Toward Fail-Operational Truck Platooning

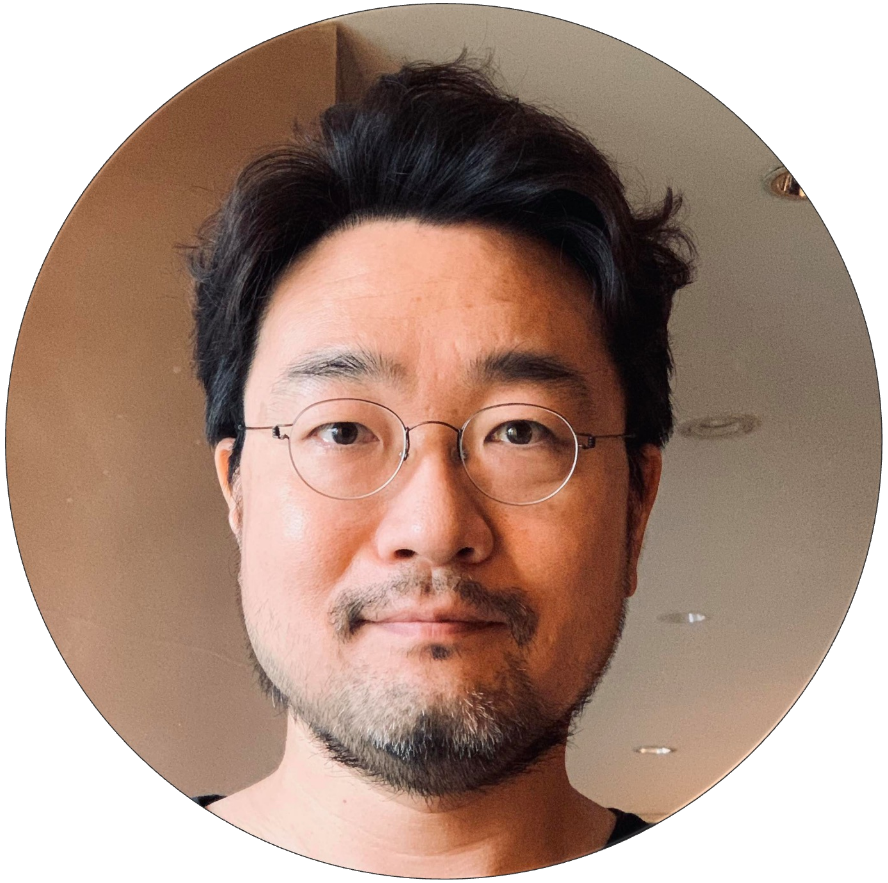
Jong-Chan Kim

Graduate School of Automobile and Mobility

KMU 국민대학교
KOOKMIN UNIVERSITY

소개: 김종찬 교수

- 국민대학교 자동차IT융합학과/자동차공학과/자동차모빌리티대학원
- 국민대학교 미래자동차 사업단 부단장
- 한국정보과학회 컴퓨터시스템소사이어티
- 한국자동차공학회 반도체/시스템SW부문
- 한국자동차공학회 전기전자시스템부문
- https://avees.kookmin.ac.kr



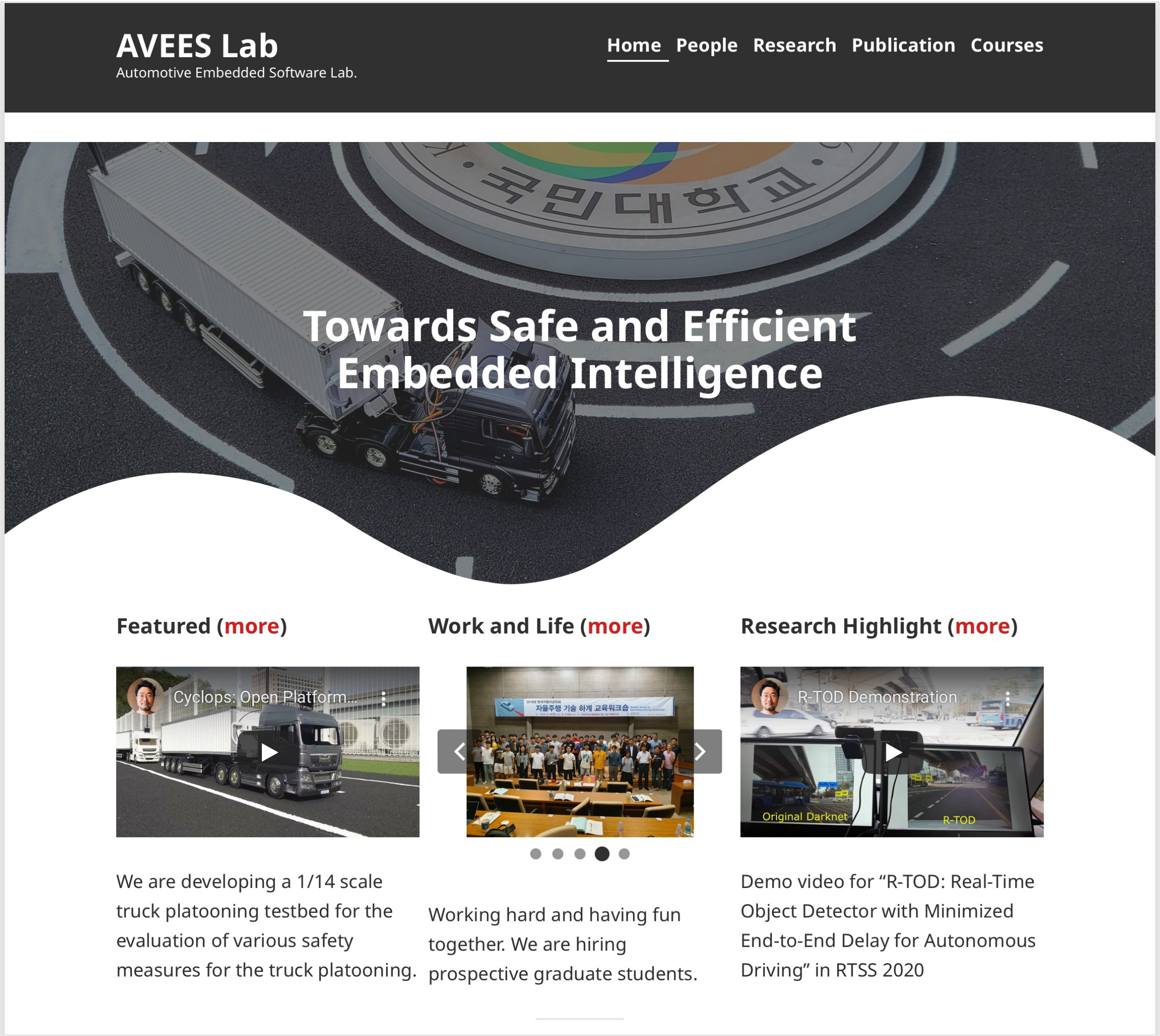
KMU UNIVERSITY SINCE 1946 KOOKMIN 국민대학교

한국정보과학회
KOREAN INSTITUTE OF INFORMATION SCIENTISTS AND ENGINEERS

KSAE

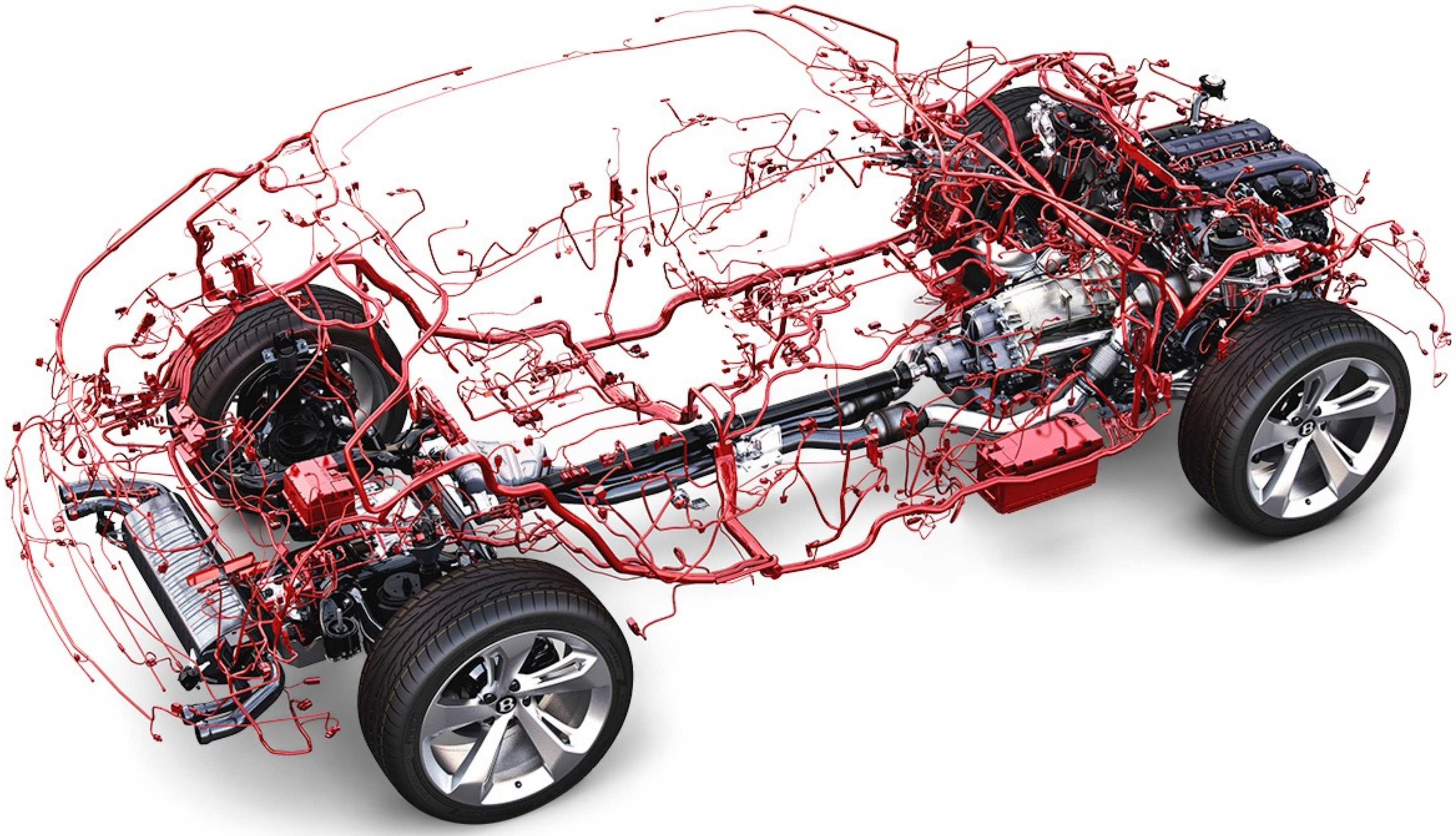
AVEES Lab. (http://avees.kookmin.ac.kr)

- Truck Platooning
  - [ICCAS '20][ICRA '22][DATE '23]
- Real-Time DNN Inference
  - [RTSS '20][RTSS '22] [ICCAS '23]
- AUTOSAR System Optimization
  - [ICCAS '18][Appl. Sci. '20]
- Energy-aware System Adaptation
  - [ISORC '21][TECS '23]
- Autonomous Driving
  - [TECS '21] [Appl. Sci. '22]

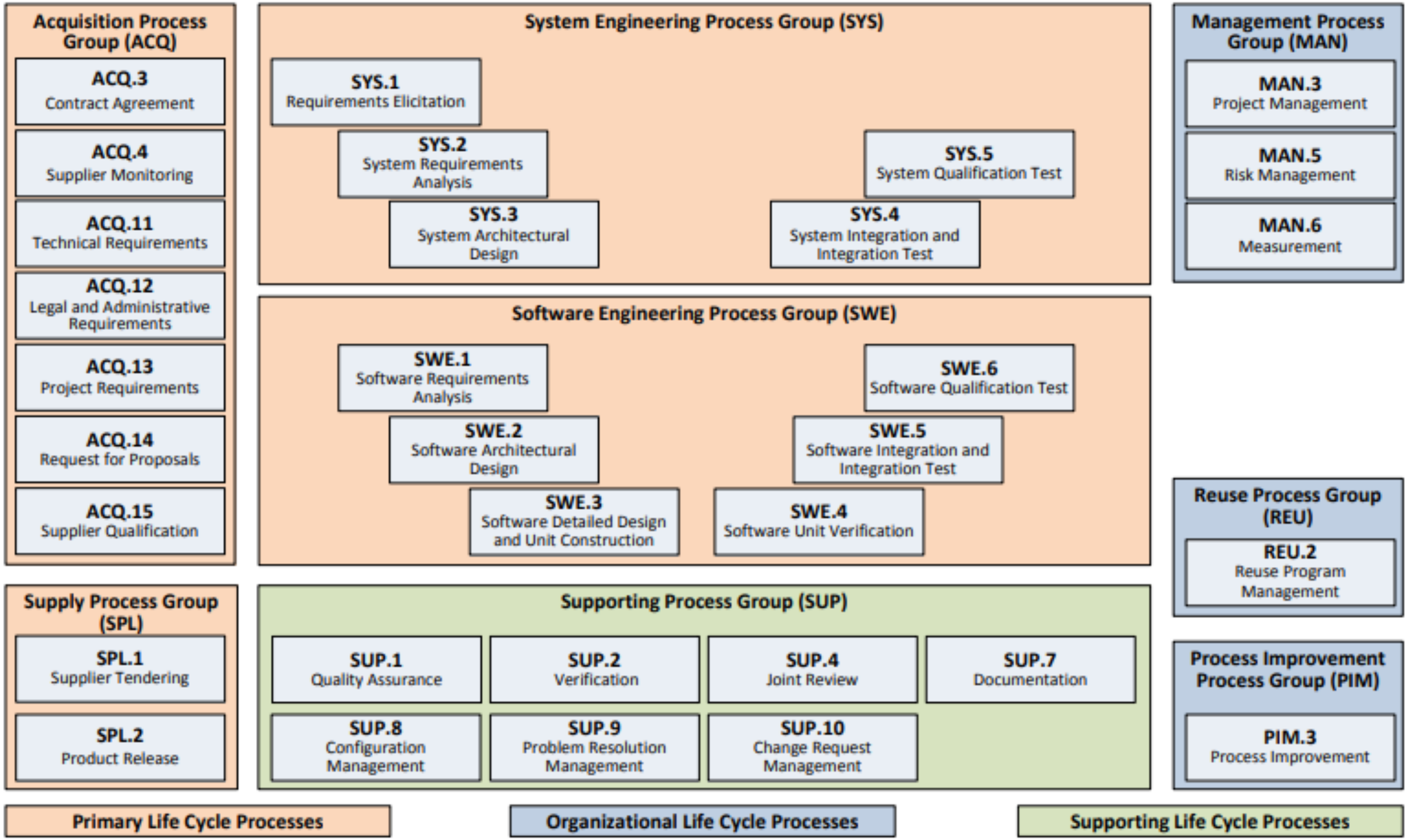


Software Engineering Standards in Automotive

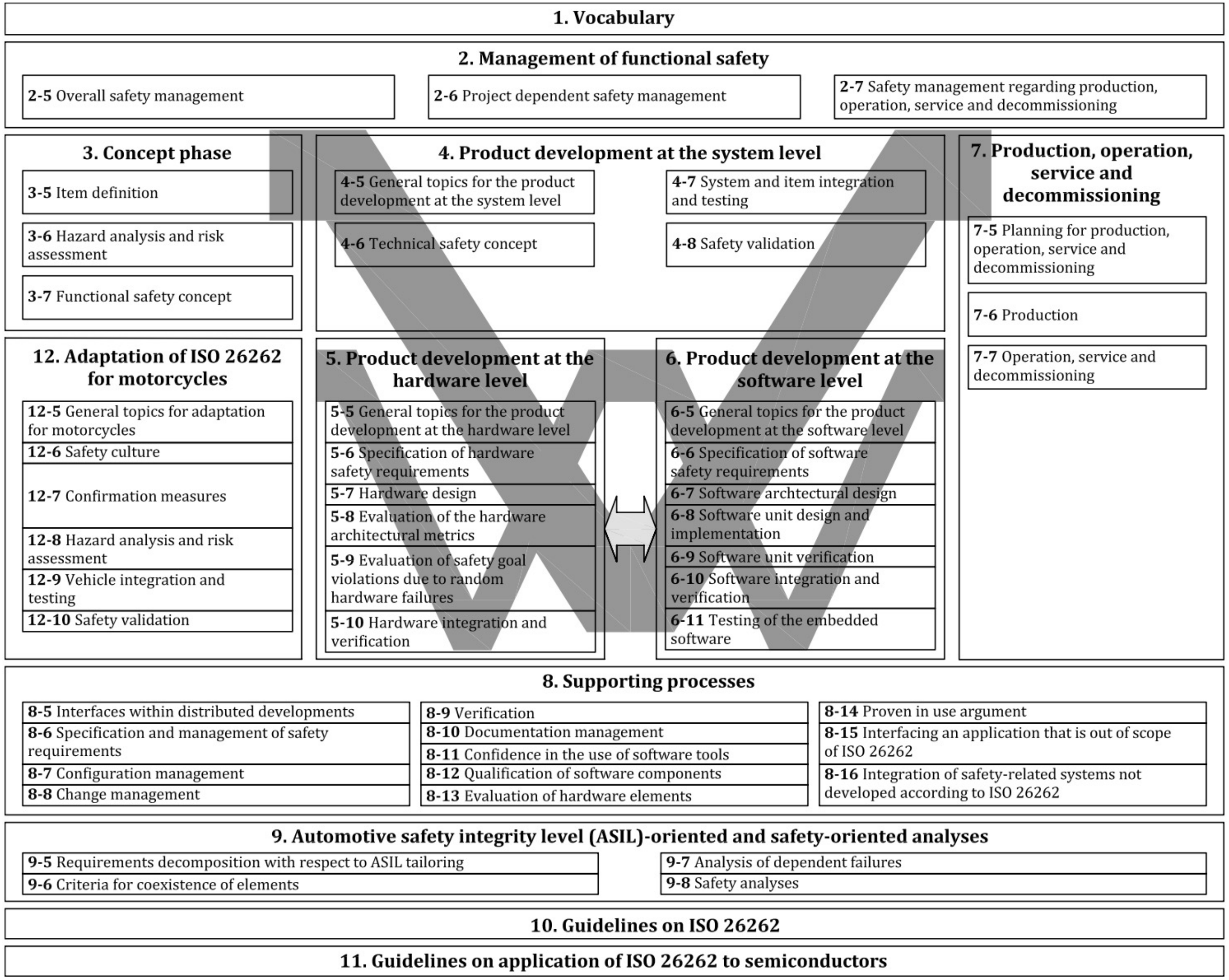
- A-SPICE
- ISO 26262
- ISO 21438
- ISO/SAE 21434
- MISRA-C
- AUTOSAR
- …



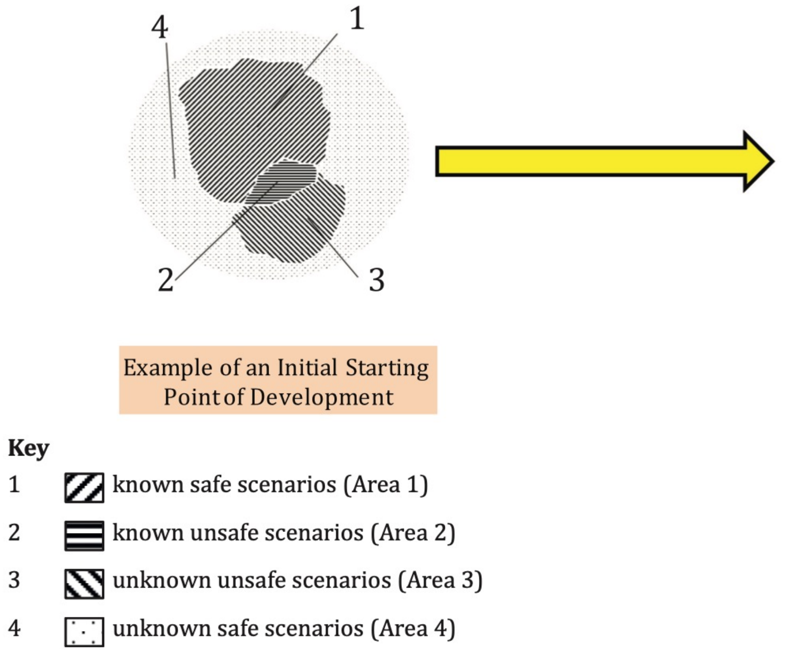
Automotive SPICE

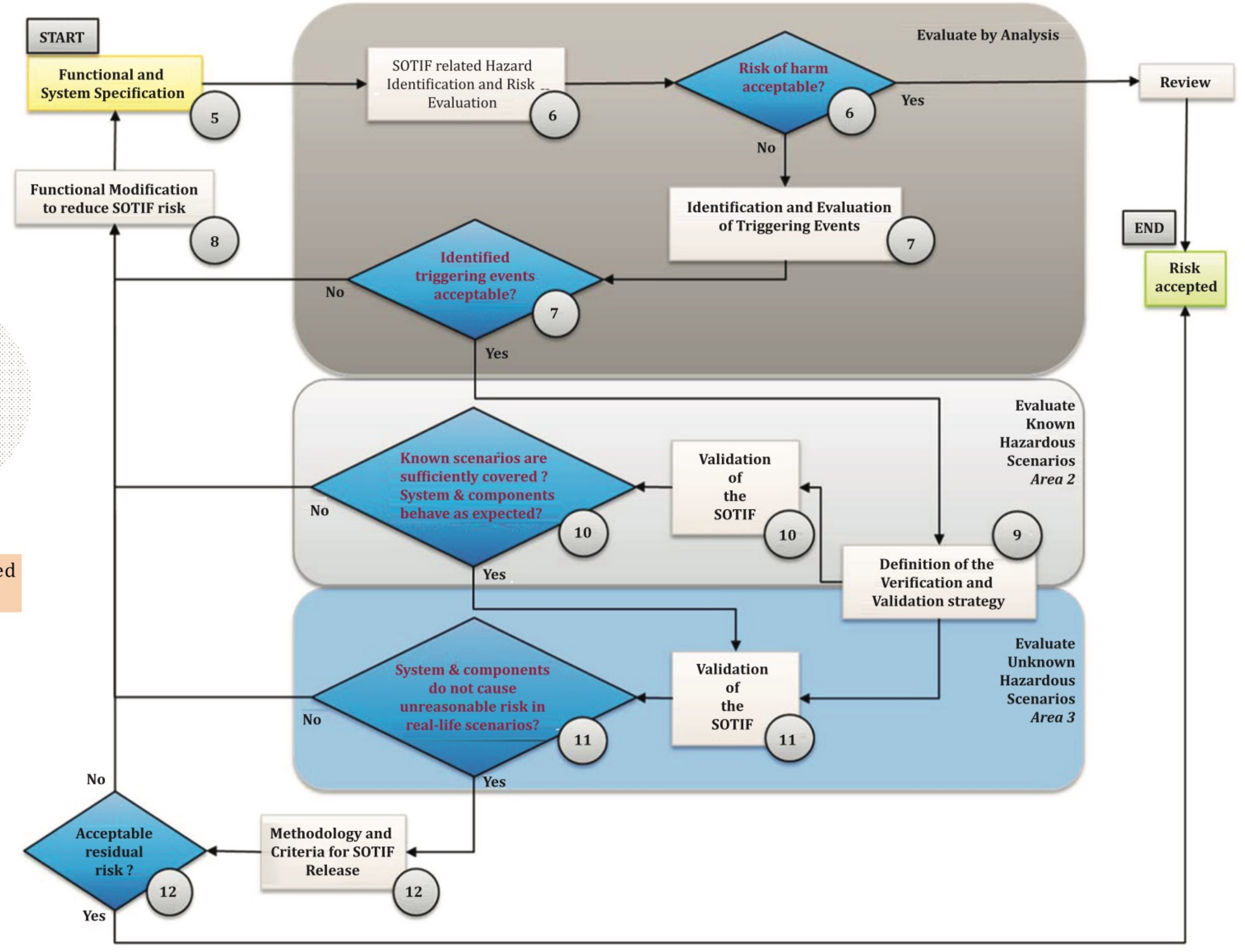


ISO 26262 Functional Safety Standard

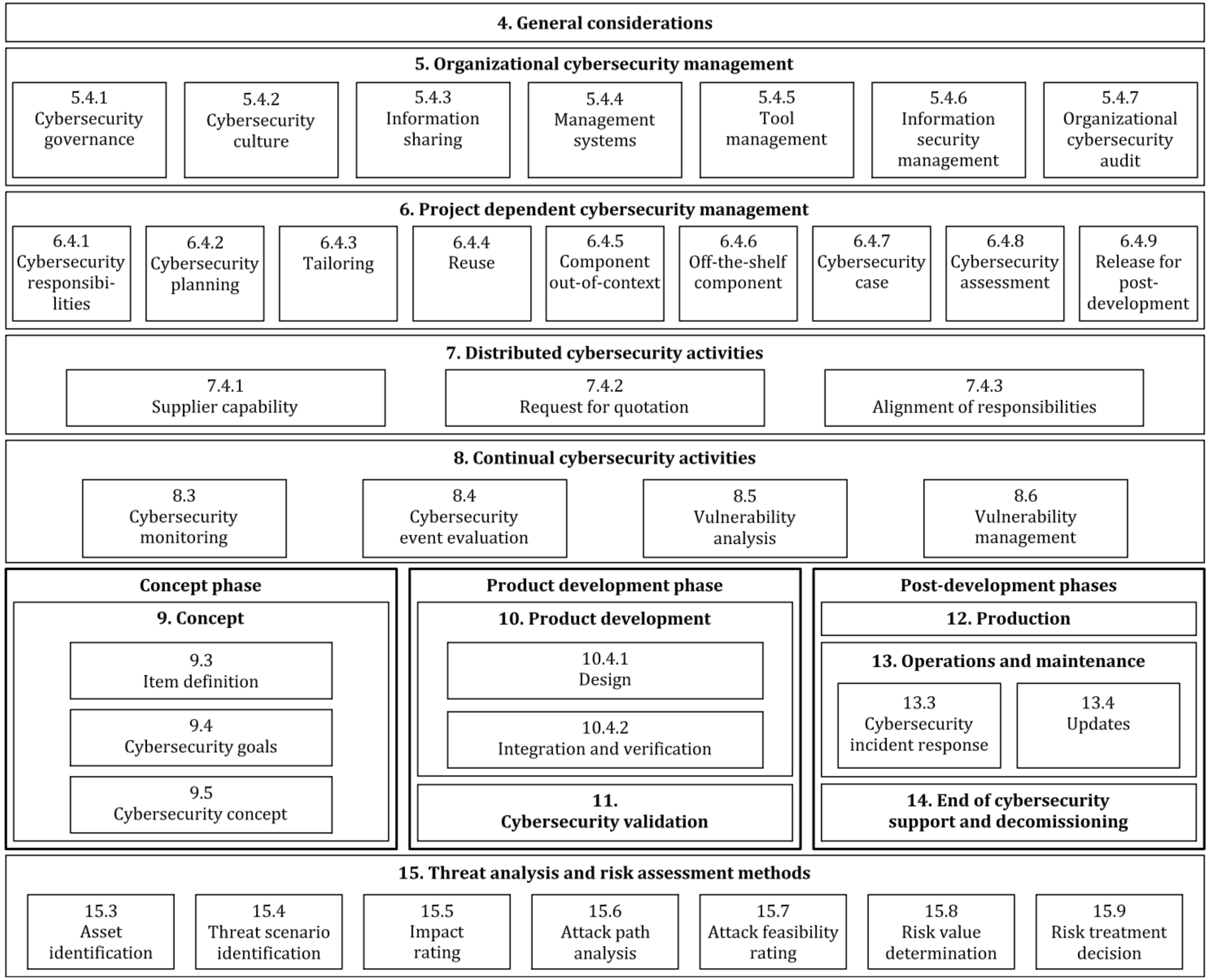


ISO 21438 Intended Safety Standard

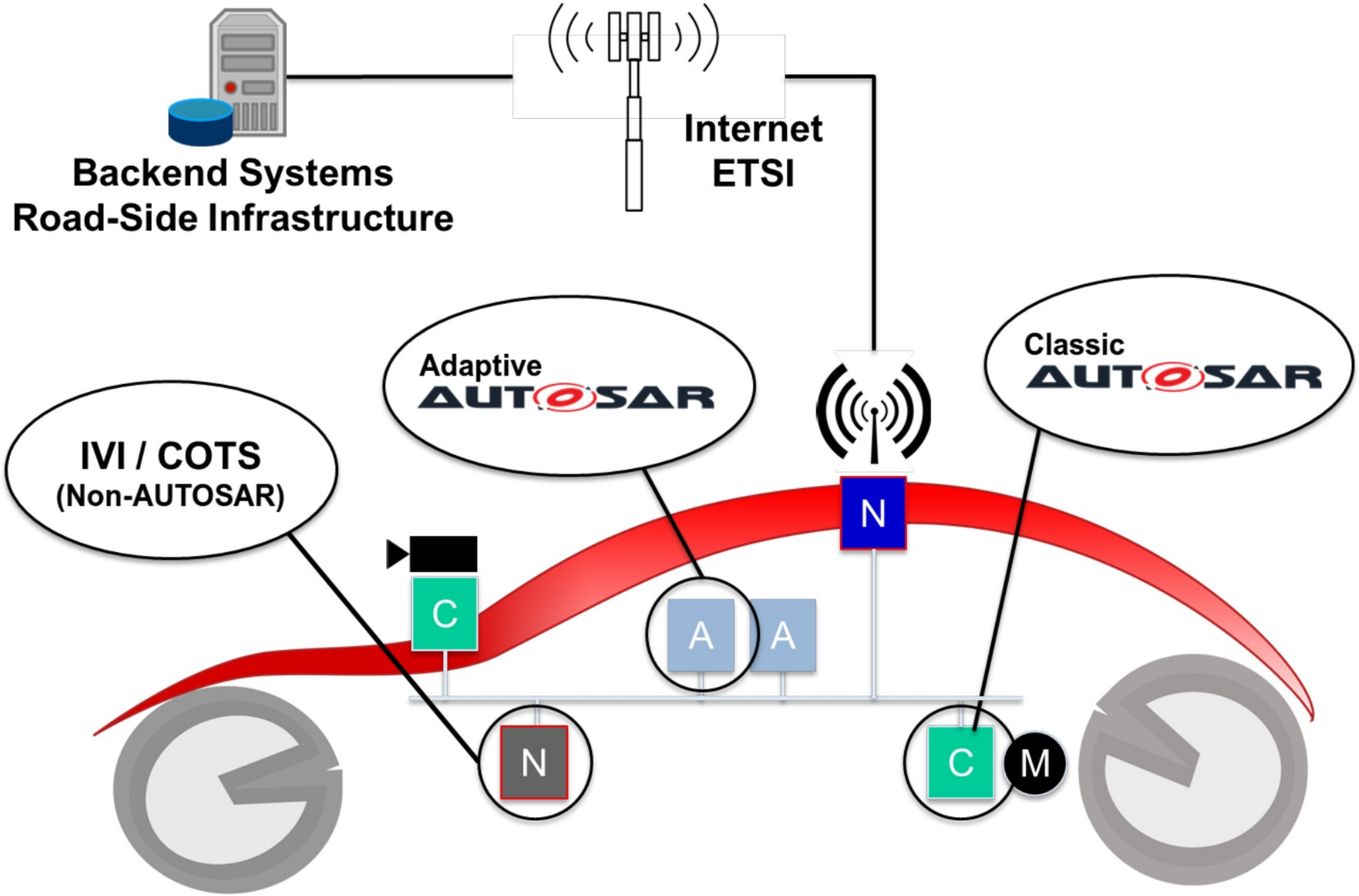




ISO/SAE 21434 Cybersecurity Standard

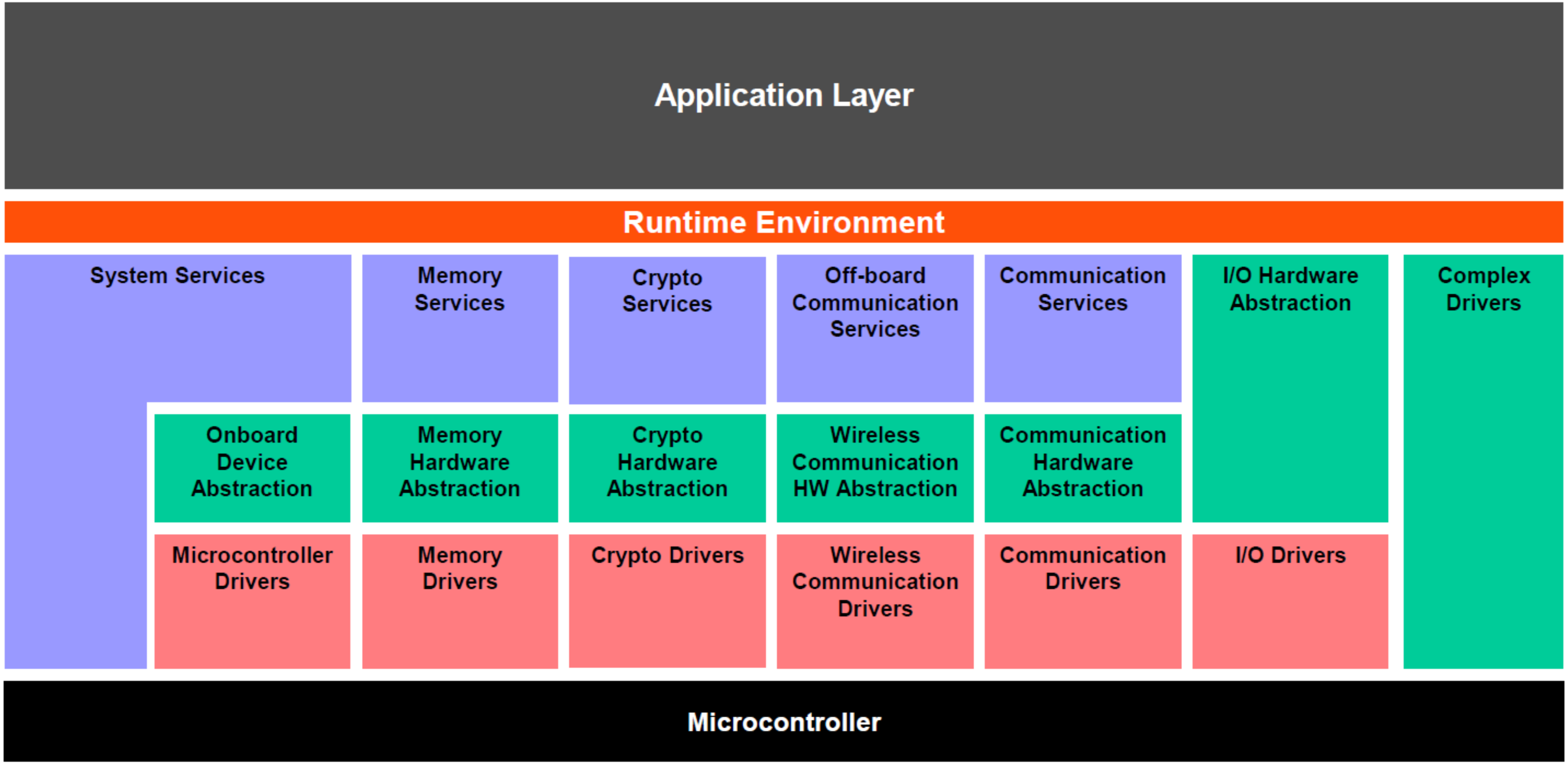


AUTOSAR

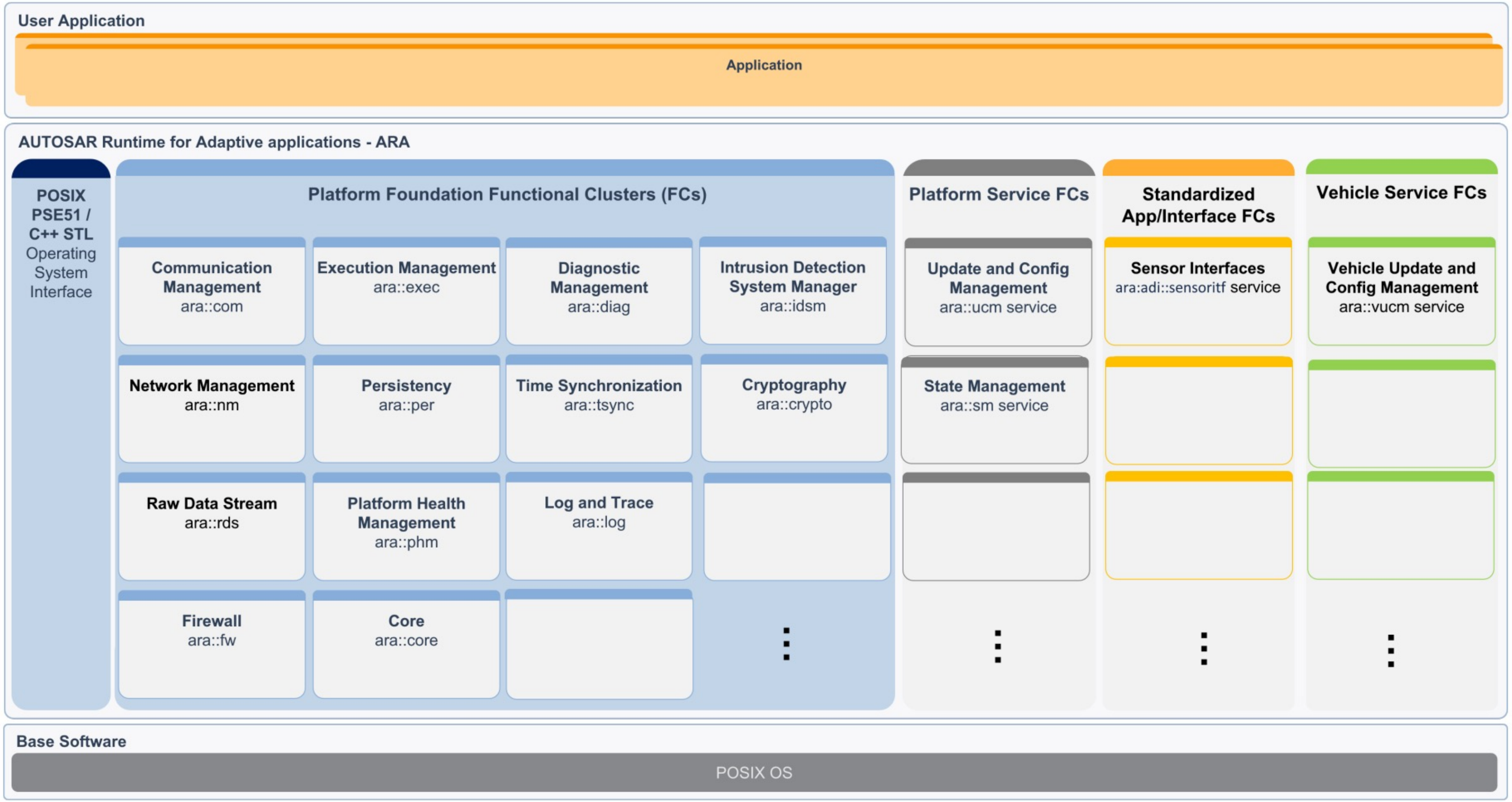


Backend Systems
Road-Side Infrastructure
Internet
ETSI
Adaptive
AUTOSAR
Classic
AUTOSAR
IVI / COTS
(Non-AUTOSAR)
N
C
A
A
N
C
M

Classic AUTOSAR

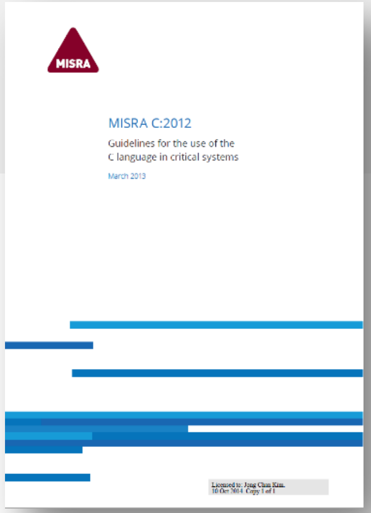


Adaptive AUTOSAR



MISRA-C

Section	Title	# of dirs
7.1	The Implementation	1
7.2	Compilation and build	1
7.3	Requirements traceability	1
7.4	Code design	13



Section	Title	# of rules
8.1	A standard C environment	3
8.2	Unused code	7
8.3	Comments	2
8.4	Character sets and lexical conventions	2
8.5	Identifiers	9
8.6	Types	2
8.7	Literals and constants	4
8.8	Declarations and definitions	14
8.9	Initialization	5
8.10	The essential type model	8
8.11	Pointer type conversions	9

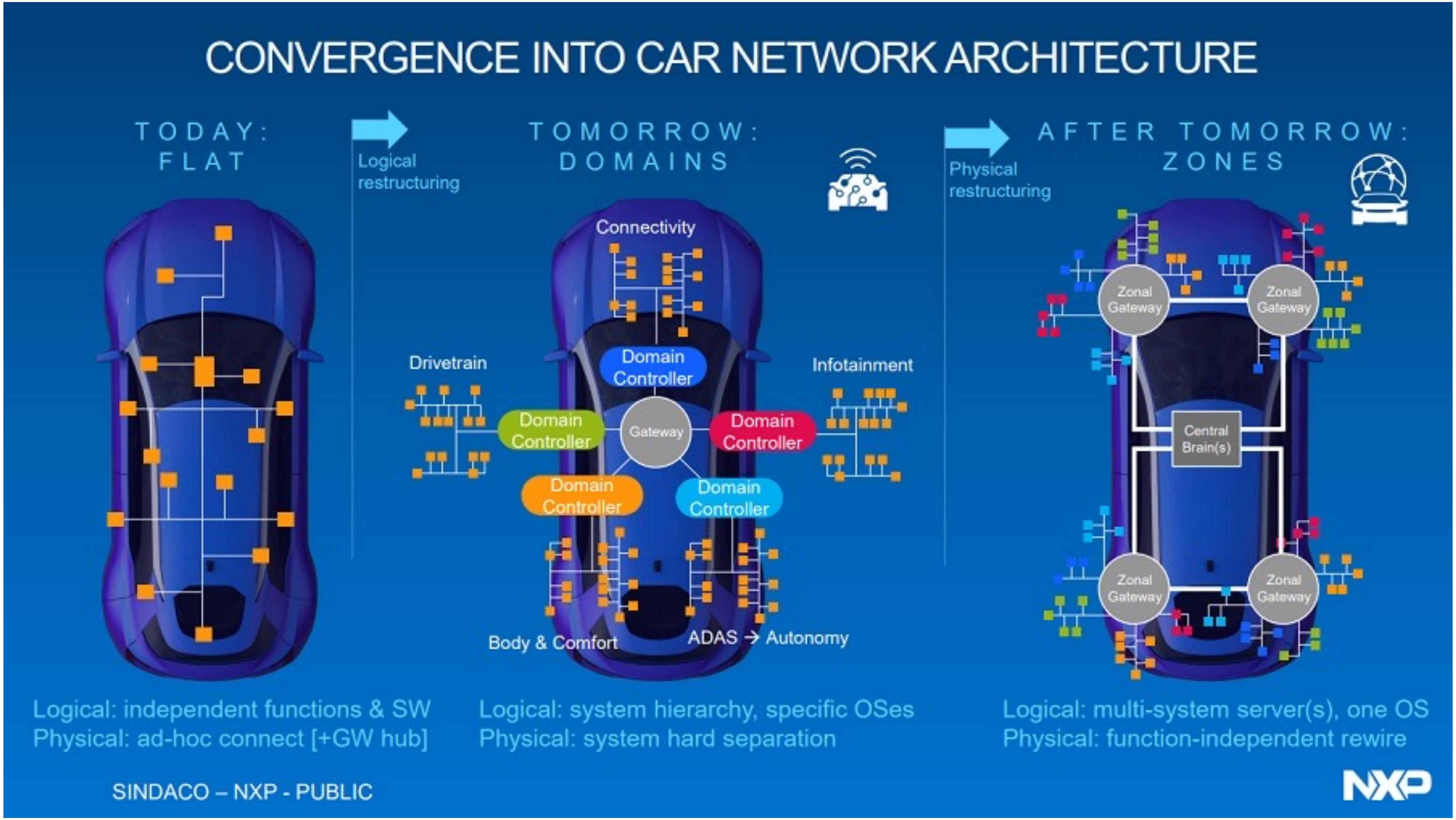
Section	Title	# of rules
8.12	Expressions	4
8.13	Side effects	6
8.14	Control statement expressions	4
8.15	Control flow	7
8.16	Switch statements	7
8.17	Functions	8
8.18	Pointers and arrays	8
8.19	Overlapping storage	2
8.20	Preprocessing directives	14
8.21	Standard libraries	12
8.22	Resources	6

AUTOSAR C++ Guideline

Document Title	Guidelines for the use of the C++14 language in critical and safety-related systems
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	839

Document Status	Final
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	17-10

E/E Architecture

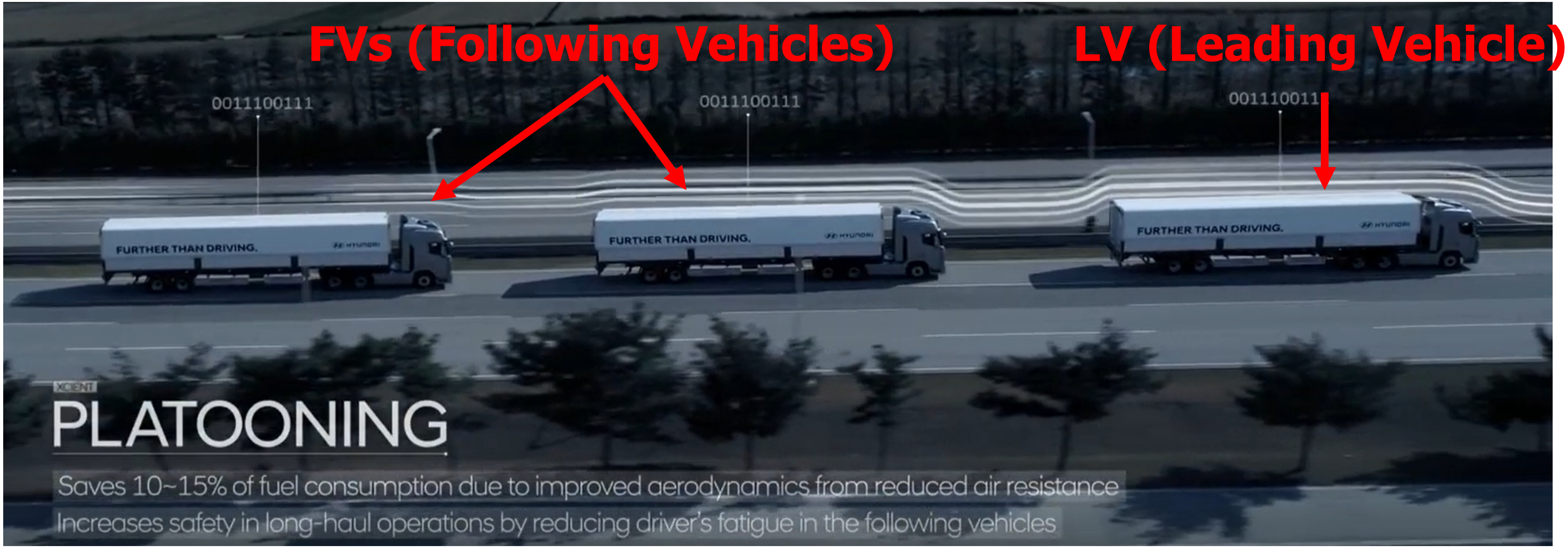


Organizations and Groups

- ARTOP: AUTOSAR User Group for Tooling
- COMASSO: AUTOSAR User Group for Reference Implementation
- COVESA: Connected Car Alliance
- SOAFEE: Open Architecture for Software Defined Vehicles
- Eclipse SDV: Open-Source Projects for Software Defined Vehicles

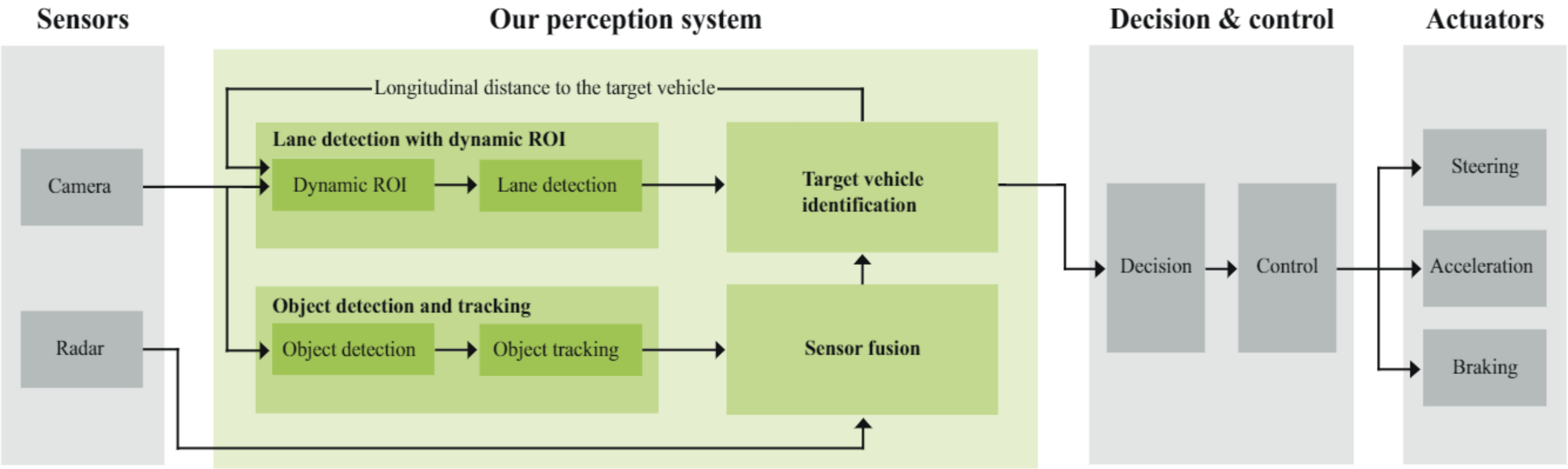
Truck Platooning

- Trucks maintain very close gap (e.g., 12.5 m at 90kph)
- LV is driven by a human driver
- FVs are autonomous



Perception in Truck Platooning

- Lane Detection
- Target Tracking
- Sensor Fusion

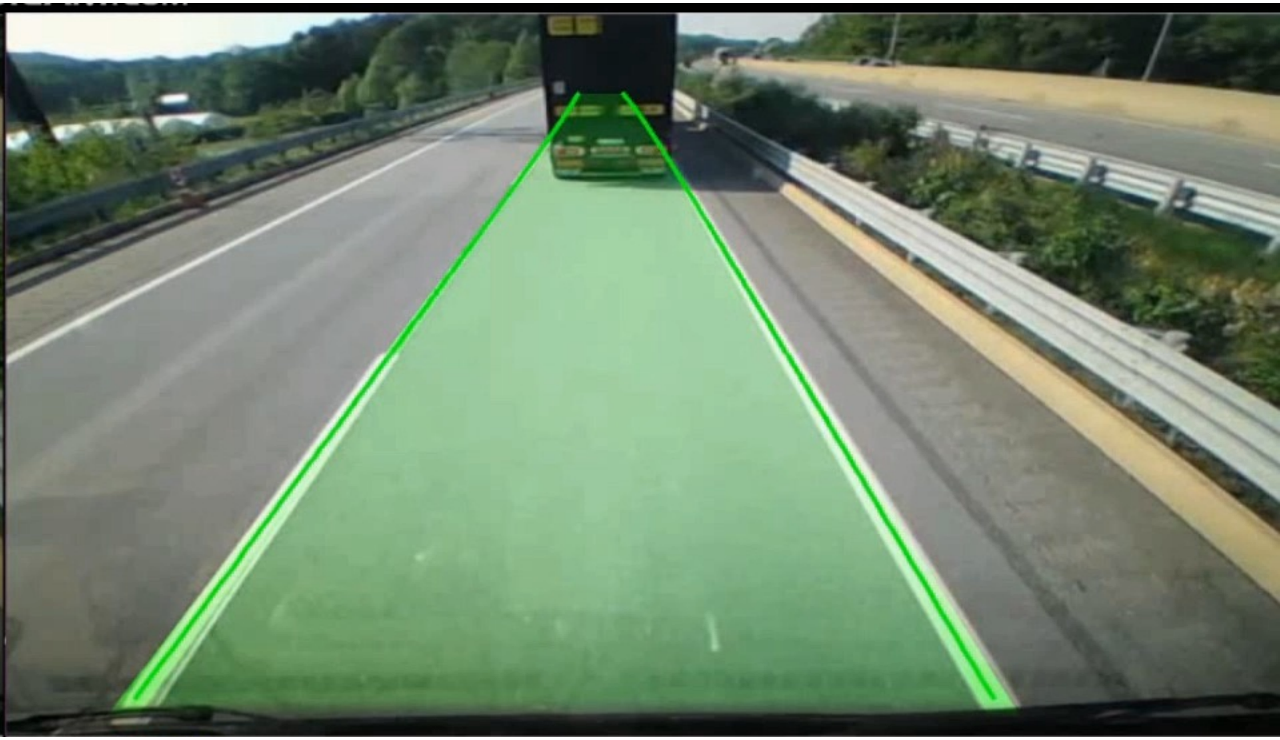


Lane Detection

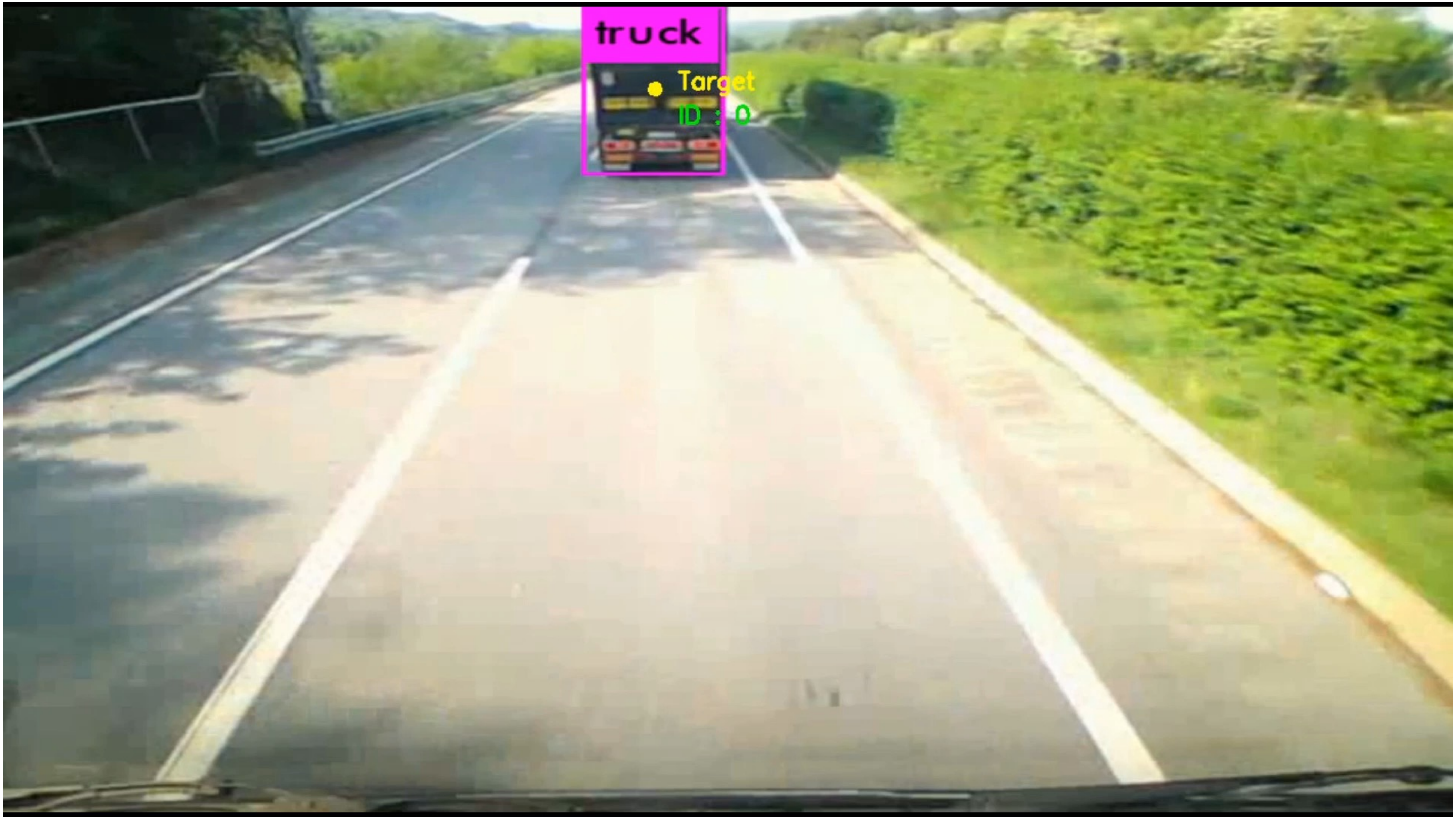
- Determines the ROI (Region Of Interest) based on the dynamic distance to the front truck, which is acquired by the radar sensor

Dynamic ROI

Static ROI



Target Tracking

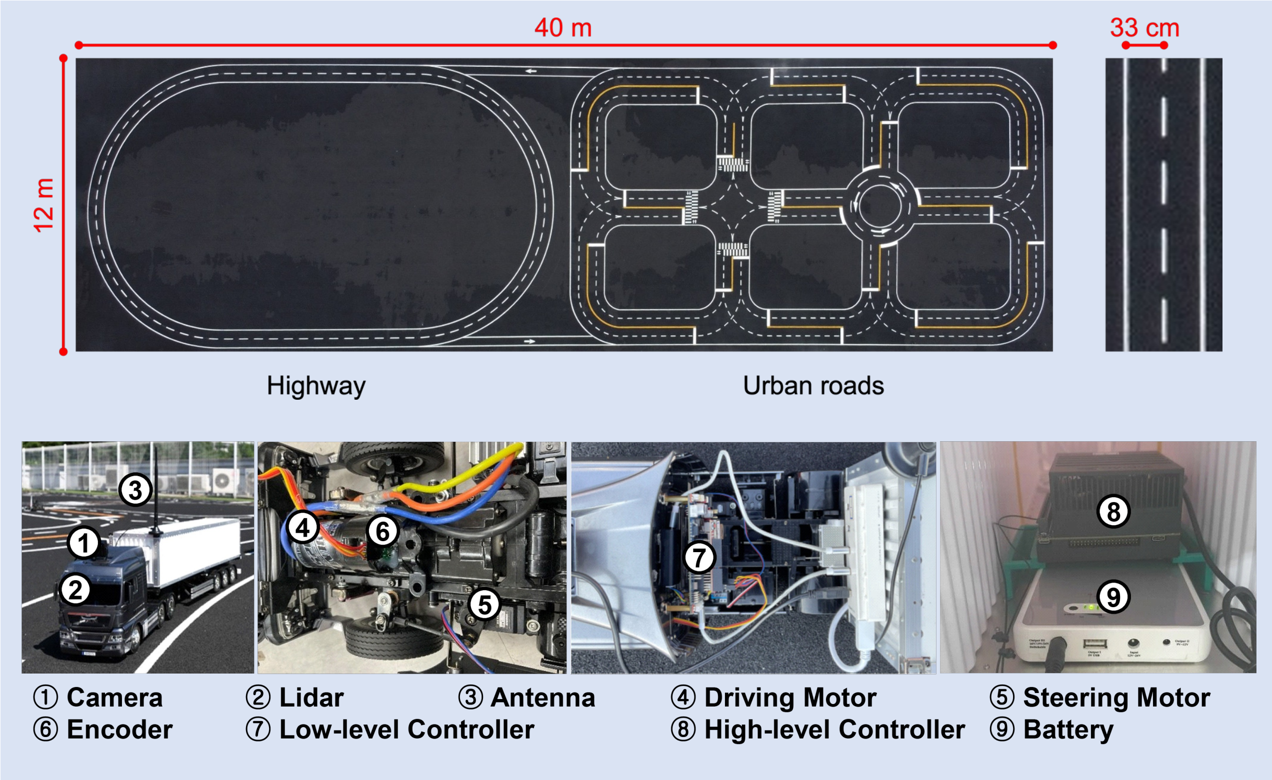


Real-World Truck Platooning (2020)

Scale Truck Platooning

- Scale truck platooning testbed
  - We can no longer access the real trucks
  - We want to test crazy (dangerous) scenarios not possible with real trucks

Hyeongyu Lee, Jaegeun Park, Changjin Koo, Jong-Chan Kim, and Yongsoon Eun, **Cyclops: Open Platform for Scale Truck Platooning**, IEEE International Conference on Robotics and Automation (ICRA 2022), May 2022.

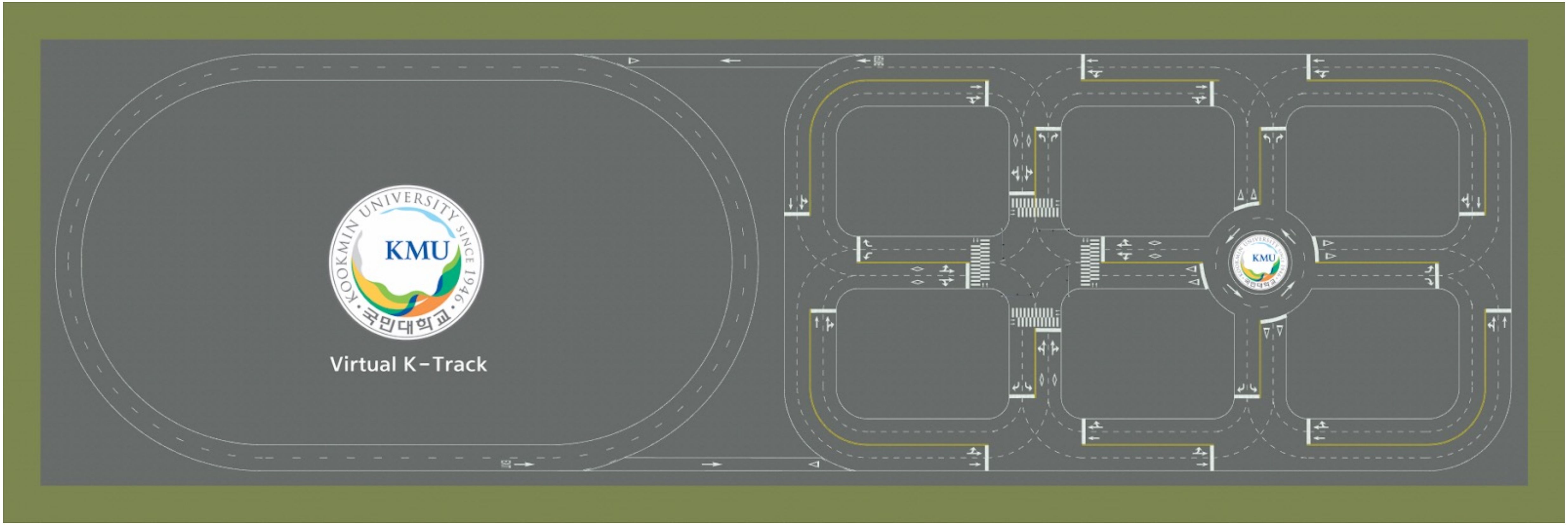


Virtual Track Design

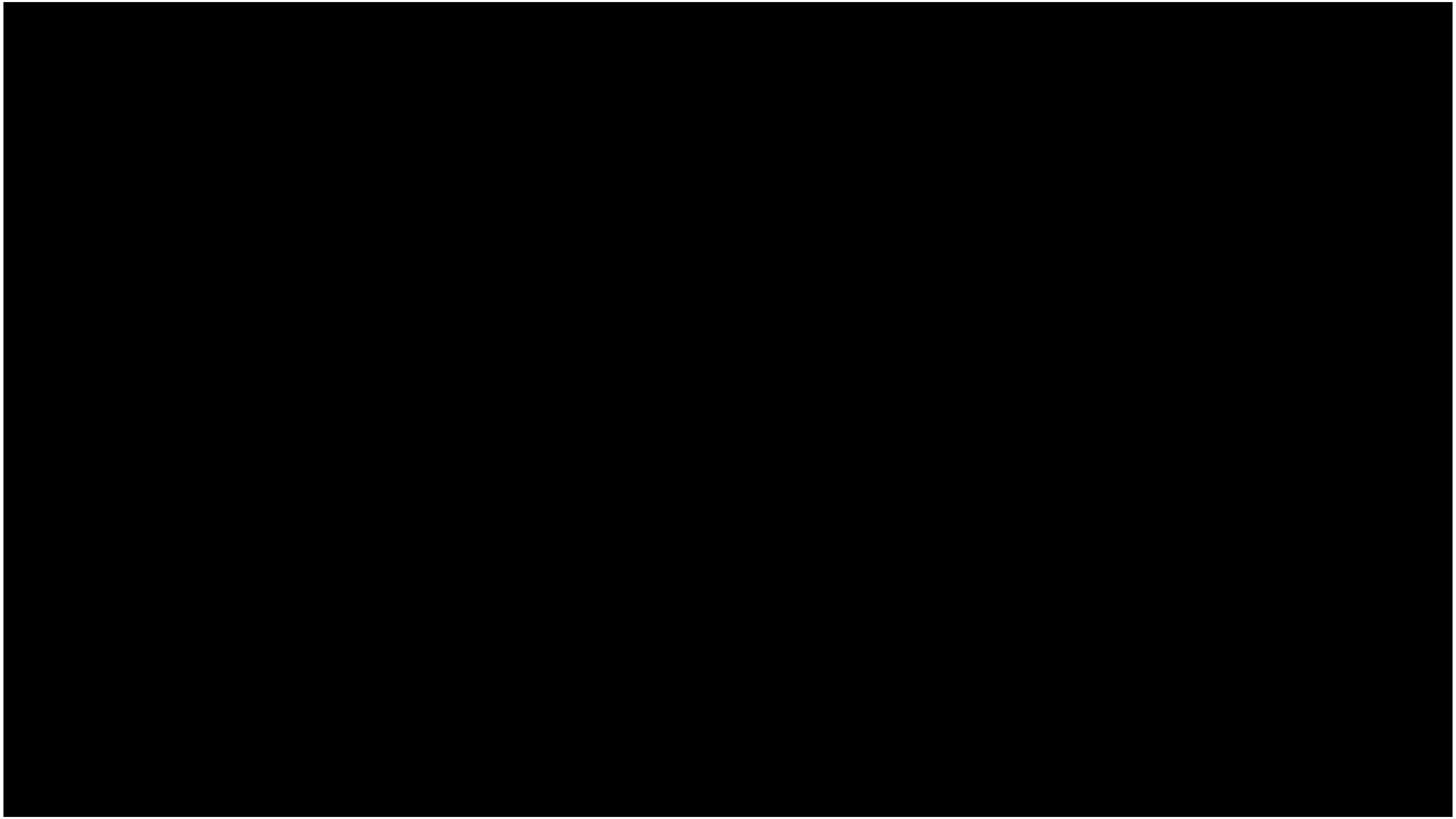
RoadRunner

MathWorks RoadRunner R2022b (1.5.0.fcdcb76871)

License Number: 968398

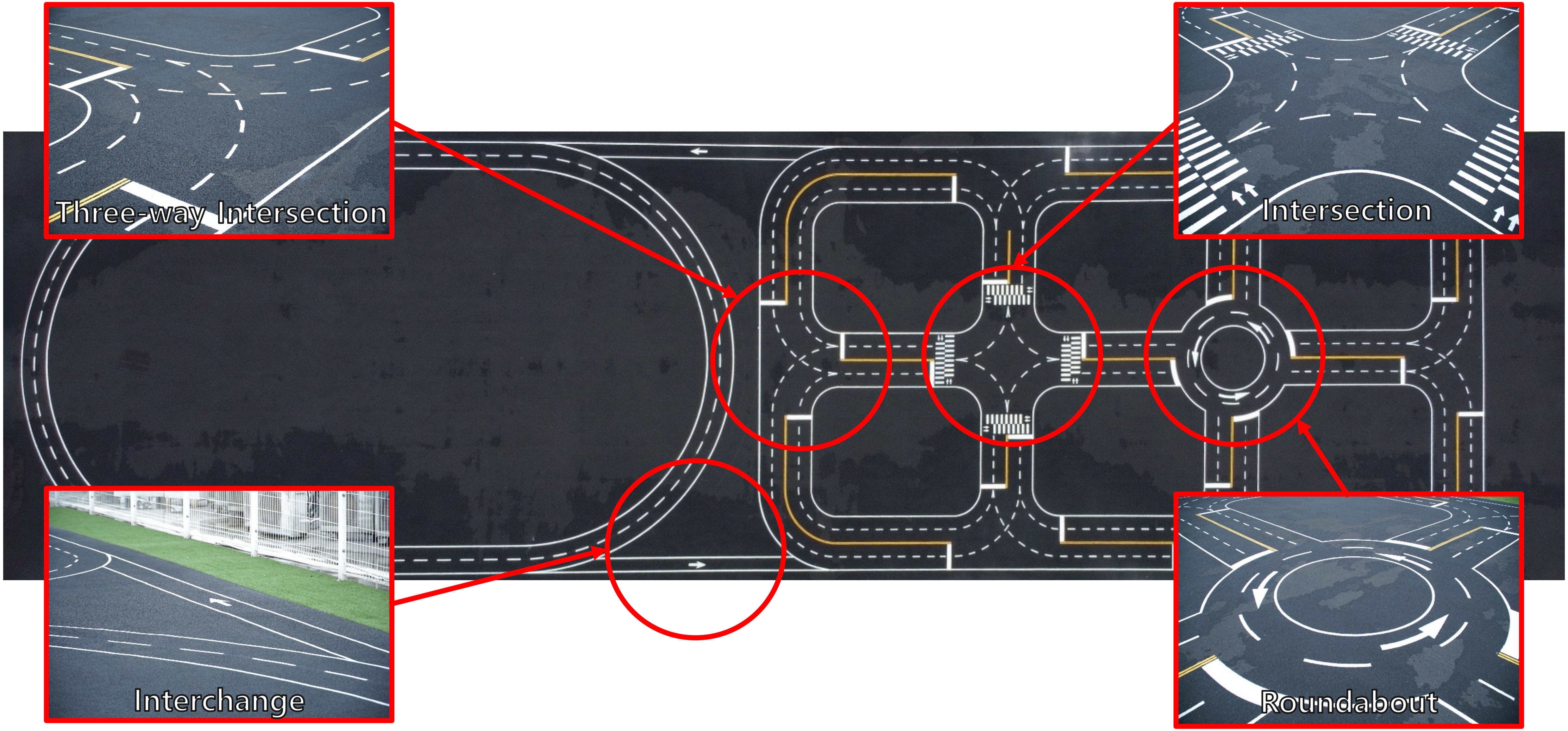


Carla-based Virtual Track Simulation

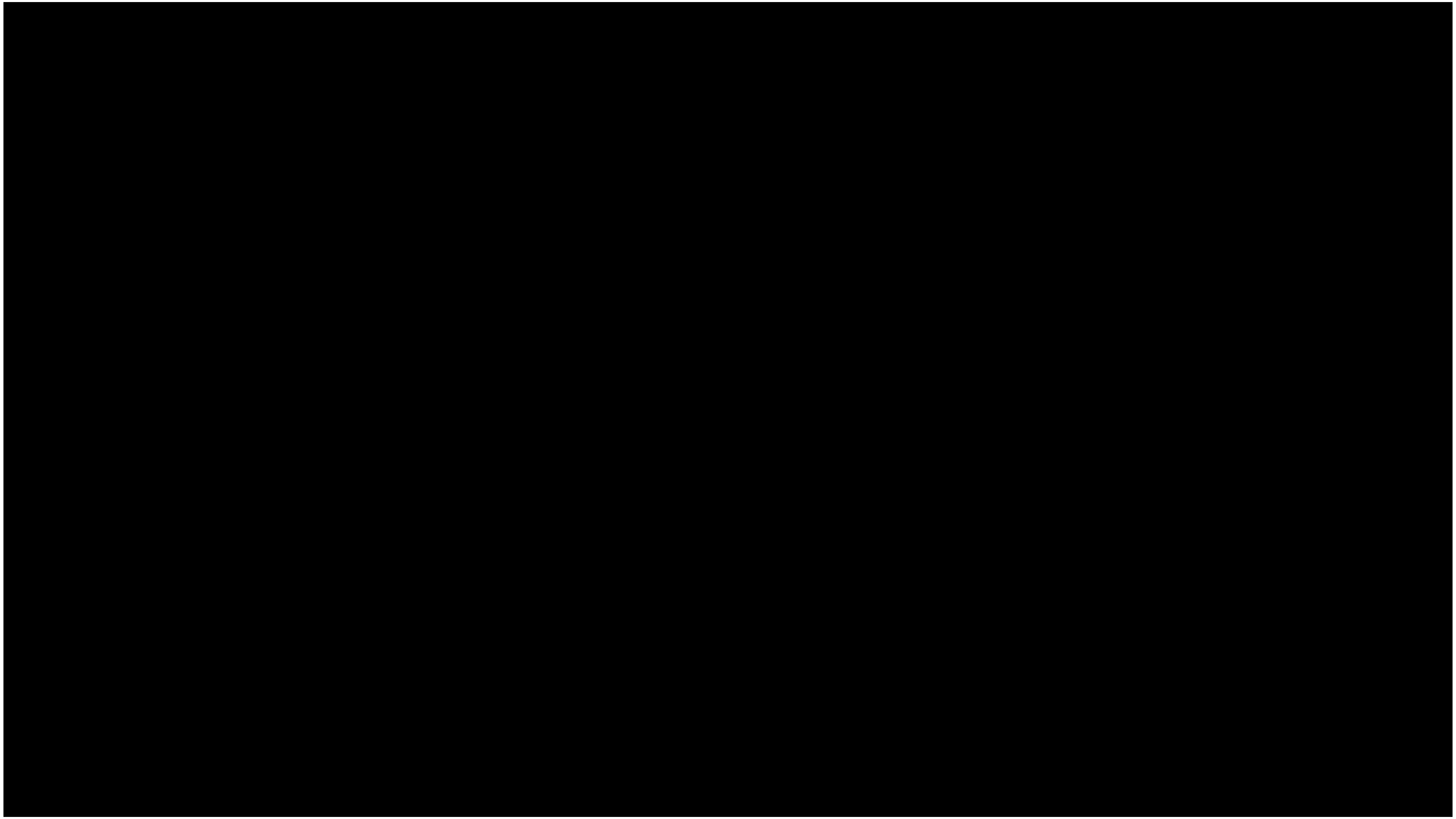


Real-World Pavement Construction

Proving Ground

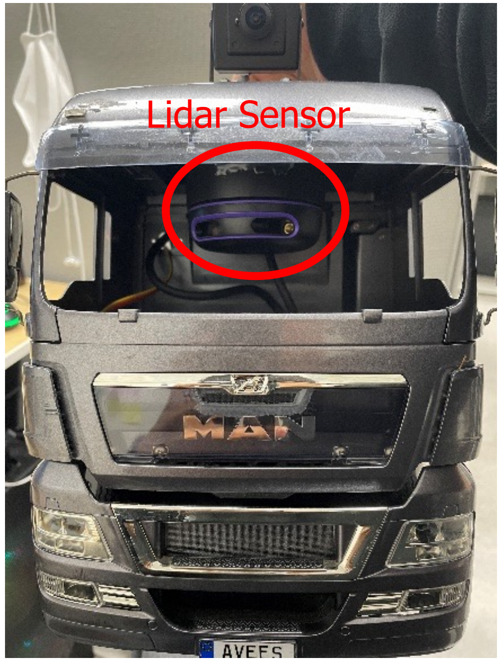


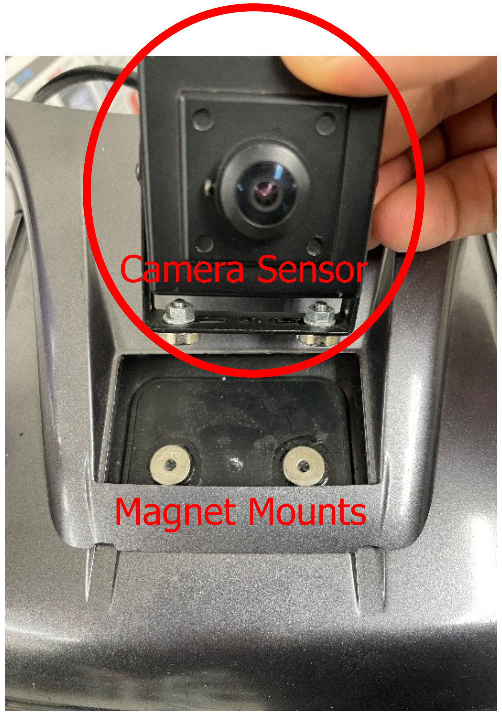
Virtual Track vs. Real Track

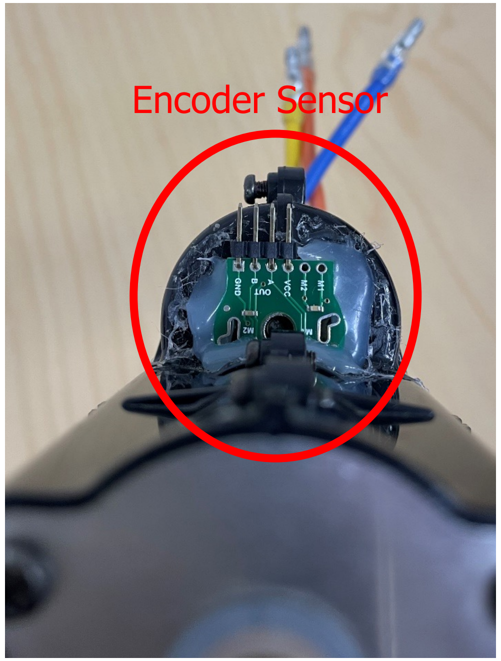


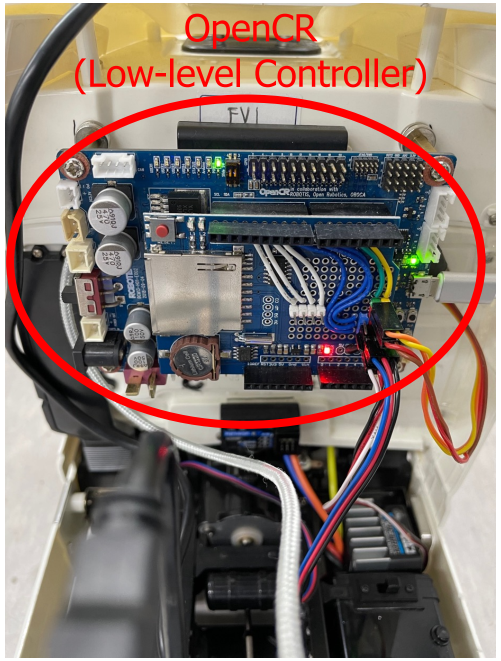
Truck Development

- Sensors: Lidar, Camera, and Encoder
- Computers: High-level controller and low-level controller
- Actuators: Steering motor and driving motor









ICRA 2022
IEEE International Conference on Robotics and Automation
May 23-27, 2022 Philadelphia (PA), USA

Cyclops: Open Platform for Scale Truck Platooning

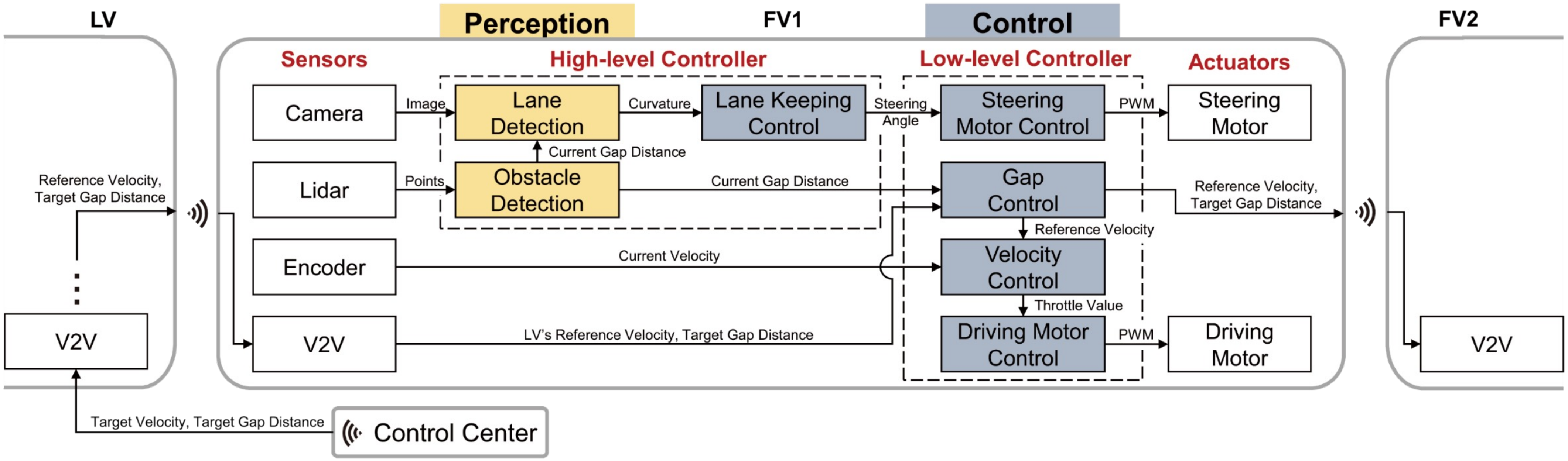
김종찬 교수

자동차공학과 / 자동차IT융합학과

KMU 국민대학교
KOOKMIN UNIVERSITY

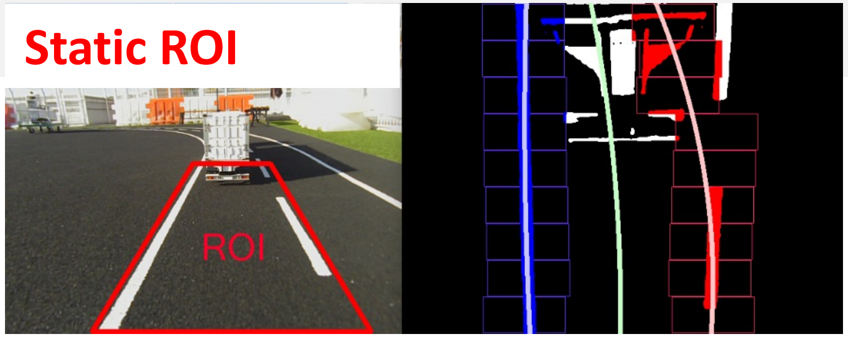
Overall Architecture

- High-level Controller (Nvidia Jetson AGX Xavier)
  - Perception and lane keeping control
- Low-level Controller
  - Gap control and velocity control
  - Motor control (PWM controls)

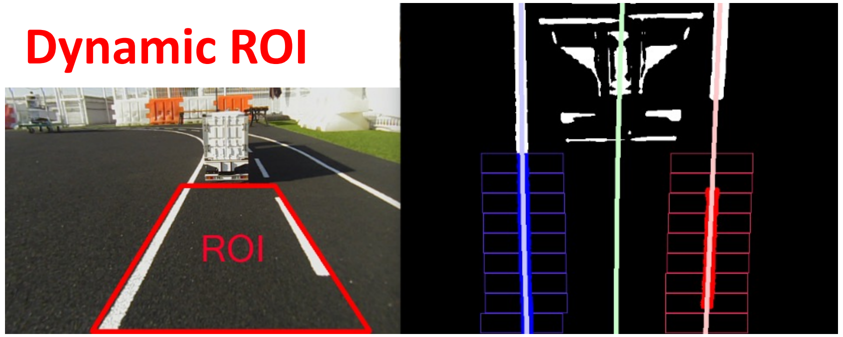


Lane Detection

- Use the conventional histogram method
  - The track lanes are relatively easy to perceive



- Lidar-based dynamic ROI setting
  - For reducing noises by the preceding trailer



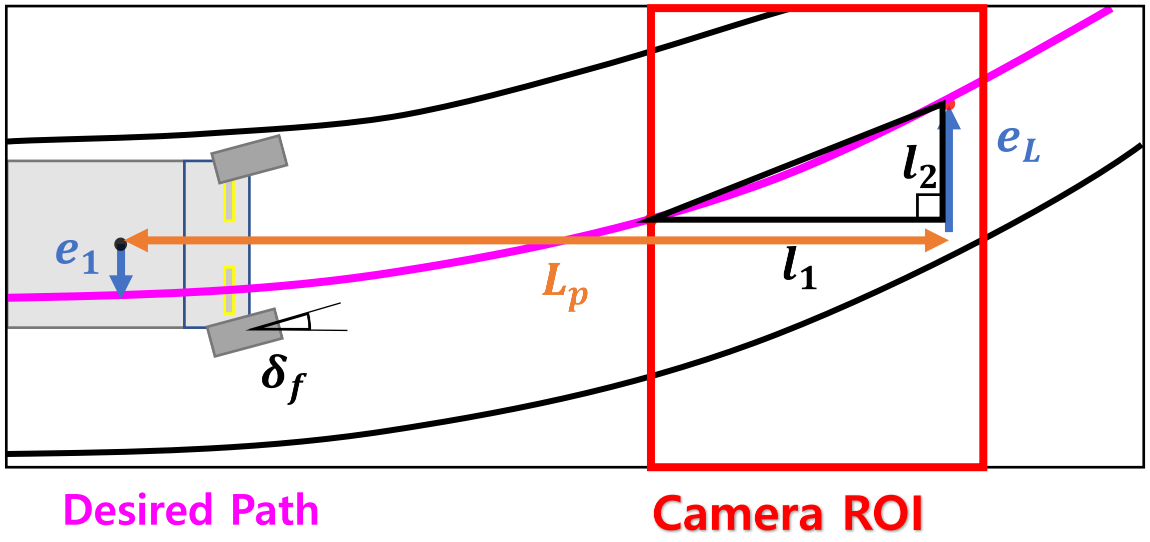
- It had to be optimized for various lighting conditions

[redacted]

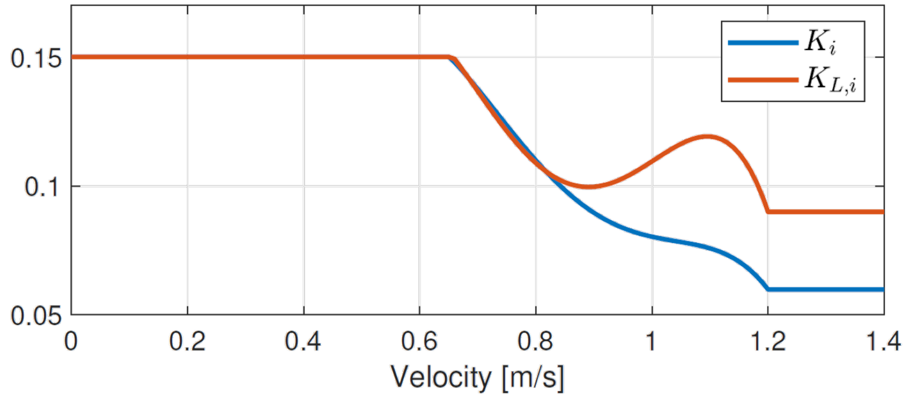
[redacted]

Lane Keeping Control

- Determine the desired path by observing lanes
- Control the steering angle to follow the desired path



i	a_i	b_i	c_i
0	-1.1446×10^{-5}	0.048278	-47.94
1	-2.0975×10^{-5}	0.08152	-76.87
2	-1.0444×10^{-5}	0.043253	-42.3682



e_1 = *Lateral Position Error*

e_L = *Preview Distance Error*

δ_f = *Steering Angle*

$$e_L \approx e_1 - \left(L_p \times \left(\frac{l_2}{l_1}\right)\right)$$

$$\delta_f \approx (-K_1 \times e_L) + (-K_2 \times e_1)$$

Gap Control

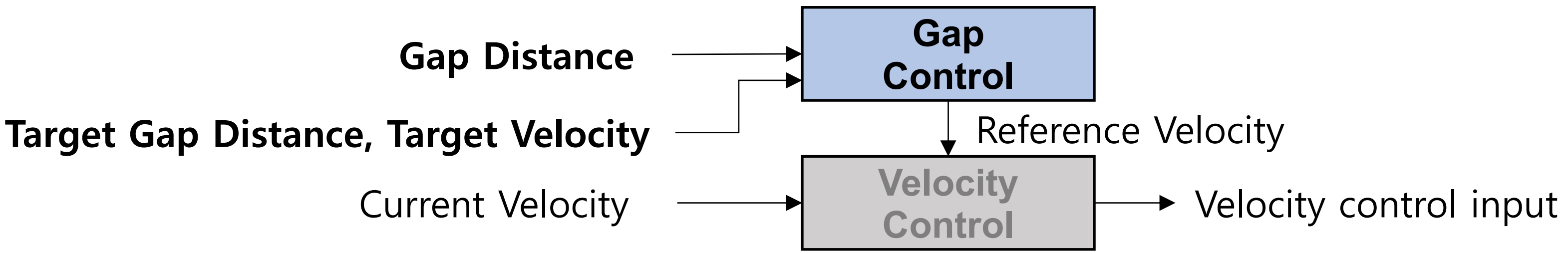
- FF (Feed Forward) + PD (Proportional-Differential) controller

$$v_{r,i}(k) = \overset{\textbf{FF}}{\left[\bar{v}_{r,i-1}(k)\right]} - \overset{\textbf{PD}}{\left[K_{GP,i}\tilde{d}_i(k) - \frac{K_{GD,i}\left(\tilde{d}_i(k) - \tilde{d}_i(k-1)\right)}{T_s}\right]}$$

$$\tilde{d}_i(k) = d_{r,i}(k) - d_i(k)$$

$$sat_i(v_{r,i}) = \begin{cases} V_r^{max}, & if\ v_{r,i} \geq V_r^{max} \\ 0, & if\ v_{r,i} \leq 0 \\ v_{r,i}, & otherwise \end{cases}$$

$\boldsymbol{i = 0(LV), 1(FV1), 2(FV2)}$
$\boldsymbol{v_{r,i} = velocity\ reference}$
$\boldsymbol{K_{GP,i}, K_{GD,i} = control\ parameters}$
$\boldsymbol{d_i = current\ gap\ distance}$
$\boldsymbol{d_{r,i} = target\ gap\ distance}$
$\boldsymbol{\tilde{d}_i = gap\ tracking\ error}$
$\boldsymbol{T_s = sampling\ time}$



Velocity Control

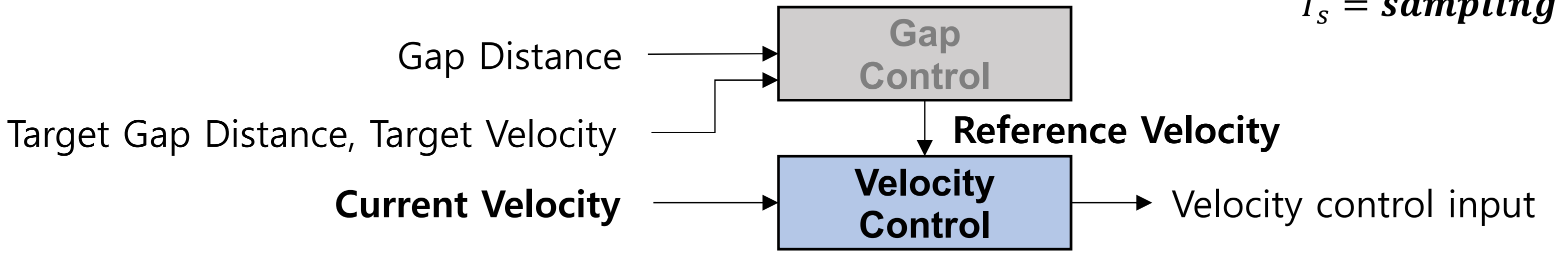
- FF (Feed Forward) + AWPI (Anti-Windup-Proportional-Integral) controller

$$u_{v,c,i}(k) = \overbrace{\left[K_{F,i} v_{r,i}(k)\right]}^{\text{FF}} + \overbrace{\left[\sum_{j=0}^{j=k-1} K_{A,i}\left(\bar{u}_{v,c,i}(j) - u_{v,c,i}(j)\right) - K_{P,i}\tilde{v}_i(k) - \sum_{j=0}^{j=k} K_{I,i}\tilde{v}_i(j) T_s\right]}^{\text{AWPI}}$$

$$\tilde{v}_i(k) = \bar{v}_{r,i}(k) - v_i(k)$$

$$sat_i(u_{v,c,i}) = \begin{cases} V_i^{max}, & if\ u_{v,c,i} \geq V_i^{max} \\ 0, & if\ u_{v,c,i} \leq 0 \\ u_{v,c,i}, & otherwise \end{cases}$$

$\boldsymbol{i = 0(LV), 1(FV1), 2(FV2)}$
$\boldsymbol{u_{v,c,i} = velocity\ control\ input}$
$\boldsymbol{K_{F,i}, K_{P,i}, K_{I,i}, K_{A,i}}$
$\boldsymbol{= control\ parameters}$
$\boldsymbol{v_i = current\ velocity}$
$\boldsymbol{v_{r,i} = reference\ velocity}$
$\boldsymbol{\tilde{v}_i = velocity\ tracking\ error}$
$\boldsymbol{T_s = sampling\ time}$



Scale Truck Platooning (ICRA 2022)

Cyclops: Open Platform for Scale Truck Platooning

Hyeongyu Lee[1], Jaegeun Park[2], Changjin Koo[1], Jong-Chan Kim[1], and Yongsoon Eun[2]

[1]Kookmin University, Graduate School of Automotive Engineering
[2]DGIST, Department of Information and Communication Engineering

국민대학교
KOOKMIN UNIVERSITY

DGIST
대구경북과학기술원

DATE 23
DESIGN, AUTOMATION & TEST IN EUROPE
17 – 19 April 2023 · Antwerp, Belgium
The European Event for Electronic System Design & Test

Phalanx: Failure-Resilient Truck Platooning System

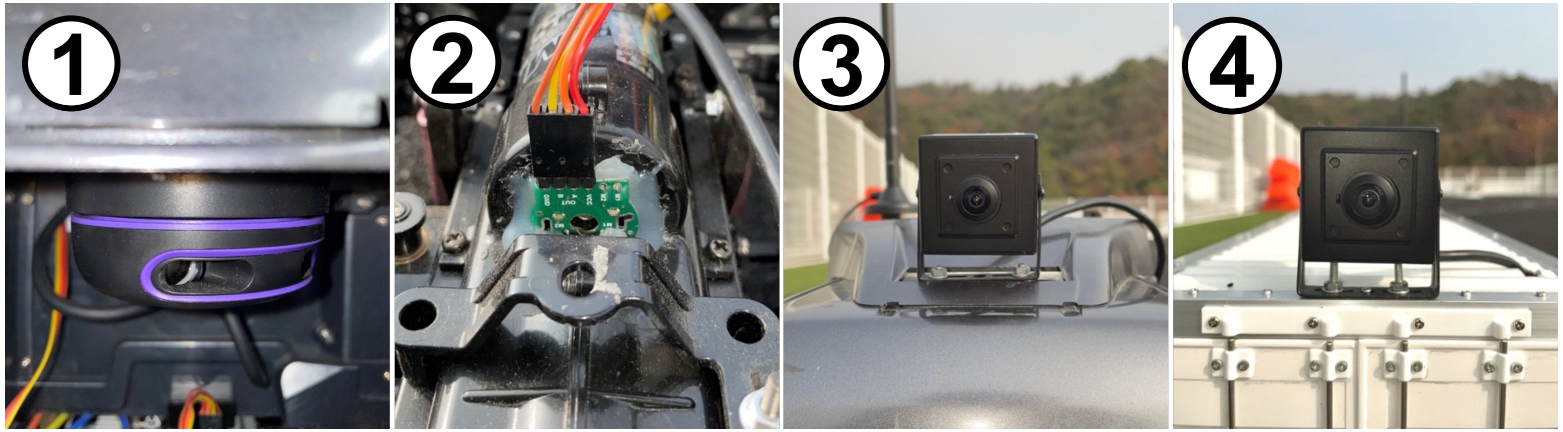
Changjin Koo[1] , Jaegeun Park[2] , Taewook Ahn[1], Hongsuk Kim[1] , Jong-Chan Kim[1] , Yongsoon Eun[2]

[1]Kookmin University, Republic of Korea
[2]DGIST, Republic of Korea

System Model

- One LV and two FVs (FV1 and FV2)
- Four sensors in each truck
  1. Lidar — Forward object detection and gap measurement
  2. Encoder — Measurement of the vehicle's current velocity
  3. Front-facing camera — Lane detection and lateral control
  4. Rear-facing camera NEW — Rear view monitoring → Non-essential in normal driving

Three essential sensors (①–③)



Sensor Failure Scenarios

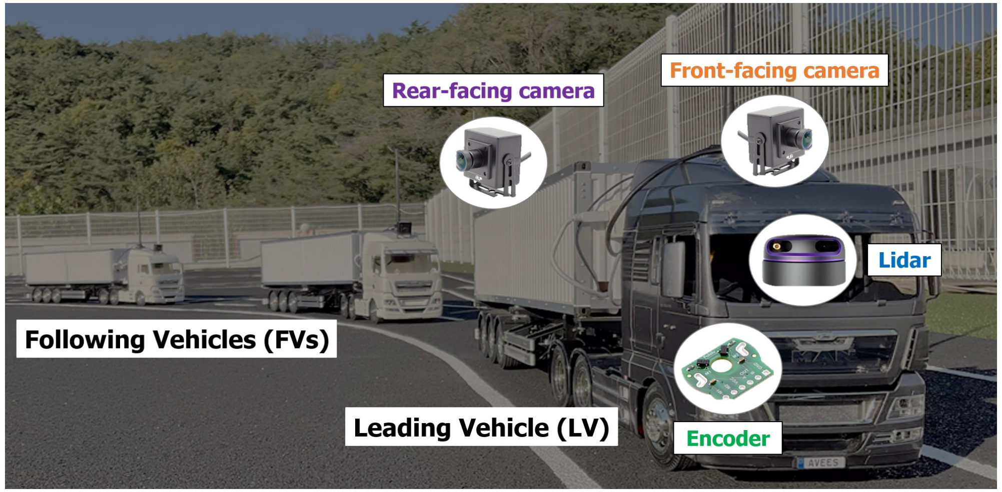
- Six single and dual sensor failure scenarios are considered
  - Lidar failure → L
  - Encoder failure → E
  - Camera failure → C
  - Lidar + Encoder failure → L + E
  - Encoder + Camera failure → E + C
  - Lidar + Camera failure → L + C

Single sensor failures

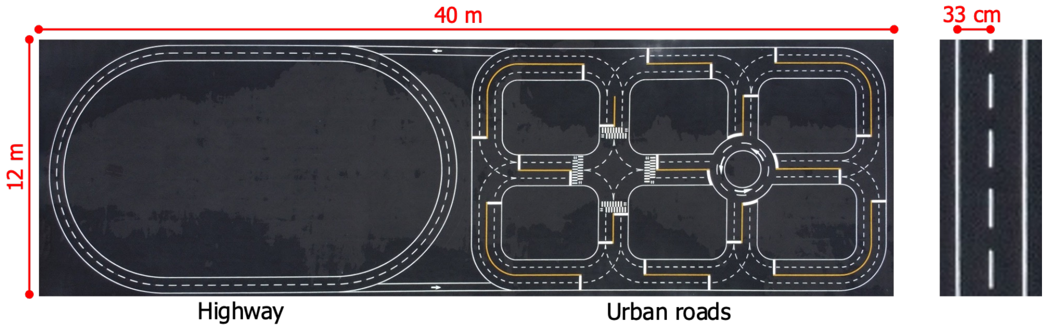
Dual sensor failures

Experiments

- Implementation
  - 1/14 scale trucks

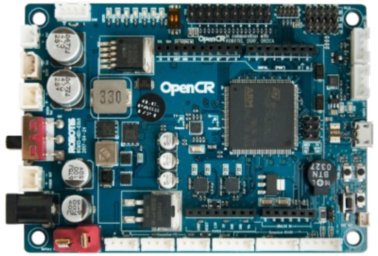


  - Proving ground



- Computing platforms
  - ✓ High-level controller
    : NVIDIA jetson Xavier

  - ✓ Low-level controller
    : OpenCR 1.0

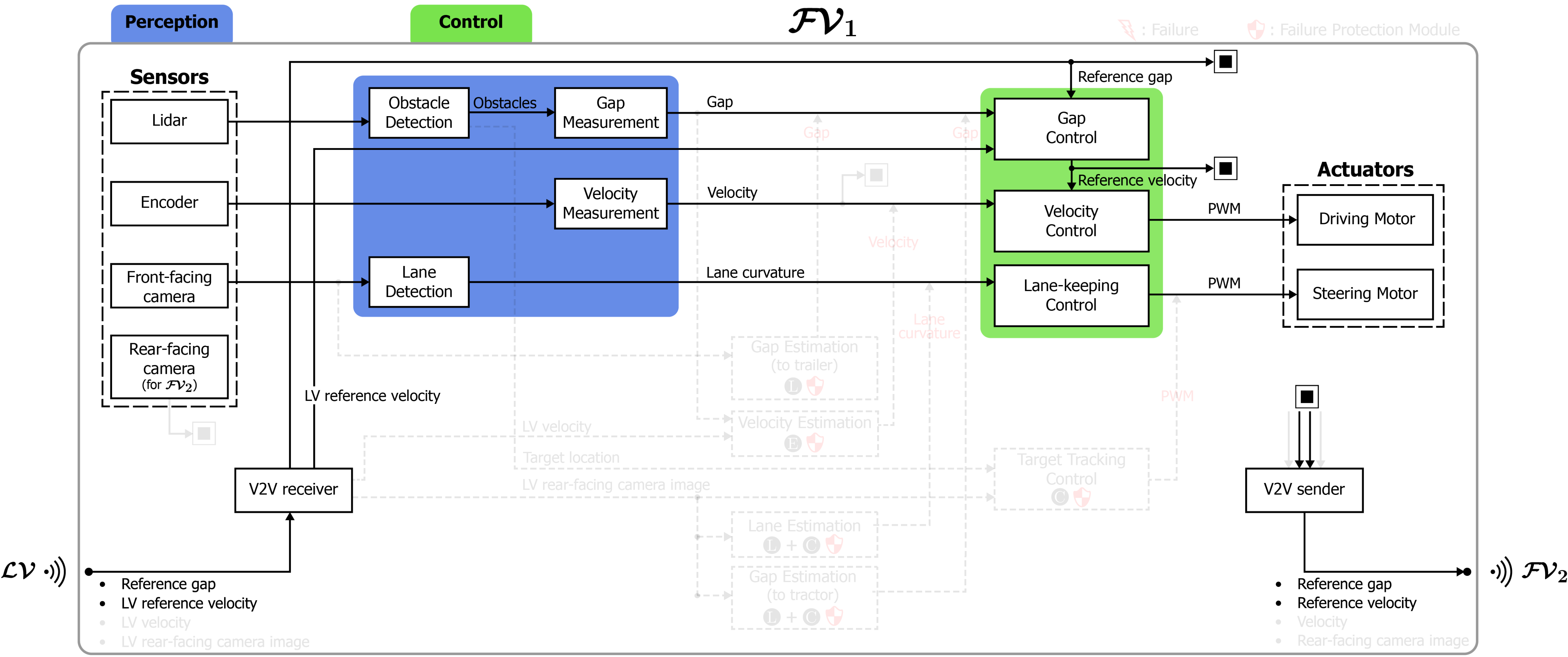


- Software platforms

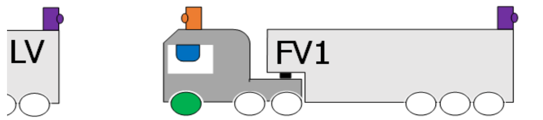
ROS OpenCV

Source: H. Lee, J. Park, C. Koo, J. -C. Kim and Y. Eun, "Cyclops: Open Platform for Scale Truck Platooning," *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8971-8977.

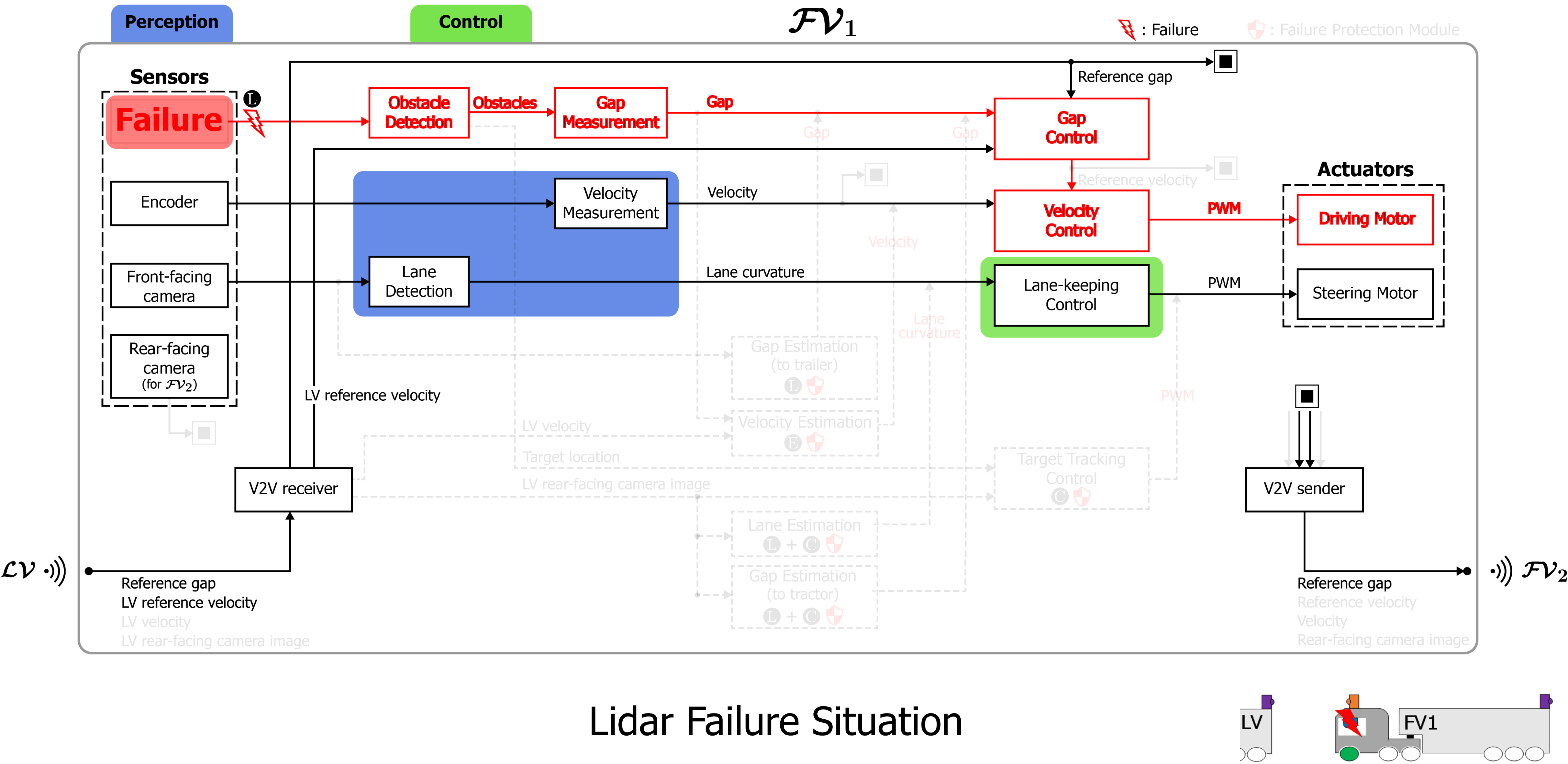
System Architecture (Normal)



Normal Situation

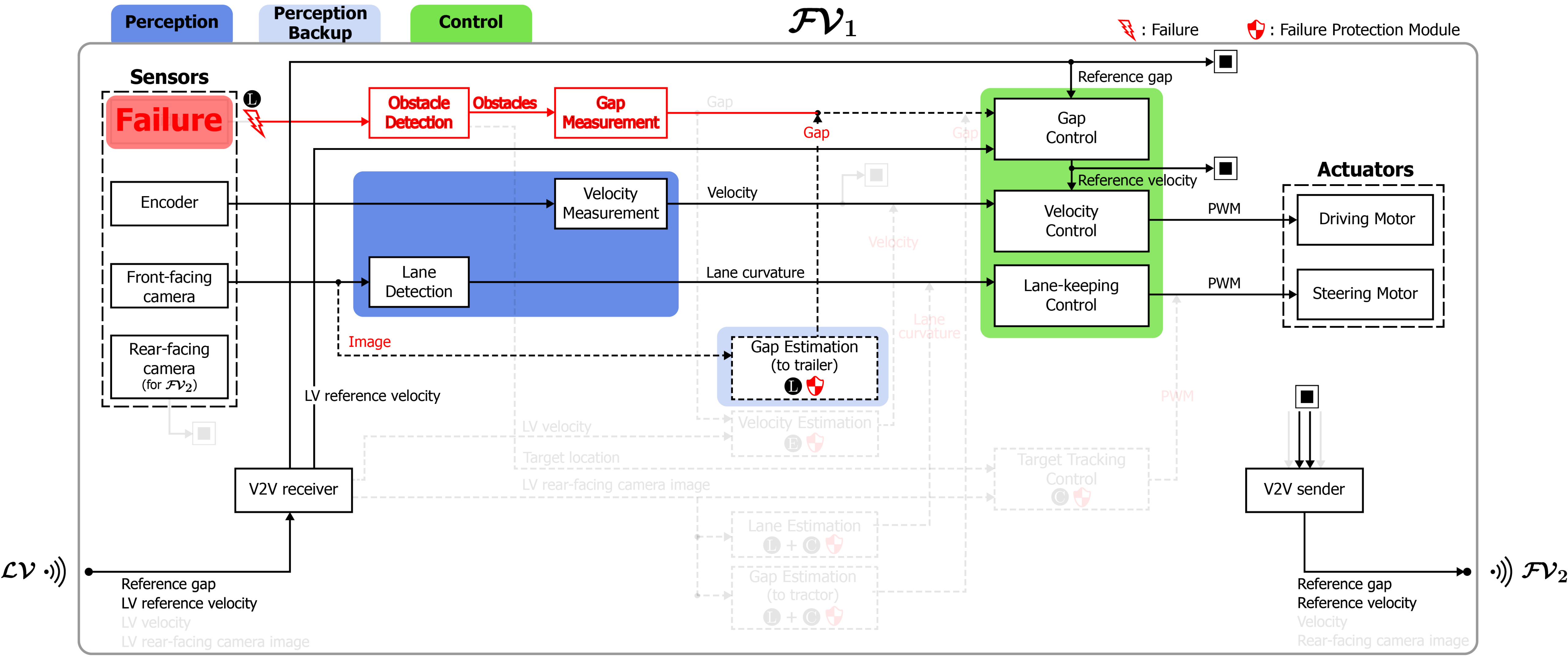


System Architecture (Ⓛ)

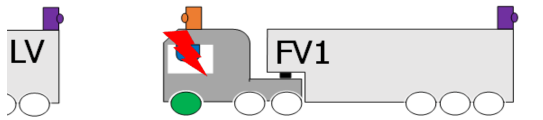


Lidar Failure Situation

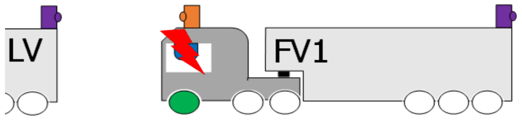
System Architecture (L)



Lidar Failure Situation

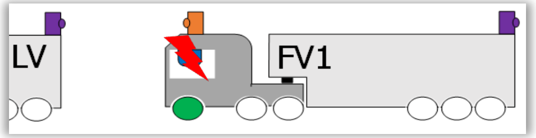


Gap Estimation for Ⓛ

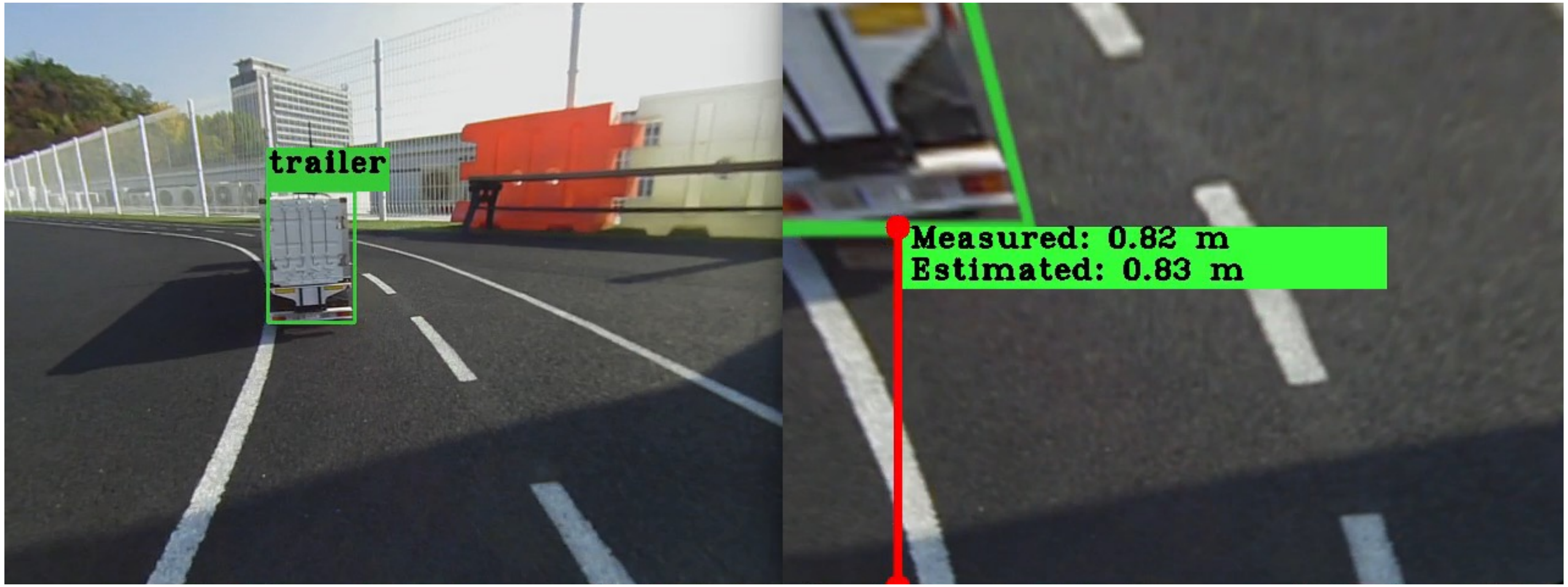


- Estimate gap distance in case of lidar failure
  1. Bounding box detection through 2D object detection
  2. Bird's-eye view (BEV) transformation
  3. Count the number of vertical pixels to the preceding trailer
  4. Convert the number of vertical pixels to distance (ratio = 490 pixels : 1m)

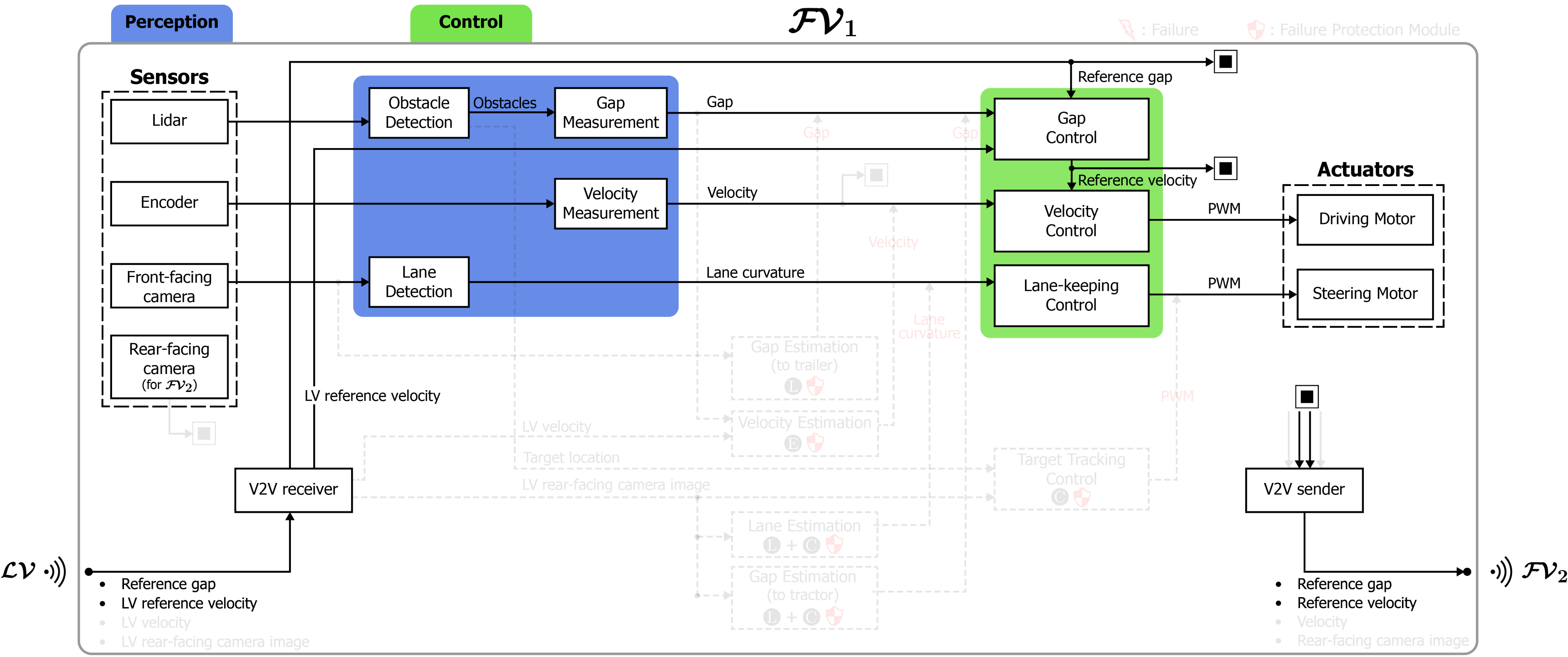
Gap Estimation for Ⓛ



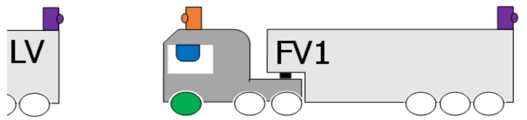
- Estimate gap distance in case of lidar failure
  ① Bounding box detection through 2D object detection
  ② Bird's-eye view (BEV) transformation
  ③ Count the number of vertical pixels to the preceding trailer
  ④ Convert the number of vertical pixels to distance (ratio = 490 pixels : 1m)



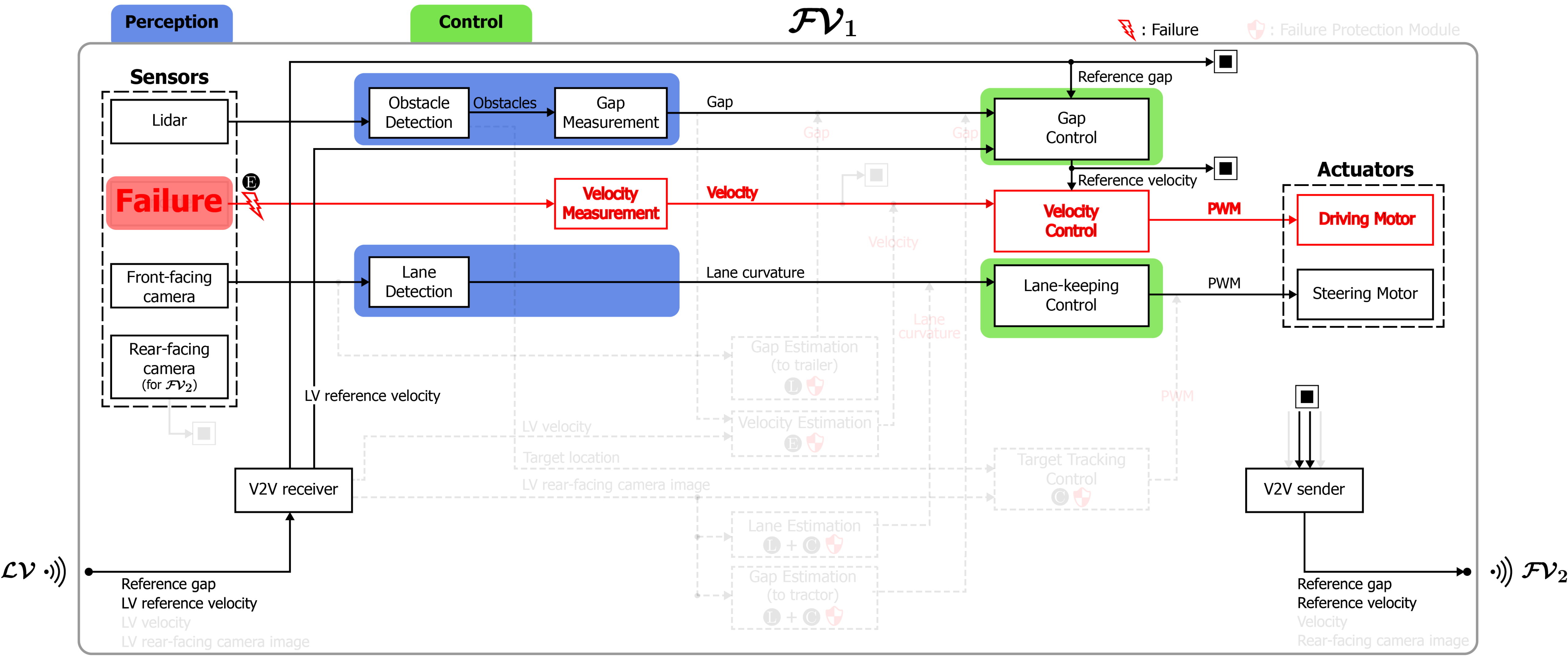
System Architecture (Normal)



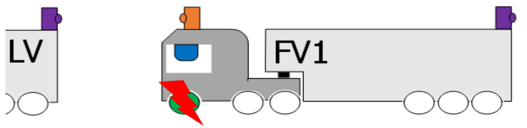
Normal Situation



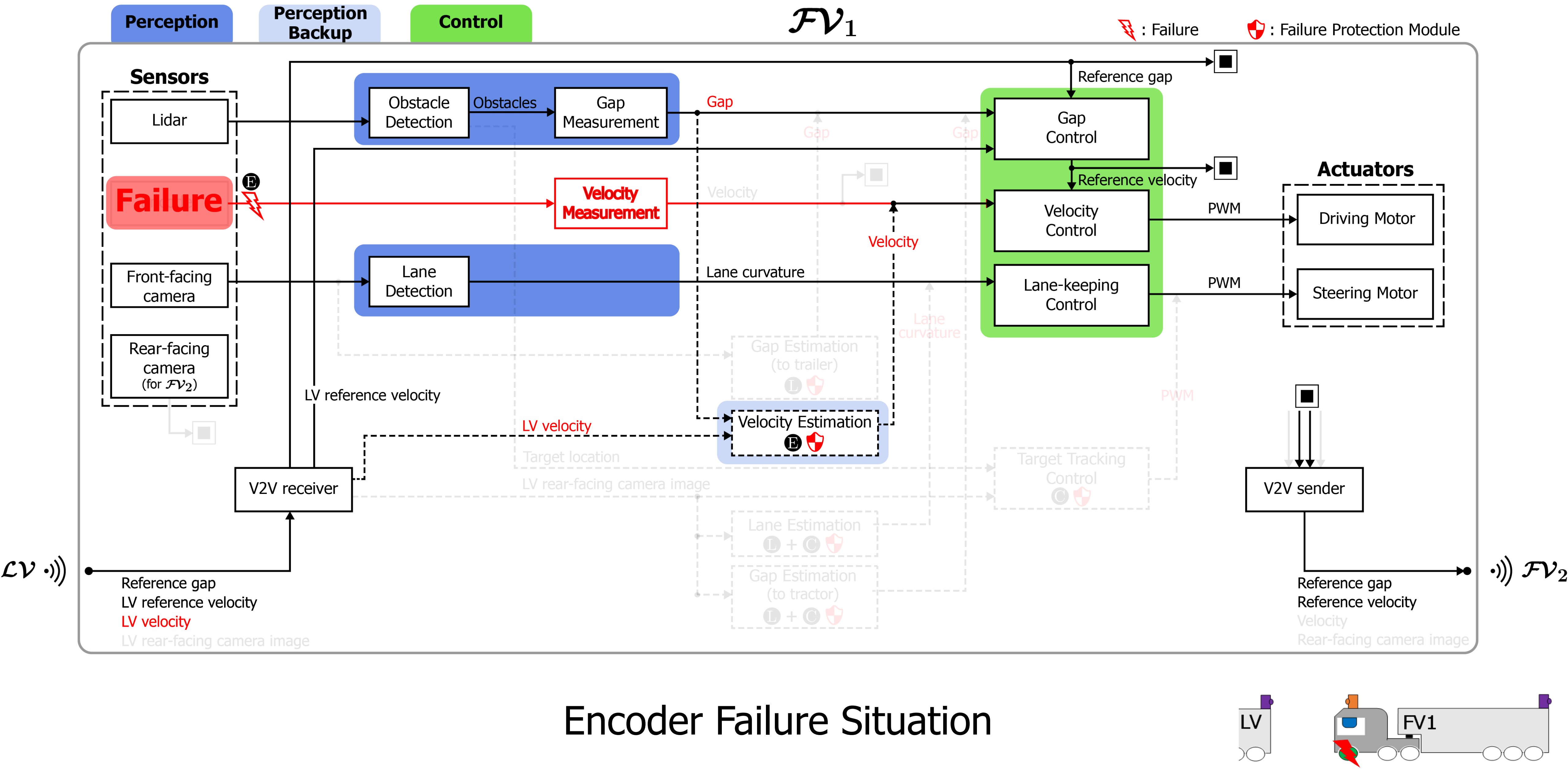
System Architecture (Ⓔ)



Encoder Failure Situation



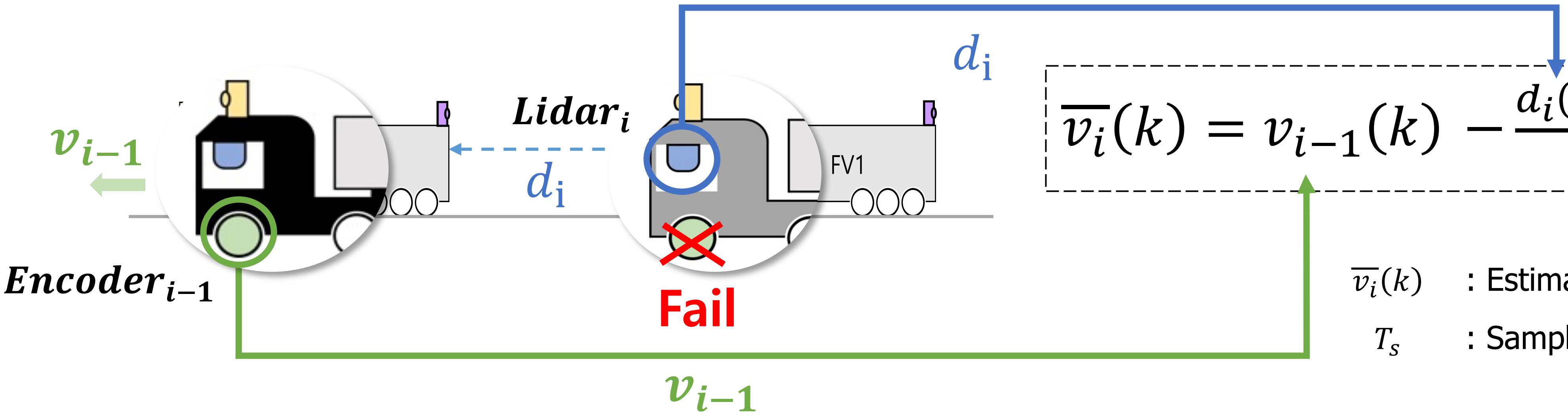
System Architecture (E)



Encoder Failure Situation

Velocity Estimation for Ⓔ

- Estimate velocity in case of encoder failure
  - ① Predict velocity through gap distance and velocity of preceding truck



$$\overline{v_i}(k) = v_{i-1}(k) - \frac{d_i(k) - d_i(k-1)}{T_s}$$

$\overline{v_i}(k)$: Estimated velocity

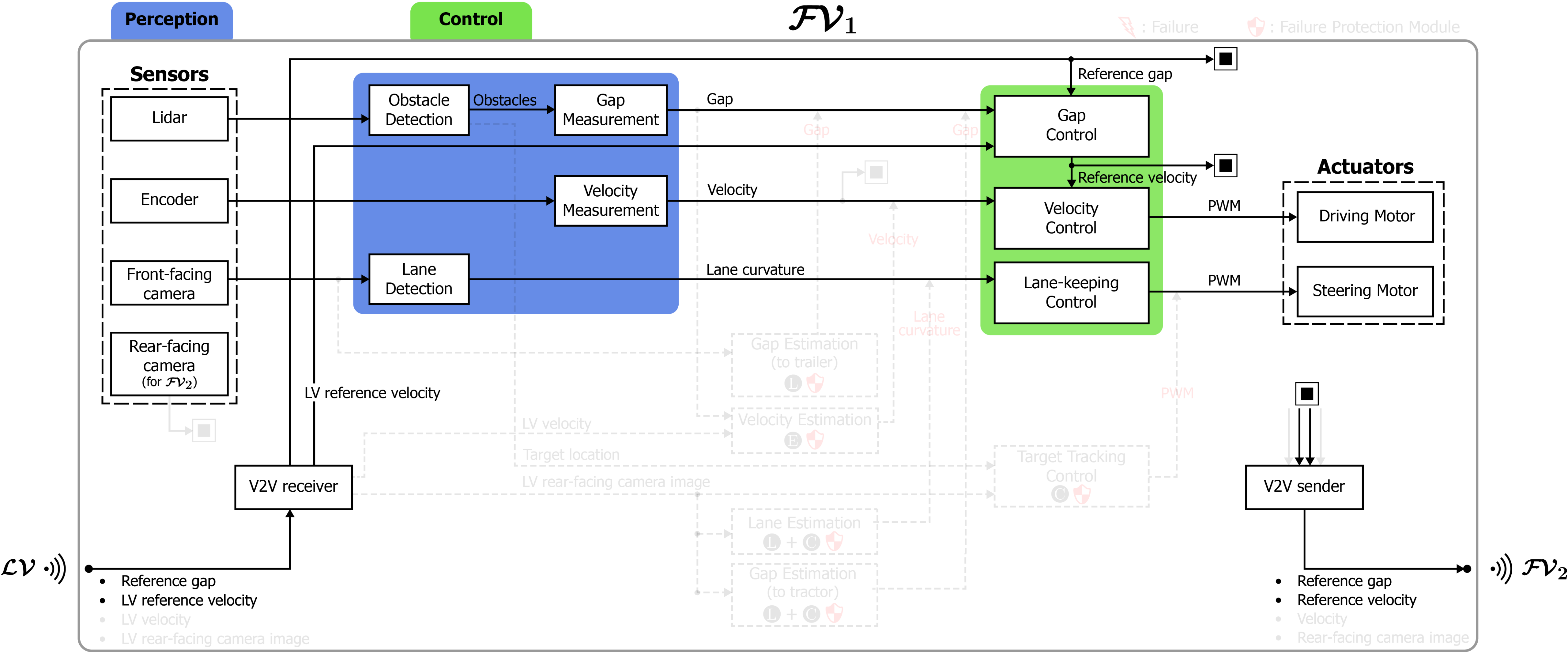
T_s : Sampling time

  - ② Remove noise by applying the first low-pass filter to the predicted velocity

$$\hat{v}_i(k) = \begin{cases} \dfrac{(\tau \times \hat{v}_i(k-1)) + (T_s \times \overline{v}_i(k))}{\tau + T_s}, & if\ k > 0 \\ \overline{v}_i(k), & if\ k = 0 \end{cases}$$

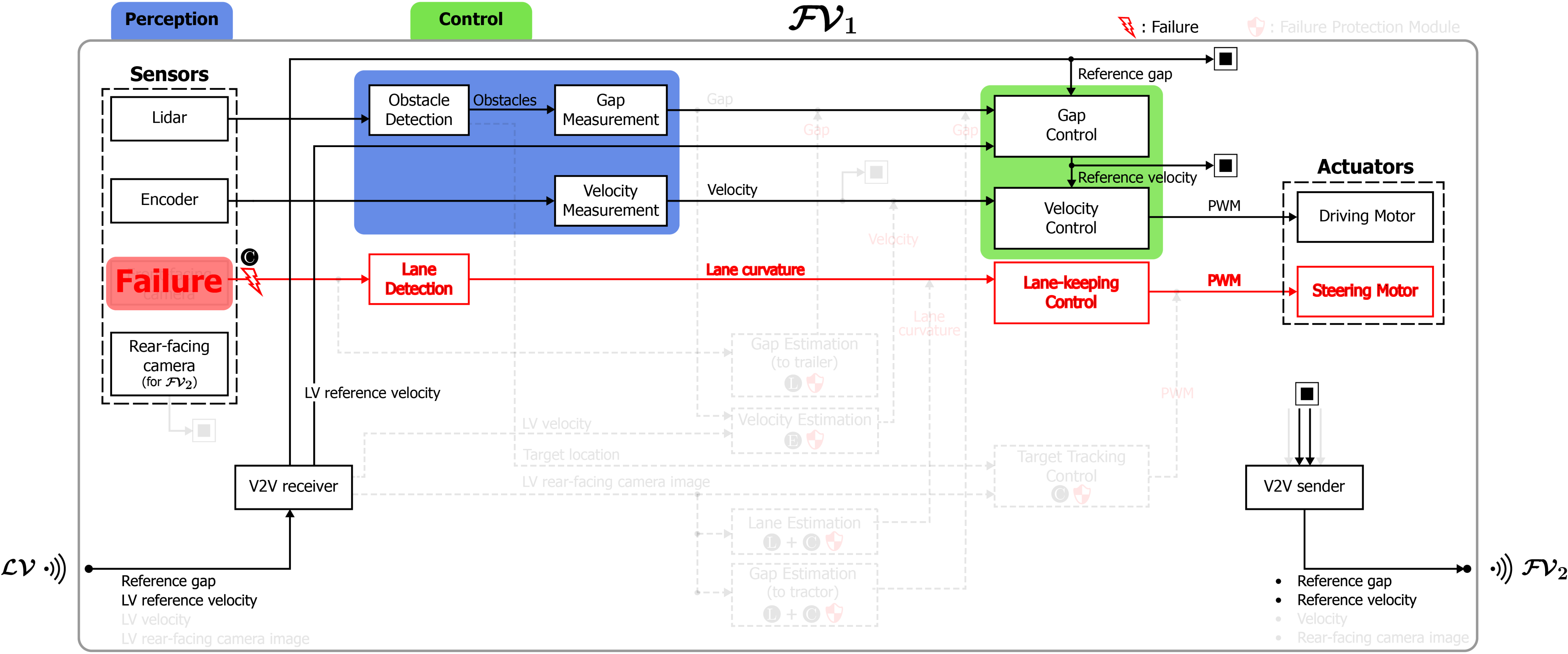
$\hat{v}_i(k)$: Filtered estimated velocity

System Architecture (Normal)

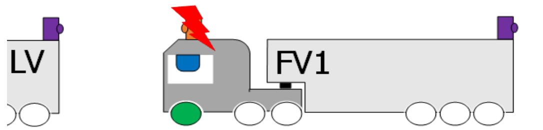


Normal Situation

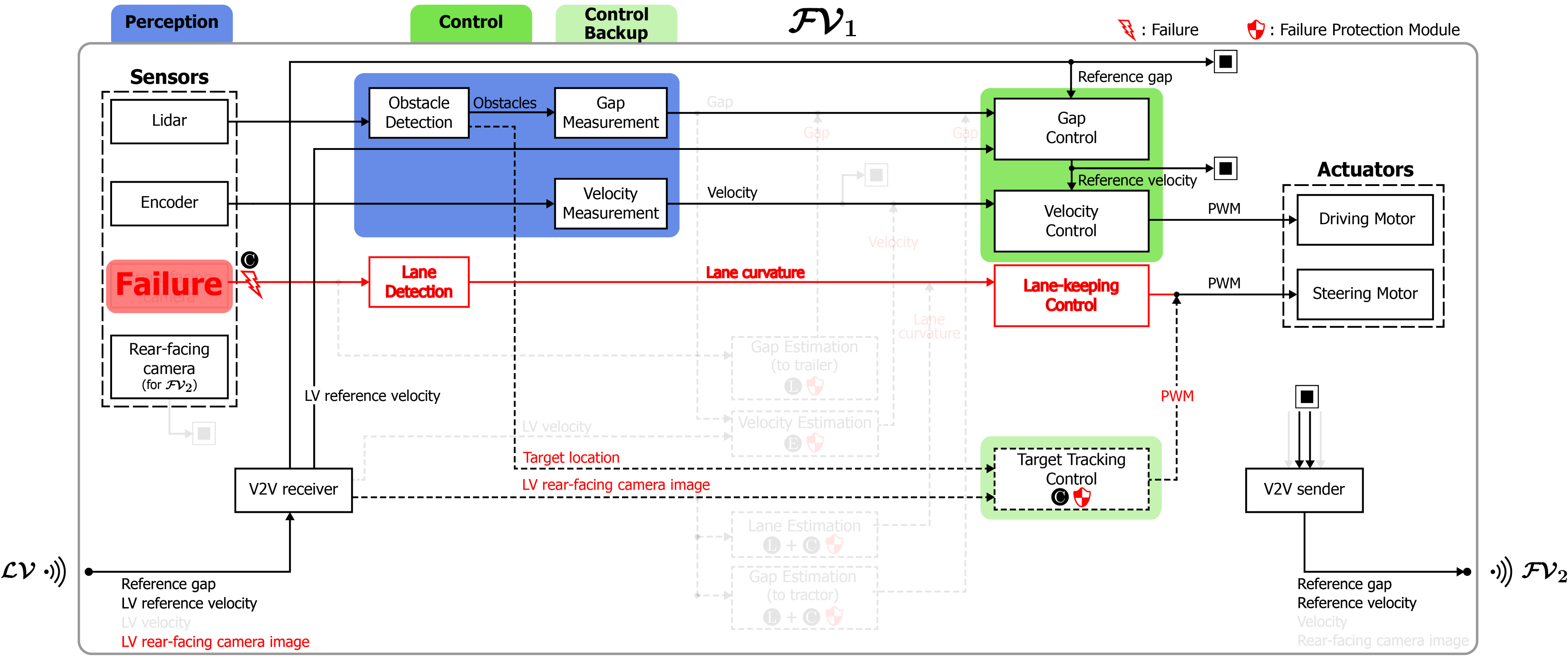
System Architecture (Ⓒ)



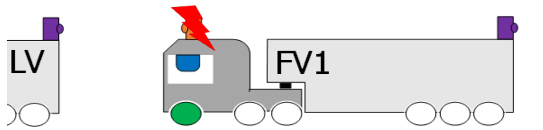
Camera Failure Situation



System Architecture (Ⓒ)

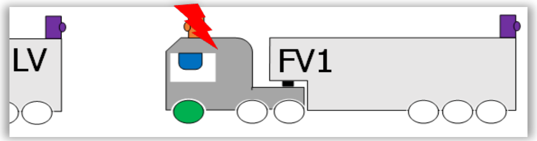


Camera Failure Situation



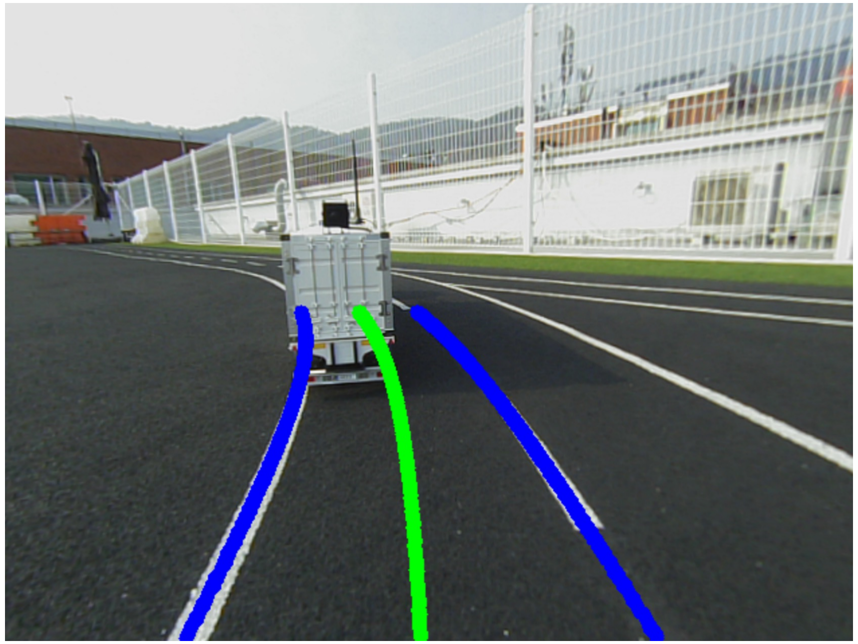
Target Tracking Control for ©

- Control mode change from lane-keeping to target tracking

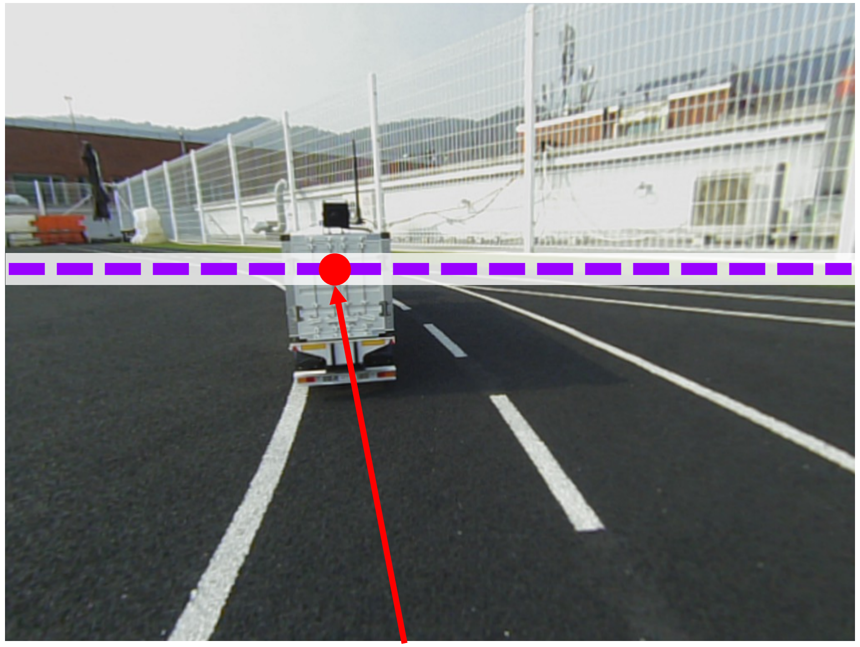


Camera-based

Lidar-based



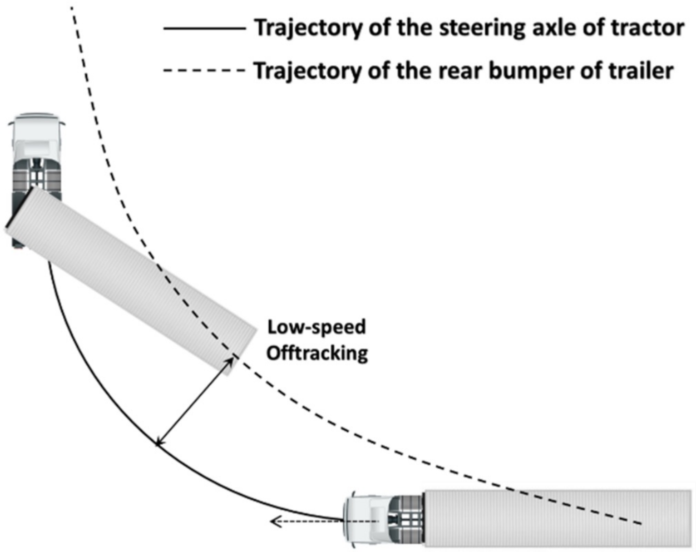
Lane-keeping control



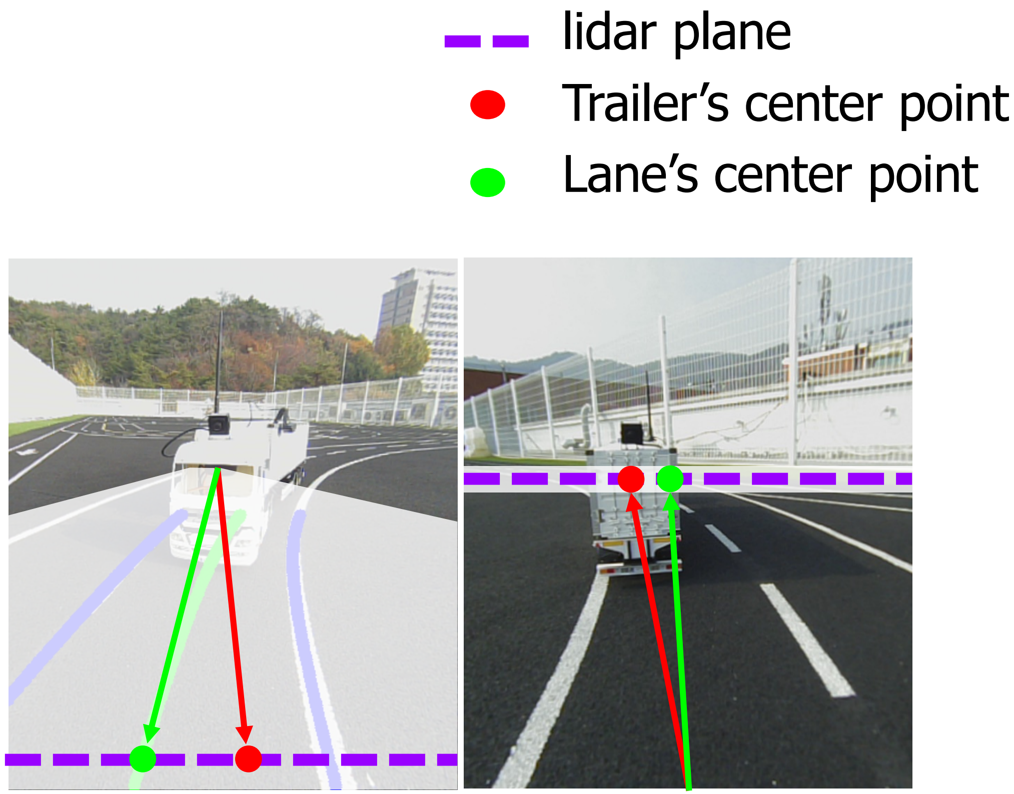
Target tracking control

Target Tracking Control for ©

- Issue: off-tracking phenomenon of semi-trailer trucks



Off-tracking phenomenon of semi-trailer truck



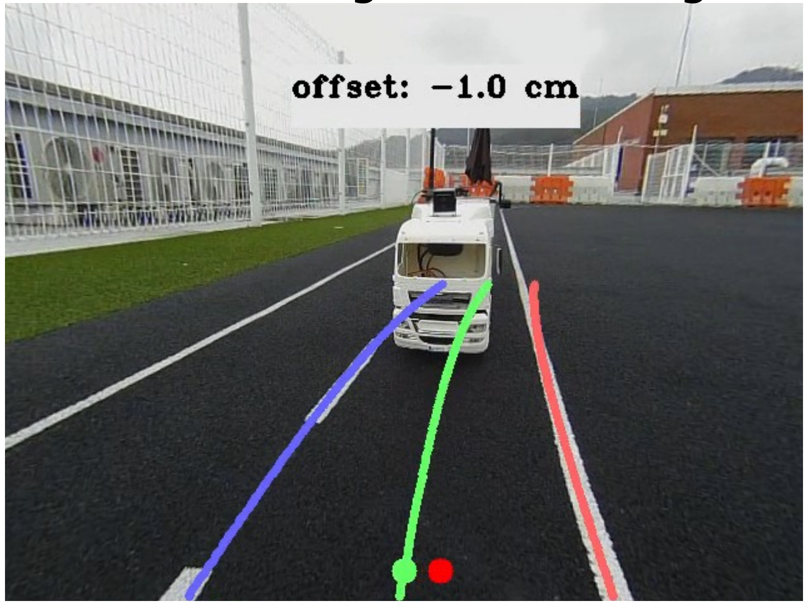
Target tracking control considering off-tracking problem

Source: Y. Lee, T. Ahn, C. Lee, S. Kim, and K. Park, "A novel path planning algorithm for truck platooning using V2V communication," Sensors, vol. 20, no. 24, p. 7022, Dec. 2020.

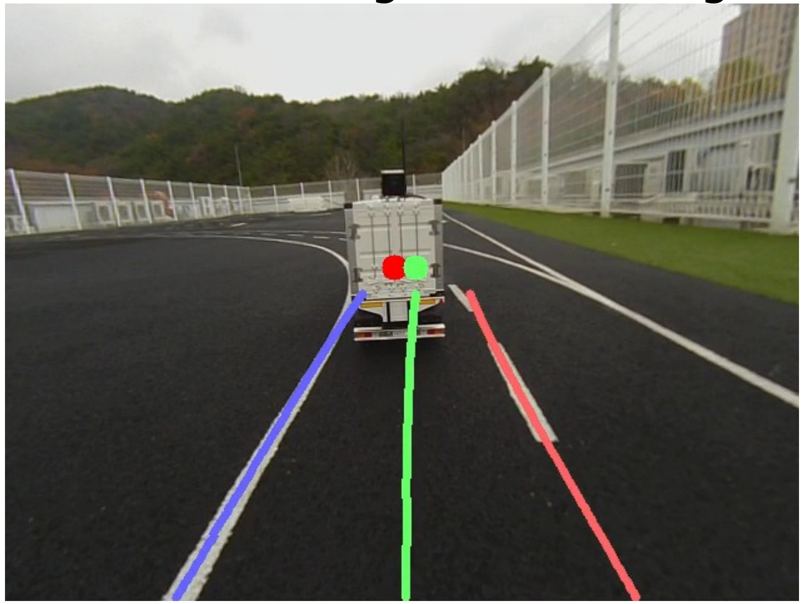
Target Tracking Control for ©

- Solution: offset correction by utilizing LV's rear-facing camera
  1. Reverse lane detection from LV's rear-facing camera
  2. Offset calculation between trailer's center and lane's center
  3. Following the lane's center (instead of trailer's center) by applying the offset

LV rear-facing camera's image

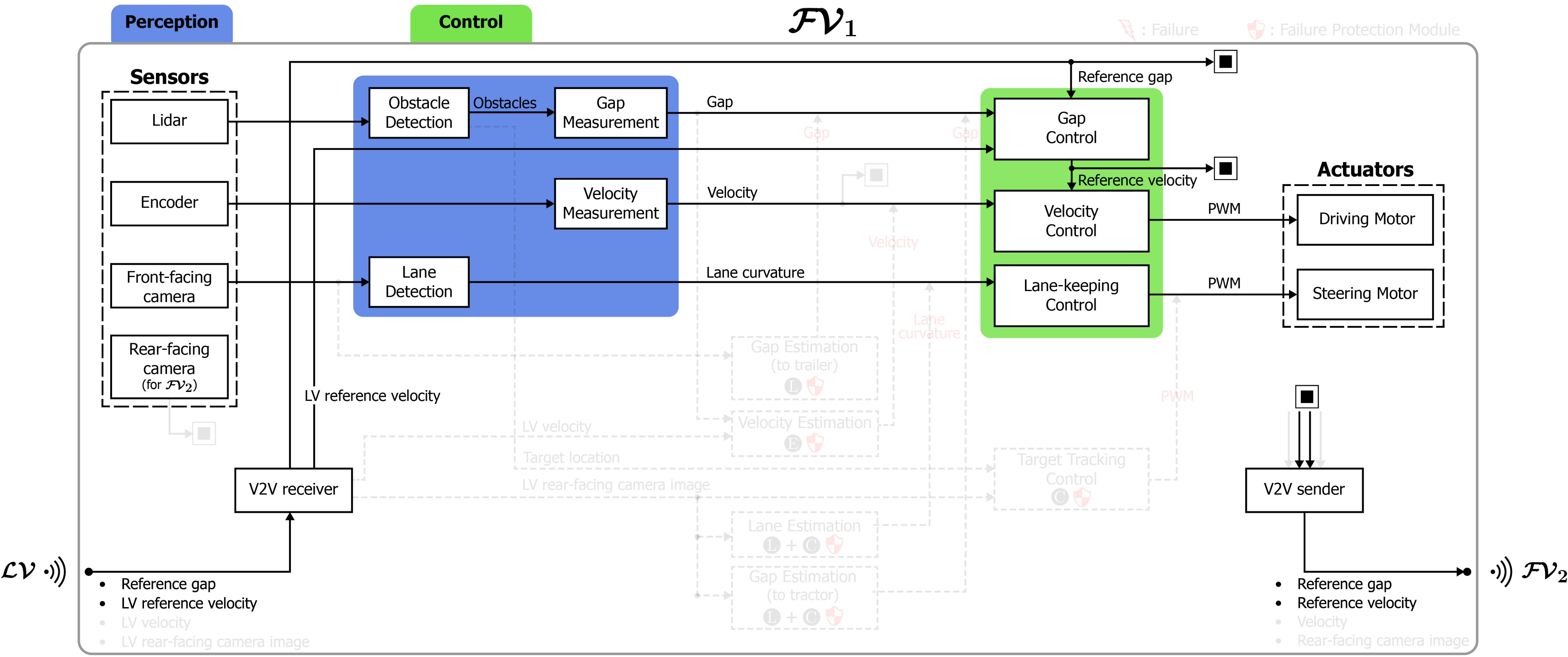


FV1 front-facing camera's image

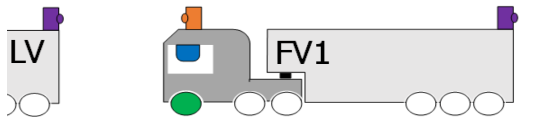


- Trailer's center point
- Lane's center point

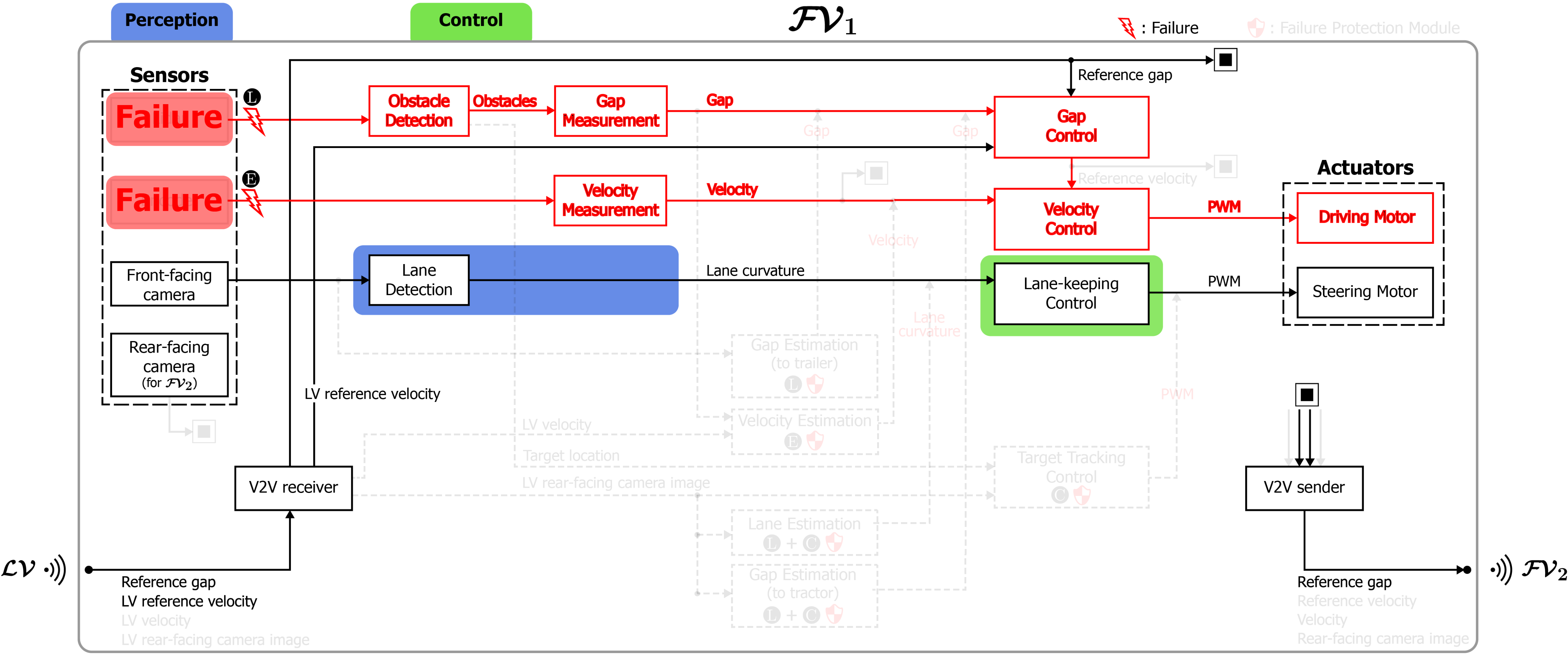
System Architecture (Normal)



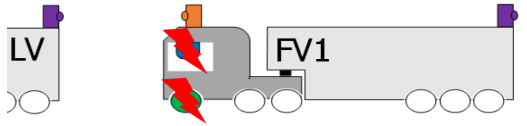
Normal Situation



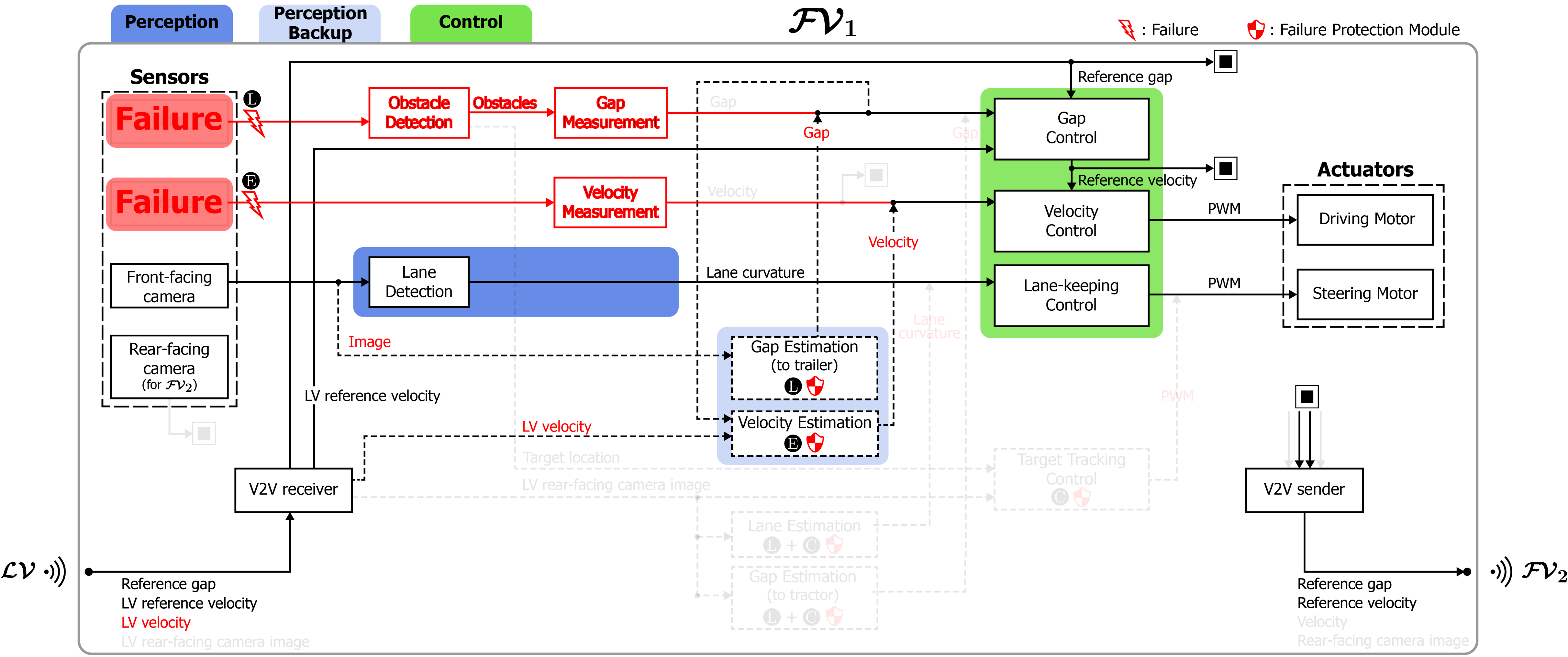
System Architecture (Ⓛ + Ⓔ)



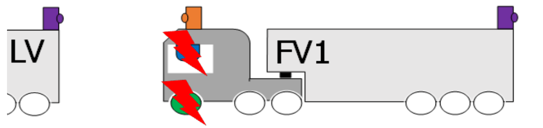
Lidar & Encoder Failure Situation



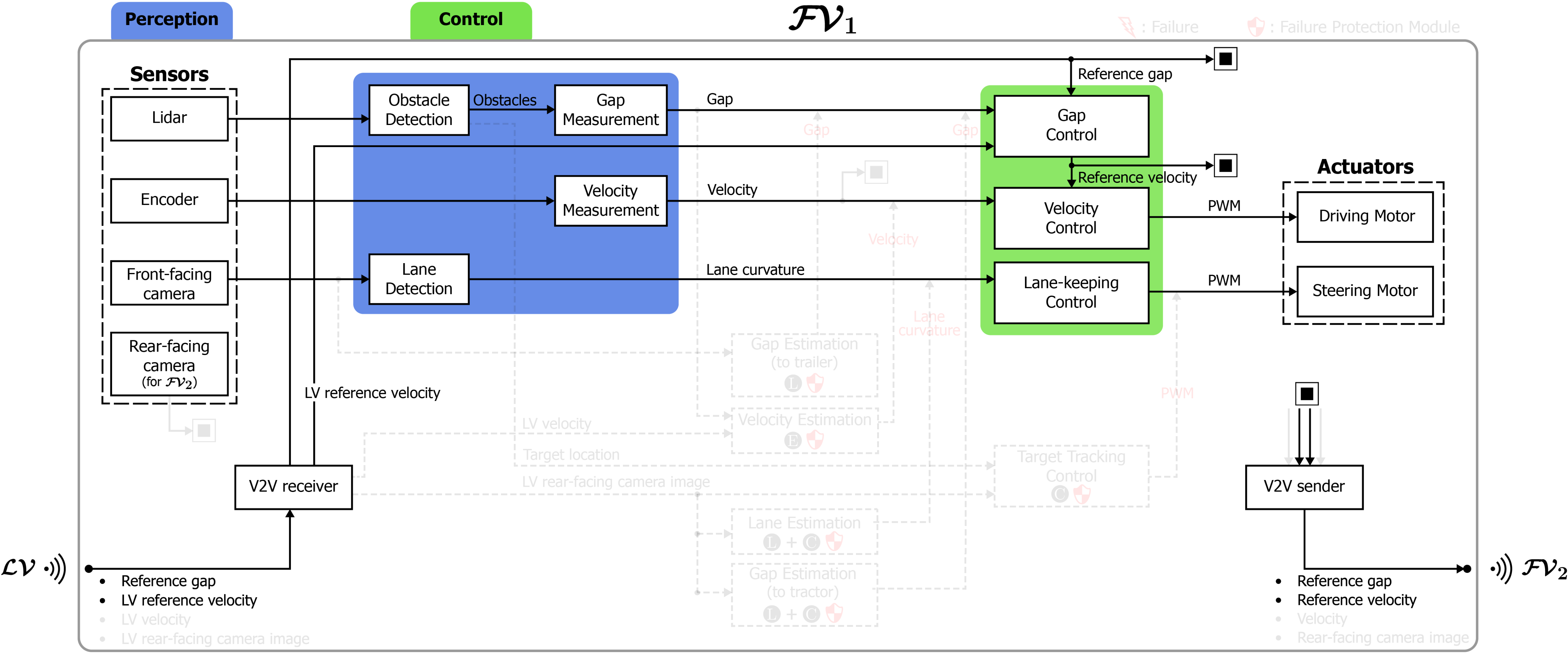
System Architecture (Ⓛ + Ⓔ)



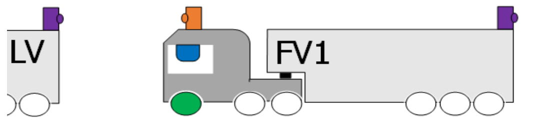
Lidar & Encoder Failure Situation



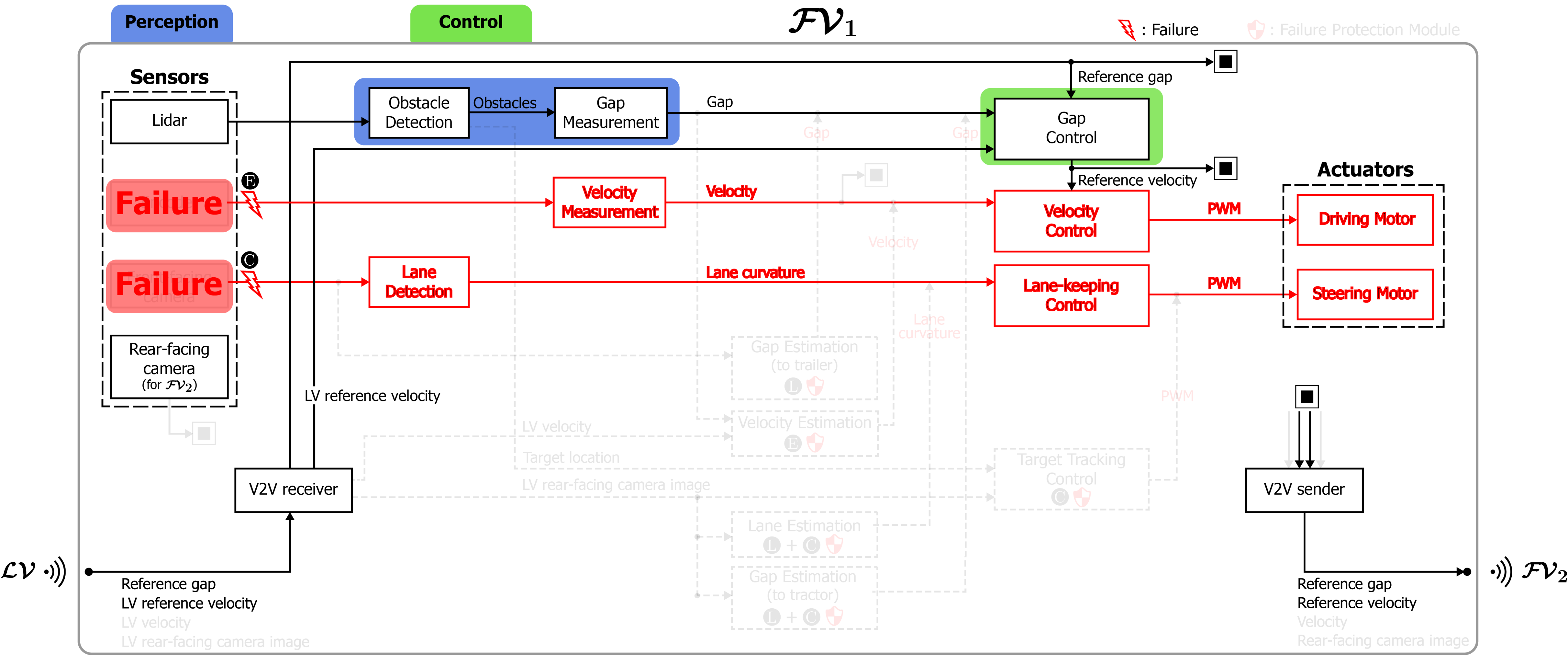
System Architecture (Normal)



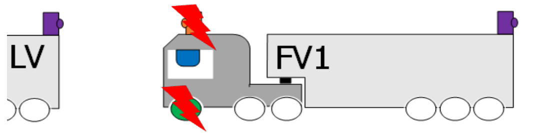
Normal Situation



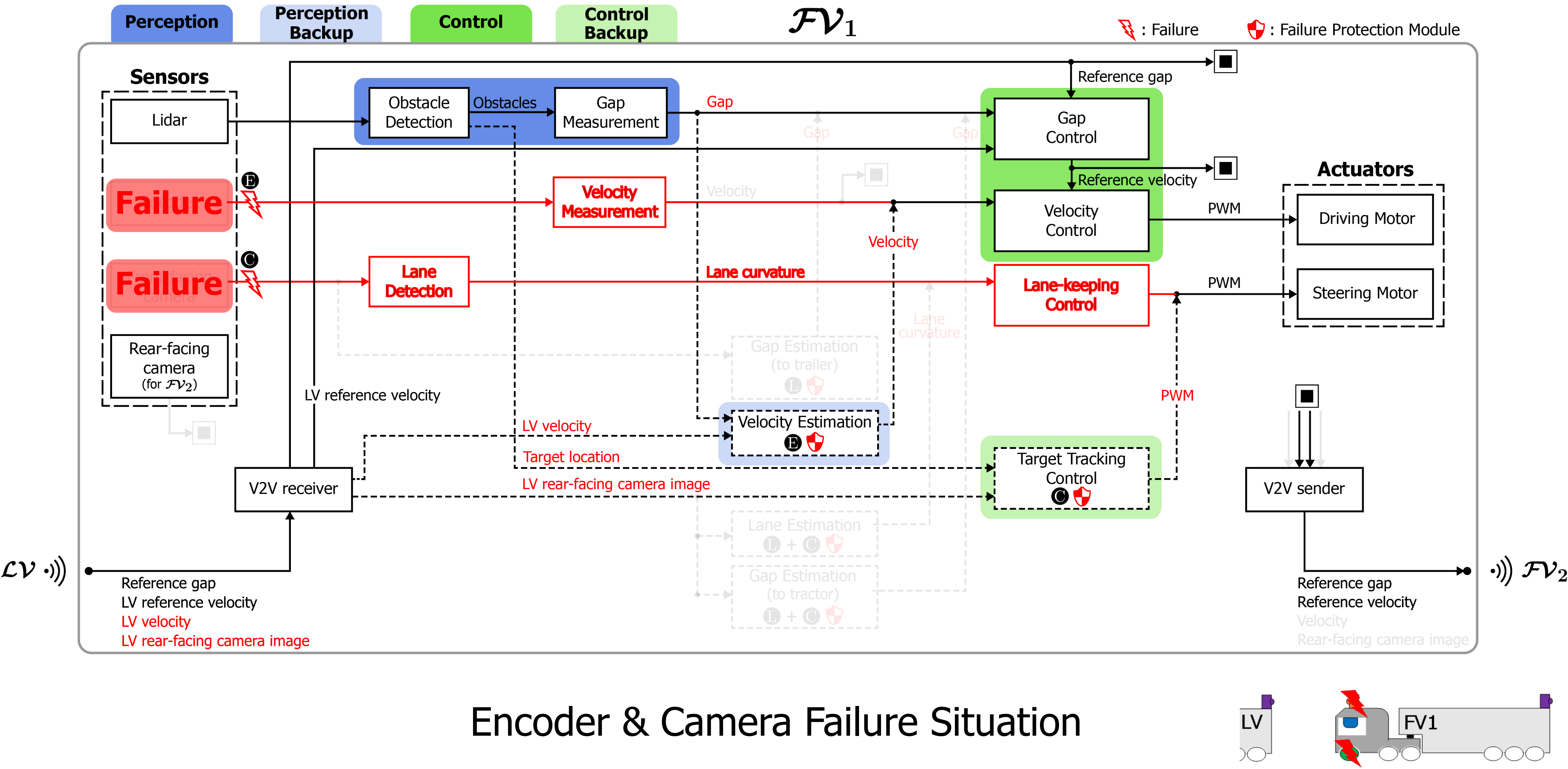
System Architecture (Ⓔ + Ⓒ)



Encoder & Camera Failure Situation

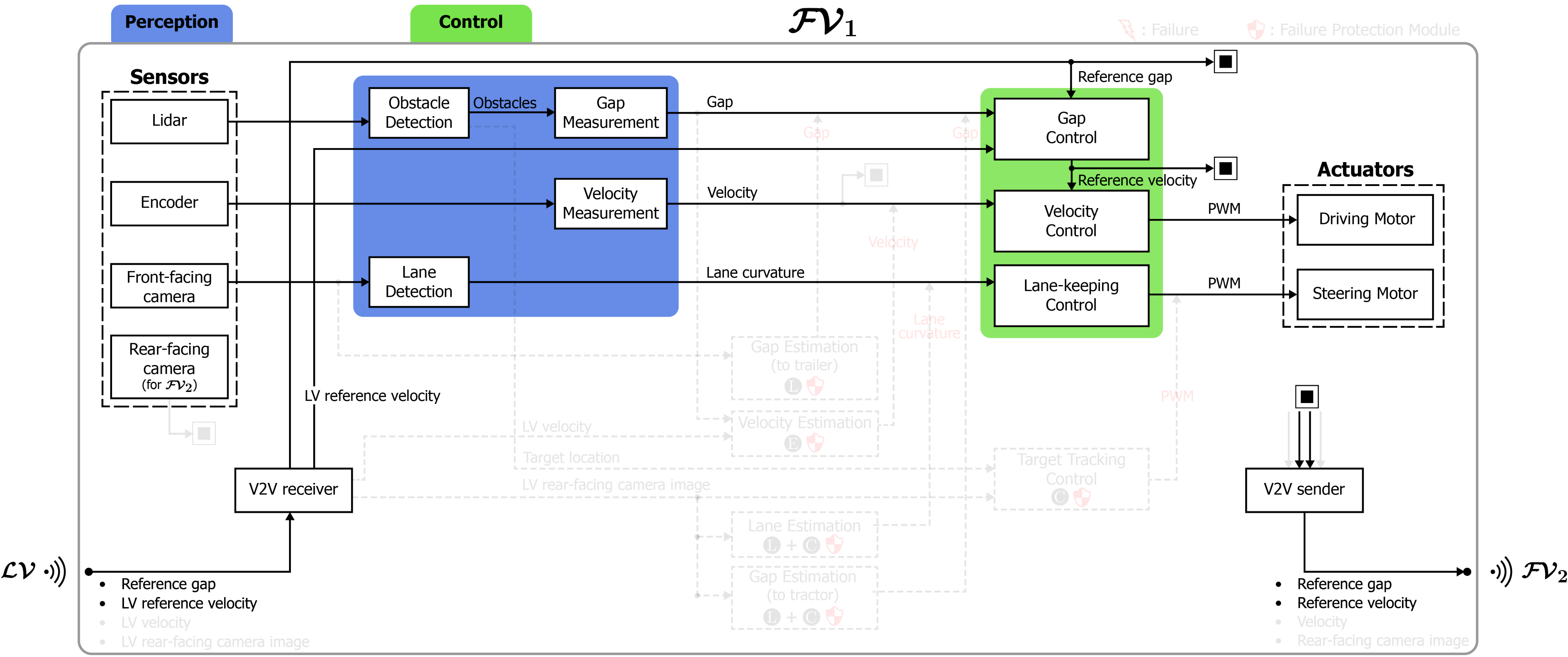


System Architecture (Ⓔ + Ⓒ)

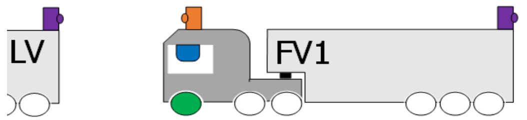


Encoder & Camera Failure Situation

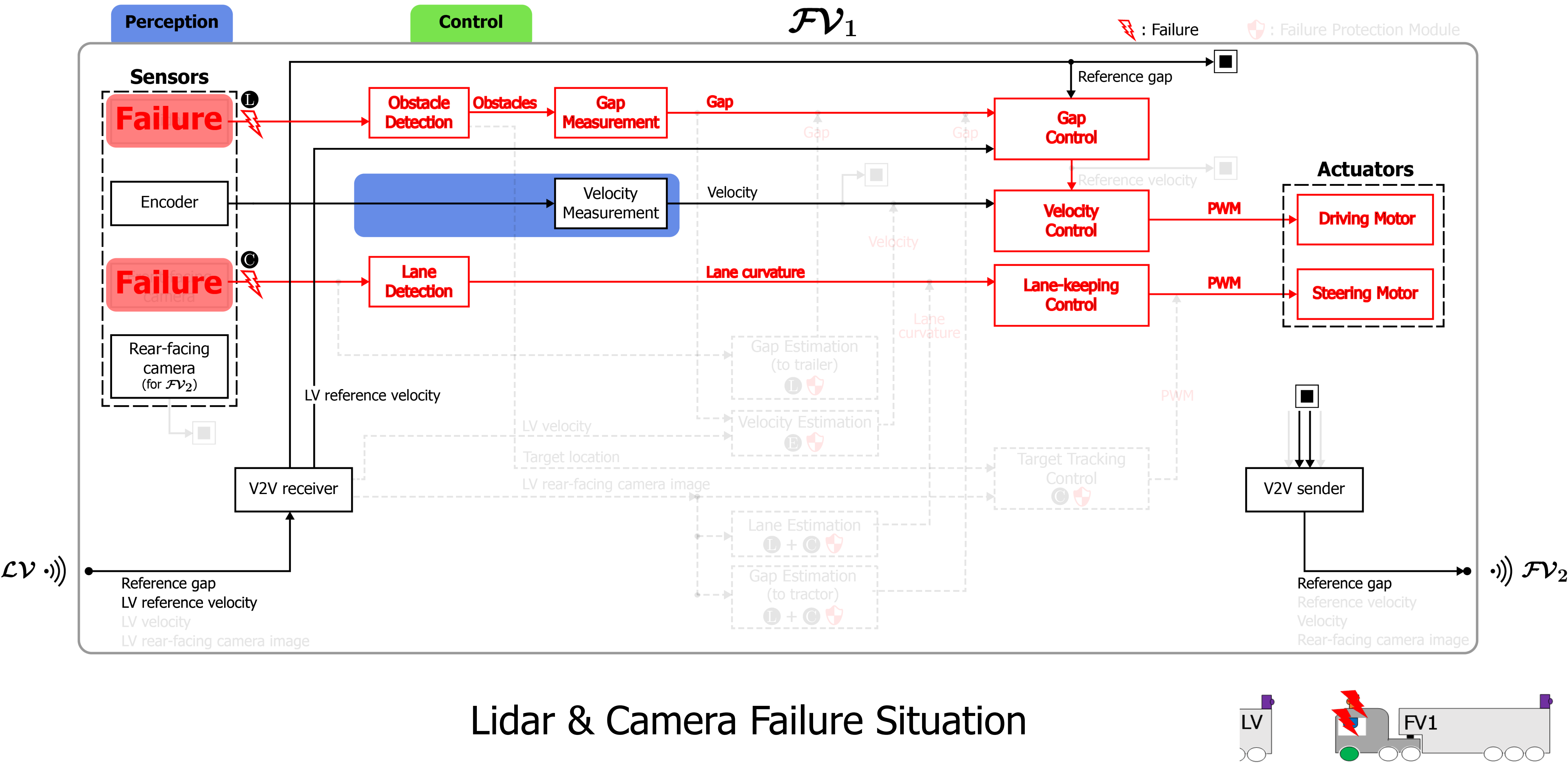
System Architecture (Normal)



Normal Situation

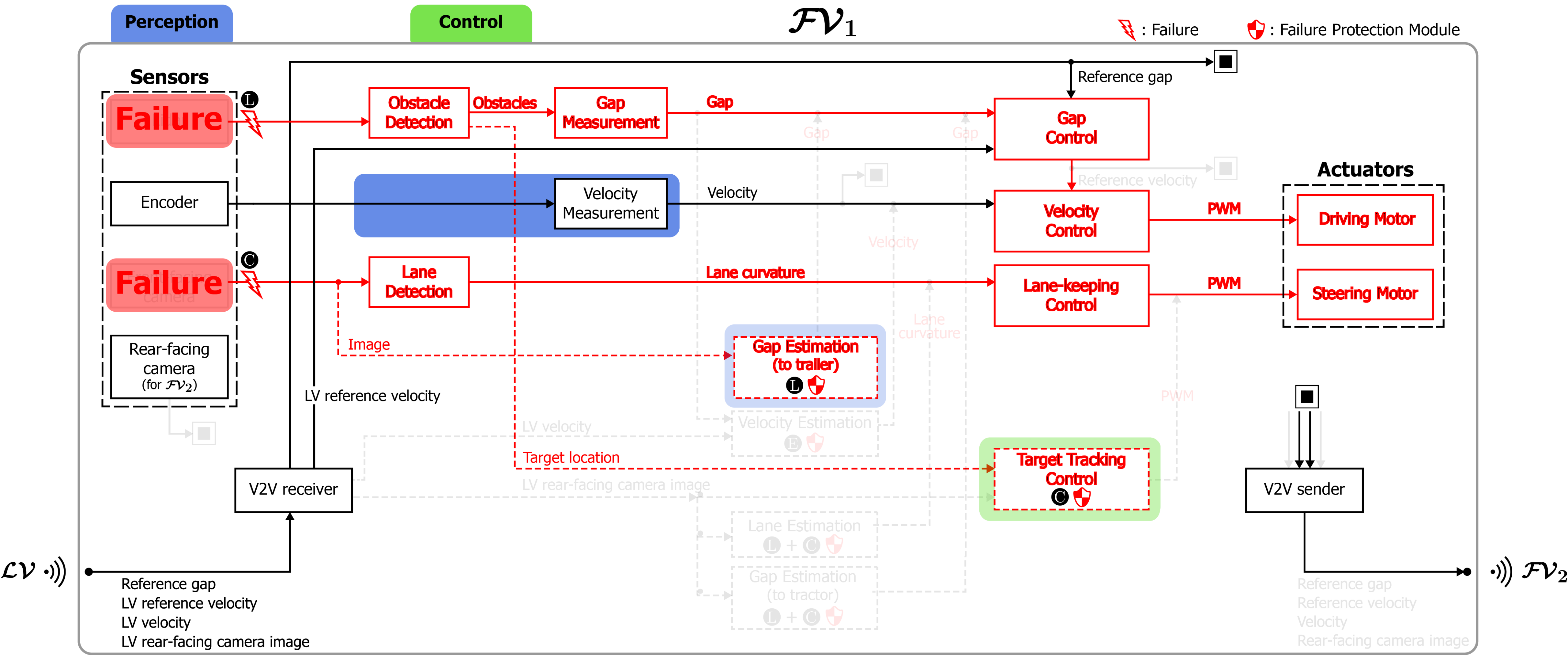


System Architecture (Ⓛ + Ⓒ)

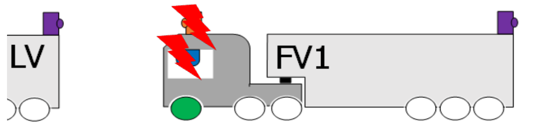


Lidar & Camera Failure Situation

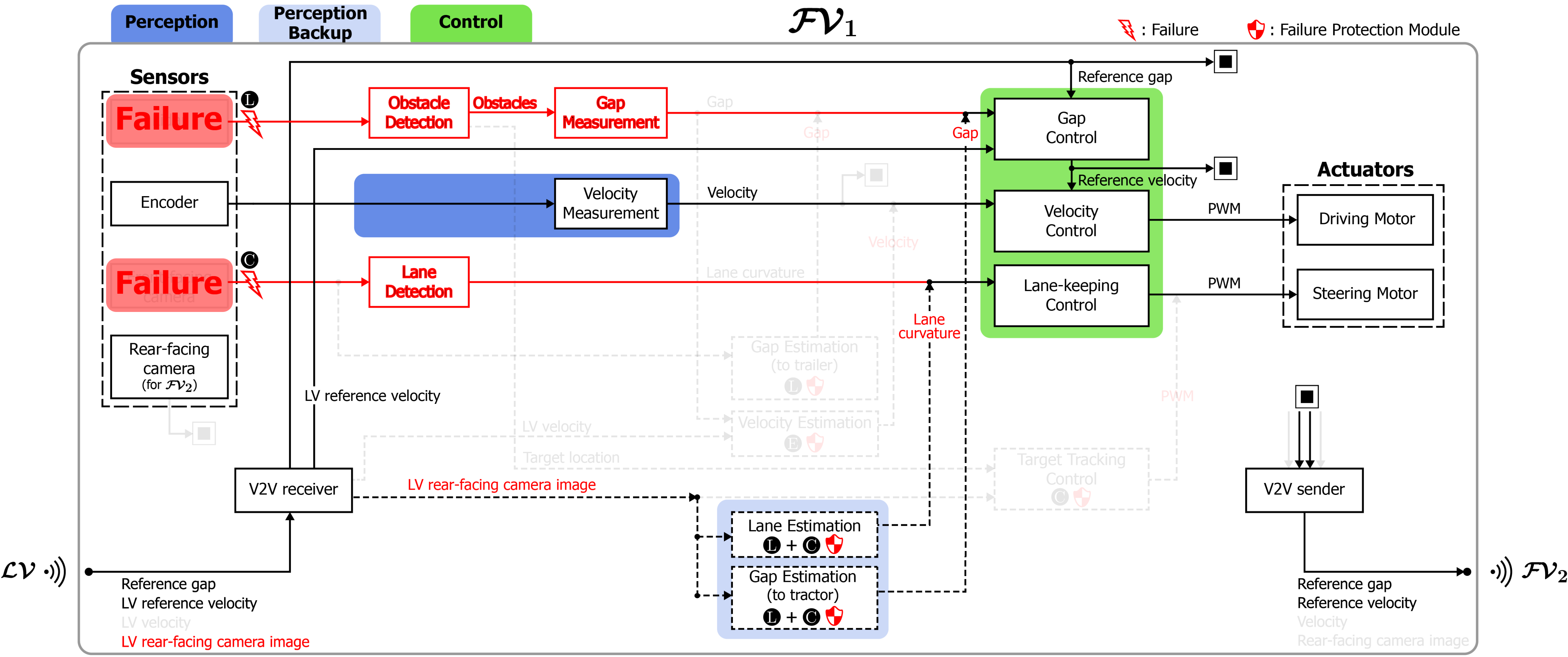
System Architecture (Ⓛ + Ⓒ)



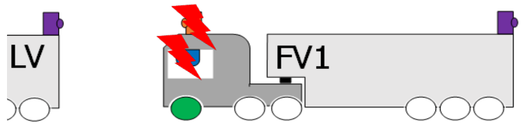
Lidar & Camera Failure Situation



System Architecture (Ⓛ + Ⓒ)



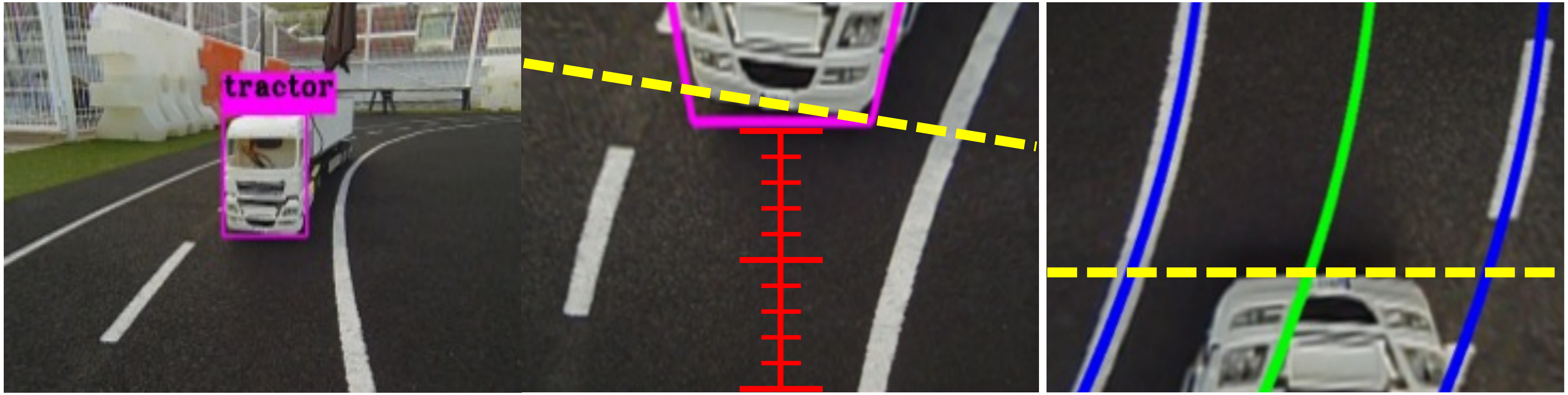
Lidar & Camera Failure Situation



Lane & Gap Estimation for Ⓛ + Ⓒ

- Lane estimation
  ① Calculate the heading angle of the ego tractor
  ② Align and detection lanes

- Gap estimation (to tractor)
  ① Calculate the distance (same as the gap estimation method in Ⓛ)

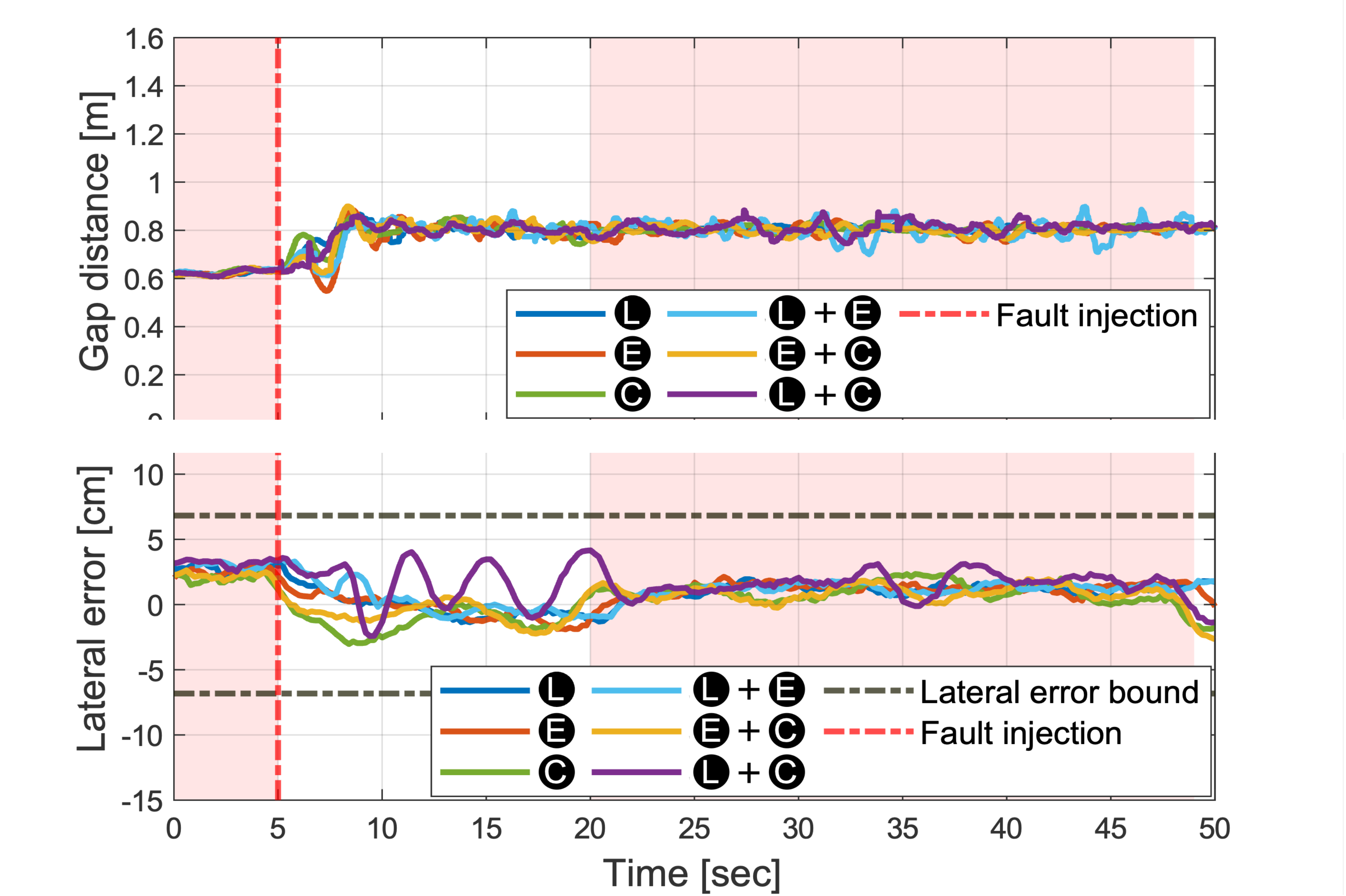


Lane & Gap Estimation for Ⓛ + Ⓒ

- Lane estimation
  ① Calculate the heading angle of the ego tractor
  ② Align and detection lanes

- Gap estimation (to tractor)
  ① Calculate the distance (same as the gap estimation method in Ⓛ)

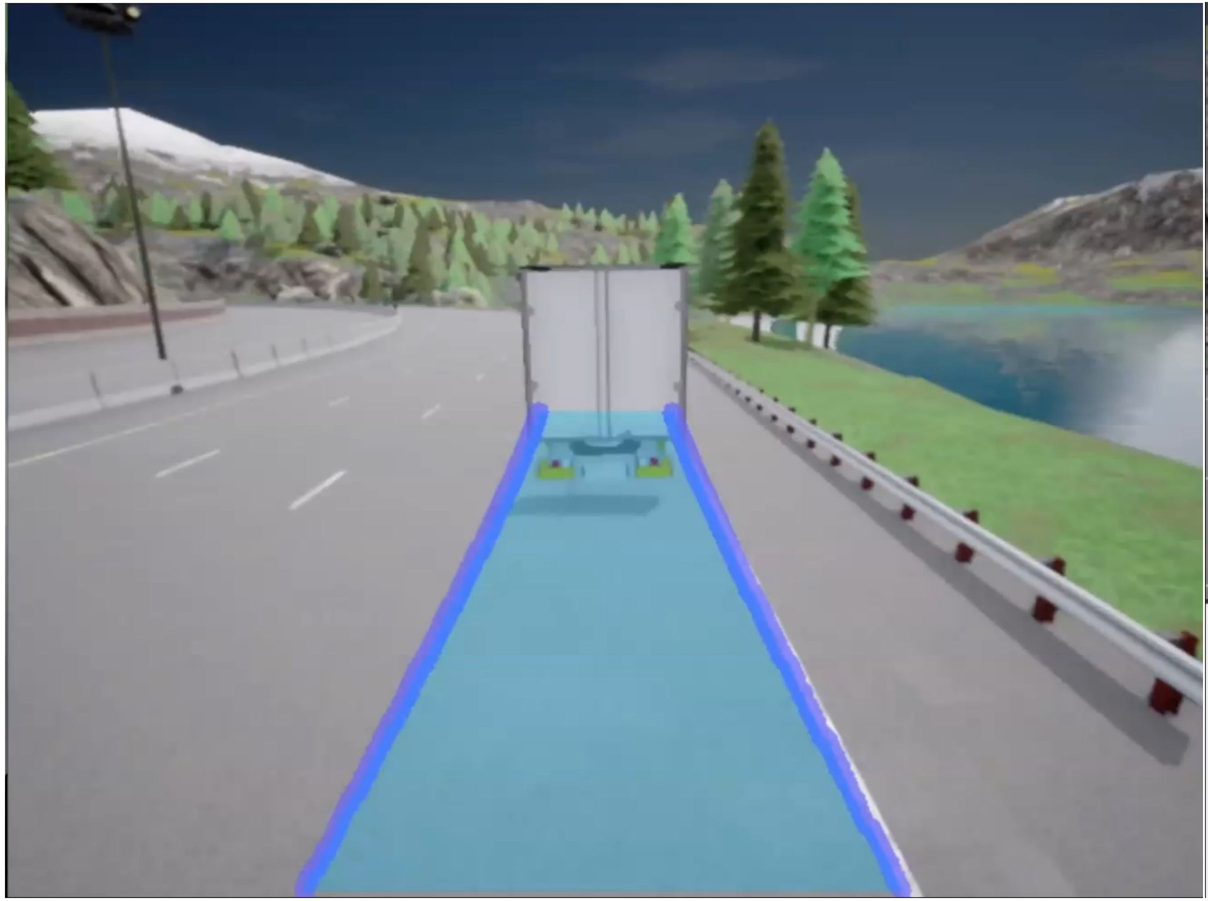
Evaluation: Six Failure Scenarios

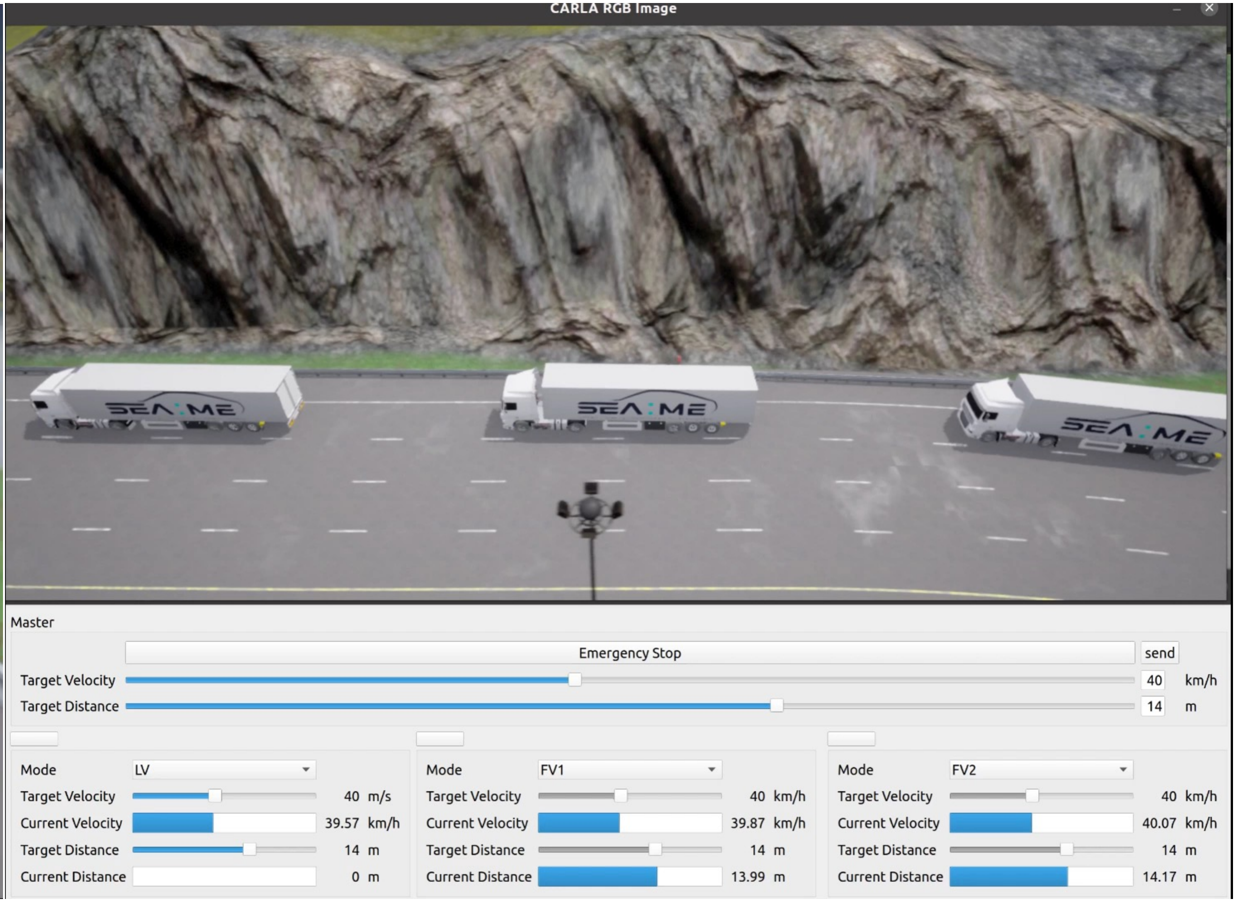


Conclusion

- Presented failure-resilient truck platooning architecture without sensor redundancy using the unique driving pattern of platooning trucks
- Six failure scenarios are considered including three single sensor failures and three dual sensor failures
- Mitigation methods for each scenario is actually implemented and evaluated on our 1/14 scale truck testbed

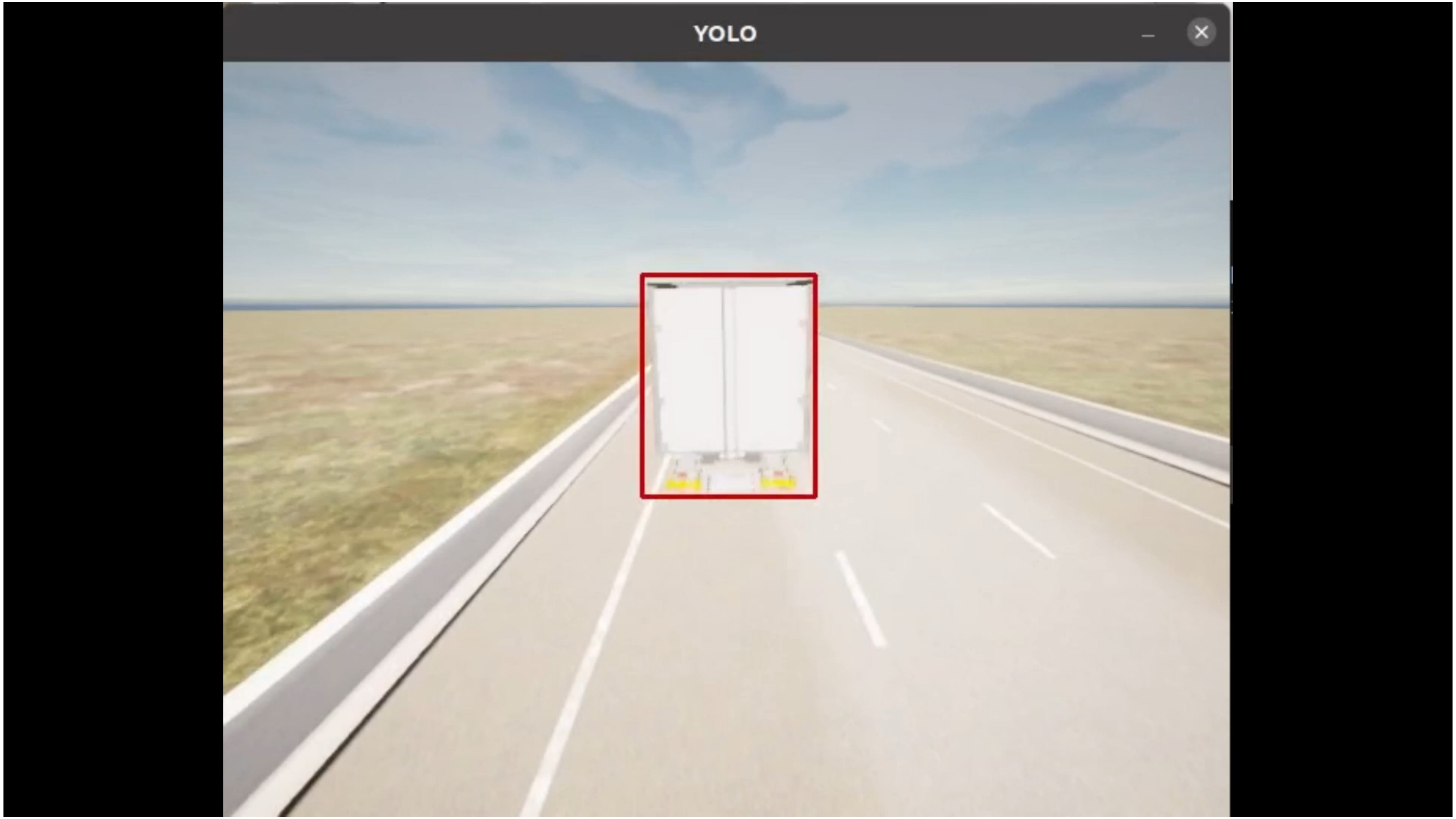
Virtual Truck Platooning





https://github.com/AveesLab/carla-virtual-platoon

Emergency Braking Scenario



Cut-in and Cut-out Scenario

Emergency Lane Change Scenario

Questions

Welcome to RTCSA 2024 in Sokcho, Korea