

8-13-2020

Hardware-in-the-Loop Simulation System for Underwater Vehicle Based on Unity3D

Fushen Ren

Northeast Petroleum University, Daqing 163318, China;

Yaqi Sun

Northeast Petroleum University, Daqing 163318, China;

Hu Qing

Northeast Petroleum University, Daqing 163318, China;

Zhaoliang Li

Northeast Petroleum University, Daqing 163318, China;

See next page for additional authors

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Hardware-in-the-Loop Simulation System for Underwater Vehicle Based on Unity3D

Abstract

Abstract: In order to provide a simulation and test platform for the performance test, control algorithm analysis and operation training of underwater robot, a hardware-in-the-loop simulation system based on Unity3D is developed, which consists of a ground control terminal, a robot body and a three-dimensional simulation system. *The system uses the same ground control terminal as the unified command input source of the three-dimensional simulation system and robot control system, and realizes the synchronization of the virtual object and robot body, real-time monitoring and parameters adjustment to the system. The position of the robot is collected by multiocular vision technology, and Kalman filtering processing is carried out on sensor data.* The robot motion simulation results show that the actual motion path of the robot is basically the same as the simulation planning path, and the hardware-in-the-loop simulation system has good dynamic response and control system synchronization.

Keywords

hardware-in-the-loop simulation, Unity3D, control system, ROV, real-time interactive

Authors

Fushen Ren, Yaqi Sun, Hu Qing, Zhaoliang Li, Pengyu Sun, and Yukun Fan

Recommended Citation

Ren Fushen, Sun Yaqi, Hu Qing, Li Zhaoliang, Sun Pengyu, Fan Yukun. Hardware-in-the-Loop Simulation System for Underwater Vehicle Based on Unity3D[J]. Journal of System Simulation, 2020, 32(8): 1546-1555.

基于 Unity3D 的水下机器人半实物仿真系统

任福深, 孙雅琪, 胡庆, 李兆亮, 孙鹏宇, 范玉坤

(东北石油大学, 黑龙江 大庆 163318)

摘要: 基于 Unity3D 研制了一种由地面控制终端、机器人本体和三维仿真系统 3 部分组成的水下机器人半实物仿真系统, 为水下机器人开发过程中的性能测试、控制算法分析和人员操作培训提供了仿真和测试平台。该系统采用同一地面控制终端作为三维仿真系统和机器人控制系统的统一指令输入源, 以实现虚拟对象和机器人本体的同步工作, 可对系统进行实时监测、调参。采用多目视觉技术对机器人位置采集, 对传感器数据进行卡尔曼滤波处理。机器人运动仿真试验结果表明, 机器人实际运动路径和仿真规划路径基本相同, 该系统动态响应性和控制系统同步性较好。

关键词: 半实物仿真; Unity3D; 控制系统; 水下机器人; 实时交互

中图分类号: TP391.9 文献标识码: A 文章编号: 1004-731X (2020) 08-1546-10

DOI: 10.16182/j.issn1004731x.joss.18-0845

Hardware-in-the-Loop Simulation System for Underwater Vehicle Based on Unity3D

Ren Fushen, Sun Yaqi, Hu Qing, Li Zhaoliang, Sun Pengyu, Fan Yukun

(Northeast Petroleum University, Daqing 163318, China)

Abstract: In order to provide a simulation and test platform for the performance test, control algorithm analysis and operation training of underwater robot, a hardware-in-the-loop simulation system based on Unity3D is developed, which consists of a ground control terminal, a robot body and a three-dimensional simulation system. The system uses the same ground control terminal as the unified command input source of the three-dimensional simulation system and robot control system, and realizes the synchronization of the virtual object and robot body, real-time monitoring and parameters adjustment to the system. The position of the robot is collected by multiocular vision technology, and Kalman filtering processing is carried out on sensor data. The robot motion simulation results show that the actual motion path of the robot is basically the same as the simulation planning path, and the hardware-in-the-loop simulation system has good dynamic response and control system synchronization.

Keywords: hardware-in-the-loop simulation; Unity3D; control system; ROV; real-time interactive

引言

随着对海洋资源的探索需求显著增加, 水下机器人在军事及民事领域获得了更加广泛的应用。由

于水下机器人设备昂贵、作业环境特殊, 需要对使用人员进行专门的操作培训^[1]。此外, 新型水下机器人研究过程中, 需要对机器人的各项性能进行评估, 因此设计并开发出一套能够模拟水下作业环境的水下机器人仿真系统, 不但能够降低水下机器人系统开发的成本, 同时也可以为现有的水下机器人提供操作培训环境, 是智能化水下机器人研究的关键技术和动态建模的有效平台^[2]。

目前, 建立仿真系统平台有很多种方法, 但是



收稿日期: 2018-12-19 修回日期: 2019-03-12;
基金项目: 中国石油天然气集团公司科学研究与技术开发项目(2019A-1013);
作者简介: 任福深(1976-), 男, 辽宁辽阳, 博士, 教授, 博导, 研究方向为智能装备及其控制理论。

<http://www.china-simulation.com>

• 1546 •

并非都适合建立半实物水下机器人仿真系统。VRML 或 OpenGL 具有很好的图形驱动引擎, 并提供了三维模型图调入的接口, 可展示相应的运动情况和场景预制, 但具有三维物理特征的场景效果开发困难, 运动仿真占用系统内存较大, 开发过程复杂, 模拟具有三维物理特征的实际效果受到了很大的限制^[3]; RT-LAB 系统平台具有很好的开放性和可扩展性, 但运行过程中需要手动修改编译模块, 比较繁琐, 平台的通用性也较差^[4]; 采用 C/C++、Java 或 Matlab/Simulink 语言进行编程开发的软件平台多对算法进行仿真, 输出的三维图像对象较为单一, 不适合用于整体环境的仿真^[5], 采用 Vxworks 系统数据交换方便, 实时性较好, 但功能扩展性较差^[6]; 采用 UWSim 软件构建虚拟场景, 需使用标准的建模软件并配置连接到外部控制程序, 同时要求安装机器人系统, 且程序修改较为复杂^[7]。Unity3D 技术在建立具有物理特征的三维场景方面具有一定的优势, 开发过程容易, 可以节省开发者大量的精力而达到预期的效果, 适合复杂的半实物仿真系统的开发。本文基于 Unity3D 技术, 采用将真实机器人控制终端与具有水下物理特征的三维仿真系统相结合的设计方法, 开发了一套水

下机器人半实物仿真系统, 为水下机器人开发过程中的性能测试、控制算法分析和操作培训提供了仿真和测试平台。

1 半实物仿真系统整体构架

如图 1 所示, 半实物仿真系统主要由摇杆手柄、地面控制终端、水下机器人本体以及三维仿真系统组成。

系统中摇杆手柄的控制信号传输到机器人地面控制终端的工控机中, 工控机通过光纤通讯的方式将控制信号传输给机器人本体的控制系统, 摇杆手柄的模拟量信号通过 A/D 转换还可以同时输入到仿真计算机的三维仿真系统中, 实现对机器人本体和仿真系统中机器人的联动控制。此外, 水下机器人作业过程中, 上位机控制终端会实时采集 ROV (Remote Operated Vehicle) 本体的潜水深度、航行速度、航行方位和机器人姿态等信息, 将这些信息重新装载到仿真系统中, 可再现水下机器人本体的作业过程, 通过对比同一控制信号下机器人实际运动与仿真环境中的运动差异, 校正仿真系统中水下环境物理参数, 确保仿真系统的真实性和有效性。

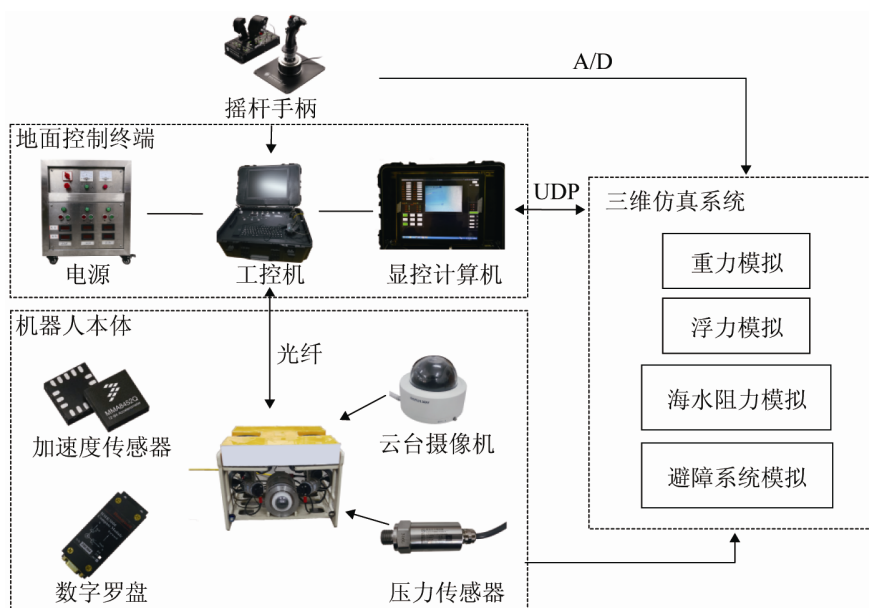


图 1 半实物仿真系统整体构架

Fig. 1 Whole framework of hardware-in-the-loop simulation system

2 可视化半实物仿真系统设计

半实物仿真系统各个功能模块的逻辑关系如图 2 所示。仿真系统中的仿真计算机通过 TTL 数字信号与工控机进行连接并建立通信，水下机器人的传感器数据经由串口数据采集模块进入仿真系统的数据管理模块。采用 Unity3D 作为仿真软件分别对水下物理环境和水动力进行仿真，通过程序设计实现照明系统、声系统、水下粒子系统、推进系统和水流阻力系统的仿真。仿真获得的数据由数据输出模块输出到地面控制终端的显控计算机中，通过 UDP 协议建立数据交互，将显控计算机中实时获取的图像信息与解算出的机器人作业状态信息进行对比，观察运动情况，及时反馈到仿真系统中对参数信息进行修正。

2.1 水下物理环境设计

基于 Unity3D 对三维仿真平台进行设计，水下机器人的运动状态信息保存为 sqlite 数据库文件，通过软件的数据库模式读取数据库文件并采用 UDP 协议进行传输。Unity3D 以其方便、高效以及直观的特点为用户所接受，可以被用来创建交互式应用程序。软件的编辑器允许开发人员管理项目(包括 3D 模型、材质、纹理、字体等等)，创建必

要的交互逻辑^[8]，利用该软件完成地形地貌、环境配置等空间设置，使所搭建完成的三维海洋场景在显示及控制方面具有良好的交互性，较大程度的还原三维实景，满足对真实环境的监控需要。仿真环境搭建流程如下：

- (1) 在 Solidworks 中依据实际参数对水下机器人模型、作业管道和设备模型进行绘制；
- (2) 利用 3ds Max 建立水生生物模型，并导出成 Unity 可识别的统一格式(.FBX 格式)，进而导入三维场景中；
- (3) 在项目工程中创建地形，利用地形编辑器，改变笔刷大小、材质和高度，设置洋中脊的距离以及起伏程度；
- (4) 使用贴图(Texture)和着色器(Shader)相配合，完善水生生物的材质、纹理以及灯光效果等信息的添加；
- (5) 调入所需应用资源包，完成天空盒子以及粒子资源的调用，海洋环境中的天空盒子可用来为周边环境设置基本颜色，粒子用来模拟螺旋桨转动时产生的气泡效果和水流效果；
- (6) 对项目中的模型添加刚体(Rigidbody)属性，勾选“Use Gravity”属性实现碰撞效果，勾选“Audio Source”属性添加水声音效；
- (7) 添加机器人相机视角、跟随视角和整体视角 3 种不同视角，进行不同角度的观察；
- (8) 提供控制接口，与显控计算机通过 UDP 进行数据交互。

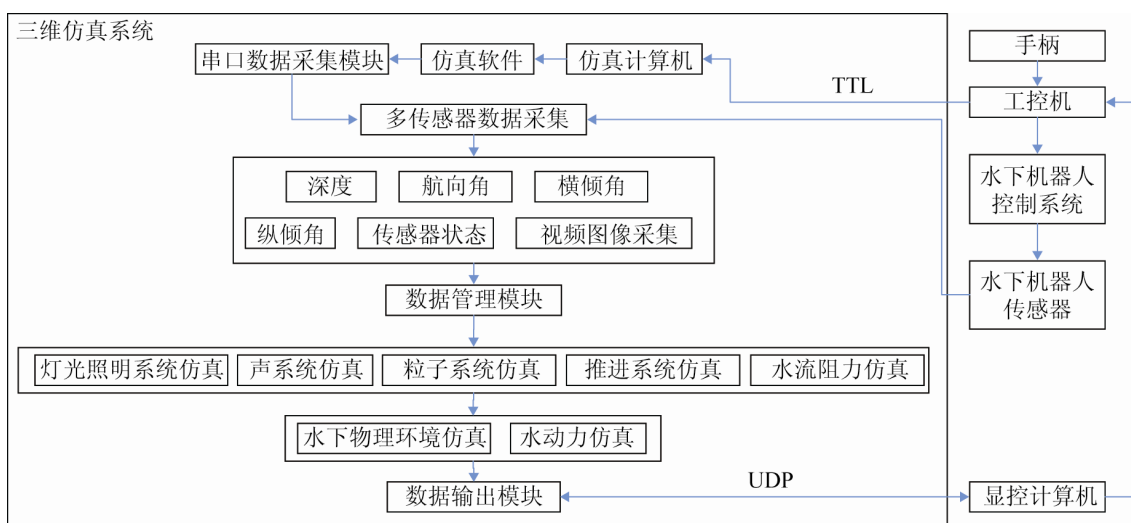


图 2 半实物仿真系统逻辑图

Fig. 2 Logic Diagram of hardware-in-the-loop simulation system

<http://www.china-simulation.com>

• 1548 •

2.2 力的仿真与计算

Unity3D 对图形、音频、物理、网络等多方面提供了引擎支持, 并使用编辑器来进行整合。利用软件的 PhysX 物理系统中的 API 函数能对机械力的运动状态进行仿真, 通过对程序中代码的编写赋予模型及仿真环境真实的物理状态。

2.2.1 推进器的仿真

ROV 本体共配置了 6 个推进器, 2 个垂直推进器平行布置, 共同实现升沉和横倾两个自由度; 4 个水平推进器采用 45° 矢量环形布置的原则, 有利于降低耦合产生的有害力矩, 在水平方向获得更好的运动性能。水平面推进器布置方式如图 3 所示。

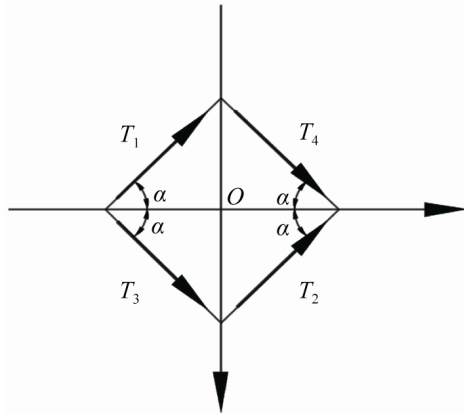


图 3 水平面推进器布局示意图
Fig. 3 Layout of horizontal thrusters

螺旋桨模型形式选用三叶 B 型螺旋桨, 根据实际设计参数计算出推力 T 大小:

$$T = \rho n^2 D^4 K_T \quad (1)$$

式中: T 为螺旋桨推力; ρ 为水密度; n 为转速; D

为螺旋桨直径; K_T 为推力系数。

根据式(1)可以求得螺旋桨的最大推力, 将所求得的推进力添加到场景中的水下机器人的推进器上。在场景中先为机器人添加刚体属性, 并检测碰撞体, 设置程序调用时间, 通过程序编写推力曲线计算方法, 为机器人的运动提供动力。

2.2.2 环境作用下的水动力仿真

环境影响是基于实际区域的洋流分布, 会在每个自由度中被转化为力或力矩。作用在机器人上的外力包括流体回复力(浮力与重力的合力)、水动力以及拖曳缆绳的张力^[9], 因此有:

$$F = F_W + F_T + F_H + F_{TH} \quad (2)$$

式中: F_W 为流体静回复力; F_T 为脐带缆张力; F_H 为作用在机器人主体上的水动力; F_{TH} 为控制各个螺旋桨的推进力的合力。

考虑机器人主体受螺旋桨造成的周围流场影响后, 分析数据构造螺旋桨的几何模型, 进而模拟螺旋桨在运动流场中的水动力特性。水流阻力主要表现为摩擦阻力和粘压阻力, 水的粘性作用不可忽略, 但可在设计过程中通过减小附体面积和改善机器人表面粗糙程度等手段减小粘性。

2.3 水下机器人运动仿真与姿态再现设计

提取机器人自身传感器数据, 经过坐标转换, 将机器人机体动坐标系下的姿态数据转换为地理坐标系下的坐标, 将转换后的姿态数据应用到三维场景中的水下机器人模型上, 坐标变换矩阵写为:

$$\eta_0 = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \varphi - \sin \psi \cos \varphi & \sin \psi \sin \varphi + \cos \psi \cos \varphi \sin \theta & 0 & 0 & 0 \\ \sin \psi \cos \theta & \cos \psi \cos \varphi + \sin \varphi \sin \theta \sin \psi & \sin \theta \sin \psi \cos \varphi - \cos \psi \sin \varphi & 0 & 0 & 0 \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & 0 & 0 & 0 & \cos \varphi & -\sin \theta \\ 0 & 0 & 0 & 0 & \sin \varphi / \cos \theta & \cos \varphi / \cos \theta \end{bmatrix} \eta \quad (3)$$

式中: $\boldsymbol{\eta}=[x\ y\ z\ \phi\ \theta\ \psi]^T$; $\boldsymbol{\eta}_0=[x_0\ y_0\ z_0\ \phi_0\ \theta_0\ \psi_0]^T$; x, y, z 为运动坐标系原点的地面坐标值; ϕ 为机器人的横滚角; θ 为机器人的纵倾角; ψ 为机器人的偏航角。

Unity3D 软件可通过数据通信端口接收数据和发送指令, 开启多个线程用于数据的接收与处理, 在数据池中对数据进行处理。提取出的数据通过代码添加到程序中水下机器人模型上, 运行程序后, 机器人的位置、姿态信息可进行在线显示。在项目的 Asset 路径下, 新建 StreamingAssets 文件夹, 利用程序代码将三维坐标实时输出到建立的 TXT 文件所在的路径中, 针对每一帧的位置变化情况, 对输出值进行记录。将输出的 TXT 文件通过 MATLAB 导入数据, 记录水下机器人执行各种命令时的运动轨迹。

3 机器人数据采集与处理研究

ROV 本体主控 ECU 板具备多传感器数据采集与交互的功能, 控制终端的工控机需要采集的信息主要包括压力传感器信息, 罗盘艏向、俯仰、纵倾的姿态角信息, 罗盘数据通过 RS485 提供串口输出。采集到的全部数据会通过 RS485 通信对上位机进行反馈, 进而提取到仿真系统中对场景中的模型进行参数赋值。

3.1 多目视觉技术采集位置信息研究

常用获取位置距离的测量方法有激光测距, 单目视觉技术, 双目视觉技术和多目视觉技术, 为获取机器人本体在试验环境中的位置信息, 本文引入了多目视觉技术, 为符合光线在空气和水两种介质的折射问题, 对一般多目视觉技术进行了改进。

在试验环境布置多个摄像机, 利用获得的多台摄像机反馈的图像信息进行立体匹配。由于机器人所处的世界坐标和摄像机图像的像素坐标不一致, 需要通过坐标系变换, 利用像素点坐标求解空间点的世界坐标^[10]。

空间点 $P=(X^w, Y^w, Z^w)$ 到以不同摄像机角度看到的点 $P_c=(X^c, Y^c, Z^c)$ 转化可表示为:

$$\begin{bmatrix} X^c \\ Y^c \\ Z^c \end{bmatrix} = \mathbf{R}_w^c \begin{bmatrix} X^w \\ Y^w \\ Z^w \end{bmatrix} + \mathbf{t} \quad (4)$$

$$\text{式中: } \mathbf{R}_w^c = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}; \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}。$$

摄像机坐标系 $O-X^cY^cZ^c$ 到成像平面坐标系 $O-x_d y_d$ 的转化可表示为:

$$\begin{cases} x_d = \frac{fX^c}{Z^c} \\ y_d = \frac{fY^c}{Z^c} \end{cases} \quad (5)$$

式中: f 为摄像机焦距。

将式(5)整理为的齐次方程形式为:

$$Z^c \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{bmatrix} \quad (6)$$

成像平面坐标系 $O-x_d y_d$ 到像素坐标 $O-uv$ 的转化关系可表示为:

$$\begin{cases} u = \frac{x_d}{dx} + u_0 \\ v = \frac{y_d}{dy} + v_0 \end{cases} \quad (7)$$

式中: dx 为像素在成像平面坐标系 X 轴方向上的物理尺寸; dy 为像素在成像平面坐标系 Y 轴方向上的物理尺寸; $(u_0, v_0)^T$ 为主点;

将式(7)整理为的齐次方程形式为:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \quad (8)$$

由于水下成像环境成像时, 光线从水中进入空气中, 在通过不同介质交界面时光线会发生折射, 折射的产生会对数据造成较大的影响, 空气中的摄像机线性模型不再适用, 因此, 为精确多目视觉定位的试验结果, 需要对摄像机成像模型进行更准确的描述, 水下成像模型如图 4 所示。

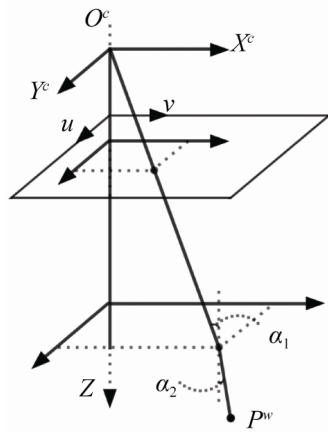


图 4 水下成像模型

Fig. 4 Model of underwater-imaging

当光线通过不同介质界面发生折射时, 成像平面坐标系与相机坐标系的转化可表示为:

$$\begin{bmatrix} \alpha_w \\ \beta_w \\ \gamma_w \end{bmatrix} = n \begin{bmatrix} \frac{1}{\sqrt{x_d^2 + y_d^2 + f^2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{x_d^2 + y_d^2 + f^2}} & 0 \\ 0 & 0 & -\sqrt{\frac{1}{n^2} + \frac{f^2}{x_d^2 + y_d^2 + f^2}} - 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \quad (11)$$

综合式(4), (6), (8), (9), (11)可得到空间点任一点 $P(X^w, Y^w, Z^w)$ 到像素坐标系中 $p(u, v)$ 的转化公式为:

$$k \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{x_d^2 + y_d^2 + f^2}}{n} & 0 & 0 & -\frac{dx_d \sqrt{x_d^2 + y_d^2 + f^2}}{nf} \\ 0 & \frac{\sqrt{x_d^2 + y_d^2 + f^2}}{n} & 0 & -\frac{dy_d \sqrt{x_d^2 + y_d^2 + f^2}}{nf} \\ 0 & 0 & \frac{1}{\sqrt{1 + \frac{n^2 f^2}{x_d^2 + y_d^2 + f^2}} - n^2} & \frac{d}{\sqrt{1 + \frac{n^2 f^2}{x_d^2 + y_d^2 + f^2}} - 1} \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} = \mathbf{M}_1 \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} = \mathbf{M}_1 \mathbf{M}_2 \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} \quad (12)$$

式中: \mathbf{M}_1 为摄像机内部参数矩阵; \mathbf{M}_2 为摄像机外部参数矩阵; 因此根据机器人本体在不同拍摄视角下、不同摄像机成像面上的像素坐标, 可求得机器人相对于像素坐标系的世界坐标系坐标。

3.2 传感器数据卡尔曼滤波

仿真模型的准确建立以及对数据的分析与处理对半实物仿真系统的搭建起着决定性作用。ROV 在水中工作,其推进器旋转和 ROV 自身的运动会导致水体发生变化,加上水体本身环境复杂,造成机器人机械结构间摩擦和碰撞以及机器人与水流的相互作用,使得机身波动中含有不确定的高频振动,最终导致传感器对机器人的姿态和运动信息测量时,其结果 \mathbf{M} 含有周期性波动和随机信号,鉴于此问题,本文采用对仿真系统获得的数据进行卡尔曼滤波处理,从而实现和实际机器人运动状态同步,提高仿真效果。

根据机器人实际情况,其过程噪声和测量噪声可写为:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \boldsymbol{\eta}(k) \quad (13)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \quad (14)$$

式中: $\boldsymbol{\eta}(k)$ 和 $\mathbf{v}(k)$ 分别为模型噪声和高斯白噪声,可近似定义为期望为 0, 方差为 \mathbf{Q} 和 \mathbf{R} 高斯白噪声序列^[11]。设计如下卡尔曼滤波算法:

$$\begin{aligned} \mathbf{X}(k+1|k) &= \mathbf{A}\mathbf{X}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{q}(k) \\ \mathbf{Y}(k+1) &= \mathbf{Y}(k+1) - \mathbf{C}\mathbf{X}(k+1|k) - \mathbf{r}(k) \\ \mathbf{P}(k+1|k) &= \mathbf{A}\mathbf{P}(k)\mathbf{A}^T + \mathbf{Q}(k) \\ \mathbf{P}(k+1) &= [\mathbf{I} - \mathbf{K}(k+1)\mathbf{C}]\mathbf{P}(k+1|k) \\ \mathbf{X}(k+1) &= \\ & \mathbf{X}(k+1|k) + \mathbf{K}(k+1)[\mathbf{Y}(k+1) - \mathbf{C}\mathbf{X}(k+1|k)] \end{aligned} \quad (15)$$

式中: \mathbf{A} , \mathbf{B} 为系统矩阵; \mathbf{C} 为系统观测矩阵; \mathbf{P} 为估计误差的协方差矩阵; \mathbf{K} 为卡尔曼增益矩阵。

由于观察噪声方差阵和控制噪声方差阵均未知,对两种方阵进行同时估计时,会造成滤波发散,考虑到观察噪声主要由人工环境原因引起,难以补偿,控制噪声可以通过其他方式抑制,因此本文仅估计测量噪声均值 r 和测量噪声方差阵 \mathbf{R} 。

其递推算法如下:

$$\begin{aligned} r(k+1) &= (1-d(k))r(k) + \\ & d(k)[Y(k+1) - C(k+1)X(k+1|k)] \\ \mathbf{R}(k+1) &= (1-d(k))\mathbf{R}(k) + \\ & d(k)[Y(k+1)Y(k+1)^T - \mathbf{C}\mathbf{P}(k+1|k)\mathbf{C}^T] \end{aligned} \quad (16)$$

式中: $d(k) = (I-b)/(I-b^{k+1})$, b 为遗忘因子,其范围为 $0 < b < 1$, 可通过在线递推进行修正。

以艏向角数据为例,对系统采集的机器人传感器数据进行处理,系统变量选择为艏向角 α , 角速度 ω_α , 角加速度 a_α , 即 $\mathbf{X}_k = [\alpha, \omega_\alpha, a_\alpha]$, 系统矩阵 $\mathbf{A} = \begin{bmatrix} 1 & -0.02 \\ 0 & 1 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 0.02 \\ 1 \end{bmatrix}$, 系统观测矩阵 $\mathbf{C} = [1 \ 0]$, 根据 ROV 在水中转艏时实际运动特点,艏向角模型噪声均值为 0, 方差 $\mathbf{Q} = \text{diag}\{0.1^2 \ 0.1^2\}$, 测量噪声的初始均值为 $\mathbf{r}_0 = [0.01 \ 0.01]^T$, 初始方差 $\mathbf{R}_0 = \text{diag}\{1^2 \ 1^2\}$ 。

对机器人艏向转角 $15^\circ, 30^\circ$ 时的传感器数据进行 2 次滤波处理,数据实际输出和滤波效果如图 5 所示。黑色曲线代表系统采集到的艏向角原始数据,蓝色曲线代表经过两次卡尔曼滤波后的艏向角变化情况,对机器人转艏运动艏向角传感器数据进行了有效滤波,输出数据比较平滑,大幅减小了噪声。

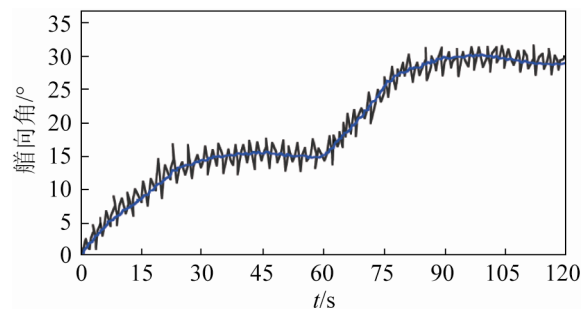


图 5 艏向角卡尔曼滤波效果
Fig. 5 Kalman filter effect of heading angle

4 系统仿真与试验

基于研制的半实物仿真系统开展了仿真实验,以验证控制系统的实用性和控制方法的准确性,半实物仿真系统实物如图 6 所示。

在室内游泳馆深水区开展了水下机器人的仿真试验,操作人员通过操作摇杆控制水下机器人的前进、后退、横移、上浮和下潜等运动,同时操作指令信号实时输入到计算机的仿真系统中。仿真系统根据接收到的指令信号,采用与机器人

本体控制系统相同的计算方法, 将操作摇杆指令数据换算为水下机器人螺旋桨的旋转速度。将试验测得的螺旋桨转速与推力添加到三维仿真系统中, 确保每个螺旋桨的推力附加到仿真系统中的机器人中的准确性, 从而驱动仿真系统中机器人实现运动。

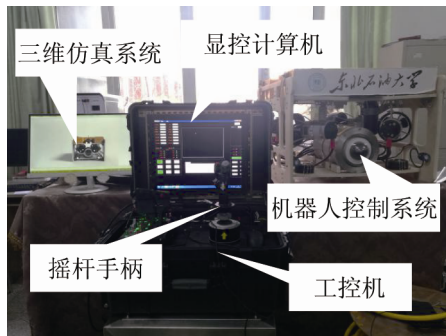


图 6 半实物仿真系统

Fig. 6 Hardware-in-the-loop simulation system

试验和仿真实物效果如图 7~8 所示, 分别开展了直线运动、曲线运动、上浮、下潜、横移和调头等运动仿真试验。机器人的运动方位和姿态通过实时采集机器人上的罗经仪数据获得; 机器人的运动位置通过架设在游泳池四周的多个摄像机记录; 机器人在水中的深度信息通过实时采集机器人上的水深传感器获得。最后, 将机器人上测得的数据信息与机器人仿真系统中提取的数据信息进行图像绘制, 通过对比, 分析机器人仿真系统的有效性和实用性。

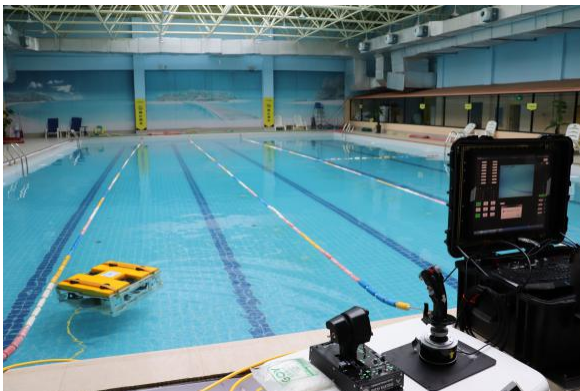


图 7 试验过程中水下机器人运动

Fig. 7 Motion of ROV in the process of test

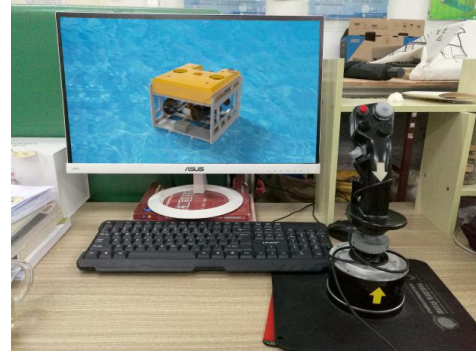
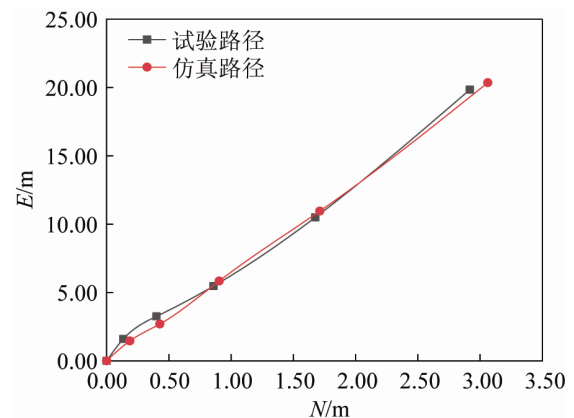


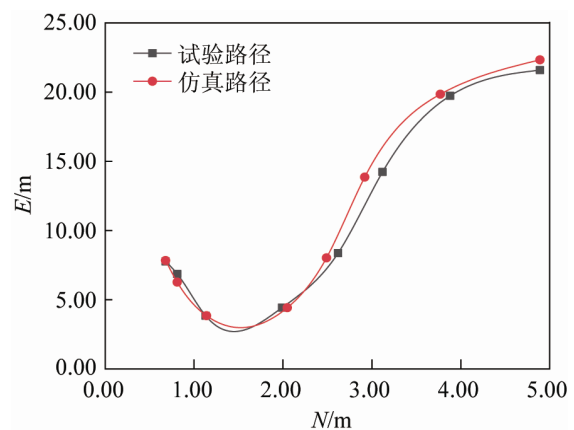
图 8 仿真系统中的水下机器人运动

Fig. 8 Motion of ROV in simulation system

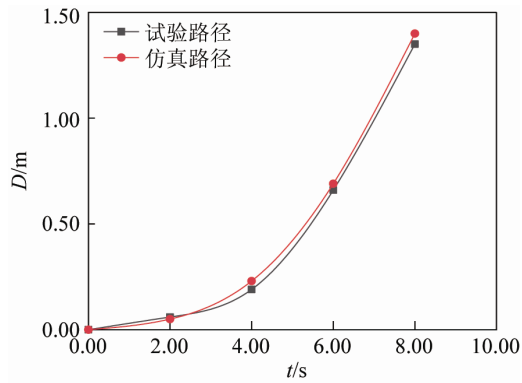
如图 9 所示, 通过试验路径和仿真路径曲线的结果对比, 可以发现由于仿真环境中关键系数如水动力、流体恢复力、水流阻力等参数难以被精确测量和不确定外界扰动会对仿真结果产生一定的影响; 操作摇杆手柄输入指令的突变会使结果出现微小的位置偏差, 但对达到最后预期位置坐标的影响不大, 试验路径和仿真规划路径基本相同。



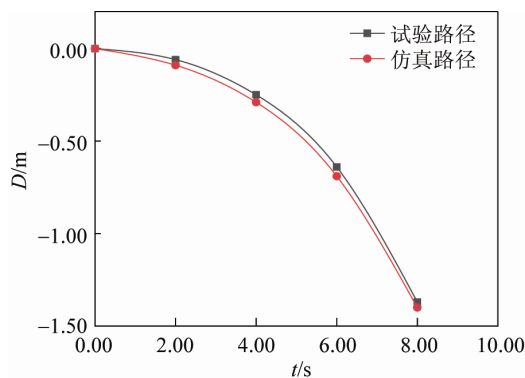
(a) 直线运动轨迹



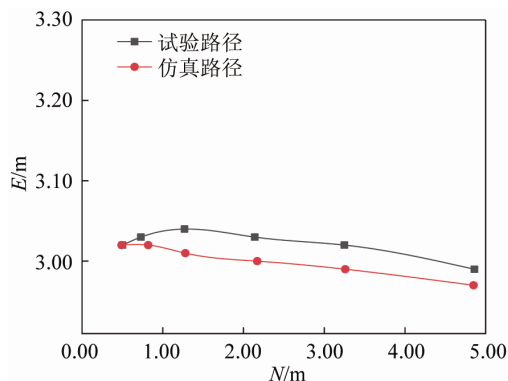
(b) 曲线运动轨迹



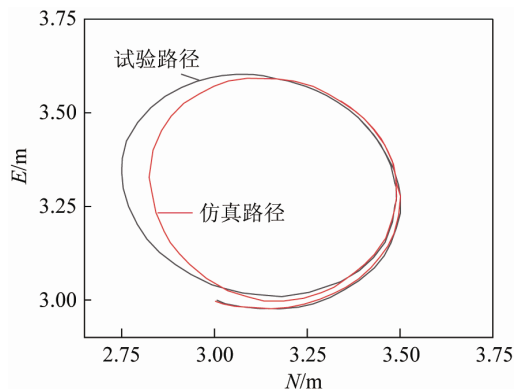
(c) 上浮垂向位移



(d) 下潜垂向位移



(e) 横移轨迹



(f) 原地回转运动轨迹

图9 轨迹对比

Fig. 9 Trajectory contrast

通过仿真试验发现,水下机器人的各种运动与仿真系统中机器人的运动基本吻合,符合预期运动状态,说明建立的具有物理特征的三维仿真环境及机器人三维模型的物理属性均满足仿真需求。

5 结论

(1) 研制了一种水下机器人半实物仿真系统,为机器人控制算法分析和操作人员培训提供实时性的仿真平台。

(2) 设计了仿真系统的布局结构,为平台添加机器人位姿和轨迹的在线可视化功能,为制定机器人控制策略提供帮助。

(3) 开展机器人运动仿真试验,检验了机器人和三维仿真系统对控制指令响应的准确性和同步性,验证了机器人仿真系统的可行性。

参考文献:

- [1] Bogue R. Underwater robots: a review of technologies and applications[J]. *Ind Robot (S0143-991X)*, 2015, 42(3): 186-191.
- [2] Talian P, Perduková D, Fedor P. Universal HIL Test Platform for Mechatronic Systems[C]. *International Conference Mechatronics*. Berlin: Springer, 2017: 338-346.
- [3] 凌家良. 基于 OpenGL 的工业机器人运动仿真软件的设计与实现[D]. 长沙: 中南大学, 2009.
Ling Jiali. The Design and Implementation of Industrial Robot Simulation System based on OpenGL[D]. Changsha: Dissertation Submitted to Central South University, 2009.
- [4] 张冀川, 徐家斌, 童亦斌, 等. 基于 RT-LAB 的 MMC 半实物仿真平台设计[J]. *电力电子技术*, 2016, 50(3): 26-28.
Zhang Jichuan, Xu Jiabin, Tong Yibin, et al. Design of Rapid Control Prototype Platform for MMC Based on RT-LAB[J]. *Power Electronics*, 2016, 50(3): 26-28.
- [5] 邢瑞毅, 马世伟, 钟建鹏. 基于 Simulink 的水下无人机器人建模与仿真[J]. *系统仿真技术*, 2018, 14(2): 109-112, 135.
Xing Ruiyi, Ma Shiwei, Zhong Jianpeng. Modeling and Simulation of Remote Operated Vehicle Based on Simulink[J]. *System Simulation Technology*, 2018, 14(2): 109-112, 135.

- 109-112, 135.
- [6] 甘永, 毛宇峰, 万磊, 等. 水下机器人半实物运动仿真系统的设计[J]. 系统仿真学报, 2006, 18(7): 2026-2029.
Gan Yong, Mao Yufeng, Wan Lei, et al. Design of Semi Physical Motion Simulation System of Underwater Vehicle[J]. Journal of System Simulation, 2006, 18(7): 2026-2029.
- [7] Pérez J, Sales J, Peñalver A, et al. Robotic Manipulation Within the Underwater Mission Planning Context[M]. Berlin: Springer International Publishing, 2015: 595-612.
- [8] Bazzano F, Gentilini F, Lamberti F, et al. Immersive Virtual Reality-Based Simulation to Support the Design of Natural Human-Robot Interfaces for Service Robotic Applications[C]. International Conference on Augmented Reality, Virtual Reality and Computer Graphics. Berlin: Springer International Publishing, 2016: 33-51.
- [9] 吴家鸣, 郁苗, 朱琳琳. 带缆遥控水下机器人水动力数学模型及其回转运动分析[J]. 船舶力学, 2011, 15(8): 827-836.
Wu Jiaming, Yu Miao, Zhu Linlin. A hydrodynamic model for a dynamic analysis of the tethered underwater robot and robot in turning motion[J]. Journal of Ship Mechanics, 2011, 15(8): 827-836.
- [10] 张洪龙, 陈涛, 庄培钦, 等. 基于立体视觉的水下三维测量系统研究[J]. 集成技术, 2018, 7(3): 1-14.
Zhang Honglong, Chen Tao, Zhuang Peiqin, et al. A Binocular Stereo Vision System for Underwater 3D Measurement[J]. Journal of Integration Technology, 2018, 7(3): 1-14.
- [11] 崔培林, 周翟和, 吕品, 等. 自适应误差四元数无迹卡尔曼滤波四旋翼飞行器姿态解算方法[J]. 西安交通大学学报, 2019, 53(3): 97-102,110.
Cui Peilin, Zhou Zhaihe, Lü Pin, et al. Adaptive Error Quaternion Unscented Kalman Filter Algorithm for Quadrotor Attitude Calculation[J]. Journal of Xi'an Jiaotong University, 2019, 53(3): 97-102,110.