

面向船舶系统的在线仿真平台的设计与实现

苏雨欣¹, 甘辉兵¹, 王安桐¹, 郭安康²

(1. 大连海事大学 轮机工程学院, 辽宁 大连 116026; 2. 中国船级社, 北京 100027)

摘要: 随着船舶行业的飞速发展, 研究内容日益细化, 一个船舶系统的开发往往需要多个团队的密切协作, 为了满足在线仿真和异地团队协同开发的需求, 针对船舶柴油发电机组调速系统, 搭建了调速系统优化的仿真模型, 并运用 Matlab Web App 技术和 Java Web 技术, 开发了一个在线仿真实验平台, 将调速系统的优化仿真模型与其同步的演示操作封装成 Matlab Web App, 实现了仿真模型的在线共享、互动及仿真的深度融合; 实践表明, 该在线仿真平台能够自定义仿真参数并顺利运行远程发布的仿真模型, 实时观察仿真结果, 极大地提高了异地团队的开发效率, 减少了重复开发的时间和成本, 为团队协作提供了一个高效、灵活的解决方案。

关键词: 船舶柴油发电机组; 调速系统; Matlab Web App; 在线仿真; 分布式开发

Design and Implementation of Online Simulation Platforms for Ship Systems

SU Yuxin¹, GAN Huibing¹, WANG Antong¹, GUO Ankang²

(1. College of Marine Engineering, Dalian Maritime University, Dalian 116026, China;

2. China Classification Society, Beijing 100027, China)

Abstract: With the rapid development of the marine industry, research is becoming increasingly specialized, and the development of a ship system often requires the close collaboration of multiple teams. To meet the needs of online simulation and collaborative development for remote teams, a simulation model is constructed to optimize the speed regulation system of marine diesel generator sets. The Matlab Web App technology and Java Web technology are used to develop an online simulation experiment platform. The optimization simulation model of the speed control system and its synchronous demonstration operation are packaged into a Matlab Web App, achieving the online sharing, interaction, and deep integration of the simulation model. Practice shows that the online simulation platform can customize simulation parameters and run remotely simulation models, observe simulation results in real-time, significantly improve the development efficiency of remote teams, reduce the time and cost of repeated development, and provide an efficient and flexible solution for team collaboration.

Keywords: marine diesel generator set; speed control system; Matlab Web App; online simulation; distributed development

0 引言

海洋运输是国际贸易重要的交通方式之一, 船舶作为海洋交通的运输载体, 对国家发展起着至关重要的作用^[1]。船舶系统复杂且体积较大, 以实船为对象开展研究较为困难且成本较高^[2]。20 世纪 50 年代, 仿真一词被首次提出, 随后被广泛应用于社会各领域, 航海领域也不例外, 建模仿真技术的引入极大提高了船舶工业的

技术发展和生产效率。但是随着船舶体积的增大以及功能的不断丰富, 对船舶体系的研究也逐渐细化, 一个完整的船舶仿真体系往往涉及多个团队的共同开发, 那么团队之间的远程资源共享就成了一个亟待解决的问题。

针对该问题, 国内外许多高校相继提出了多种解决方案。马里博尔大学^[3]基于 Matlab Web Server (MWS) 和 Matlab GUI 开发了两个远端实验平台, 在电气工程、控制系统领域实现了 Web 仿真, 用于电气工程和控制在

收稿日期:2024-08-29; 修回日期:2024-10-12。

基金项目:国家重点研发计划资助项目(2022YFB4301400)。

作者简介:苏雨欣(1999-),女,硕士研究生。

通讯作者:甘辉兵(1981-),男,博士,教授,博士生导师。

引用格式:苏雨欣,甘辉兵,王安桐,等. 面向船舶系统的在线仿真平台的设计与实现[J]. 计算机测量与控制, 2025, 33(10):208-215, 224.

系统领域的 Web 仿真。该平台允许用户远程自主设计控制函数,具有一定的交互性和灵活性,但其实现对 MWS 的高度依赖,使得客户端无法完全脱离 Matlab 环境,限制了用户的使用范围和平台的扩展性。美国俄勒冈大学开发的虚拟实验室平台采用 Java 技术,提供了多种虚拟实验供学生选择。用户可以通过登录平台进行仿真实验,但该平台还是要求学生在运行实验之前,从服务器下载客户端程序,增加了用户的技术交易所和操作复杂性。国内高校也采用了不同的方式实现团队内部资源共享,例如,北京邮电大学开发一种基于 C/S 架构的通信原理虚拟实验室,使用 Matlab 作为底层运算环境,并通过 Matlab Builder Ja 将实验所需组件封装成 Java 类和方法,结合 Java 的网络和界面编程技术,实现了平台内资源共享,从而使得系统用户可以自主搭建和修改实验^[4]。云南大学分别利用了 Matlab Builder Ja 组件结合 J2EE 构架以及 LabVIEW 自带的 Web 发布功能,均实现了将实验流程搬到 Web 端,终端用户通过一个浏览器即可完成已发布的实验^[5]。长江大学同样也采用 Matlab Builder Ja 组件和 Java 技术相结合的方式,开发了虚拟实验室,以 FIR 滤波器设计和拓展实验拍频信号在线演示为例,实现了系统内用户在 Web 端调用 Matlab 资源完成实验^[6]。上述平台的开发,以 Matlab 作为底层运算环境,采用 Java 语言开发管理系统和前端界面,关键技术采用的是 Matlab Builder Ja 组件,该组件可以将 Matlab 函数封装成 Java 类或方法,以 jar 包的形式嵌入到 Java 开发环境中,在被调用时通过启动 Matlab Compiler Runtime (MCR) 来完成底层计算^[7],再将结果显示到浏览器界面。但是该方法无法脱离 Matlab 服务器,也就是说使用该平台的每一个用户计算机上均需要装有 Matlab 软件,并且拥有正版许可证,而 Matlab 软件占据内存较大,且正版费用较贵,这就对成员的主机和团队经费都提出了较高的要求。另外,采用 Matlab Builder Ja 组件支持将 Matlab 函数封装 jar 包供 Java 调用,但是该方法不能直接与 Simulink 模型交互。因此本文利用 Matlab Web 技术代替传统的 Matlab Builder Ja 组件开发 Matlab Web App,利用 Java Web 技术开发平台的前端界面和管理系统,实现了 Simulink 模型以及 Matlab 函数的在线仿真,提高了团队的开发效率。

Matlab Web App 是一种基于 Matlab App Designer 开发的应用程序,能够通过网络浏览器实现交互式功能^[8]。Matlab Web App 集成了强大的计算能力,可以直接调用 Matlab 和 Simulink 的功能进行数据处理、仿真分析和算法测试。同时具备丰富的交互性和易用性,借助 Matlab App Designer 的拖拽式界面设计,用户可以轻松创建包含按钮、图形、输入框等元素的可视化界

面。这种高可定制化的界面设计使得科研人员能够开发符合实际需求的应用。最重要的是 Matlab Web App 可以实现快速部署和共享,通过 Matlab Web App Server^[9],可以将本地开发的应用程序发布到网络,团队成员只需通过浏览器输入 URL 即可访问应用,无需安装 Matlab 客户端^[10],从而降低了部署和维护成本。因此,对于工程仿真、模型验证以及复杂的数据分析,Matlab Web App 都为用户提供了高度集成的解决方案,本设计深度融合理论模型的搭建和在线仿真,开发更加适合团队内部 Matlab 资源共享的在线仿真平台。

基于 Matlab 开发在线仿真平台,传统的开发方式依赖 MCR,在适用范围方面存在较大的局限性^[11],相比之下,采用 Matlab Web 技术能够实现客户端完全脱离 Matlab 平台,并支持与 Simulink 模型的交互,极大提升了团队内部的资源共享效率。本文的研究思路如下:首先,本文设计并优化了船舶柴油发电机组的调速系统模型,通过模糊 PID 算法提升系统性能;其次,利用 Matlab Web 技术将该模型及优化算法封装为 Web App,生成唯一的 URL,便于在后台管理系统中集成和调用;随后,基于 Java Web 技术开发后台管理平台,为用户提供直观的导航界面,并管理各个应用程序;最后,本文详细探讨了该平台在船舶系统异地开发中的应用场景,并分析其对团队协作效率的提升作用。

1 整体设计

本章将从平台的设计思想、整体架构和实现的技术路线三部分来阐述平台的整体设计。

1.1 设计思想

该 Web 仿真平台整体上采用 B/S 架构^[12],该平台从逻辑层面上大致分为 3 层: UI 层、应用逻辑层以及模型层,其中模型层为核心模块。设计逻辑如图 1 所示。

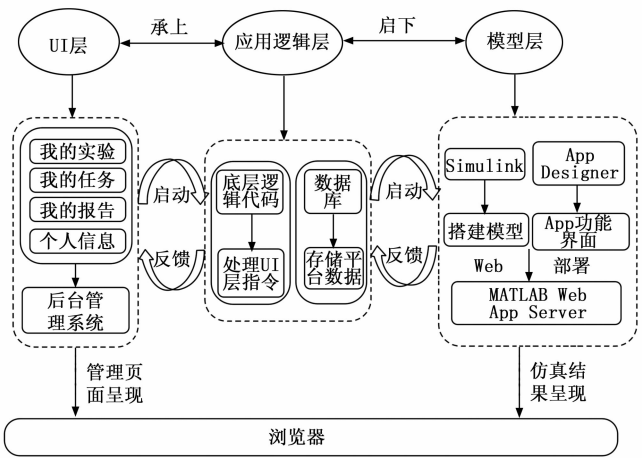


图 1 在线仿真实验平台设计图

UI 层包含我的实验、我的任务、我的报告以及个人信息 4 个功能模块,主要是平台的管理和导航界面,

负责用户身份的验证，并在平台内为用户提供导航功能。该层旨在指导用户选择适当的学习资料和仿真实验，同时提醒用户记录实验数据并提交实验报告。

应用逻辑层包含底层逻辑代码和数据库两个功能模块，在整个平台中起到了承上启下的作用。它上接前端界面所传递的各类运行指令，并将这些指令转发至模型层以启动仿真过程。同时，该层通过数据库的支持，将平台的学习资料、用户的个人信息、实验报告等重要数据反馈至前端界面，实现数据的实时展示功能。此层不仅负责平台的整体运行逻辑的梳理，还支持平台的正常运转，其核心包括后端逻辑代码、平台运行服务器以及数据库的集成使用。

模型层是平台仿真实验的核心支撑部分，负责为平台提供内容支持。该层的主要开发工具包括应用程序设计器（App Designer）、Simulink 插件，以及 Matlab Web App 服务器，通过这些工具实现仿真实验的开发与部署，为用户提供全面的仿真服务和技术支持。

基于 Web 的在线仿真平台的设计思想主要可以概括为以下几点:

1) 模块化设计与整合。在线仿真平台基于 Matlab Web App 的模块化特性, 实现了 Web App 关键要素 (仿真实验以及仿真示例的操作指南、Simulink 模型、模型架构、交互操作流程以及仿真结果等) 的在线整合, 完成 Simulink 模型的在线仿真, 形成学习和仿真模块的一体化平台。

2) 理论模型和应用程序设计器的集成构建。Web App 的设计将理论模型的搭建与应用程序设计器的开发有机结合, 确保模型和应用的同步开发, 保证模型和对应 Web App 的功能完美契合, 提升用户体验和仿真精度。

3) 基于模板化的设计。本研究以柴油发电机组调速系统为例,规定了在线仿真平台 Web App 的设计模板,团队内部成员可在此模板基础上根据不同模块特性添加相应的参数控件。采用模板化设计避免了 Web App 功能的重复开发,节省了团队的开发成本,同时保证了团队协作开发一致性。

4) 在线仿真与在线管理的有机结合。平台不仅支持 Simulink 模型的在线仿真及仿真结果可视化, 同时整合 Java Web 技术实现了实验、人员信息管理、资料学习和实验报告等功能, 打造了完整的在线仿真与管理

1.2 整体架构

如图 2 所示, 在线仿真平台由 Java Web 管理系统和 Matlab Web App 组成, 其中 Web App 为平台提供核心技术内容支持, 管理系统负责实验管理和后台资源管理, 辅助模块由 Java Web 页面组成。平台的核心模块要求团队成员按照任务分配和项目要求组织构建各自的

Web App, 为团队其他成员提供在线仿真的资源。团队内部成员可以通过平台管理模块在线发布和运行 Web App, 发布和查看实验报告。后台管理模块管理网站的基本参数和用户信息等内容。

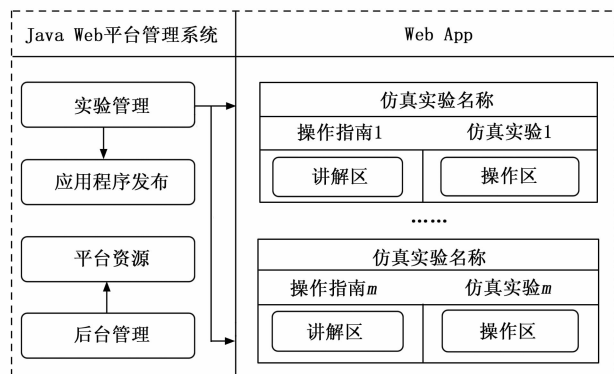


图2 在线仿真平台整体架构图

1.3 技术路线

在线仿真实验平台的开发主要依托于 Matlab Web 技术和 Java Web 技术。其技术路线如图 3 所示。

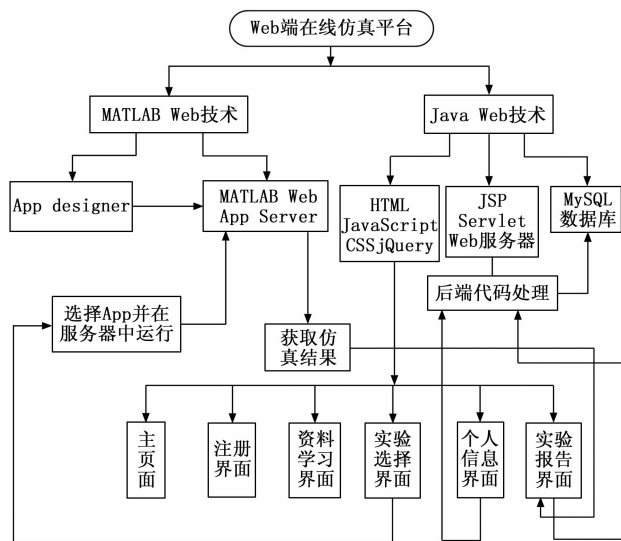


图3 在线仿真平台的技术路线

平台的核心模块采用 Matlab Web 技术开发。首先,在 Simulink 组件中搭建理论模型,并完成编译和运行。随后,通过应用程序设计器构建 Web App 功能的界面,实现与理论模型的通信,从而支持模型的在线交互式仿真。最后,将设计完成的应用程序部署到 Matlab Web App 服务器中形成 Web 应用程序,并为其生成唯一的 URL,用于后续在平台的辅助管理模块中进行集成,为平台提供核心技术支持,同时确保模型的仿真与计算过程仍以 Matlab 为底层运算环境。

平台的辅助管理模块采用 Java Web 技术开发,包括前端开发、后端开发以及数据库管理。通过前端开发

实现管理平台的界面功能(如主页面、平台用户信息注册界面、资料学习界面、实验列表管理界面、实验报告界面以及个人信息中心界面)。后端开发采用 JSP 与 Servlet 技术, Tomcat 作为 Web 服务器。MySQL 作为数据库系统。团队用户可以通过注册界面添加个人信息, 这些信息经后端代码处理后存储至数据库。登录该平台后, 用户可在实验列表界面点击相应的仿真链接, 跳转至对应的 Web App 进行仿真, 待仿真完成后, 用户可以记录结果数据, 完成实验报告进行提交, 实验报告同样经过后端代码处理存储至数据库, 以供团队其他成员使用。

2 在线仿真平台的整体实现

在线仿真平台的实现主要分为两个部分: 首先是 Web App 的设计与实现, 为平台提供核心内容; 其次是后台管理系统的实现, 主要为客户端提供信息管理服务 and 导航界面。

2.1 Web App 的设计与实现

如上文所述, Web App 的设计实现主要分为 3 个步骤: 理论模型的搭建、App 功能界面的设计以及最终的应用程序在 Web 端的部署。

本研究针对发电机组调速系统优化模型, 设计了 Web App 模板, 并在 Matlab R2022b 的应用程序设计工具中开发。船舶体系结构模块有难易之分, 用户可在此模板的基础上根据不同的模型修改仿真参数, 设计适合不同模型的 Web App。

2.1.1 模型的搭建

Matlab 和 Simulink 紧密集成, 开发者可以在 Matlab 环境中调用 Simulink 模型并使用 Matlab 工具箱进行数据分析和可视化等操作, 因此 Web App 的制作和实现也是基于这一功能来实现。

船舶柴油发电机组是船舶电力系统不可或缺的组成部分, 调速系统作为发电机的重要组成部分, 对其正常运行发挥着重要意义^[13]。柴油发电机的转速直接影响

船舶电网频率的稳定性以及发电机有功功率的输出, 进而影响船舶电力系统的稳定性, 关乎着船舶的正常运行和人员安全^[14]。本研究以柴油发电机组调速系统的优化为例, 详细介绍 Web App 的设计与实现过程。该 Web App 涉及调速系统的两种工况(小负载启动和负载突变)以及两种控制方法(PID 和模糊 PID), 即共涉及 4 个模型。

船舶柴油发电机组调速系统由速度控制器、执行器、柴油原动机、同步发电机以及三相电力负载组成^[15], 其控制如图 4 所示。

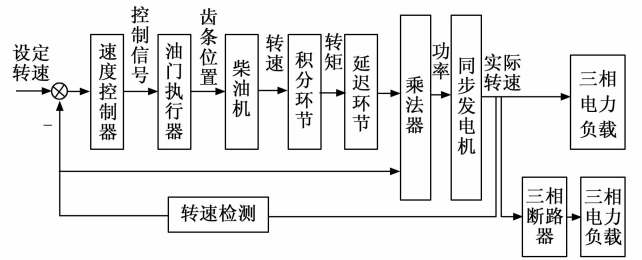


图 4 船舶柴油发电机组调速系统原理方框图

1) 速度控制器: 本研究分别采用 PID 控制和模糊 PID^[16]控制对调速系统进行控制, 通过 Web 端可以实时观察和对比柴油发电机在不同工况以及不同控制方式下的转速响应表现。PID 和模糊 PID 的结构如图 5 所示。其中经典 PID 控制的传递函数如下:

$$G(s) = K_p + K_i/s + K_d s \tag{1}$$

式中, K_p 为比例系数, K_i 为积分系数, K_d 为微分系数。

在本系统中, $K_p=15, K_i=5, K_d=0.25$ 。

模糊 PID 各参数计算公式如下:

$$K_p = K_p^0 + \Delta K_p \times f_p \tag{2}$$

$$K_i = K_i^0 + \Delta K_i \times f_i \tag{3}$$

$$K_d = K_d^0 + \Delta K_d \times f_d \tag{4}$$

式中, K_p^0, K_i^0 以及 K_d^0 为参数初始值, $\Delta K_p, \Delta K_i, \Delta K_d$ 为模糊控制器的输出, f_p, f_i 和 f_d 为解模糊因子。

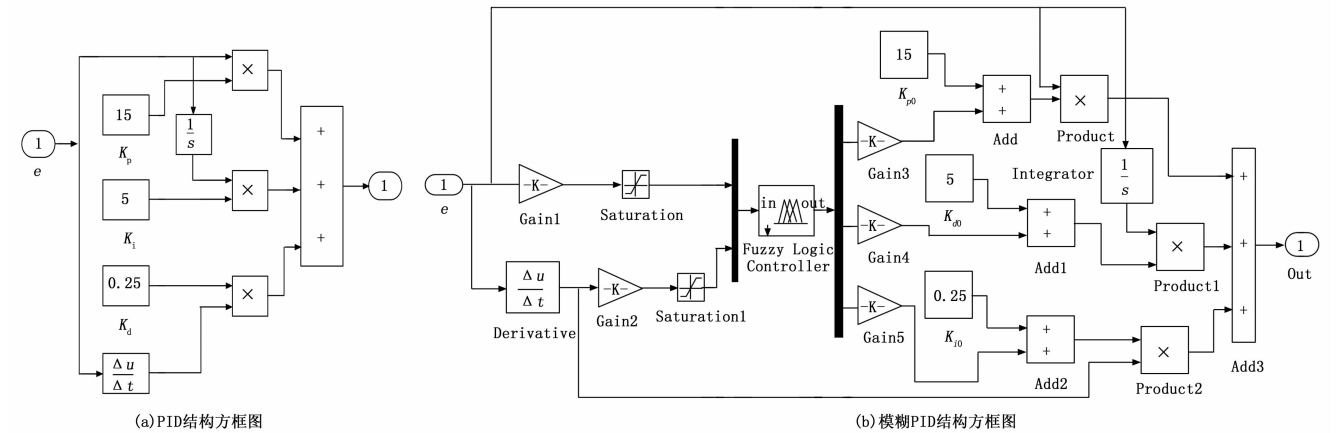


图 5 调速系统的两种控制方式

本系统中，参数初始值即为经典 PID 控制中各参数对应值， $f_p=5$ ， $f_i=2$ ， $f_d=0.05$ 。

2) 油门执行器：该模块采用电磁执行器，建模过程中采用一阶惯性环节代替电磁执行器，其传递函数如下：

$$G(s)=\frac{k}{T_k s+1} \tag{5}$$

式中， T_k 为执行器的惯性系数； k 为常数。

柴油机：柴油机带载时的动态方程为^[17]：

$$T_d s \alpha(s)+T_g(s) \alpha(s)=\beta(s)-\gamma(s) \tag{6}$$

当柴油机空载时，无 $\gamma(s)$ 项，对上式进行拉普拉斯变换，得到下式：

$$G(s)=\frac{1}{T_d s+T_g} \tag{7}$$

式中， T_d 为加速时间常数， T_g 为柴油机自稳定系数。

4) 积分环节：柴油机输出转速经过积分环节形成转矩。

5) 延迟环节：该环节的传递函数可以表达为：

$$G_\tau(s)=e^{-\tau s} \tag{8}$$

式中， τ 为该环节的滞后时间常数。

6) 乘法器：经延迟环节输出的转矩与柴油发电机实际转速相乘得到功率用于驱动柴油发电机的运行。

7) 反馈环节：获取柴油发电机的实际转速，并将其转换成与转速成正比的电压信号，该环节可以看作一个比例环节，放大系数为。

8) 同步发电机：同步发电机参考某远洋大型集装箱船舶（69 285 载重吨位），其相关参数如表 1 所示。

表 1 同步发电机相关参数

额定功率/kW	额定电压/V	额定转速/r·min ⁻¹
2 500	460	1 000

2.1.2 Web App 功能界面的设计与实现

Web App 的功能界面的设计与实现是通过 Matlab 提供的一种可视化应用程序设计工具^[18]来完成的，应用程序设计器提供了一个直观的图形界面并且支持多种 UI 组件和控件的集成，另外还兼容 HTML5，在开发过程中可以嵌入网页、多媒体和动画等效果，开发者可以自行设计应用程序的前端界面布局并且通过添加组件的回调函数来实现交互性和动态性^[19]。

船用柴油发电机组调速系统在线仿真实验包括模型验证以及控制方式优化的检测对比等内容，主要让团队成员获取调速系统部分的相关数据的软件实现。该 Web App 的开发强调模型设计与在线仿真的同步协调展开，设计了两个并行展示的功能区域：操作指南学习区和仿真实验操作区，可以同步展开学习和实操，其 Web App 模板如图 6 所示。操作指导区通过视频为团队其他成员提供该实验的操作步骤与注意事项，实验区

提供与该讲解视频相适配的仿真实验，具备与 Simulink 模型通信、自定义参数、仿真执行、结果输出以及结果对比功能。



图 6 调速系统仿真实验 Web App 模板

图 6 中的操作指南讲解区采用应用程序设计工具中的 HTML5 组件实现，其呈现效果如图 6 左侧所示，其中讲解视频转成 ogv 格式并采用 1 080 p（1 920×1 080）的分辨率^[20]，利用前端页面开发技术将视频文件通过 video 标签嵌入到 html 页面中，随后将该 html 文件添加到 Matlab 的存储路径中，以便 HTML5 组件的调用。

图 6 的右侧是该 Web App 的实验操作区，为了实现用户与模型的在线交互功能以及仿真结果的直观性，通过切换按钮组控件按照参数设定、结果展示和结果对比 3 个环节同步设计了该 Web App 的仿真实验操作区，用户可以按钮切换到对应模块。

1) 设定模型环节的实现，如图 6 右侧所示。在参数设定区采用编辑字段让用户调节设定转速、停止时间等参数，允许用户自定义数据，调整初始值；采用下拉框实现模型控制方式和运行工况的选择，其中控制方式包括“PID 控制”和“模糊 PID 控制”，运行工况包括“小负载启动”和“负载突变”；通过“Simulation”和“compare”按钮确定仿真的开始和结果对比的计算及显示。在模型结构区采用坐标组件显示对应参数的模型结构，帮助团队内部成员更好地了解该模型。在此环节中，用户需要设定参数并初始化仿真模型，这涉及应用程序设计器和 Simulink 模型之间的通信问题。

在应用程序设计器和 Simulink 模型的通信中，主要有两种实现方式：一种是通过工作区作为缓冲区传递参数，另一种是利用模型的输入和输出端口进行通信。

首先是通过工作区通信：在这种方式中，开发者可以通过应用程序设计器的界面组件设定模型参数，并利用 assignin 函数将参数传递并保存在工作区，模型的变量名需要和工作区中对应的参数变量名保持一致，以确保参数的正确传递。

第二种是通过输入输出模块通信：首先确定模型的输入端口数量，并使用 externalInput 函数以矩阵的形

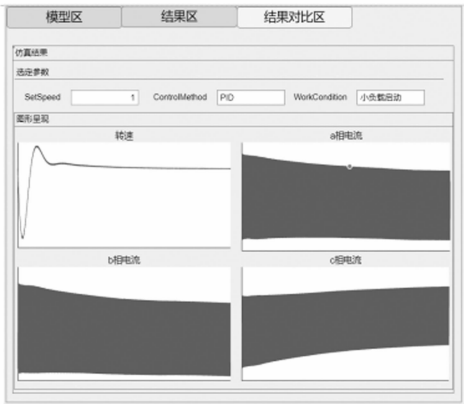
式为每个输入端口赋值, 接下来, 利用 `sim` 函数指定仿真模型并传递输入信号。待仿真结束后, 利用面向对象编程的思想获取仿真结果, 指定外部输入、调整仿真时间、设置仿真模式等。两者的结合使得用户灵活地控制 Simulink 模型的输入数据, 实现动态仿真控制, 支持与外部应用程序的实时交互。

在使用模型的输入输出端口实现数据通信时, 对模型的要求较为严格, 首先, 在 Simulink 中打开“模型设置”窗口, 并导航至“数据导入/导出”模块。在此模块中, 将“输入”和“初始状态”字段设置为空矩阵。接下来, 在“保存到工作区或文件”部分, 仅勾选“输出”和“单一仿真输出”选项, 同时指定“时间”和“输出”变量的名称。为了在应用程序设计器中准确获取仿真结果, 索引时需要严格按照设置的变量名进行。最后, 将数据格式设置为“带时间的结构体”, 以完成对模型的配置。

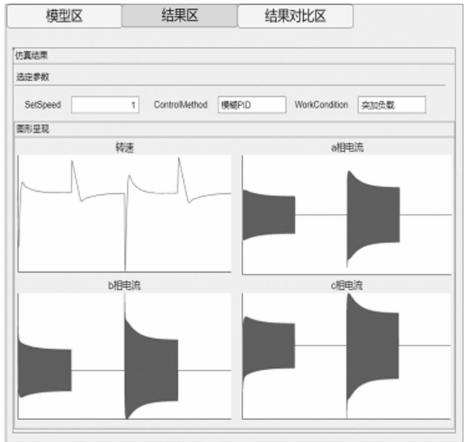
在实践过程中, 发现通过工作区通信时, 常常出现编译错误和信息遗漏等问题, 导致部署在 Matlab Web App Server 上的应用程序无法正常运行。相比之下, 采用输入输出端口通信可以减少数据传输中的潜在错误, 并实现对模型输入和输出数据的实时控制和监视。因此, 本研究选择通过模型的输入输出端口来实现应用程序设计器和模型的通信, 在该环节指定模型并完成对模型的初始化。

2) 结果显示环节的实现, 如图 7 所示, 其中 (a) 图是小负载 (15% 负载) 以 PID 控制方式启动时柴油发电机的转速及三相电流的变化趋势, (b) 图是在模糊 PID 控制方式下, 给系统进行两次突加突减负载 (第一次 25%, 第二次 50%) 时柴油发电机的转速及三相电流的变化趋势。在界面的第一栏采用编辑字段显示当前仿真结果对应的模型及参数。在界面的第二栏采用坐标轴绘制对应工况及控制方式下的仿真结果, 包括转速、柴油发电机的 a 相电流、b 相电流以及 c 相电流。通过结果显示区可以验证模型的正确性, 帮助团队成员更精准地开发其他模块。

3) 结果对比环节的实现, 如图 8 所示。该环节在坐标组件中绘制两种控制方式的结果对比曲线图。在表格组件中, 输出相应优化指标的结果, 小负载启动工况的优化指标为最大超调值、稳定时间、上升时间; 突加突减负载工况的优化指标为两次突加突减负载时的最大超调值和稳定时间。(a) 图展示了小负载启动工况下, 柴油发电机组调速系统在 PID 控制与模糊 PID 控制方式下的性能对比。由表格数据可知, 与传统 PID 控制方式相比, 模糊 PID 控制方式显著改善了系统的动态性能, 其中最大超调值降低了 51.14%, 稳定时间缩短了 38.95%, 上升时间减少了 17.16%。(b) 图反映了



(a) 小负载启动工况仿真结果图



(b) 突加突减负载工况仿真结果图

图 7 两种工况下的模型仿真结果图

在系统两次突加突减负载的工况下, 柴油发电机组调速系统在 PID 控制与模糊 PID 控制方式下的结果对比。具体来说, 当系统突加 25% 负载时, 采用模糊 PID 控制后, 最大超调值下降了 57.78%, 稳定时间缩短了 23.63%; 在负载突然下降 25% 时, 最大超调值减少了 22.27%, 稳定时间缩短了 34.65%。另外, 系统在第二次突加 50% 负载时, 模糊 PID 控制依然表现出显著的优化效果, 最大超调值下降了 57.99%, 稳定时间缩短了 17.2%; 当负载突然减少 50% 时, 最大超调值下降了 24.08%, 稳定时间缩短了 33.09%。该环节实现了调速系统的优化, 验证了模糊 PID 控制的有效性, 帮助团队成员更加直观地观察仿真结果以及优化对比, 还可以将相关优化算法运用到自己的模型当中, 实现更好的控制效果。

2.1.3 应用程序的 Web 部署

应用程序设计完成后还需要使用 Web App Compiler 打包成 Web 应用程序, 如图 9 所示, 整体分为 3 个步骤: 首先在 MAIN FILE 中添加设计完成的 `mlpp` 文件, 在应用程序详情中可以自行编辑作品名字, 添加作者名字以及应用程序的详细描述, 其次在必要文件中添加确保该应用程序正常运行的相关文件, 这里涉及了必

为了具体说明该平台的应用场景, 本节以船舶动力模块为例进行详细阐述。船舶动力模块可以大致划分为主机模块和辅机模块, 这两个模块通常由不同的异地团队分别开发。辅机模块的开发需要依赖主机模块的相关数据。如果主机模块采用 Matlab 开发, 并完成 Web App 的设计与实现, 则辅机模块开发团队可以采用其他仿真软件开发, 不用安装 Matlab 软件并且无需相关许可证, 即可在 Web 端获取主机模块的相关数据。在具体应用中, 辅机开发团队可以通过账户和密码登录平台的后台管理系统, 进入“实验项目列表”界面, 查找与所需数据对应的主机模块仿真项目。在找到对应项目后, 团队成员只需点击“Web App 地址”中的链接, 即可在浏览器中跳转到该模块的仿真实验页面。通过界面提示输入必要的仿真参数后, 即可实时获取并记录所需要的数据, 并用在辅机模块的开发模型中, 有效地避免了模型的重复开发, 极大地提高了开发效率, 实现了多团队之间的数据共享和模型协作, 为复杂系统的开发提供了高效、便捷的解决方案^[23]。

4 结束语

随着 Web 技术逐步迈入 Web3.0 时代, 利用 Web 端进行资源共享已经证明能够显著节省团队的开发时间和成本, 并有效提升开发效率。目前, LabVIEW 和 Matlab 等主流建模仿真软件已支持通过 Web 端发布和共享模型资源来满足团队协同开发的需求, 基于柴油发电机组调速系统的模型特性以及 Matlab 丰富的工具箱, 本研究采用了 Matlab Web 技术来实现模型资源的共享和在线仿真。

本文深入探讨了 Simulink 模型与在线仿真技术的深度融合, 开发了一个基于 Matlab Web 技术和 Java Web 技术的在线仿真实验平台, 重点集成和应用用于柴油发电机组调速系统的优化模型。该平台支持多用户在线访问和使用, 通过 Web 界面和 Web App 的结合, 用户不仅可以实时观察仿真过程和记录结果, 还能根据需要自定义仿真参数以匹配特定的模型, 满足多样化的实验需求。实践证明, 该在线仿真平台不仅具备 Matlab 的底层计算能力, 还结合了 Java Web 在后台管理系统开发中的优势, 使用户无需安装 Matlab 等软件即可通过浏览器访问并运行发布在平台上的 Web App。该平台在实际应用中展现出了卓越的效果, 不仅在很大程度上减少了团队内部重复开发的风险, 还显著节省了购买软件和相关学习的时间、金钱及人力成本。通过这一平台, 团队成员可以在任何地点、任何设备上便捷地访问和使用共享模型, 而无需安装复杂的软件环境, 从而大幅度简化了协同开发的流程, 显著提升项目的整体开发效率, 增强了团队的协作能力和灵活性。

该平台在以下两个方面需要进一步改进。首先是模型内容的支持部分。目前, 平台仅以柴油发电机组调速系统的优化模型为例, 开发了一个 Web App 模板。因此后续开发者可根据不同模块的需求和特性, 按照上述的开发流程, 在 Simulink 中构建各个模块的仿真模型。基于每个模型的输入输出参数和动态特性, 开发者可以在 Matlab 应用程序设计器中, 使用已有的 Web App 模板扩展出对应的 Web App, 把这些应用程序部署到 Matlab Web App 服务器中, 生成唯一的 URL。之后, 将每个模块的 URL 上传到后台管理系统实现新模型的发布, 从而实现对船舶系统中多个模块的集成, 供团队其他成员使用, 支持更广泛的应用场景和复杂系统的模拟与分析。其次, 平台的后台管理系统目前只支持用户以访客身份登录, 所有用户看到的界面都是相同的。未来, 可以开发一个管理员模块, 用于更好地管理用户和模型, 分配权限, 管理实验报告和发布通知等。这样可以提升平台的管理功能和用户体验, 进一步推动团队协作和研究工作的进行。

参考文献:

- [1] 叶韦辰. 船舶运输环境观测与数据传输系统的构建 [J]. 中国航务周刊, 2021 (32): 52-53.
- [2] 张勇亮, 张均东, 张志政. 三维船舶轮机虚拟实验室的设计和实现 [J]. 计算机应用与软件, 2019, 36 (1): 171-175.
- [3] URAN S, JEZERNIK K. Virtual laboratory for creative control design experiments [J]. IEEE Transactions on Education, 2008, 51 (1): 69-75.
- [4] 赵越. 通信原理虚拟实验平台的设计与实现 [D]. 北京: 北京邮电大学, 2012.
- [5] 程赛. 基于 Web 的通信信号处理实验平台设计与实现 [D]. 昆明: 云南大学, 2013.
- [6] 喻盼. 基于 Java 和 Matlab 的虚拟仿真实验系统的设计与实现 [D]. 荆州: 长江大学, 2018.
- [7] 钟熙, 孙祥娥. 基于拍频现象教学演示的 Matlab Builder JA 方法 [J]. 电脑知识与技术, 2019, 15 (16): 124-126.
- [8] REN R, SUN X E, HU L. A new method for hosting and sharing Matlab Web App [J]. Scientific Reports, 2022, 12 (1): 21645.
- [9] MathWorks. Matlab web app server user's guide [EB/OL]. (2024-07-19) [2024-8-21]. <https://ww2.mathworks.cn/products/matlab-web-app-server.html>.
- [10] FAUZI M F A, NORDIN R, ABDULLAH N F, et al. Machine learning-based online coverage estimator (MLOE): advancing mobile network planning and optimization [J]. IEEE Access, 2023, 11: 3096-3109.

(下转第 224 页)