CSC 580 – AI2 – HW6

ProVE-Lab for Ensuring Safety with Cyber-Physical Systems

Name: Raju Meesala, **ID:** 2119844

Introduction

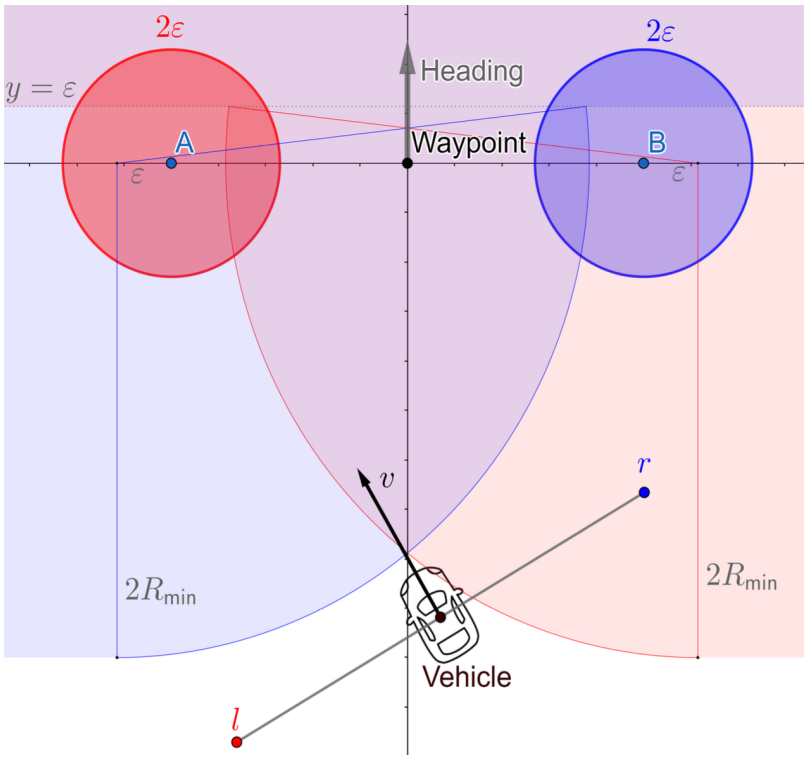
The professor Stefan Mitsch's research focused on using machine learning, mainly reinforcement learning to solve tasks in cyber physical systems like autonomous driving car, aircraft. These systems use learned controllers that make decisions based on a lot of training that they go through. Since all these systems have potential safety implications at runtime, it is important to have strong guarantees about the correctness of those controlled systems.

CPS

CPS is any computer system, or the computer algorithm interacts with the physical world. Examples are, autonomous driving car, aircraft and any other systems where computer systems have interactions with the physical world.

Reinforcement Learning for Autonomous Driving Car

In this example of autonomous driving car, we have control tasks at different levels. Let's discuss how reinforcement learning can solve control tasks at different levels.



Navigation

The car needs to navigate from certain position to a goal and traverse some waypoints along the way. This involves making turns at intersections and not take straight away across the field.

- **High Level Navigation:** Need to be able to pick those waypoints and in between those waypoints.
- **Mid-Level Trajectory Planning:** Need to stick to a route, and we need to stay on pre-planned trajectory and stick to that.
- **Low Level Control:** Steering, braking and acceleration.

Reward Function: The reward function is used to guide the learning process. The car receives rewards for getting closer to the waypoints and sticking to the path. That can be physical road or some virtual road that is being pre-planned. This helps to learn to repeat the actions that led to rewards. This approach helps the car to learn complex driving behaviors that are complex to program manually.

Positive rewards for staying within the safe path and reaching waypoints

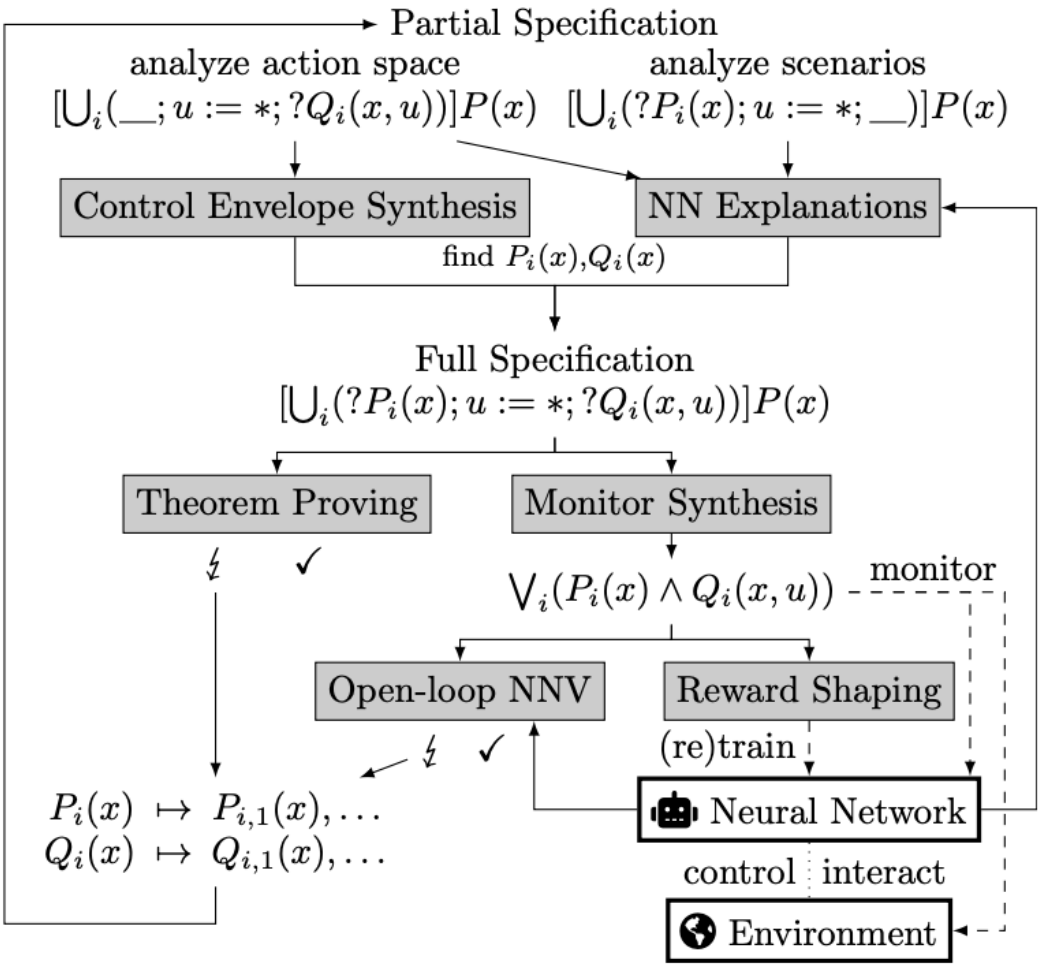
Negative rewards for collisions, deviations from the path, or inefficient movements.

At first the car randomly explores its environment, but it learns to perform better over time through reinforcement learning. The reward function helps to shape the car's behavior by guiding it to follow the path and reach waypoints. The main challenge is that the RL agent to be able to generalize well to real-world scenarios.

Challenges in Defining Rewards

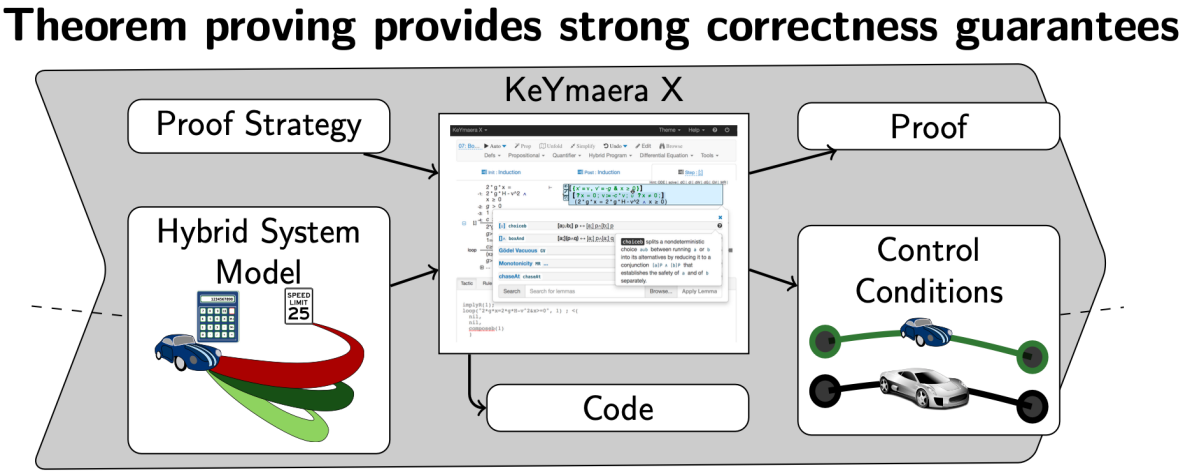
- **Reward Function Complexity:** In autonomous driving there are multiple factors like distance to the next waypoint, current position, and car orientation need to be considered. Ensuring the car is in correct position for the next movement and preventing to get stuck in undesired situations.
- **Real World Environment Uncertainty:** There are always limitations with sensor as we can never fully known about the real-world environment. It is important to consider factors like road conditions and weather conditions while giving control input. Also, the model needs to allow for deviations for learning more about the environment while keeping the vehicle safe.
- **Verification and Theorem Proving:** In testing-based approach, at first they design scenarios, and then they let the controller run and then hey test whether it had the desired outcome. The problem is that the traditional testing-based approach has the downside that it is possible to test only a finite number of examples. So even then testing in simulation is very costly to do that case by case.

Verification Framework Overview (Theorem Proving)



The Verification Framework consists of following key actions:

- **Partial to Full Specification:** This control envelope synthesis and neural network analysis fill in the specification holes. In this partial specification analyzes different control scenarios and full specification defines entire control envelope.
- **Theorem Proving:** Is to ensure the correctness of the control models.
- **Monitor Synthesis:** This is to create real time monitoring systems to find out the deviations. This turns formal model into program and quantifier free input-output conditions
- **Open Loop NNV** analyzes neural network for satisfying formal model
- **Reward Shaping** attempts to train neural network according to formal model.



In the above framework there are three main parts

1. **Hybrid System Model**: Every model is (somewhat) wrong and makes assumptions. These systems have both software control and continuous physical process in any cyber physical systems.
2. **KeYmaera X:** Contains code to verify system safety properties mathematically. This is a theorem prover used to verify and validate the model. Also, this helps to make sure that RL-generated policies are provably safe.
3. **Control Conditions:** This is a safe action space which an autonomous system make decisions by following safety measurements. This system verifies the model assumptions at runtime.

Runtime Monitors for learned Control

As the offline proofs are mostly based on models, it is necessary to check that the actual execution aligning with the assumptions. The system continuously verifies whether the learned controller actions are within the pre validated safety range.

Bridging the Gap Between Models and Real-Time Execution

The gaps can be the parameters like car dynamics, environmental effects like wind, sensor noise and other factors. The traditional simulation-based verification tests a finite number of scenarios, it is not sufficient for real world deployment. The runtime monitors are used to detect the deviations of the model in real world execution.

Formal CPS Models

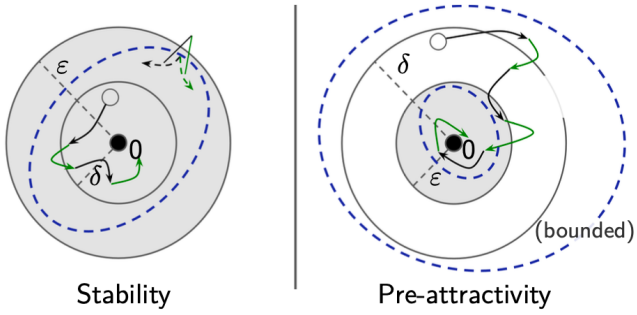
Hybrid Programs (alpha)

The hybrid programs contain a function f(x) which can give a value that helps to decide the action to be taken by the CPS.

Contains: Discrete assignments to represent changes in the variables, Test condition (Q), Differential equation f(x), nondeterministic choice in terms or alpha and beta, sequential composition, and nondeterministic repeat which is alpha.

- In hybrid programs neural network says what not to do.
- Trigonometric functions are used to integrate acceleration, current position, and steering updates to model the vehicle motion.
- Differential logic is used to prove that the CPS will never violates the safety constraints in real world environment.

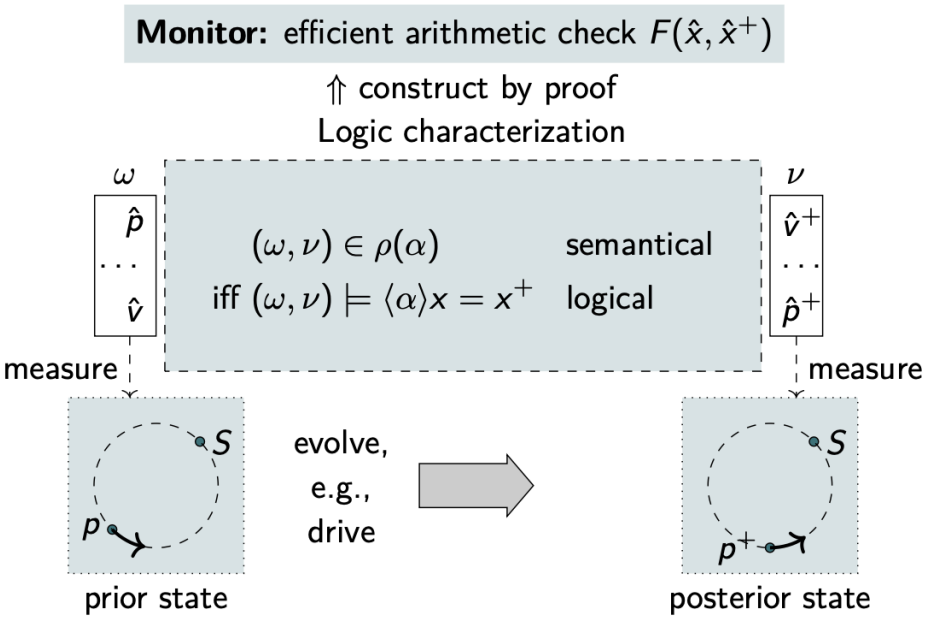
Stability: Robustness to Real-World Disturbances



- Stability is to make sure that the small disturbance to the system does not push the system too far.
- Pre-attractivity guarantees the system will return to stable state even if the disturbance pushes it away temporarily.
- Using differential dynamic logic to ensure the stability in control policies

ModelPlex: Model Validation and Proof Transfer

ModelPlex is a framework for runtime monitoring and validation. It gives mechanism for checking a mathematically verified model is aligning with actual behavior of a CPS.



- Prior state: System starts in an initial state S
- Measure: Measured system's real-world state at runtime
- Monitor: Displays the athematic check
- Posterior State: New state that the system reaches after the execution

Result: This framework helps to ensure the safety proofs remain valid in runtime and plays important role for safety critical applications like autonomous driving car, and aircraft. In any CPS, ModelPlex checks the real world execution matches with verified model. If any discrepancies come in the time of execution, preventive measures can be taken.

Generate Reward Artifacts

1. Safety reward: Distance to boundary of safe region
2. Conflict with achieving goals
3. Prioritize some aspects like speed vs distance

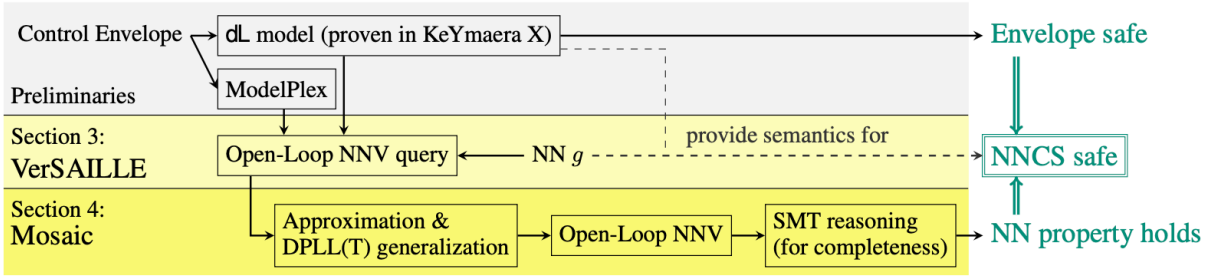
Positive rewards for staying within the safe path and reaching waypoints

Negative rewards for collisions, deviations from the path, or inefficient movements.

Provably Safe Neural Network Controllers

Analyze neural network controller

- Interpret ModelPlex as input-output relation of an unknown controller
- Check neural network controller symbolically for meeting expectations
- Find counter examples



Control Envelope: dL model (proven in KeYmaera X) defines safe region for control decisions, then ModelPlex validates that the system is staying within the safety region in real world by continuously monitoring in runtime.

If the control envelope is safe, it means neural network controller is also safe.

Verification of Safe AI via Logically Linked Envelopes (VerSAILLE): This is a method to translate the neural network behavior into Open-Loops NNV query.

This query helps to check the neural network is following the desired control behavior.

Mosaic Framework:

- Approximation & DPLL(T) generalization: Calculates NN behavior for symbolic reasoning.
- Open-Loop NNV: Checks NN outputs stays within safety region.
- SMT reasoning: Proves the NN meets the safety measurements.

This complete method is to get Neural Network Control System (NNCS), If all the conditions are satisfied, then the NNCS are declared safe.

Result: The System can identify counter examples before deployment, this helps to retrain the models to eliminate the failures.

Technical Analysis:

Further Research Opportunities

As I passionate about autonomous driving technology, this research helps me to understand what kind of preventive measures to be taken while training a CPS. Gave me a detailed overview about the frameworks that are used to make sure how our model performs in real world environment. As current reinforcement learning models are reward given, but they are lack of built in safety guarantees. Research to explore hybrid models where reinforcement learning policies are trained with pre verified safe regions.

In autonomous vehicles, there are other concepts to ensure safety in real world environment. Below are some examples

Cooperative driving: Where vehicles exchange information and make cooperative decisions for safety and overall system efficiency.

Vehicular Ad Hoc Networks: Vehicles within the same area create VANETs and work as nodes for improving safety and navigation.

Sensing Systems: Instead of using traditional sensor devices, using **LIDAR** to collect real time environment data, which is more effective for obstacle detection, and navigation. LIDAR can provide 360 degree visibility and accurate distance measurements

Verification Methods for Deep Learning Models:

Theorem proving works well for symbolic models, but when it comes to deep learning based controllers, it is difficult to verify. Further research can helps to improve neural network verification for safety analysis.

References:

https://www.sciencedirect.com/science/article/pii/S2352146518302606

https://www.youtube.com/watch?v=H2-Yp30TGk4

https://www.youtube.com/watch?v=14fOqMBn9aw

Relevance to AI

This research is closely related to my courses Reinforcement Learning, Neural Networks and Deep Learning, and Machine Learning since it addresses fundamental problems of applying AI-driven control in real-world systems. Using Reinforcement Learning, this research showing how difficult it is to create reward functions that achieve a balance between safety and performance. And demonstrated how formally verified constraints can helps to improve RL-based decision-making in Cyber-Physical Systems (CPS).

From the Neural Networks and Deep Learning, this research taught about verifying black-box neural controllers, a very challenging for deploying the AI in safety-critical applications like self-driving cars and aircrafts. Symbolic reasoning with deep learning models to show the way formal verification can ensure that neural networks will behave predictably in all situations.

In Machine Learning, the work is on the generalization problem in learned controllers, showcasing how ML models can go wrong in edge cases and how runtime monitoring (ModelPlex) helps detect unsafe behaviors at runtime. Overall, this work relates AI-based learning methods and formal verification, establishing concepts from my studies while demonstrating their crucial role in the creation of safe, autonomous systems.

Relevance to AI in general

This research is highly relevant to AI in general as it is about safety challenges in deploying AI systems into real world environment. RL and NN models are extensively being used in domains like healthcare, automobile, financial and others. It is very crucial to validate these models are following the safety constraints. Integrating formal verification methods, symbolic reasoning, and runtime monitoring, this research helps making AI driven control systems provably safe.

Critical Reflections:

Personal Takeaways

1. Approaches to ensure the AI model can follow safety constraints in real world environment.
2. Learning based controllers used to optimize for performance, but they can lead to unsafe behaviors without safety constraints
3. Learned about the importance of verification in AI driven CPS
4. Understood about the different frameworks that are used for validating the model behavior in real world environment

Open Questions:

1. Can we implement pre-defined neural networks like NVDIA's DAVE-2, and PilotNet for autonomous vehicle action control?
2. How can RL reward functions automatically shaped for both optimal performance and safety concerns.
3. Up to what level it is possible to simulate the real-world environment while testing?

Future Directions:

Expanding runtime monitoring frameworks for continuous adaption to real world unexpected conditions.

Implementing latest technologies like LIDAR, VANETs can helps improve safety.

As the main goal is about creating AI models that can learn effectively while staying provably safe, it is important develop better techniques for combining RL with formal verification.

Conclusion

This research focusses on ensuring the safety of using neural network controllers (NNCSs) in Cyber-Physical Systems (CPS), especially in autonomous vehicles. By combining formal verification (differential dynamic logic), runtime monitoring (ModelPlex), and neural network verification (NNV), it effectively bridges the gap between machine learning driven control and provable safety. The study highlights that while AI-based controllers are quite adaptable, they must follow strict safety constraints to avoid real-world failures. With techniques like symbolic analysis, counterexample detection, and theorem proving, this framework makes sure that AI decision-making remains predictable and reliable in all situations. Also, it stresses the importance of runtime monitoring to catch any deviations from expected behavior, enabling real-time corrections and interventions when needed.

Also, this research helps me to understand about the different approaches to ensure safety in AI models correctness.

Final Thoughts: Why this topic matters for researchers/practitioners.

In AI driven CPS, safety and reliability are critical concerns for researchers as well as practitioners particularly in industries like robotics, aircraft, autonomous vehicles, and automation industries. As machine learning model like neural networks and RL agents have become more integrated to real world decision making, it is challenging and also important to ensure their safety. This research mainly focusses on key challenges like verifying and monitoring AI-based controllers to overcome failures in safety critical applications.

Personal Feeling: I am feeling extremely sad for missing an opportunity to be involved in this research, I am personally very passionate about automobiles, I did my under-graduation in Mechanical Engineering from IIIT and worked in Renault Nissan for 3 years out of my passion. Also, my main reason for pursuing a Masters in AI is to work on autonomous driving vehicles. After I came here to DePaul, first thing I started to find out is, if any research is going in autonomous driving. But unfortunately, I was unable to find anyone to guide me, and later I moved on, thinking that there was no research is going on related to it. But now, this is my final quarter.

References

https://neurips.cc/virtual/2024/poster/95085

https://arxiv.org/pdf/2411.14163

https://www.sciencedirect.com/science/article/pii/S2352146518302606

https://www.youtube.com/watch?v=H2-Yp30TGk4

https://www.youtube.com/watch?v=14fOqMBn9aw

https://dl.acm.org/doi/pdf/10.1145/3302504.3311802

Thank you