A Practical Degradation Model for Mixed Criticality Systems

Vijaya Kumar Sundar, **Arvind Easwaran**

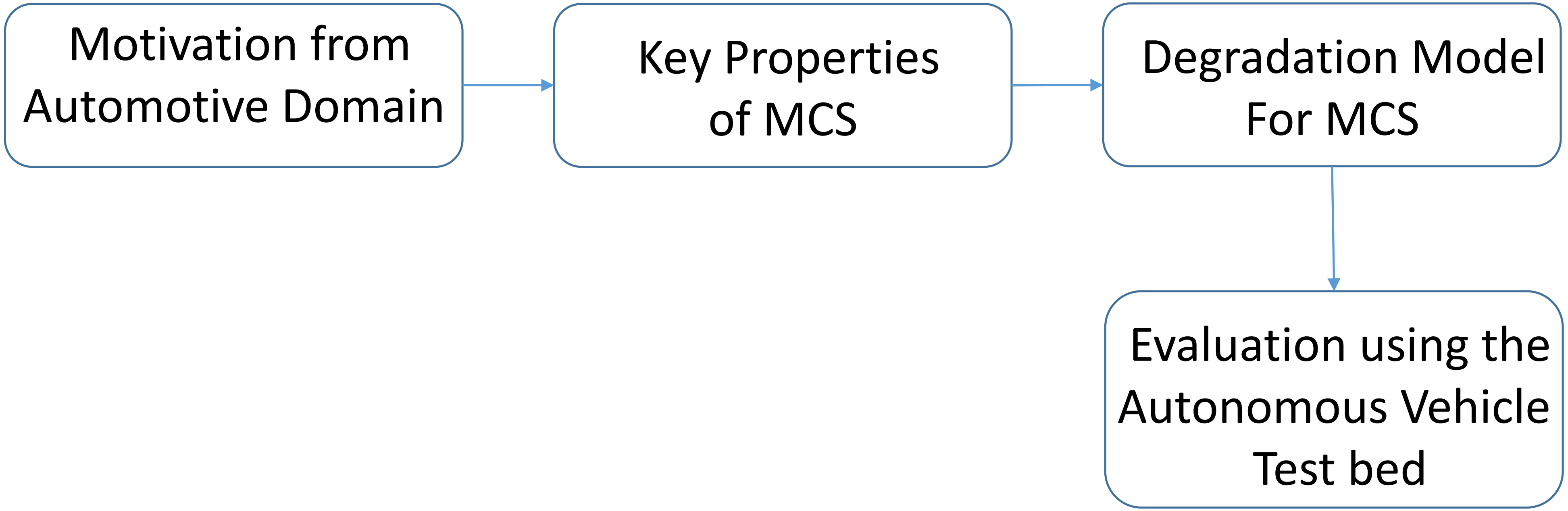
Nanyang Technological University (NTU), Singapore

May 8, 2019

NANYANG
TECHNOLOGICAL
UNIVERSITY

Research Outline

Research Objective: A new degradation model for Mixed Criticality System



NANYANG TECHNOLOGICAL UNIVERSITY

Current Trends in Automotive Systems

Current Trends in Automotive Systems

Trend 1 : Automotive is shifting from mechanical centric to electronic and software centric domain.

Current Trends in Automotive Systems

Trend 1 : Automotive is shifting from mechanical centric to electronic and software centric domain.

- ➢need for Autonomous Driving
- ➢increase in software intensive Driver Assistance Systems for safety and comfort.

NANYANG TECHNOLOGICAL UNIVERSITY

Current Trends in Automotive Systems

Trend 1 : Automotive is shifting from mechanical centric to electronic and software centric domain.

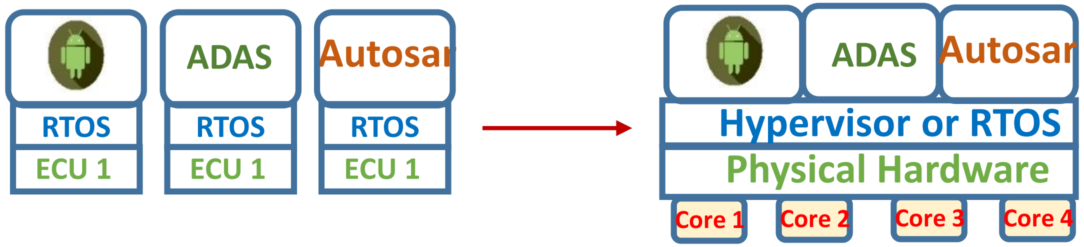
- ➢need for Autonomous Driving
- ➢increase in software intensive Driver Assistance Systems for safety and comfort.

Challenge : Increase in ECUs → Increase in harness weight, cost and reduced fuel efficiency.

Trend Towards ECU Consolidation

- A dedicated ECU for each functionality is not a sustainable solution!
  - Increase in demand for safety and comfort features
  - Increase in harness weight, cost and network complexity

- Car manufacturers are moving towards ECU consolidation
  - Shared sensors and actuators between applications
  - Reduced communication latency
  - Applications having varied importance/criticality execute on a single hardware platform

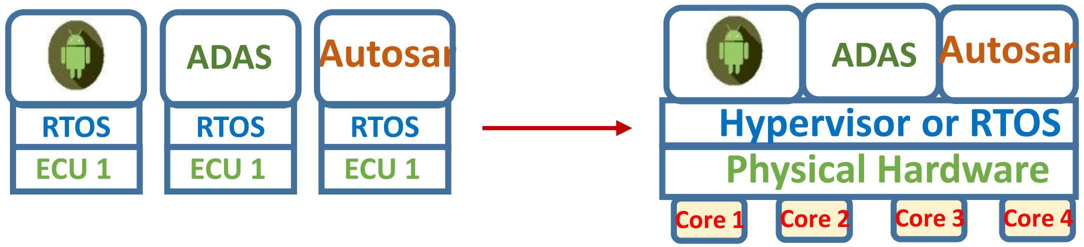


NANYANG TECHNOLOGICAL UNIVERSITY

Trend Towards ECU Consolidation

- A dedicated ECU for each functionality is not a sustainable solution!
  - Increase in demand for safety and comfort features
  - Increase in harness weight, cost and network complexity

- Car manufacturers are moving towards ECU consolidation
  - Shared sensors and actuators between applications
  - Reduced communication latency
  - Applications having varied importance/criticality execute on a single hardware platform



Innovation in hardware and software platforms have made automotive, a Mixed Criticality System

Mixed Criticality Systems (MCS)

- A system with multiple applications that are “certified” to different levels of criticality and share hardware resources
  - Example, Antilock Braking (ABS), a highly critical application sharing hardware with Parking Assist, a relatively less critical application

Mixed Criticality Systems (MCS)

- A system with multiple applications that are "certified" to different levels of criticality and share hardware resources
  - Example, Antilock Braking (ABS), a highly critical application sharing hardware with Parking Assist, a relatively less critical application

- MCS is not new in the safety-criticality industry
  - Integrated Modular Avionics (IMA) was commercially introduced in the 1990s
  - AUTOSAR has been around for about 10 years now

Mixed Criticality Systems (MCS)

- A system with multiple applications that are "certified" to different levels of criticality and share hardware resources
  - Example, Antilock Braking (ABS), a highly critical application sharing hardware with Parking Assist, a relatively less critical application

- MCS is not new in the safety-criticality industry
  - Integrated Modular Avionics (IMA) was commercially introduced in the 1990s
  - AUTOSAR has been around for about 10 years now

- Important challenges for computing platforms in MCS
  - Resource allocation strategy ensuring safety with acceptable compromise on performance
  - Software architectures for enabling MCS

Resource Allocation in MCS

- How to ensure safety?
    - Guaranteed allocation for critical applications
    - Satisfaction of safety standards such as ISO26262

Resource Allocation in MCS

- How to ensure safety?
    - Guaranteed allocation for critical applications
    - Satisfaction of safety standards such as ISO26262

- How to efficiently utilize resources?
    - To ensure safety, utilization estimates are needed for applications
        - Example, Worst-Case Execution Time (WCET) estimates
    - Estimates are typically generous for critical applications
    - Leads to over-allocation and consequently wastage

Resource Allocation in MCS

- How to ensure safety?
  - Guaranteed allocation for critical applications
  - Satisfaction of safety standards such as ISO26262

- How to efficiently utilize resources?
  - To ensure safety, utilization estimates are needed for applications
    - Example, Worst-Case Execution Time (WCET) estimates
  - Estimates are typically generous for critical applications
  - Leads to over-allocation and consequently wastage

How to reconcile seemingly conflicting requirements of safety and efficiency?

ISO26262 Resource Allocation Requirements

ISO26262 Resource Allocation Requirements

If an ECU runs SW-Cs of different ASILs, ISO26262 provides two options

ASIL D

+

ASIL C

ECU with mixed ASILs

ISO26262 Resource Allocation Requirements

If an ECU runs SW-Cs of different ASILs, ISO26262 provides two options

ASIL D
+
ASIL C

ECU with mixed ASILs

Option 1 : Raise the ASIL

ASIL D
+
ASIL C → ASIL D

- For all SW-Cs, assign the highest ASIL among all coexisting SW-Cs
- Do this if evidence for **freedom from interference** is not feasible
- Increased certification cost, larger footprint

ISO26262 Resource Allocation Requirements

If an ECU runs SW-Cs of different ASILs, ISO26262 provides two options

ASIL D
+
ASIL C

ECU with mixed ASILs

Option 1 : Raise the ASIL

ASIL D
+
ASIL C → ASIL D

- For all SW-Cs, assign the highest ASIL among all coexisting SW-Cs
- Do this if evidence for **freedom from interference** is not feasible
- Increased certification cost, larger footprint

Option 2 : Allow co-existence

ASIL D
+
ASIL C

- Do this if evidence for **freedom from interference** is feasible

Freedom from Interference (as defined in ISO26262)

- Quote*

"Interference is the presence of cascading failures from a SW-C with no ASIL assigned, or a lower ASIL assigned, to a SW-C with a higher ASIL assigned leading to the violation of a safety requirement of the SW-C"

- Definition 1.49 in ISO26262*

"Freedom from interference is the absence of cascading failures between two or more SW-Cs that could lead to the violation of a safety requirement"

*Paraphrased (replaced elements and sub-elements with SW-Cs)

Motivation for Permitting Interference

- Resource utilization estimates are pessimistic to ensure safety
  - Hardware/Software complexity adds to the pessimism

Motivation for Permitting Interference

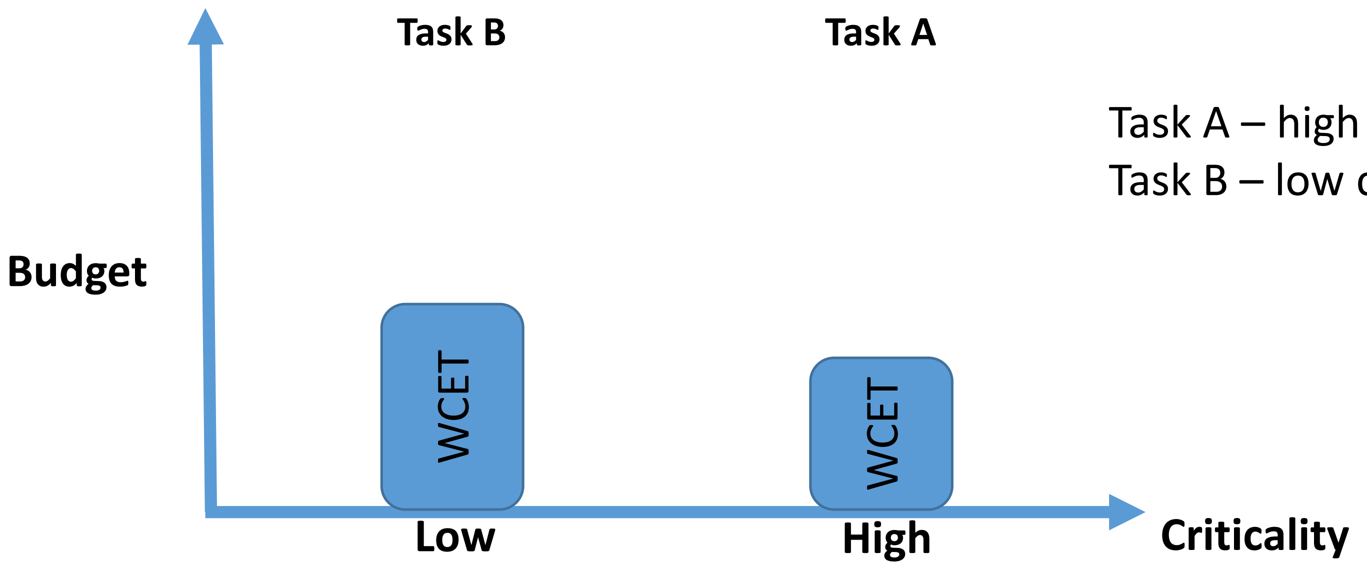
- Resource utilization estimates are pessimistic to ensure safety
  - Hardware/Software complexity adds to the pessimism

- Higher criticality → More stringent safety requirements → More pessimism

Motivation for Permitting Interference

- Resource utilization estimates are pessimistic to ensure safety
  - Hardware/Software complexity adds to the pessimism

- Higher criticality → More stringent safety requirements → More pessimism

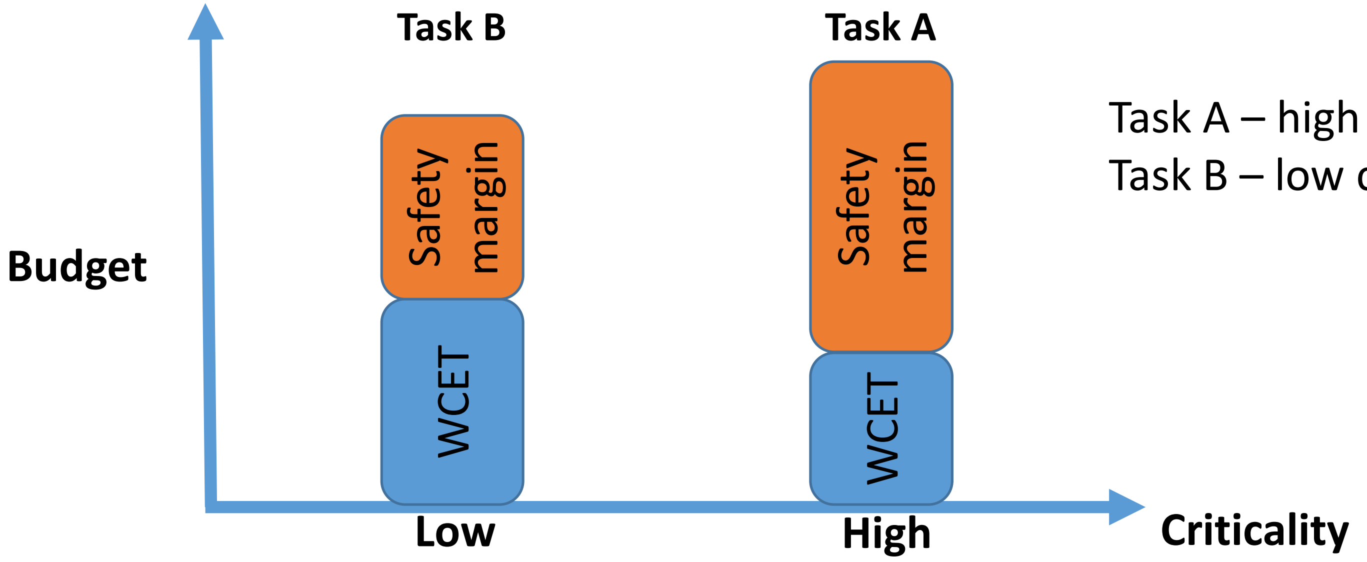


Task A – high criticality
Task B – low criticality

Motivation for Permitting Interference

- Resource utilization estimates are pessimistic to ensure safety
  - Hardware/Software complexity adds to the pessimism

- Higher criticality ➔ More stringent safety requirements ➔ More pessimism



Task A – high criticality
Task B – low criticality

Motivation for Permitting Interference

- Resource utilization estimates are pessimistic to ensure safety
    - Hardware/Software complexity adds to the pessimism
- Higher criticality → More stringent safety requirements → More pessimism
- Permit interference within the safety margin
    - Safety is not compromised
    - Resource utilization is vastly improved

Motivation for Permitting Interference

- Resource utilization estimates are pessimistic to ensure safety
  - Hardware/Software complexity adds to the pessimism

- Higher criticality → More stringent safety requirements → More pessimism

- Permit interference within the safety margin
  - Safety is not compromised
  - Resource utilization is vastly improved

- How to ensure interference is safe when WCET is unknown?
  - Permit interference and recover prior to a safety violation
  - Recovery may impact the execution of some (lower criticality) tasks
  - As long as there is no impact on safety, . . .

NANYANG TECHNOLOGICAL UNIVERSITY

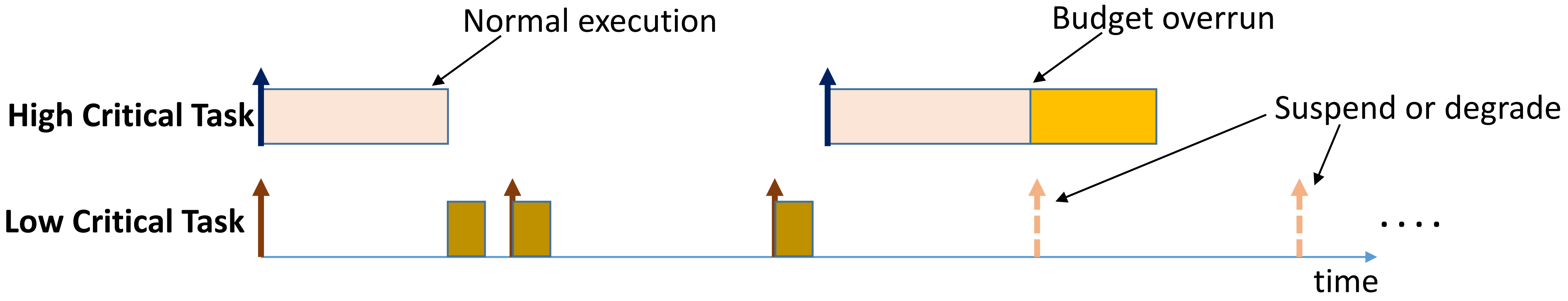
A Popular MCS Model in Academia

- Reserve less “pessimistic” budgets for tasks
  - →Allows interference
  - →Improves efficiency

A Popular MCS Model in Academia

- Reserve less “pessimistic” budgets for tasks
  - →Allows interference
  - →Improves efficiency

- In the (hopefully) “rare” case that a budget is overrun, prioritize allocations to critical tasks
  - →Recovers prior to a safety violation
  - →No impact on critical tasks



A Popular MCS Model in Academia

- Reserve less “pessimistic” budgets for tasks
  - →Allows interference
  - →Improves efficiency

- In the (hopefully) “rare” case that a budget is overrun, prioritize allocations to critical tasks
  - →Recovers prior to a safety violation
  - →No impact on critical tasks

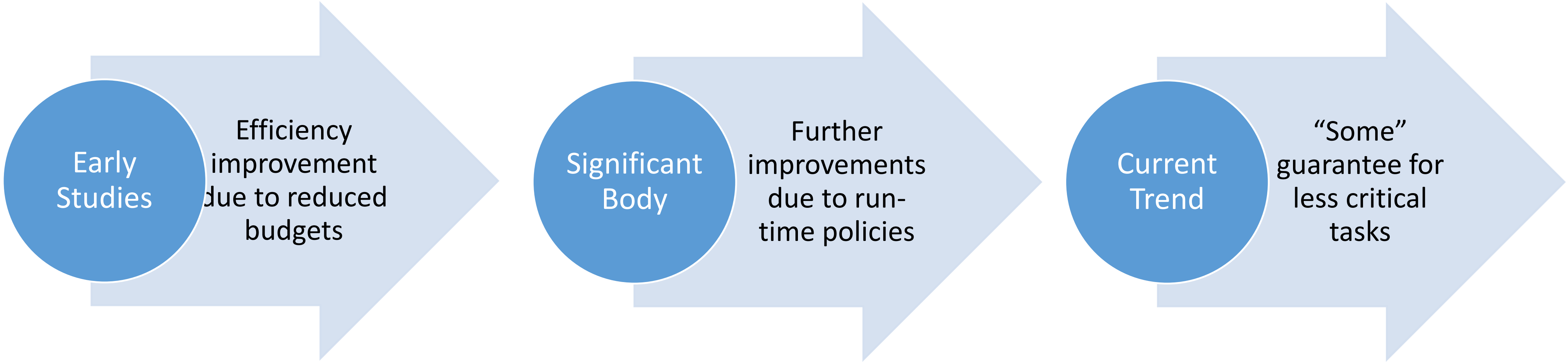
A Popular MCS Model in Academia

- Reserve less "pessimistic" budgets for tasks
  - →Allows interference
  - →Improves efficiency

- In the (hopefully) "rare" case that a budget is overrun, prioritize allocations to critical tasks
  - →Recovers prior to a safety violation
  - →No impact on critical tasks

- Possible impact on the execution of less critical tasks
  - →What is an acceptable impact, given safety specifications?

MCS Models – Research Trends



- Focus was on design time improvements to resource utilization
- Did not consider the impact on less critical tasks
- Executed them using best effort upon budget overrun for more critical tasks

MCS Models – Research Trends

Early Studies → Efficiency improvement due to reduced budgets

Significant Body → Further improvements due to run-time policies

Current Trend → “Some” guarantee for less critical tasks

Early Studies

- Focus was on design time improvements to resource utilization
- Did not consider the impact on less critical tasks
- Executed them using best effort upon budget overrun for more critical tasks

Significant Body

- Completely suspended less critical tasks upon budget overruns
- Further improvements to resource utilization, but at the cost of all guarantees for less critical tasks
- May not be reasonable from the perspective of impact on safety

MCS Models – Research Trends

Early Studies — Efficiency improvement due to reduced budgets

- Focus was on design time improvements to resource utilization
- Did not consider the impact on less critical tasks
- Executed them using best effort upon budget overrun for more critical tasks

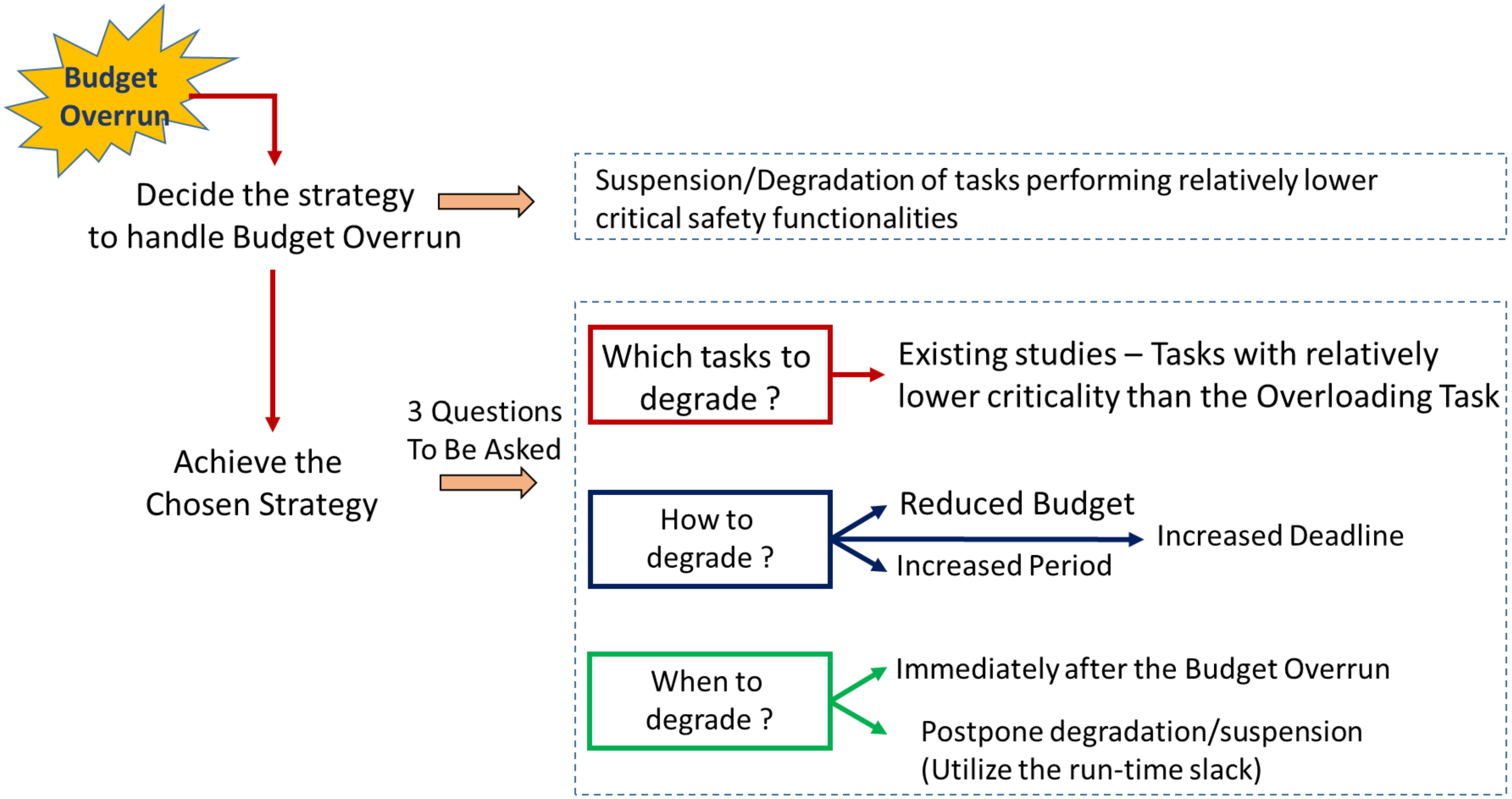
Significant Body — Further improvements due to run-time policies

- Completely suspended less critical tasks upon budget overruns
- Further improvements to resource utilization, but at the cost of all guarantees for less critical tasks
- May not be reasonable from the perspective of impact on safety

Current Trend — "Some" guarantee for less critical tasks

- Provides "some" guaranteed resource allocation to even less critical tasks at all times
- Trend in the right direction
- What is a reasonable guarantee to ensure no impact on safety?

Related Work - Classification



Challenges in Automotive MCS

1. Issues in adopting existing MCS models for automotive
2. Key questions that need to be asked for MCS with more than two criticality levels:

Which tasks can be allowed to overrun their budgets ?

Which tasks can be allowed to be degraded ?

Does relatively lower criticality mean lower importance ?

NANYANG TECHNOLOGICAL UNIVERSITY

Challenges in Automotive MCS

- Adopting existing MCS task models for automotive

Issue 1: Criticality is an abstraction of three or more factors.

Automotive Safety Integrity Level (ASIL)

- ASIL,
  - Describes the level for required risk reduction of a safety functionality
  - Classifies the hazards to 4 levels [A to D]; A – low, D – high

Automotive Safety Integrity Level (ASIL)

- ASIL,
  - Describes the level for required risk reduction of a safety functionality
  - Classifies the hazards to 4 levels [A to D]; A – low, D – high
  - For an hazard, ASIL depends on 3 factors.

Severity (S) + Controllability (C) + Probability Of Exposure (E)

Automotive Safety Integrity Level (ASIL)

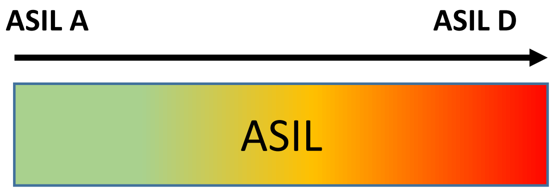
- ASIL,
  - Describes the level for required risk reduction of a safety functionality
  - Classifies the hazards to 4 levels [A to D]; A – low, D – high
  - For an hazard, ASIL depends on 3 factors.

Severity (S) + Controllability (C) + Probability Of Exposure (E) →

		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

ASIL decision chart

=



Automotive Safety Integrity Level (ASIL)

Automotive Safety Integrity Level (ASIL)

Classes of Severity

S0	S1	S2	S3
No Injuries	Light and moderate injuries	Severe and life threatening injuries(survival probable)	Life-threatening injuries (survival uncertain), fatal injuries

NANYANG TECHNOLOGICAL UNIVERSITY

Automotive Safety Integrity Level (ASIL)

Classes of Severity

S0	S1	S2	S3
No Injuries	Light and moderate injuries	Severe and life threatening injuries(survival probable)	Life-threatening injuries (survival uncertain), fatal injuries

Classes of Probability of Exposure

E0	E1	E2	E3	E4
Incredible	Very low probability	Low probability	Medium probability	High probability

Automotive Safety Integrity Level (ASIL)

Classes of Severity

S0	S1	S2	S3
No Injuries	Light and moderate injuries	Severe and life threatening injuries(survival probable)	Life-threatening injuries (survival uncertain), fatal injuries

Classes of Probability of Exposure

E0	E1	E2	E3	E4
Incredible	Very low probability	Low probability	Medium probability	High probability

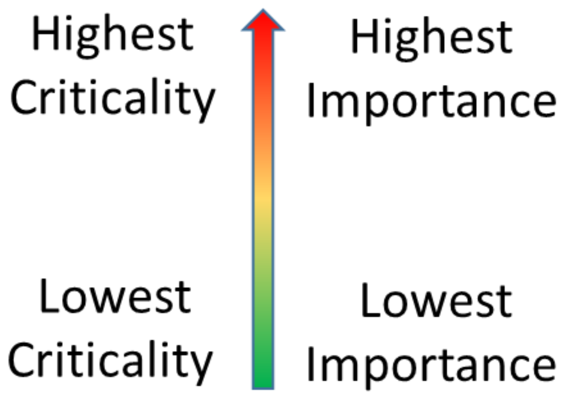
Classes of Controllability

C0	C1	C2	C3
Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable

Challenges in Automotive MCS

- Adopting existing MCS task models for automotive

Issue 2: Lower criticality as lower importance



Automotive MCS Challenges

Consider the following applications,

Automotive MCS Challenges

Consider the following applications,

Adaptive Cruise Control (ACC) → ASIL C → High critical

Hill Start Assist (HSA) → ASIL B → Low critical

Automotive MCS Challenges

Consider the following applications,

Adaptive Cruise Control (ACC) → ASIL C	→ High critical
Hill Start Assist (HSA) → ASIL B	→ Low critical

If driving in hills is considered as a rare event,

NANYANG TECHNOLOGICAL UNIVERSITY

Automotive MCS Challenges

Consider the following applications,

Adaptive Cruise Control (ACC) → ASIL C → High critical

Hill Start Assist (HSA) → ASIL B → Low critical

If driving in hills is considered as a rare event,

Hill Start Assist → low probability of exposure → ASIL B

Automotive MCS Challenges

Consider the following applications,

Adaptive Cruise Control (ACC) → ASIL C → High critical

Hill Start Assist (HSA) → ASIL B → Low critical

NANYANG TECHNOLOGICAL UNIVERSITY

Automotive MCS Challenges

Consider the following applications,

Adaptive Cruise Control (ACC) → ASIL C → High critical

Hill Start Assist (HSA) → ASIL B → Low critical

But, if the vehicle is actually riding on a hill, then

Degradation of HSA may not be acceptable

Automotive MCS Challenges

Consider the following applications,

Adaptive Cruise Control (ACC) → ASIL C → High critical

Hill Start Assist (HSA) → ASIL B → ~~Low critical~~ High critical

But, if the vehicle is actually riding on a hill, then

Degradation of HSA may not be acceptable

Automotive MCS Challenges

Consider the following applications,

Adaptive Cruise Control (ACC) → ASIL C → High critical

Hill Start Assist (HSA) → ASIL B → ~~Low critical~~ High critical

But, if the vehicle is actually riding on a hill, then

Degradation of HSA may not be acceptable

Sometimes, lower ASIL applications can be considered highly important

Automotive MCS Challenges

Decomposition of safety requirements may lead to lower criticality

ASIL D Safety Requirements = ASIL C Safety Requirements + ASIL A Safety Requirements

↓

Lead Vehicle Detection = Radar + V2V

Redundant and Independent Sensors

Lower criticality task may perform safety functionality with higher importance !

Automotive MCS Challenges

Consider the following applications,

Acceptable Interference -

Acceptable Degradation -

NANYANG TECHNOLOGICAL UNIVERSITY

Automotive MCS Challenges

Consider the following applications,

Adaptive Cruise Control (ACC) → ASIL D	→ High critical
Hill Start Assist (HSA) → ASIL B	→ Low critical

Acceptable Interference -

Acceptable Degradation -

NANYANG TECHNOLOGICAL UNIVERSITY

Automotive MCS Challenges

Consider the following applications,

Adaptive Cruise Control (ACC) → ASIL D → High critical

Hill Start Assist (HSA) → ASIL B → Low critical

Acceptable Interference -

Can HSA be allowed to overrun its budget ?

Acceptable Degradation -

Automotive MCS Challenges

Consider the following applications,

Adaptive Cruise Control (ACC) → ASIL D - - - - - → High critical
Hill Start Assist (HSA) → ASIL B - - - - - - → Low critical

Acceptable Interference -

Can HSA be allowed to overrun its budget ?

Acceptable Degradation -

Can ACC be degraded to handle overrun of HSA ?

Automotive MCS Challenges

Issue 3 (Possibility) Can we consider degradation of higher critical functionality to handle budget overrun of relatively lower critical functionality ?

Functionality of ACC (Adaptive Cruise Control)

- Safety Aspect — Main inter-vehicle distance, Apply Acceleration and Brake
- Performance Aspect — Smoothness in control

Degradation of higher critical functionality without affecting its safety !

NANYANG TECHNOLOGICAL UNIVERSITY

Automotive MCS Challenges

LIDAR Processing Task

Normal Operation		Degraded Operation		
		Mode 1	***Mode 2***	***Mode 3***
Scanning fre-quency	*10 Hz*	*5 Hz*	*5 Hz*	*5 Hz*
Angular resolu-tion	*0.25 °*	*0.25 °*	*0.125°*	*0.25 °*
Aquisition angle	*270 °*	*270 °*	*270 °*	*270 °*
Echo Amplitudes	*yes*	*yes*	*no*	*no*
TCP payload at each sensor	*43430 Byte / s*	*21715 Bytes /s*	*21715 Bytes /s*	*41 Bytes / s averaged*
Normalized computing power in ECU	*1*	*0.450*	*0.303*	*0.032*
Update rate	*10 pictures / s*	*5 pictures / s*	*5 pictures / s*	*~1 picture / s*

Multiple ways of degrading a task's budget can be considered!

Source: **Graceful Degradation for Driver Assistance Systems**, H. U. Michel, S. Holzknecht, E. Biebl, Springer 2009

Key Properties of any MCS

	Properties of a Mixed Criticality System
Property 1	A lower critical task does not mean that it always performs a functionality of lower importance.
Property 2	Degradation of higher criticality tasks is possible.
Property 3	Multiple ways of degrading a task's budget is possible.
Property 4	A specific degradation of a task depending on the overloading task can be useful.

Research Objective: A new degradation model for MCS based on above 4 properties

NANYANG TECHNOLOGICAL UNIVERSITY

Intuition for Context-Aware MCS Model

Budget
Overrun

Intuition for Context-Aware MCS Model

Budget Overrun

Which tasks to degrade ?

Use **criticality** to choose Tasks to be degraded

Relatively lower criticality tasks are degraded

Majority of the existing Studies

Intuition for Context-Aware MCS Model

Which tasks to degrade ?

Use **criticality** to choose
Tasks to be degraded

Relatively lower criticality
tasks are degraded

How to degrade ?

Fixed type of degradation
Reduced budget/Inc. period
Decide offline

Majority of the
existing Studies

Budget
Overrun

Intuition for Context-Aware MCS Model

Which tasks to degrade ?

How to degrade ?

Use **criticality** to choose Tasks to be degraded

Fixed type of degradation
Reduced budget/Inc. period
Decide offline

Majority of the existing Studies

Budget Overrun

Relatively lower criticality tasks are degraded

Allow the designer to choose the task to be degraded

This Paper

NANYANG TECHNOLOGICAL UNIVERSITY

Intuition for Context-Aware MCS Model

Which tasks to degrade ?

How to degrade ?

Use **criticality** to choose
Tasks to be degraded

→

Fixed type of degradation
Reduced budget/Inc. period
Decide offline

Majority of the
existing Studies

Relatively lower criticality
tasks are degraded

Budget
Overrun

Allow the designer
to choose the task
to be degraded

→ Single task
overrun →

Specific degraded
Budget based on overrun task

This
Paper

NANYANG TECHNOLOGICAL UNIVERSITY

Intuition for Context-Aware MCS Model

Budget Overrun

Which tasks to degrade ?

Use **criticality** to choose Tasks to be degraded

Relatively lower criticality tasks are degraded

How to degrade ?

Fixed type of degradation
Reduced budget/Inc. period
Decide offline

Majority of the existing Studies

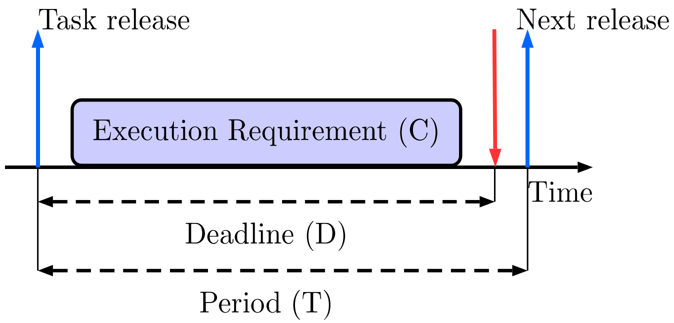
Allow the designer to choose the task to be degraded

Single task overrun → **Specific degraded Budget** based on overrun task

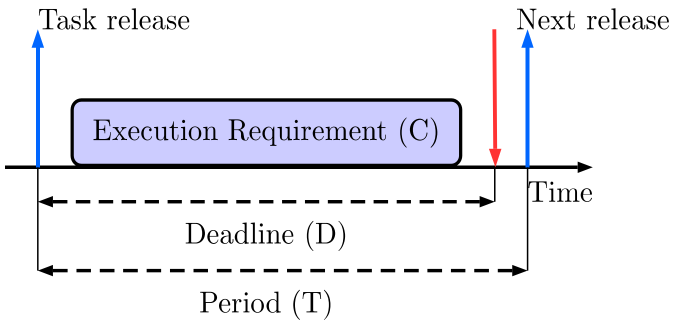
Multiple tasks overrun → **Specific degraded budget** based on the **Criticality** of overrun tasks

This Paper

Context-Aware MCS Model - Syntax



Context-Aware MCS Model - Syntax



$$\tau_i = \{\mathcal{C}_i, T_i, \mathcal{B}_i, \mathcal{L}_i\}$$

T_i - Time period

$\mathcal{C}_i = \{C_i^0, C_i^1, C_i^2, ..., C_i^{n-1}, C_i^n\}$

C_i^0 - Normal Budget (lower pessimism)

C_i^i - Safe Budget (higher pessimism)

$C_i^1, C_i^2, ..., C_i^{n-1}$ - Degraded Budgets (higher pessimism)

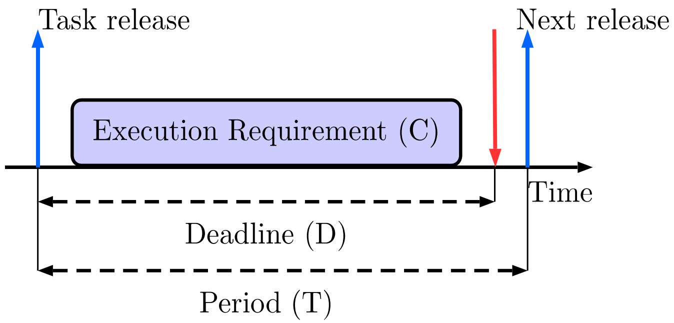
$\mathcal{B}_i = \{B_i^0, B_i^1, ..., B_i^n\}$ - A set of behaviours

$B_i^k \in \{Normal, Safe, Degraded\}$

$\mathcal{L}_i$ - Criticality Level

NANYANG TECHNOLOGICAL UNIVERSITY

Context-Aware MCS Model - Syntax



Assumptions regarding budgets:

$\forall k$ in the range $1 \leq k \leq n$, $C_i^k \leq C_i^0$, if $B_i^k = Degraded$

$C_i^k = C_i^i$, if $B_i^k = Safe$

$C_i^k = C_i^0$, if $B_i^k = Normal$

$\forall \tau_i \in T$, $C_i^0 \leq C_i^i$

τ_i is allowed to overrun only its C_i^0

$$\tau_i = \{\mathcal{C}_i, T_i, \mathcal{B}_i, \mathcal{L}_i\}$$

T_i - Time period

$\mathcal{C}_i = \{C_i^0, C_i^1, C_i^2, ..., C_i^{n-1}, C_i^n\}$

C_i^0 - Normal Budget (lower pessimism)

C_i^i - Safe Budget (higher pessimism)

$C_i^1, C_i^2, ..., C_i^{n-1}$ - Degraded Budgets (higher pessimism)

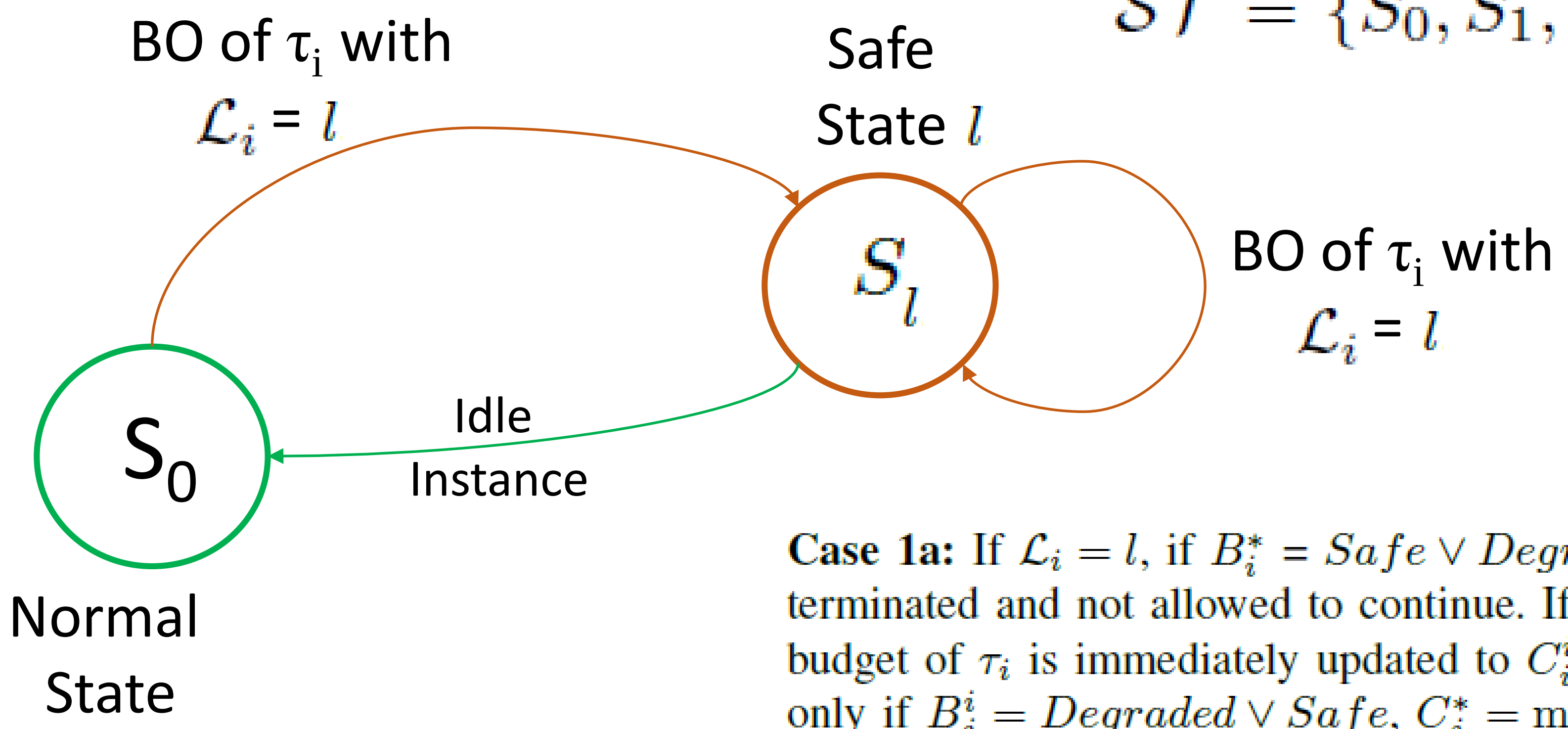
$\mathcal{B}_i = \{B_i^0, B_i^1, ..., B_i^n\}$ - A set of behaviours

$B_i^k \in \{Normal, Safe, Degraded\}$

$\mathcal{L}_i$ - Criticality Level

Context-Aware MCS Model - Semantics

$$\mathcal{ST} = \{S_0, S_1, S_2, ..., S_L\}$$

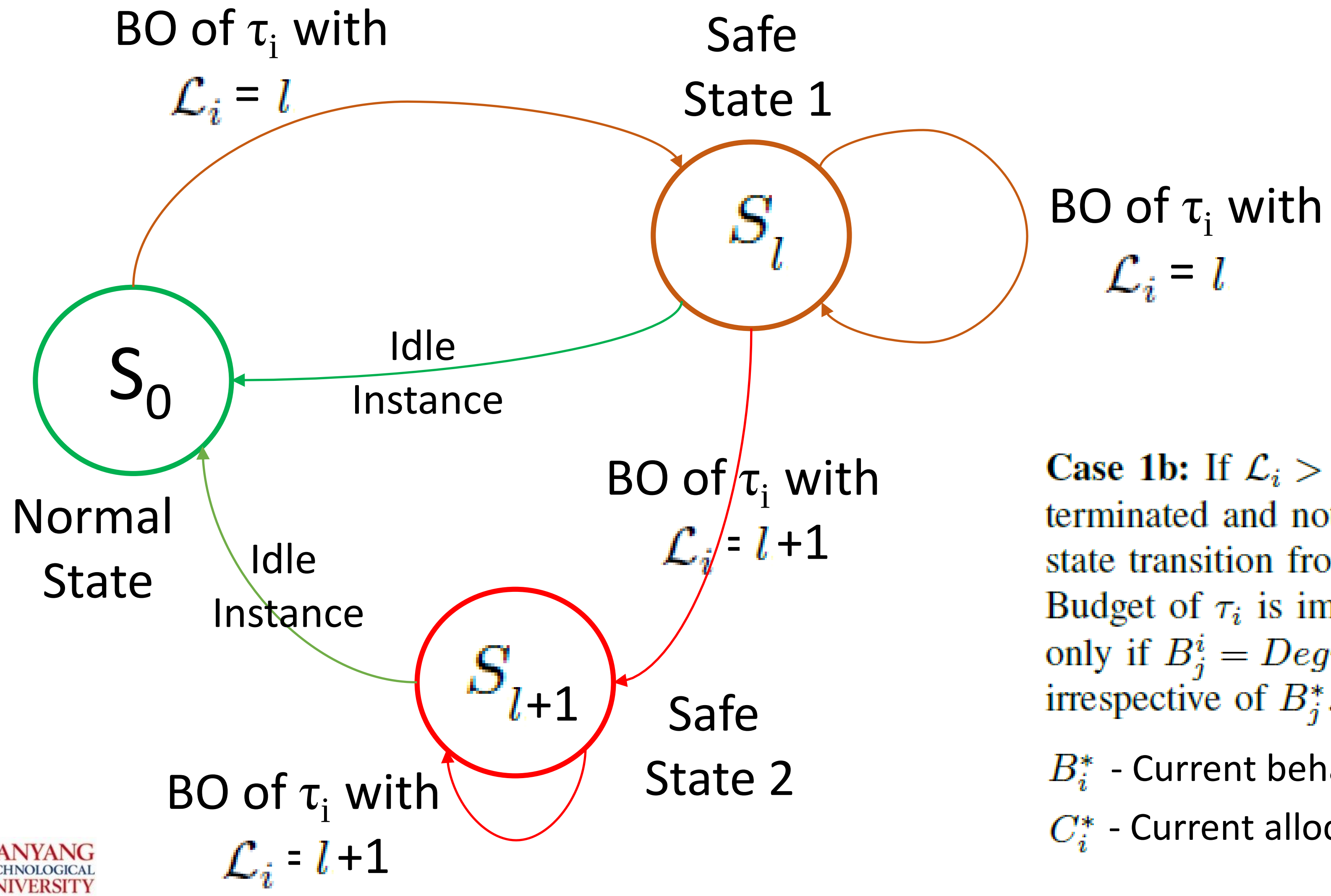


Case 1a: If $\mathcal{L}_i = l$, if $B_i^* = Safe \vee Degraded$, J_i is terminated and not allowed to continue. If $B_i^* = Normal$, budget of τ_i is immediately updated to C_i^i. $\forall \tau_j \in T \setminus \tau_i$, only if $B_j^i = Degraded \vee Safe$, $C_j^* = \min(C_j^*, C_j^i)$ is updated irrespective of B_j^*.

B_i^* - Current behaviour of τ_i

C_i^* - Current allocated budget of τ_i

Context-Aware MCS Model - Semantics



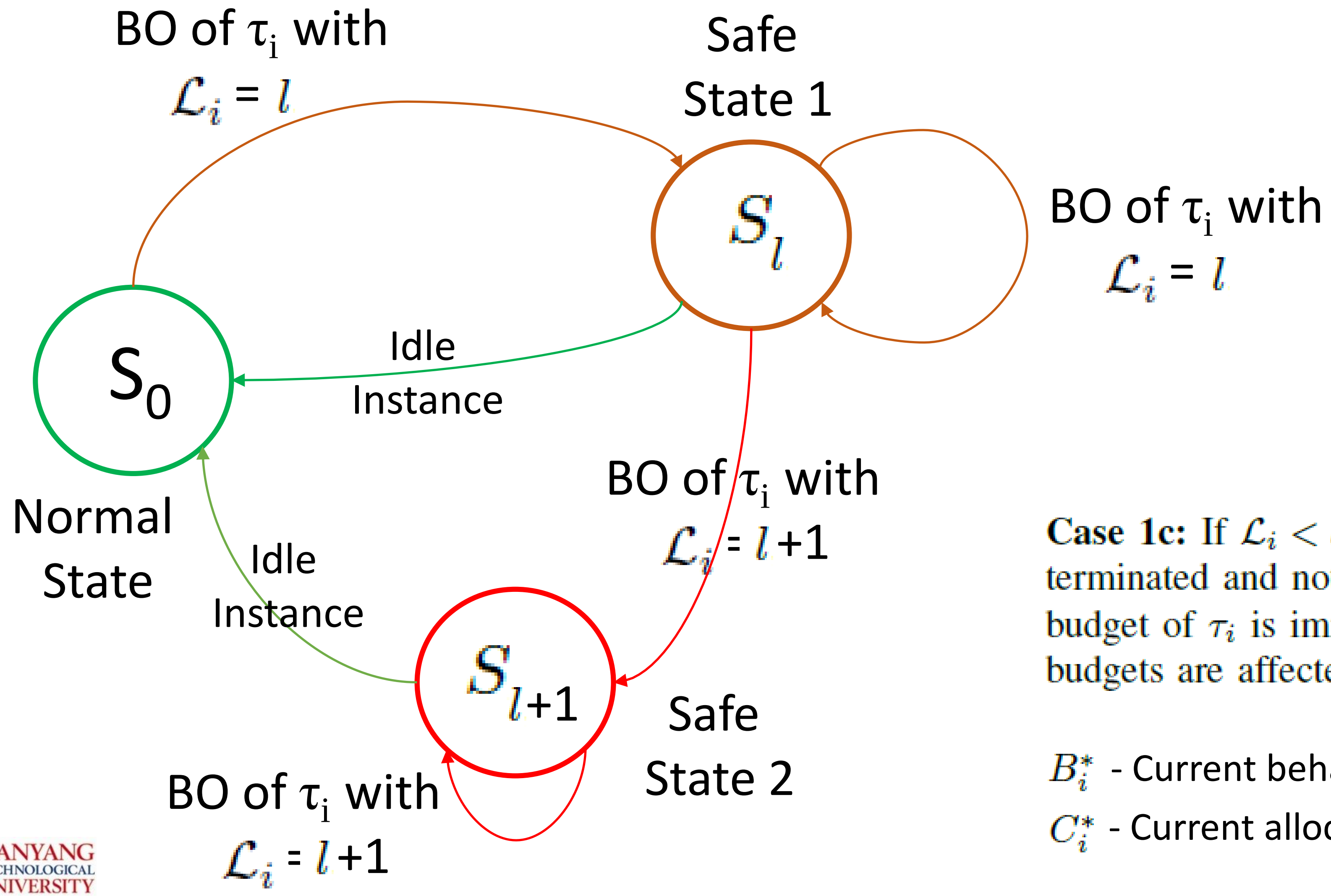
Case 1b: If $\mathcal{L}_i > l$, if $B_i^* = Safe \vee Degraded$, J_i is terminated and not allowed to continue. If $B_i^* = Normal$, state transition from $\mathcal{S}_l$ to $\mathcal{S}_{\mathcal{L}_i} \in \mathcal{ST}$ is immediately done. Budget of τ_i is immediately updated to C_i^i. $\forall \tau_j \in T \setminus \tau_i$, only if $B_j^i = Degraded$, $C_j^* = C_j^i$ is updated to τ_j irrespective of B_j^*. Otherwise, tasks' budgets are not affected.

B_i^* - Current behaviour of τ_i

C_i^* - Current allocated budget of τ_i

NANYANG TECHNOLOGICAL UNIVERSITY

Context-Aware MCS Model - Semantics

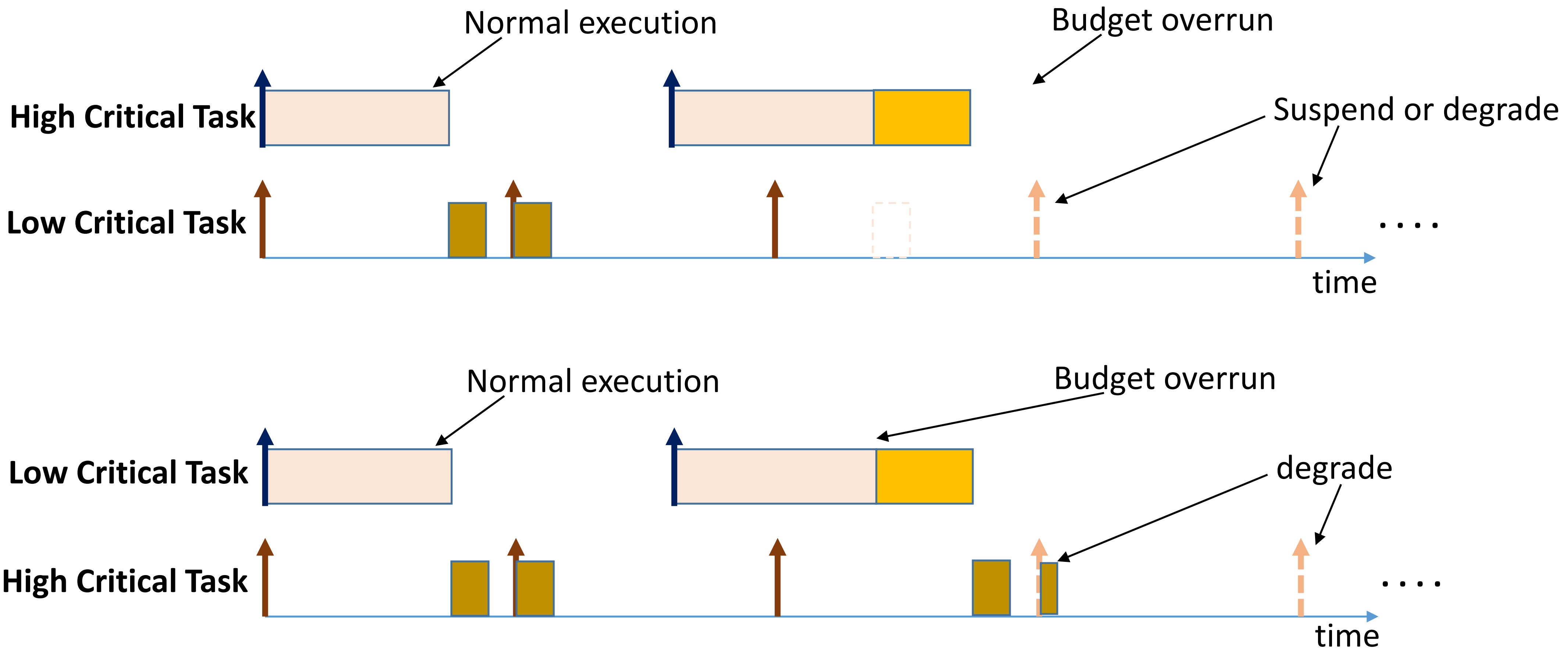


Case 1c: If $\mathcal{L}_i < l$, if $B_i^* = Safe \vee Degraded$, J_i is terminated and not allowed to continue. If $B_i^* = Normal$, budget of τ_i is immediately updated to C_i^i. No other tasks' budgets are affected.

B_i^* - Current behaviour of τ_i

C_i^* - Current allocated budget of τ_i

Updation of Budgets



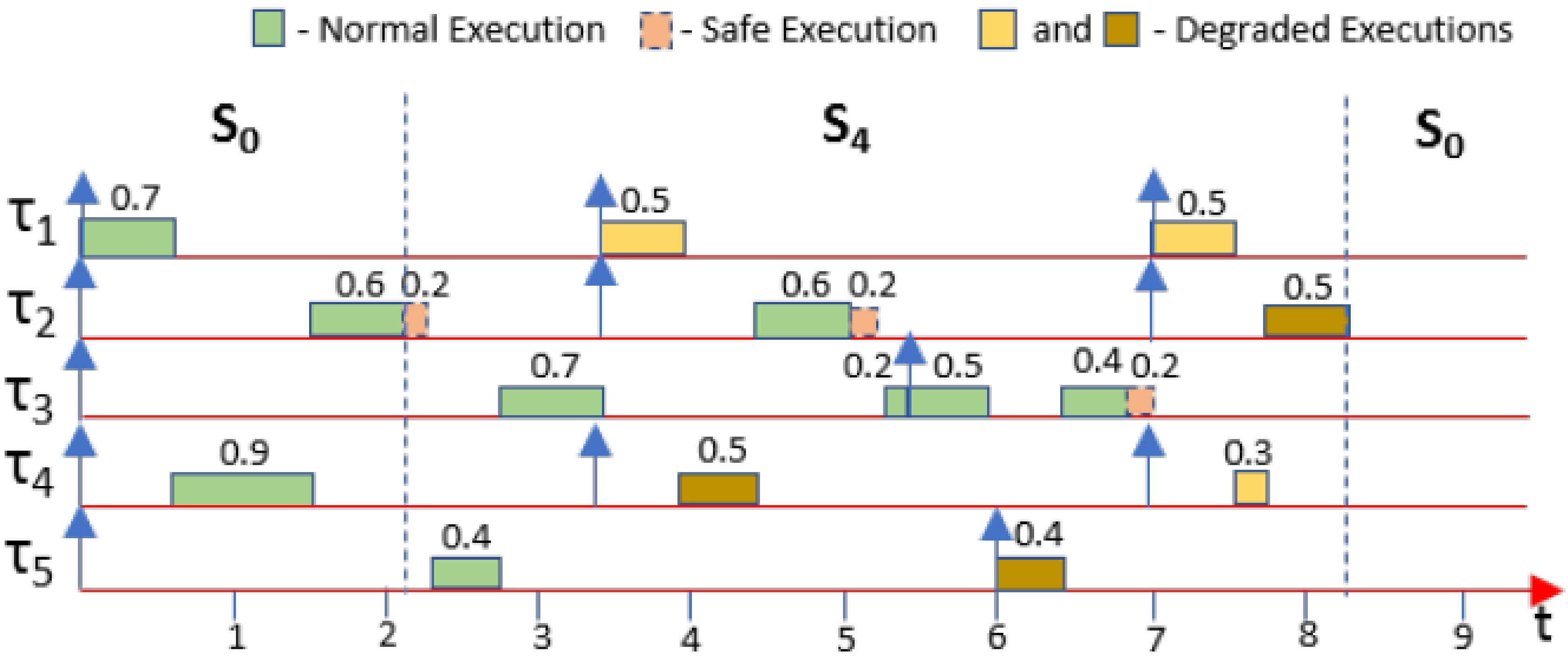
NANYANG TECHNOLOGICAL UNIVERSITY

Example Taskset

EXAMPLE TASK SET

$\tau_i = \{C_i, T_i, B_i, \mathcal{L}_i\}$	P_i
$\tau_1 = \{\{0.7, 1.1, 0.5, 0.6, 0.6, 0.5\}, 3.5, \{N, S, D, D, D, D\}, 3\}$	1
$\tau_2 = \{\{0.6, 0.3, 0.8, 0.5, 0.5, 0.5\}, 3.5, \{N, D, S, D, D, D\}, 4\}$	3
$\tau_3 = \{\{0.9, 0.8, 0.9, 1.1, 0.3, 0.2\}, 5.5, \{N, D, N, S, D, D\}, 4\}$	5
$\tau_4 = \{\{0.9, 0.9, 0.5, 0.3, 1.2, 0.4\}, 3.5, \{N, N, D, D, S, D\}, 5\}$	2
$\tau_5 = \{0.4, 0.4, 0.4, 0.4, 0.4, 0.5\}, 6, \{N, N, N, D, N, S\}, 2\}$	4

$N - Normal; D - Degraded; S - Safe$



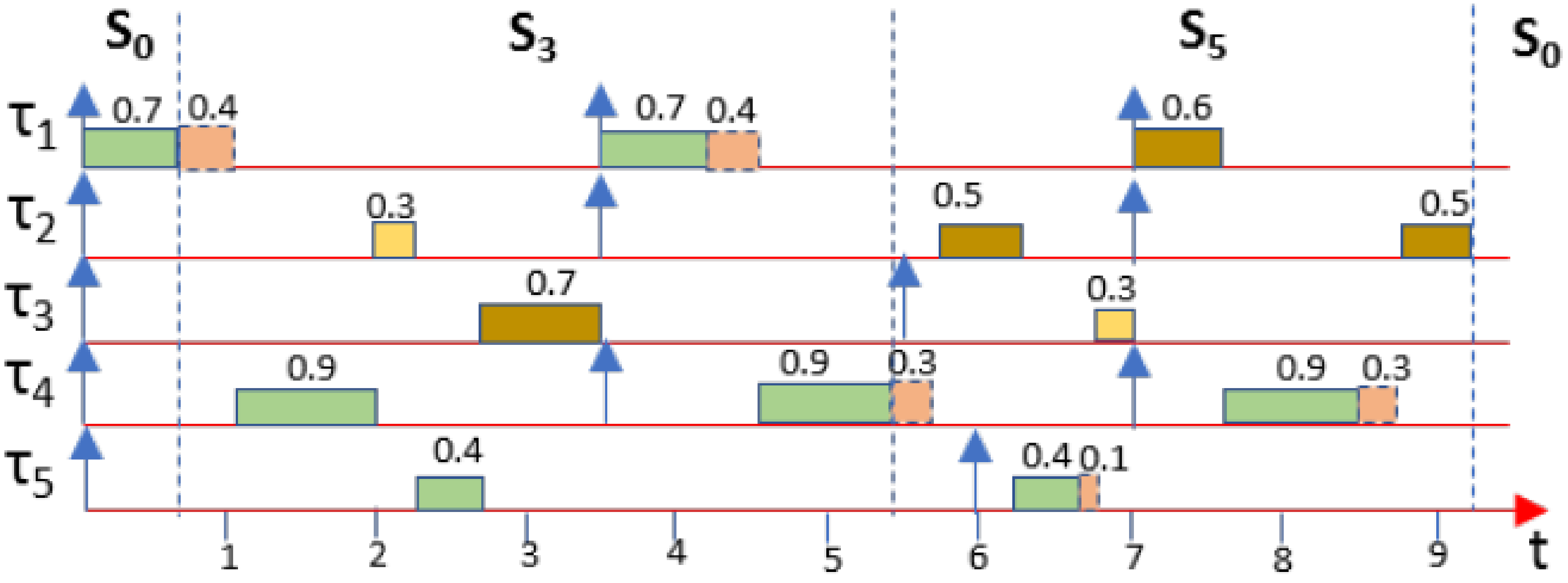
Schedule 1: Overrun tasks with same criticality

Example Taskset

EXAMPLE TASK SET

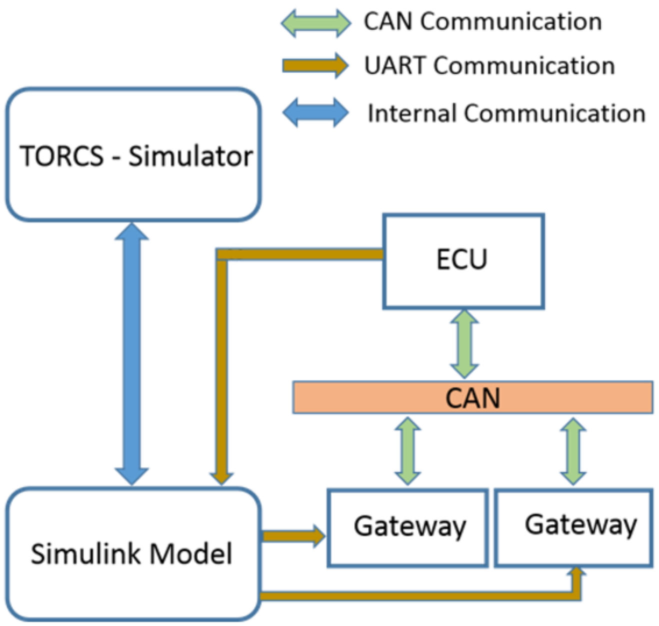
$\tau_i = \{C_i, T_i, B_i, \mathcal{L}_i\}$	P_i
$\tau_1 = \{\{0.7, 1.1, 0.5, 0.6, 0.6, 0.5\}, 3.5, \{N, S, D, D, D, D\}, 3\}$	1
$\tau_2 = \{\{0.6, 0.3, 0.8, 0.5, 0.5, 0.5\}, 3.5, \{N, D, S, D, D, D\}, 4\}$	3
$\tau_3 = \{\{0.9, 0.8, 0.9, 1.1, 0.3, 0.2\}, 5.5, \{N, D, N, S, D, D\}, 4\}$	5
$\tau_4 = \{\{0.9, 0.9, 0.5, 0.3, 1.2, 0.4\}, 3.5, \{N, N, D, D, S, D\}, 5\}$	2
$\tau_5 = \{\{0.4, 0.4, 0.4, 0.4, 0.4, 0.5\}, 6, \{N, N, N, D, N, S\}, 2\}$	4

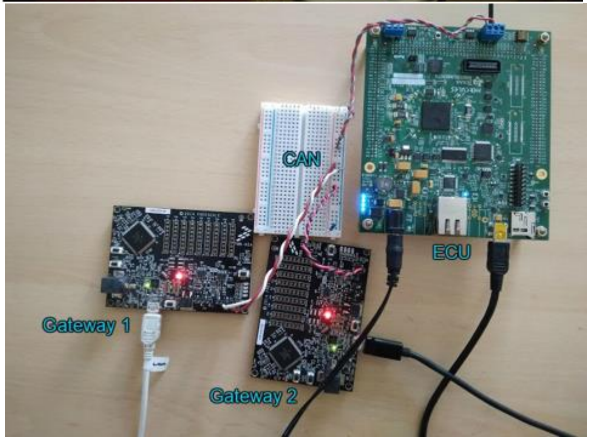
$N - Normal; D - Degraded; S - Safe$

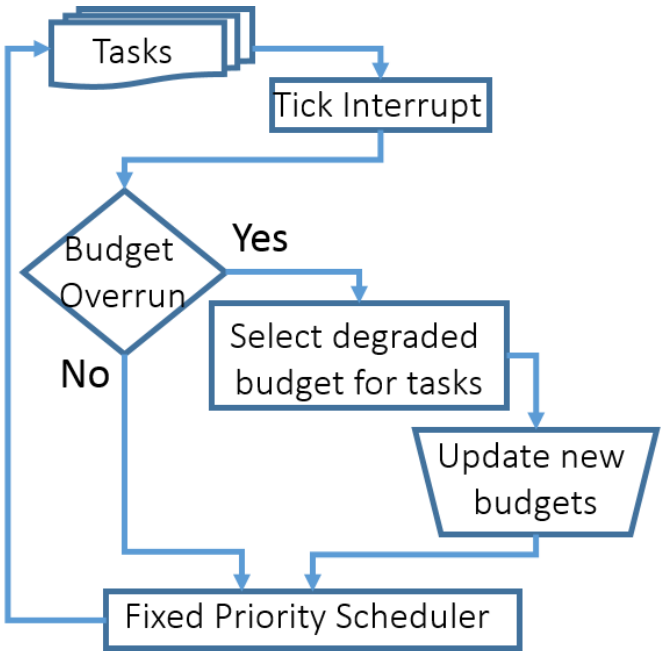


Schedule 2: Overrun tasks with different criticality

Testbed Architecture







Applications Implemented

- Lead Vehicle Detection
  - Pseudo radar task – ASIL C, V2V task – ASIL B
- Longitudinal Vehicular Control
  - Adaptive Cruise Control (ASIL C)
    - PID or ONOFF control mechanism
    - Intelligent Speed Adaptation in curves
  - Collision Avoidance (ASIL D)
- Lateral Vehicular Control
  - Steer Control – ASIL D

Budgets (Ticks)	Tasks					
	Radar	V2V	CC	CA	SC	CAN
Normal	8771	8153	162524	5810	4618	9950
Safe	17861	16532	324590	10234	8432	10785

NANYANG TECHNOLOGICAL UNIVERSITY

Context-Aware Degradation in the Testbed

- ACC implemented with both safety and performance aspects
  - Degradation type 1 : PID → ONOFF
  - Degradation type 2 : No ISA

Types	V2V Task		CC Task	
	Budget (Ticks)	Period	Budget (Ticks)	Period
Type 1	8153	60 ms	146534 (ON-OFF)	20ms
Type 2	NA		154885 (No DSA)	20ms

Steer Task Overrun → Degrade ACC type 2 (with No ISA) → Impacts only the heading error

CA Task Overrun → Degrade ACC type 1 (PID→ONOFF) → Impacts only the acceleration

What can TORCS provide you ?

- A variety of sensor values (around 25 different parameters) at run-time.

Position, Acceleration, Velocity (X,Y,Z) of self , other vehicles

Roll, Pitch and yaw

Temperature, Pressure, Fuel remaining

Details for overtaking like overtake radius, overtake distance, velocity

Track length, curvature, remaining distance, car width, length, weight

Refer car.h and car.cpp in TORCS source code

Sensor data used in the Test-bed

Sensor data	Data Type	Description
position[3]	Double - Array	Global position [m]
velocity[3]	Double - Array	Global velocity [m/s]
acceleration[3]	Double - Array	Global acceleration [m/s/s]
angle[3]	Double - Array	Roll/Pitch/Yaw [rad]
Angular Velocity[3]	Double - Array	Roll/Pitch/Yaw rates [rad/s]
Heading Error	Double	Error between vehicle heading and track heading (at current location) [rad]
lateral Error	Double	Lateral error between car (CoG) and track centreline (at current location) [m]
roadDistance	Double	Distance travelled along track from start/finish line [m]
roadCurvature	Double	Curvature of track (at current location), left turns = +ve curvature, right turns = -ve curvature
engineRPM	Double	Engine RPM

NANYANG TECHNOLOGICAL UNIVERSITY

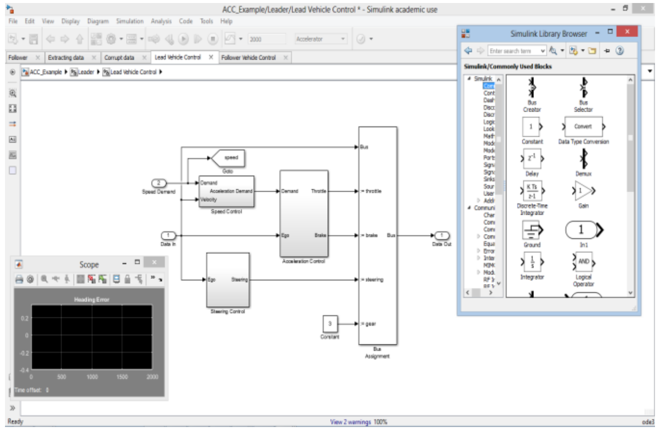
Simulink Model

Sensor data from TORCS

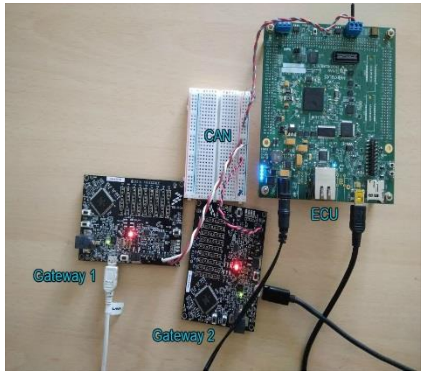
Simulink

- Run time data monitoring
- Perform sensor fusion
- Feed data to a model or to an algorithm
- Record values, results
- Plot graphs
- Direct values to any external hardware

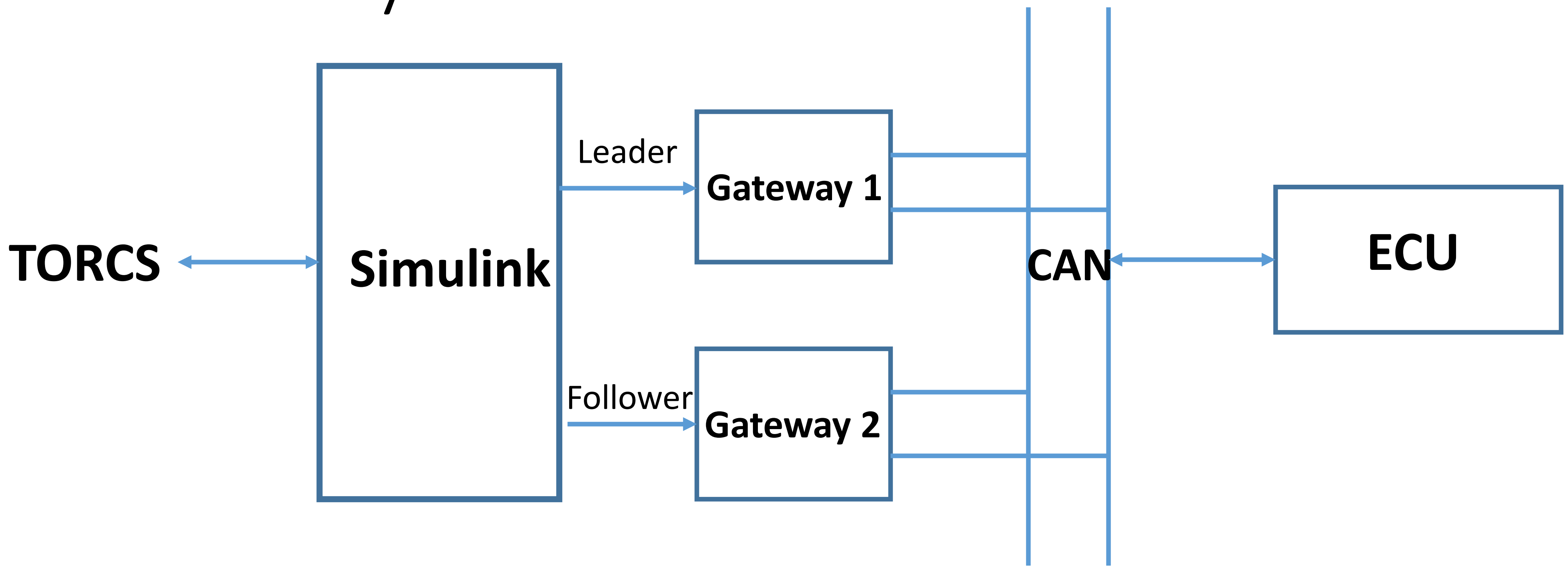
Simulink functionalities



Electronic Control Unit



Gateway



Objective :

Convert sensor values from TORCS into CAN messages

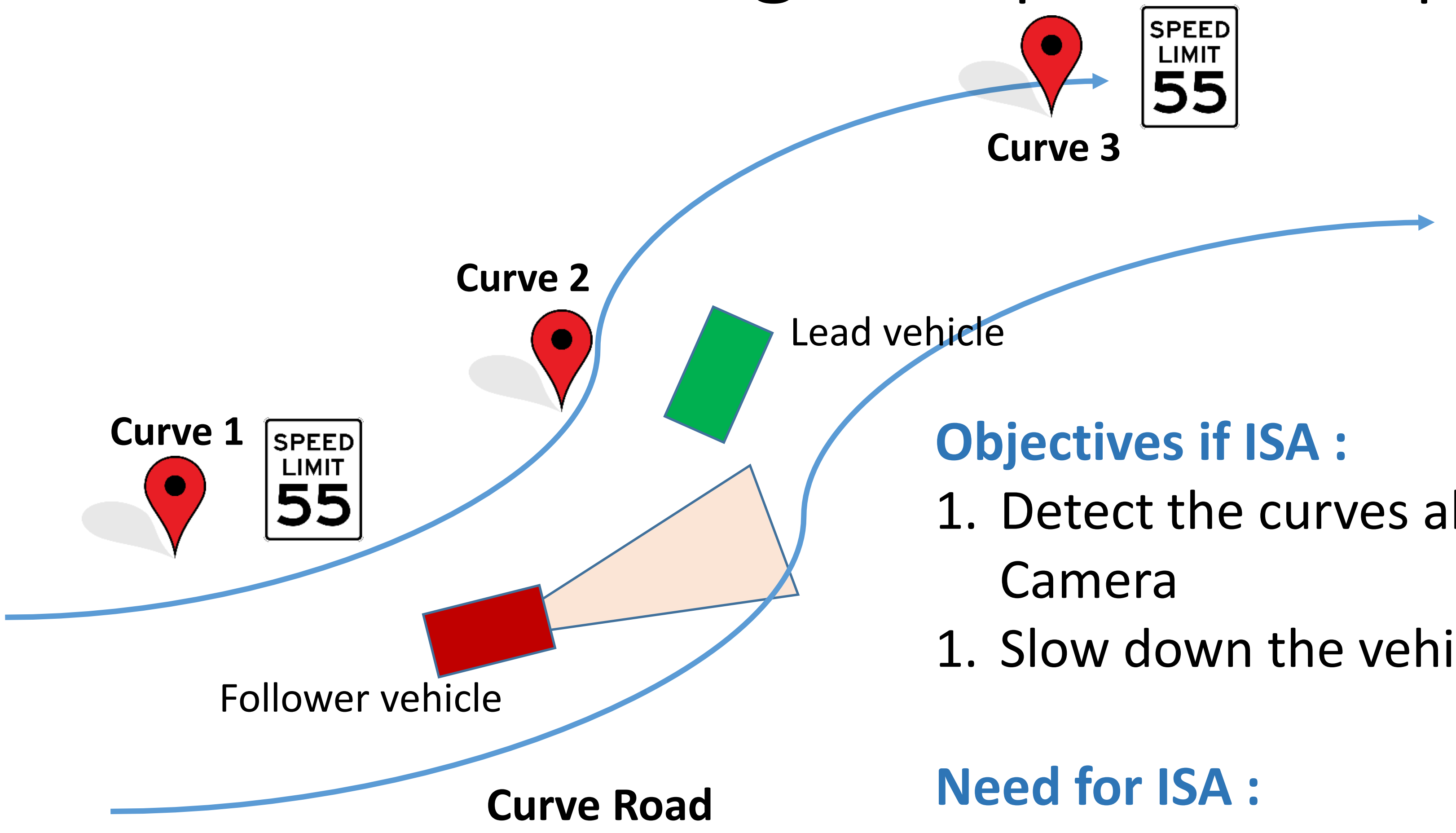
FreeRTOS

- FreeRTOS is a real-time operating system for embedded devices - ported to 35 microcontrollers.

Technology Highlights - FreeRTOS	
Pre-emptive scheduling option	Easy to use message passing
Co-operative scheduling option	Round robin with time slicing
Fast task notifications	Mutexes with priority inheritance
6K to 12K ROM footprint	Recursive mutexes
Configurable / scalable	Binary and counting semaphores
Chip and compiler agnostic	Very efficient software timers
Some ports never completely disable interrupts	Easy to use API

Src : FreeRTOS

ACC with Intelligent Speed Adaptation



Objectives if ISA :

1. Detect the curves ahead with sensors like GPS, Camera
1. Slow down the vehicle to safe velocity

Need for ISA :

1. Radar may fail to detect the lead vehicle in curves
2. Manoeuvre curve with higher velocity can lead to collision

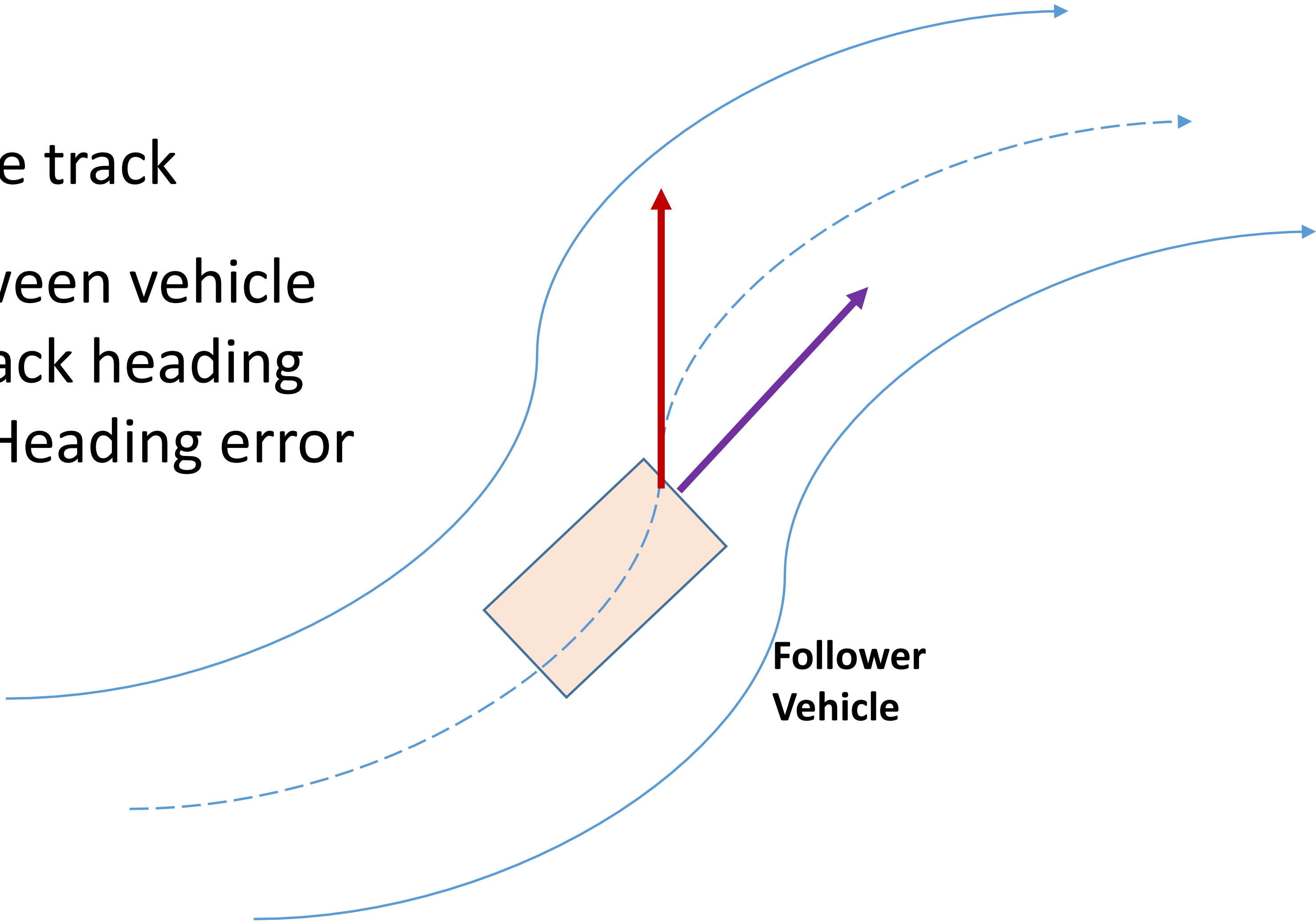
Steering Control

Objective of Steering Control :

Keep vehicle at the centre of the track

Heading Error = Difference between vehicle heading and Track heading

Steer Command = Function of Heading error



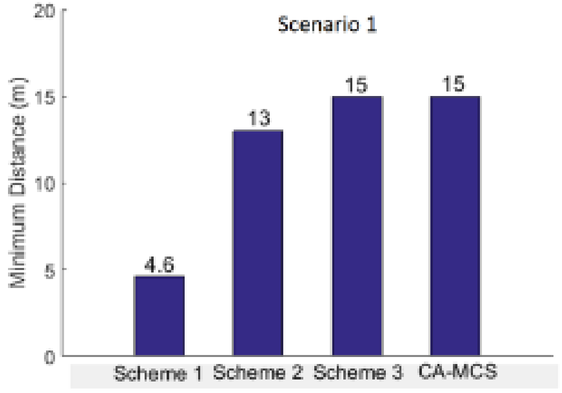
Experiments and Results

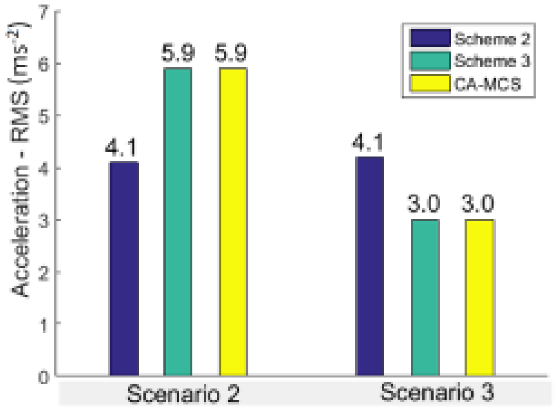
Parameters monitored:

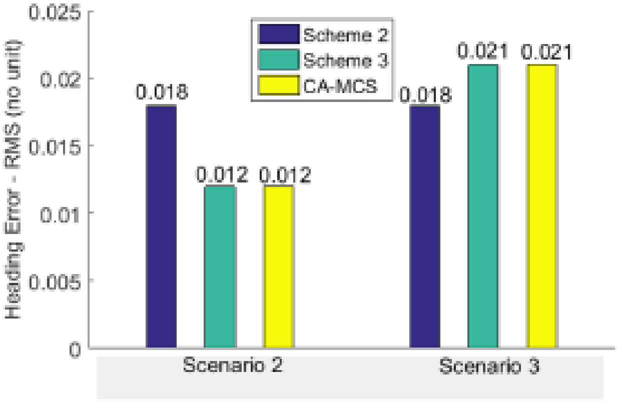
- Minimum Distance maintained between the vehicles – Safety parameter
- Heading Error – performance of lateral vehicular control
- Acceleration – performance of longitudinal vehicular control

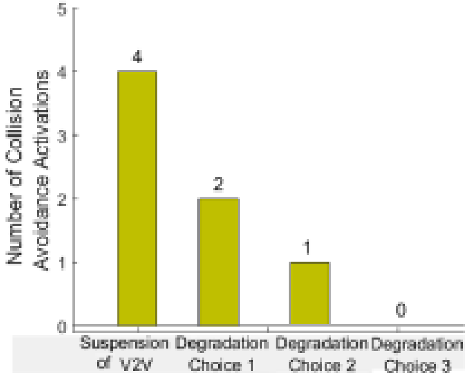
Experiments and Results

Degradation Schemes	Scenario 1 (Pseudo-Radar Overrun)	Scenario 2 (Collision Avoidance Overrun)	Scenario 3 (Steer Control Overrun)	Scenario 4 (Pseudo-Radar (and Steer Control Overrun)
Scheme 1	Suspension of V2V task.			
Scheme 2	Type 1 degradation of V2V task.	Type 1 degradation of V2V task and Type 1 degradation of CC task.		
Scheme 3 Context Aware - MCS	Type 1 degradation of both V2V and CC task.	Type 1 degradation of CC task.	Type 2 degradation of CC task.	1. Type 1 degradation of V2V and CC. 2. Type 1 degradation of V2V and Type 2 degradation of CC tasks. 3. Type 2 degradation of CC task.









Conclusion and Future Work

- CA-MCS model can effectively isolate effects of performance degradation between applications of different criticality

- Future work will be focussed to integrate mode change protocols with the CA-MCS model to achieve graceful degradation.

Thank You ☺