

中图分类号: TP29

论文编号: 10006SY2041109

北京航空航天大学  
硕士学位论文

数字孪生平台研发及其  
数据和知识建模方法研究

作者姓名 蒋亮亮

学科专业 电子信息与科学技术

指导教师 胡远奇

培养学院 集成电路科学与工程学院



A Dissertation Submitted for the Degree of Master

**Candidate:** veypi jiang

**Supervisor:** yuanqi hu

School of Integrated Circuit Science and Engineering

Beihang University, Beijing, China



中图分类号：TP29

论文编号：10006SY2041109

## 硕 士 学 位 论 文

# 数字孪生平台研发及其 数据和知识建模方法研究

作者姓名 蒋亮亮

申请学位级别 硕士

指导教师姓名 胡远奇

职 称

学科专业 电子信息与科学技术

研究方向 数字孪生

学习时间自 2020 年 9 月 1 日

起至 2023 年 1 月 1 日止

论文提交日期 (年) 年 (月) 月 (日) 日

论文答辩日期 (年) 年 (月) 月 (日) 日

学位授予单位 北京航空航天大学

学位授予日期 (年) 年 (月) 月 (日) 日



## 关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写过的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：\_\_\_\_\_

日期：      年      月      日

## 学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：\_\_\_\_\_

日期：      年      月      日

指导教师签名：\_\_\_\_\_

日期：      年      月      日



## 摘 要

数字产品定义 (Digital product definition,DPD) 经历了从二维到三维的发展。业界提出了以产品设计为核心的数字仿真 (DMU)、虚拟样机等技术, 产品设计信息的定义和表达也越来越完善。近年来, 国内外制造业的经验表明, 3D 数字化定义的产品模式已经成熟, 其效益被反复验证。但是, 目前的数字化产品定义还存在描述不涉及制造、运行和维护阶段、产品定义和过程定义之间缺乏相关性、后续模拟基于“理想定义”对真实产品的指导有限等问题。为了解决上述问题, “数字孪生”技术应运而生, 被誉为有望改变未来航空制造“游戏规则”的顶级技术。

数字孪生技术目前发展方向或者难点存在在数据质量和精度, 可视性和交互性提升, 模型和知识分析存储, 数学分析和仿真技术, 模型快速构建技术, 多领域落地应用, 实时性应用落地等方向。本论文在以下三个方面进行研究, 提出性能高效、使用方便、场景构建迅速的通用数字孪生平台方案, 并尝试进行实践应用。

在数据质量和精度方面, 探索设计了自主分布式实时消息通信网络, 保证了在硬件网络内消息实时性的要求, 在全系统内保证消息的发送、分发、存储、响应效率。

在模型与知识分析存储方向, 探索设计了一套使用图谱或语义构建逻辑模型的体系, 及其方便用户去编辑场景内各个节点交互响应逻辑以及虚拟场景内仿真分析逻辑, 参数化设计模型及模型间交互逻辑, 并设计合适的模型存储结构, 极大方便了模型与知识在各端的存储和传输。

在实时性应用方向, 在前两个技术基础上, 探索了系统全生命周期使用的方案技术, 从历史数据分析到实时性场景内数据监控和逻辑响应再到未来部分数据推演仿真, 将单个孪生场景的历史、现在、未来数据统一整合, 极大拓展孪生技术应用方向范围。

在真实的实践场景中, 还会遇到很多复杂的现实问题, 如海量信息的建模、处理问题, 模型的结构化专有化问题等等。单一场景都有可能面临构建几千孪生体和处理每秒几十万量级数据, 如何用已有的知识去快速推演模型, 如何去关联、分析、响应海量异构数据, 仍是在目前在探索研究前两个技术方向时去尝试研究和解决的现实难点。

**关键词:** 数字孪生, 分布式消息网络, 知识模型, 数据推演

## **Abstract**

**Key words:** digital twins

# 目 录

|                     |    |
|---------------------|----|
| 第一章 绪论 .....        | 1  |
| 1.1 数字孪生概述 .....    | 1  |
| 1.2 研究背景及现状 .....   | 2  |
| 1.3 研究意义 .....      | 3  |
| 1.4 研究目标及方案 .....   | 5  |
| 1.4.1 研究目标 .....    | 5  |
| 1.4.2 研究方案 .....    | 5  |
| 1.4.3 主体平台 .....    | 5  |
| 1.4.4 分布式消息网络 ..... | 6  |
| 1.4.5 状态与事件机制 ..... | 6  |
| 1.4.6 规则模型 .....    | 7  |
| 第二章 系统设计 .....      | 9  |
| 2.1 系统设计原则 .....    | 9  |
| 2.1.1 高可用 .....     | 9  |
| 2.1.2 高性能 .....     | 9  |
| 2.1.3 可扩展性 .....    | 9  |
| 2.1.4 安全性 .....     | 9  |
| 2.1.5 单一职责原则 .....  | 9  |
| 2.2 整体结构设计 .....    | 9  |
| 2.3 功能设计 .....      | 11 |
| 2.4 数据库设计 .....     | 11 |
| 2.5 UI 设计 .....     | 12 |
| 第三章 分布式消息网络设计 ..... | 13 |
| 3.1 概述 .....        | 13 |
| 3.2 通讯机制 .....      | 13 |
| 3.3 通讯协议 .....      | 15 |
| 3.4 路由算法 .....      | 15 |
| 3.5 分布式选举协议 .....   | 16 |
| 3.6 硬实时通信 .....     | 17 |
| 3.7 终端通信 .....      | 17 |

|                           |           |
|---------------------------|-----------|
| 3.8 性能分析 .....            | 18        |
| <b>第四章 规则模型设计 .....</b>   | <b>21</b> |
| 4.1 基本模型 .....            | 21        |
| 4.2 程序接入机制 .....          | 21        |
| 4.3 代码解析机制 .....          | 21        |
| 4.4 公式解析算法 .....          | 22        |
| 4.5 代码规则案例-碰撞检测 .....     | 23        |
| 4.5.1 点与 AABB .....       | 24        |
| 4.5.2 AABB 与 AABB .....   | 24        |
| 4.6 公式规则案例-刚体运动力学模型 ..... | 24        |
| 4.6.1 空间位置 .....          | 25        |
| 4.6.2 空间姿态 .....          | 26        |
| <b>第五章 状态与事件模型 .....</b>  | <b>29</b> |
| 5.1 状态机制 .....            | 29        |
| 5.2 事件模型 .....            | 29        |
| 5.3 时序模型 .....            | 30        |
| <b>第六章 应用案例 .....</b>     | <b>33</b> |
| 6.1 规则模型-PID .....        | 33        |
| 6.2 自组织网络-机械臂 .....       | 33        |
| <b>结论 .....</b>           | <b>35</b> |
| <b>参考文献 .....</b>         | <b>37</b> |

# 插图目录

|     |                  |    |
|-----|------------------|----|
| 图 1 | 系统结构图 .....      | 6  |
| 图 2 | 平台主体结构 .....     | 10 |
| 图 3 | 通讯网络 .....       | 14 |
| 图 4 | 选举协议 .....       | 16 |
| 图 5 | 消息编码速率分布 .....   | 18 |
| 图 6 | 消息编码速率拟合曲线 ..... | 18 |
| 图 7 | 消息分发速率压测结果 ..... | 19 |
| 图 8 | 规则模型上传过程 .....   | 22 |



## 表格目录

|     |               |    |
|-----|---------------|----|
| 表 1 | 通讯帧 .....     | 15 |
| 表 2 | 消息编码 .....    | 19 |
| 表 3 | 节点状态转移表 ..... | 29 |



## 主要符号对照表

|                        |                                    |
|------------------------|------------------------------------|
| HPC                    | 高性能计算 (High Performance Computing) |
| cluster                | 集群                                 |
| SMP                    | 对称多处理                              |
| MBD                    | 基于模型的设计                            |
| API                    | 应用程序编程接口                           |
| Digital Twin           | 数字孪生                               |
| entity                 | 实体对象                               |
| physical entity        | 物理对象                               |
| virtual entity         | 虚拟对象                               |
| physical domain(space) | 物理域 (物理空间)                         |
| $L$                    | 角动量                                |
| $I$                    | 惯性张量                               |
| $\tau$                 | 力矩                                 |
| $\omega$               | 角速度                                |



# 第一章 绪论

## 1.1 数字孪生概述

数字产品定义 (Digital product definition,DPD) 经历了从二维到三维的发展。业界提出了以产品设计为核心的数字仿真 (DMU)、虚拟样机等技术, 产品设计信息的定义和表达也越来越完善。近年来, 国内外制造业的经验表明,3D 数字化定义的产品模式已经成熟, 其效益被反复验证。但是, 目前的数字化产品定义还存在描述不涉及制造、运行和维护阶段、产品定义和过程定义之间缺乏相关性、后续模拟基于”理想定义”对真实产品的指导有限等问题。为了解决上述问题,”数字孪生”技术应运而生。数字孪生技术被誉为有望改变未来制造”游戏规则”的顶级技术。这项技术使用数据馈送来映射物理实体, 并正在对工业的许多领域产生颠覆性的影响。德国信息技术和新媒体协会预测, 在制造业市场中, 数字孪生的价值是巨大的, 到 2025 年将超过 780 亿欧元。Gartner 已经连续两年 (2016、2017) 将数字孪生列为十大战略技术发展趋势之一。2017 年 11 月, 世界上最大的武器制造商洛克希德 马丁公司将数字孪生列为未来国防和航空航天工业的六大顶级技术。

数字孪生是指将物理对象、流程或服务数字化建模, 并利用这些模型来推断其行为和性能的技术。它是一种全面的数字化方法, 可以从产品生命周期的开始到结束对其进行全面的描述和分析, 实现可视化、可交互、可预测的物理世界虚拟化。在数字孪生中, 物理系统被视为网络, 由一系列特征点组成的对象集合, 这些特征点包括物理机制、虚拟特征和与人的交互关系。数字孪生技术可以映射物理世界中的任何物体, 例如发动机、电场、建筑物和城市等。此外, 数字孪生还可以用于映射流程, 并通过收集和分析数据来预测它们的行为和性能。数字孪生本质上是一种计算机程序, 它利用现实世界的数据来创建可以预测产品或流程行为和性能的虚拟模型。数字孪生技术可以通过整合物联网、人工智能和软件分析等技术, 来提高效率和精确度, 并在现代工程中推动创新和提高性能。随着机器学习和大数据等因素的进步, 数字孪生技术的应用范围将会越来越广泛, 为人们提供更加全面的数字化解决方案。

在真实的实践场景中, 数字孪生往往会遇到复杂的现实问题, 如海量信息的建模、处理问题, 模型的结构化专有化问题等等。单一场景即有可能需要构建几千孪生体并处理每秒几十万量级的数据, 如何根据已有知识构建模型, 如何描述模型特征和状态, 如何关联、分析、响应大量数据, 是本篇文章的主要研究内容。

## 1.2 研究背景及现状

数字孪生 (Digital Twin) 的概念最早出现在 2003 年, 由 Grieves 教授在密歇根大学的产品生命周期管理课程中提出。2010 年, ”Digital Twin” 一词在 NASA 的技术报告中被正式提出, 并被定义为”集成了多物理量、多尺度、多概率的系统或飞行器仿真过程”。2011 年起, 美国国防部将数字孪生的概念引入了航天器的健康维护等问题解决中。

NASA 将数字孪生定义为: 以飞机或系统为导向, 充分利用最佳物理模型、传感器和运行历史数据, 集成多学科和多尺度概率仿真过程, 映射对应物理飞机的状态。

美国空军对数字孪生的定义是基于 NASA 将”面向飞机或系统”扩展到”面向完成建模的系统”和”映射”扩展到”映射和预测”, 并给出了机身图形的孪生内涵示例 (包括几何模型、材料性能、飞行试验数据、制造/检验/维修数据、气动模型、有限元模型、损伤演化模型)。

庄存波等人对数字孪生的定义是产品数字孪生, 是指产品物理实体在信息空间中的工作状态和工作进度的全要素重构和数字映射。它是一种集成的多物理、多尺度、超逼真的动态概率仿真模型, 可用于仿真、诊断、预测和控制真实环境下产品物理实体的实现过程。

Grieves 等人对数字孪生的定义是一组虚拟信息结构, 对复杂产品从微观原子层次到宏观几何层次全面描述潜在生产或实际制造产品。构建一个数字孪生的最好结果是, 任何可以通过检查实际制造的东西, 产品获得的信息也可以从它的数字孪生获得。在研究基于数字孪生的复杂产品/复杂产品/系统行为分类系统问题预测时, 明确指出构建数字孪生的目的是基于仿真预测, 最大程度地消除复杂产品/系统”不可预测和不受欢迎的行为”, 以避免不可知的灾难性问题。以火箭发射为例, 在虚拟空间中, 模拟真实火箭的数字孪生, 即使失败了, 也可以用新的虚拟空间快速替换, 继续启动。

数字孪生是以数字化方式创建物理实体的虚拟实体, 借助历史数据、实时数据以及算法模型等, 模拟、验证、预测、控制物理实体全生命周期过程的技术手段, 建立在对一系列多维度的大规模的、实时的真实世界的的数据测量, 并辅以一定的物理和行为建模进行状态复现和预测的概率过程。

现阶段, 除了航空航天领域, 数字孪生领域还应用于电力、矿产采集、城市管理、工业制造等, 特别是在智能制造领域, 数字孪生被认为是一种实现物理世界、信息世界及人类意识世界交互的重要手段。

从数字孪生的多方陈述中可以看出其一部分与仿真类似, 仿真技术是用包含了确定性规律和完整机理的数字模型模拟物理世界的方法, 目的是依靠正确的模型和完整的信息、环境信息分析物理世界的特性和参数, 而数字孪生有仿真的内核, 但是多了实时性、

交互性, 其信息模型和物理模型可以相互关联相互影响, 其主要技术包括信息建模、信息采集同步、信息分析、智能决策等, 目前虽然已经取得了很大发展, 但是由于现在世界模型机理复杂、知识图谱建立困难以及有效数据采集分析困难等问题, 限制了其实践形势。

### 1.3 研究意义

数字孪生技术是近年来在工业领域和其他领域快速发展的一项技术, 它将物理对象、流程或服务的数字表示与现实世界相连结, 提供了一种在虚拟环境中创建、测试、优化和管理物理系统的新方法, 这对于许多行业的发展具有重要的意义, 在工业界得到了广泛的关注, 全球最具权威的 IT 研究与顾问咨询公司 GARTNER 连续两年将数字孪生列为当年十大战略科技发展趋势之一, 世界最大的武器生产商洛克希德马丁公司于 2017.11 将数字孪生列为未来国防和航天工业 6 大顶尖技术之首;2017 年 12 月 8 日中国科协智能制造学术联合体在世界智能制造大会上将数字孪生列为世界智能制造十大科技进展之一, 达索、西门子等公司已初步将数字孪生技术应用于产品设计、生产、维护过程中, NASA、美国空军研究实验室等将其运用在具体型号研发中, 并获得 2016 年美国国防制造技术奖。

数字孪生技术在众多行业得到及其醒目的关注, 并陆续将其作为未来发展技术, 在于它对生产全生命周期都具有重要意义。

首先, 数字孪生技术可以提高产品设计和生产过程的效率。通过数字孪生技术, 工程师可以在虚拟环境中设计、测试和验证产品, 从而减少了制造过程中的错误和重复。此外, 数字孪生技术还可以用于优化生产过程, 通过实时数据监测和分析来改善生产效率, 降低能源消耗, 减少废品率, 提高产品质量和生产效率。

其次, 数字孪生技术可以帮助企业更好地了解其产品和服务在市场上的表现。通过数字孪生技术, 企业可以将现实世界中的数据和虚拟模型相结合, 来模拟客户体验和产品信息。这有助于企业更好地了解客户需求和市场趋势, 从而更好地调整产品和服务, 提高客户满意度。

另外, 数字孪生技术还可以用于改进运营和维护。通过数字孪生技术, 企业可以利用虚拟环境中的数据和进行分析来监测和预测设备运行情况, 并进行远程监控和维护。这可以减少设备故障和停机时间, 降低维修成本, 提高设备利用率和生产效率。

数字孪生技术在提高效率、改进产品设计和生产过程、提高客户满意度、降低成本、提高生产效率、产品运营和维护等方面具有重要的意义。随着技术的不断发展, 数字孪生技术有望成为各行各业推动数字化转型和实现可持续发展的重要工具, 同时在实际生

产使用过程中也还面临很多复杂的现实问题。

**数据问题** 数据采集的尺度或计量单位的一致性，涉及物理数据、几何数据、时间数据等。如构建实物的三维模型的坐标与计量单位不一致，会导致不同模型之间无法融合，需要增加数据接口与编写数据翻译器；如工厂内的生产计划数据采集过程中，不同时间单位的生产计划数据会导致数字孪生模型出现数据读取错误。数据采集参数及格式的一致性。针对同一对象，多维虚拟模型采集的数据格式不一致、参数类型和数量不对等，同样会出现不同模型在数据融合时出现问题，不能进行交互。数据采集周期问题。如生产设备的数据产生一般以毫秒记，用采集的数据来驱动数字孪生模型时，往往需要放大时间尺度，否则模型的仿真运行压力过大会导致崩溃。除此之外，数据采集的稳定性和准确度、海量数据的存储和处理能力、通信接口协议不一、多源异构数据整合、数据歧义等均对数据提出了挑战。

**模型问题** 模型类别单一，在现有的数字孪生系统中，专有化程度较高，模型结构比较单一，虚拟世界基础依赖知识单一，在从分析预测阶段向自主控制的智能化分析探索，对现有基础模型知识库提出海量的需求。由于目前知识库的数据结构和模型没有统一的标准，多模型互操作难，数据语义、语法不统一，标准化的知识图谱体系尚需探索。在本文中将模型按层次分为几何模型、物理模型、规则模型，其中几何模型和物理模型适用于虚拟世界，其真实作用是方便人更直观的去观察系统变化，规则模型涉及实体、虚体、系统、人四方数据流通和逻辑判断，其模型建立上现在都是通过特定场景应用下硬编码实现，无法通用，更新和建立模型也十分复杂。

**实践问题及多系统融合** 数字孪生技术是一项综合技术，与数据采集处理、数字孪生与数字模型、数字孪生与 PLM 产品全生命周期、数字孪生与大数据分析、数字孪生与 CPS 信息物理系统、数字孪生与工业互联网等多技术融合，当前需要数字孪生基础理论及相关的技术融合突破，开展设备泛在接入、工业通信协议适配、异构系统集成、虚实融合等核心关键构件研发，突破多协议数据转换、海量异构数据汇聚、感知数据驱动、数字孪生精准映射等关键技术研究促进数字孪生应用。在多技术融合下，其商业模式不能得到充分保障，其实践重要性仍需关键突破应用，从单一环境仿真到大世界综合场景应用仍需底层技术发展和相关数据标准统一。数字孪生的多系统特性即反映在物理空间，也反映在虚拟空间，在数据、模型和交互各环节均有表达。数字孪生融合物理世界与数字世界，是一个多维系统的融合。首先面临的是物理世界的多系统挑战。据不完全统计，制造业现在的设

备数字化率约为 47%，局域联网率只有 40%，可接入公网的只有 20% 左右，底层 OT 跟 IT 的融合仍然是极其核心的基础性问题。

针对以上三个问题，本文在结合实践形式和数字孪生技术特点，分析了相关技术理论，分别提出了相应的改进方案，探索数字孪生技术在全产品生命周期的应用前景和能力。

## 1.4 研究目标及方案

### 1.4.1 研究目标

数字孪生作为一种实现物理实体向信息空间数字化模型映射的关键技术，通过充分利用布置在物理系统各部分的传感器，对物理实体进行数据分析与建模，形成多学科、多物理量、多时间尺度、多概率的仿真过程，将物理实体在不同真实场景中的全生命周期过程反映出来。但是在落地使用上面临较为复杂的现实问题，目前比较成功的案例都局限在特定场景特定数据下，尚未存在较为成熟的通用技术方案，针对上文总结的三个问题，分别提出相应的方案去解决或者部分解决现实面临的困境。

- 1、数据困境：编写设计一套自组织分布式消息网络，部分节点实现硬实时通信，整体实现所有节点自组织形成分布式通信网络。
- 2、模型困境：针对规则模型编写设计一套通用机制，物理节点、虚拟节点、服务节点可分别使用和设计不同规则，用户生成规则可以选择内置逻辑、函数、真值表、代码等实现。
- 3、实践困境：以分布式消息网络为基础，编写设计模型通用特征，在系统所有层次都有公开设计的通信接口，任何节点可以选择使用非内置节点，使用第三方或者自行编写的逻辑，提供现有主流的 PLC 通信协议，并提供将任何协议转内置通信协议的方案或物理节点。以上述方案为基础，构建数据推演模型，整合历史、现在、未来数据，扩充数字孪生平台应用范围、降低应用难度。

### 1.4.2 研究方案

本文开发的数字孪生平台主要涉及 4 大主体内容，包含主体平台、分布式消息网络、节点状态与事件机制，规则模型。

### 1.4.3 主体平台

图 1 所示，A 为主体平台，是数字孪生平台的主体，包含了基本的数据存储、数据分析、核心消息节点、用户交互、模型展示、知识存储等功能，其余所有一切功能都

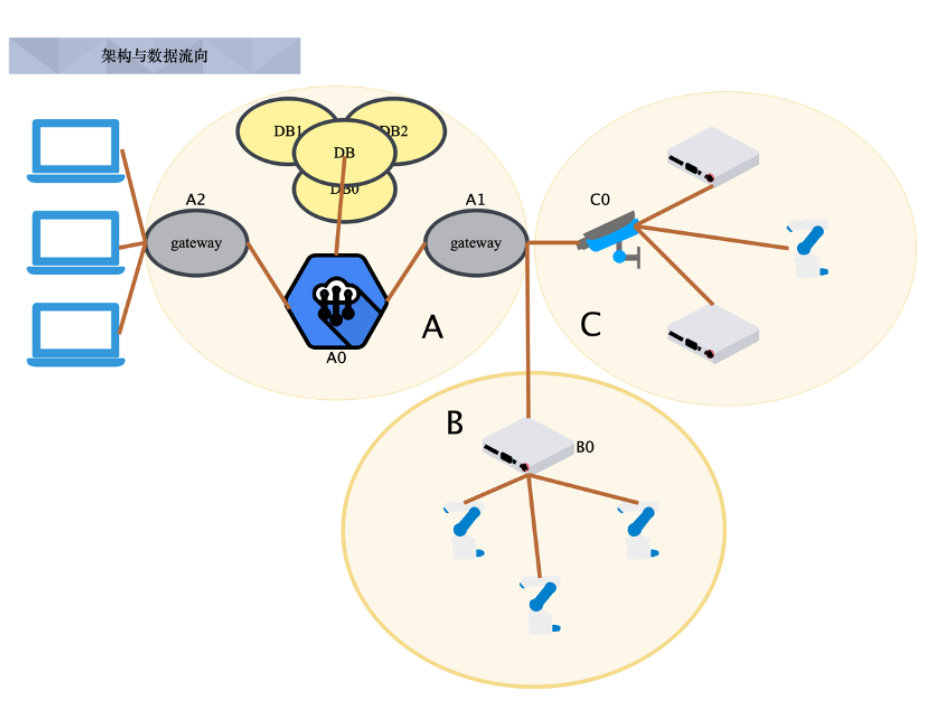


图 1 系统结构图

要借助主体平台得以显示和控制。用户通过 A 集群查看、分析和操作所有集群。

#### 1.4.4 分布式消息网络

图 1 所示, ABC 三集群之间, 集群内部, 用户与主体平台之间均使用的是该消息网络, 但是在不同的节点不同的集群, 根据节点需要, 该消息网络有不同的特征和性能要求, 比如在用户与主体平台之间, 实时性要求低, 对精细数据无要求, 但涉及大量重复数据给不同用户分发, 其消息网络注重分发性能设计。BC 物理集群, 涉及硬件操作, 对实时性要求极高, 要求存在硬中断和优先级, 所以其设计偏向硬实时消息网络。

#### 1.4.5 状态与事件机制

状态与事件机制为系统核心逻辑, 任意物理虚拟服务节点在本身, 主体平台以及途径的所有消息节点都有维护一套状态机制, 节点本事就是一个状态机, 其状态会在消息网络中广播最后到达主体平台, 事件机制为各节点交互的基础, 每个节点仅对自己负责, 优先级最高的是维护自身状态, 事件输入触发节点逻辑, 状态改变和节点输出皆为事件输出对外广播, 节点仅负责自己的状态和逻辑, 其输入输出和状态改变皆为一条事件, 其核心运行逻辑就是加入消息网络, 订阅需要的事件, 接收事件, 触发逻辑运算, 输出事件到消息网络, 其输出是否有用有消息网络判定, 节点主体只负责接收、运算及维护状态、输出。事件机制为主体平台和部分物理集群核心节点所独特管理的逻辑机制, 其

基础是订阅制，在订阅制基础上附上用户设计的其他规则逻辑。

#### 1.4.6 规则模型

规则模型是在以消息网络和事件机制为基础上搭建的逻辑判定机制，其涉及的规则运行主体、客体为系统内任意物理虚拟服务节点或者任意事件，其规则生成可以使用平台内置的逻辑，也可以由用户输入函数、真值表、伪代码片段或者接入程序实现。



## 第二章 系统设计

### 2.1 系统设计原则

#### 2.1.1 高可用

高可用的衡量指标通常为几个 9，它是衡量系统稳定工作的指标。在本文中主要指系统服务的稳定性，消息网络的稳定性。

#### 2.1.2 高性能

在本平台设计中，核心性能指标有消息网络的分发能力，硬实时消息的响应速度，主体平台对多种规则模型的处理速度，海量消息的存储能力，用户访问时模型的操作性能、仿真性能等。

#### 2.1.3 可扩展性

可扩展性是本平台设计中核心设计的一点，不单单指服务的可扩展性，比如服务节点可以水平扩增等，还指集群网络的开放程度，能对接现有消息协议，开放消息网络对接和规则模型设计。

#### 2.1.4 安全性

在保持消息网络的开发性上去保证整体平台、节点的安全性，在实践中，消息网络如何在开放协议的基础上保证通信安全。

#### 2.1.5 单一职责原则

任何模块保持职责和功能的最小化，专一化，模块之间不含功能交集。

### 2.2 整体结构设计

在设计数字孪生整体结构时，需要考虑用户使用、硬件通信、云服务特性去考虑设计，同时考虑平台的通用性，可扩展性，容灾恢复，安全性等等。

在整体上采用微服务架构。将系统按照业务功能模块拆分成若干个独立的微服务，每个微服务都能够独立部署和扩展。微服务架构可以提高系统的可伸缩性和可用性，并且可以降低系统的维护成本。

前端采用前后端分离的架构，使用现代化的前端框架 vue，通过 API 连接到后端服务，从而实现前后端分离，提高系统的可维护性和可伸缩性。

后端采用分布式架构，将不同的业务模块拆分成独立的微服务，通过 API 或消息队列进行通信，提高系统的可伸缩性和可用性。同时，采用缓存技术，如 Redis 等，提高系统的性能。

数据库采用分布式数据库架构，将不同的数据拆分到不同的数据库实例中，通过数据分片技术进行分布式存储，提高系统的可伸缩性和性能。同时，采用读写分离技术，将读操作和写操作分别路由到不同的数据库实例中，从而提高系统的性能。根据数据的特点，分别选用不同类型的数据，主体结构数据存入 mysql 中，时序数据存入 es 中，非参数化模型存入文本或 oss 中。

在安全设计中需要采用统一的非对称密钥认证机制，用户访问和节点通信皆使用私人密钥本地解密远程密文进行认证, 例如身份验证、访问控制、节点接入、数据加密等，保障系统的安全性。

如下图所示，为云平台主体结构，通用数字孪生平台的架构设计需要考虑多方面因素，通过采用现代化的架构技术和安全机制，实现高性能、可伸缩、可用和安全的系统。

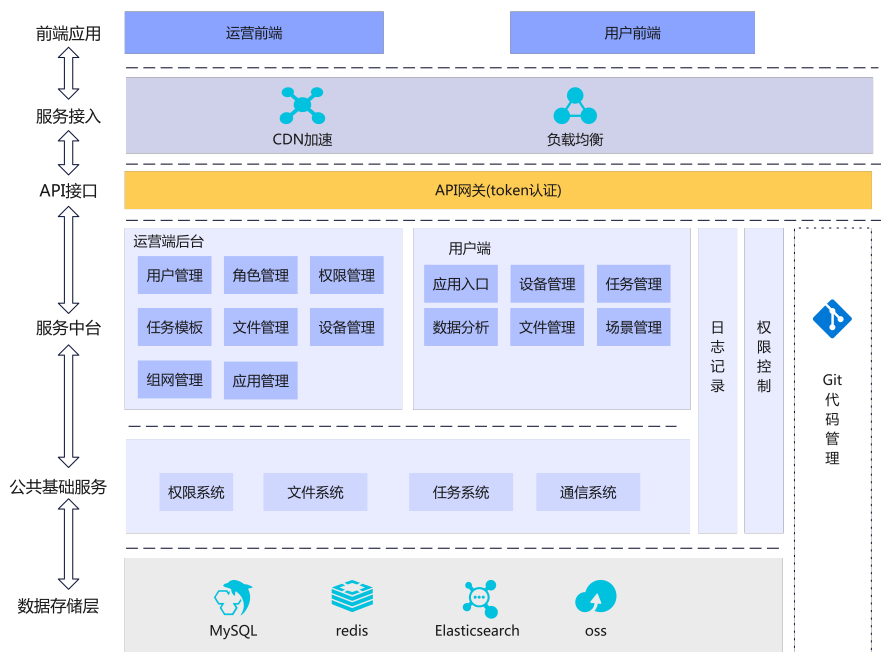


图 2 平台主体结构

## 2.3 功能设计

在设计该平台功能时，重点考虑其通用性和扩展性后，将平台功能点分为基础功能和拓展功能，拓展功能以本文提供的数据通信方案和规则模型方案为基础构建，由基础功能和额外编写的逻辑规则拓展而成，用户也可以此设计拓展功能。

- 1、数字孪生建模功能：提供用户友好的建模工具，支持各种模型类型和复杂性。用户可以通过可视化界面创建、编辑和删除模型。同时，平台应该提供模型验证、合法性检查、版本管理等功能。
- 2、规则编写：提供多种方式的规则输入体系，使得用户方便的去构建场景和节点交互逻辑和内置规则。
- 3、数据采集和集成功能：数字孪生平台需要支持各种数据源的集成，包括传感器、设备、网络、云服务等等，提供丰富的 API 和数据连接功能，以便数据源的快速接入和管理。
- 4、数据分析和可视化功能：数字孪生平台需要提供数据分析和可视化工具，以便用户对数字孪生系统中的数据进行分析 and 探索。平台提供可视化工具和分析功能，支持数据探索、数据可视化和可视化交互等等。
- 5、仿真和预测功能：数字孪生平台需要支持仿真和预测功能，以便用户可以基于数字孪生系统中的数据进行模拟和预测分析。平台应该提供各种仿真和预测工具，包括数据建模、算法开发和模型训练等等。
- 6、安全性和可扩展性：数字孪生平台需要具备高可靠性和可扩展性。平台应该提供安全保障机制，包括数据安全、身份认证和访问控制等等。同时，平台应该具备可扩展性，以便用户可以轻松地扩展数字孪生系统的功能和规模。
- 7、API 和开放平台：数字孪生平台需要提供 API 和开放平台，以便用户可以轻松地开发自己的应用程序和服务。平台应该提供完整的 API 文档和示例代码，同时支持第三方开发者的集成和扩展。

## 2.4 数据库设计

在设计该平台数据库时，需要考虑以下几个方面：

- 1、数据库的范围和目的：数字孪生平台通常需要处理大量的数据，包括实时数据、历史数据、静态数据等。因此，需要确定数据库的范围和目的，包括存储数据的类型、数据的来源和流向、数据的格式等。
- 2、数据库的结构和模式：为了支持数字孪生平台的多样化需求，数据库需要具有灵

活的结构和模式。可以采用关系型数据库或非关系型数据库，或者二者结合使用。同时，需要确定数据库的表结构、字段类型、主键、外键等。

- 3、数据库的安全性和可靠性：数字孪生平台通常涉及到重要的实时数据和设备信息，因此需要确保数据库的安全性和可靠性。可以采用多重备份、加密、权限控制等方式来保证数据的完整性和保密性。
- 4、数据库的性能和扩展性：数字孪生平台需要支持高效的数据处理和分析，因此需要考虑数据库的性能和扩展性。采用分布式数据库、内存数据库等技术来提高性能，并采用分区、分片等方式来实现扩展性。
- 5、数据库的数据采集和清洗：数字孪生平台需要从多个来源获取数据，并进行数据清洗和处理。因此需要设计适合的数据采集和清洗模块，确保数据的准确性和一致性。

综上所述，数字孪生平台的数据库设计需要考虑多个方面，包括数据库的范围和目的、结构和模式、安全性和可靠性、性能和扩展性以及数据采集和清洗等。在设计过程中需要综合考虑各个方面的需求，灵活应用不同的技术和方法，确保数字孪生平台的高效、安全、可靠运行。

## 2.5 UI 设计

## 第三章 分布式消息网络设计

### 3.1 概述

现代软件应用程序很少孤立存在，而现在通常的做法是依赖服务或使用远程实体提供的信息。在这种分布式架构中，集成是关键。近年来，消息传递是解决分布式性质挑战的参考解决方案，例如网络不可靠性，生产者和消费者的强烈耦合以及应用的异质性。由于强大的社区以及对标准和整合的共同努力，消息代理如今已成为许多项目和服务的传输层构建块，近年来出现了很多的消息服务，如 MQTT, RabbitMQ, RocketMQ, Kafka 等。

现代分布式消息网络是一种基于分布式系统和消息传递模式的网络架构，它支持高可用性、可扩展性和可靠性的通信。它通常由多个节点组成，这些节点可以是物理机器、虚拟机或容器。

消息网络的核心是消息队列，它是一种异步通信模式，可以在多个节点之间传递消息。消息队列通常包括生产者、消费者和代理（broker），生产者可以将消息发送到队列中，消费者可以从队列中接收消息进行处理，代理则是消息队列的中心节点，负责维护队列、路由消息和确保消息的可靠性传递。

现代分布式消息网络通常采用发布/订阅模式或点对点模式进行消息传递。在发布/订阅模式中，消息生产者发布消息到主题（topic），消费者订阅感兴趣的主题并接收相应的消息。在点对点模式中，消息生产者发送消息到队列，只有一个消费者能够接收和处理该消息。

这些消息框架现在已经发展的非常成熟，但是不太适合本文的使用场景，通常用于云服务之间的消息通信，单纯工控用的 PLC 通信又十分的僵硬，适合构建信息物理系统（CPS）使用，涉及到大量设备、服务、人和虚拟实体四方消息互动目前来看没有适合的通信方案，在这四方之间消息传递的需求和性能要求皆不同，所以尝试去设计一套新的消息网络去解决数字孪生下的数据困境。

### 3.2 通讯机制

下图为具体某一集群内的消息节点图，在同一子网下的 client 节点上线时会进行广播，发现内网节点后进行选举，产生一个子网内核心节点，其余节点连接该节点，核心节点在向上连接，最终链接到核心服务器集群，服务器集群也会选举产生一个核心节点，这个节点为整个网络的核心节点，用来维护节点状态并向下同步状态，整体网络中任意

节点下线，皆会触发重新选举或状态更新，核心节点因故障下线时，会导致当前层次子网重新选举，中断服务 1s，但是不影响其余层次网络集群功能，最终上线后会同步节点状态，实现整体网络的最终一致性。

对于任意节点来说，其上级节点其实与下级节点一致，在功能逻辑上存在上下级，但是在消息分发上不存在上级，连接的所有节点都是下级或者说同级，会维护一套消息表，记录消息标记和需要发往的目标节点。如节点 A 刚开始上线时该表为空，当 A 需要订阅消息时，就广播给周边的节点；B 收到记录下消息类型和目标 A 以及路径长度 1，在广播给除了 A 以外的临近节点，同时路径长度加 1；每个节点收到同步消息时根据表内数据，没有则添加，有则根据路径长度比较，短则替换目标，长则抛弃。这样，任意节点订阅消息时将会在整个消息网络广播其订阅记录，并在每个节点记录的是最短传播路径。当节点收到消息时，根据消息表内记录，存在标记则转发，不存在则抛弃。

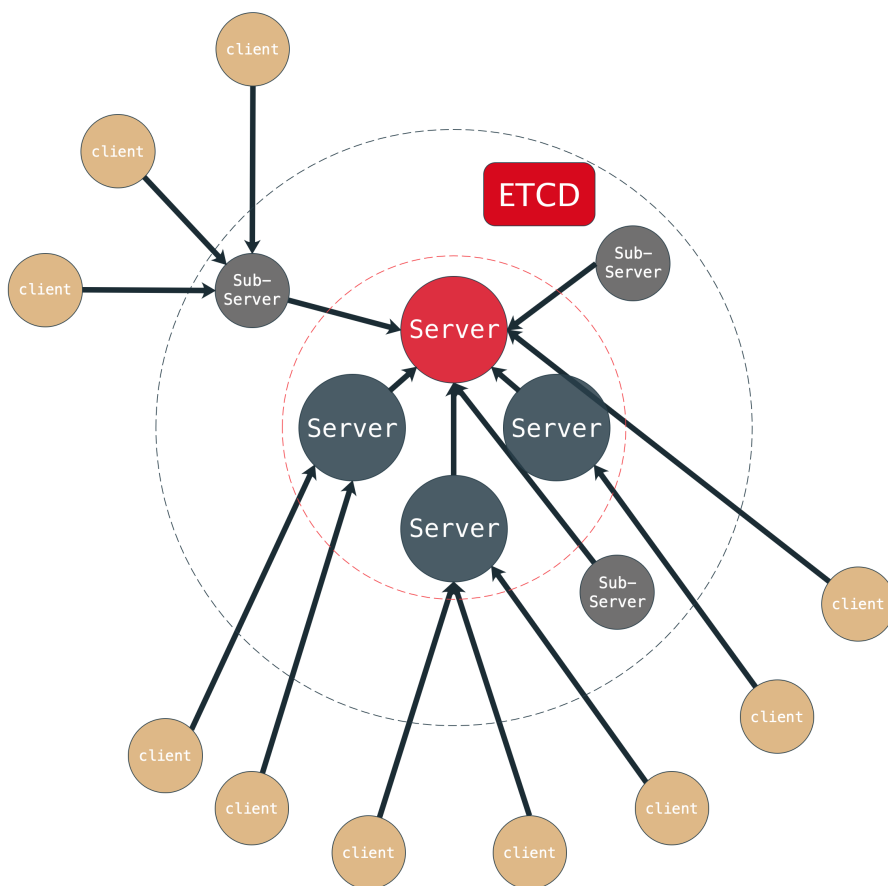


图3 通讯网络

表 1 通讯帧

|        |     |        |
|--------|-----|--------|
| prefix |     | count  |
| typ    | tag | source |
| target |     | data   |
| data   |     |        |
| ....   |     |        |
| data   |     |        |

### 3.3 通讯协议

- prefix(8bit): 帧前缀，用于判断消息帧的开始
- count(8bit): 消息长度
- typ(4bit): 消息类型
- tag(12bit): 消息标记
- source(8bit): 消息来源节点
- target(8bit): 消息目标节点
- payload: 搭载的数据

### 3.4 路由算法

在订阅消息时存在父子订阅需求，对于很多节点来说可能需要订阅的是某个具体的消息，对于某些规则类节点或者虚拟节点来说，订阅的是某个类别的消息，所以设计树状订阅机制，其订阅子树仅会收到子树消息，订阅父节点则会收到下属所有子树消息，这样在进行消息分发时不能简单的通过判断标记想到与否转发，所以采用了 rie 树算法加速消息分发。

rie 树又被称为前缀树、字典树是一种用于快速检索的多叉树结构。字典树把字符串看成字符序列，根据字符串中字符序列的先后顺序构造从上到下的树结构，树结构中的每一条边都对应着一个字符。字典树上存储的字符串被视为从根节点到某个节点之间的一条路径，并在终点节点上做个标记”该节点对应词语的结尾”，正因为有终点节点的存在，字典树不仅可以实现简单的存储字符串，还可以实现字符串的映射，只需要将相对应的值悬挂在终点节点上即可。

Trie 的核心思想是空间换时间，有如下基本性质：

- - 根节点不包含字符，除根节点外每一个节点都只包含一个字符
- - 从根节点到某一节点，路径上经过的字符连接起来，为该节点对应的字符串

- - 每个节点的所有子节点包含的字符都不相同

字典树能够利用字符串中的公共前缀，这样可能会节省内存，利用字符串的公共前缀可以减少查询字符串的时间，能够最大限度的减少无谓的字符串比较，同时在查询的过程中不需要预知待查询字符串的长度，沿着字典树的边进行匹配，查询效率比较高。

### 3.5 分布式选举协议

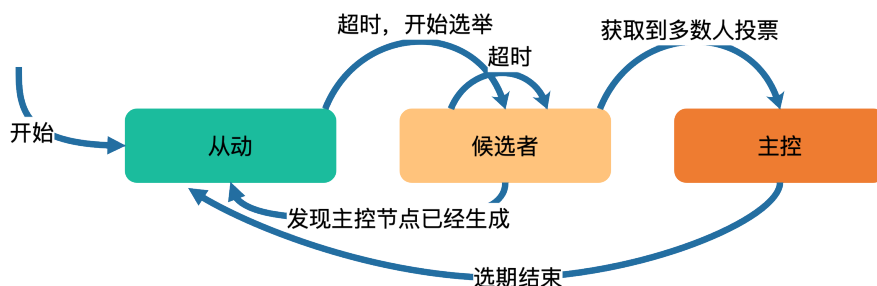


图4 选举协议

每个节点上线时如果有现有的消息网络会直接连入现有的消息网络，当大量节点同时上线或者子网间关键节点下线时会触发节点选举，选举产生一位核心节点代表他们和其他集群进行消息通信，并在这些核心节点中再次选举产生一个核心节点，这样所有的节点会自动的根据网络环境分层出消息层，并在消息传递之间保持最小路径在分布式选举协议中，一个节点任一时刻处于以下三个状态之一：

- 从动
- 候选者
- 主控

如上图所示，可以看出所有节点启动时都是从动状态；在一段时间内如果没有收到来自主控的心跳，从从动切换到候选者，发起选举；如果收到大多数的同意票（含自己的一票）则切换到主控状态；如果发现其他节点比自己更新，则主动切换到从动。

总之，系统中最多只有一个主控节点，如果在一段时间里发现没有主控，则大家通过选举-投票选出主控。主控会不停的给从动发心跳消息，表明自己的存活状态。如果主控节点故障，那么从动节点会转换成候选状态，重新选出主控。

### 3.6 硬实时通信

在工业控制领域实时 (Real Time) 是一个核心要求。实时系统是指计算的正确性不仅依赖于逻辑的正确性而且依赖于产生结果的时间, 如果系统的时间限制不能得到满足, 系统将会产生故障。在工业领域这种故障可能造成灾难性的结果。

在 CPU 资源调度方面, OS 主要提供一个多任务 (multitasking) 的运行环境, 以方便应用的开发。在开发某个应用时首先把工作拆解成多个任务 (Task/Thread), 每个任务都可以简化成一个简单的无限循环:

```
1 void MyTask (void)
2 {
3     while (1) {
4         Wait for an event to occur;
5         Perform task operation;
6     }
7 }
```

代码 3.1 基本调度过程

如上面代码所示, 任务 (Task) 都是等待 event, 然后处理事务。任何一个任务得以运行, 都是因为它收到了一个 Event, 这个 Event 可能是一个中断、也可能是超时到期、还有可能是其他任务发出的 IPC 信号, 继续追查发出 IPC 信号的任务最后的源头 Event 肯定是一个外部设备硬件中断或者是内部的 Timer 中断。中断引起了 Event 传递, 形成了逐个运行多个任务的链条 (Chain)。一个系统内部会存在很多条这种链条。

对实时 (Real Time) 系统来说, 不仅仅要求 OS 能提供多任务环境, 更要求任务能在极短的时间之内响应外部的中断事件。

对于终端节点来说硬实时通讯较好实现, 利用时钟定时中断即可强制切换到高级别消息发送, 但是终端节点一般作为信息发送和执行节点, 不具备逻辑处理功能, 在整体通讯延迟受最慢一级也就是逻辑处理层一般也是主机节点影响最大, 为处理此问题, 对消息进行分级, 高优先级消息会先触发中断响应, 并切换到执行状态, 且不可被抢占。

### 3.7 终端通信

在上述协议下可以实现在场景内部的毫秒级实时通信需要, 对于已有的不同终端, 根据通信协议和线材不同可连入已有的物理节点进行消息转发, 同时目前已开发了 c/rust 通信依赖库可供开发使用, 直接在设备内嵌入通信逻辑。用户可以选择是外接设备还是内嵌代码接入到整个数字孪生平台, 该物联终端接口可同样应用于自定义仿真算法和逻

辑算法节点通信接入。

### 3.8 性能分析

以下性能测试皆使用 2.4 GHz 八核 Intel Core i9 作为测试芯片，并进行相关内核性能调优后得到的测试结果。

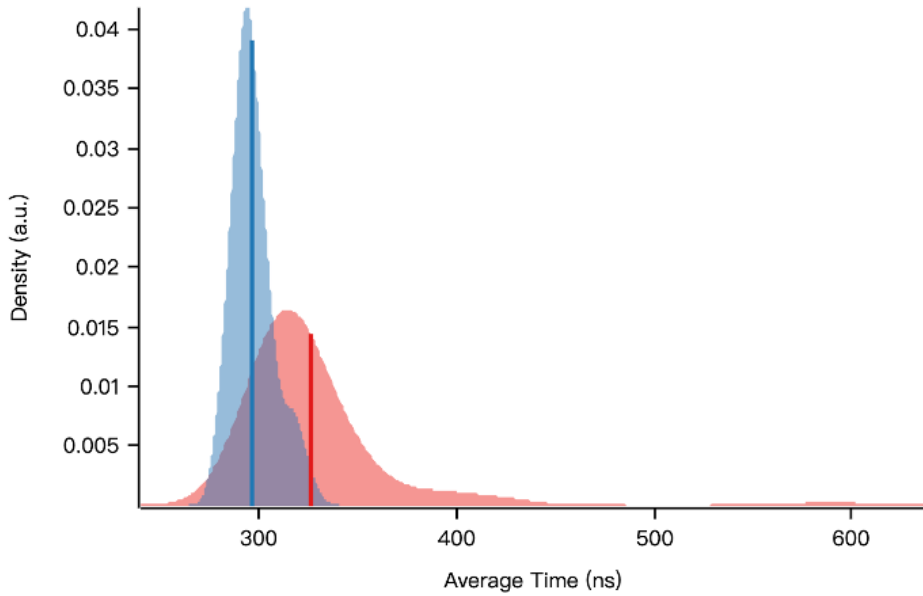


图 5 消息编码速率分布

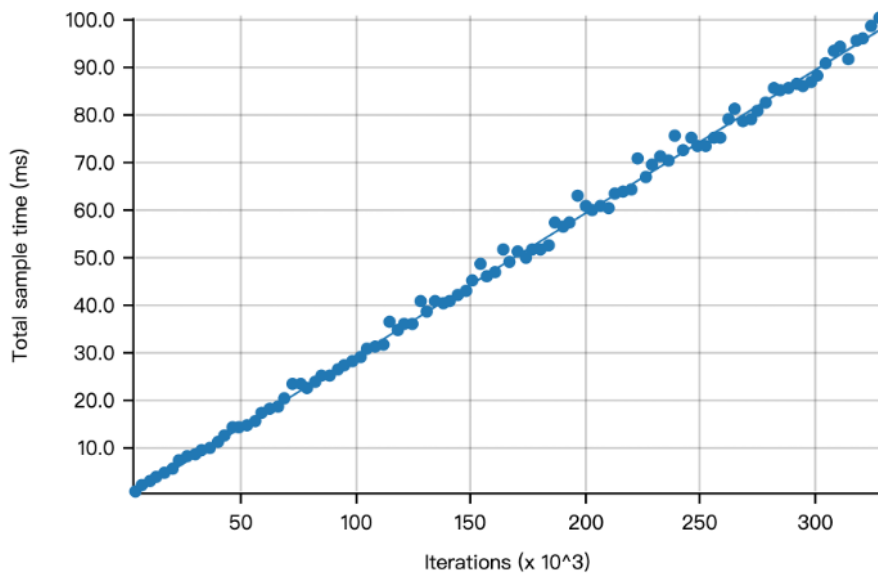


图 6 消息编码速率拟合曲线

由图 4 和表格可知，消息分发网络中消息编码时间在 300ns 左右，设备负载对消息

表 2 消息编码

| 编码时间   | 下限        | 估值        | 上限        |
|--------|-----------|-----------|-----------|
| 回归斜率   | 296.52 ns | 298.26 ns | 300.04 ns |
| 拟合度    | 0.9228055 | 0.9263008 | 0.9225949 |
| 均值     | 295.37 ns | 297.35 ns | 299.41ns  |
| 样本标准偏差 | 8.6420 ns | 10.325 ns | 11.813 ns |
| 中值     | 293.45 ns | 295.80 ns | 297.40 ns |
| 绝对中位差  | 6.1489 ns | 8.4741 ns | 11.089 ns |

编码速率分布有较大影响. 针对消息分发速率, 使用 wrk 工具对系统进行压力测试, 得到图 5

```
→ ~ wrk -t8 -c1000 -d15s http://127.0.0.1:4000/api/topic/derive
Running 15s test @ http://127.0.0.1:4000/api/topic/derive
8 threads and 1000 connections
Thread Stats   Avg      Stdev     Max    +/-  Stdev
Latency       6.33ms    1.83ms   59.51ms  98.16%
Req/Sec       19.82k    2.76k    22.24k   97.25%
2366694 requests in 15.01s, 291.16MB read
Socket errors: connect 0, read 3291, write 0, timeout 0
Requests/sec: 157651.06
Transfer/sec:   19.39MB
```

图 7 消息分发速率压测结果

由图 6 可以看出, 消息分发在未绑定任何响应函数的情况下单机可以达到每秒 157651 条, 在未读取数据库情况下消息分发延迟 6.33ms, 以上数据目前满足数字孪生平台要求。



## 第四章 规则模型设计

### 4.1 基本模型

在建立数字孪生虚拟世界过程中,会有大大小小的规则、算法等规则纳入其中,任何规则算法归纳来讲都是描述任意虚拟或实体对象相关作用关系。本文归纳了这些规则、算法的共同特征,建立了一个一般性的规则模型,该模型是一个动态模型,其规则求解过程可根据特点纳入平台内运算或使用独立进程进行运算求解。基本模型需要先建立三个外部特征,触发域、输入域、输出域。触发域主要用于设定触发求解过程的条件。输入域是指定该模型需要哪些虚拟/实体对象的什么参数。输出域是指定该模型输入影响范围和参数。对于不同的规则或者说规则,根据其作用域和功能特点,将其分为现实类规则、虚拟类规则、仿真类规则

**现实类规则** 该类规则是具体场景下最底层规则,其触发域和输入域仅能为实体对象数据,虚拟场景运行与否不影响该类规则触发和生效作用。

**仿真类规则** 该类规则为虚拟环境中与实体对象对应的虚拟对象的数据的关联规则,其输出域可以设定于虚拟世界,也可作用于现实世界。

**虚拟类规则** 该类规则其输出域仅能作用于虚拟世界。

如图 8 所示,基本模型内部求解过程可以根据类型和功能特点选择提交公式或算法代码,或直接运行独立的求解程序。本章后面以构建碰撞检测和刚体运动力学规则为例讲解代码和公式确定规则模型。

### 4.2 程序接入机制

如图 8 所示,用户仅需要申请访问密钥,调用本平台开发的依赖库即刻纳入到通信网络中作为一个计算节点执行计算任务,与内部的计算节点无任何差别,唯一要考虑的是跨地区网络后消息实时性无法满足要求,不适合去做实时性分析人物。目前支持的依赖库 python/go/c/rust。

### 4.3 代码解析机制

如图 8 流程所示,代码在用户上传完毕后会根据需要在平台上生成运行环境,并根据需要提供一定的运算资源和数据背景,提供更高阶的数据查询和接入方法。目前代码解析部分采取现成的语法,根据语言特性和场景需要,目前仅支持 js 和 python,尽管采用了一定的沙盒机制去运行,但是仍然面临很大的安全问题,所以现在的代码机制仅供

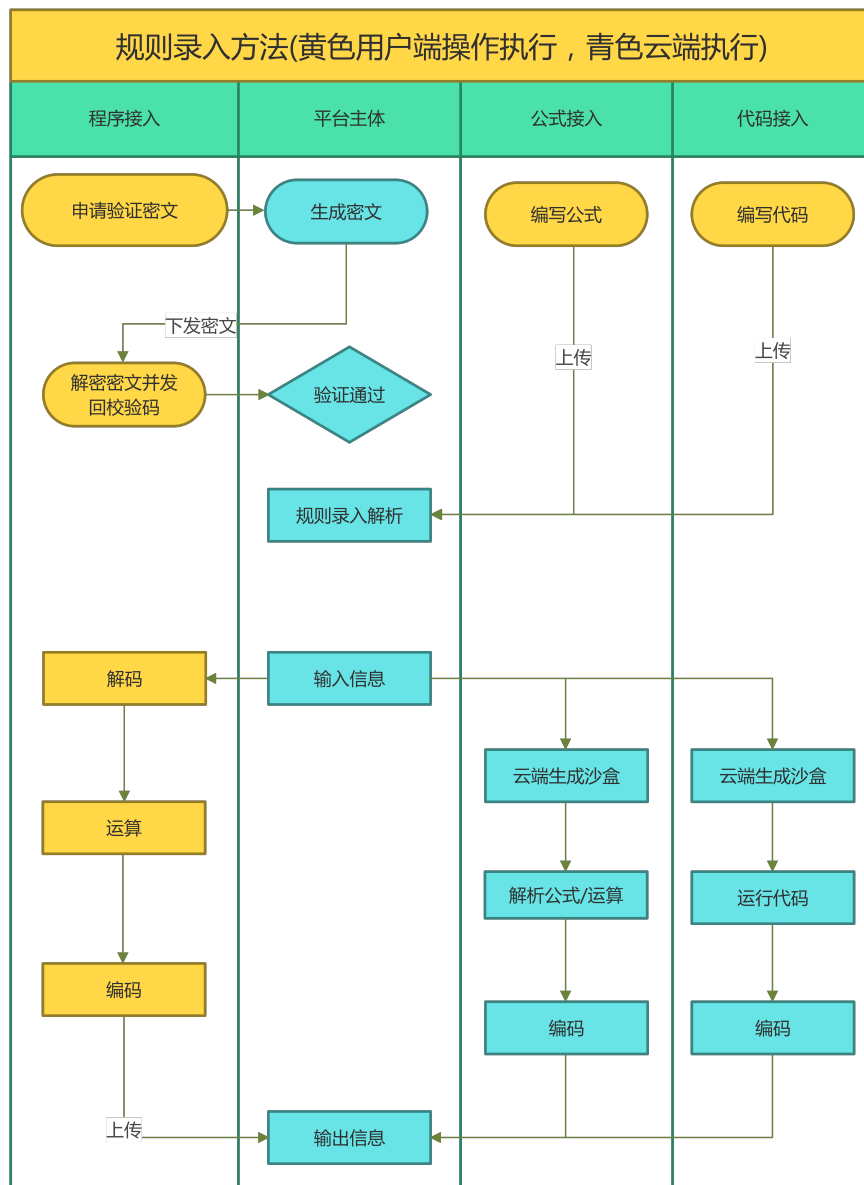


图 8 规则模型上传过程

内部使用，后续根据需要开放出自定义代码或者代码检查。

#### 4.4 公式解析算法

在构建规则模型其中很重要的一步是支持公式输入，对于一条规则，我们可以提供节点数据并转换成合适的格式，比如对于某一理想形状体，我给予 100N 的力 10s，如何通过规则模型解析公式获得该物体 10s 内的运动状态，首先我们需要去解析公式，确定输入变量数量和格式，在确定输出变量数量和格式，根据公式以一定的频率去输出结果。在公式撰写规则上直接采用先用的 latex 语法去语义化编写公式，采用 AST 算法去

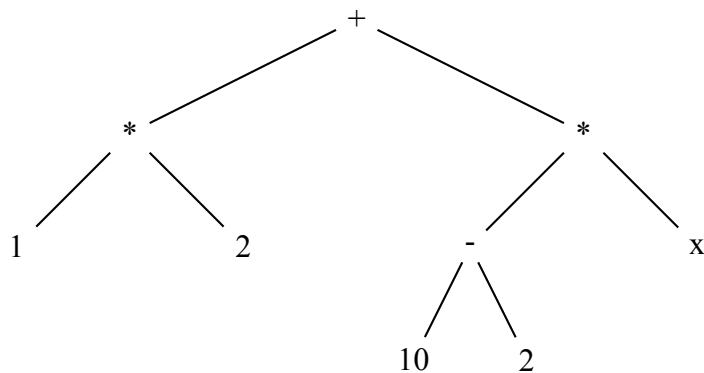
解析公式结构. 以该公式为例:

$$y = 1 * 2 + (10 - 2) * x$$

将右侧表达式转换为 RPN 表达式

$$1\ 2\ *\ 10\ 2\ -\ *\ x\ +$$

根据 RPN 表达式构建出的 AST 语法树为



以 AST 树构建公式算法, 输入目前支持浮点数及矩阵格式, 支持常用函数表达, 整体具体公式需要去做输入输出数据格式转换。

#### 4.5 代码规则案例-碰撞检测

碰撞检测目前定义为虚拟类规则, 暂时其输出域数据无法作用于现实世界, 主要用于虚拟世界用户交互检测。为了简化物体之间的碰撞检测运算, 通常会对物体创建一个规则的几何外形将其包围。在本系统中, 碰撞检测中将物体分为三种检测模型, 点、AABB、球体。其中, AABB (axis-aligned bounding box) 包围盒被称为轴对齐包围盒。

轴对齐包围盒是判断两个物体是否重叠的最快算法, 物体被包裹在一个非旋转的 (因此轴对齐的) 盒中, 并检查这些盒在三维坐标空间中的位置, 以确定它们是否重叠。

由于性能原因, 轴对齐是有一些约束的。两个非旋转的盒子之间是否重叠可以通过逻辑比较进行检查, 而旋转的盒子则需要三角运算, 这会导致性能下降。如果你有旋转的物体, 可以通过修改边框的尺寸, 这样盒子仍可以包裹物体, 或者选择使用另一种边界几何类型, 比如球体 (球体旋转, 形状不会变)。

在该例中, 根据不同需要适合使用代码片段去定义该规则。

### 4.5.1 点与 AABB

如果检测到一个点是否在 AABB 内部就非常简单了—我们只需要检查这个点的坐标是否在 AABB 内;分别考虑到每种坐标轴。如果假设  $P_x, P_y$  和  $P_z$  是点的坐标,  $B_{minX} - B_{maxX}$ ,  $B_{minY} - B_{maxY}$ , 和  $B_{minZ} B_{maxZ}$  是 AABB 的每一个坐标轴的范围, 我们可以使用以下代码计算两者之间的碰撞是否发生:

```

1 function isPointInsideAABB(point, box) {
2   return (point.x >= box.minX && point.x <= box.maxX) &&
3         (point.y >= box.minY && point.y <= box.maxY) &&
4         (point.z >= box.minY && point.z <= box.maxZ);
5 }
```

### 4.5.2 AABB 与 AABB

检查一个 AABB 是否和另一个 AABB 相交类似于检测两个点一样。我们只需要基于每一条坐标轴并利用盒子的边缘去检测。

```

1 function intersect (a, b) {
2   return (a.minX <= b.maxX && a.maxX >= b.minX) &&
3         (a.minY <= b.maxY && a.maxY >= b.minY) &&
4         (a.minZ <= b.maxZ && a.maxZ >= b.minZ);
5 }
```

## 4.6 公式规则案例-刚体运动力学模型

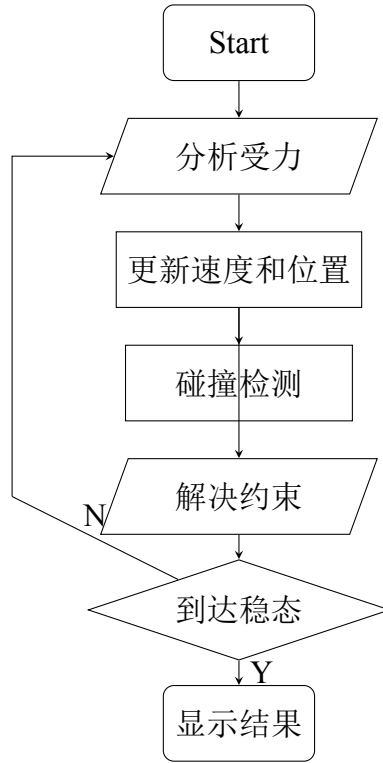
在虚拟世界中为关联现实设备运动状态, 需要根据已有的传感器数据, 如加速度、里程信息等估计实体对象运动姿态, 或者由虚拟对象指导影响实体对象运动, 两者相互作用皆需要实现基本的刚体运动。刚体运动的典型计算特征可用于验证公式规则模型, 其评估效率和验证性能直观有效。

刚体的运动主要基于牛顿三大定律来模拟:

- 1. 惯性物体在不受力时, 总是保持速度不变.
- 2. 力, 质量, 加速度力在物体上产生加速度, 满足  $F = ma$ .
- 3. 力的作用是相互的

基于牛顿三大定律, 在计算机中来模拟刚体的运动流程如此:

对于每个物体, 使用循环的方式来模拟:



最后将所有运动分解为粒子运动和旋转运动, 每秒以一定频率分别计算其空间位置和空间姿态, 最终合成为完整运动分析过程。

#### 4.6.1 空间位置

先不考虑物体的形状和旋转, 把物体当成理想粒子来对待, 根据牛顿定律来循环计算物体的速度和位置:

$$dt = t_{i+1} - t_i$$

$$v(t_{i+1}) = v(t_i) + \left(\frac{f(t_i)}{m}\right)dt$$

$$p(t_{i+1}) = p(t_i) + v(t_{i+1})dt$$

将其泰勒展开

$$p(t_{i+1}) = p(t_i) + p'(t_i)dt + p''(t_i)\frac{dt^2}{2!} + p'''(t_i)\frac{dt^3}{3!} + \dots$$

对于浏览器按像素点实时位置渲染精度而言, 3 阶泰勒展开已符合精度要求。

因此对于理想点的运动规则模拟可以提交公式

$$1 \quad p(t_{i+1}) = p(t_i) + \dot{p}(t_i)dt + \frac{\ddot{p}(t_i)}{2!}dt^2 + \frac{\dddot{p}(t_i)}{3!}dt^3$$

#### 4.6.2 空间姿态

要计算物体的空间姿态，即物体绕某一定点在三自由度上的旋转量，可以用四元数  $q$  来表示刚体在极短时间内的旋转量，

惯性张量

在三维空间中任取一点  $Q$  及一个直角坐标系  $Q_{xyz}$ ，可以得到物体的惯性张量：

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yz} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (4.1)$$

对角元素  $I_{xx}, I_{yy}, I_{zz}$  是物体分别相对于  $x, y, z$  轴的转动惯量。

$$I_{xx} = \int (r_y^2 + r_z^2)dm$$

$$I_{yy} = \int (r_x^2 + r_z^2)dm$$

$$I_{zz} = \int (r_x^2 + r_y^2)dm$$

计算惯量积

$$I_{xy} = I_{yx} = - \int (r_x r_y)dm$$

$$I_{xz} = I_{zx} = - \int (r_x r_z)dm$$

$$I_{yz} = I_{zy} = - \int (r_z r_y)dm$$

根据惯性张量计算力矩  $\tau$ ，角将速度  $d\omega$ ，角速度  $\omega$ 。

$$\tau = \frac{dL}{dt} = I \frac{d\omega}{dt}$$

$$d\omega = I^{-1} \tau dt$$

$$\omega(t_{i+1}) = \omega(t_i) + (I^{-1}\tau(t_i))dt$$

用四元数  $q$  来表示刚体的旋转状态，刚体在  $dt$  时间内沿着  $\omega$  旋转了  $|\omega dt|$  的角度，得到

$$q(t_{i+1}) = q(t_i) * [\cos \frac{|\omega dt|}{2}, \sin \frac{|\omega dt|}{2} \frac{\omega}{|\omega|}] \quad (4.2)$$

对于获取刚体旋转状态的规则特征可以提交为

$$\begin{array}{l} 1 \\ 2 \\ 3 \end{array} \left\{ \begin{array}{l} q(t_{i+1}) = q(t_i) * [\cos\{\frac{|\omega dt|}{2}\}, \sin\{\frac{|\omega dt|}{2}\} \frac{\omega}{|\omega|}] \end{array} \right.$$



## 第五章 状态与事件模型

### 5.1 状态机制

在所有的节点集群中，将节点分为三大类：

**服务节点**：主要为系统服务节点，运行平台核心功能或一部分核心规则

**物理节点**：一个接入到消息网络的真实物理设备，其响应会在物理世界生效

**虚拟节点**：主要为规则节点或者虚体逻辑节点，代表一个虚拟物品或者一段规则。

每个节点在接入消息网络时会自动同步输入输出接口，为方便节点运算管理和展示，对节点库内置有限状态机逻辑，可在算法内直接调用和修改。除了在节点算法内自行配置状态机外，还可以在用户端对节点状态转移表进行直接控制，如下表所示根据不同的输入编辑状态转换逻辑

**表 3 节点状态转移表**

|      | 状态 A | 状态 B | 状态 C |
|------|------|------|------|
| 事件 A | .... | ...  | ...  |
| 事件 B | .... | ...  | ...  |
| 事件 C | .... | 状态 C | ...  |
| 事件 D | .... | ...  | ...  |

### 5.2 事件模型

有泛含某种特定逻辑的单条或者多条消息在本平台归纳为事件，事件是在整个平台流转的最小价值信息，也是消息网络订阅主题的发布内容，每个主题仅会发布同种事件，事件具有特定的数据结构，不一定与节点输入输出数据结构相符，所以事件在到达和传出节点时会做一次结构转换。这样特定主体的事件就可以在整个消息网络和节点网络传播，并且即时生效和作用。事件也是数据存储的单位，每个事件都会作为一条消息存储在数据库中，后续所有基于数据的数据分析实质都是在这些时序事件中发现规则。其数据格式设计为：

```

1 {
2   "distinct_id": "2b0a6f51a3cd6775",
3   "time": 1434556935000,
4   "type": "track",

```

```

5  "anonymous_id": "2b0a6f51a3cd6775",
6  "event": "",
7  "properties": {
8      "ip" : "180.79.35.65",
9      "data": {
10         "demo1": test_data,
11         "demo2": .....,
12         .....
13     }
14 }
15 }

```

### 5.3 时序模型

时序模型是本平台最为核心的一个基础功能和分析手段。在已经运行的集群网络，会收集所有的历史事件，这些历史数据是所有数据分析的基础。整个集群会包含三个状态，过去，现在，未来。现在状态就是物理集群目前的状态，会按照设定好的规则和节点逻辑运行。过去和未来状态皆是模拟场景，发生在用户端，用户在客户端可以自由的调整时间进度条，往前拉，虚拟场景会自动重新模拟当时的事件数据并按照当时的逻辑规则运行，往后拉，会根据历史的事件数据进行拟合，去预测未来的集群状态和数据。相当于在虚拟世界可以自由的调整时间线，平台会根据时间戳自动重播历史事件消息或者预测未来数据。

本文采用时序分解法去预测未来数据，一个时间信号主要有四部分趋势影响叠加合成：

**长期趋势 (Secular trend,T)** 指在较长时期内持续发展变化的一种趋向或状态。

**季节变动 (Seasonal Variation,S)** 由于季节的变化引起的规则变动

**循环波动 (Cyclical Variation,C)** 指以若干年为期限，不具严格规则的周期性连续变动

**不规则波动 (Irregular Variation,I)** 指由于众多偶然因素对时间序列造成的影响

对于不同的信号，其特征不同，预测算法参数也应该有偏重，具体的预测算法参数和模型可以由用户配置，也可使用默认配置。

预测模型可采用：

**加法模型**  $Y_t = T_t + S_t + C_t + I_t$

乘法模型  $Y_t = T_t * S_t * C_t * I_t$

加乘模型  $Y_t = T_t * S_t * C_t + I_t$

对于四种因素预测，可采用最典型的移动平均法，也可以如第四章所示自行提交预测函数进行预测



## 第六章 应用案例

### 6.1 规则模型-PID

### 6.2 自组织网络-机械臂



## 结论



## 参考文献

- [1] GLAESSGEN E, STARGEL D. The digital twin paradigm for future nasa and u.s. air force vehicles[C]//Aiaa/asme/asce/ahs/asc Structures, Structural Dynamics ; Materials Conference Aiaa/asme/ahs Adaptive Structures Conference Aiaa. [S.l.: s.n.], 2012.
- [2] TUEGEL E J, INGRAFFEA A R, EASON T G, et al. Reengineering aircraft structural life prediction using a digital twin[J]. International Journal of Aerospace Engineering, 2011, 2011(1687-5966).
- [3] GRIEVES M, VICKERS J. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems[J]. Springer International Publishing, 2017.
- [4] LEE J. Integration of digital twin and deep learning in cyber-physical systems: Towards smart manufacturing[J]. 2020, 38(8):901-910.
- [5] UHLEMANN H J, LEHMANN C, STEINHILPER R. The digital twin: Realizing the cyber-physical production system for industry 4.0[J]. Procedia CIRP, 2017, 61:335-340.
- [6] ALAM K M, SADDIK A E. C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems[J]. IEEE Access, 2017, 5:2050-2062.
- [7] RAMTILAK A, JOSEPH A, SIVAKUMAR G, et al. Digital twin spark ignition for improved fuel economy and emissions on four stroke engines[C]//SIAT 2005. [S.l.: s.n.], 2005.
- [8] CERRONE A, HOCHHALTER J, HEBER G, et al. On the effects of modeling as-manufactured geometry: Toward digital twin[J]. International Journal of Aerospace Engineering, 2014, 2014.
- [9] QI Q, TAO F. Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison[J]. IEEE Access, 2018:3585-3593.
- [10] ANWER, NABIL, SCHLEICH, et al. Shaping the digital twin for design and production engineering[J]. CIRP Annals, 2017.
- [11] 陶飞, 张萌, 程江峰, 等. Digital twin workshop:a new paradigm for future workshop[J]. Computer Integrated Manufacturing Systems, 2017, 23(第 1 期).
- [12] BANERJEE A, DALAL R, MITTAL S, et al. Generating digital twin models using knowledge graphs for industrial production lines[C]//9th ACM Web Science Conference, Industrial Knowledge Graphs. [S.l.: s.n.], 2017.
- [13] WAGNER, C, GROTHOFF, et al. The role of the industry 4.0 asset administration shell

- and the digital twin during the life cycle of a plant[J]. IEEE Int C Emerg, 2017.
- [14] 韩辉. Digital Twin Service towards Smart Manufacturing[J]. 疾病监测, 2019.
- [15] FISHER C R, NYGREN K E, BEAUDOIN A J. Validation of materials-informed digital twin: Mapping residual strains in hsla steel weldment using high energy x-rays[J]. Journal of manufacturing processes, 2022(74-Feb.).
- [16] PAU F, JOAN G, V 柳 CtorGIC, 等. SARS-CoV-2 Spread Forecast Dynamic Model Validation Thorough Digital Twin Approach, Catalonia Case Study[J]. 2021.
- [17] WANG Y C, ZHANG N, LI H, et al. Research on digital twin framework of military large-scale uav based on cloud computing[J]. Journal of Physics: Conference Series, 2021, 1738 (1):012052 (6pp).
- [18] ZHOU H, YANG C, SUN Y. Intelligent ironmaking optimization service on a cloud computing platform by digital twin - sciencedirect[J]. 工程 (英文), 2021.
- [19] HADLEY F, DUNLAP T A, POON T. Precision treatment with machine learning and digital twin technology for optimal metabolic outcomes[Z]. [S.l.: s.n.], 2021.