

CSC 144V: Autonomous Vehicles

Lecture 1.3

HW/SW Platforms and Ethics Exercises Solution

Department of Computer Science,
Hofstra University

Exercise: Compute platforms

For each requirement, choose the best compute platform (**CPU, GPU, FPGA or ASIC/NPU**) and give a one-sentence reason.

#	Requirement	Platform	Reason (one sentence)
1	Batch-1 camera DNN inference with low and deterministic latency; the network may be updated after the vehicle is deployed.		
2	Mass-produced vehicle with a fixed DNN, where power and cost per TOPS matter most.		
3	Highly parallel matrix multiplication for large-batch DNN throughput.		
4	Control-flow-heavy general software: operating system, task scheduling, planning logic.		

Exercise Solution: Compute platforms

#	Requirement	Platform	Reason (one sentence)
1	Batch-1 camera DNN inference with low and deterministic latency; the network may be updated after the vehicle is deployed.	FPGA	Low, deterministic latency at batch 1; reprogrammable after deployment.
2	Mass-produced vehicle with a fixed DNN, where power and cost per TOPS matter most.	ASIC / NPU	Most power- and cost-efficient per TOPS at volume; the price is lost flexibility.
3	Highly parallel matrix multiplication for large-batch DNN throughput.	GPU	Many parallel compute elements suit matrix multiplication (large batches raise throughput but also latency).
4	Control-flow-heavy general software: operating system, task scheduling, planning logic.	CPU	Flexible; strong at sequential control flow, OS and scheduling code.

Exercise: AUTOSAR

Choose **AUTOSAR Classic** or **AUTOSAR Adaptive** for each function and give one reason.

- (i) The steering-actuator control loop on a resource-constrained ECU.
- (ii) A perception service on a high-performance multicore ECU that receives service updates over the network.

Exercise Solution: AUTOSAR

- (i) **Classic:** hard real-time, statically configured, signal-based (OSEK) software for resource-constrained, safety-critical ECUs.
- (ii) **Adaptive:** runs on high-performance multicore ECUs (POSIX), dynamically configured, service-oriented communication (SOME/IP), and can be updated at run time.

	AUTOSAR Classic	AUTOSAR Adaptive
Target ECU	Resource-constrained microcontrollers	High-performance multicore
Operating system	OSEK, hard real-time	POSIX
Configuration	Static	Dynamic
Communication	Signal-based	Service-oriented (SOME/IP)
Updates	Fixed at build time	Updatable at run time

Exercise: Trolley problem

Trolley variant with a **probability threshold**.

- Option A: you kill a pedestrian with probability 1, but it is not your fault.
 - Option B: you kill a different pedestrian with probability p , and it is your fault.
 - You choose A if $p \geq p_{\text{thr}}$, otherwise B.
- (i) Your threshold is $p_{\text{thr}} = 0.3$ and $p = 0.5$. Which option do you choose?
- (ii) According to the lecture, what has actually caused AV accidents in recent years, rather than trolley-problem-style choices?

Exercise Solution: Trolley problem

- (i) $p = 0.5 \geq p_{\text{thr}} = 0.3$, so choose **A** (kill with probability 1, but not at fault). The threshold encodes how much extra weight is given to “my fault” versus certain harm.
- (ii) Failures in **sensors or perception algorithms**; none of the recent AV accidents involved an ethical decision of the trolley type.