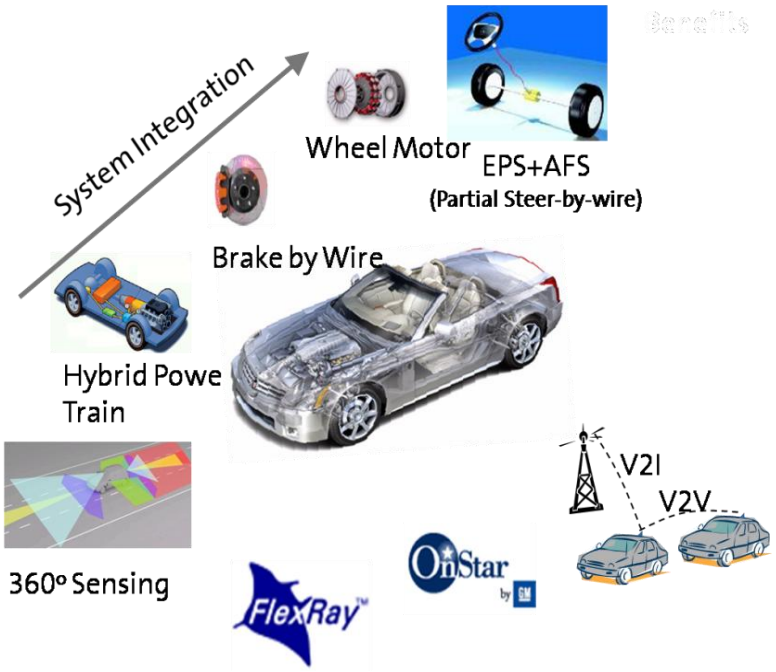
Early Time-Budgeting: A Methodology for Developing Component Based Embedded Control Systems

Manoj Dixit and S. Ramesh, ISL, GM R&D

Pallab Dasgupta, IIT Kharagpur

Background

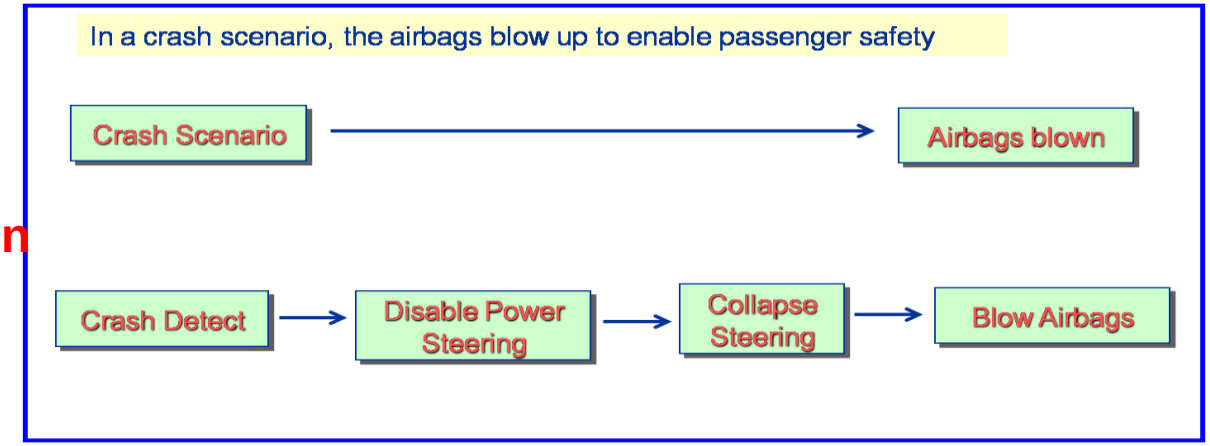


- Modern embedded control systems have large number of features
- Safety features have strict real-time end-to-end requirements
- Many components interact together to meet system level requirements
- System is distributed in nature

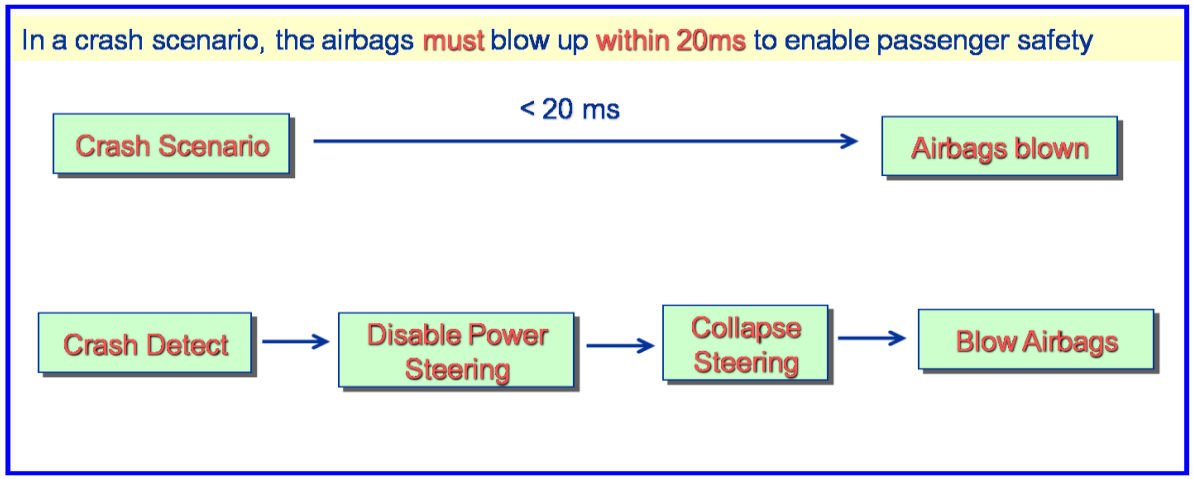
How to do a timing layout of entire system to meet end-to-end real-time requirements?

Illustrative Example

Functional Specification

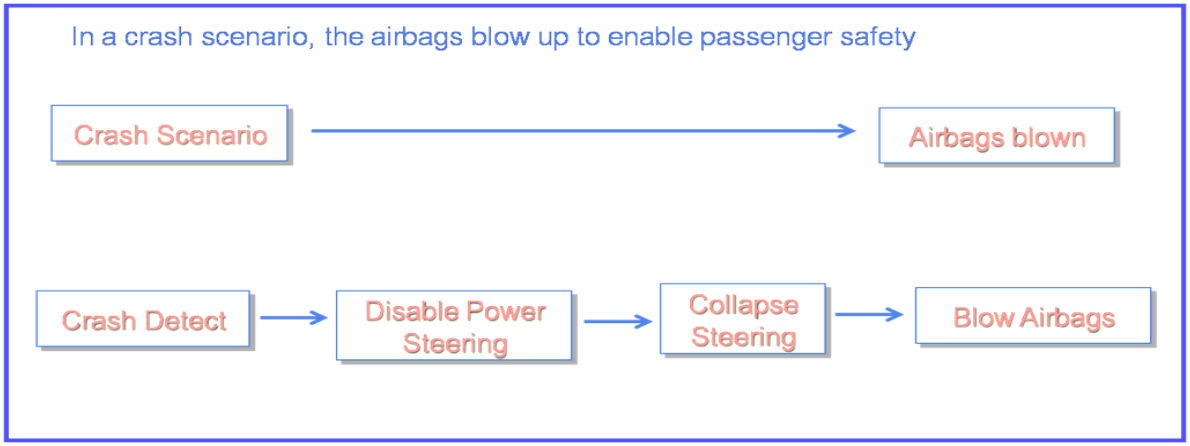


Safety Requirement



illustrative Example

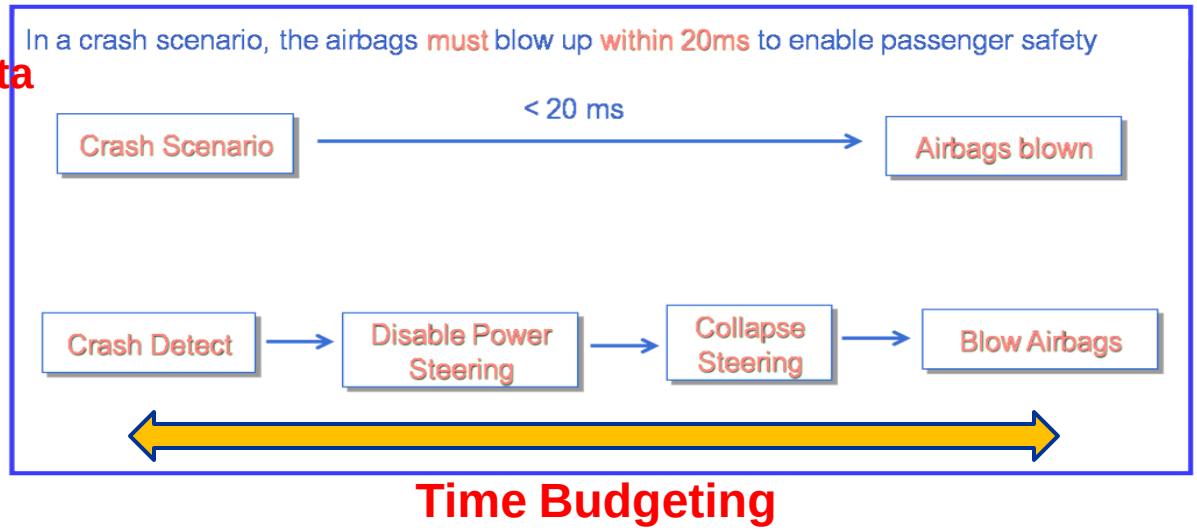
Functional Specification



Standards/Statistical data

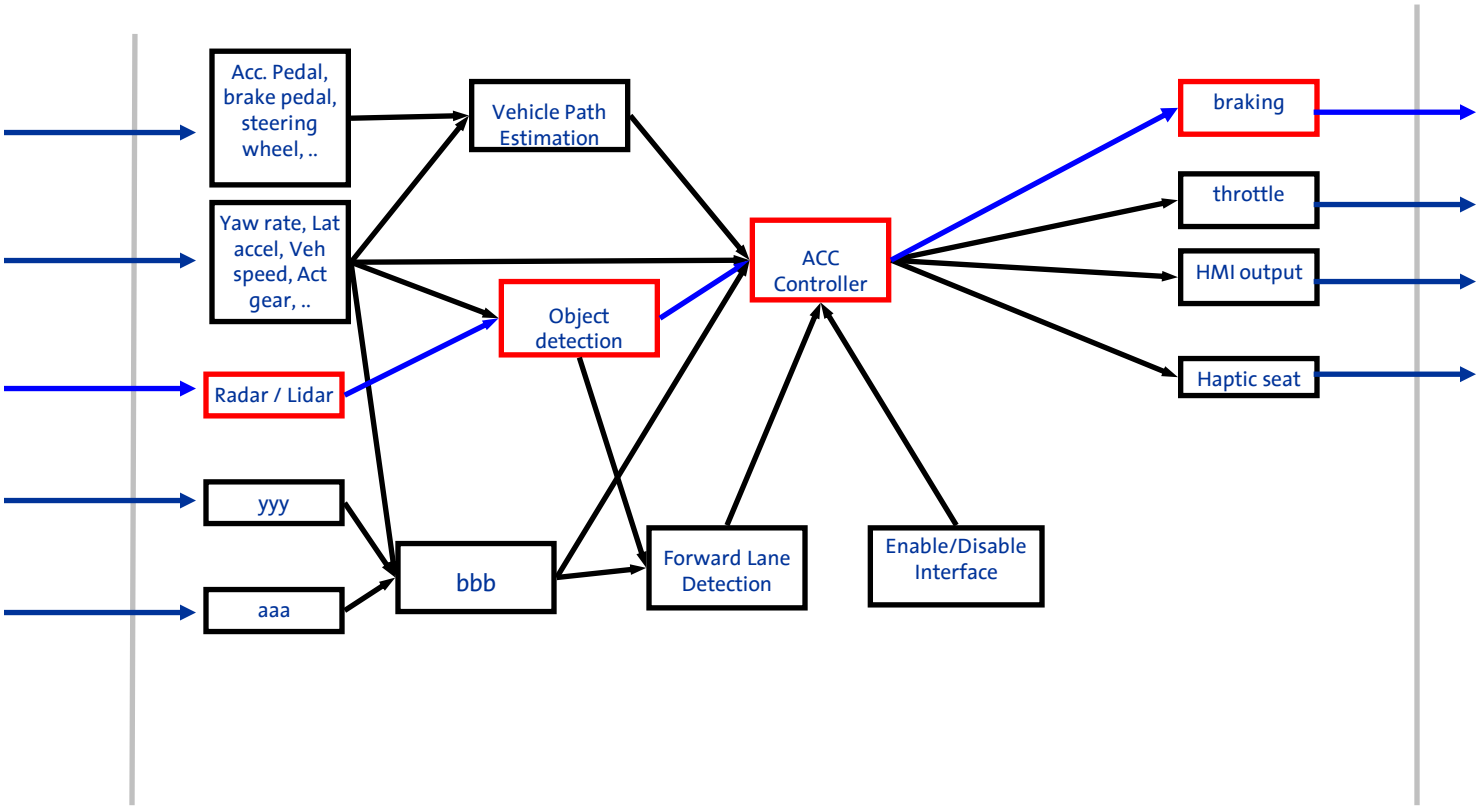
Safety Requirement

Actual Implementation



Time Budgeting

More General Interactions



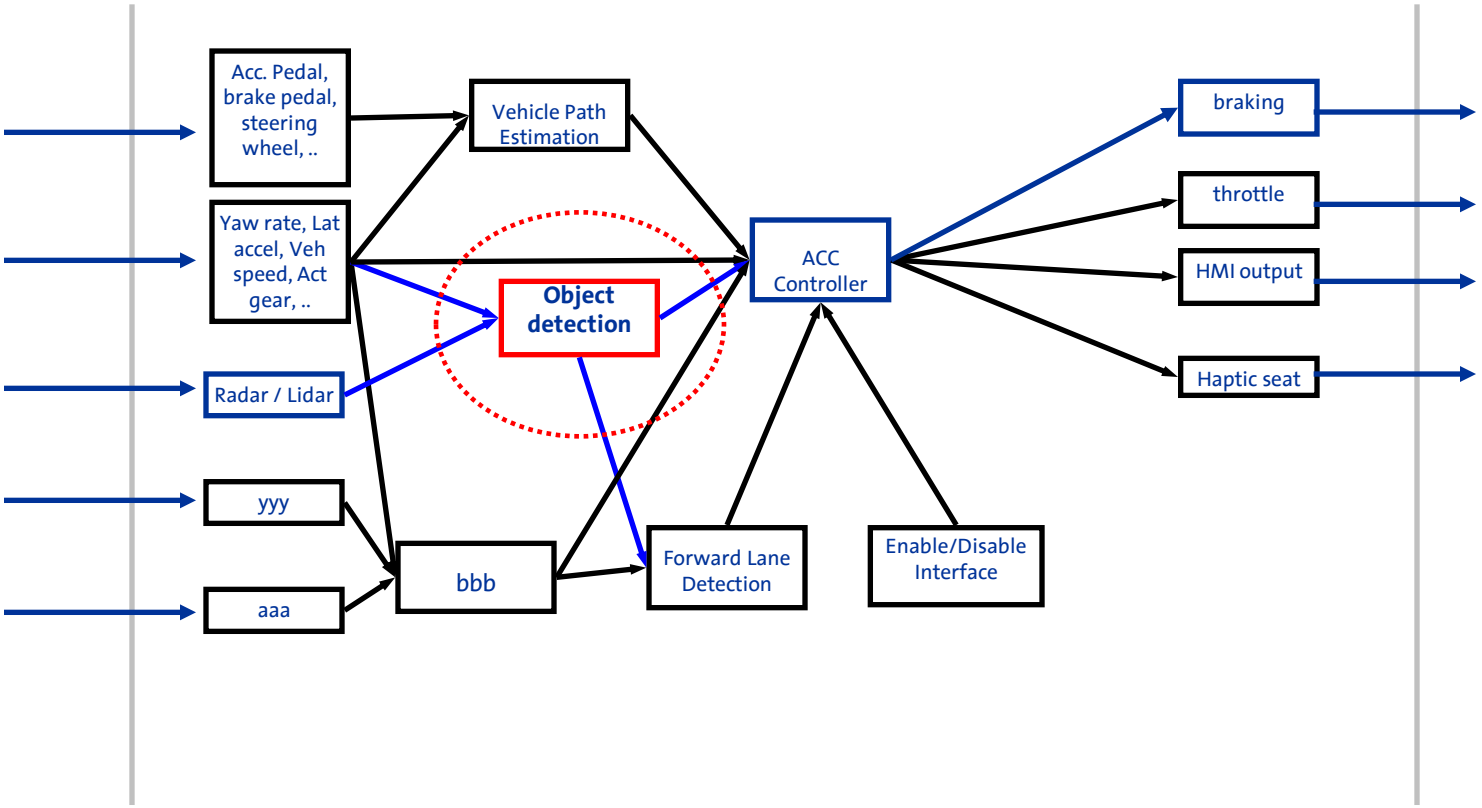
Sensors → Actuators

End-to-End Latency < 100 ms

Emerging Challenges

- **Increasing complex features**
- **Multiple functions in a single computational unit, e.g. AUTOSAR**
- **More component sharing promoted by the smaller component sizes**
- **Need for advance planning of resources for extensibility**
- **This is leading to...**
  - Increasing Real-time interdependencies between components

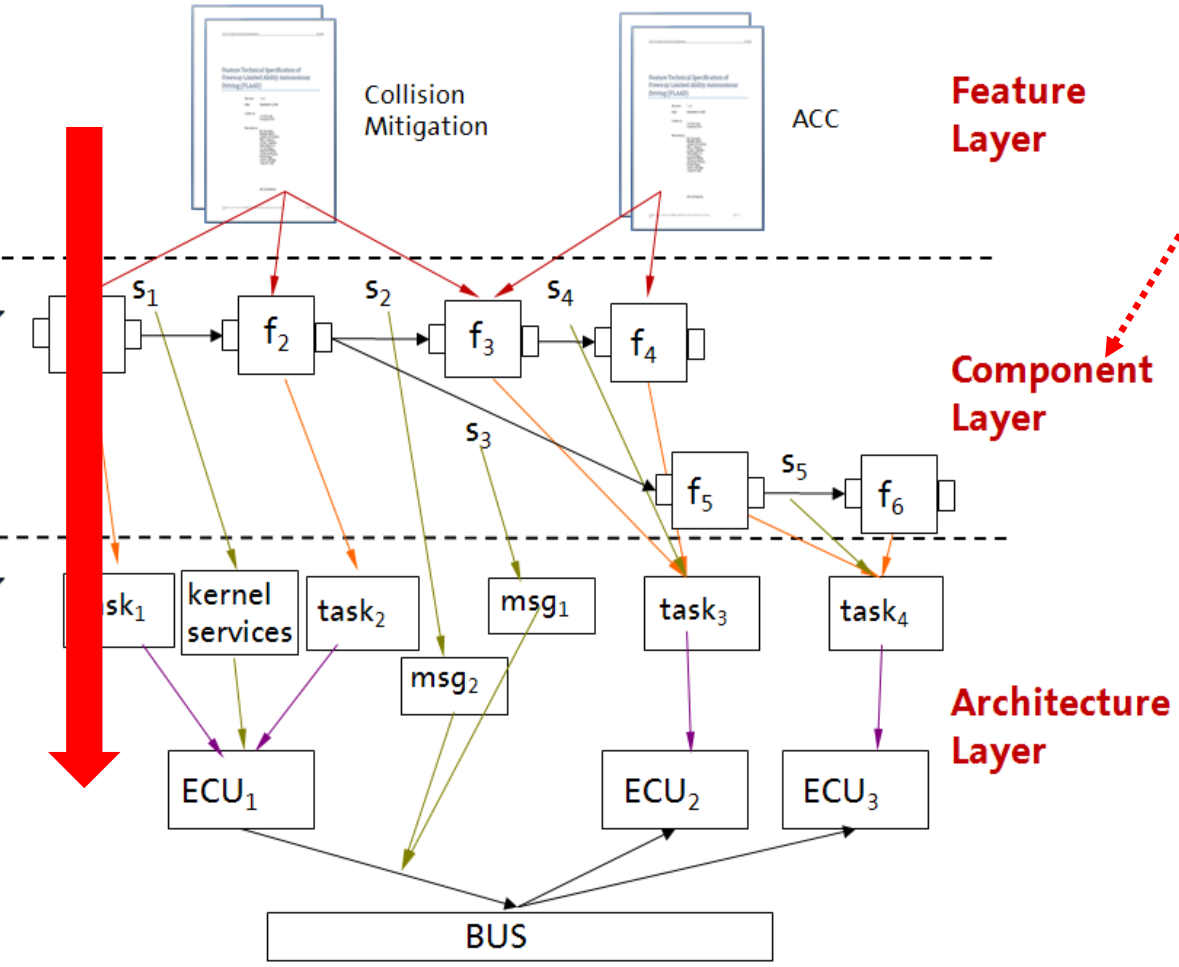
Example: Timing Dependencies



Sensors → Actuators

End-to-End Latency < 500 ms

Prevalent Approaches

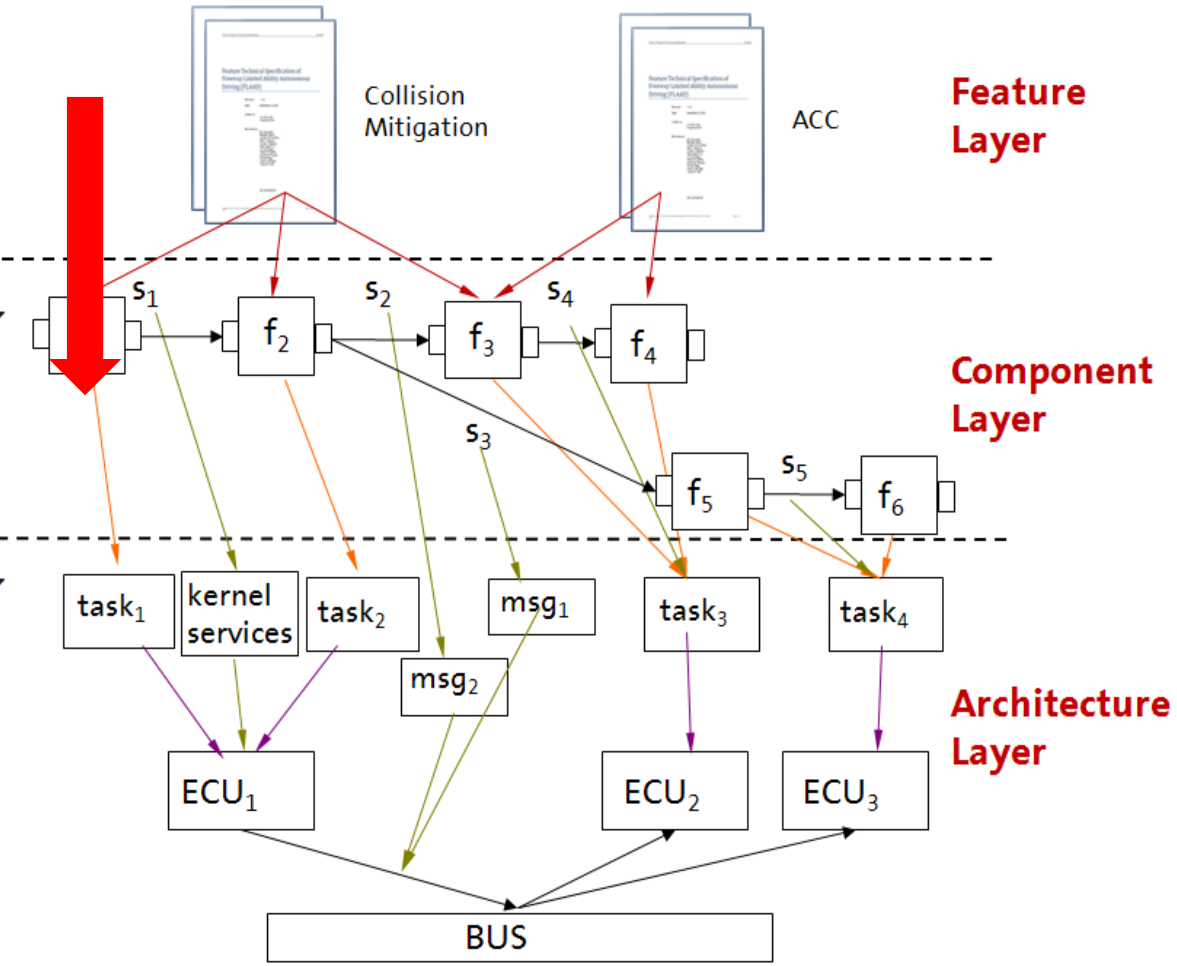


- Ad-hoc estimates about component response time
- Architecture exploration to do component-task mapping, component-ECU mapping etc.
- On failure, difficult to trace the culprit component

How to budget time for each component is not clear

Part of the figure from:
Embedded System Design for Automotive Applications, Marco Di Natale et.al.
IEEE Computer2007,

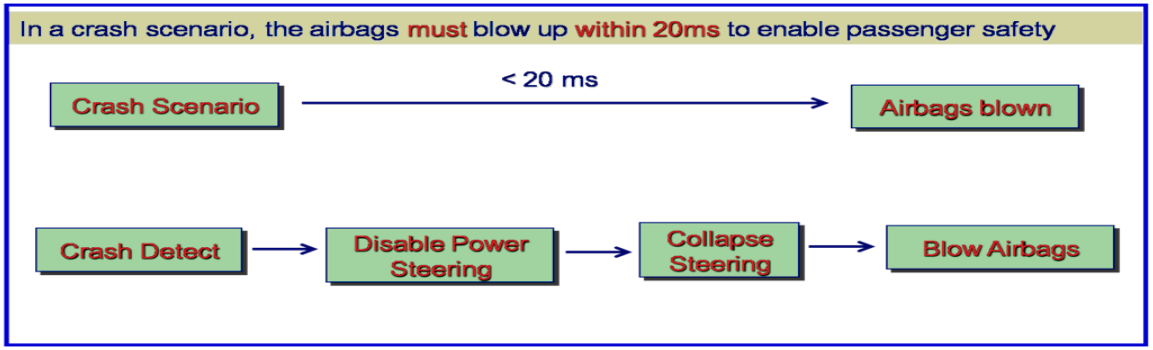
Our Proposal

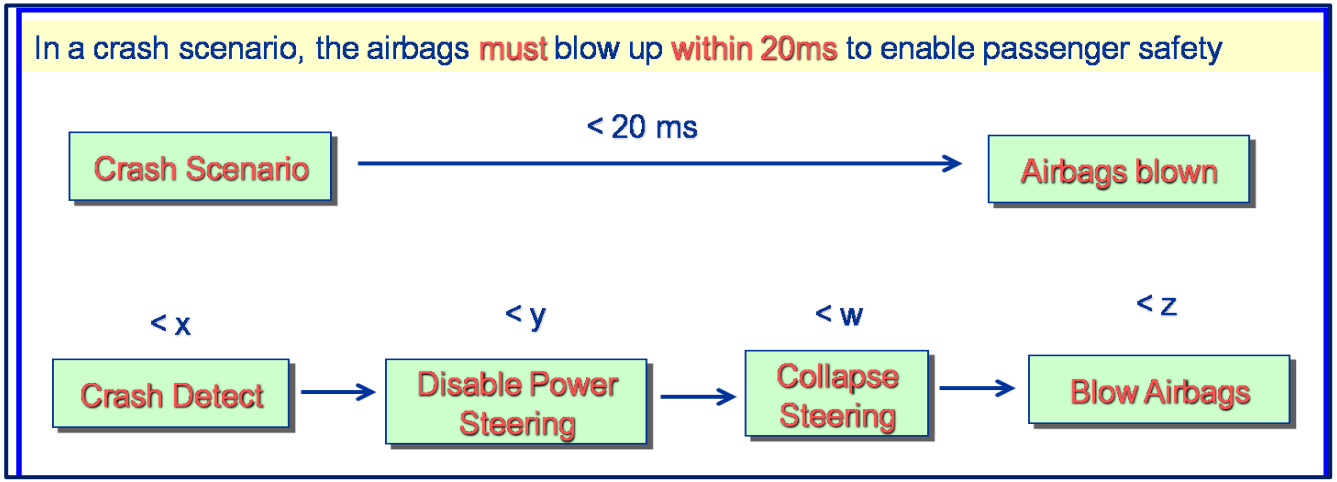


Part of the figure from:
Embedded System Design for Automotive Applications, Marco Di Natale et.al. IEEE Computer2007,

- **Rigorous-Systematic methodology**
- **To be applied during early stages at the requirements level**
- ***Timing requirements are captured in parametric form***
- ***Use formal specification and analysis methodology to generate constraints on parameter valuations***

How we propose to do this?





Parametric Specifications

Time-Budgeting: What values of x, y, w, z are good-enough?

The Problem

- We are given a set of features and their real-time requirements
- We are given a set of components and their parametric-time requirements for implementing these features
- Propose Early stage Time-Budgeting Methodology
  - Find constraints over parameter values
  - Design space exploration to select suitable valuation
  - Scalable

Our Contributions

1. Manoj G. Dixit, Pallab Dasgupta and S. Ramesh. Taming the Component Timing: A CBD Methodology for Real-time Embedded Systems. *Design Automation and Test in Europe*, pages 1649-1652, 2010.

2. Manoj G. Dixit, S. Ramesh and Pallab Dasgupta. Some Results on Parametric Temporal Logic. *Information Processing Letters*, Elsevier, Accepted for Publication.

3. Manoj G. Dixit, S. Ramesh, and Pallab Dasgupta. Parametric Analysis of Real-time Response Guarantees on Interacting Software Components, *World Intellectual Property Organization*, Publication Number: WO/2009/129089, International Application No.: PCT/US2009/039837

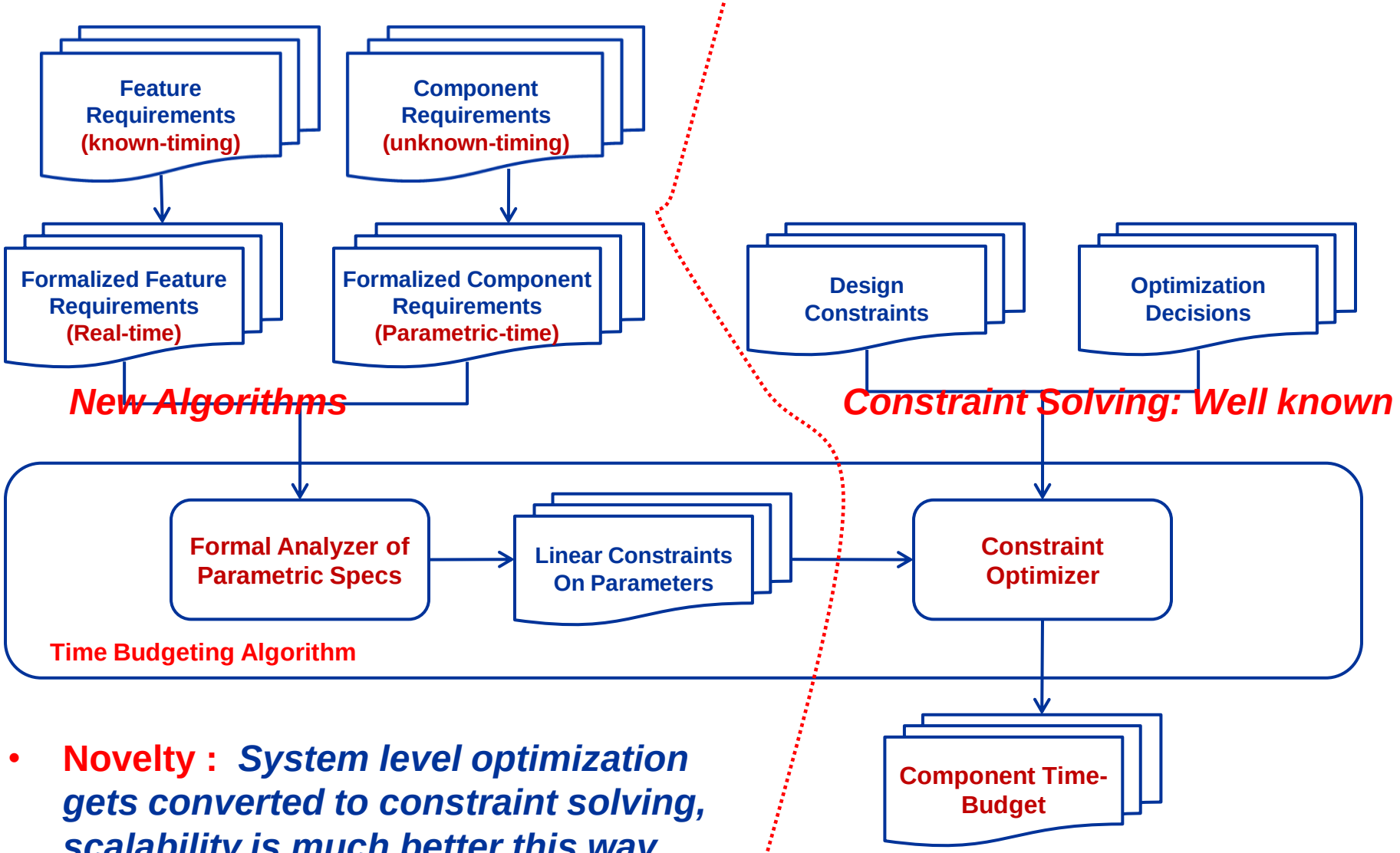
4. Manoj G. Dixit, S. Ramesh and Pallab Dasgupta. A Prune and Search Approach for an Early Time-Budgeting in Real-Time Systems. Under Prepration.

5. Manoj G. Dixit, S. Ramesh and Pallab Dasgupta. Hierarchical Time-Budgeting using Parameterized Bounded-Response Requirements. Under Prepration.

Hierarchical Time-Budgeting Methodology

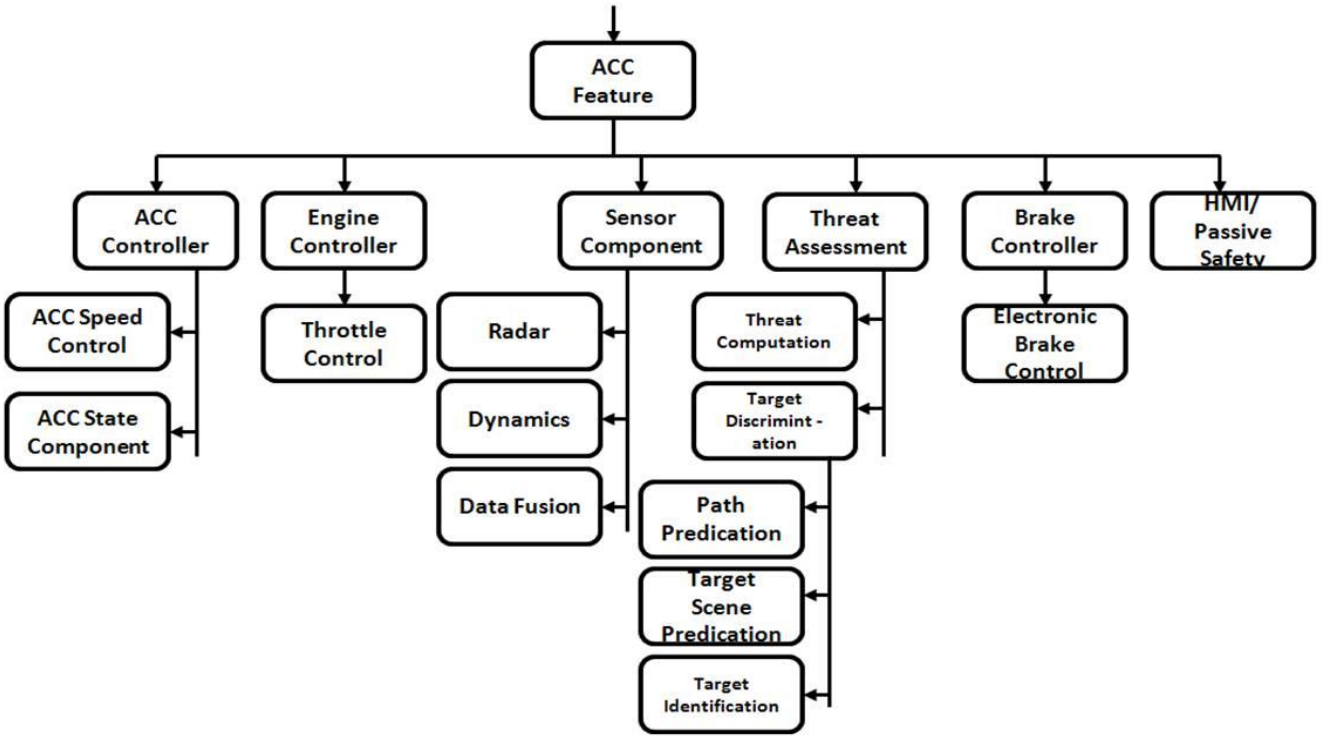
- **Preliminary version in DATE 2010**
- **Patent under review**

Time-Budgeting Methodology



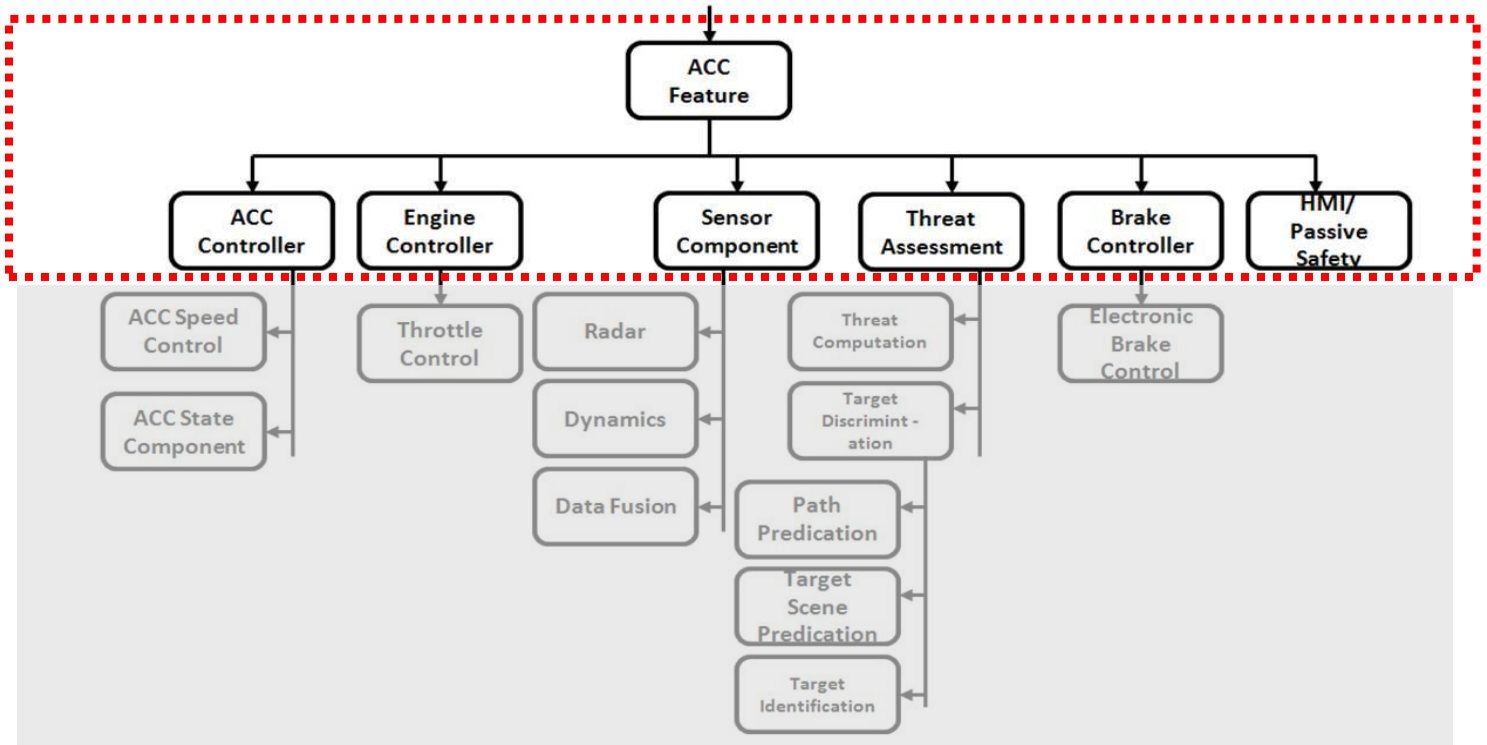
- **Novelty :** ***System level optimization gets converted to constraint solving, scalability is much better this way***

Time-Budgeting Challenges



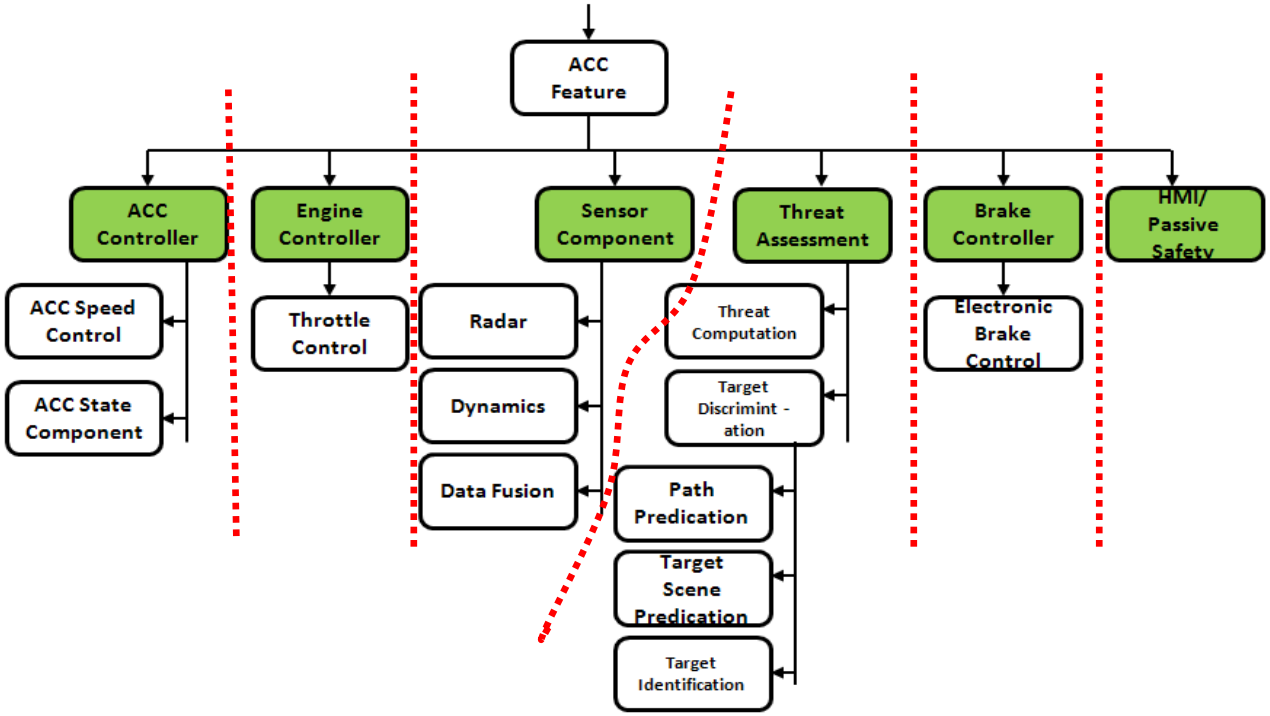
- Component Decompositions are hierarchical: DAG
- Need to budget timing for all components
- Each feature component has 10s of requirements
- *simultaneous budgeting not advisable*

Hierarchical Time-Budgeting Algorithm



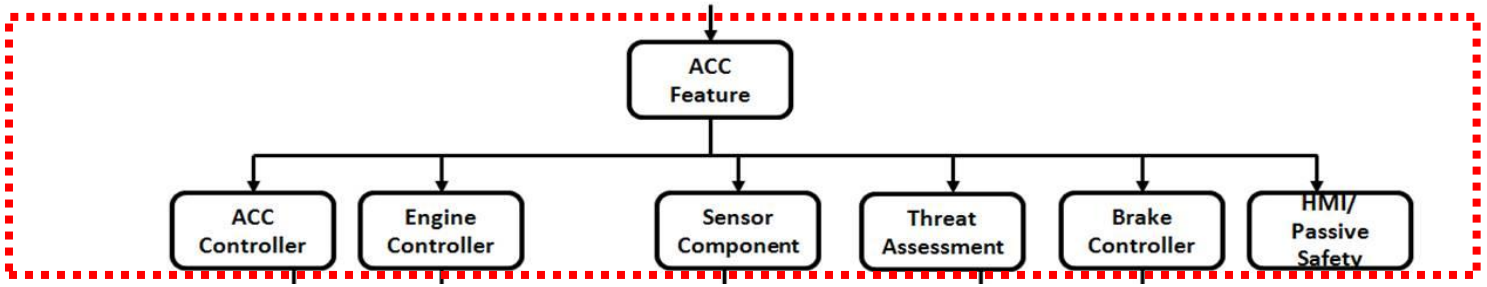
- **Budget time for all top level components from feature requirements**
- **Requirements of features and top level components are analyzed together**

Hierarchical Time-Budgeting Algorithm



- Split component time-budgeting into smaller subproblems and repeat
- Compositional approach
- DF-traversal with back-tracking takes care of component re-use case

Main Step in Time-Budgeting

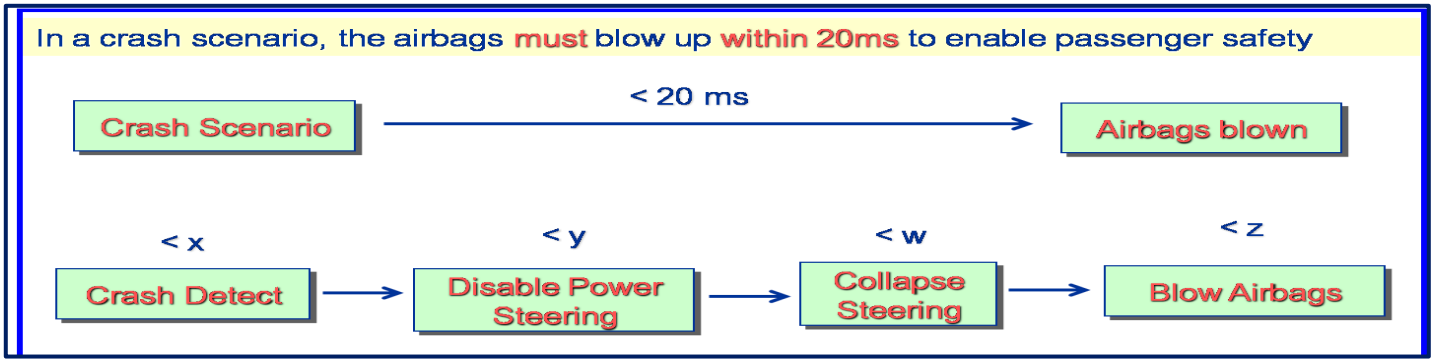


- A collection of real-time requirements which is refined into a collection of parameterized requirements
- Find suitable parameter valuation so that, the instantiated requirements satisfy all real-time requirements from which they have been derived
- Actually a standard verification check!! But reduces to constraint synthesis here!

Formalization of Refinement Step

- **Requirements refinement step is formalized as a collection of *requirement decomposition pairs***
- ***f* : feature requirement**
- **$g_1, \ldots, g_k$ are component requirements identified for *f***
- **$(f, \{g_1, \ldots, g_k\})$ is a requirement decomposition pair**

Theorem: It is enough to analyze each pair separately, compute validity constraint. Any solution to conjoined constraint defines a suitable time-budget

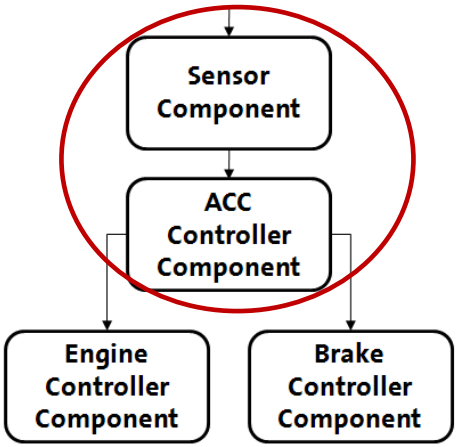


Formal Specification for Requirements

- **Parametric Linear Temporal Logic (PLTL)**
  - Extends well known Linear Temporal Logic
  - Semantics is defined by using a parameter valuation

Feature/Component Name	PLTL Formula
ACC Feature	$\phi_1 : \Box(lead_slow \Rightarrow \Diamond_{\leq 500}\ apply_brake)$
Sensor Component	$\psi_1 : \Box(lead_slow \Rightarrow \Diamond_{\leq x_1}\ lead_kinematics_info)$
ACC Controller Component	$\psi_2 : \Box(lead_kinematics_info \Rightarrow \Diamond_{\leq x_2} apply_brake)$

$(\phi_1, \{\psi_1, \psi_2\})$ *is the RDP*

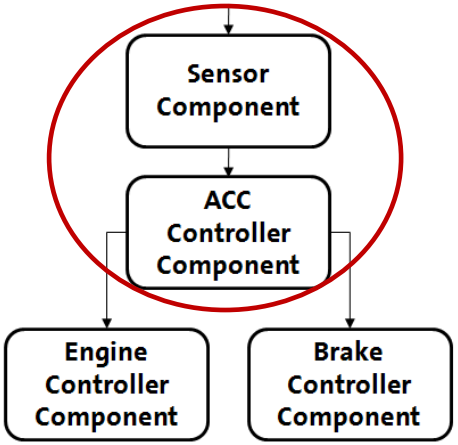


Validity of a Requirement Decomposition Pair

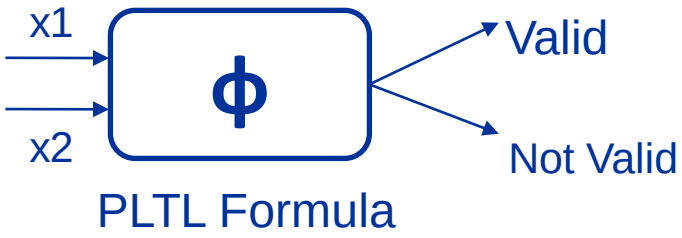
Feature/Component Name	PLTL Formula
ACC Feature	$\phi_1 : \Box(lead_slow \Rightarrow \Diamond_{\leq 500}\, apply_brake)$
Sensor Component	$\psi_1 : \Box(lead_slow \Rightarrow \Diamond_{\leq x_1}\, lead_kinematics_info)$
ACC Controller Component	$\psi_2 : \Box(lead_kinematics_info \Rightarrow \Diamond_{\leq x_2} apply_brake)$

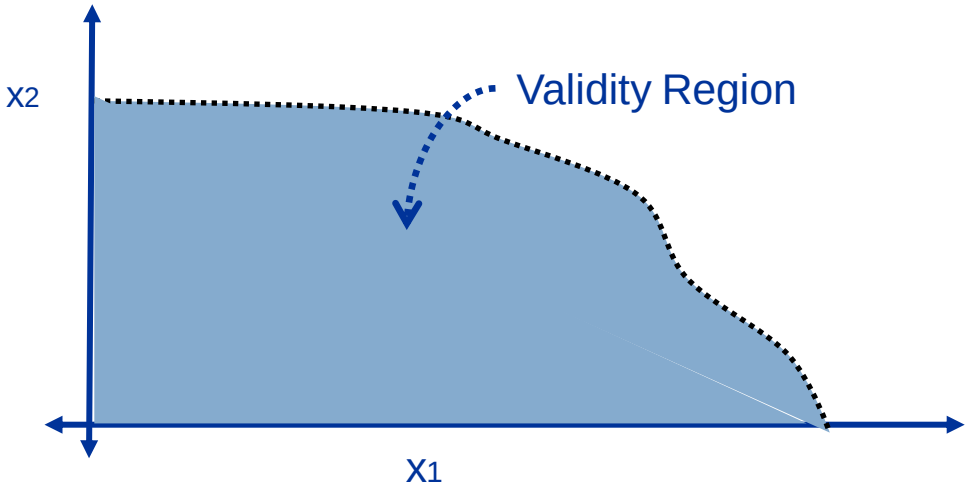
The requirement decomposition pair is *valid* if and only if PLTL formula $\psi_1 \wedge \psi_2 \Rightarrow \phi_1$ is valid.

Due to the parameters, this reduces to constraint computation



Abstractly…

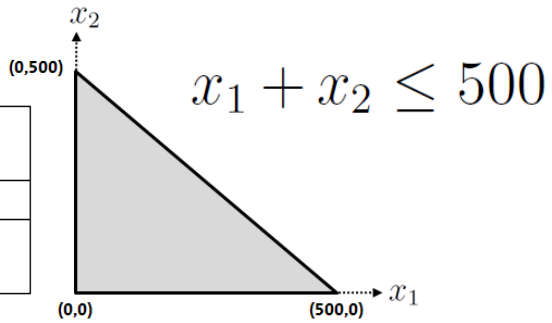




Given a PLTL formula ϕ, we want to find the representation of the solution region in the form of a constraint.

Validity Region Characteristics…

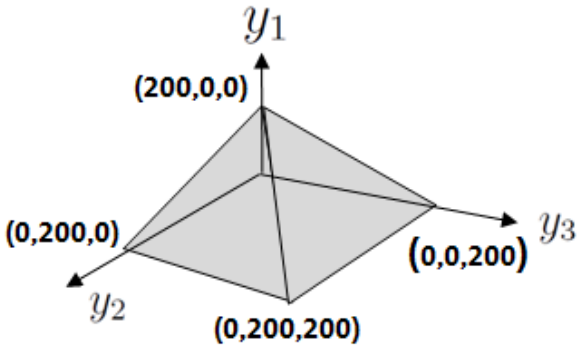
Feature/Component Name	PLTL Formula
ACC Feature	$\phi_1 : \Box(lead_slow \Rightarrow \Diamond_{\leq 500}\, apply_brake)$
Sensor Component	$\psi_1 : \Box(lead_slow \Rightarrow \Diamond_{\leq x_1}\, lead_kinematics_info)$
ACC Controller Component	$\psi_2 : \Box(lead_kinematics_info \Rightarrow \Diamond_{\leq x_2} apply_brake)$



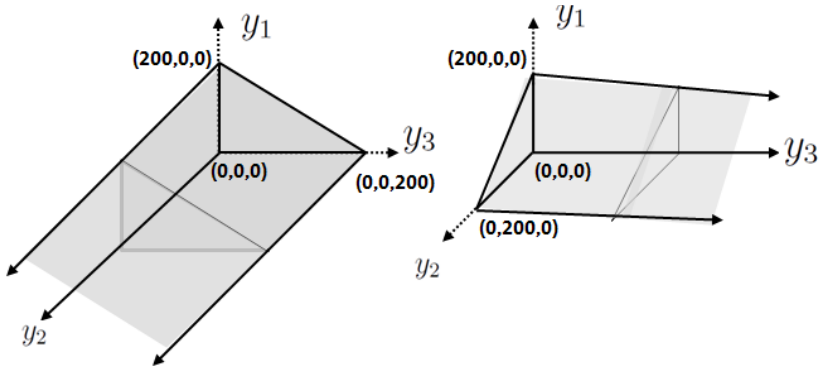
ACC Feature	$\phi_2 : \Box(apply_brake \Rightarrow \Diamond_{\leq 200}\, host_decel)$
ACC Controller Component	$\psi_3 : \Box(apply_brake \Rightarrow \Diamond_{\leq y_1}[(thrtl_adj) \bigvee (brake_adj)])$
Engine Controller Component	$\psi_4 : \Box(thrtl_adj \Rightarrow \Diamond_{\leq y_2} host_decel)$
Brake Controller Component	$\psi_5 : \Box(brake_adj \Rightarrow \Diamond_{\leq y_3} host_decel)$

AND replaces OR

$$(y_1 + y_2 \leq 200) \wedge (y_1 + y_3 \leq 200)$$



$$(y_1 + y_2 \leq 200) \vee (y_1 + y_3 \leq 200)$$



Validity Region: Some Takeaways

- **Validity region can be bounded/unbounded, convex/non-convex**
- **Different points in the region can satisfy different constraints- we need to combine them**
- **Does not merely depend on underlying component data-flow graph**
- **Linear predicates looks good representation**
  - but are they enough?

Emptiness, Universality and Finiteness of Validity Region

+

Unsuitability of Linear Predicates for Representing Validity Region

Information Processing Letters

Emptiness, Universality and Finiteness Problems

Formula Φ	Emptiness			Universality			Finiteness		
	F	S_Φ	V_Φ	F	S_Φ	V_Φ	F	S_Φ	V_Φ
PLTL$_\Box$	$\{\Phi(\alpha_0)\}$	$\Leftrightarrow$, sat	$\Leftrightarrow$, val	$\{\widetilde{\Phi}\}$	$\Rightarrow$, sat	$\Leftrightarrow$, val	$\{\widetilde{\Phi(0_{Y\setminus\{y\}})} \mid y \in Y\}$	$\Rightarrow$, any member sat	$\Leftrightarrow$, any member val
PLTL$_\Diamond$	$\{\widetilde{\Phi}\}$	$\Leftrightarrow$, sat	$\Rightarrow$, val	$\{\Phi(\alpha_0)\}$	$\Leftrightarrow$, sat	$\Leftrightarrow$, val	$\{\widetilde{\Phi}\}$	$\Leftrightarrow$, sat	$\Rightarrow$, val
PLTL	$\{\widetilde{\Phi(0_Y)}\}$	$\Leftrightarrow$, sat	$\Rightarrow$, val	$\{\widetilde{\Phi(0_X)}\}$	$\Rightarrow$, sat	$\Leftrightarrow$, val	$\{\widetilde{\Phi(0_Y)}\}$	$\Leftrightarrow$, sat	$\Rightarrow$, val

α_0:	Parameter valuation assigning 0 to all parameters.
0_X:	Partial parameter valuation assigning 0 to all members of X.
0_Y:	Partial parameter valuation assigning 0 to all members Y.
$0_{Y\setminus\{y\}}$:	Partial parameter valuation assigning 0 to all members of $Y \setminus \{y\}$.
$\Leftrightarrow$:	Solution provides both necessary and sufficient condition.
$\Rightarrow$:	Solution provides only sufficient condition.
sat:	Satisfiability check required of a member formula.
val:	Validity check required of a member formula.

- **Defined *parameter abstraction operation* for PLTL**

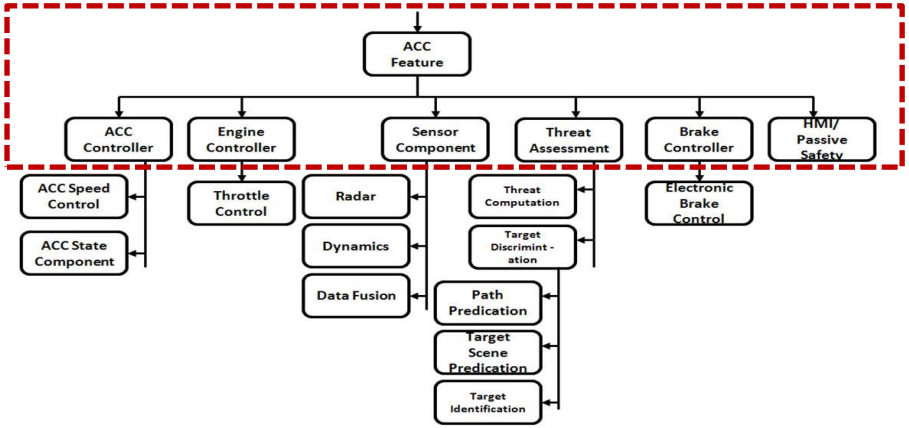
$$\Phi = \Box_{\leq y}\, p_1 \wedge \Box_{\leq 10}\, p_2 \qquad \widetilde{\Phi} = \Box\, p_1 \wedge \Box_{\leq 10}\, p_2$$

- **Improved Complexity**
  - Our complexity: $O(c_\Phi 2^{n_\Phi})$. Earlier: $O(c_\Phi^{k_\Phi+1} k_\Phi^{k_\Phi} 2^{n_\Phi(k_\Phi+1)})$

Bounded-Response Constraint Extraction Method

Under Submission to Journal

A Scalable Method for a widely used Requirement Pattern



$$\Box\left(\phi \Rightarrow \Diamond_{\leq x}\psi\right)$$

Bounded-response formula in PLTL

- At higher levels most of the requirements are based on a specific pattern... *bounded-response*

- ϕ : Antecedent formula
- ψ : Succedent formula
- x : Parameter or constant

- We consider validity checks of requirement decomposition pairs using this pattern

Constraint Extraction Method

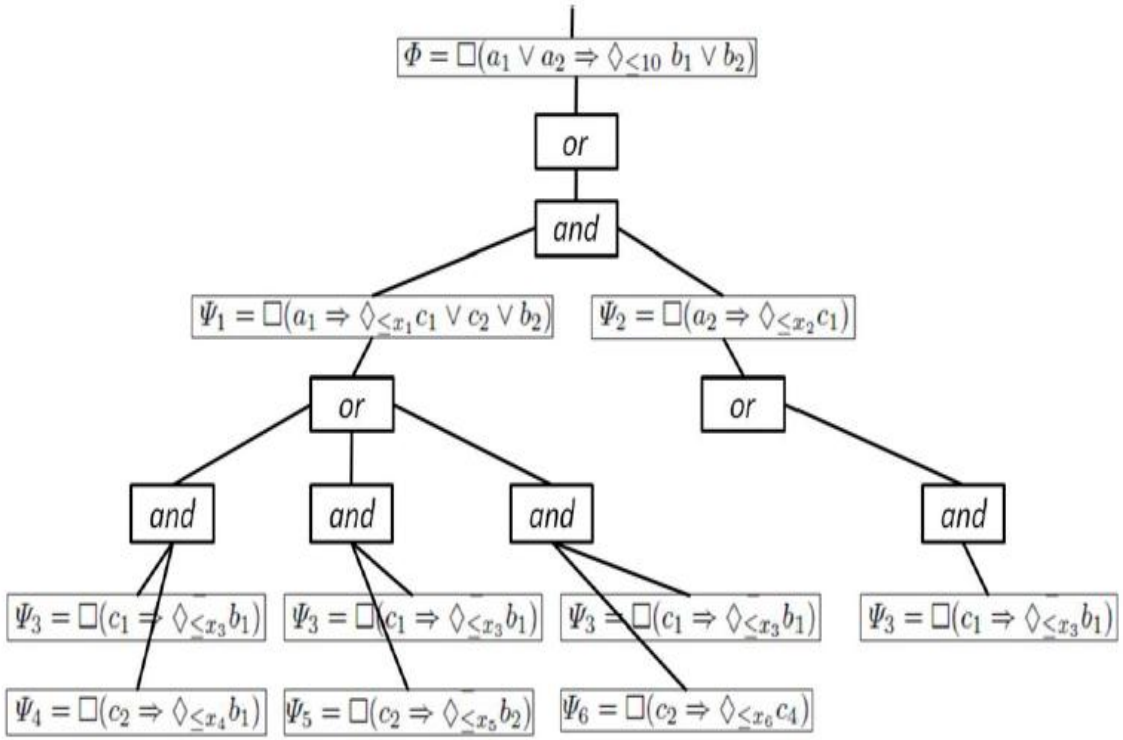
- **Reasoning over temporal formulae reduced to Boolean reasoning… hence scalable**
- **And-or tree constructed from formulae**
  - We have defined a notion of an *irreducible cover* for Boolean formulae
- **Assign path constraints**
- **Final constraint is conjunction/disjunction of path constraints**

Let γ and $\gamma_1, \cdots, \gamma_n$ be Boolean formulae.

An irreducible cover of γ is a minimal subset H of $\{\gamma_1, \cdots, \gamma_n\}$ such that $\gamma \Rightarrow \bigvee_{\gamma' \in H} \gamma'$ is valid.

Example: Bounded-Response Tree

$\Phi = \Box(a_1 \vee a_2 \Rightarrow \Diamond_{\leq 10}\ b_1 \vee b_2)$	$\Psi_1 = \Box(a_1 \Rightarrow \Diamond_{\leq x_1} c_1 \vee c_2 \vee b_2)$; $\Psi_2 = \Box(a_2 \Rightarrow \Diamond_{\leq x_2} c_1)$ $\Psi_3 = \Box(c_1 \Rightarrow \Diamond_{\leq x_3} b_1)$; $\Psi_4 = \Box(c_2 \Rightarrow \Diamond_{\leq x_4} b_1)$ $\Psi_5 = \Box(c_2 \Rightarrow \Diamond_{\leq x_5} b_2)$; $\Psi_6 = \Box(c_2 \Rightarrow \Diamond_{\leq x_6} c_4)$ $\Psi_7 = \Box(a_1 \wedge a_2 \Rightarrow \Diamond_{\leq x_7} c_3)$



$$\{[(x_1+x_3 \leq 10) \wedge (x_1+x4 \leq 10)] \vee [(x_1+x_3 \leq 10) \wedge (x_1+x_5 \leq 10)] \vee [(x_1+x_3 \leq 10) \wedge \texttt{false}]\} \wedge (x_2+x_3 \leq 10)$$

Corner Point Constraint Extraction Method

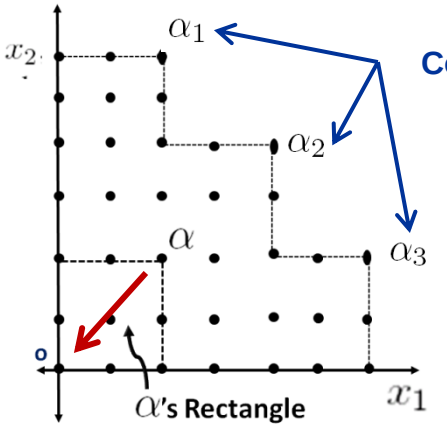
Preliminary Version in DATE 2010

Journal Version under Submission

A General Constraint Extraction Method for PLTL

$$\Box_{\leq m}\phi \Rightarrow \Box_{\leq (m-1)}\phi$$
$$\Diamond_{\leq m}\phi \Rightarrow \Diamond_{\leq (m+1)}\phi$$

Monotonicity of PLTL operators



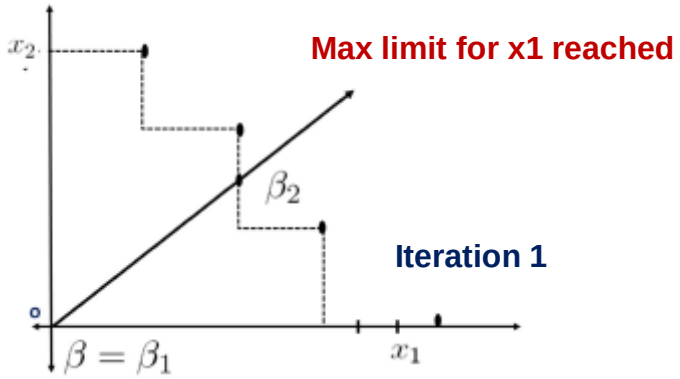
Downward Closure Property

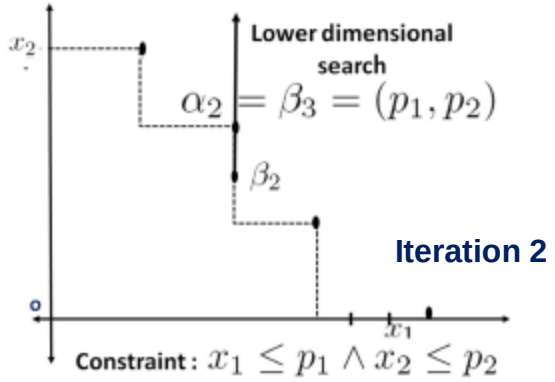
$$\bigvee_{i=1}^{3} x_1 \leq \alpha_i(x_1) \wedge x_2 \leq \alpha_i(x_2)$$

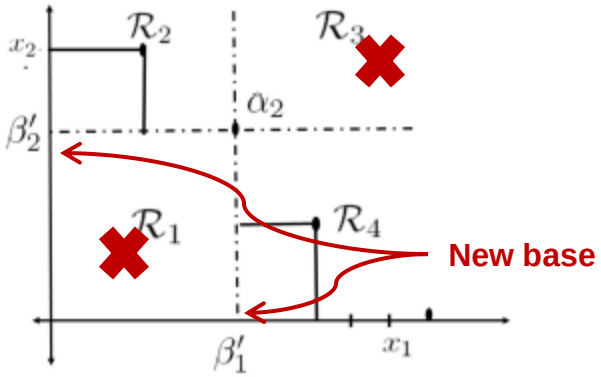
- **Suitable for complex temporal properties**
- **We focus on PLTL fragments and their geometric properties**
- **PLTL Global properties are downward-closed**
- **Downward Closed Region have finite number of *Corner-Points***
  - **Include Point-at-infinity**
  - **Constraint definition using Corner-Points**

Algorithm Overview

- **Prune and Search Approach**
  - Find a corner point
  - Partition the further search
- **Search Step**
  - Obtain a farthest useful valuation along diagonal starting from a *base*
  - Decide sub-set of parameters for which max limit is reached
  - Fix them and re-iterate till all parameters are over
- **Prune Step**
  - Partition and identify the region(s) where no corner-point can lie
  - Adjust new base valuation so that those regions get ignored from later search
  - Repeat Search step recursively for them

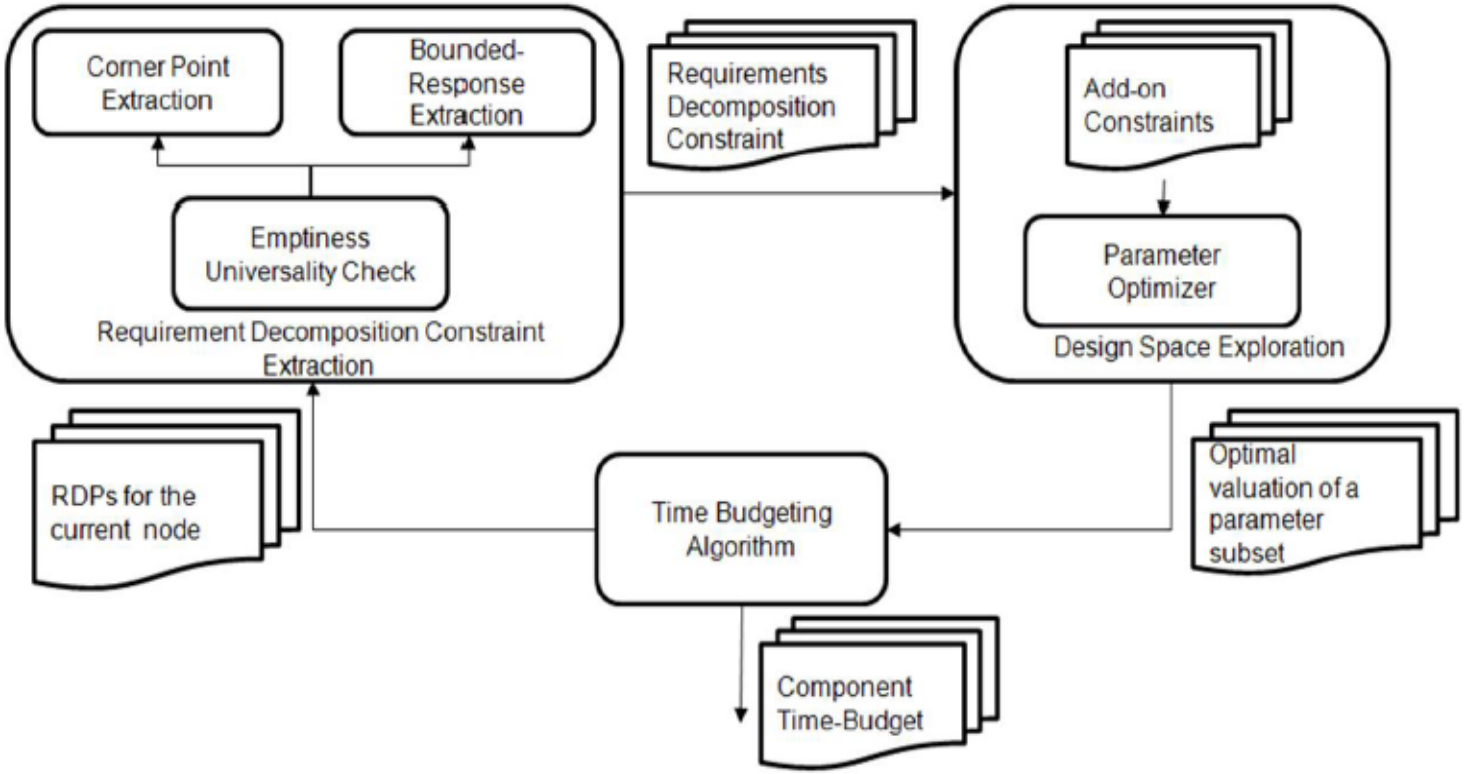






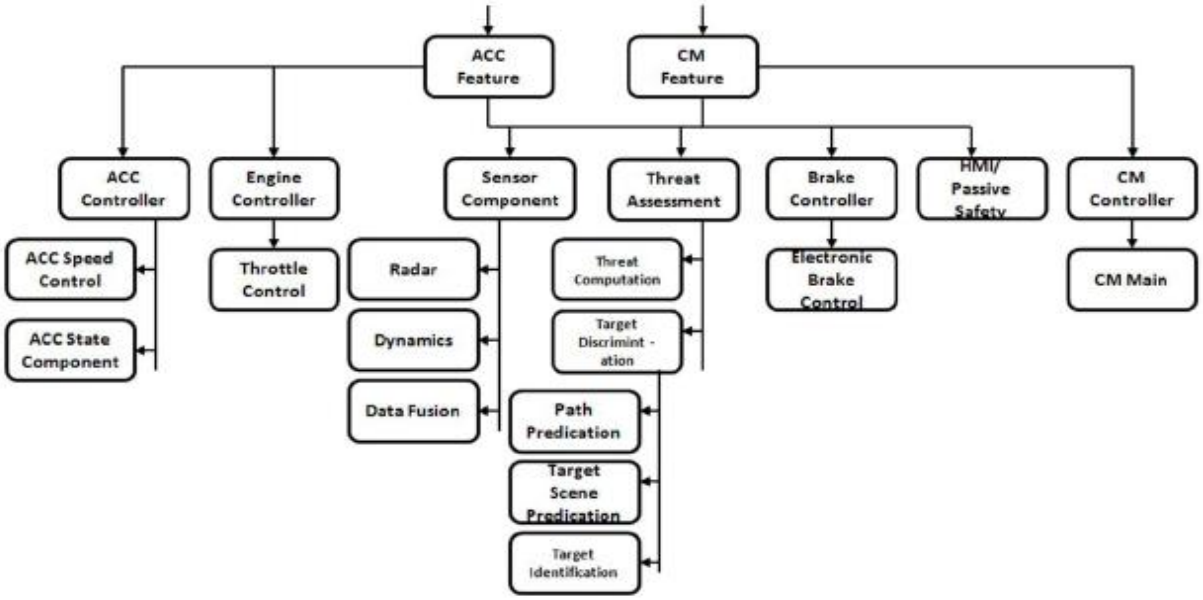
Demonstration of the Methodology

Integrated Time-Budgeting Methodology



- **NuSMV for LTL checks**
- **Yices for constraint solving**
- **Eclipse, Java**

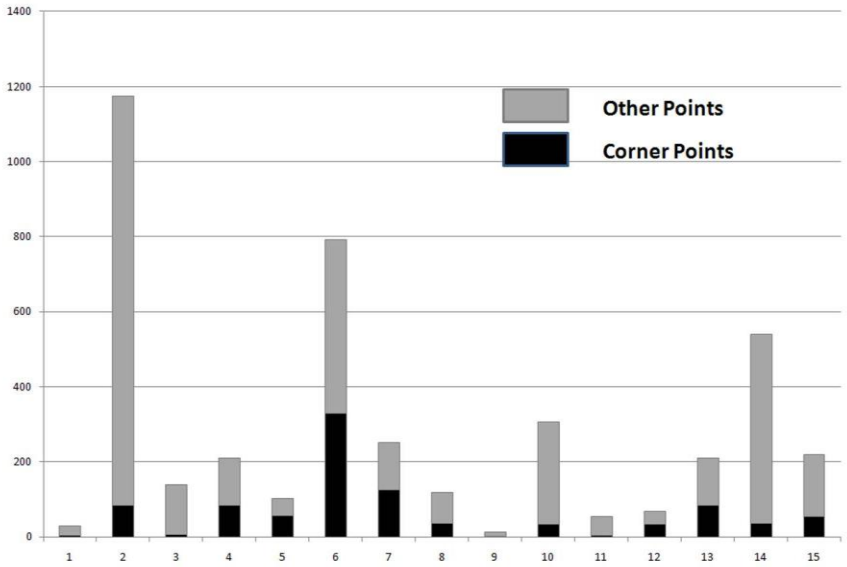
Case Studies



- **Adaptive Cruise Control, Collision Mitigation**
- **120+ feature and component properties**
- **100+ add-on constraints in the exploration**
- ***Budgeting for 3 feature combinations: ACC only, CM only and ACC-CM***

Some Results

Component	Parameter	Only ACC	Only CM	ACC-CM
Brake Controller Component	w_1	200	160	160
	w_2	70	-	70
	w_3	100	-	100
	w_4	100	-	100
	w_5	10	-	10
Electronic Brake Control	w_6	150	100	100
	w_{11}	35	20	20
	w_{12}	100	80	80
	w_{13}	30	20	20
	w_{14}	10	40	20
	w_{15}	100	100	100
	w_{16}	60	-	60
	w_{17}	60	-	60



Decomposition Type	Num Components	RDP Nos	Num. SAT Checks (Average)	Constraint Computation Time (Average) (sec)
Single Level	12	15	314 (LTL)	1280
Multi Level	22	11 (Bounded-Resp), 20 (General)	83 (Boolean SAT), 57 (LTL)	2.36 (Boolean SAT), 19.25 (LTL)

Summary of Contributions

- **A Hierarchical Time-Budgeting Methodology**
  - Integrates all of the below techniques
- **Emptiness, Universality and Finiteness Problems for PLTL**
  - Non-triviality of the solution region
- **Bounded-Response Constraint Extraction Method**
  - A specially tuned method for a widely used requirements pattern
- **Corner Point Constraint Extraction Method**
  - Complex temporal relationships
- **Case Studies**
  - Tool framework and demonstration on automotive features ACC and CM

Our Publications

1. Manoj G. Dixit, Pallab Dasgupta and S. Ramesh. Taming the Component Timing: A CBD Methodology for Real-time Embedded Systems. *Design Automation and Test in Europe*, pages 1649-1652, 2010.

2. Manoj G. Dixit, S. Ramesh and Pallab Dasgupta. Some Results on Parametric Temporal Logic. *Information Processing Letters*, Elsevier, Accepted for Publication.

3. Manoj G. Dixit, S. Ramesh, and Pallab Dasgupta. Parametric Analysis of Real-time Response Guarantees on Interacting Software Components, *World Intellectual Property Organization*, Publication Number: WO/2009/129089, International Application No.: PCT/US2009/039837

4. Manoj G. Dixit, S. Ramesh and Pallab Dasgupta. A Prune and Search Approach for an Early Time-Budgeting in Real-Time Systems. Under Prepration.

5. Manoj G. Dixit, S. Ramesh and Pallab Dasgupta. Hierarchical Time-Budgeting using Parameterized Bounded-Response Requirements. Under Prepration.

References

R. Alur, K. Etessami, S. La Torre, and D. Peled. Parametric Temporal Logic for "Model Measuring". *ACM Transactions on Computational Logic*, 2(3):388–407, 2001.

C. Bartolini, G. Lipari, and M. Di Natale. From Functional Blocks to the Synthesis of the Architectural Model in Embedded Real-time Applications. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 458–467, 2005.

M. Di Natale and A. L. Sangiovanni-Vincentelli. Moving From Federated to Integrated Architectures in Automotive: The Role of Standards, Methods and Tools. *Proceedings of the IEEE*, 98(4):603–620, 2010.

H. Blom, R. Johansson, and H. Lönn. Annotation with Timing Constraints in the Context of EAST-ADL2 and AUTOSAR - the Timing Augmented Description Language. In *Workshop on the Definition, evaluation, and exploitation of modelling and computing standards for Real-Time Embedded Systems*, 2009.

Thank You