00
Model-3	LEC	Model	0.0012003	16:31:50.00
Model-4	LEC	Model	0.0012003	16:32:50.00
Model-5	LEC	Model	0.0012003	16:33:50.00
Model-6	LEC	Model	0.0012003	16:34:50.00
Model-7	LEC	Model	0.0012003	16:35:50.00
Model-8	LEC	Model	0.0012003	16:36:50.00
Model-9	LEC	Model	0.0012003	16:37:50.00
Model-10	LEC	Model	0.0012003	16:38:50.00

Jupyter notebook with training loss plots and postprocess script for the trained LEC analysis

Reusability - The jupyter notebook is always stored in a date checkpointed folder

**ALC Toolchain**

System modeling → ROS package generation → LEC construction & training → Deploying toolchain generated files → Continual learning

Output

- Top level ROS launch file.
- Component ROS package with ROS nodes and launch files
- Trained LEC weights.

1. System modeling, ROS node generation, LEC construction and training for DeepNNCar

Deploy

Run

end-to-end image to steering control

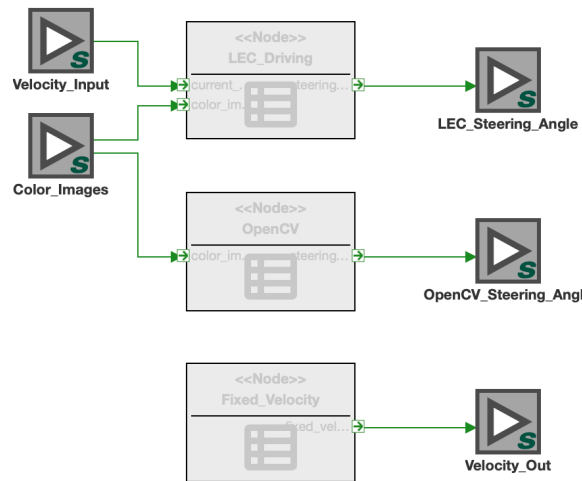
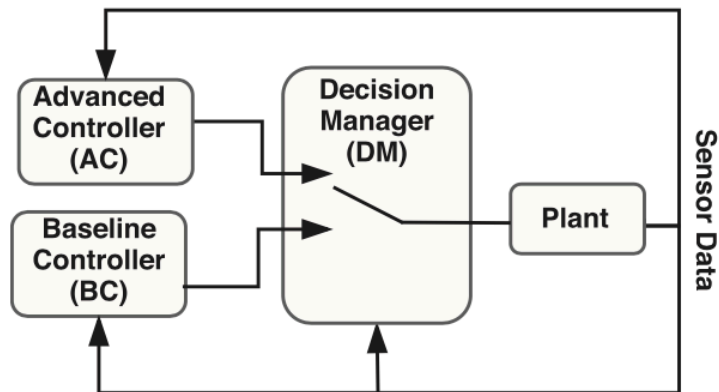
DeepNNCar

Components of the car:
 

- Raspberry Pi3 (RPi3)
- 8.4V NiMH Battery
- LIDAR
- USB Webcam
- Electronic Speed Control
- IR Optocoupler
- Titan 12T Motor
- RPi3 Power Supply

# Simplex Strategy Implementation on ALC Toolchain

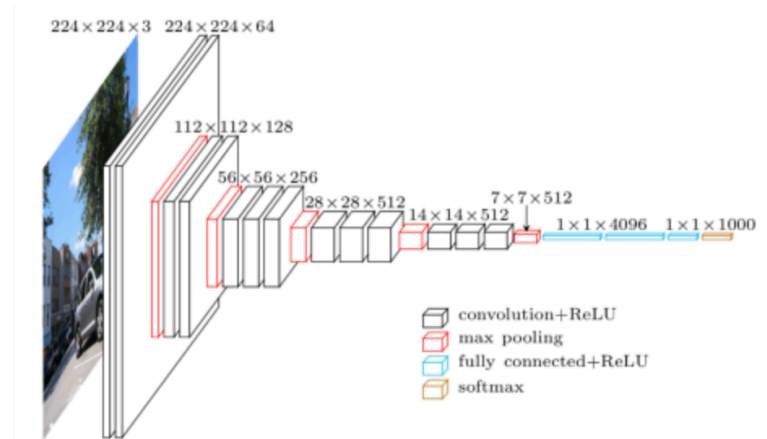
- Goal: Evaluate the the strength of the ALC Toolchain when applied for modeling an autonomous car setting
- Fixed-weight simplex strategy to compute the steering angle
- Two controllers: LEC\_Driving and OpenCV



Source: Shreyas Ramakrishna, Charles Harstell, Matthew P. Burruss, Gabor Karsai, Abhishek Dubey. Dynamic-weighted simplex strategy for learning enabled cyber physical systems. Journal of Systems Architecture, Volume 111, 2020, 101760, ISSN 1383-7621,

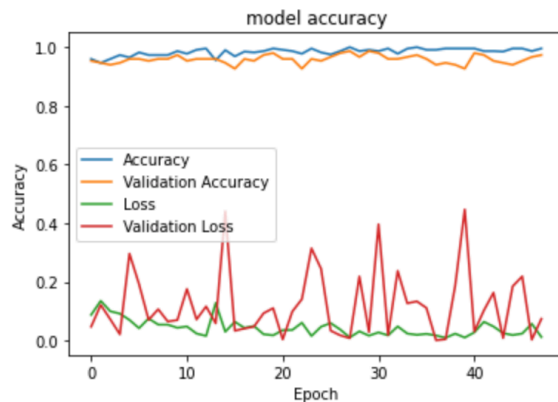
# VGG16 Model Implementation on Google Colab

- VGG16 Model
- 2 classes: vehicles and non-vehicles.
- The model can be used to decide when to stop
- Implemented with Keras.
- Dataset: 3425 images of vehicles and 3900 images of non-vehicles.
- Trained and tested on Google Colab.



Architecture of VGG16

Source: <https://arxiv.org/abs/1505.06798>



- Validation accuracy: 98%
- Testing accuracy: 96.67%
- Lack of reproducibility
- Lack of traceability

# Summary

---

- Future work:
  - Integrate the Dynamic-weighted Simplex Strategy into the Toolchain
  - Improve the training and testing experience within the Toolchain
- What I learned:
  - ALC Toolchain allows for collaboration between multiple users while maintaining reproducibility and traceability during all stages of the development cycle of a CPS system
  - An automated modeling environment can save a lot of time
- Challenges:
  - Work and communicate remotely