QHRSCS：可防御漏洞攻击的四重化动态异构冗余分布式工业控制器架构–仿真报告

马卓 [1)] 周石伟 [2)] 强宇琛 [2)] 陶羽石 [2)]

队伍名称：设计安全小分队

[1)] 江苏警官学院

[2)] 东南大学

摘 要 随着工业控制系统复杂性和网络信息安全威胁的加剧，确保系统的可靠性和稳定性变得愈加重要。传统主动防御技术旨在通过额外添加模块防御攻击，但这些模块本身的安全问题就值得商榷，如拜占庭容错架构难以抵御同质漏洞攻击，移动目标防御也无法应对主动后门攻击。针对这些局限，网络空间内生安全从内在功能防护的角度出发，利用动态异构冗余、广义鲁棒控制机制等理念，实现对协同故障的强抑制，从而达到“高可靠、高可信、高可用”的内生安全功能。基于此，本文提出了一种可防御漏洞攻击的四重化动态异构冗余分布式工业控制器架构（**Q**uadruplicated **H**eterogeneous **R**edundant **S**ecurity **C**ontrol **S**ystem，**QHRSCS**），以应对动态变化的外部攻击和潜在的内在故障。**QHRSCS** 首先监测各分布式控制器系统（DCS）的安全状态并发送攻击请求，若检测到 DCS 模块在线且检测概率有效，则视为成功防御攻击，**QHRSCS** 随后分析 DCS 返回的检测结果，将检测概率转化为攻击类型并与预期匹配，成功识别后记录攻击事件；如果模块离线，系统则触发冗余切换，将负载转移至备用模块，并动态将故障模块调整为备用控制器，以确保系统的稳健性和可恢复性。**QHRSCS** 通过在线监测、冗余切换及攻击响应提升系统的稳态可用性，从而为工业控制系统提供智能化安全防护。为了贴合真实场景，本文构建了恒温箱作为真实工业控制场景，模拟不同攻击场景下的系统行为，验证了 DHR 架构在提高系统安全性和稳定性方面的有效性。实验结果表明，**QHRSCS** 架构在稳态可用性、平均无故障工作时间、平均恢复时间系统架构弹性等指标上均超过同类的防御方法。最后，本文还从控制论、信息论和博弈论等理论角度论证了 **QHRSCS** 的安全性。

关键词 分布式工业控制系统；动态异构冗余架构；内生安全；网络弹性；四重化冗余架构

1 可防御漏洞攻击的四重化冗余分布式控制器架构

针对赛题要求，本文基于 IEC61508[1] 的冗余表决架构，结合网络空间内生安全设计理念，设计了一个可防御漏洞攻击的四重化冗余分布式控制器架构模型 **QHRSCS**（**Q**uadruplicated **H**eterogeneous **R**edundant **S**ecurity **C**ontrol **S**ystem），具体的架构图可见图 3。

接下来，本文将详细阐述所提出系统设计方案的具体步骤及设计思路，以展示其如何在多层防护中有效提升系统的安全性与稳定性。

1.1 分布式控制器系统设计思路

如图 3 所示，本文的设计的 **QHRSCS** 系统采用了层次化架构，具体有以下几层：现场设备层包含异构传感器，用于采集多厂商数据，适应多场景需求。现场控制层引入基本冗余执行器和零信任安全机制，确保设备和用户的身份验证；通过微隔离和动态访问控制，限制潜在攻击的横向移动，集成异构 PLC/RTU

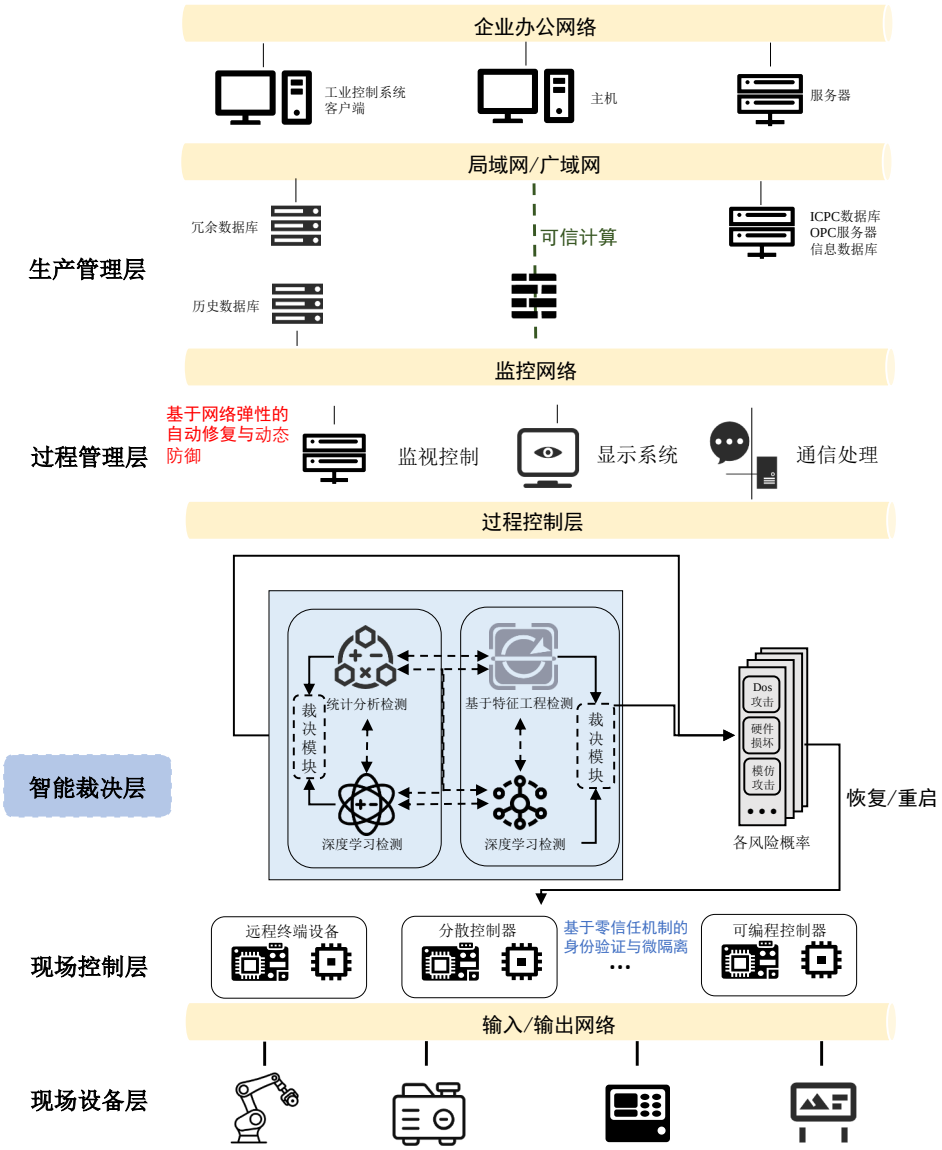


图 3　动态异构冗余内生安全工业控制架构示意图

实现兼容性，并提供冗余通信网络以保障实时可靠的控制信号传输。智能裁决层采用四重化冗余异构控制模块的 2oo4D 表决机制，将控制模块分为两组，以确保控制信号的准确性与稳定性。过程管理层包括基本冗余监控节点，引入网络弹性机制和自愈功能，实现自动修复和调整应对故障和攻击，同时多路径通信技术增强了监管信号传输的抗毁性。生产管理层中的 MES 系统通过控制层表决数据优化生产调度，支持异构 MES 集成并确保不同厂区的调度系统互通。企业资源层中的 ERP 系统基于表决数据调整生产计划，实现异构数据源的整合和跨系统数据共享，提高资源管理的效率。

在智能裁决层中，系统负责管理分布式控制器系统（DCS）各模块的安全状态监测与攻击响应。具体流程如下：

首先，系统会按指定的攻击类型向各个 DCS 模块发送请求，并监控其响应状态；当收到正常响应时，系统提取并分析 DCS 返回的数据，包括模块名称、攻击类型、在线状态和检测概率等关键信息。

其次，智能裁决层对攻击进行识别与处理：当 DCS 模块在线且具备有效的检测概率时，裁决层视其为

成功防御攻击，并将响应时间计入系统正常运行时间。同时，裁决层还会分析 DCS 返回的检测结果，并通过特定函数将检测概率值转化为对应的攻击类型。若转化结果与预期攻击类型匹配，则记录一次成功识别。

对于故障管理与冗余切换，若 DCS 模块显示离线状态，则裁决层触发冗余切换流程，将负载切换至冗余模块，并记录故障发生的时间与次数。若冗余 DCS 模块响应良好，则进行恢复操作并更新恢复次数。若冗余模块也处于故障状态，则标记攻击为有效攻击。此外，系统动态调整故障模块为备用控制器，确保系统的稳健性和可恢复性。

通过以上机制，智能裁决层在攻击识别和故障管理过程中实时统计系统的正常运行时间、故障时间、故障次数和恢复次数，最终汇总并返回各模块的识别结果和状态，以便后续进行系统安全性分析。裁决层通过对分布式控制模块的在线监测、冗余切换及攻击响应来提升系统的稳态可用性，为整个工业控制系统提供智能化的安全防护与响应机制。

1.2 真实场景控制系统安全与攻击模式模拟设计

为了最大限度地模拟真实工业控制场景，本文设计了一个恒温箱控制实验。恒温箱是一种用于精确控制内部温度的设备。其基本功能是通过内部传感器实时监测箱内温度，并根据设定的目标值自动调节温控系统，以维持一个稳定的温度环境。这种设备通常具备升温和降温功能，并可在极端温度条件下保持长时间的稳定性。恒温箱的设计旨在确保环境因素的最小化干扰。具体而言，本文采用“安全控制”与“攻击模式”两种操作模式，以仿真不同安全状态下的控制系统响应。恒温箱初始设定了温度范围，“安全控制”模式旨在确保恒温箱温度始终保持在该范围内，并具备应对温度异常的自动响应机制；而“攻击模式”则反其道而行，旨在模拟系统被攻破后的不稳定状态。

在“安全控制”模式下，控制系统会根据当前温度状态，自动执行以下调节措施：

(1) 当温度高于设定的最高值时，控制系统将主动降低温度，以确保其不超过上限。

(2) 当温度低于设定的最低值时，控制系统将提升温度，维持恒温箱的正常运行。

(3) 若温度处于设定的正常范围内，系统将不采取调控操作，确保温度的自然稳定。

在系统受到攻击时，控制系统向恒温箱服务发送攻击请求，迫使其温度控制发生异常响应。在“攻击模式”下，系统会根据传感器反馈的当前温度实施以下操作，以模拟受攻击状态下的高风险行为：

(1) 当温度超过最高设定值时，系统将进一步升温，以模拟攻击下温度失控的风险情境。

(2) 当温度低于最低设定值时，系统将进一步降低温度，破坏正常的温度调控。

(3) 若温度处于正常范围内，系统则随机选择升温或降温，以增加温度波动并降低控制系统的稳定性。

1.3 本章小结

本章节具体描述了一种基于 IEC61508[1] 标准的四重化冗余分布式控制器架构 **QHRSCS**，以增强工业控制系统在网络安全威胁下的抗攻击性与稳健性。该架构结合内生安全理念，设计了包括现场设备层、控制层、智能裁决层、过程管理层、生产管理层和企业资源层的多层级防御体系。各层级通过引入异构传感器、冗余执行器、微隔离、动态访问控制、网络弹性和多路径通信等技术，实现从物理设备到高层管理的全面安全防护。智能裁决层在系统内生安全设计中尤为关键，负责动态监测分布式控制器系统（DCS）各模块的状态和攻击响应：系统若检测到 DCS 模块离线，则触发冗余切换以保证任务的正常执行；若检测到模块

在线并成功识别攻击，系统将记录事件并执行相应防御措施。为验证架构的安全性与稳健性，本文设计了恒温箱控制实验，模拟了“安全控制”和“攻击模式”两种情境。在安全模式下，系统基于温度调节实现稳定控制；在攻击模式下，模拟温度失控情境，评估系统在受攻击状态下的响应与恢复机制。**QHRSCS** 为工业控制系统的智能安全防护提供了新的架构参考。

2 仿真报告

本章节首先阐述了本文仿真实验的具体设置细节，然后对 **QHRSCS** 在不同攻击下的平均故障间隔时间（MTTF）、平均恢复时间（MTTR）和可用性（Availability）等安全性指标进行计算，同时还考虑了冗余、异构等设计的有效性，以及真实场景下的网络弹性指标。

2.1 指标设计与仿真目标

在现代工业控制系统中，平均故障间隔时间（MTTF）、平均恢复时间 (MTTR) 和可用性是评估系统性能与可靠性的关键指标。MTTF 衡量系统在正常运行条件下的平均工作时间，其重要性在于能够有效评估系统的可靠性与故障预测能力，为决策提供支持；而 MTTR 则指系统发生故障后恢复至正常状态所需的平均时间，低 MTTR 表明系统具有良好的恢复能力，能够提高运维效率并提升客户满意度。可用性则反映了系统在特定时间段内提供服务的概率，直接影响服务水平承诺及业务连续性保障。通过对 MTTF、MTTR 和可用性的监测与分析，从而实现系统性能的持续提升与风险的有效控制。因此，确保高水平的 MTTF、低 MTTR 和高可用性是实现工业控制系统安全性与稳定性的基础。

冗余机制通过增加备用组件来提高系统的可靠性，确保在关键模块故障时能够迅速切换到冗余模块，从而降低系统停机时间；异构设计则通过引入多样化的组件和环境，增强了系统对不同攻击类型的适应性和容错能力。这种设计理念能够有效分散潜在的风险，避免因单一故障导致的全面崩溃。此外，网络弹性指标作为评估系统在面对网络攻击或故障时维持正常运行能力的重要参数，能够反映系统对外部冲击的适应能力和恢复能力。在真实恒温箱场景下，通过对冗余与异构设计的有效性进行综合评估，结合网络弹性指标的监测，能够实现对系统安全性与稳定性的全面把控，为实际应用提供有力支持。

2.2 仿真实验设置

2.2.1 概览

为了尽可能仿真系统的**异构**性质，本文在实验中采用了 `Docker` 作为异构 DCS 的载体，使用 `Python` 模拟 DCS、主控制器与真实工业场景的恒温箱的行为，通过 `Flask` 框架实现 DCS、主控制器与恒温箱通信。具体的文件树如下表所示。

DCS_Simulator 文件结构

```
DCS_Simulator
├── docker-compose.yml                    # 多 docker 启动脚本
├── Dockerfile.DCS1                       # DCS1 的 Dockerfile, 下同
├── Dockerfile.DCS2
├── Dockerfile.DCS3
├── Dockerfile.Incubator                  # 恒温箱的 Dockerfile
└── Dockerfile.RedundantPLC
```

```
├── ics_security
│   └── plc_simulator.py                    # DCS 控制器模拟脚本
├── incubator_simulation.py                 # 恒温箱具体的模拟代码
├── main_controller.py                      # 主程序
└── README.md
```

具体而言，异构分布式控制系统（DCS）和恒温箱实例通过不同的 `Dockerfile` 创建，通过主控制器来控制攻击行为。

本文提出的 **QHRSCS** 首先仿真了主控制器，该控制器负责向分布式控制系统（DCS）发送攻击请求，并触发恒温箱服务以模拟攻击。同时，系统根据 DCS 的响应和恒温箱服务的反馈计算安全性指标。

2.2.2 重要假设

- **QHRSCS** 控制的每一个异构 DCS 模块有至少一个固定的漏洞，也就是说，**每一个 DCS 都有至少一个不同攻击类型能被其攻击下线；**
- 即使异构 DCS 模块成功防御攻击，其仍有一定概率将其**误判为其他类型的攻击**；
- 当异构 DCS 模块被攻击下线，冗余 DCS 会**顶替**下线的 DCS，此时下线的 DCS 经过重启等恢复操作重新成为新的冗余 DCS。

2.2.3 攻击与防御仿真

在攻击类型的选择上，系统随机选择五种攻击类型，包括恶意软件（malware）、钓鱼（phishing）、勒索软件（ransomware）、分布式拒绝服务（DDoS）及未知类型（unknown）。通过对 DCS 返回的检测结果进行分析，系统能够将检测结果转换为具体的攻击类型。

如图 4 所示，系统通过发送攻击请求至各 DCS，获取其在线状态及检测概率。输出的检测概率是 One-Hot 编码，类似于 `[0.0061, 0.9787, 0.00377, 0.0078, 0.0034]`，代表了模型对每个类别识别的置信度。总体而言，DCS 有两种状态，一种是抵御住了这次攻击，仍旧在线；一种是被攻击成功，离线。如果 DCS 在线且检测出的攻击类别正确则认为防御成功，值得注意的是，即使 DCS 在线，其仍有一定的概率返回错误的识别结果，也就是有误判的概率。当 DCS 离线时，系统将触发冗余 DCS 的继续上述步骤；如果冗余 DCS 也被攻击离线，则认为攻击成功。

系统总结所有冗余 DCS 的防御/攻击结果，进行表决并将结果发送至恒温箱，如图 5 所示。如果防御成功，则进行正常逻辑的控制恒温箱操作；如果攻击成功则进行被攻击状态下的破坏恒温箱温度操作，如图 5 所示。

实验使用的温度数据取自恒温控制系统模拟实验。设定的温度阈值范围为 $[36.0, 38.0]$ 度。

2.3 仿真结果与分析

通过对比不同方案的仿真结果，可以得出如下结论：

如表 1 所示，在安全性指标方面，本文方案在平均故障间隔时间（MTTF）、平均修复时间（MTTR）和可用性（Availability）上表现出了明显优势。具体而言，本文方案的 MTTF 为 0.0261，远高于其他变体，表明其具有更高的系统可靠性。MTTR 为 0.0145，显著低于其他方案，说明其在故障发生时能够更快速地恢复。可用性方面，本文方案为 0.3932，明显优于其他方案，表明其在大多数情况下能够保持较高的系统运行效率。

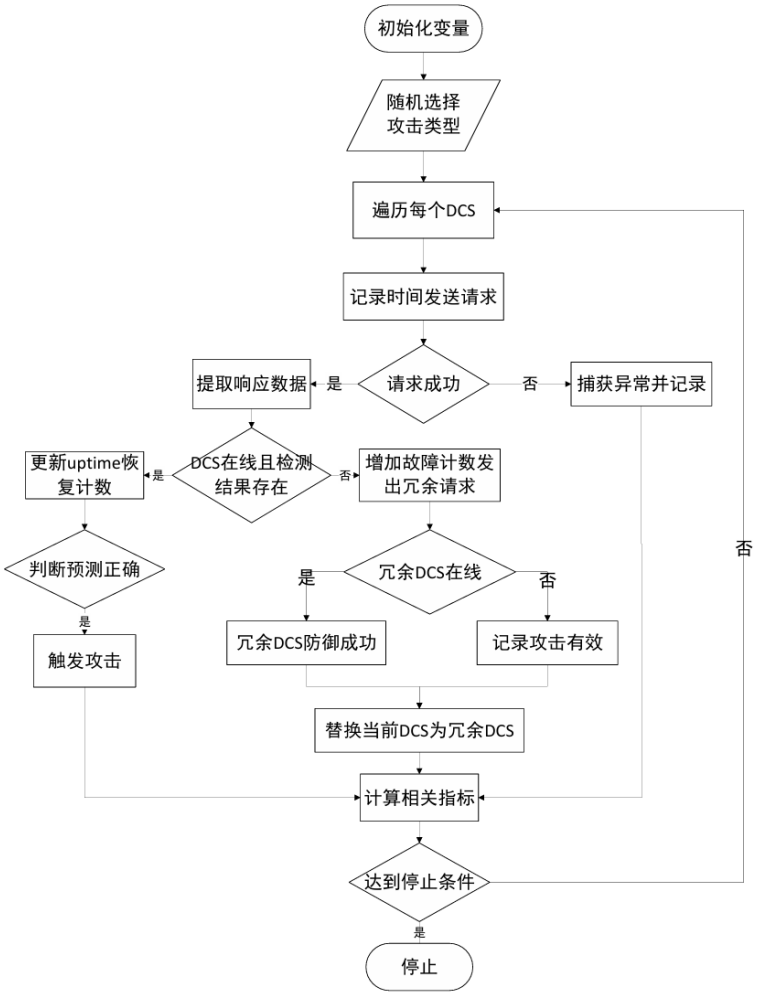


图 4 主控制器流程示意图

Table 1: 不同方案下的平均安全性指标

方案	MTTF	MTTR	Availability
三重化变体	0.0068	0.0408	0.3230
去掉冗余机制的变体	0.0086	0.1844	0.0460
去掉同质的变体	0.0103	0.1186	0.0809
本文方案	**0.0261**	**0.0145**	**0.3932**

如表 2 所示，在网络弹性方面，本文方案与三重化变体在 **在阈值范围内的时间比例**上相同，均为 0.94，表明两者在维持系统稳定性方面具有相似的表现。然而，去掉冗余机制的变体和去掉同质的变体在此项指标上的值较低，分别为 0.87 和 0.90，显示出其在稳定性控制方面的不足。在 **超出阈值范围的偏离平均值**方面，本文方案的偏离值为 0.65，低于其他变体，表明其能够更好地控制网络的异常波动，提供更为稳定的性能。尤其是在 **恢复时间（步数）**方面，本文方案仅需 1 步即能恢复系统，而其他变体需要 4 步或 5 步，体现出本文方案在故障恢复上的迅速响应能力，显著降低了系统的停机时间。

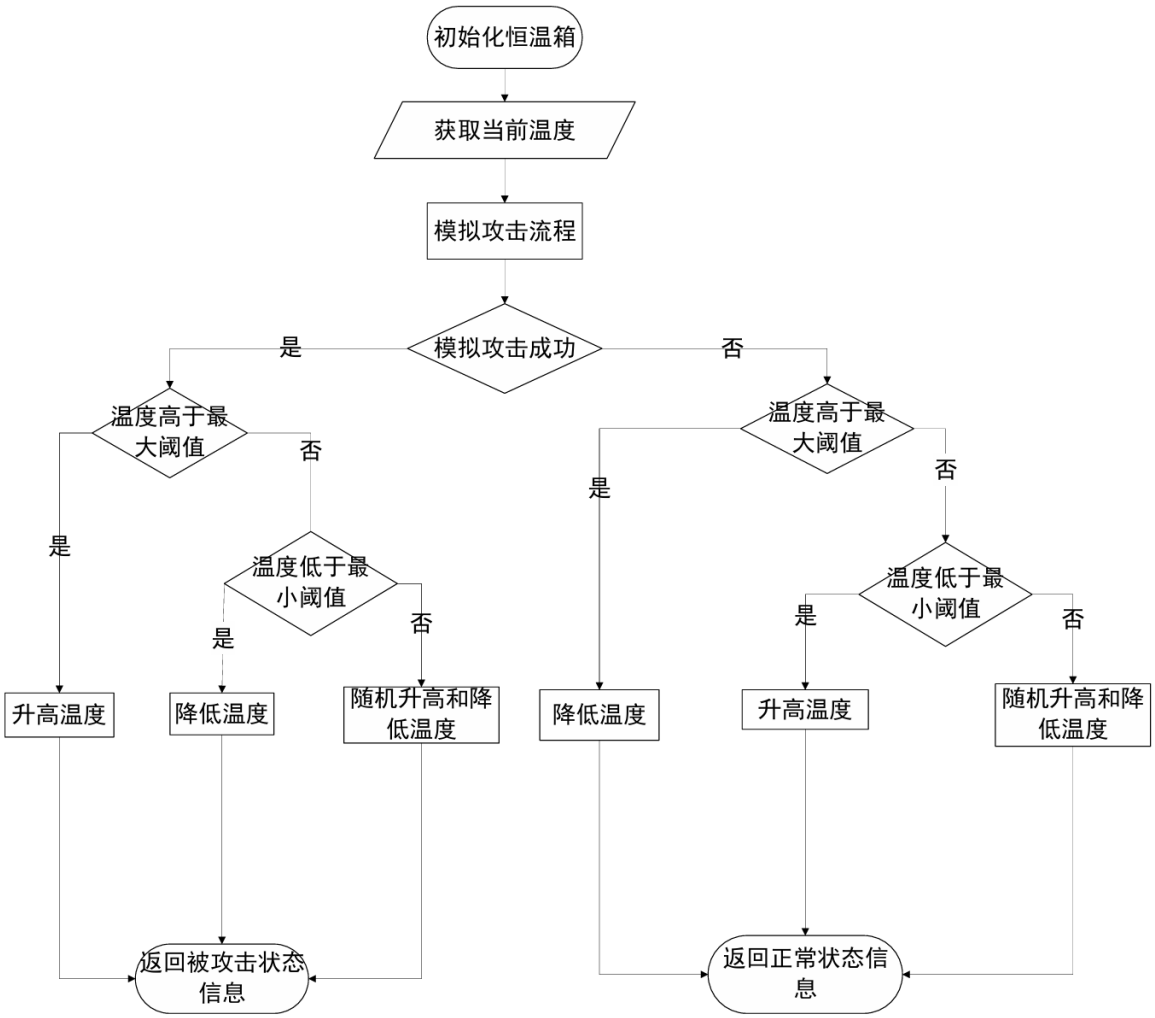
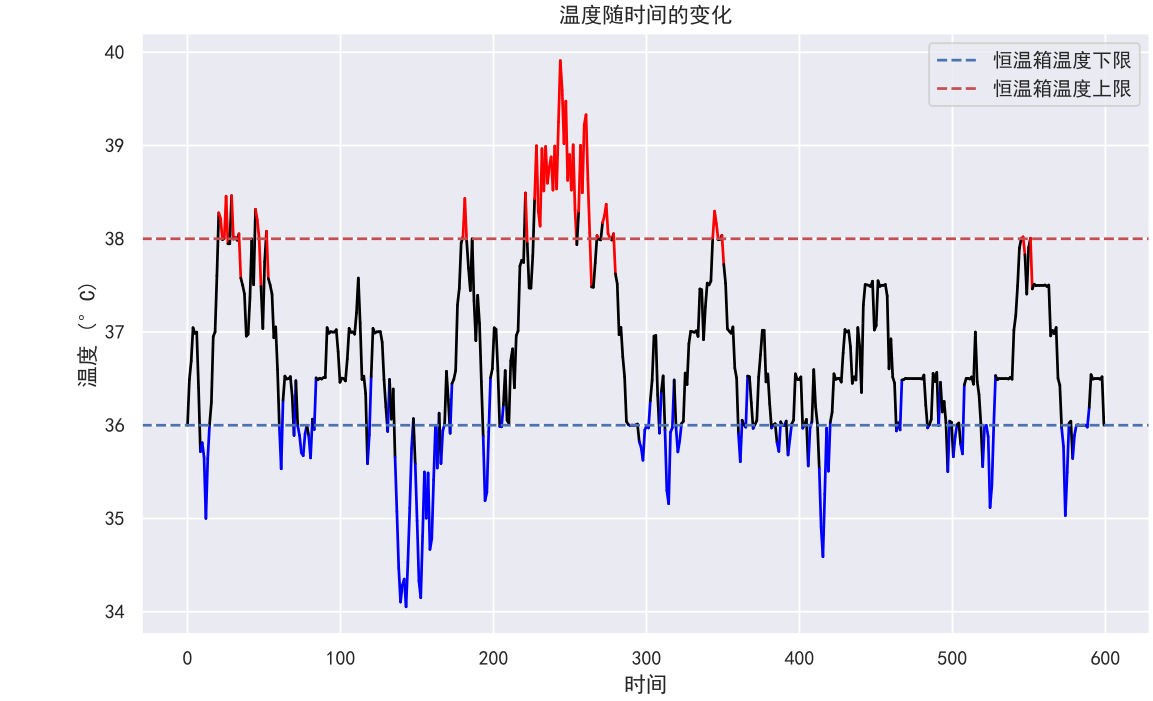


图 5　恒温箱流程示意图

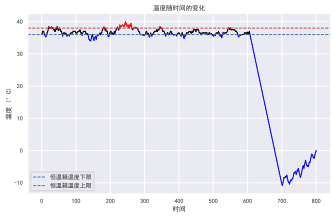
Table 2: 不同方案下的网络弹性指标

方案	在阈值范围内的时间比例	超出阈值范围的偏离平均值	恢复时间（步数）
三重化变体	**0.94**	0.73	4
去掉冗余机制的变体	0.87	0.78	5
去掉同质的变体	0.90	0.70	5
本文方案	**0.94**	**0.65**	**1**

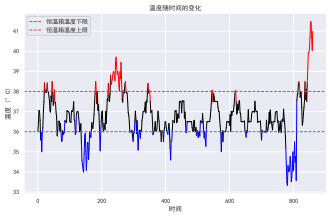
综上所述，本文方案在网络弹性和系统安全性方面均展现出了较强的优势，尤其在快速恢复、系统稳定性及高可用性方面表现突出。与其他变体相比，本文方案在故障恢复时间、稳定性控制以及系统可靠性方面均有显著改善，因此，在实际应用中具有更好的适用性，尤其适合于对系统可靠性和恢复能力有较高要求的场景。



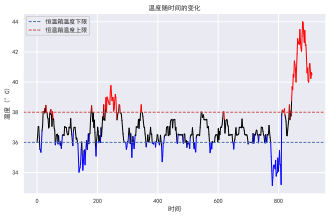
(a) 本文方案



(b) 三重化变体



(c) 去掉冗余机制的变体



(d) 同质变体

图 6 恒温箱温度随时间变化曲线

2.4 结论

本次仿真验证了基于动态异构冗余设计的分布式控制系统在温度控制场景下的性能表现。通过计算温度保持时间比例、偏离度和恢复时间，证明了该系统在稳定性和抗干扰能力方面的优越性。未来工作将进一步优化控制策略，以提升系统的响应速度和容错能力。

参 考 文 献

[1] BELL R. Introduction to iec 61508[C]//Acm international conference proceeding series: Vol. 162. Citeseer, 2006: 3-12.