Adaptive Dynamic Scheduling on Multi-functional Mixed-Criticality Automotive Cyber-Physical Systems

阅读报告

白洋

2017.06.09

论文信息

- Transactions on Vehicular Technology, PP(99):1-1.1 2017年录用
- 关键点：自适应调度，结合AUTOSAR adaptive platform的新特性、功能安全、混合关键级、 DAG

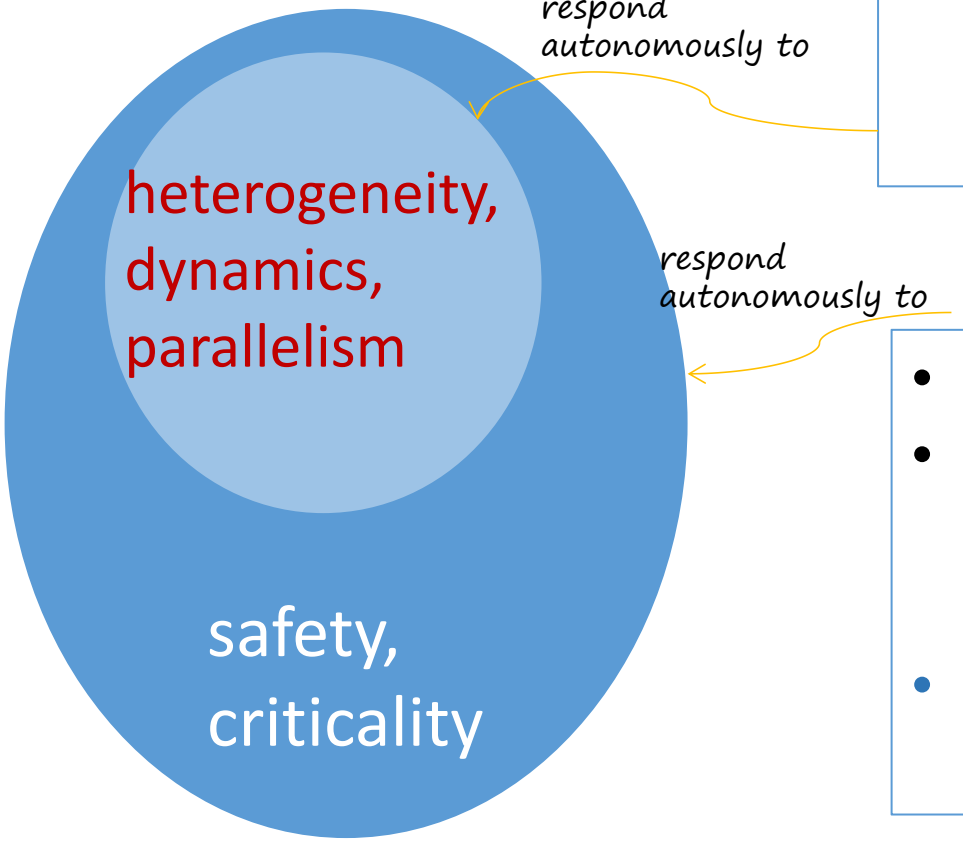
Adaptive Dynamic Scheduling on Multi-functional Mixed-Criticality Automotive Cyber-Physical Systems

Guoqi Xie, *Member, IEEE,* Gang Zeng, *Member, IEEE,* Zhetao Li, Renfa Li, *Senior Member, IEEE,* and Keqin Li, *Fellow, IEEE*

Outline

- Abstract
- Motivation
- Contribution
- Model
- *Fairness-based Dynamic Scheduling*
- *Adaptive Dynamic Scheduling*
- *Experiment*
- 思考 问题

Abstract



公平动态调度算法

- fairness-based dynamic scheduling algorithm FDS MIMF
- to minimize the individual makespans (i.e., schedule lengths) of functions from a high performance perspective

自适应动态调度算法

- adaptive dynamic scheduling algorithmADS MIMF
- to achieve low deadline miss ratios (DMRs) of safetycritical functions from a timing constraint perspective
- while maintaining the acceptable overall makespan of ACPS from ahigh performance perspective

However, ACPS should deal with joint challenges of heterogeneity, dynamics, parallelism, safety, and criticality, and these challenges are the key issues that will be solved in the next generation AUTOSAR adaptive platform

Motivation

- 有效调度：work for 充分利用ECU、达到高的性能

- 新挑战：
- 1. Functions 变复杂：周期性释放＋动态释放 （interact with environment） =>静态调度要变为动态调度
- 2. 动态释放的function并不只是来源于driver assistance，也来自安全相关的动态控制、主动安全和被动安全系统
- 3. AUTOSAR adaptive platform 在基础软件架构上进行的改变也反映了heterogeneity, dynamics, parallelism, safety, and criticality

Adaptive scheduling approach should be realized by adapting the architecture of ACPS at runtime to respond autonomously to changes in environments or within themselves.

Contributions

- 1. 模型
- 2. 公平动态调度算法
- 3.自适应动态调度算法

Outline

- Abstract
- Motivation
- Contribution
- Model
- *Fairness-based Dynamic Scheduling*
- *Adaptive Dynamic Scheduling*
- *Experiment*
- 思考 问题

模型

- *A. Architecture*
  - 集成式CAN集群：CAN因其event-triggered特性，天然适合dynamic
- *B. Criticality Level*
  - *ISO26262*标准定义：A B C D 四个等级；严重性、可靠性、可控性
  - 只考虑严重性，安全关键级退化为S = {S0, S1, S2, S3}
- *C. Mixed-criticality Function Model*
  - Fm =(N , W , M , C)； remaining attributes (arrivaltime, criticality, lowerbound, deadline, and makespan)
- *D. Mixed-criticality System Model*
  - M S = {F1, F2,...,F|MS|}. 假设是非抢占式调度
  - **Dynamic模式怎么来？**
- *E. Motivating Example*
- *F. Problem Description*

模型

- *D. Mixed-criticality System Model*
  - M S = {F1, F2,...,F|MS|}. 假设是抢占式调度
  - *Dynamic模式怎么来?*
  - To implement the requirement of **assigning tasks to different ECUs in a dynamic fashion**, **we emphasize the following conditions**:
  - 1) source code for each task needs to be presented on each ECU
  - 2) the data for the task needs to be available on the ECU which requires **dynamic sending of data via messages** 没看明白
  - 3) all ECUs have to be certified to the highest criticality level;
  - 4) a TIER 1 supplier of one ECU has to allow the execution of a task possible designed by another TIER 1 supplier which raises questions regarding liability.(遵循标准即可自动满足)

模型

- *E. Motivating Example* 多ECU，多功能，多关键级

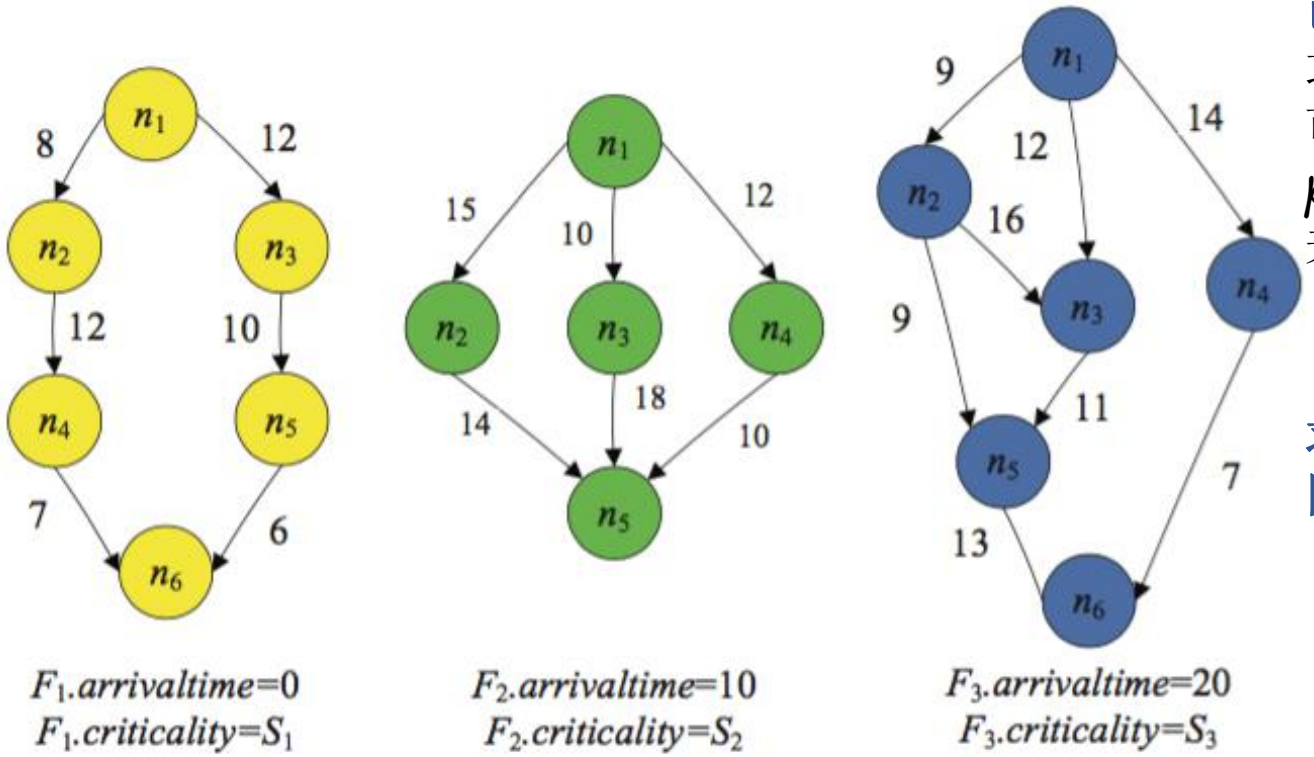


Fig. 2: Motivating example of ACPS containing three distributed functions with different criticality levels ($F_1.criticality = S_1$, $F_2.criticality = S_2$, and $F_3.criticality = S_3$).

已知：

功能集合MS = {F1, F2,…,F|MS|}

可执行功能的异构ECU 集合 P = {p1, p2, …, p|P |}

关键级集合S = {S0, S1, S2, S3}

求：

同时降低总体makespan和高关键级死线错失率

Outline

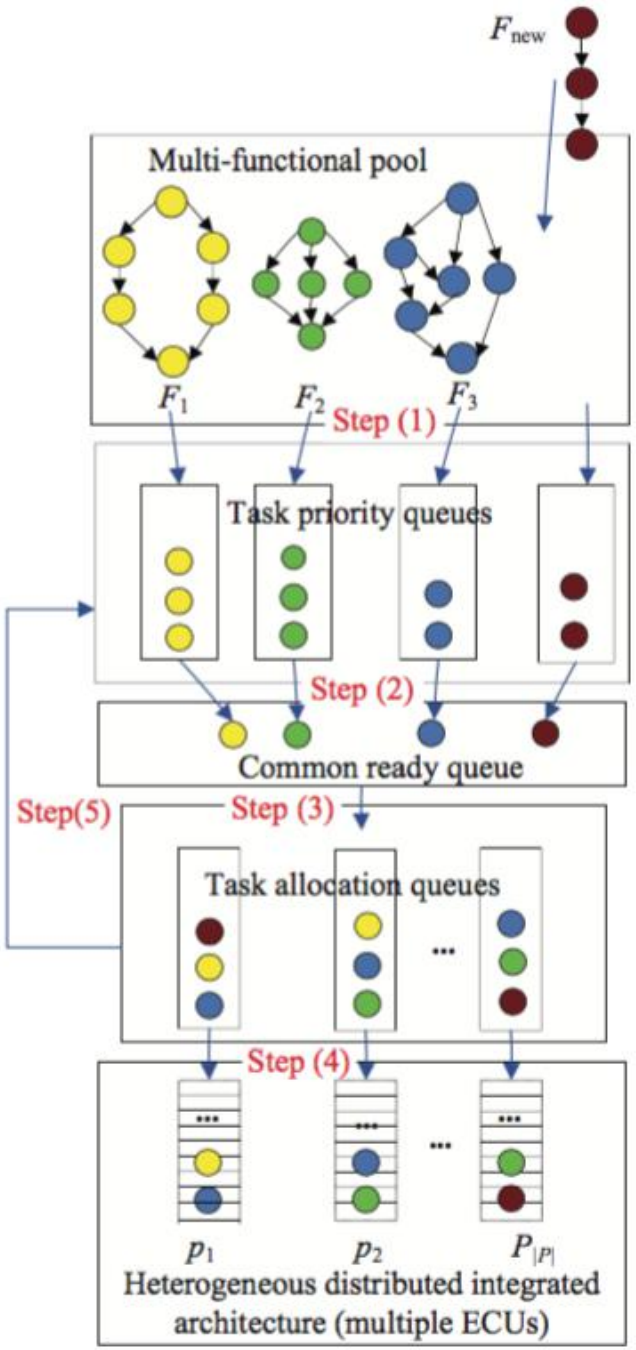
- Abstract
- Motivation
- Contribution
- Model
- Fairness-based Dynamic Scheduling
- *Adaptive Dynamic Scheduling*
- *Experiment*
- 思考 问题

FAIRNESS-BASED DYNAMIC SCHEDULING

- *A. Lower Bound and Deadline*
  - 根据已有算法计算优先级、lowerbound、dealine（相对值由认证机构提供）
- *B. Dynamic Scheduling Framework*
- *C. Fairness-based Dynamic Scheduling Algorithm*
- *D. Example of the FDS MIMF Algorithm*

FAIRNESS-BASED DYNAMIC SCHEDULING

- *Dynamic Scheduling Framework*
  - *两个组成部分：1. 功能池 2. ECU集合*
- 三个队列：
  - 1) The task priority queue （对每一个功能来说）
    - ordered according to descending ranku(Fm.ni).
  - 2) The common ready queue (对整体)
    - storing ready tasks
    - also ordered according to descending ranku(Fm.ni)
  - 3) The task allocation queue (对每一个ECU来说)
    - of each ECU is
    - for storing allocated tasks.



FAIRNESS-BASED DYNAMIC SCHEDULING

- *Fairness-based Dynamic Scheduling Algorithm*

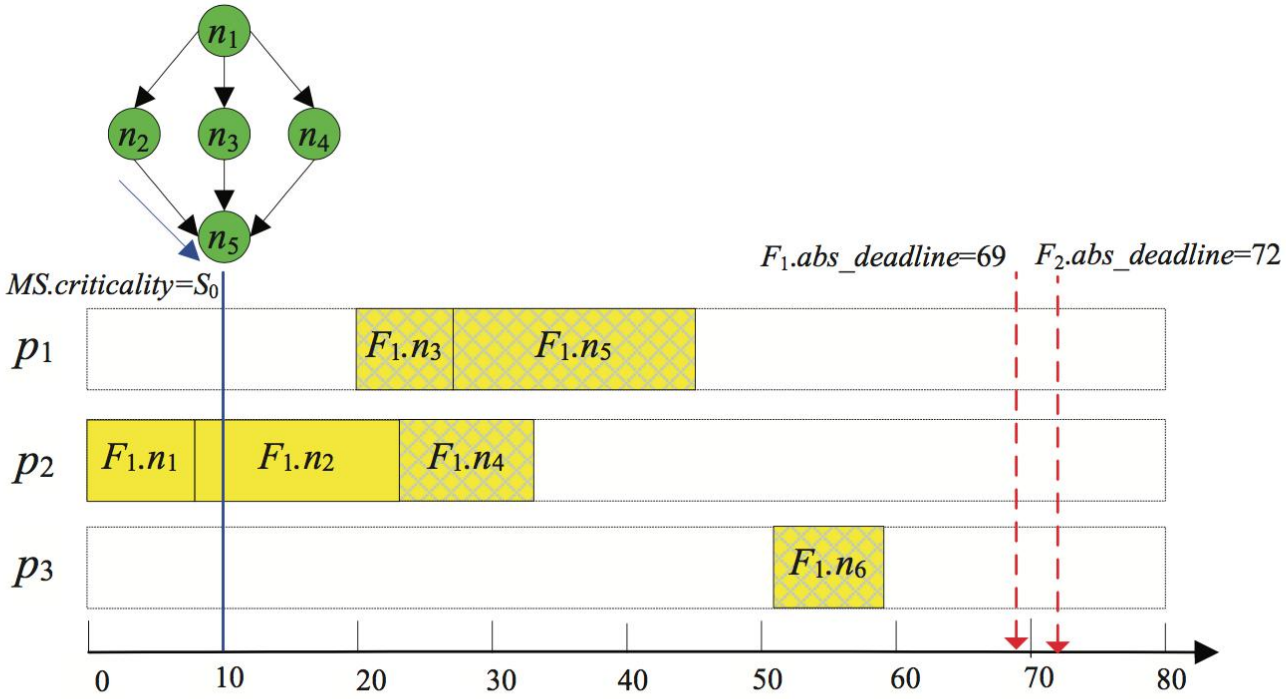


Fig. 5: Example of dynamic arrival, and the current time instant is 10 when F_2 arrives.

公平：为了使新功能可以即时地公平地加入调度过程，而不是等待上一轮调度结果完成

关键：Each task is first allocated to the task allocation queue of the ECU regardless of whether the ECU is idle when using our strategy (Step (4)), whereas FWDS needs to determine whether is idle and the task should wait for scheduling when the ECU is not idle

TABLE IV: Properties of functions in Fig. 2.

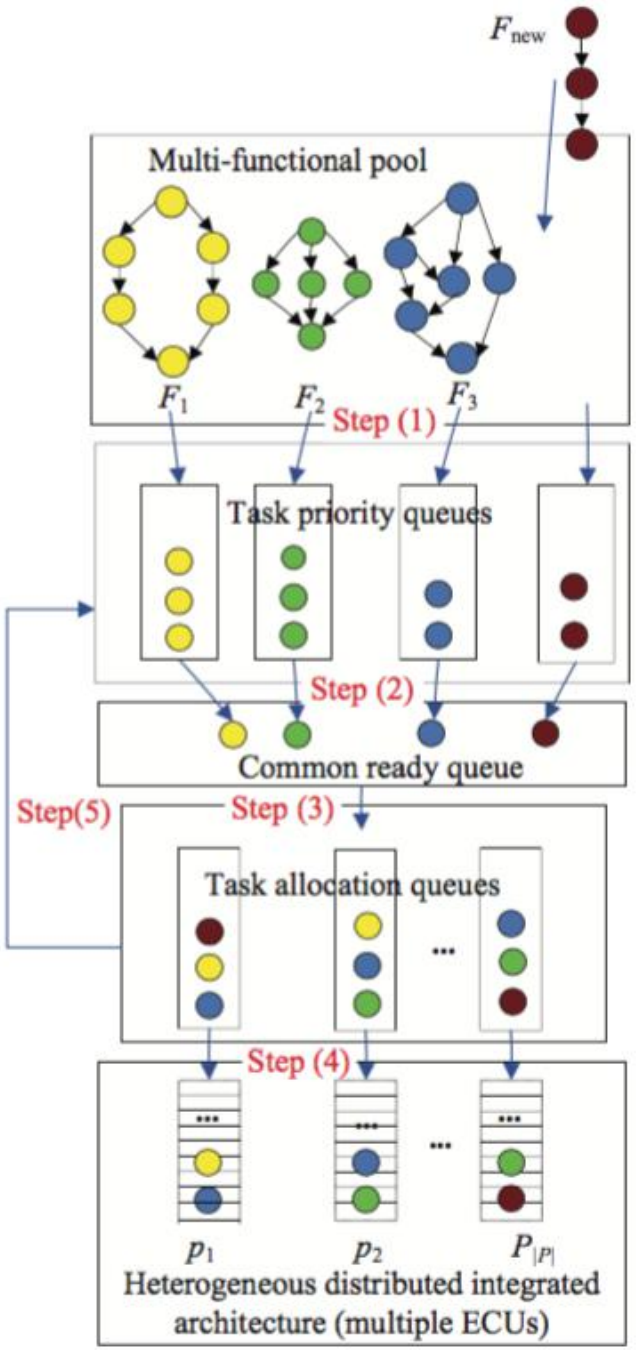
	F_1	F_2	F_3
Task priority	$F_1.n_1, F_1.n_2, F_1.n_3, F_1.n_4, F_1.n_5, F_1.n_6$	$F_2.n_1, F_2.n_3, F_2.n_4, F_2.n_2, F_2.n_5$	$F_3.n_1, F_3.n_2, F_3.n_3, F_3.n_5, F_3.n_4, F_3.n_6$
arrivaltime	0	10	20
criticality	S_1	S_2	S_3
lowerbound	59	52	54
deadline	69	62	64
abs_deadline	69	72	84

FAIRNESS-BASED DYNAMIC SCHEI

- *Fairness-based Dynamic Scheduling Algorithm*

公平：为了使新功能可以即时地公平地加入调度过程，而不是等待上一轮调度结果完成

关键：Each task is first allocated to the task allocation queue of the ECU regardless of whether the ECU is idle when using our strategy (Step (4)), whereas FWDS needs to determine whether is idle and the task should wait for scheduling when the ECU is not idle



FAIRNESS-BASED DYNAMIC SCHEDULING

- *Example of the FDS MIMF Algorithm*
  - *论文里的各个调度图*

TABLE V: Task allocation steps of the motivating example using the FDS_MIMF algorithm.

Step	Figure	Current instant	System's criticality	Operation	Operated tasks and orders
1	Fig. 3	0	S_0	Allocation	$F_1.n_1, F_1.n_2, F_1.n_3, F_1.n_4, F_1.n_5, F_1.n_6$
2	Fig. 5	10	S_0	Cancel	$F_1.n_3, F_1.n_4, F_1.n_5, F_1.n_6$
3	Fig. 6	10	S_0	Allocation	$F_2.n_1, F_1.n_3, F_2.n_3, F_1.n_4, F_2.n_4, F_1.n_5, F_2.n_2, F_1.n_6, F_2.n_5$
4	Fig. 7	20	S_0	Cancel	$F_1.n_3, F_1.n_4, F_1.n_5, F_1.n_6, F_2.n_2, F_2.n_4, F_2.n_5$
5	Fig. 8	20	S_0	Allocation	$F_3.n_1, F_1.n_3, F_2.n_4, F_3.n_2, F_1.n_4, F_2.n_2, F_3.n_3, F_1.n_5, F_2.n_5, F_3.n_5, F_1.n_6, F_3.n_4, F_3.n_6$

问题：保证公平，但是没考虑错失率DMR

Outline

- Abstract
- Motivation
- Contribution
- Model
- Fairness-based Dynamic Scheduling
- Adaptive Dynamic Scheduling
- *Experiment*
- 思考 问题

ADAPTIVE DYNAMIC SCHEDULING

- *A. Deadline-slack：用来计算abs_deadline*
  - 可以理解为实际finish time的正常取值（不错失）的弹性空间
  - $Fm.deadlineslack = Fm.deadline - Fm.lowerbound.$
- *B. The ADS MIMF Algorithm*
  - *公平算法为了公平，系统的关键级设置为S0，最低级：但高关键级任务DMR被无视了*
  - 自适应性算法：目标是维持dmr，保证高关键级任务尽可能小的错失率；
- *C. Example of the ADS MIMF Algorithm*

ADAPTIVE DYNAMIC SCHEDULING

- *B. The ADS MIMF Algorithm*
  - *公平算法为了公平，系统的关键级设置为S0，最低级：但高关键级任务DMR被无视了*
  - 自适应性算法：保证高关键级任务尽可能小的错失率
  - 核心思想：
    - 当公平策略不能保证高关键级任务Fm 的错失率，即提升系统关键级到Fm同级
    - 对大于等于Fm的功能，进行公平调度
    - 当可以满足Fm的错失率时，回落关键级到S0，以公平调度所有任务

Experiment

- *比较overall makespan 和DMR*
- *不同规模的功能集*

总结

- 所提出的公平调度算法，——>对应dynamic 特性，dynamic是趋势
- 所提出的自适应（＋公平）调度算法，——>对应的是动态、安全、
  - 自适应的目标：保证安全
- 方法上收获：首先，提出的基本策略要能处理系统的动态变化，其次，基于此再根据目标去调整算法、方法已使得目标被动态地满足。

问题

- 动态任务调度（本文）与 运行时任务放置 有什么区别？
- 与前几次看TUM的论文相比
  - 1以软件组件为对象（component），解决软件组件在ECU上的runtime配置
  - 2以功能为对象，解决动态任务调度，以满足高性能

  - 1是在一个既定的集合S1里找解，不考虑性能（某个软件组件只能放在哪几个ECU之一上）
  - 2是在所有可能性的解空间集合S里找解，以满足某方面性能
- ？疑问：本文是否考虑了其实有的调度方案是不可行的（解不在出厂时给的范围（类比））？1是不是2的前一步行为？那对2有没有限制？
- ？支持消息动态发送是前提，这个消息动态发送啥意思？
- 对于本文的算法2，系统关键级变化遵循“低起点/快提升/快回落”,＝》“快提升慢回落”？
- （跟楠博士看的论文主要区别在哪）

?

- such similar automotive E/E architecture can also be found in some ACPS design [2], [12] and is basically similar to the hardware requirement (i.e., sensor and actuators are redundant or accessible via network) of 1oo2D (1 out of 2 Diagnosis) solution using dynamic reconfiguration by Elektrobit [53].
- The above requirement is basically similar to the software requirement (i.e., functions can be dynamically relocated) of 1oo2D salutation using dynamic reconfiguration by Elektrobit [53].

Thankyou!