



东风汽车集团有限公司技术中心
DONGFENG MOTOR CORPORATION TECHNICAL CENTER

MATLAB在混动动力总成控制策略软件开发中的应用

张德全, 东风汽车集团有限公司技术中心



MATLAB EXPO

主要内容

- 动力总成简介及实现路径
- 发动机管理控制器软件架构设计
- 控制策略软件搭建工具链
- 总结及建议

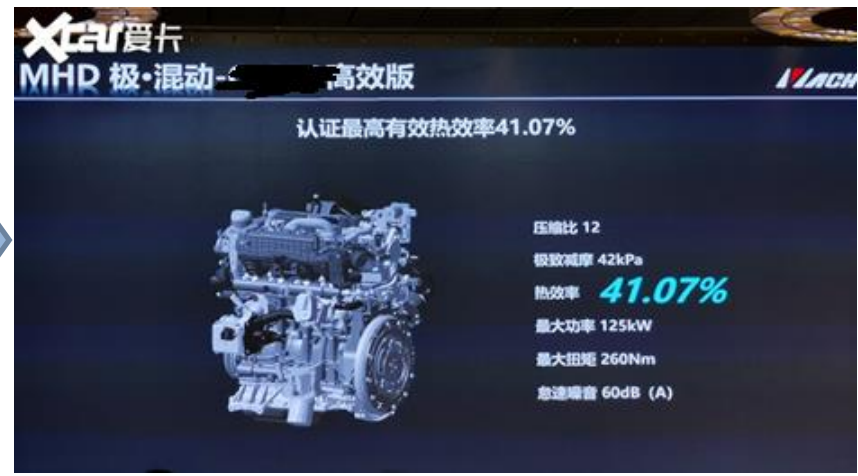
动力总成简介及策略实现路径



●2018年

●2020年

●2023年



产品认证证书
PRODUCT CERTIFICATION CERTIFICATE
能效之星
最高有效热效率 45.18 %

新技术/新策略导入

- 高效抗爆震快速燃烧控制系统
- 混动专用VGT（可变截面涡轮增压）控制技术
- 低压EGR(废气再循环)全域耦合控制技术
- 350bar缸内直喷系统
- 智能热管理系统

AUTOSAR标准正向开发

- AUTOSAR标准平台搭建-V4.0
 - 应用软件层 ★
 - 运行时环境
 - 基础软件层
 - 服务层
 - ECU抽象层
 - 微控制器抽象层
 - 复杂驱动层

ISO26262功能安全支撑

- 可验证性
- 可追溯性
- 可配置性
- 灵活性
- 可测性
- 可维护性

CMMI-3过程管控体系

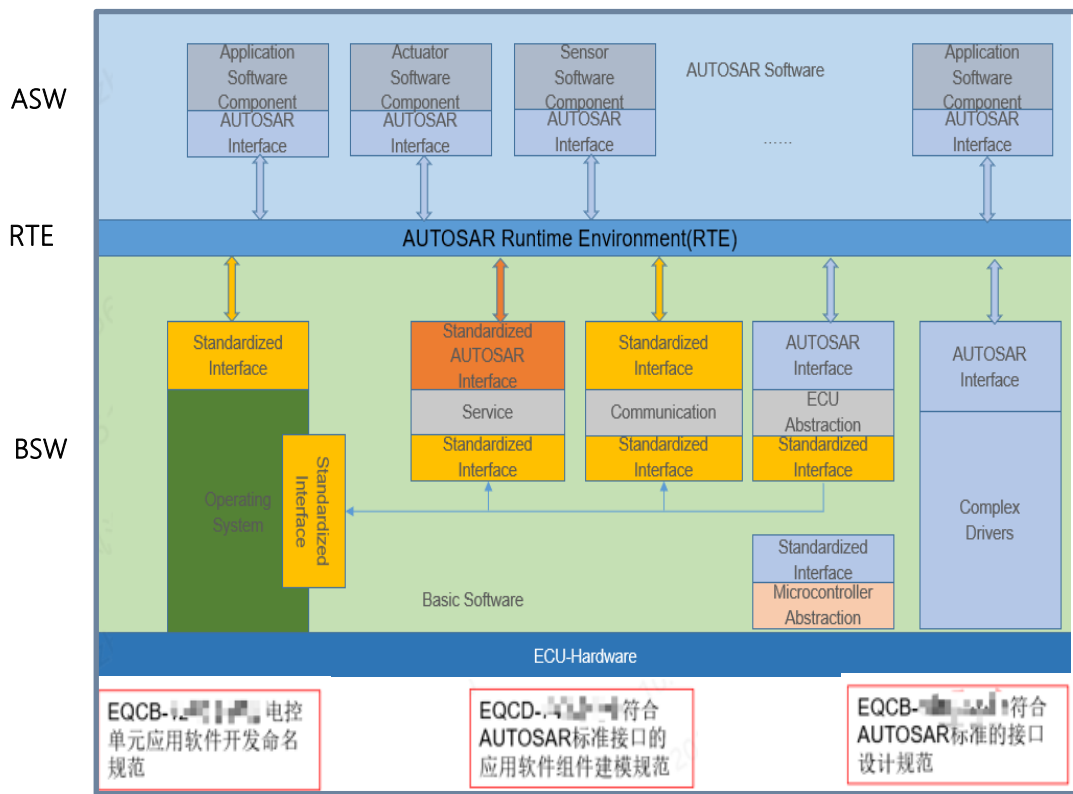
- 需求管理-Systemweaver
- 软件架构设计-Systemweaver
- 模型开发-MATLAB ★
- 单元测试-MATLAB
- MIL测试-MATLAB
- HIL测试

发动机管理控制器软件架构设计

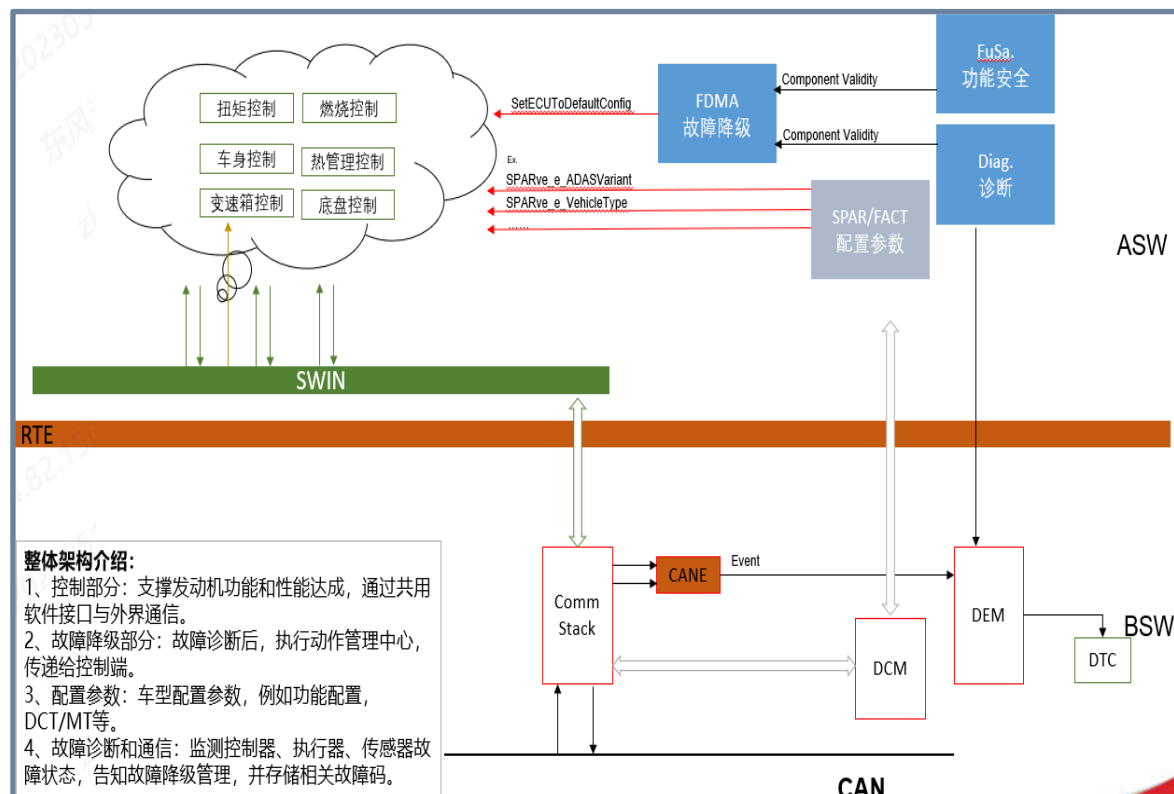
软件架构--AUTOSAR

软件架构按照AUTOSAR标准构建，包含应用层、运行时环境（RTE，Run Time Environment）、基础层。

- 电子控制器属于嵌入式系统，其特点之一是嵌入式软件强依赖于与硬件，执行效率高但通用化较差；
- Autosar核心理念是通过定义标准化的接口层使得应用软件可以脱离硬件的束缚，实现高复用率和功能配置化管理；
- 形成通用化软件接口搭建标准，全集团规范使用。



✓标准化AUTOSAR架构定义

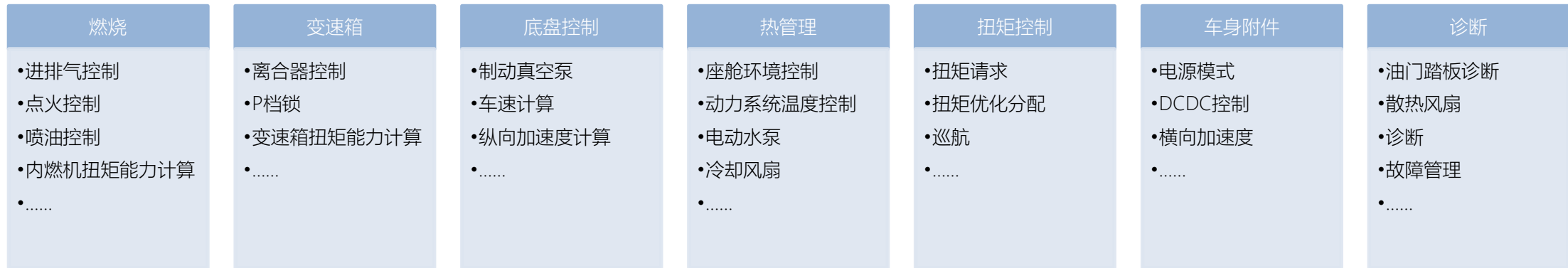


✓符合AUTOSAR的软件架构目标

发动机管理控制器软件架构设计

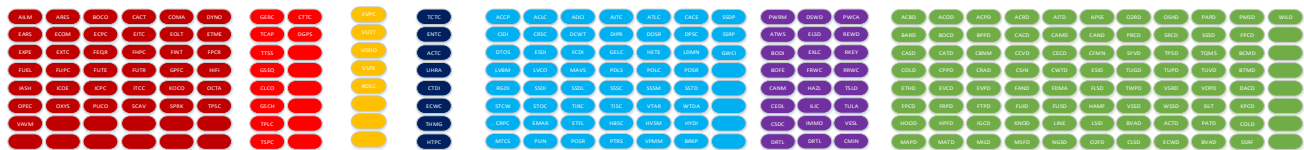
软件架构—功能域

主要包含了燃烧控制、变速箱控制、底盘控制、热管理、扭矩控制、车身附件、诊断7个功能域



该动力控制策略应用软件包含约250基本控制模块单元(SWU)

- 燃烧域包含38个SWU，主要实现发动机呼吸控制，点火控制、喷油控制、爆震控制、排放控制及相关执行器控制等功能
- 扭矩流包括42个SWU，实现驾驶意图识别、驾驶扭矩计算及融合、车辆扭矩仲裁、扭矩驾驶性修正、发动机启停、动力系统能力预测、能量管理、执行器扭矩动态协调、执行器控制等功能
- 车辆状态实现系统参数配置、休眠唤醒、显示信息处理等
- 附件控制实现真空泵、水泵、冷却风扇等的控制
- 诊断实现对传感器、执行器的故障诊断及故障后的功能降级处理



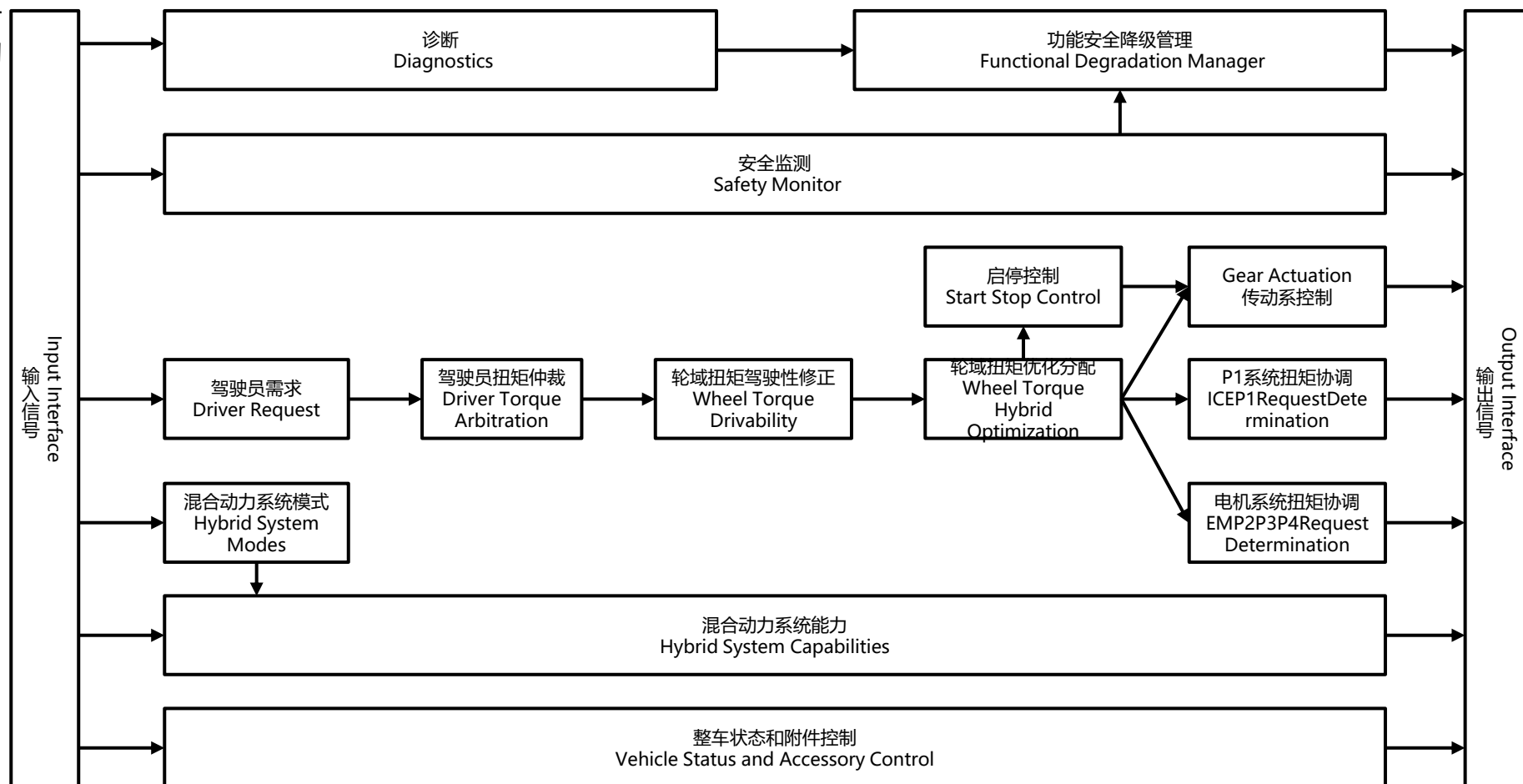
发动机管理控制器软件架构设计

软件架构—扭矩系统框图

应用层软件以扭矩流为中心，实现对驾驶员意图识别、动力总成和车辆的控制、诊断和安全管理；

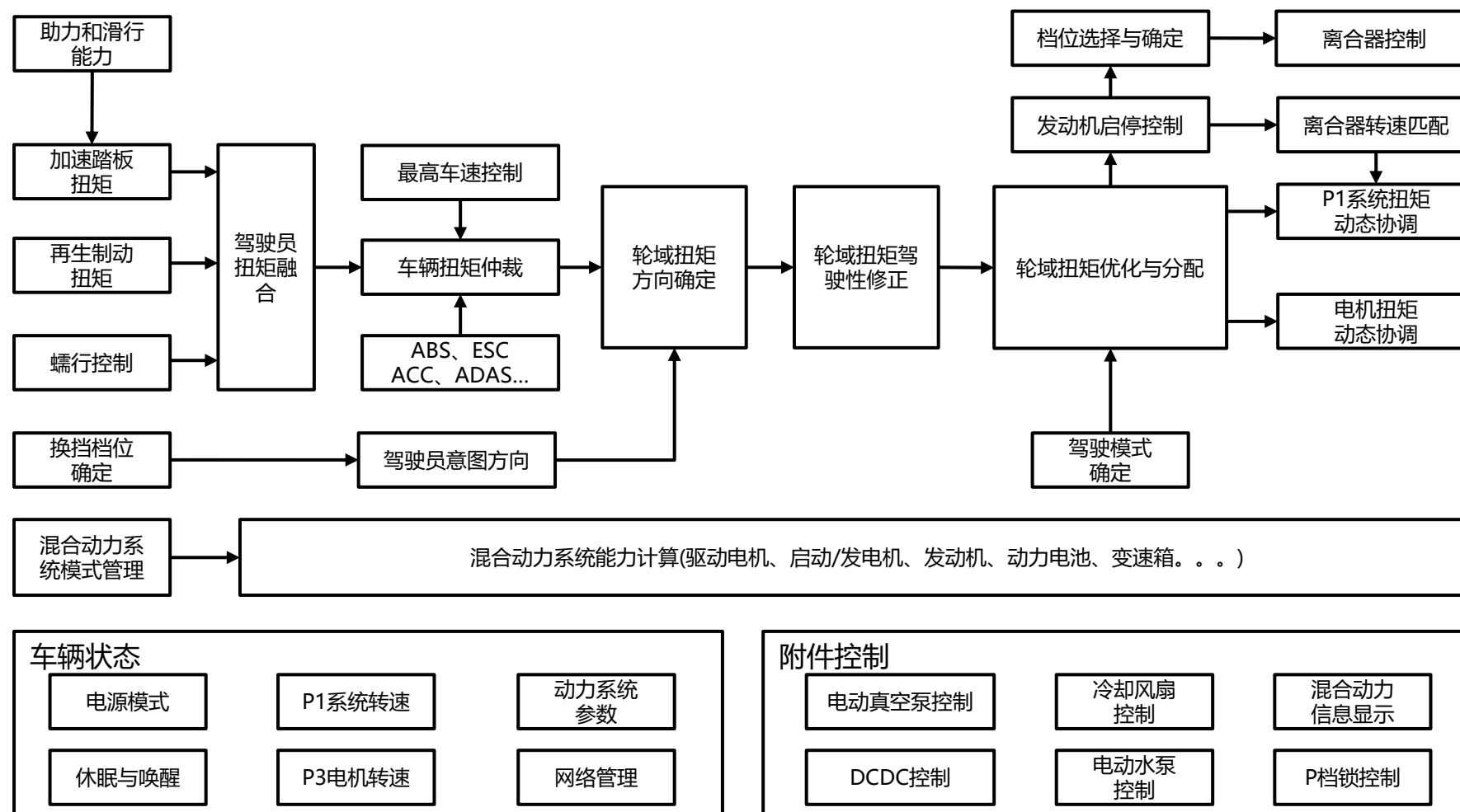
主要包括：

驾驶员需求、
混合动力系统模式、
混合动力系统能力、
驾驶员扭矩仲裁、
扭矩驾驶性修正、
扭矩管理及分配、
启停控制、
传动系控制、
P1系统扭矩协调、
电机系统扭矩协调、
整车状态和附件控制、
诊断、
安全监测、
功能安全降级管理等14大块。



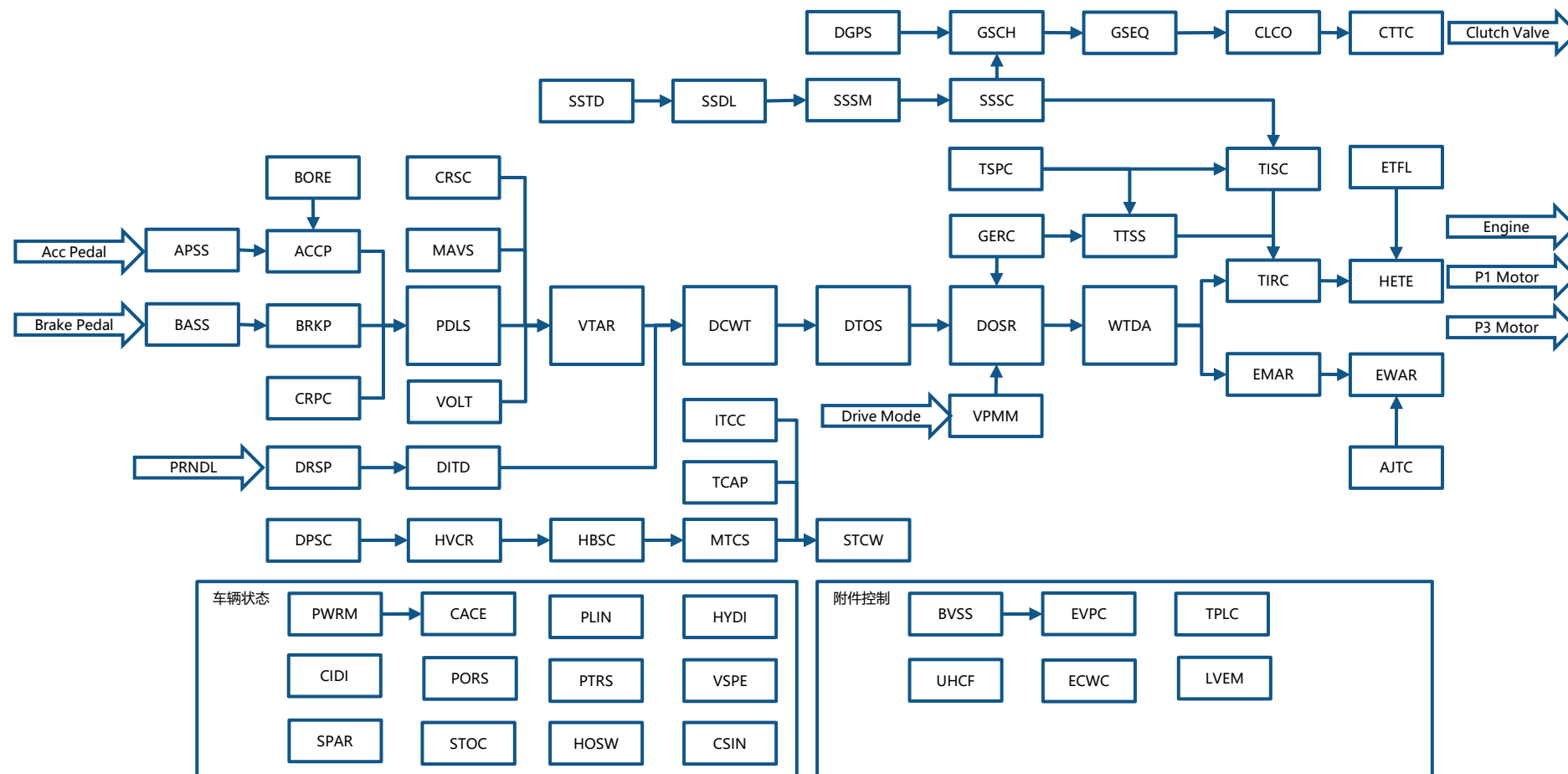
发动机管理控制器软件架构设计

软件架构—扭矩系统框图-功能级



发动机管理控制器软件架构设计

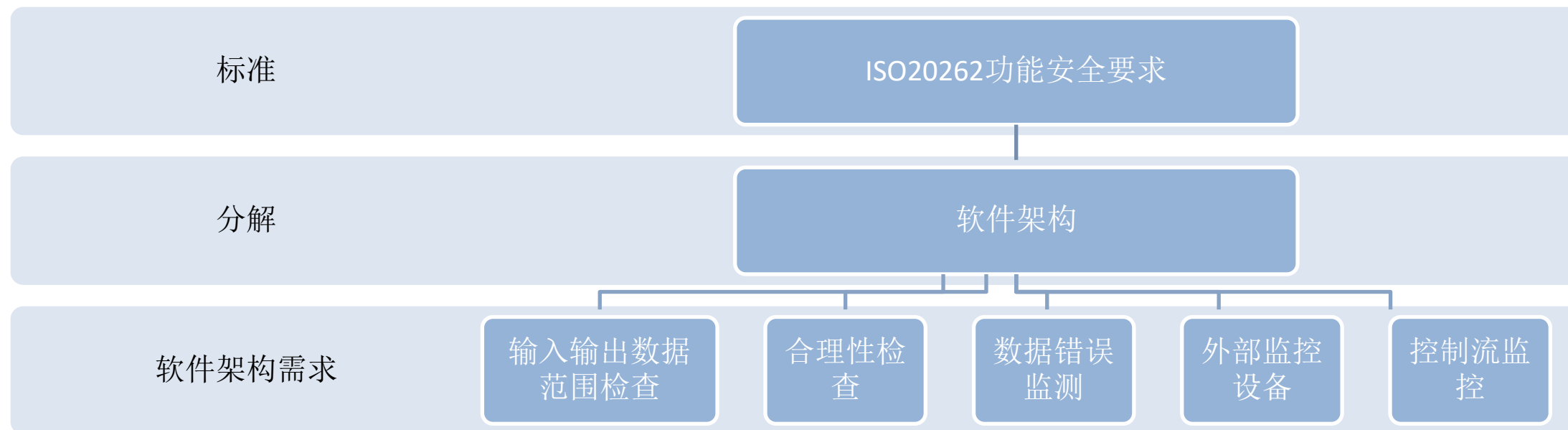
软件架构—扭矩系统框图设计级



发动机管理控制器软件架构设计

软件架构—功能安全

遵循ISO26262标准，最高功能安全级别达到ASIL-C标准；同时满足EGAS欧盟标准。

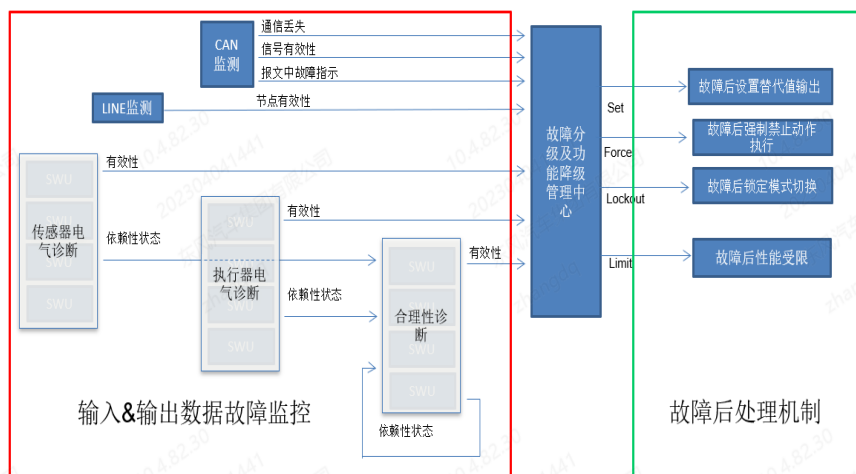
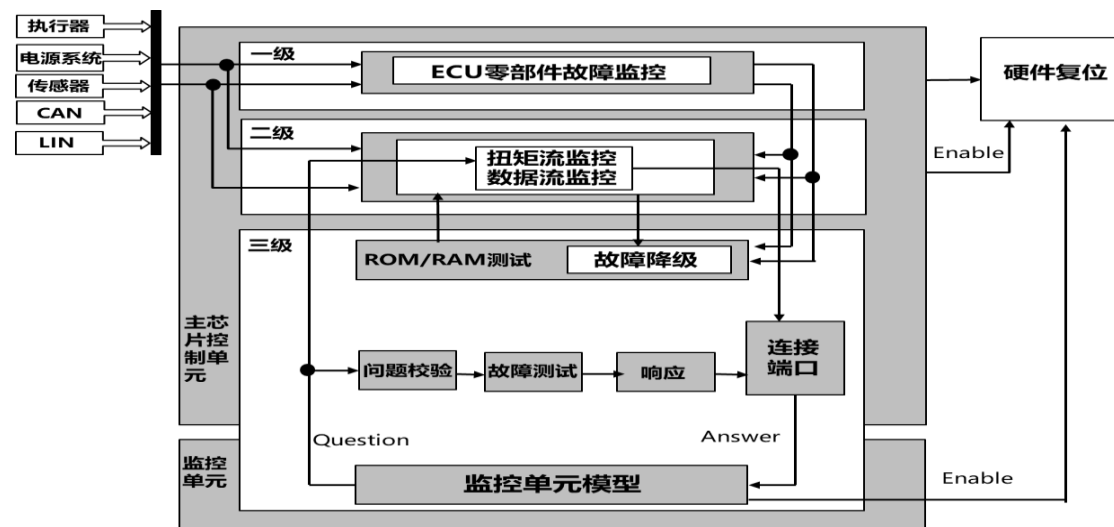


发动机管理控制器软件架构设计

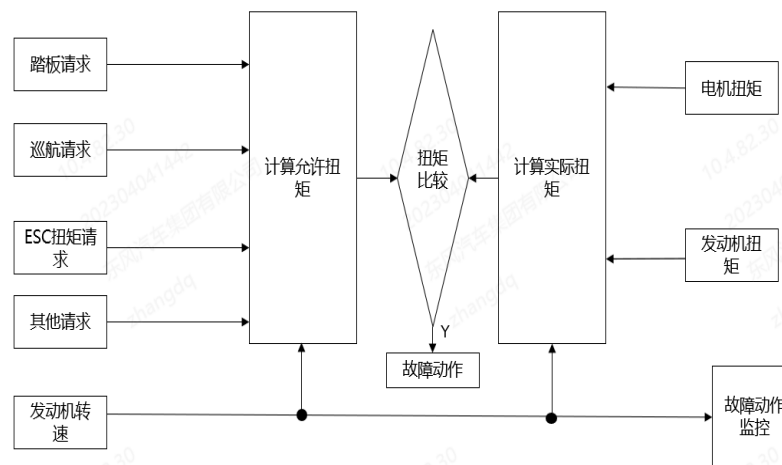
软件架构—功能安全

遵循ISO26262标准，最高功能安全级别达到ASIL-C标准；同时满足EGAS欧盟标准。

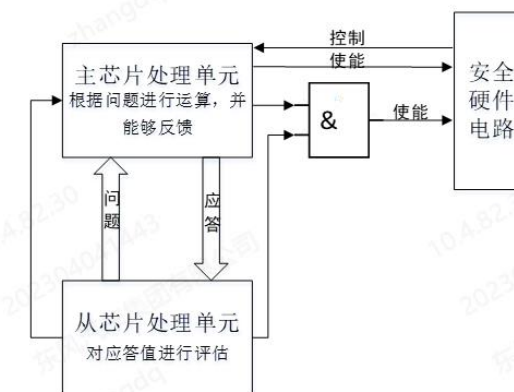
- ❑ 一级主要针对零部件故障监控
- ❑ 二级主要针对控制路径监控，例如扭矩流和程序数据流
- ❑ 三级主要针对内部处理器监控，例如ROM/RAM监控



✓ 零部件故障监控



✓ 控制路径监控



✓ 处理器安全监控

发动机管理控制器软件架构设计

软件架构到设计实现案例介绍-处理器安全监控

D.2.3.5 Reciprocal comparison by software in separate processing units 需求输入

NOTE This technique/measure is referenced in Table D.4.

Aim: To detect, as early as possible, failures in the processing unit, by dynamic software comparison.

Description: Two processing units exchange data (including results, intermediate results and test data) reciprocally. A comparison of the data is carried out using software in each unit and detected differences lead to a failure message (see Figure D.3). This approach allows for hardware and software diversity if different processor types are used. The design includes methods to avoid false failures (e.g. loop jitter, communication delays, etc.).

Paths can be implemented using separate hardware or software. The design includes analysis to understand communication between the two cores.

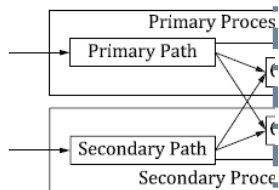


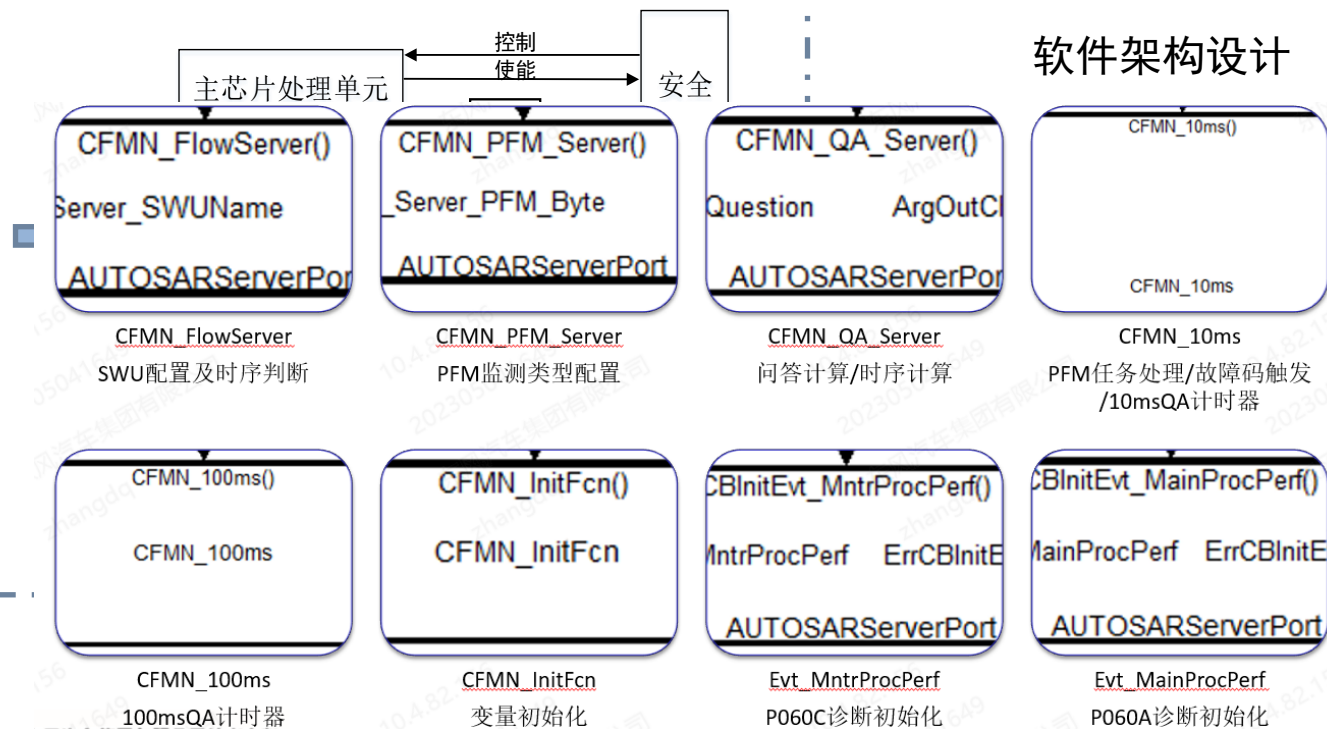
Figure D.3 — Redundant software architecture

主从芯片处理单元数据流运算逻辑监控算法，保证各芯片处理单元按照正常的顺序进行执行，整个控制系统包含主芯片处理单元，从芯片处理单元，安全硬件电路，及主从芯片处理单元串行传输通道，在此硬件配置的基础上进行故障监测。

目标:

- 算术逻辑单元 (ALU, Arithmetic Logic Unit) 运行无故障
- 程序数据流 (PFM, Program Flow Monitoring) 运行无故障

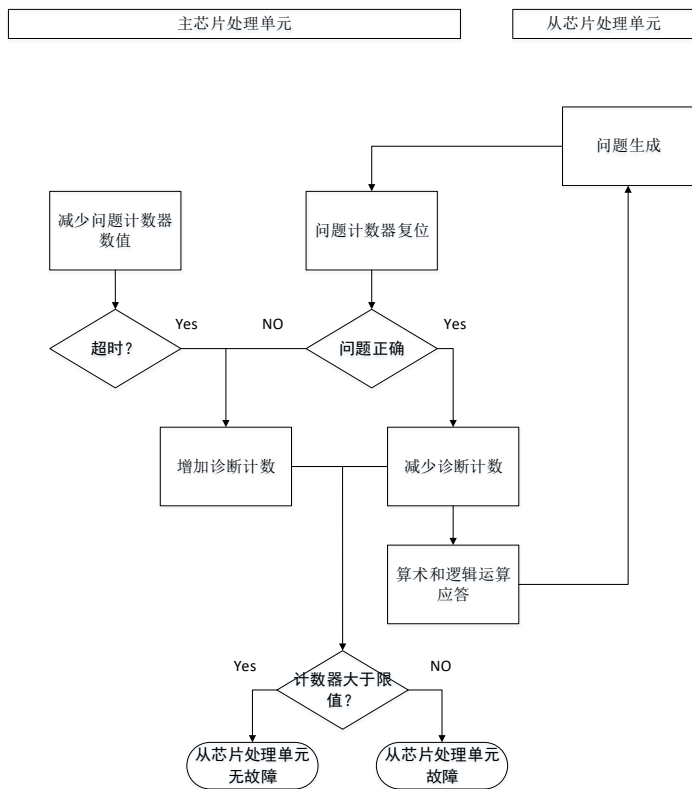
方案设计



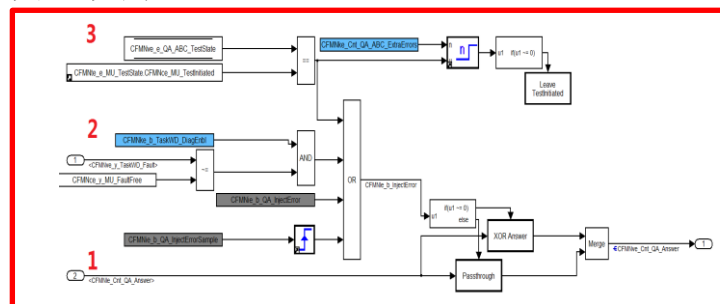
软件架构设计

发动机管理控制器软件架构设计

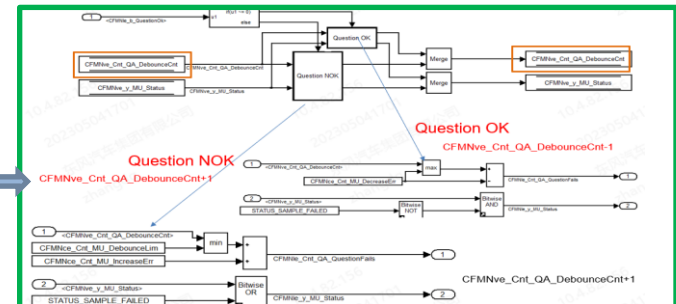
软件架构到设计实现案例介绍-处理器安全监控



设计实现



Step2: 从芯片发送Question



Step3: 主芯片发送Answer



```

package MUTR_pkg.MUTR_sw

import MUTR_pkg.MUTR_if.*
import TENG_pkg.TENG_dt.*
import ArcCore.Services.IoHwAb.*

// component declarations
component complexDeviceDriver MUTR {
  ports {
    server MUTR_ErrorInformation provides ErrorInformation
    server MUTR_PFM_Answer provides PFM_Answer
    server MUTR_NDIS_Status provides NDIS_Status
    server MUTR_QA_AntiBounceCounter provides QA_AntiBounceCounter
    client CFMN_PFM_Server requires PFM_Server
    client CFMN_QA_Server requires QA_Server
    sender MUTRve_U_ADTestVoltageDiff provides MUTRve_U_ADTestVoltageDiff
    client MUTR_Adc5V_Read requires AnalogServiceRead
  }
}
    
```

Step1: 底层触发Question

```

static boolean dtclsGroup(uint32 dtc, Dem_DTCFormatType dtcFormat, uint32 *groupLower, uint32 *groupUpper)
{
  const Dem_GroupOfDtcType * DTCGroups = configSet->GroupOfDtc;
  boolean groupFound = FALSE;
  if( DEM_DTC_FORMAT_UDS == dtcFormat ) {
    while( (FALSE == DTCGroups->Arc_EOI) && (FALSE == groupFound) ) {
      if( dtc == DTCGroups->DemGroupDTCs ) {
        *groupLower = DTCGroups->DemGroupDTCs;
        groupFound = TRUE;
      }
      DTCGroups++;
    }
    *groupUpper = DTCGroups->DemGroupDTCs - 1u;
  }
  return groupFound;
}
    
```

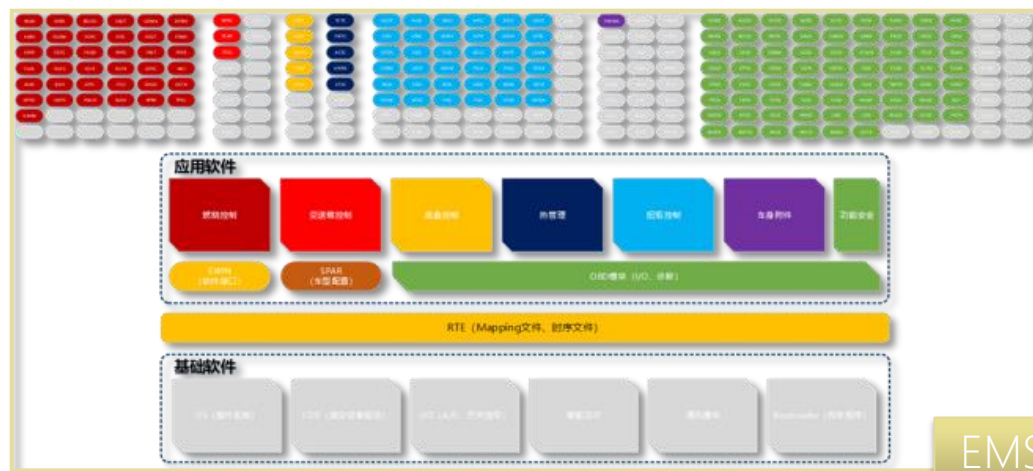
Step4: 故障触发及故障保护

ASW

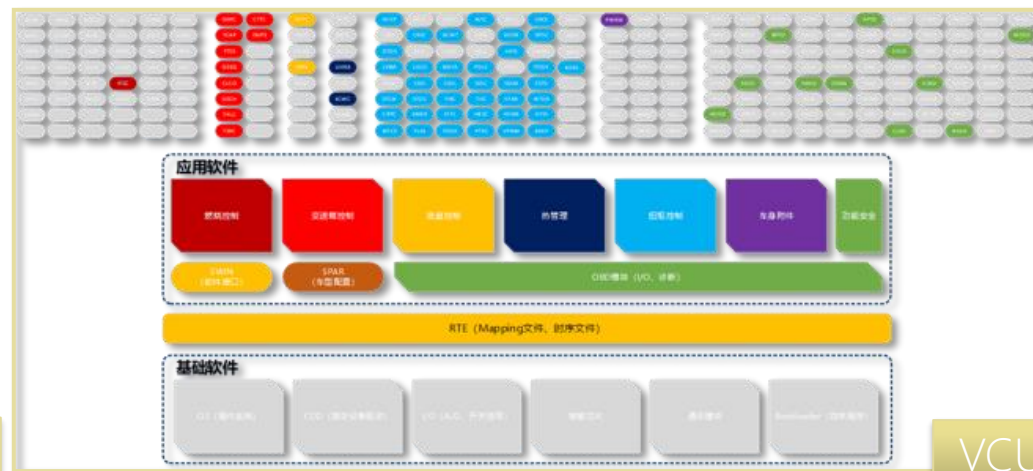
BSW

发动机管理控制器软件架构设计

软件架构—形成的通用的MATLAB软件模块库，可配置其他控制器开发



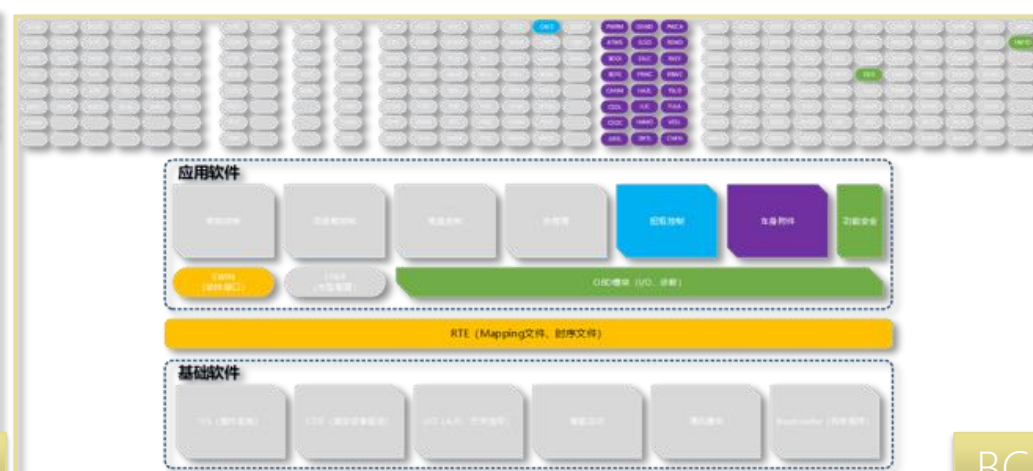
EMS



VCU



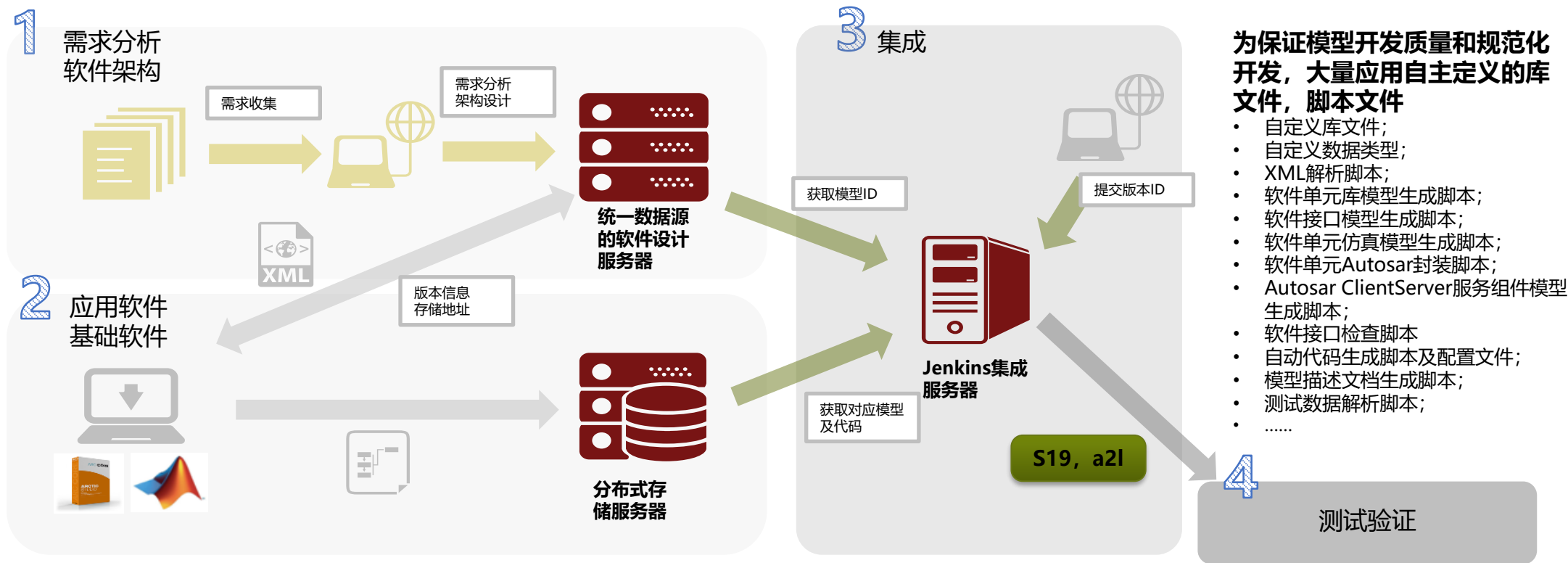
GW



BCM

控制策略软件搭建工具链

- MATLAB在软件开发环节中的应用
 - 软件开发工具链应用体系



为保证模型开发质量和规范化开发，大量应用自定义的库文件，脚本文件

- 自定义库文件;
- 自定义数据类型;
- XML解析脚本;
- 软件单元库模型生成脚本;
- 软件接口模型生成脚本;
- 软件单元仿真模型生成脚本;
- 软件单元Autosar封装脚本;
- Autosar ClientServer服务组件模型生成脚本;
- 软件接口检查脚本
- 自动代码生成脚本及配置文件;
- 模型描述文档生成脚本;
- 测试数据解析脚本;
-



控制策略软件搭建工具链

■ MATLAB在软件开发环节中的应用

- 需求分析, 架构设计, 模型搭建, 单元测试, 模型检查, MIL测试, 代码生成;

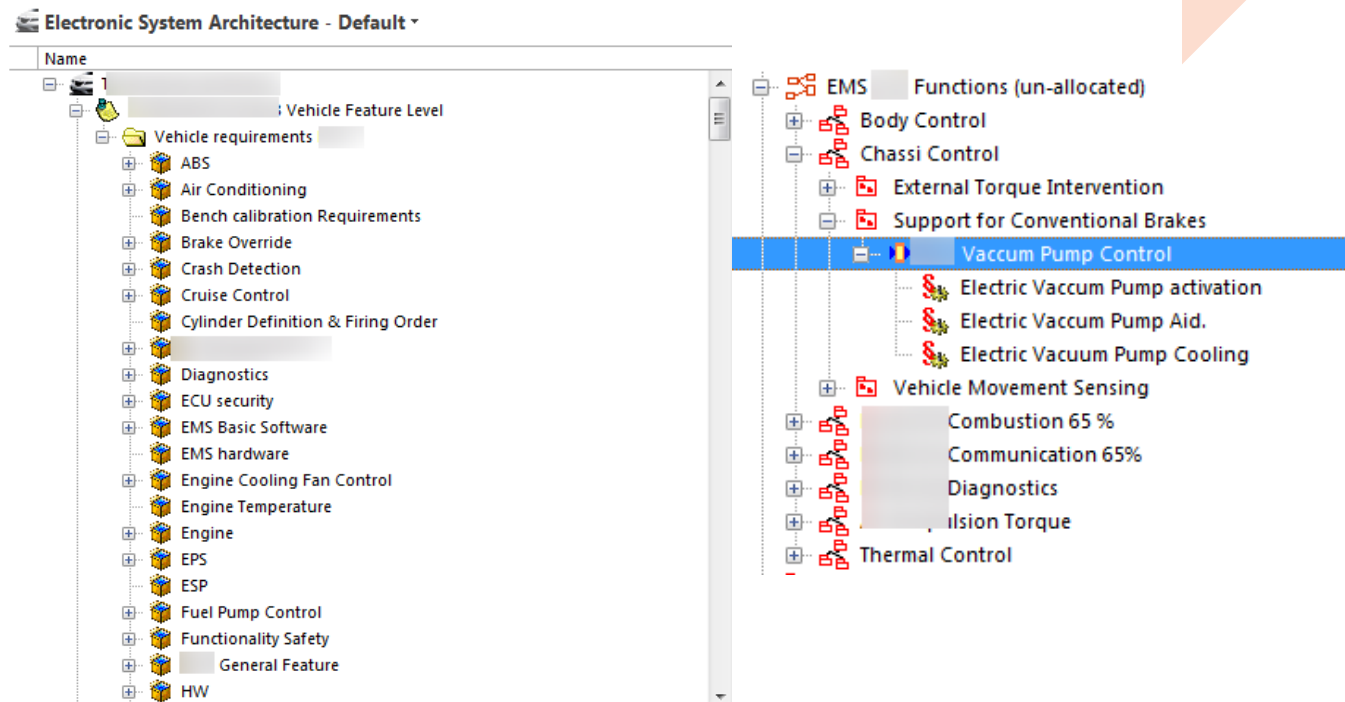


- 整车配置定义
- 由配置拆解至功能
- 由功能细化、分析成为设计需求
- 由设计需求结合设计规范、算法性能要求等形成实施级需求定义。
- 实现需求分析和软件实现的自动MAPPING关系, 查看需求实现覆盖度。

Project Requirements

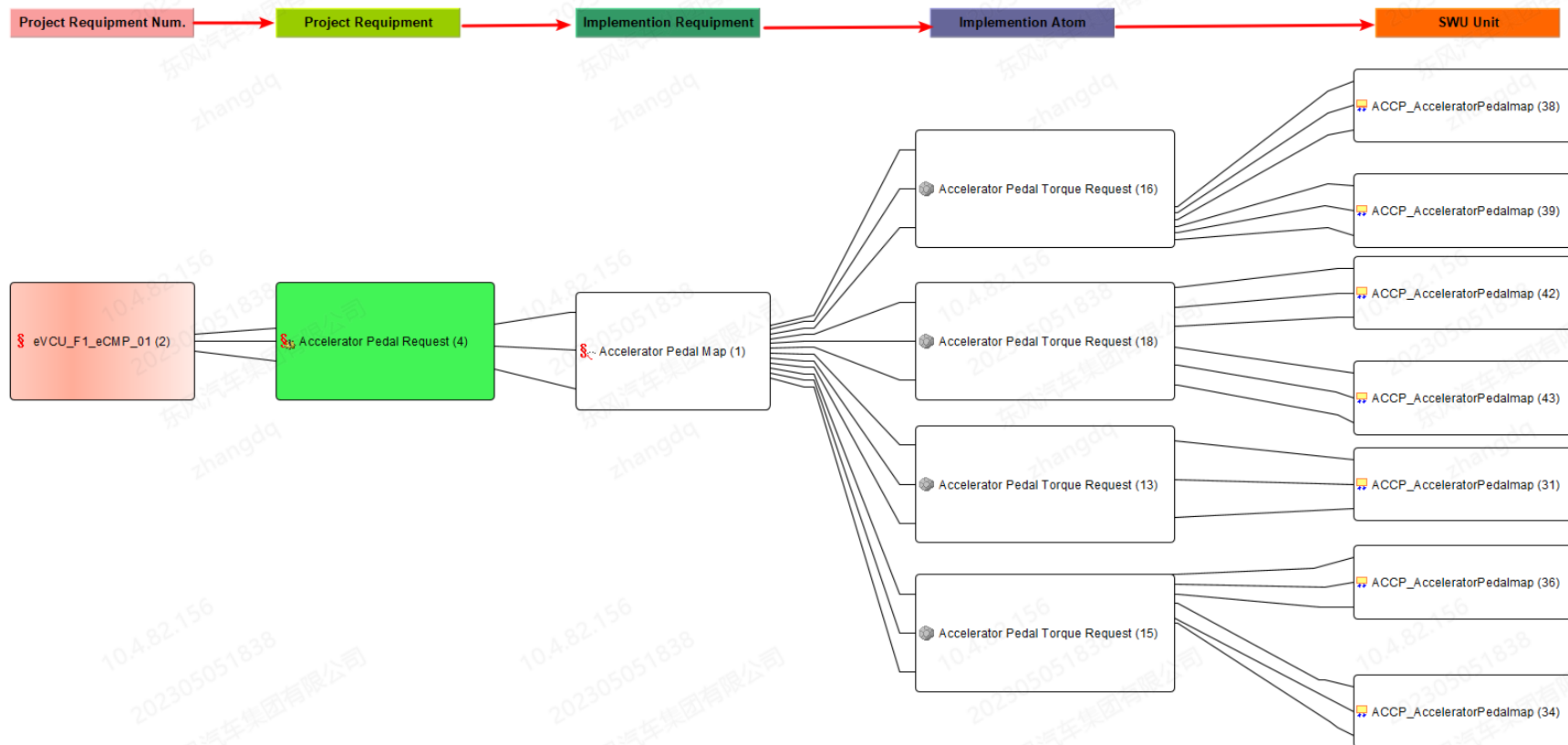
Function

Design Requirement

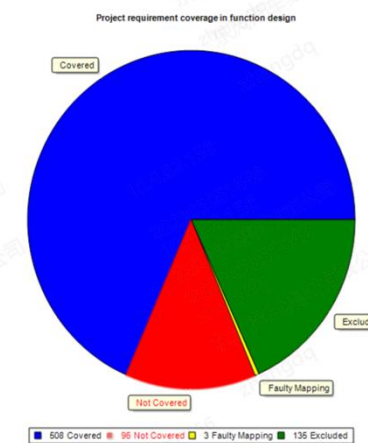


控制策略软件搭建工具链

- MATLAB在软件开发环节中的应用
 - 需求分析, 架构设计, 模型搭建, 单元测试, 模型检查, 代码生成, MIL测试;
 - ✓ 软件需求追溯;
 - ✓ 软件需求覆盖率;



✓需求分解及软件实现MAP



✓需求覆盖率统计

控制策略软件搭建工具链

■ MATLAB在软件开发环节中的应用

➢ 需求分析，架构设计，**模型搭建**，单元测试，模型检查，代码生成，MIL测试；



- 导出xml文件；
- 使用Matlab脚本生成包含已定义的子系统结构及接口的空模型；
- 填充算法，使用自定义库文件、数据类型，以减少代码量并保持数据类型一致性；
- 使用GUI界面添加测量量、标定量并使用m文件存储。

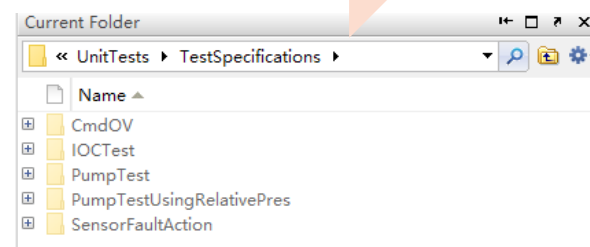
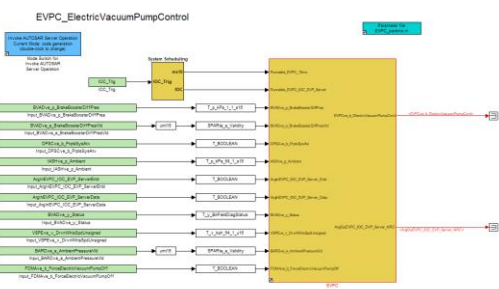
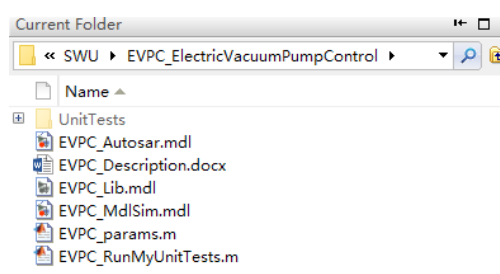
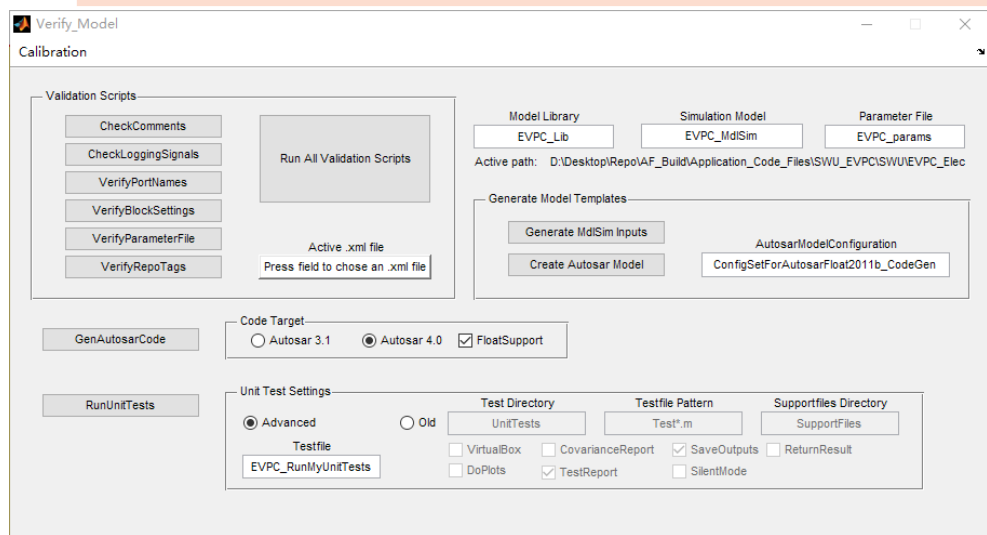
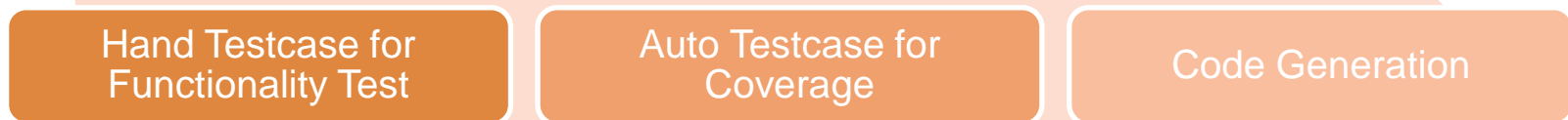
Name	Status	Version
EVPC_ElectricVacuumPumpControl	Frozen	
Vacuum Pump Actuator	CS Released	
Electric Vacuum Pump Control	CS Released	
Electric Vacuum Pump On	CS Released	
Electric vacuum pump cool down	CS Released	
Electric vacuum pump Off	CS Released	
Electric Vacuum Pump activation hysteresis	CS Released	
PumpControlAlwaysON	CS Released	
TurnPumpOn	CS Released	
t_TurnPumpOn	CS Released	
TurnPumpOff	CS Released	
t_TurnPumpOff	CS Released	
MaxPumpOnTime	CS Released	
PumpCoolTime	CS Released	
VacuumDigActionDisable	CS Released	
t_MaxRunTimeDisplayDelay	CS Released	
Electrical Vacuum Pump Command	CS Released	

控制策略软件搭建工具链

- MATLAB在软件开发环节中的应用
 - 需求分析，架构设计，模型搭建，单元测试，模型检查，代码生成、MIL测试；



- 使用脚本生成仿真模型进行单元测试；
- 编制测试用例进行功能测试；
- 使用自动测试用例进行覆盖率测试；
- 使用脚本进行Autosar封装；
- 生成代码。

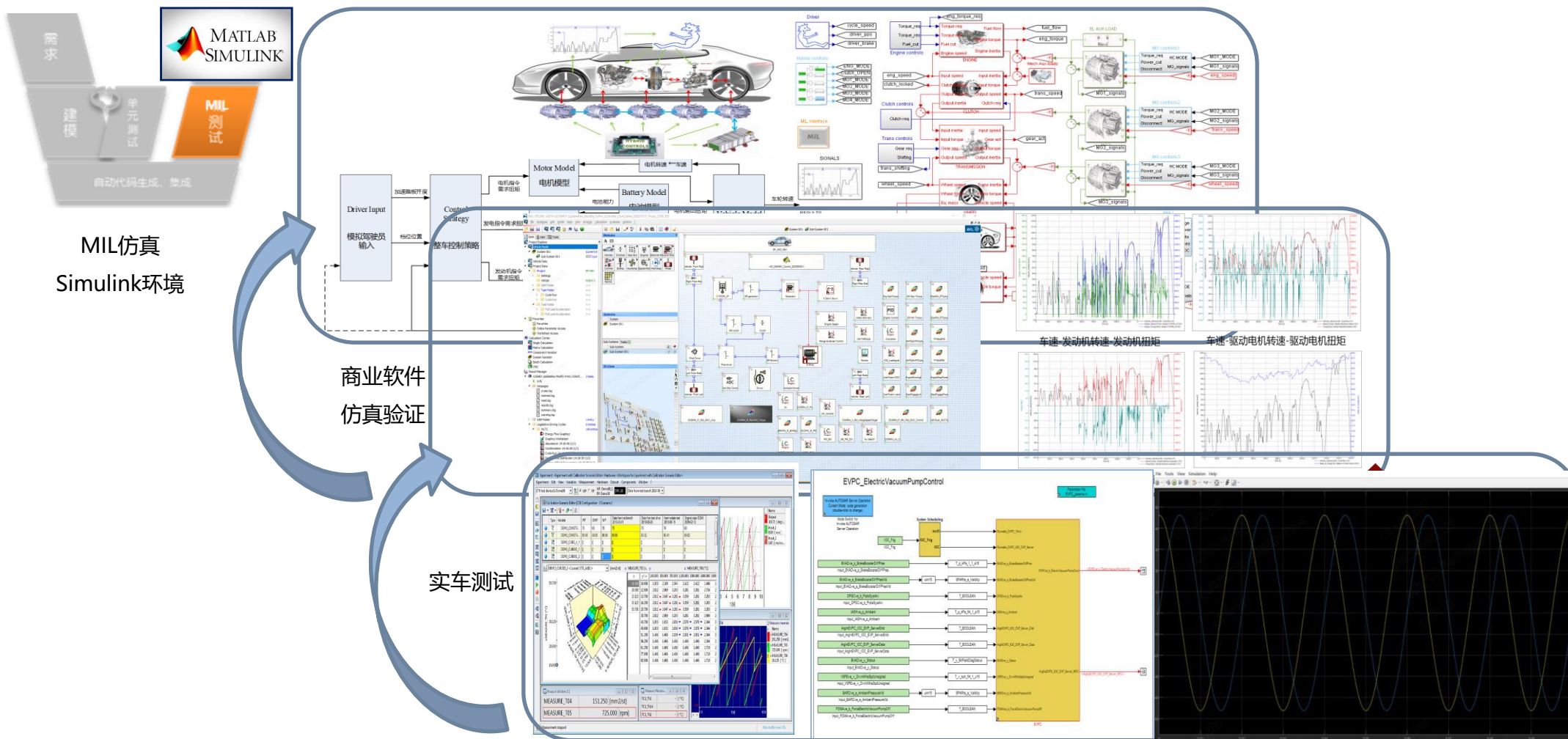


Time	Test Case	Result
57	EVPC_e_BrakeBoostPres	0.00
58	EVPC_e_BrakeBoostPres	0.00
59	EVPC_e_BrakeBoostPres	0.00
60	EVPC_e_BrakeBoostPres	0.00
61	EVPC_e_BrakeBoostPres	0.00
62	EVPC_e_BrakeBoostPres	0.00
63	EVPC_e_BrakeBoostPres	0.00
64	EVPC_e_BrakeBoostPres	0.00
65	EVPC_e_BrakeBoostPres	0.00
66	EVPC_e_BrakeBoostPres	0.00
67	EVPC_e_BrakeBoostPres	0.00
68	EVPC_e_BrakeBoostPres	0.00
69	EVPC_e_BrakeBoostPres	0.00
70	EVPC_e_BrakeBoostPres	0.00
71	EVPC_e_BrakeBoostPres	0.00
72	EVPC_e_BrakeBoostPres	0.00

1	Unit Test summary:	
2	Test coverage:	88.20%
3	Test Case	Test Stat Test run date:
4	CmdOV	PASSED 16-Apr-18
5	IOCTest	PASSED 16-Apr-18
6	PumpTest	PASSED 16-Apr-18
7	PumpTestUsingRelativePres	PASSED 16-Apr-18
8	SensorFaultAction	PASSED 16-Apr-18

控制策略软件搭建工具链

- MATLAB在软件开发环节中的应用
 - 需求分析，架构设计，模型搭建，单元测试，模型检查，代码生成、MIL测试；

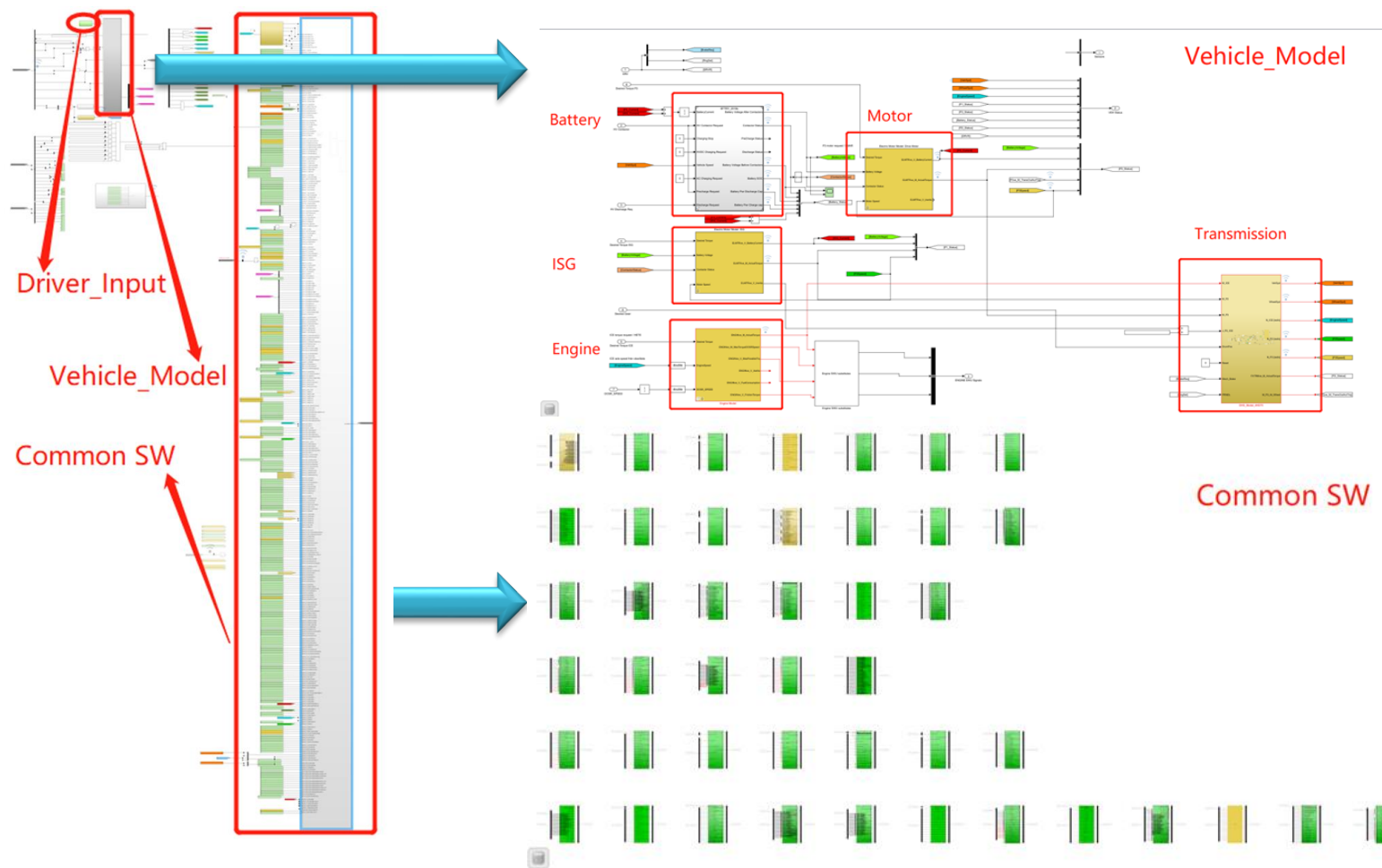


控制策略软件搭建工具链

■ MATLAB在软件测试环节的应用

➤ 虚拟仿真实例介绍-MIL仿真模型架构-MIL-Architecture

✓ MIL仿真模型架构中包含三个大的功能组块：驾驶员模型、车辆模型、通用软件（包含所有核心功能模块）。



控制策略软件搭建工具链

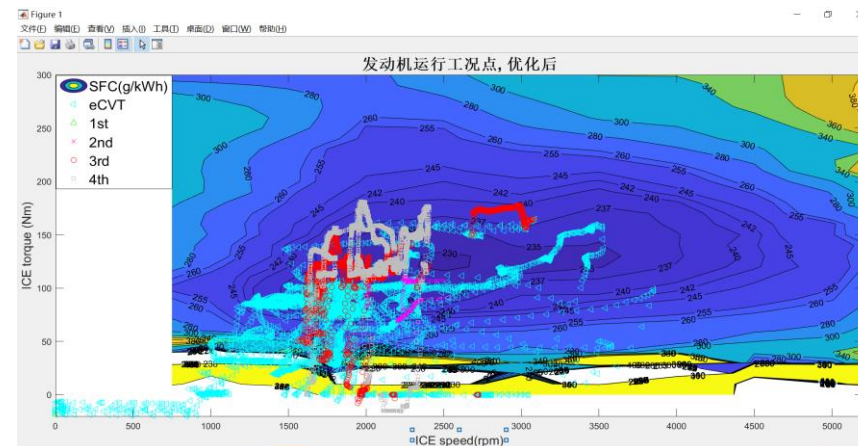
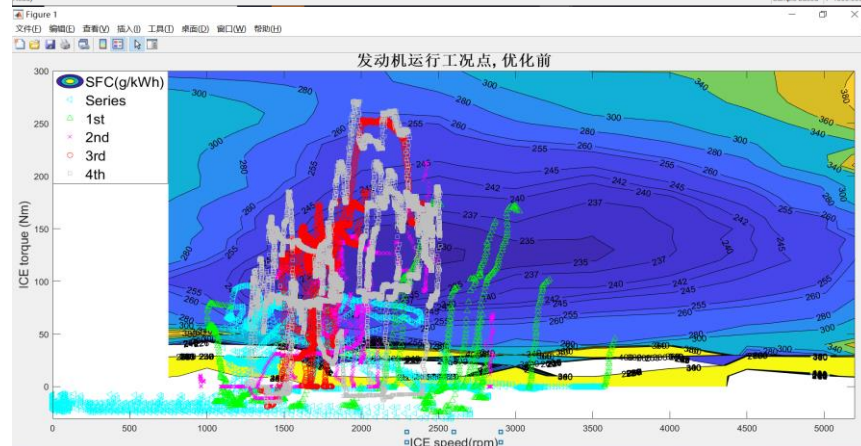
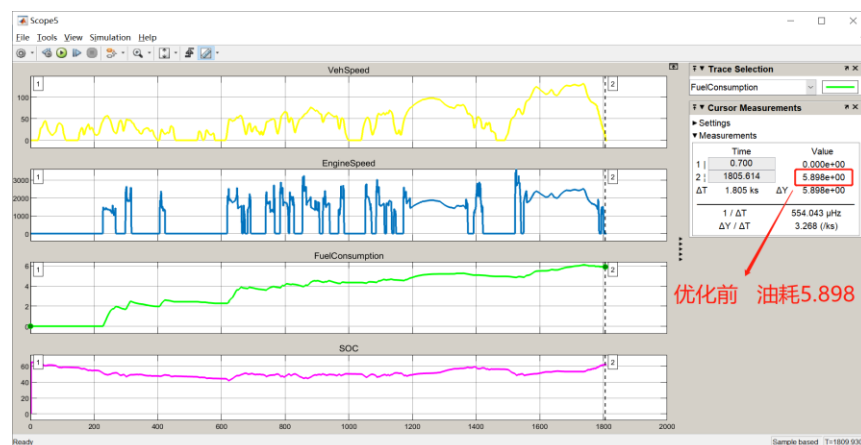
■ MATLAB在软件测试环节的应用

➤ 虚拟仿真实例介绍-使用MIL仿真模型进行经济性优化

✓ 优化混动多挡箱项目发动机启停及挡位选择：

✓ 优化前主要使用串联即并联1/2/3/4挡，发动机启停频繁；优化后主要使用eCVT和并联3/4挡，启停次数减少，工况点分布也得到优化；

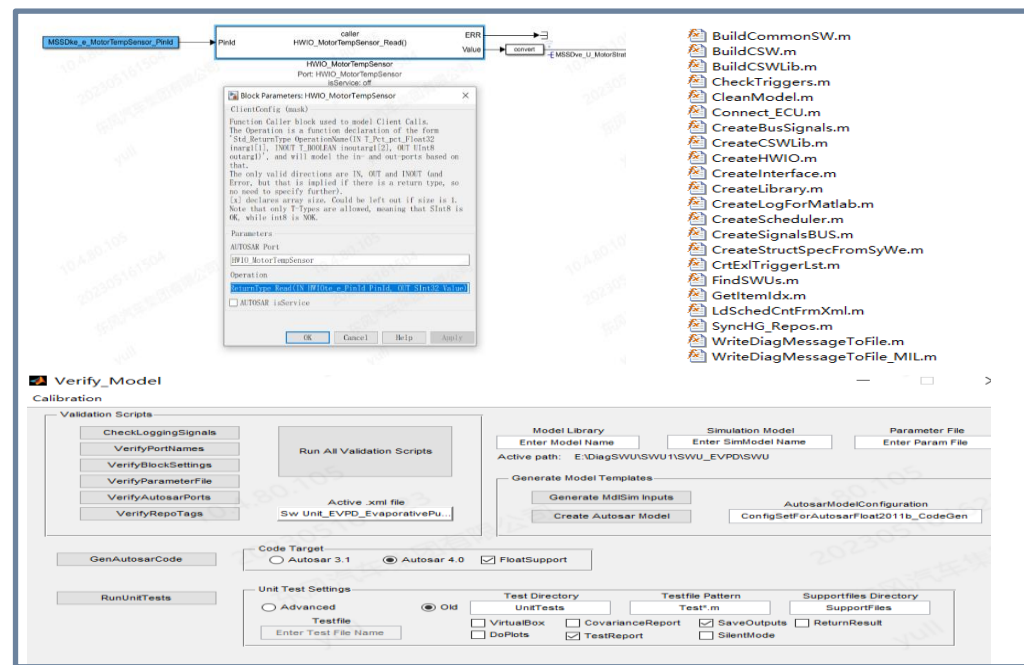
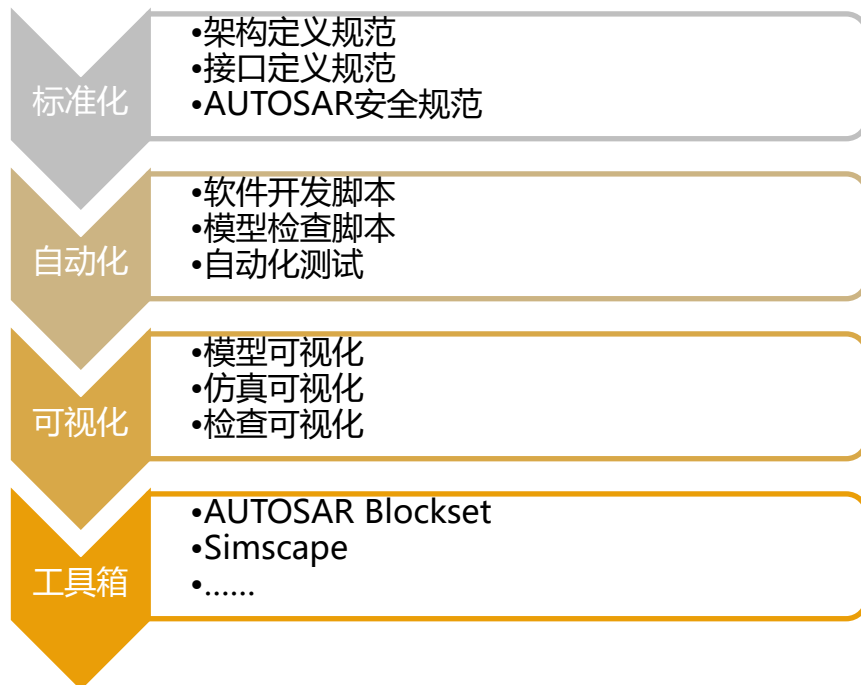
✓ MIL仿真模型验证，油耗有明显提升，实车测试也确认了该结论；



总结及建议

■ 总结:

- 1、MATLAB支持软件标准化建立;
- 2、MATLAB脚本功能支持自动化开发和测试体系建立;
- 3、MATLAB工具箱支持控制策略模型、仿真模型快速建立;
- 4、MATLAB帮助创建可视化界面, 快速继承。



创造·快乐

CREATE SMILE

感谢聆听

责任意识 现场意识 用户意识 专业意识 成本意识

责任意识 现场意识 用户意识 专业意识 成本意识



MATLAB EXPO

Thank you



© 2023 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.