实时计算系统设计

Introduction to Course

Prof. Jiyao AN

College of Computer Science and Electronic Engineering, HNU

网络+计算的时代变迁

Internet

IoT-Internet of Things

CPS-Cyber Physical System

Industry 4.0

Course Information

- Instructor
  - Jiyao AN（jt_anbob@hnu.edu.cn）
- TA
  - See TA ppt.
- Office
  - Tel: 18670005215
  - e-mail is the best way to communicate

TA Information

序号	助教姓名	联系电话	E-mail or QQ	助教班级
1				
2				
3				

助教职责：（课代表交电子作业或者e-mail联系时，请CC一份给我）

1. 跟班听课；
2. 批改作业（全部批改，并将学生作业中出现的普遍问题及时反馈主讲教师，必要时讲解习题。课程结束后将作业成绩交给主讲教师。）；
3. 带实验（全程辅导所助教班级的课程实验环节，协助教师考勤、验收实验并批改实验报告，课程结束后将所记录的情况反馈教师。）；
4. 答疑（2次集中答疑+6次个别答疑，请各班课代表与助教商定地点+时间）

Course Information

- *Introductory*

  (Advance) course in Real-time Computing System Design
- Who is this course for?
  - Undergrads, graduate students
- Prerequisites:
  - Discrete Mathematics,
  - Data Structures, Database,
  - Computing Method, Signal and System,
  - Operating Systems,
  - C++, Python,
  - Good programming skills.

Course Information

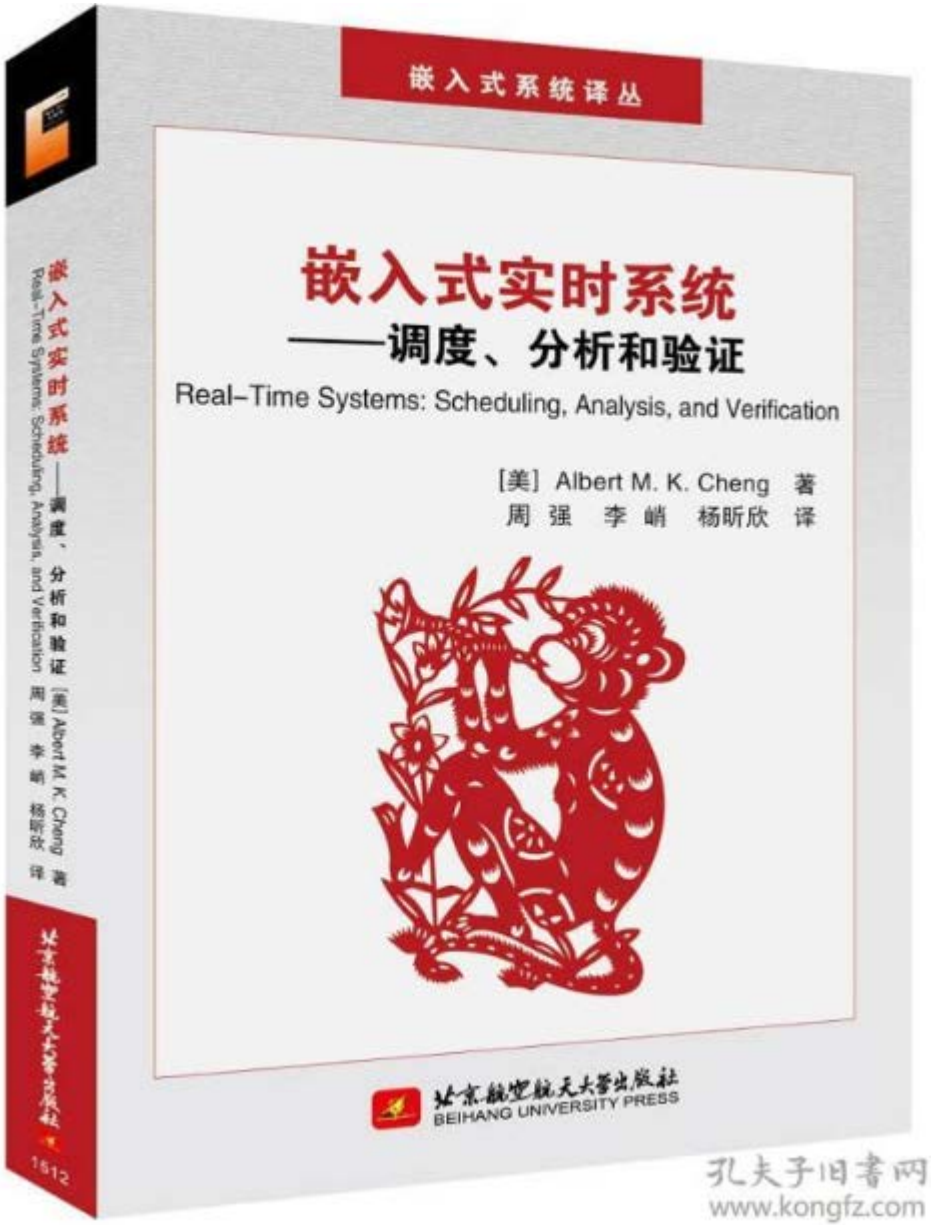
- Teaching resource:
  - Text: Real-Time Computing System Design
  - Slides
  - Assignment
- A little about me
  - Automotive CPS, IS (Intelligent System), IC (Intelligent Computing), Big data analysis, etc.

Reference

– 教材/参考书

- Alan Burns, Andy Wellings, Real-Time Systems and Programming Language, 2004.
- Hermann Kopetz. Real-Time Systems: Design Principles for Distributed Embedded Applications, 2011.
- Liu, Jane. Real-Time Systems. Prentice Hall, 2000.
- Lee, Edward Ashford, and Sanjit Arunkumar Seshia. Introduction to embedded systems: A cyber physical systems approach. Lee & Seshia, 2011.
- （美）阿尔伯特.陈著，周强，李峭，杨昕欣译, 嵌入式实时系统——调度、分析和验证, 北京航空航天大学出版社，**2015.12**

Reference



Grading Policy

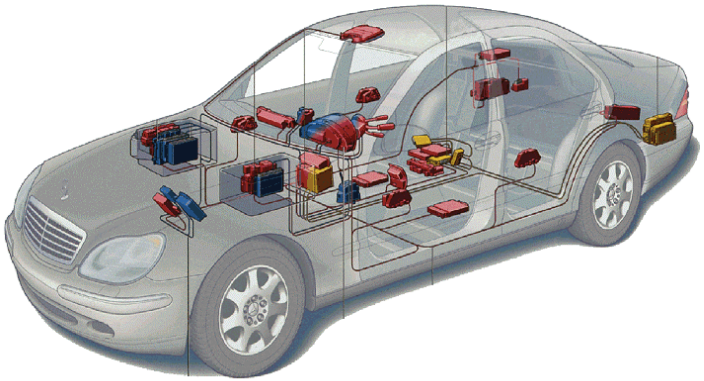
- Class attendance 10%
- Assignments & Experimentation 20%
- Mid exam 20%
- Final exam 50%

课程内容

课程文化**2W2H**：

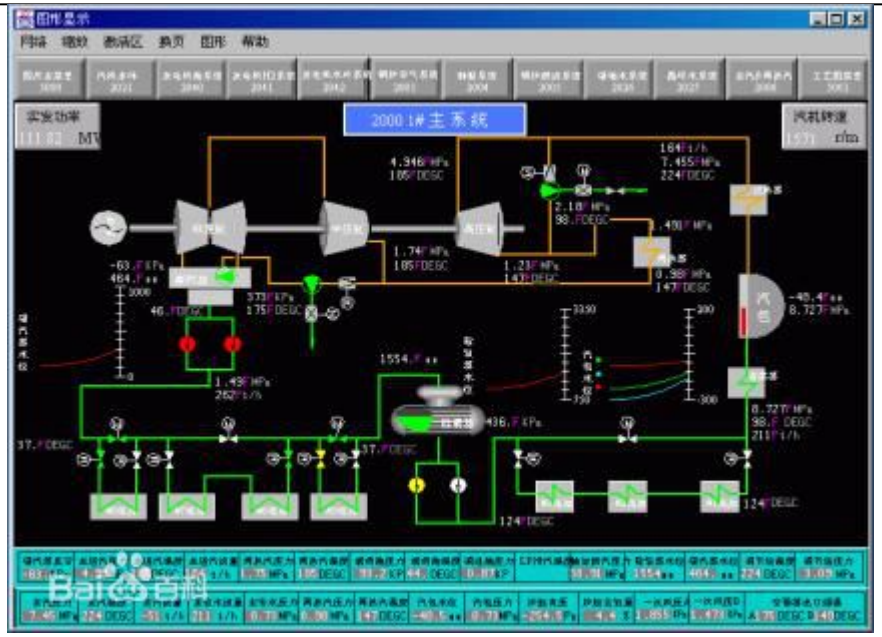
Why?　What?　How?　How Much?

Why?



Make it faster!

什么是实时系统



1. 一个实时系统是指计算的正确性不仅取决于程序的逻辑正确性，也取决于结果产生的时间，如果系统的时间约束条件得不到满足，将会发生系统出错。

2. **所谓“实时”，是表示“及时”，而实时系统是指系统能及时响应外部事件的请求，在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致的运行。**

3. 实时系统（Real-time system,RTS）的正确性不仅依赖系统计算的逻辑结果，还依赖于产生这个结果的时间。实时系统能够在指定或者确定的时间内完成系统功能和外部或内部、同步或异步时间做出响应的系统。因此实时系统应该在事先定义的时间范围内识别和处理离散事件的能力；系统能够处理和储存控制系统所需要的大量数据。

What?

- 实时计算系统概述
- 实时系统的硬件体系
- 实时计算系统基本概念
- 实时软件设计基础Elements of Real-Time Software Design
- 实时调度技术Real-Time Scheduling
- 资源分配Resource Allocation
- 多处理器与分布式系统中的实时调度
- 实时操作系统比较与分析Real-Time Operating Systems,RT-OS
- 实时系统的形式模型与形式验证

How?

- 课堂教学为主线，
- 学生实践、课程指导为主导，
- 作业、实验程序为辅线。

How Much?

实践内容

实践是检验真理的唯一标准。

美好的智能计算时代已经到来，

年轻的你还在等什么？

课堂大讨论——topic

实时计算系统RT-CS

普通计算系统GP-CS

几个Hot Spot

RTSS 2018

39th IEEE Real-Time Systems Symposium

11th -14th December 2018
Nashville, Tennessee, USA

http://2018.rtss.org/

- Hardware/Software Architecture
- Real-time Operating Systems and Scheduling
- Real-time Programming Languages and Software
- Reliability, Safety, and Fault Tolerance
- Performance Evaluation
- Real-time Sensing and Control
- Robotics and Integrated Manufacturing
- Case Studies

Photo credit: Courtesy of Nashville Convention & Visitors Corporation

RTSS是实时计算领域的国际顶级会议。

The scope of RTSS‘18

The IEEE Real-Time Systems Symposium (RTSS) is the premier conference in the field of real-time systems, presenting innovations with respect to both theory and practice. RTSS provides a forum for the presentation of high-quality, original research covering all aspects of **real-time systems, including theory, design, analysis, implementation, evaluation, and experience.** RTSS’18 continues the trend of making RTSS an expansive and inclusive symposium, looking to embrace new and emerging areas of real-time systems research.

RTSS’18 welcomes submissions in **all areas of real-time systems research, including but not limited to operating systems, networks, middleware, compilers, tools, modelling, scheduling, QoS support, resource management, testing and debugging, design and verification, hardware/software co-design, fault tolerance, security, power and thermal management, embedded platforms, and system experimentation and deployment experiences.**

In addition to the main real-time track (Track 1), submissions are also welcomed in the specialized areas of **Cyber-Physical Systems**, **HW-SW integration** and system level design, and **Internet of Things (IoT).** Together, these specialized areas comprise the CPS/HW-SW integration/IoT track (Track 2).

RTCS+AI

一个主论坛和11个分论坛（仿生机器人分论坛，机器人行业应用分论坛，CV分论坛，智能安全分论坛，金融科技分论坛；自动驾驶分论坛，NLP分论坛，AI+分论坛，AI芯片分论坛，IoT分论坛，投资人分论坛）


专场设置
6.29
人工智能
主会场
6.30
仿生机器人
专场
机器人行业应用
专场
CV
专场
智能安全
专场
金融科技
专场
7.1
智能驾驶
专场
NLP
专场
AI+
专场
AI芯片
专场
IoT
专场
投资人
专场

会议由一个主论坛、若干主题论坛、创新竞赛、以及一系列体验和展示组成。

主论坛	包括开幕式和全体会议两部分。开幕式包括政要致辞、学界代表发言、企业代表发言。全体会议主要对重点议题进行全面交流。
主题论坛	主要从技术、产业、应用等角度对人工智能技术和产业的发展现状进行深入探讨、包括脑机融合、人工智能算法、框架平台、智能芯片、大数据、智能图像、智能语音、智能驾驶、投融资等主题论坛，以及部分国际国内顶级企业举办的论坛。
创新竞赛	聚焦重点应用场景，评选出若干行业标杆性的人工智能产品或方案，进行宣传推广，营造全社会共同参与的创新创业氛围。
体验和展示	根据主会场周边的空间特点，在主会场及周边AI PARK 区域集中呈现最佳人工智能体验和展示方案。

一类典型的实时计算系统，

Automotive Cyber-Physical System

Cyber-physical systems

From Wikipedia

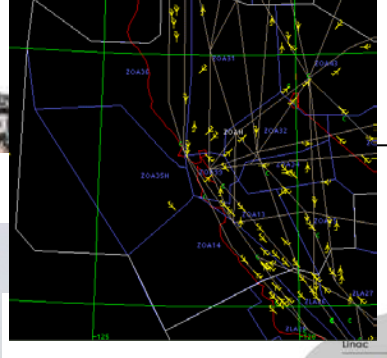
- “Cyber-Physical Systems (CPS) are integrations of computation, networking, and physical processes.”

- “Embedded computers and networks monitor and control the physical processes, with feedback loops where physical processes affect computations and vice versa.”

CPS Application Domains

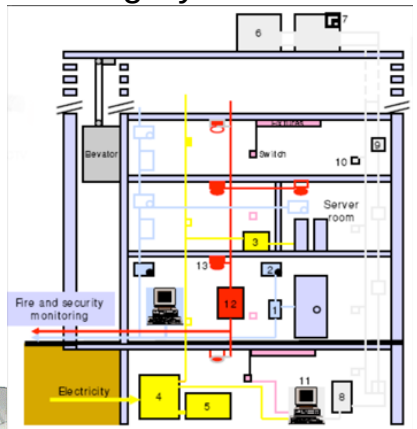
(source Lee & Seshia – UC Berkeley)

Avionics



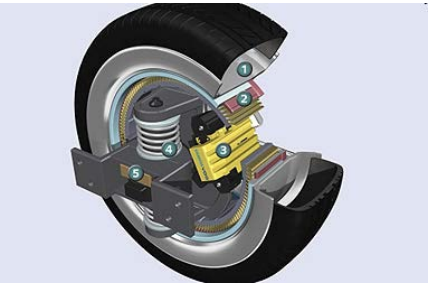
Transportation (Air traffic control at SFO)

Building Systems

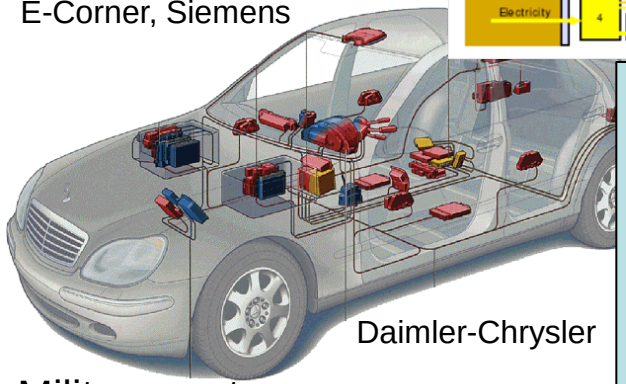


Telecommunications

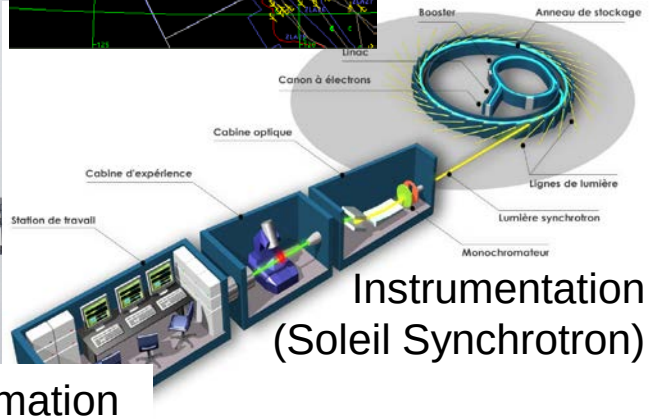
Automotive



E-Corner, Siemens



Daimler-Chrysler



Instrumentation (Soleil Synchrotron)

Factory automation

Courtesy of Kuka Robotics Corp.

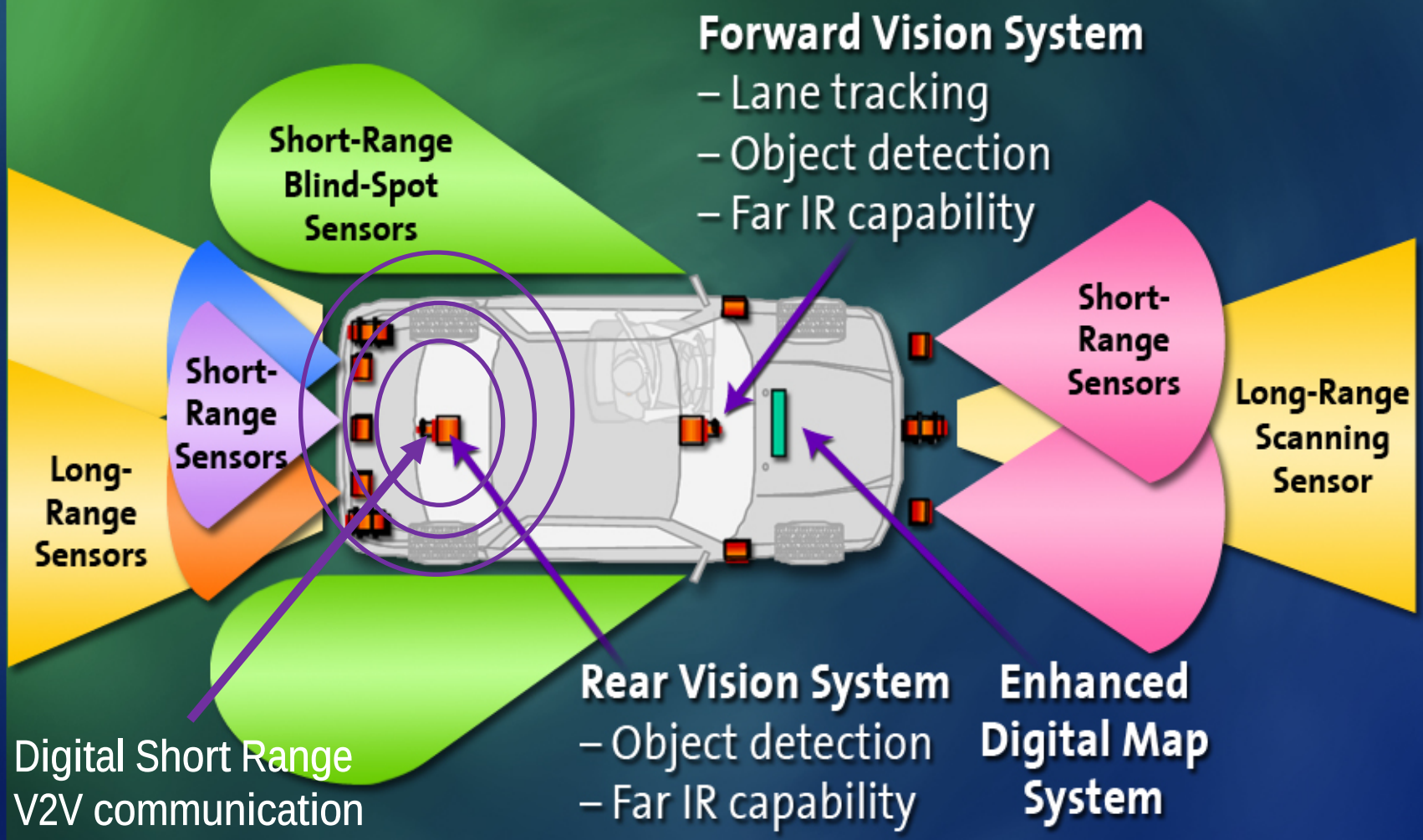
Power generation and distribution

Courtesy of General Electric

Military systems:

Courtesy of Doug Schmidt

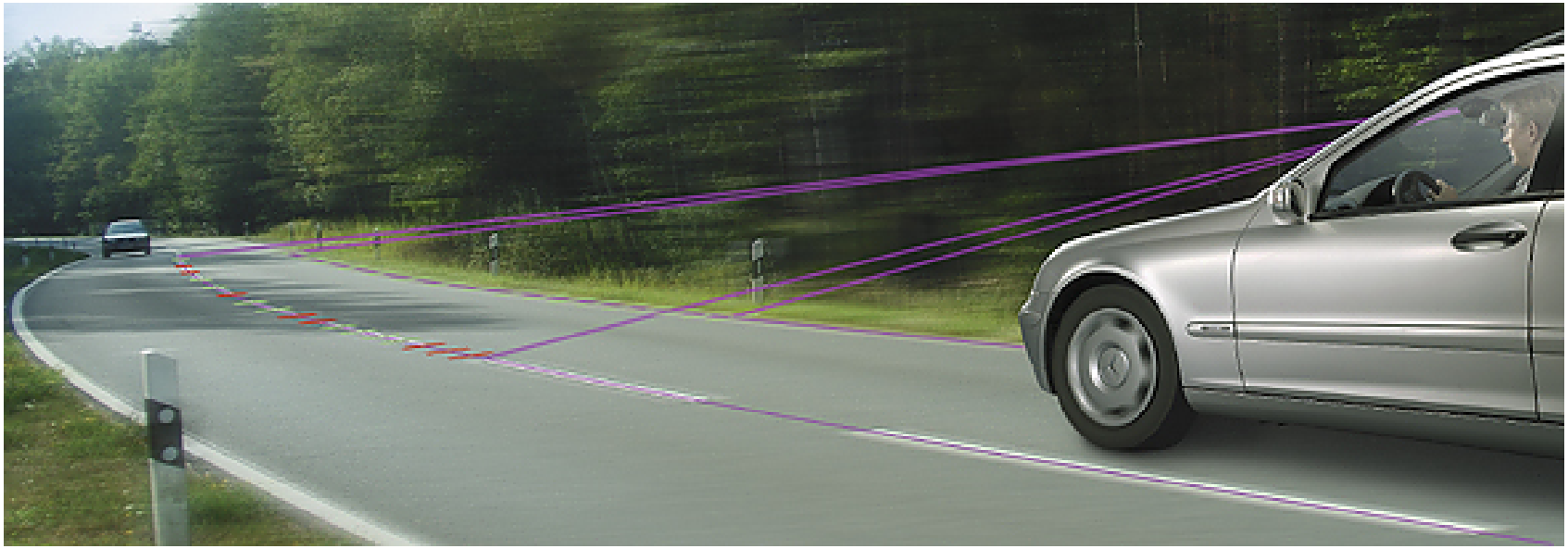
The refuse-to-collide car



Source: Larry Burns, General Motors

An Example Automotive Feature

Lane Keeping System (LKS)



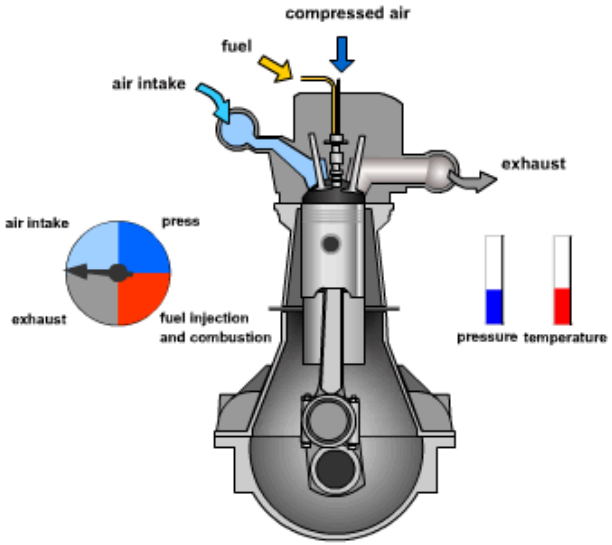
"… alerts the driver with acoustical or haptic warnings before his vehicle is about to leave the lane
….could **prevent just about half of the accidents** caused in this way.

… additionally responds through a gentle intervention in the steering, which the driver can counteract at any time. This can save additional time to react appropriately – **each and every second counts**."

Source: Continental Website

Example: Engine Control Unit

- Read sensors (temp, pedal position, exhaust) and control fuel injector timing and spark timing
- Control engine fan and other actuators
- Handle the serial data communication that is common in cars.
- Interface with climate and other passenger controls
- Provide diagnostics



http://automobile-us.blogspot.com

Source: Partha Dasgupta

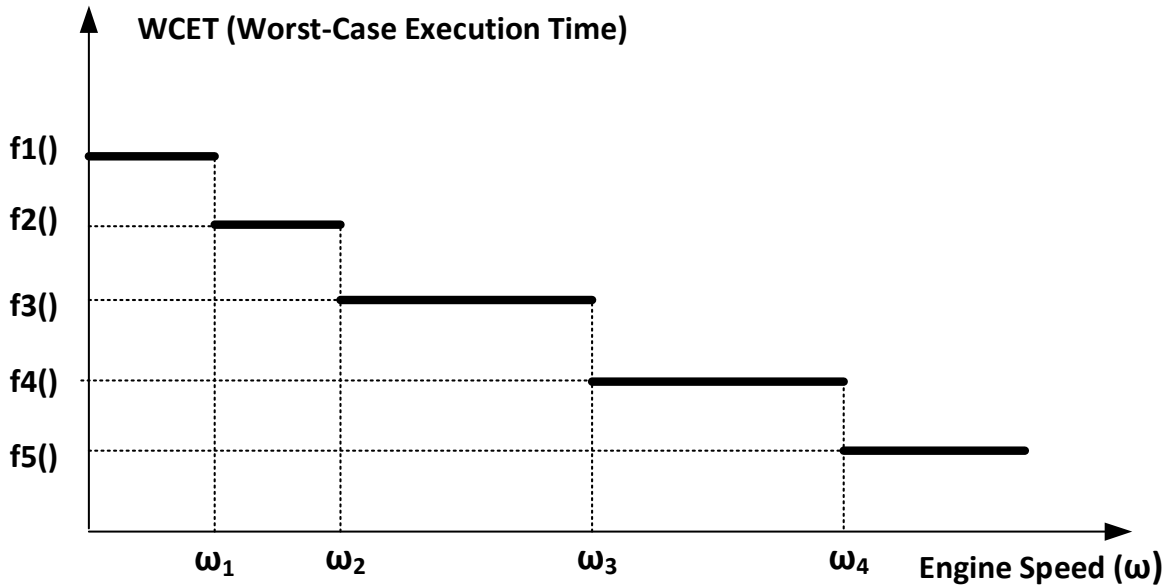
Software Implementation of Engine Control

```
#define ω1 1000
#define ω2 2000
#define ω3 4000
#define ω4 6000

task EC_task {
      ω = read_engine_speed();

      if (ω ≤ ω1)          f1();
      else if (ω ≤ ω2)     f2();
      else if (ω ≤ ω3)     f3();
      else if (ω ≤ ω4)     f4();
      else                 f5();
}
```

(a)



(b)

- "Adaptive" Variable Rate
  - Triggered at certain crankshaft rotation angle
  - *Better, more sophisticated* control strategy at *lower* engine speed interval

Real-time

Make it faster!

What if you need "absolutely positively on time"?

Today, most embedded software engineers write code, build system, and test for timing. Model-based design seeks to specify dynamic behavior (including timing) and "compile" implementations that meet the behavior.

(source Lee & Seshia – UC Berkeley)

A Key Challenge on the Cyber Side: Real-Time Software

- *Correct execution of a program in C, C#, Java, Haskell, etc. has nothing to do with how long it takes to do anything. All our computation and networking abstractions are built on this premise.*

Timing of programs is not repeatable, except at very coarse granularity.

Programmers have to step outside the programming abstractions to specify timing behavior.

(source Lee & Seshia – UC Berkeley)

Techniques Exploiting the Fact that Time is Irrelevant

- Programming languages
- Virtual memory
- Caches
- Dynamic dispatch
- Speculative execution
- Power management (voltage scaling)
- Memory management (garbage collection)
- Just-in-time (JIT) compilation
- Multitasking (threads and processes)
- Component technologies (OO design)
- Networking (TCP)
- …

(source Lee & Seshia – UC Berkeley)

Sequential Processing

- They have one “thread” of execution
- One step follows another in sequence
- One processor is all that is needed to run the algorithm

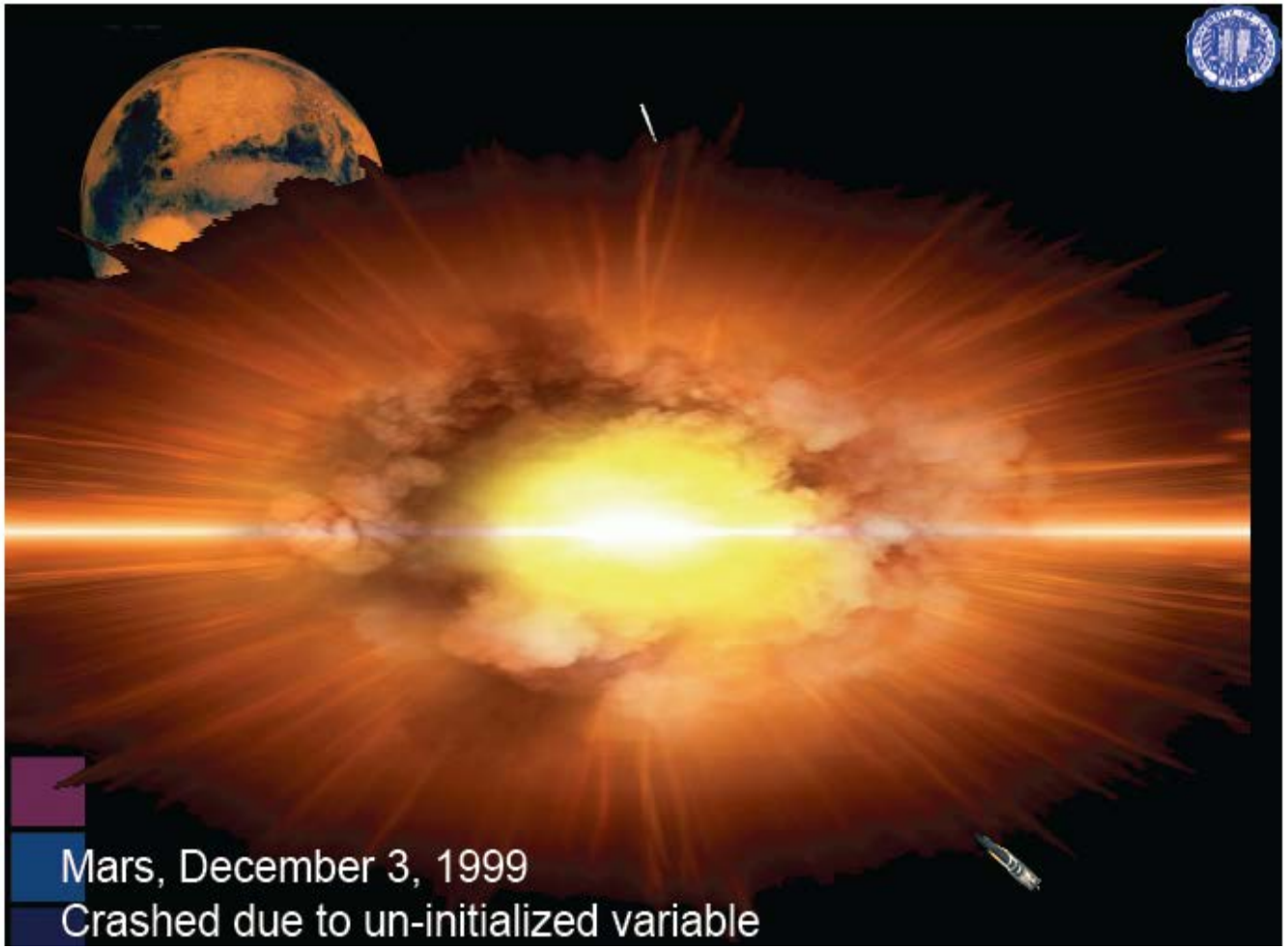
An example of non-sequentiality

- Your car has an onboard digital dashboard that "simultaneously":
  - Displays the speed on the speedometer
  - Checks your engine oil level
  - Checks your fuel level and calculates fuel consumption
  - Monitors the temperature of the engine and turns on a light if it is too hot
  - Monitors your alternator to make sure it is charging your battery
  - ….

Example: Ariane 5

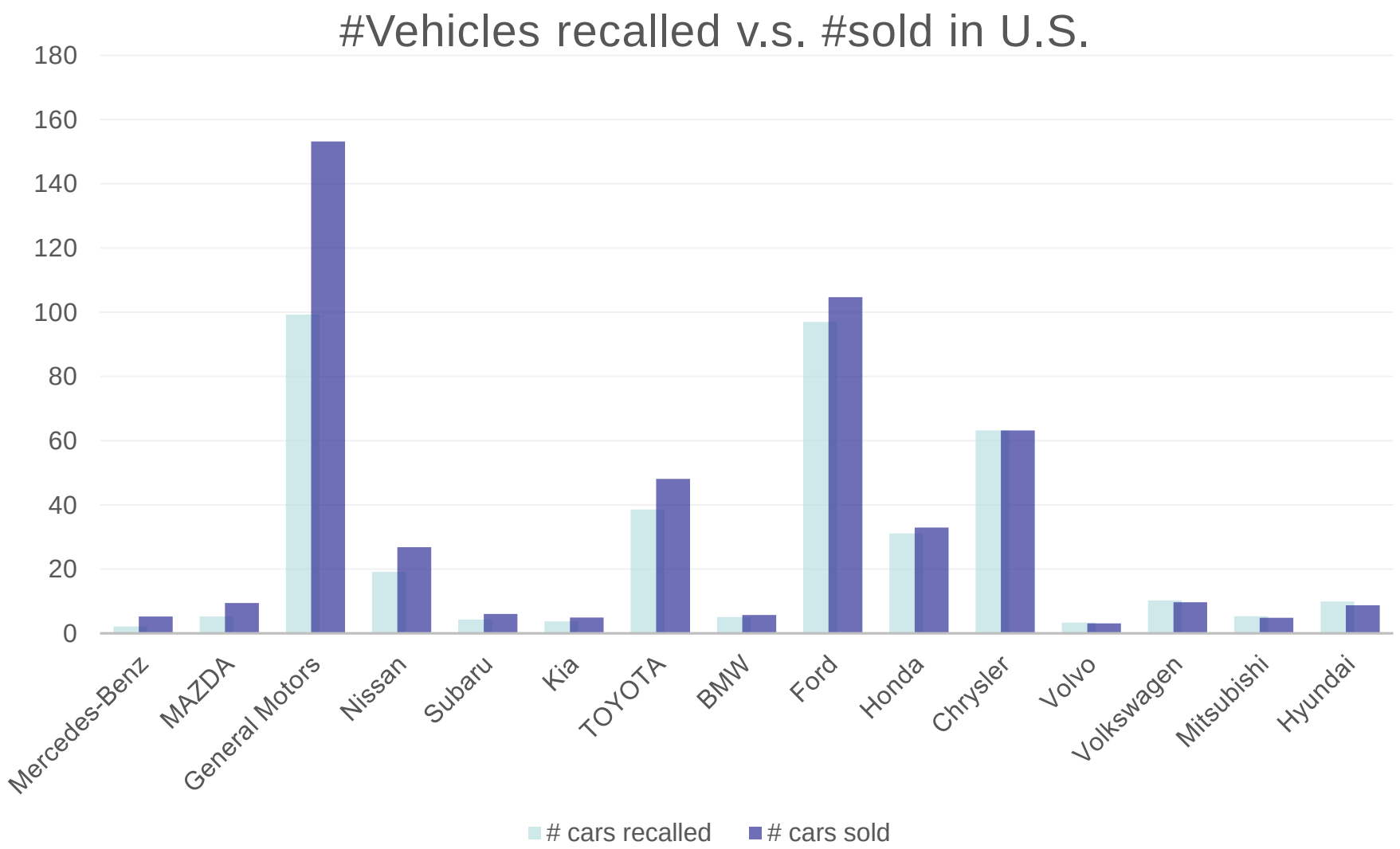
- Ariane 5 is a European expendable launch system designed to deliver payloads into geostationary transfer orbit or low Earth orbit. It succeeded Ariane 4, but does not derive from it directly. **Its development took 10 years and cost $7 billion.**
- Ariane 5's first test flight (Ariane 5 Flight 501) on 4 June 1996 failed, with the rocket self-destructing 37 seconds after launch because of a malfunction in the control software, which was arguably one of the most expensive computer bugs in history. A data conversion from 64-bit floating point to 16-bit signed integer value had caused a processor trap (operand error). The floating point number had a value too large to be represented by a 16-bit signed integer.
- ***The Ariane 5 software reused the specifications from the Ariane 4, but the Ariane 5's flight path was considerably different and beyond the range for which the reused code had been designed.*** Specifically, the Ariane 5's greater acceleration caused the back-up and primary inertial guidance computers to crash, after which the launcher's nozzles were directed by spurious data. Pre-flight tests had never been performed on the re-alignment code under simulated Ariane 5 flight conditions, so the error was not discovered before launch.

Example: Mars



The Failure Review Board concluded that the most likely cause of the mishap was a software error that incorrectly identified vibrations, caused by the deployment of the stowed legs, as surface touchdown

Automotive Safety Recalls



Data Source: iseecars.com

New Challenges in Networked Car: Connectivity

- Remotely hacked Chrysler Jeep Cherokee (2015.07)
  - Using Wi-Fi in infotainment
  - Relaying commands to dashboard functions, steering, brakes, transmission, etc.
  - 1.4 million cars recalled
  - https://www.youtube.com/watch?v=MK0SrxBC1xs

Source: wired.com

Homework

1. 针对大数据的实时批量计算，了解与分析一种实时可靠的开源分布式实时计算系统——Storm。
2. 了解：VxWorks、μClinux、μC/OS-II和eCos等4种性能优良并广泛应用的实时操作系统RTOS。

- Questions/Issues?