

面向海量数据的全船振动分析 管理系统研究

杜德锋¹, 周文进¹, 何江贤¹, 孟凡凯²

(1. 中国人民解放军 91388 部队, 广东 湛江 524002; 2. 海军工程大学, 武汉 430033)

摘要: 目前广泛采用的振动测试系统侧重于数据采集的精度、采样率等指标的提高, 但数据预处理环节不足, 智能化程度不高, 测试人员面对分散数据, 查询对比分析存在很大困难; 为集中有效管理海量预处理数据, 实现数据高性能查询分析, 分别设计了面向查询结果和面向数据解耦关系的两套数据库; 采用 MySQL + Python 架构实现了海量数据背景下全船振动数据的快速查询与分析等功能; 经过多艘船舶的数据处理实践表明, 该系统有效解决了振声数据的综合分析问题, 具有一定的工程实际应用价值。

关键词: 海量数据; 振动噪声; 关系型数据库; 混合编程

Research on Analysis and Management System for Massive Data of Whole Ship Vibration

DU Defeng¹, ZHOU Wenjin¹, HE Jiangxian¹, MENG Fankai²

(1. 91388 Unit of PLA, Zhanjiang 524002, China; 2. Naval University of Engineering, Wuhan 430033, China)

Abstract: At present, the widely used vibration test system focuses on the improvements of data acquisition accuracy, sampling rate and other indicators, but the data preprocessing is insufficient and the degree of intelligence is not high. In the face of scattered data, the tester has great difficulties in query and comparative analysis. In order to effectively manage massive preprocess data and realize the high performance of query and data analysis, two sets of databases that oriented to the query results and data decoupling are designed respectively. The architecture of the MySQL and Python is used to realize the rapid query and data analysis of whole ship vibration under the background of the massive data. The data processing practice of several ships shows that the system effectively solves the comprehensive question of vibration and sound data, which has certain engineering practical application value.

Keywords: massive data; vibration noise; relational database; hybrid programming

0 引言

随着人类对海洋资源的开发利用不断深入, 大型船舶数量井喷式增长。振动的评价与控制是大型船舶日常管理的重要方面。机械设备的振动噪声过大, 不仅影响其使用寿命, 且易对船体结构造成损坏, 还会对船舶的远洋航行安全造成危害^[1-2]。船员长期处于嘈杂的舱室环境中, 也易对身心造成损害, 引发健康问题^[2]。随着振动噪声测试系统的不断发展、对设备振动噪声的认识不断提高, 在整个船舶上布设振动噪声测试系统越来越受到重视。但因大型船舶船体结构庞大、机械设备繁多、管路复杂^[3-5], 对全船的振动噪声测试系统的数据融合性提出了较高要求^[6]。

现行的船舶振动测试标准对机械设备、船体结构、推进装置等振动测试分别做出了具体规定。机械设备的振动

评价需要在机脚、基座、管路、隔振装置、设备与船体连接处上布设振动测点^[7-8]; 船体结构振动评价需要在沿船体方向在船壳上布设振动测点; 船舶推进装置的振动评价需要在动力装置、轴系、以及各种支撑轴承上布设振动测点。安装于船舶不同舱室、隶属于不同类型设备的振动测点, 又因其安装空间位置、隔振层级、以及不同的测试时间与组合工况的不同, 具有复杂的多重属性^[9-14]。可以看出, 全船的振动测试评价数据量巨大、振动测点属性多, 在不同的应用场景下, 一方面需要利用数据冗余形成结构性数据块进行批量分析; 另一方面需要根据测点属性内在的联系, 将测试数据与测点属性解耦剥离开来分别形成易扩展维护的数据模块, 进行深度的数学挖掘。

Brüel & Kjær (BK) 测试系统^[15-16]和 LMS 测试系

收稿日期: 2022-04-13; 修回日期: 2022-05-18。

基金项目: 国家自然科学基金(11974429)。

作者简介: 杜德锋(1983-), 男, 湖北宜昌人, 博士, 工程师, 主要从事振动测试方向的研究。

引用格式: 杜德锋, 周文进, 何江贤, 等. 面向海量数据的全船振动分析管理系统研究[J]. 计算机测量与控制, 2022, 30(10): 202-208.

统^[17-21]都是国际领先、被广泛应用于大型船舶振动噪声测试的主流代表, 均具有一定的智能化管理数据能力, 可以进行初步的过滤和排序, 对于简单的振动测试分析可以胜任。但对于大型船舶海量振动测试, 因设备间复杂的耦合关系、隔振方式的多样性及传递路径的复杂性, 这些系统能够开放给用户的数据管理功能还很有限, 不能实现历史数据工程化的查询与调用, 还无法满足全系统、全周期管理。本文聚焦 BK 振声测试系统的数据分类、可视化及运行效率问题, 为有效控制振声数据颗粒度, 基于 MySQL 关系型数据库和 Python 混合编程^[22], 分别设计面向查询结果的数据库和面向数据解耦的数据库交互平台。着力解决数据颗粒度低、交互性弱、提取效率不足等问题, 为后续海量数据特征的深度挖掘打下基础。

1 振声数据库的设计

MySQL 由瑞典 MySQL AB 公司开发以来, 目前已成长为最流行的关系型数据库管理系统, 有着体积小、速度快、支持跨平台、开源、数据存储量大(支持 TB 级数据)等优点, 可以称得上是目前运行速度最快的 SQL 语言数据库之一。在工程领域得到广泛应用。因此以 MySQL 作为船舶测试系统振声数据库的开发平台。

在大型船舶振声测试中主要以压电加速度传感器来测量旋转机械及船体的振动冲击^[23-24], 以传声器(麦克风)来测量舱室的空气噪声, 这两种测量元件所采集的数据经各种预处理会得到多样的数据类型。

下面以 BK 测试系统为例来阐明整个数据采集及预处理的主要过程: 1) 采集阶段, 按照舱室分布采集前端, 各舱室传感器汇总连接至所在舱室的采集前端(具体数量由该舱室的传感器数量和采集前端的容量决定), 利用网络交换机将各舱室的采集前端汇总至终端采集软件, 并进行前端配置以便数据录取, 每单次采集的数据以单个二进制文件格式存储于硬盘介质上, 每单次采集即是一个工况下一个单程的总数据, 这样在多次不同工况的测试下, 便得到多个二进制文件, 其文件总数等于各工况乘以其单程数并求和; 2) 预处理阶段, 根据振动噪声评价标准需要: 针对压电加速度传感器获取的数据, 需要根据不同的 Reflex 数据处理链对二进制原始数据在振动加速度、振动烈度上进行预处理, 振动加速度预处理数据包含振动加速度功率线谱(FFT)、振动加速度特征线谱(CLS, characteristic line spectrum,)、三分之一倍频程加速度级(CPB)、振动加速度总级(OVERALL)、振动加速度频域提取(RMS); 振动烈度预处理数据包含振动烈度功率线谱(FFT)、振动烈度特征线谱(CLS)、振动烈度频域提取(RMS); 针对传声器获取的数据, 需要根据 Reflex 数据处理链在空气噪声上进行预处理, 空气噪声预处理数据包含空气噪声三分之一倍频程声压级(CPB)、空气噪声 A 计权声压级、空气噪声总声级^[15, 25-27], 以上共计 11 种数据类型。除振动加速

度、烈度的特征线谱数据(CLS)需要根据功率谱(FFT)计算人工干预分析获得以外, 其他 9 种数据类型存储在不同的预处理结果文本中, 各文本文件依靠文件夹系统分类管理, 每个数据文本都有其独特的格式, 如图 1 所示。通常在进行振声数据处理过程中, 以非标准数据格式实现调用, 数据抽取效率低, 给后续的机械振动、船体振声特性、振动能量传递特性的标准化分析带来困难。

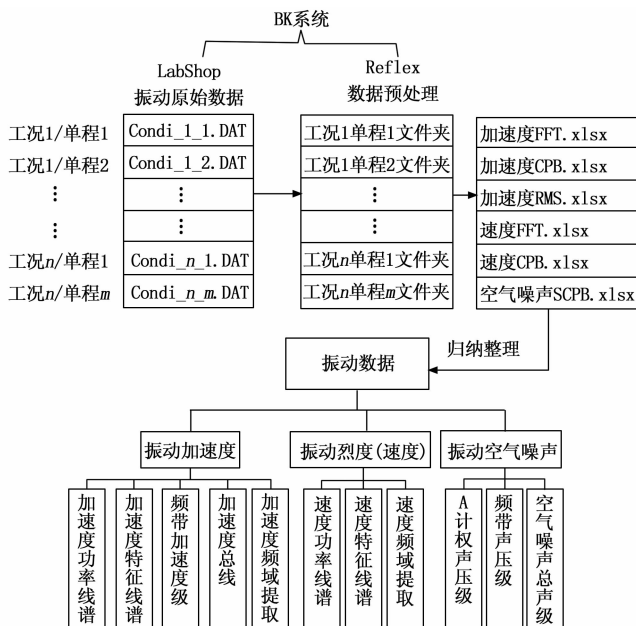


图 1 振动测试系统数据预处理流程框架图

从数据逻辑属性和后续分析来看, 这 11 种数据类型附属于某个工况下某个单程的某个测点上, 记做数据元(工况, 单程, 测点)。当数据元三要素确定了, 数据也就唯一确定。数据元与这 11 种数据类型的对应关系可划分为两类, 一类是一对一关系: 振动加速度总级、振动加速度频域提取、振动烈度频域提取、空气噪声 A 计权声压级、空气噪声总声级; 另一类是一对多关系: 振动加速度功率线谱、振动加速度特征线谱、三分之一倍频程振动加速度级、振动烈度功率线谱、振动烈度特征线谱、三分之一倍频程声压级。

一对一对应是一个数据元确定了一个一维数据; 一对多对应是一个数据元确定了一个二维数据。以某空调系统测试为例, 额定工况下测得该空调左侧机脚测点原始数据、空调机体烈度数据、空调开启时周围空气噪声, 经 BK 处理链分析得 11 种数据类型, 其中一维数据为: 该左侧机脚的振动加速度总级(OVERALL)、振动频域提取(RMS)、机体烈度测点频域提取(RMS)、设备空气噪声 A 声级、总声级; 二维数据为: 该左侧机脚的振动加速度功率谱(FFT)、振动加速度特征谱线(CLS)、振动加速度三分之一倍频程加速度级、机体烈度功率谱、机体烈度特征谱线(CLS)、设备空气噪声三分之一倍频程声压级。

为有效管理数据,本文面向查询结果和数据解耦设计了两种数据库系统。

1.1 面向查询结果的关系型数据库

在多次测试并存储全船振动数据信息时,可将每次测试结果信息解构为三大层面:船舶层面信息、测点属性层面信息、测试数据层面信息。其中船舶层面信息查询变化最少,测点属性层面信息查询变化居中,测试数据层面信息查询变化最大。

船舶层面信息包含:测试船型(posship)(便于同类型船舶横向比较)、测试等级(posstate)(便于船舶修理前后纵向比较),属于一维数据。

测点属性层面信息包含:测点名称(posname)、测点所属工况(poscondition_name)、测点所属单程编号(pos_single-passnumber)、测点所处舱室(poscabin)、测点所处方位(posorientation)、测点所处振动传递级别(posposition_level)、测点所属设备(posequipment)、测点在空间的三维坐标分量(poscoordinate_x、poscoordinate_y、poscoordinate_z)(便于抽取振动空间分布特性),属于一维数据。

测试数据层面信息包含:测点数据 ID(posid)、测点数据单位(posunit)、测试原始数据选取时间起点(posstarttime)、测试原始数据选取时间终点(posendtime)、测试处理对应的文件名称(posfilename)(便于后续复查)、测点数据,其中测点数据就是文中前述的 11 种数据类型,既有一维数据,也有二维数据。

从数据元三要素来看,工况、单程、测点所涉及的外延信息主要是船舶层面信息、测点属性层面信息两部分构成,数据涉及的外延信息主要来自测试数据层面信息。从数据物理属性来看,数据可以分为振动加速度数据、振动烈度(速度)数据、空气噪声数据;从数据元与数据对应关系来看,数据又可分为一维数据(振动加速度 OVER-ALL、振动加速度 RMS、空气噪声 A 计权声压级、空气噪声总声级)、二维数据(振动加速度 FFT、CPB、CLS; 振动烈度 FFT、CLS; 空气噪声三分之一倍频程声压级)。

根据振声数据常规分析方法,结合预处理数据文本各自特点,利用数据的物理属性和维度属性交叉组合,可以建立 9 个自完备的数据关系表,每张表信息都是可以独立检索,不涉及关系表之间的交叉查询。以加速度信息为例,其关系表包含:1) 振动加速度一维总表(其中测点数据分解为两个字段 posoverall、posrms),如表 1 所示;2) 振动加速度功率线谱二维表(其中测点数据分解为两个字段 posx、posy),如表 2 所示。只需要将测点数据分解为对应的字段,同时调整一下测点属性层面的字段,就可以建立其他 7 个数据关系表:3) 振动三分之一倍频程加速度级二维表(其中测点数据分解为两个字段 poscpbx、poscpby);4) 振动加速度特征线谱二维表(其中测点数据分解为两个字段 pos_peakx、pos_peaky);5) 振动烈度功率线谱二维表(其

中测点数据分解为两个字段 posx、posy);6) 振动烈度功率特征线谱二维表(其中测点数据分解为两个字段 pos_peakx、pos_peaky);7) 振动烈度频域提取一维表(其中测点数据字段为 posrms);8) 振动空气噪声一维表(其中测点数据字段可分解为两个字段 pos_a_weighted_level 和 pos_totalsound_level);9) 振动空气噪声三分之一倍频程频带声压级表(其中测点数据分解为两个字段 poscpbx、poscpby)。

表 1 加速度单值总表

字段名称	字段含义	数据类型	约束
posid	测点 ID	int IDENTITY (1,1)	NOT NULL
posname	测点名称	nvarchar(50)	NULL
poscondition_name	测点所属工况名称	nvarchar(50)	NULL
posorientation	测点方位	nvarchar(50)	NULL
pospositon_level	测点级别	nvarchar(50)	NULL
posunit	测点单位	nvarchar(50)	NULL
posrms	频域提取值	numeric(18,10)	NULL
posoverall	总级	numeric(18,10)	NULL
poscabin	测点所属舱室	nchar(10)	NULL
posequipment	测点所属设备	nchar(20)	NULL
posship	测点所属船型	nchar(10)	NULL
posstate	测试级别	nchar(10)	NULL
posstarttime	测点处理起始时间	datetime	NULL
dosendtime	测点处理结束时间	datetime	NULL

表 2 加速度功率线谱表

字段名称	字段含义	数据类型	约束
int	测点 ID	int IDENTITY (1,1)	NOT NULL
posname	测点名称	nvarchar(50)	NULL
poscondition_name	测点所属工况名称	nvarchar(50)	NULL
posorientation	测点方位	nvarchar(50)	NULL
pospositon_level	测点级别	nvarchar(50)	NULL
posunit	测点单位	nvarchar(50)	NULL
posx	加速度线谱频率	smallint	NULL
posy	加速度线谱幅值	numeric(18,10)	NULL
poscabin	测点所属舱室	nchar(10)	NULL
posequipment	测点所属设备	nchar(20)	NULL
posship	测点所属船型	nchar(10)	NULL
posstate	测试级别	nchar(10)	NULL
posstarttime	测点处理起始时间	datetime	NULL
posendtime	测点处理结束时间	datetime	NULL

该架构数据库采用 InnoDB 存储引擎,各表信息是自完备的,没有利用到多表查询,因而存在数据冗余,但是在撰写振动图谱报告时,查询逻辑简单。也正是因为数据冗

余, 所占硬盘空间大, 在实际测试中, 随着测试数据的累积, 数据库越来越大, 查询速度变得越来越慢, 已经不再适合快速给出查询结果。为此需要建立数据解耦的关系表, 将数据元信息与数据剥离开来, 分别建表存储, 利用表的主外键关联, 建立多表查询, 这样一来, 这些与测试数据本身关联的测试变量可以独立维护, 可扩充新的需求变量字段, 数据库灵活性更高。

1.2 面向数据解耦的关系型数据库

在数据分析处理过程中, 所有需要检索的数据条目都建立在测点基础上, 当被测船舶、测试工况、测点 3 个因素明确时, 被检索的数据内容即被锁定。

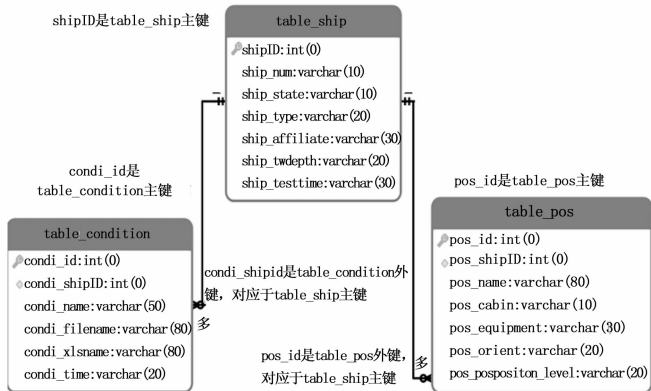


图 2 被测船舶信息表、测试工况信息表、测点信息表三表之间的关系

被测船舶信息表的固有属性字段包含: 对象名称、测试等级 (状态分类)、船舶类型、管理单位、测试水文条件、振声数据采集时间等。测试工况信息表的固有属性字段包含: 工况名称、原始采集数据名称、预处理数据名称、工况实施时间等。测点信息表的固有属性字段包含: 测点名称、测点所属舱室、测点所属设备、测点方位、测点级别等。被测船舶信息表、测试工况信息表、测点信息表三表之间的关系如图所示, 3 个表之间信息是独立的, 用主外键形式将三表固连在一起, 被测船舶信息表中字段 shipID 分别对应测试工况信息表的字段 conди _ shipID 和测点信息表的字段 pos _ shipID。

建立的面向数据解耦的关系型数据库也采用 InnoDB 存储引擎, 所有关系表总计 7 个, 如图 3 所示, 相较于面向查询结果的关系型数据库 (如表 1 和表 2 所示), 数据结构进一步优化、单表字段明显减少、表与表之间的关系更加紧密。测点固有属性字段归纳到 Table _ Ship、Table _ Condition、Table _ Pos 三个关系表中, 通过关系表中主外键约束关系, 把测点数据特征映射到 Table _ singlevalue、Table _ fft、Table _ cpb、Table _ peak 中去, 实现测点名称固有属性和测点数据的分离。

在设计面向查询结果的关系型数据库时, 主要考虑用户查询、分析时, SQL 语言编写简单直观, 因此采用反范

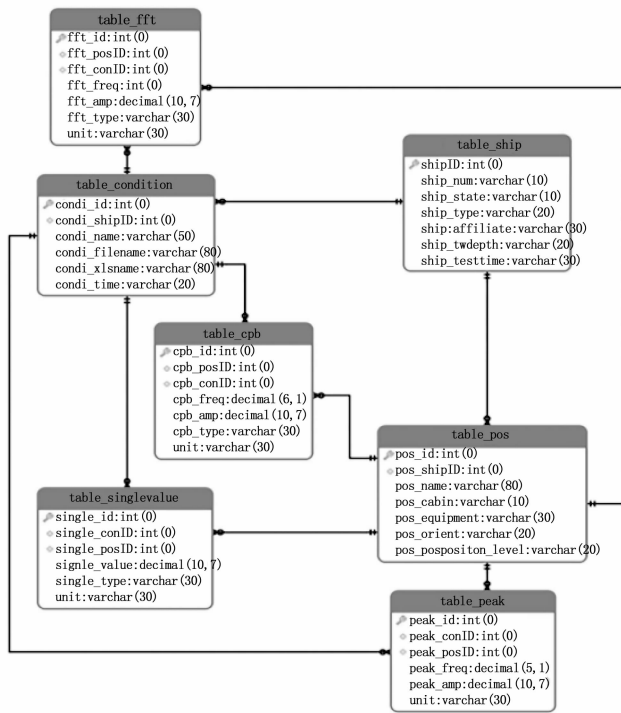


图 3 面向数据解耦的关系型数据库表

式化的 schema, 让参与分析的所有数据都在一张表中, 很好地避免关联, 如果不需要关联, 即使表中没有使用索引, 当数据比内存大时这可能比关联要快得多^[28]。但其弱点是: 在数据入库时要插入大量重复的信息, 入库时间慢, 修改测点信息时需要修改大量数据。为此, 面向数据解耦的关系型数据库采用范式化的 schema, 将重复的信息抽离出来单独建表, 利用表之间的关联查询信息, 数据冗余小, 磁盘占用空间小, 更新数据比反范式要快。但这样存储和查询逻辑复杂, 不按既定规则来存储就会混乱。两种数据库各有优点, 完全的范式化和完全的反范式化都不是必须追求的, 根据用户的需要进行选择。

2 数据入库

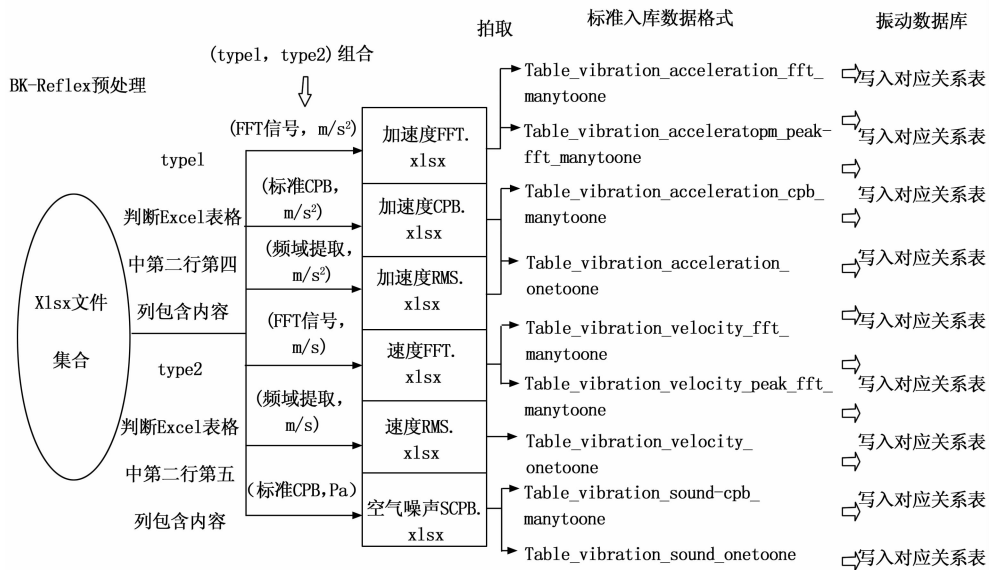
为保证振动数据能高效自动入库, 数据库要同时满足测点名称规范、入库数据格式统一两个要素, 具体步骤如下:

2.1 规范测点名称

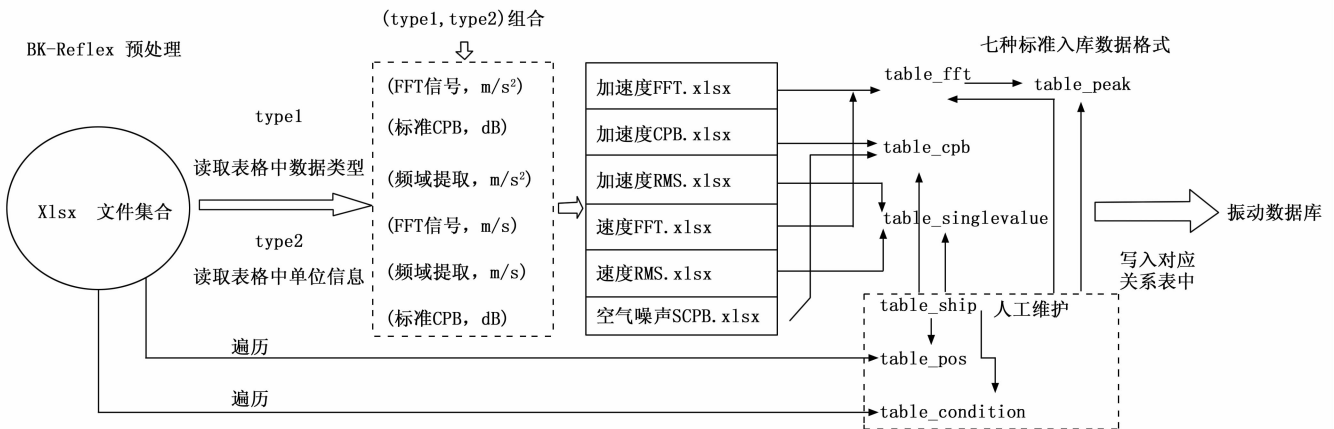
在进行首次测试或增加测点时, 先依次读取各数据类文本文件进行测点名称遍历汇总, 去重得到测点名称集, 进行同一测点的不同名称数据清洗, 确保各测点都有规范唯一的名称, 形成标准测点名称字典, 实现各测点名称变量到名称字典间的标准化映射。

2.2 统一入库数据格式

振声测试系统预处理后的类文本数据文件的通常格式较为单一, 但未能按图 1 中数据类型区分, 比如: 加速度类文本文件包含频带加速度级和总级, 空气噪声声压频带级类文本文件包含频带声压级、A 计权声压级、声压总级。



(a) 面向数据类型的数据入库示意图



(b) 面向数据解耦的数据入库示意图

图 4 数据入库示意图

因此需要从类文本数据格式中抽取特征数据，转化标准格式。针对面向查询结果的关系型数据库，不同类型数据分别隶属不同关系表中，标准格式入库数据分别对应不同关系表，因此数据类型间不存在约束关系，见图 4 (a) 所示。

在以上标准入库数据格式转化基础上，基于 Python 与 MySQL 混合编程^[8]，实现数据的智能入库，利用 Python 数据库包 pandas，将预处理数据按照预先指定的字典格式加载到 DataFrame 后，再使用 DataFrame 属性和方法与 MySQL 进行交互，伪代码如下：

```
import pandas as pd
from sqlalchemy import create_engine
sql_link = 'MySQL+pyMySQL://root:password@localhost:3306/database? charset=utf8'
engine = create_engine(sql_link)
```

循环遍历插入标准入库数据格式

```
table_insert = pd.DataFrame(标准入库数据格式字典)
table_insert.to_sql(name='table_fft', con=engine, if_exists='append', index=False)
```

在 BK 产生的结果文件中，可通过文件中标记位置信息自动识别数据文件类型如图 4 所示。

利用 pandas 库中的数据结构 DataFrame 与关系型数据库进行数据交互。首先将数据按照数据库中关系表形式转化为字典，后写入到数据结构中 DataFrame 去形成数据集。这样数据入库要比传统的单条数据逐一入库效率要高很多。

3 数据快速查询

为实现数据的可视化及快速调用，需要在数据库中进行数据查询。

以加速度功率谱为例，查询"XXX号"船舶，在

“YYYY 年-MM 月-DD 日” 时 “XXX 等级修理前”, 工况 “Condi_01” 下, “01 舱” 舱室内 “1# 空调风机” 设备上 “左前” “机脚” 上振动测点的功率谱。其中, 面向查询结果的关系型数据库的查询语句为:

```
Sql="SELECT posname, posx as 'f/(Hz)', posy as 'a/(m/s2)' FROM Database1.Table_vibration_acceleration_fft_manytoone where posship='XXX 号' and posstate='XXX 等级修理前' and convert(char(10), posstarttime, 120) = 'YYYY-MM-DD' and poscondition_name = 'Condi_01' and poscabin='01 舱' and posequipment = '1# 空调风机' and posorientation = '左前' and pospositon_level = '机脚'"
```

面向数据解耦的数据库的查询 SQL 语句为:

```
SELECT shipID into @shipid from Database2.table_ship where ship_num = 'YYY 号' and ship_state='XXX 等级修理前';
```

```
SELECT condi_id into @conid from Database2.table_condition WHERE condi_shipID = @shipid and condi_name = 'Condi_01' and STR_TO_DATE(condi_time, '%m/%d/%Y') = STR_TO_DATE('YYYY-MM-DD', '%Y-%m-%d');
```

```
SELECT pos_id, pos_name into @posid, @posname from Database2.table_pos WHERE pos_shipID = @shipid and pos_cabin = '01 舱' and pos_equipment = '1# 空调风机' and pos_orient = '左前' and pos_pospositon_level = '机脚';
```

```
SELECT @posname as '测点名称', fft_freq as 'f/(Hz)', fft_amp as 'a/(m/s2)' from Database2.table_fft where fft_posID=@
```

posid and fft_conID=@conid

频带加速度级、频带声压级、总级、频域提取等数据提取方法与上述查询语句类似。

在振动测试中, 常见的分析有设备振动隔振效果计算、设备烈度计算, 以及沿船体的振动分布曲线绘制。隔振效果、烈度可依据计算相关公式进行 SQL 语句的编写, 对于批量设备的隔振效果排序、烈度排序能够快速得出结果。对于沿船体的振动分布曲线, 既可以在数据入库后, 按照物理位置排序的测点集合依次查询提取数据绘图, 也可以在 table_pos 关系表中增加测点三维物理位置字段, 在执行查询时, 任意选取一维坐标进行排序后数据提取绘制, 用来观测船体振动状态, 快速定位异常振动源。

4 实例分析

以某型船舶全船振动噪声分析为例, 该型船舶共分两个型号, 共测得 6 艘该型船舶、9 个条次的振动噪声测试历史数据, 数据量累积达 15Tb。利用建立的上述分析管理系统建立数据库。由于振动数据库中各个振动测点不是严格按照设备、舱室分类依次出现, 无法直接从工况提取数据集进行图谱分析, 不便于进行查阅。在撰写分析报告之前, 应预先给定一个设备测点的顺序, 按照这个顺序依次查询相关数据进行编写, 如图 5 所示。

针对常见的分析所需结果, 例如设备振动隔振效果计

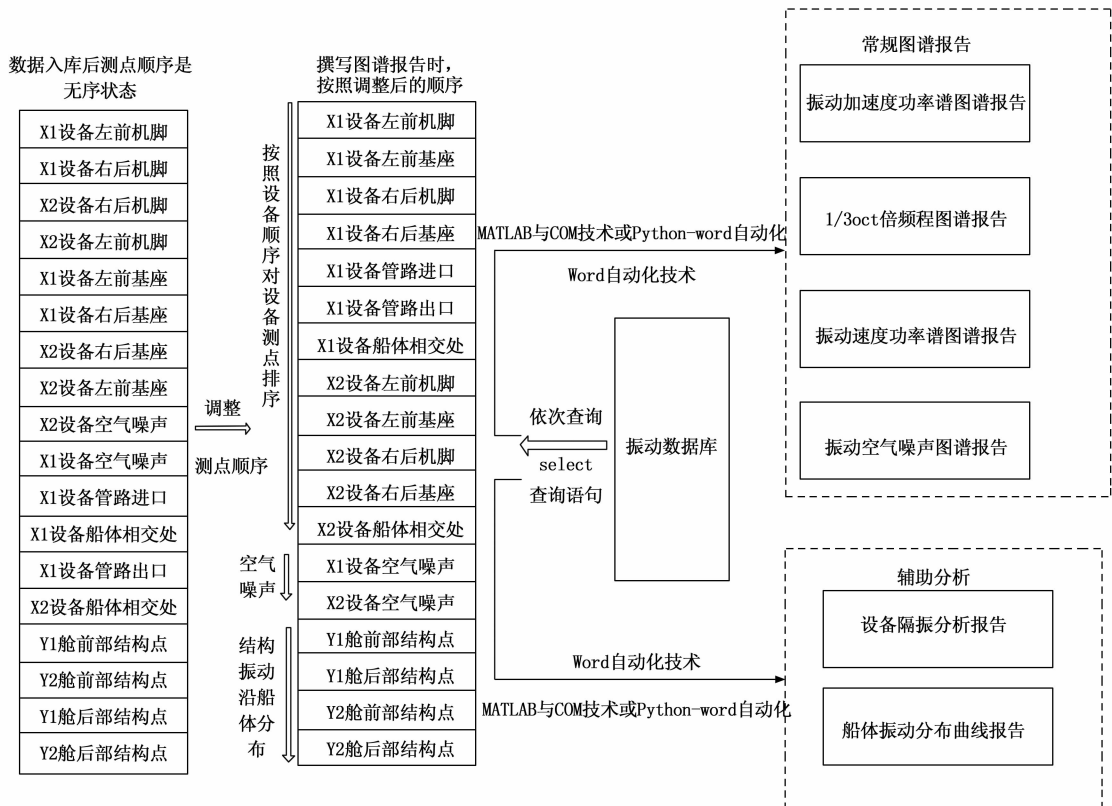


图 5 数据分析报告撰写示意图

算、设备烈度计算,以及沿船体的振动分布曲线绘制等要素,都可以根据上述所建数据库进行数据提取分析。隔振效果、烈度可依据计算相关公式进行 SQL 语句的编写,结果显示:对于批量设备的隔振效果排序、烈度排序能够快速得出结果。对于沿船体的振动分布曲线,既可以在数据入库后,按照物理位置排序的测点集合依次查询提取数据绘图,也可以在 table_pos 关系表中增加测点三维物理位置字段,在执行查询时,任意选取一维坐标进行排序后数据提取绘制,结果显示:该系统可以用来观测船体振动状态,快速定位异常振动源。

5 结束语

本文根据大型船舶振动噪声测试海量数据分析的实际需要,将基于 MySQL 和 BK-Reflex 预处理数据文件,按照直接面向查询结果和数据耦合关系分别设计了两套振动噪声数据库,在 Python 环境下与 MySQL 进行交互,完成了数据库的智能入库、快速查询和批量分析。在此基础上,还可以按照分析意图,进行数据挖掘,利用 SQL 灵活编写查询语句用于辅助分析,辅助发现船体振动状态,快速定位异常振动噪声源。

经过实际的应用检验,两套数据库均能达到预期效果,对比发现:1)面向查询结果的关系型数据库,在数据查询分析时更加友好,sql 语句相对简单、好理解,缺点是数据冗余信息多,占用空间大;2)面向数据解耦的关系型数据库,各表之间交互紧密,附加信息字段容易扩展,因逻辑复杂编写的 sql 语句较复杂,但扩展性好,适合更加复杂的分析,同样的测试数据,数据库所占空间相是前者的 1/10。这两套数据库的优缺点在实际运用也充分展现,可以根据不同需要进行选择。结合故障诊断学,本文的数据库也可为故障机器学习奠定基础。

参考文献:

- [1] 古龙, 闵捷. 船舶振动噪声控制技术的现状与发展 [J]. 舰船科学技术, 2019, 41 (23): 1-5.
- [2] 张福生. 船舶的噪声与振动控制 [J]. 电子制作, 2019 (8): 96-97.
- [3] 柴凯, 楼京俊, 朱石坚, 等. 船舶典型管路系统低噪声设计研究 [J]. 噪声与振动控制, 2021, 41 (2): 156-162.
- [4] 何春雨, 况阳, 李增光. 船舶海水管路系统振动特性计算与分析 [J]. 造船技术, 2020 (3): 61-65.
- [5] 叶栗栗, 王献忠, 吴卫国, 等. 船舶通风空调管路系统的噪声控制 [C] // 昆明: 第十八届船舶水下噪声学术讨论会论文集, 2021.
- [6] 吴武辉, 向阳, 罗文, 等. 复杂机械隔振系统阻抗与传递损失关联性研究 [J]. 噪声与振动控制, 2019, 39 (1): 224-229.
- [7] 胡以怀, 蒋佳炜, 方云虎, 等. 船舶机舱旋转机械的振动测试与分析 [J]. 船舶与海洋工程, 2020, 36 (2): 36-40.
- [8] 岳增峰, 刘近龙. 振动测试技术与工程应用 [J]. 福建建筑, 2021 (9): 159-162.
- [9] 王飞. 主被动混合隔振与可调频动力吸振技术研究 [D]. 北京: 中国舰船研究院, 2018.
- [10] 李志远. 柴油发电机组浮筏隔振装置优化设计与性能研究 [D]. 镇江: 江苏科技大学, 2019.
- [11] 喻雄飞, 邵利民, 徐慧峰. 船舶动力装置整舱隔振技术及其应用分析 [J]. 装备制造技术, 2015 (5): 97-99.
- [12] 柳祺. 船舶浮筏隔振装置建模及减振仿真分析研究 [D]. 武汉: 华中科技大学, 2016.
- [13] 李洪亮, 巴海涛. 船舶轮机振动噪声控制分析 [J]. 船舶物资与市场, 2019 (12): 33-34.
- [14] 殷长春, 张欢, 姚利锋, 等. 船舶推进电机隔振技术研究 [J]. 柴油机, 2018, 40 (6): 40-42.
- [15] 魏来. 基于台架试验的实船设备激励载荷计算方法研究 [D]. 哈尔滨: 哈尔滨工程大学, 2017.
- [16] 杨竞千. BK 振动仪在内燃机车上的应用 [J]. 内燃机车, 1993 (4): 48-49.
- [17] 张振, 楼京俊, 刘锦春, 等. 基于 LMS Test.Lab 的电磁作动器振动测试与分析 [J]. 机电信息, 2015 (30): 124-125.
- [18] 王竞杰, 王书文. 基于 LMS Test.Lab 的电机振动测试与分析 [J]. 农业装备与车辆工程, 2017, 55 (2): 11-16.
- [19] 覃海峰, 李露露, 穆建华, 等. 基于 LMS Test.Lab 的某三缸汽油机增压器油管振动问题优化 [J]. 小型内燃机与车辆技术, 2019, 48 (6): 63-67.
- [20] 李鹏, 辛敏成, 张海涛, 等. 基于 LMS Test.Lab 的振动试验中断处理方法研究 [J]. 环境技术, 2019, 37 (5): 90-94.
- [21] 王若平, 黄杰. 基于 LMS 的怠速状态方向盘振动试验研究 [J]. 郑州大学学报 (工学版), 2016, 37 (3): 83-87.
- [22] 薛建波, 谭凌岚, 罗佳. 基于 Python 混合编程 [J]. 电子测试, 2018 (12): 80-81.
- [23] 陈至坤, 逢鹏, 王福斌, 等. 基于压电加速度传感器的振动测量系统研究 [J]. 计算机测量与控制, 2014, 22 (12): 3923-3925.
- [24] 吕萍, 崔巍, 刘晓晨, 等. 压电加速度计在不同安装方式下的响应分析 [J]. 计算机测量与控制, 2020, 28 (4): 217-221.
- [25] 颜健勇. 考虑多故障特征的滚动轴承故障诊断优化方法研究 [D]. 郑州: 郑州大学, 2021.
- [26] 吕晴. 倍频程分析新方法及其声功率检测应用研究 [D]. 长沙: 湖南大学, 2018.
- [27] 李勇, 侯文浩, 彭丽军, 等. 基于随机噪声的 A 计权 1/3 倍频程分析 [J]. 新型工业化, 2020, 10 (5): 65-66.
- [28] BARON S, ZAITSEV P, TKACBENKO V. 高性能 MySQL [M]. 第 3 版. 宁海元, 周振兴, 彭立勋, 等译. 北京: 电子工业出版社, 2013.