

# 2020 MathWorks 中国汽车年会

基于模型预测控制的电池热管理策略

赵明如  
联合汽车电子有限公司

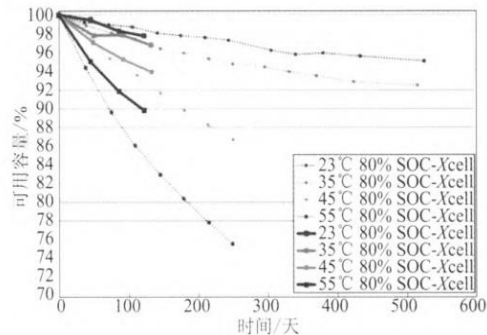


1. 项目背景介绍
2. MPC介绍
3. 算法应用及对比
4. 总结与展望

# 项目意义

## 1. 电池工作温度过高的危害

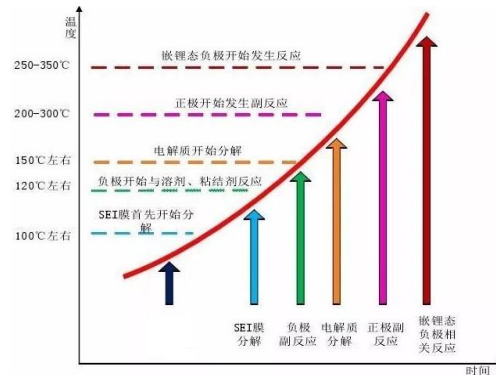
### 损害电池寿命



工作温度与电池日历寿命的关系

温度/°C	寿命/天
23	6238
35	1790
45	670
55	272

### 热失控



## 2. 常见解决方案

### 冷却方式:

被动冷却, 消耗能量很少, 但散热能力十分有限

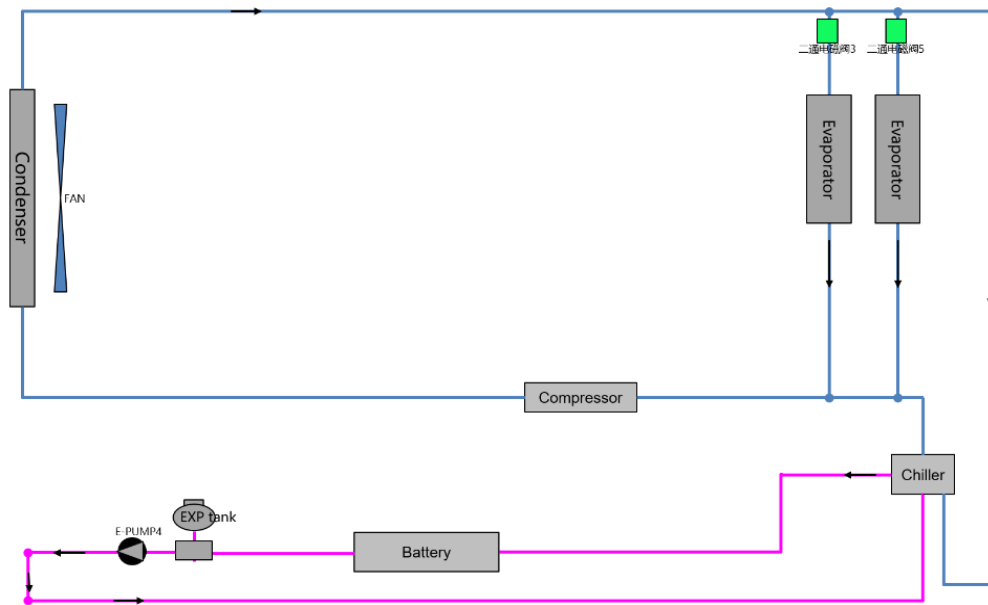
主动冷却, 散热能力强, 但能耗较高

### 控制策略

控温

节能

# 项目背景



## 任务:

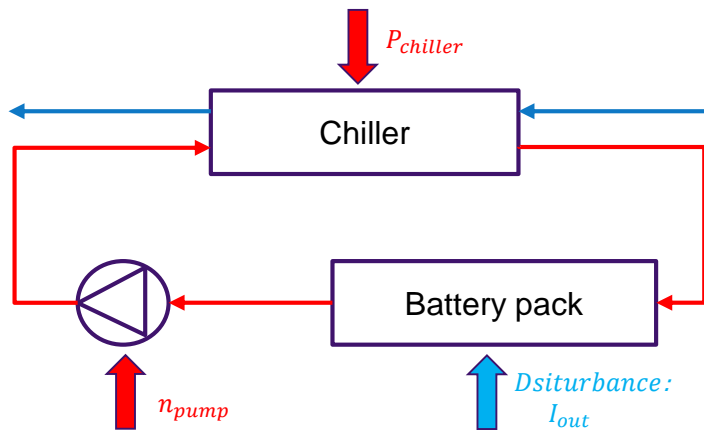
**控制对象:** 基于主动冷却的电池热管理系统;

**控制目标:** 1, 保证电池温度在安全范围; 2, 减少执行器用能;

□ **难点:** 电池冷却循环和制冷循环在能量交换以及执行器控制上相互耦合

# 系统需求

初步简化：将制冷循环当作执行器考虑，  
将电池冷却系统简化为如下结构；



执行器：电池循环泵，制冷循环  
扰动参数：电池充放电需求

系统需求：

- 使电池包的温度处于或接近适宜的范围（ $25\sim 30^{\circ}\text{C}$ ）；
- 满足控温的前提下，减少执行器能耗；
- 满足一定的实时性要求；

经过文献调研发现，相比传统控制方法，MPC算法可以实现更好的控温和节能的效果

# 文献调研

## 1. MPC 和 Rule-based Control 的结果对比

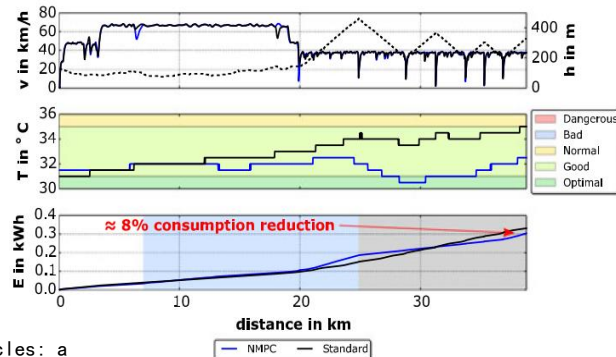
### ➤ 实车道路实验 [1]

车型: Golf GTE (PHEV)

控制方法: NMPC (vs rule-based)

计算平台: Intel Core i5 @ 2.6 GHz

冷却方式: 液冷



[1] Lopez-Sanz J, Ocampo-Martinez C, Alvarez-Florez J, et al. Thermal Management in Plug-In Hybrid Electric Vehicles: a Real-Time Nonlinear Model Predictive Control Implementation. [J]. IEEE Transactions on Vehicular Technology, 2017:1-1.

## 2. MPC 和 PID 的结果对比

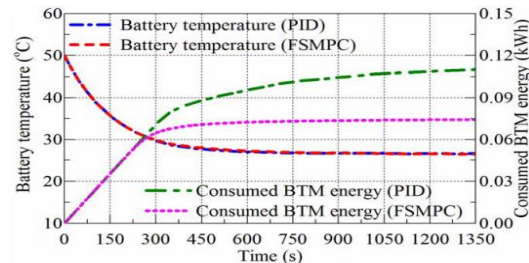
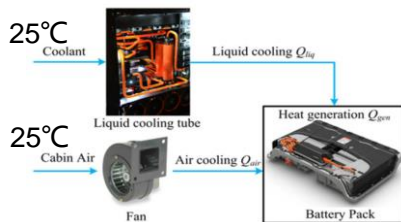
### ➤ HIL平台实验 [2]

虚拟车型: 联网自动驾驶汽车 (CAV)

控制方法: NMPC (vs PID)

计算平台: dSPACE MicroAutoBox

冷却方式: 风冷 + 液冷



在相近的控温效果下, 附件耗能比PID减少30% (under UDDS cycle)

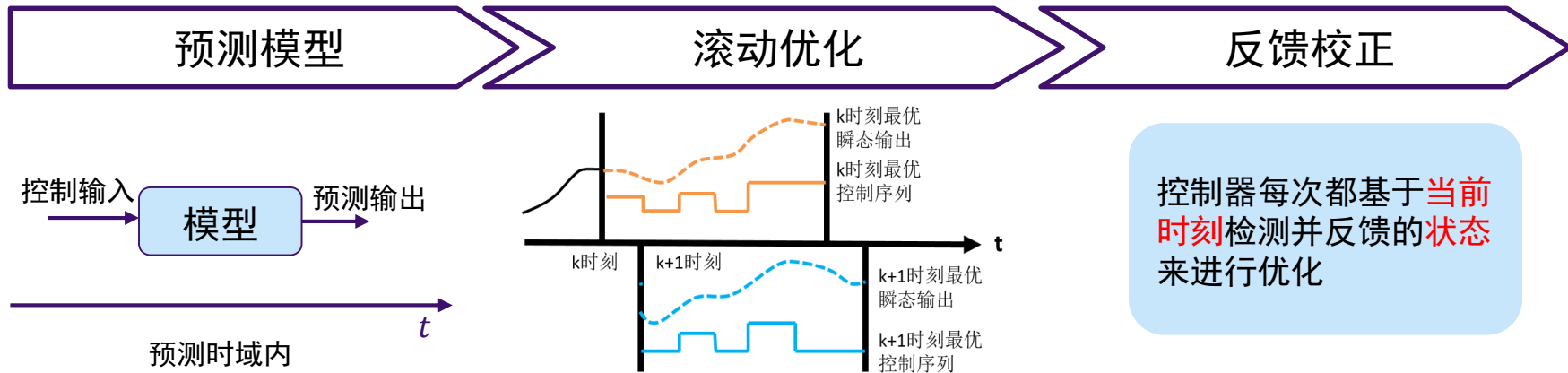
注: Reference Temperature = 27°C

[2] Chong Z, Fei L, Hua Z. Robust Predictive Battery Thermal Management Strategy for Connected and Automated Hybrid Electric Vehicles [J]. IEEE, 2018:1-1.

1. 项目背景介绍
2. MPC介绍
3. 算法应用及对比
4. 总结与展望

# 方案概念

## MPC原理介绍



### 系统特点:

- ❑ 用于**多参数控制**时，控制器的复杂度相对单参数控制不会明显增加
- ❑ 可处理**时变且非线性**系统，并方便实现**约束**条件下的优化
- ❑ 基于预测信息的输入可明显改善控制效果，顺应未来**智能网联**的技术趋势



# 方案概念

预测模型

滚动优化

反馈校正

对模型要求：速度快，精度高

主要建模部件：电池包，Chiller

建模难点：

电池包：等效电路模型，产热传热过程

状态量（4个）：

$V_1$ ：一阶RC环节电压

$T_{cell}$ ：电池包平均温度

$T_{liq}$ ：电池包冷却水出口温度

$T_{ch}$ ：Chiller冷却水出口温度

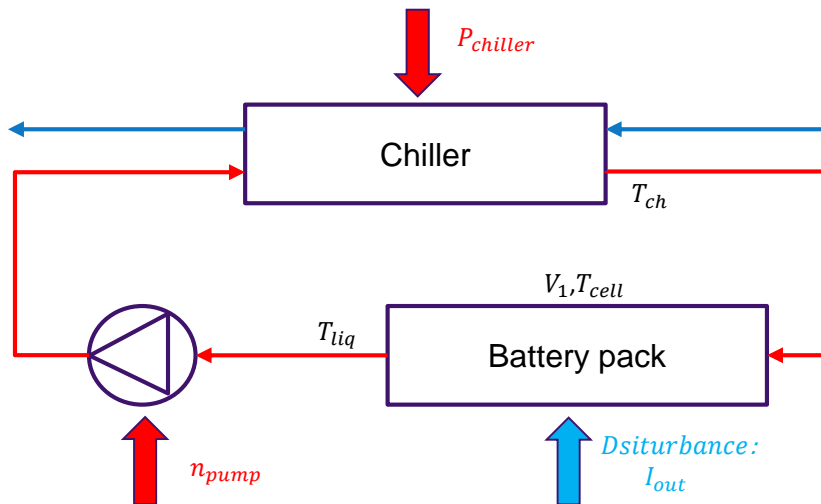
控制量（2个）：

$n_{pump}$ ：循环泵转速

$P_{chiller}$ ：Chiller换热量

扰动量（1个）：

$I_{out}$ ：电流需求



$n_{pump}$ 和 $P_{chiller}$ 都可作为控制量，此处为方便对比展示，将 $P_{chiller}$ 设为定值，仅选取 $n_{pump}$ 为控制量

# 方案概念

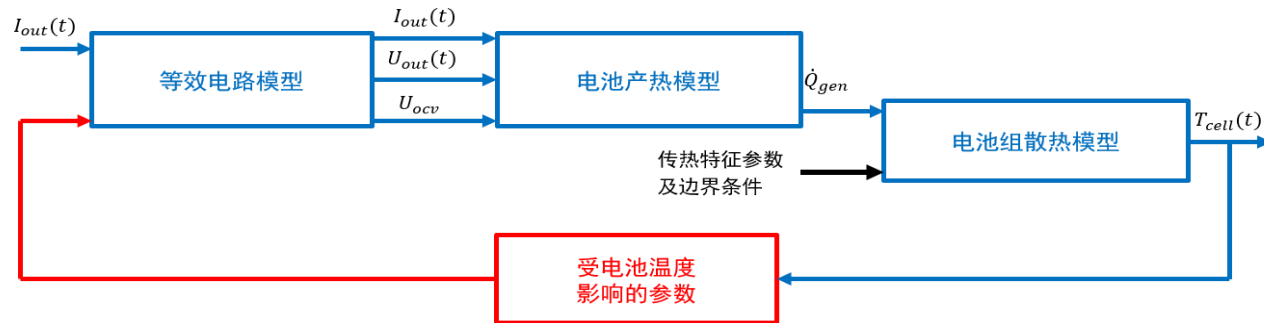
预测模型

滚动优化

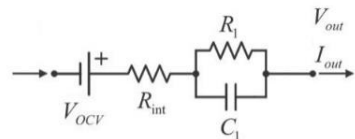
反馈校正

电池包模型:

非线性时变系统



等效电路-产热模型



其中  $V_{OCV}$ ,  $R_{int}$ ,  $R_1$ ,  $C_1$  = 函数集 ( $\text{sig}(I_{out}), T_{cell}, SOC$ )

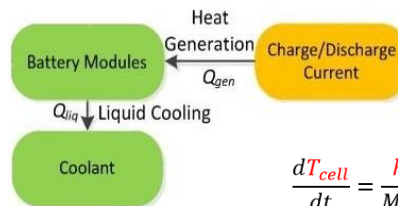
$$\frac{dV_1}{dt} = -\frac{1}{R_1 C_1} \cdot V_1 + \frac{1}{C_1} \cdot I_{out}$$

$$V_{out} = V_{OCV} - V_1 - I_{out} R_{int}$$

$$SOC(t) = SOC_0 - \frac{\int_0^t I_{out} dt}{C_{nom}}$$

$$\dot{Q}_{gen} = I_{out}(V_{OCV} - V_{out})$$

散热模型



$$m_{cell} c_{cell} \frac{dT_{cell}}{dt} = \dot{Q}_{gen} - \dot{Q}_{liq}$$

其中:

$$\dot{Q}_{liq} = hA(T_{cell} - T_{liq})$$

整理后:

$$\frac{dT_{cell}}{dt} = \frac{hA}{M_{cell}} (T_{liq} - T_{cell}) + \frac{N \cdot I_{out}(V_1 + I_{out} R_{int})}{M_{cell}}$$

# 方案概念

预测模型

滚动优化

反馈校正

## 目标函数

温度跟随控制，同时考虑能耗优化 (LQP型)

$$J = \int_{t_0}^{t_0+H_p} [(T(t) - T_r)R_T(T(t) - T_r)^T + uR_Pu^T] dt$$

$T(t)$  电池温度

$u$  执行器能耗

$T_r$  电池目标温度

$R_T, R_P$  权重系数

## 优化算法

MATLAB工具箱: Model Predictive Control Toolbox

- ① NMPC算法 (内置fmincon函数, 采用SQP寻优算法)
- ② AMPC算法 (采用LQP寻优算法)

## 反馈校正

MATLAB: Extended Kalman Filter (非线性), Kalman Filter (线性)

1. 项目背景介绍
2. MPC介绍
3. 算法应用及对比
4. 总结与展望

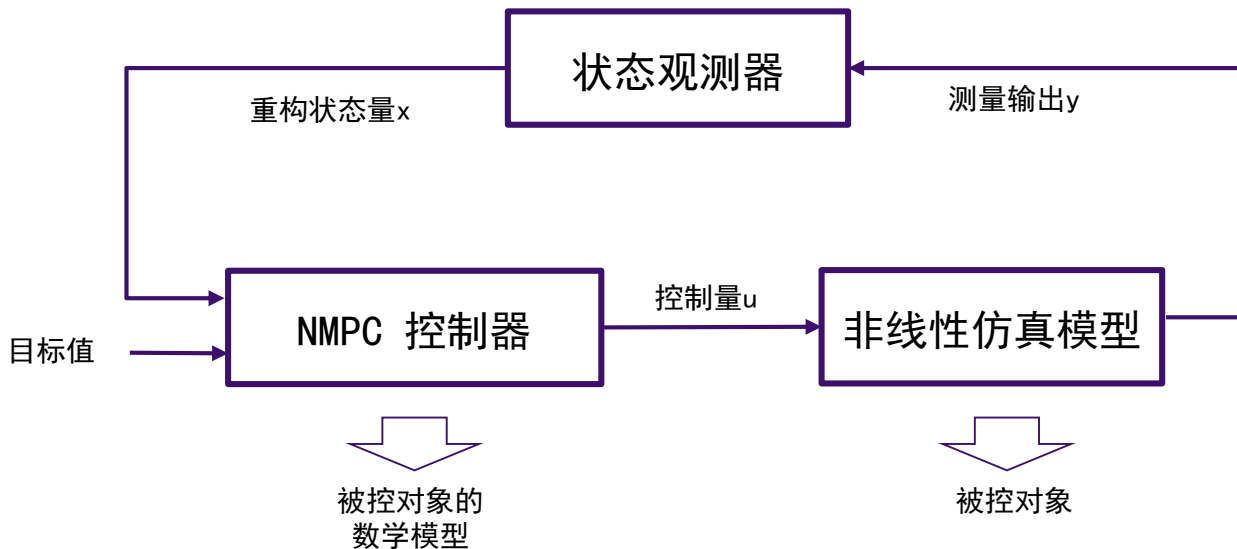
# 工具箱介绍

## MATLAB 工具箱 (Model Predictive Control Toolbox)

问题描述	常用方法	优化算法	
线性模型, 线性约束, 线性cost	线性MPC控制器	LQP	可生成 C/C++ Code
非线性模型, 线性约束, 线性cost 优化问题的结构不随工况变化	Adaptive MPC控制器	LQP	
非线性模型, 线性约束, 线性cost 优化问题的结构随工况变化	Gain-Scheduled MPC控制器	LQP	
高度非线性模型, 非线性约束, 非 线性cost	Nonlinear MPC控制器	SQP (梯度寻优)	2020a 可 生成代码

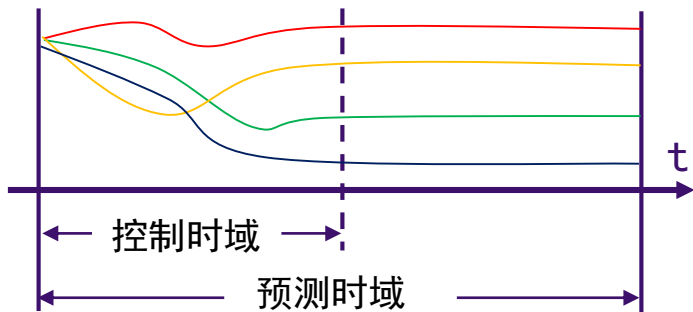
优点: 算法成熟, 鲁棒性强, 生成代码效率高;

## Nonlinear-MPC结构



## NMPC 控制器

预测时域内可能出现的u的轨迹



实质为非线性优化问题，目的是找到令预测时域内Cost function最小的u的轨迹

缺点：耗时长且无法保证全局最优

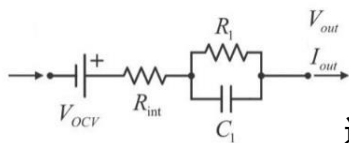
需要先在MATLAB工作区中定义“nlobj”函数体，该函数体内可以定义：

- 1, NLMP问题结构 (states, inputs, outputs等) ;
- 2, 特征参数 (时间步长, 预测时域, 控制时域等) ;
- 3, 非线性数学模型;
- 4, 限制条件 (Constraints);
- 5, 目标函数 (Cost function) 以及权重 (Weights) ;
- 6, 求解方法 (默认为基于SQP的梯度寻优)

## 状态观测器

作用：1，根据测量值和数学模型，反推出不可测的状态量；2，通过对测量信号的滤波及修正，提升反馈信号的精度；

1, 重构不可观测的状态量：RC环节电压V1



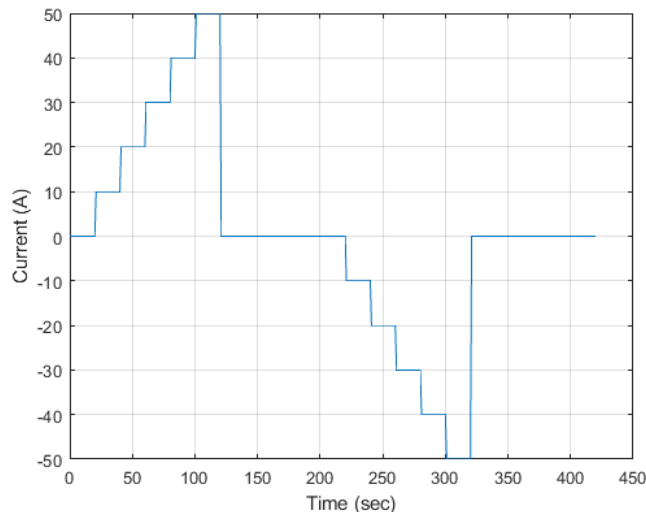
$$V_{out} = V_{OCV} - V_1 - I_{out}R_{int}$$

通过观测Vout，重构状态量V1

2, 扩展卡尔曼滤波：基于测量值和模型预测值的置信度，得到最接近真值的最优估计值

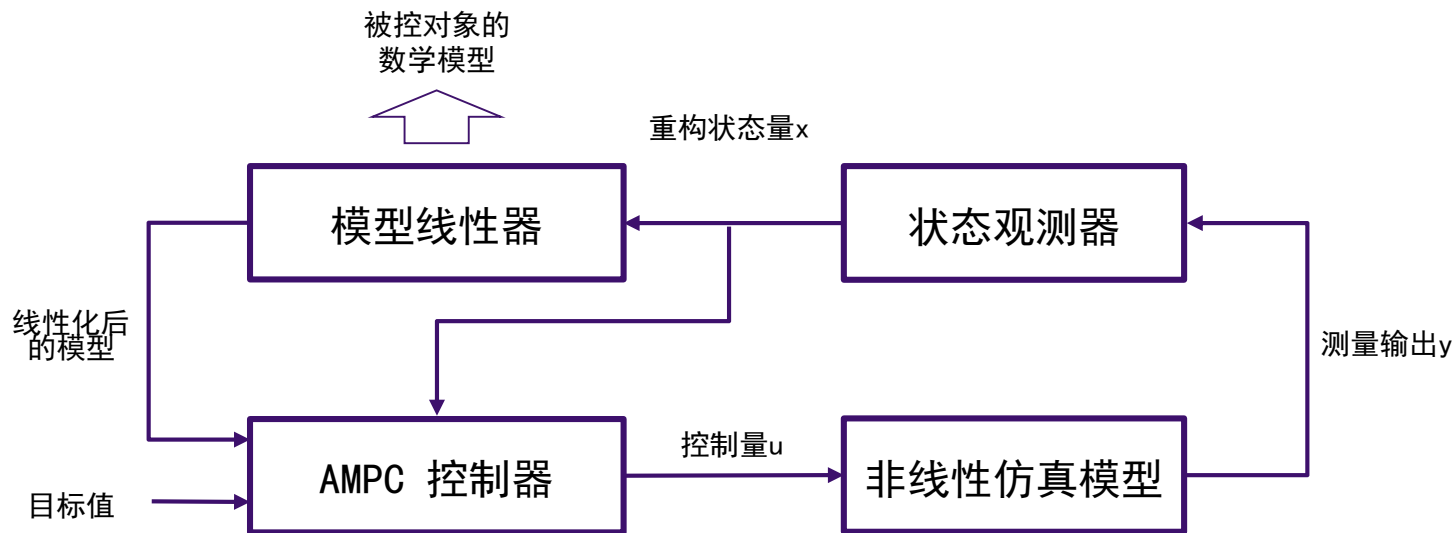
## 扰动参数

用如下阶跃电流作为系统输入  $I_{out}$

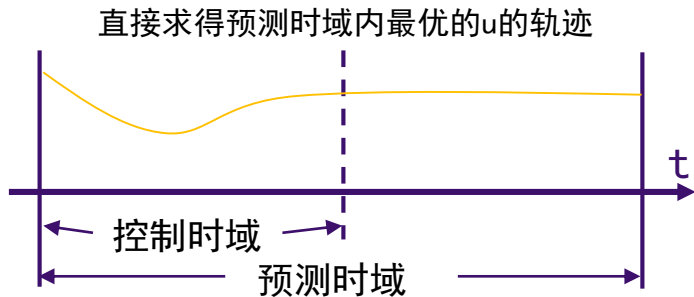




## 自适应MPC (Adaptive-MPC) 结构



## AMPC 控制器



近似转化为线性MPC问题，采用线性二次规划（LQP）可以得到全局最优解析解；  
缺点：无法自定义CostFunction，必须用线性二次规划（LQP）标准形式

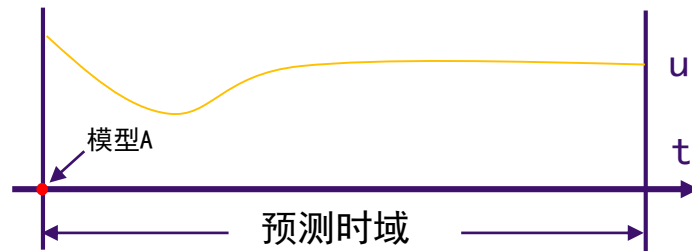
需要先在MATLAB工作区中定义“mpcobj”函数体，该函数体内可以定义：

- 1, MPC问题的结构（states, inputs, outputs等）；
- 2, 特征参数（时间步长, 预测时域, 控制时域等）；
- 3, 线性初始化数学模型；
- 4, 限制条件（Constraints）；
- 5, 权重（Weights）；
- 6, 求解方法（默认为基于LQP的线性优化求解）

## 模型线性器（手工代码）

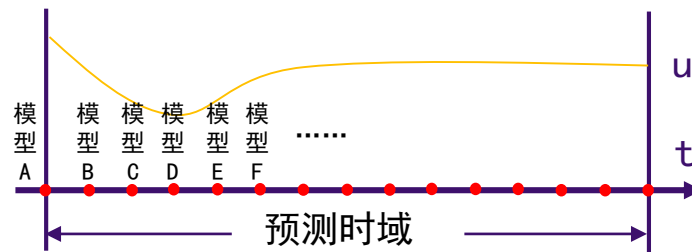
情况一：在预测时域内工况变化范围小，模型的变化可以忽略；

这时只需要在每次预测时域开始之前对模型在当前工况下进行一次线性化，在预测时域内模型不变；



情况二：在预测时域内工况变化范围大，模型的变化不可以忽略；“Successive Linearizer”

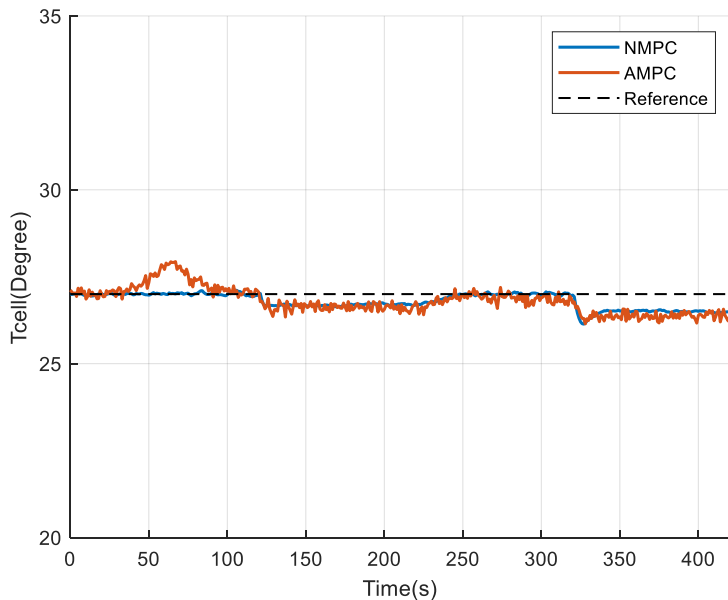
这时需要得到预测时域内的每一次时间节点下的线性化模型，在预测时域内理论上处处模型都不同；



# 仿真结果对比

## NMPC与AMPC结果对比

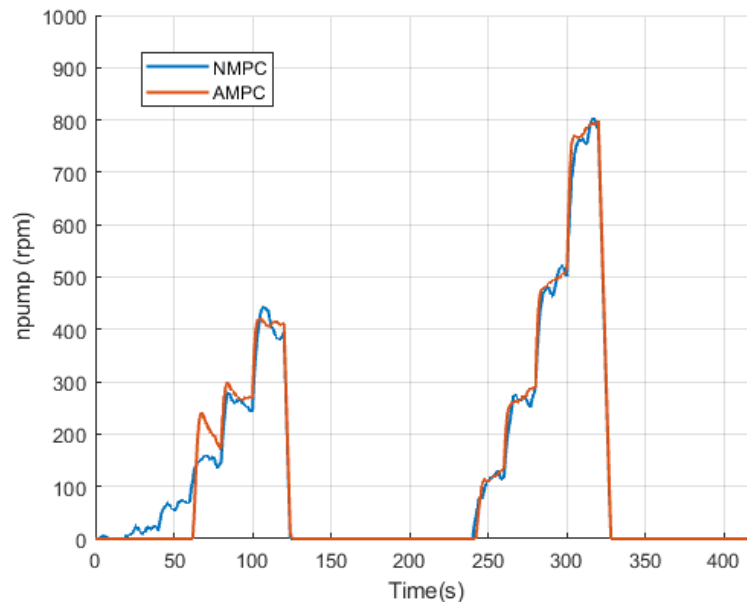
(无未来信息, 即预测时域内 $I_{out}$ 不变)



时间步长: 1s 预测时域: 20s 控制时域: 10s

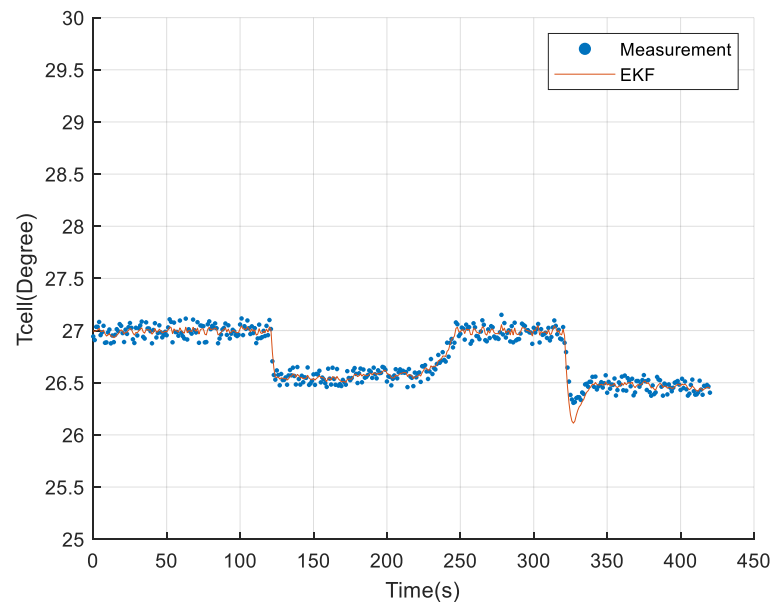
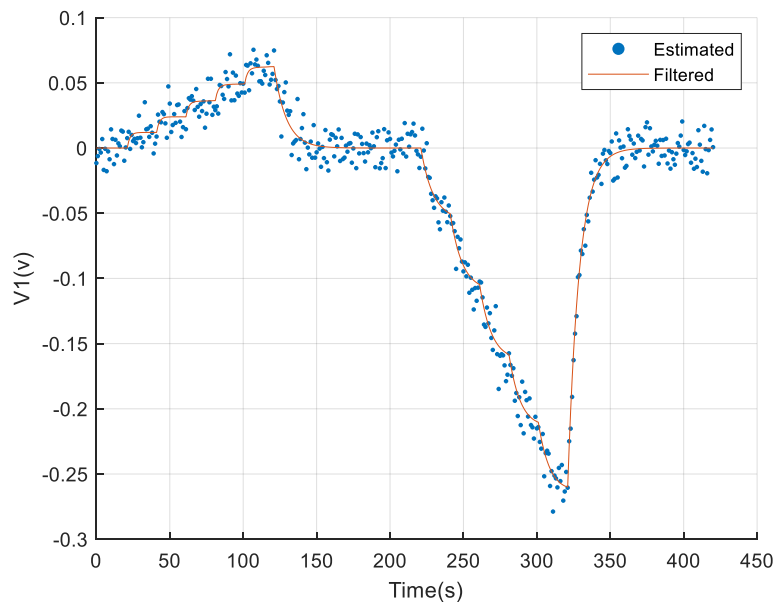
控制量constraints: [0, 1000]

控制量变化率Constraints: [-50, 50]



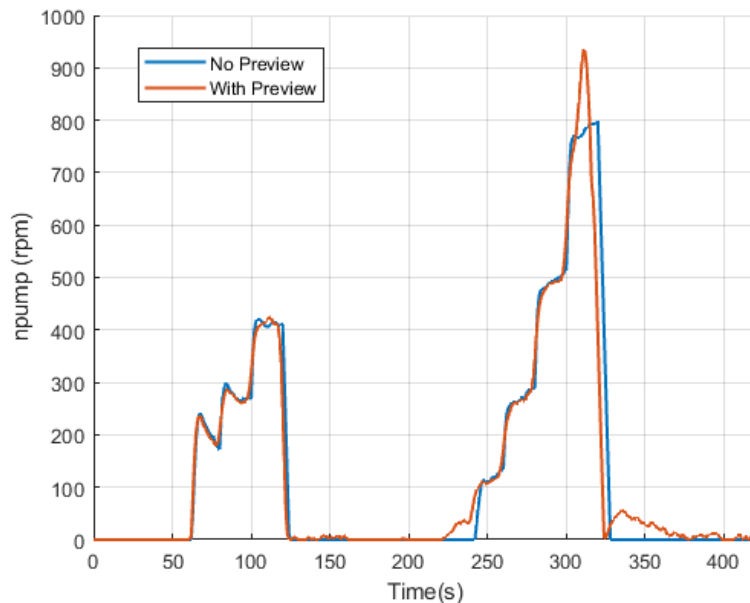
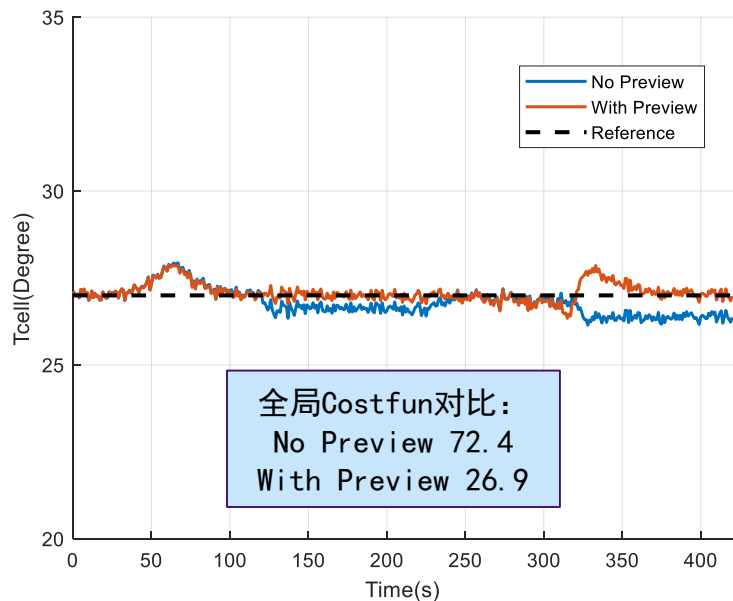
# 仿真结果对比

## 状态重构 + 扩展卡尔曼滤波器效果



# 仿真结果对比

## 有无未来信息结果对比 (预测时域中加入未来 $I_{out}$ 信息)



1. 项目背景介绍
2. MPC介绍
3. 算法应用及对比
4. 总结与展望

# 总结

- 1, MPC算法的初步尝试证明该算法可以很好的控制非线性时变系统
- 2, 在相对极小的算力要求下, AMPC可以实现与NMPC十分接近的控制效果
- 3, 加入未来信息后, MPC可以更好地实现预测控制的效果
- 4, MATLAB工具箱可以自动生成代码, 方便应用于嵌入式系统

## 在硬件中的算力消耗情况



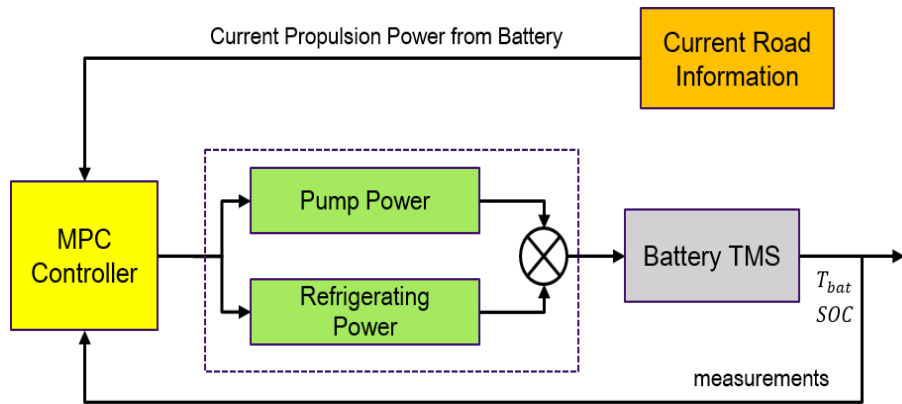
生产厂家: NXP  
主频: 1G Hz  
RAM: 2GB  
eMMC: 16GB

Algorithm	Pre-Request		Resource Exhausting		
	Thread(s)	Schedule	CPU Load	RAM	Flash
Model Predictive Control (MPC)	1	100ms	25.2%(单核)	2024KB	9M
	1	10ms	99% (单核)	2024KB	9M

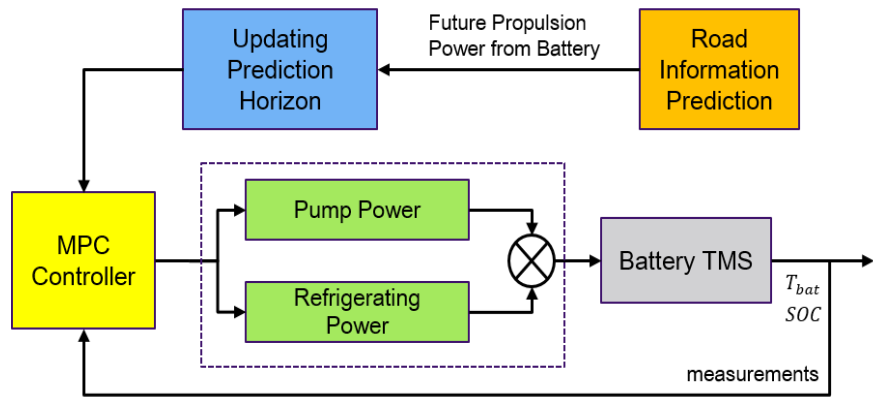


# 方案展望

## 1, 嵌入式系统方案



基础方案 (不联网)

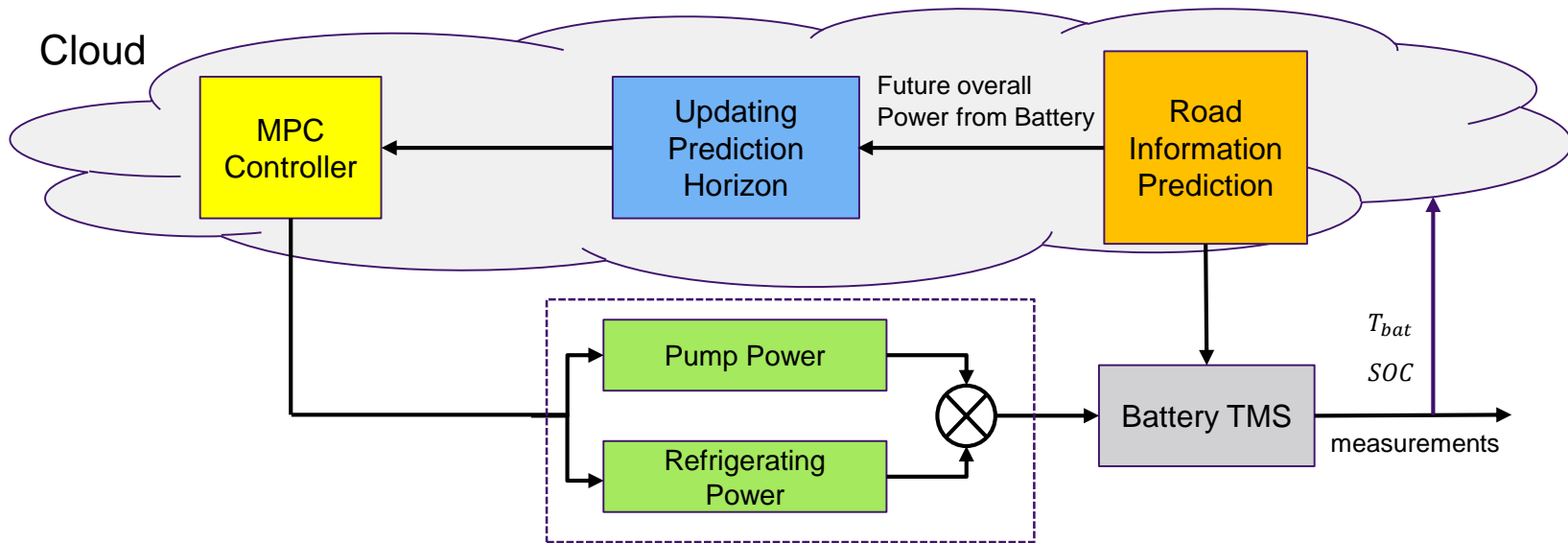


车联网方案

# 方案展望

2, 热管理控制的实时性要求不高（秒级），是有必要在嵌入式系统中计算？

云端有更强的算力和存储空间，并且工作环境不像车规级严苛





No. **1**  
In Driving  
Technology in  
China