

基于 Web 的摩托车远程诊断系统^①

苏 亮, 李仁旺, 罗 昊

(浙江理工大学 先进制造研究所, 杭州 310018)

摘 要: 针对摩托车市场的现状和问题, 设计并构建了一套远程摩托车故障诊断系统。阐述了该系统的基本框架和诊断流程, 并结合实际开发中所遇到的问题给出技术解决方案, 最后, 分析摩托车的故障类型并针对不同故障给出诊断算法。

关键词: 摩托车; 远程诊断; 故障算法; 串口通信; 网络控制

Motorcycle Web-Based Remote Diagnosis System

SU Liang, LI Ren-Wang, LUO Hao

(Advanced Manufacturing Research Institute, Sci-tech University of Zhejiang, Hangzhou 310018, China)

Abstract: For the motorcycle market situation and problems, we design and build a set of remote fault diagnosis system for motorcycle. The paper describes the basic framework of the system and its diagnostic processes, and combined with practical problems encountered in developing gives technology solutions. Finally, with diagnosis to different failure types of motorcycle, different fault algorithm is given.

Keywords: motorcycle; remote diagnostics; fault algorithm; serial communication, network control

我国是世界摩托车生产量最大的国家, 年产量达一千多万台。而摩托的故障诊断与维修, 作为其售后服务的一个重要组成部分, 在摩托车行业中起着举足轻重的作用。而远程诊断是一种新型的诊断技术, 他是在网络快速发展的基础上发展而来的。在汽车行业中已得到较为广泛的使用, 通用公司已将远程测试与维护系统作为通用汽车的标准配置之一, 丰田、大众、宝马等公司也在研制相应的设备^[1]。而在摩托车行业中, 目前尚没有远程诊断系统投入使用。

本系统的故障检测, 实际是基于飞亚电子的 ECU(电子控制单元)强大的功能基础上实现的。利用串口通信程序与摩托车 ECU 的交互, 根据 ECU 通信的要求发送并获取信息, 并将信息经 Internet 发送至服务器端解析并返回数据。从而实现自动诊断、故障记录上传及远程实时调试与诊断的功能。

1 系统整体结构与流程

如图 1 所示, 系统主要分为两大部分, 客户端、服

务端, 其中, 服务端又包括服务中心端和诊断人员端。系统各部分通过 Internet 网及本地局域网相互连接, 以达到功能要求。



图 1 系统结构

远程调试的过程主要包括自动故障诊断、用户资料解析、车辆运行状态上传及远程实时调试四个步骤, 流程图如图 2。

第一, 自动故障诊断: 客户端读取故障码, 将其

① 基金项目:浙江省杰出青年团队项目(R6080403);浙江省自然科学基金(Z6090572)

收稿时间:2010-07-06;收到修改稿时间:2010-08-17

发到服务器端，利用自动故障检测算法解析故障码并查询数据库返回可能的故障信息及解决方案。

第二，用户资料解析：客户端软件读写串口数据，获取 ECU 编号，并将其通过 Internet 网络传至 F A I 网络中心的服务器端计算机。服务器端根据 ECU 编码查询数据库，而后将相应客户资料取出返回给用户从而完成资料读取。

第三，车辆运行状态上传：客户端将要求客户对车辆进行一定的操作比如轰油门等等并且将记录这些车辆运行数据而后将其上传至服务器，服务器调用专家系统对其进行初步的分析并给与建议并将分析结果、建议及运行记录一起保存，等待调试人员调用以便进行诊断，最后，再由调试人员将故障信息反馈给用户。

第四，远程实时调试：服务器端将客户端的连接转给对应的调试人员端，而后调试人员通过软件远程控制摩托车的部分功能并实时读取运行数据，再利用工具软件进行诊断调试从而实现远程实时诊断的功能。

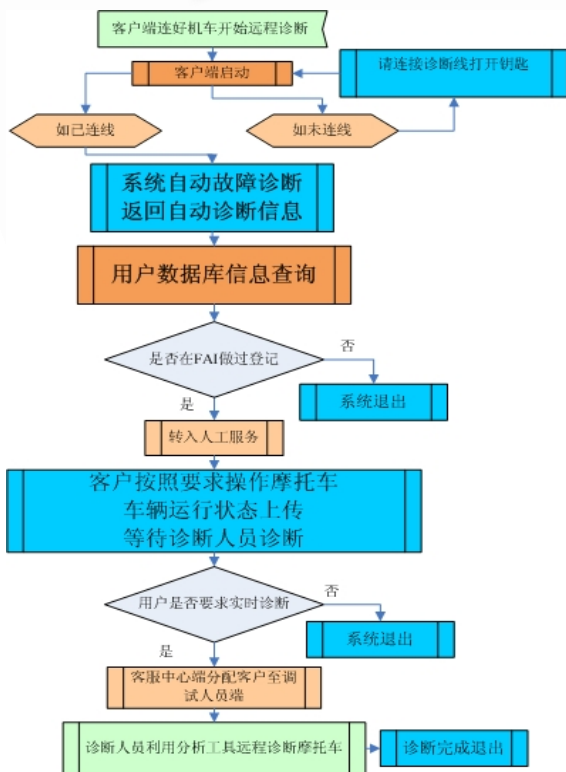


图 2 系统诊断流程

2 客户端系统中的问题及方案

客户端是系统中与客户实现直接交互的部分。在他的设计与实现过程中主要存在两个问题，一个是串

口数据读写，另一个是客户端程序的封装。

2.1 串口数据读写

客户端利用 Java 串口读写包 comm.Jar 及驱动 windll.dll 实现与 ECU 之间的串口读写。读写数据的模式有三种，第一、监听模式，第二、多线程等待模式，第三、多线程循环模式。监听模式，即在类中实现串口监听接口，在每次串口事件发生时激活函数，从而实现数据的读写。多线程等待模式，即启用两个线程，分别实现串口的读和写操作，读线程要运行于写线程之前开始等待，在写线程结束 ECU 返回值之后读取数据并继续运行线程内后续代码。多线程循环模式，同样是利用多线程，不同点在于读线程不会一直等待，而是在一个段代码内循环，定时查看数据是否读入，当有数据读入时跳出循环^[2]。

由于操作系统的的时间片是根据优先级算法分配，而且串口缓存的读写也是和系统相关，因此，不同的读写模式适合不同系统要求。在本系统中，根据系统数据特点，经试验证明第一种模式易产生数据错位与丢失的问题，而第二种模式速度过慢不满足要求，第三种模式才是较为适合的模式。

在调试过程中如何尽量减少数据读写错误的发生是一个非常重要的问题，在控制程序调试过程中，由于上位机程序运行速度较快，当 ECU 正在向上位机返回数据时，若再给 ECU 发出新命令，ECU 便会停止当前数据发送而开始发送新的数据，而这便会导致数据的错位，为了解决这个问题，在写线程的编程上，系统采用了独子模式，即保证整个系统只用一个写线程实例来写数据，并对此写线程之间的最小间隔加以设定，从而防止了此类错误的产生。

虽然系统中已经为防止数据错误做了很多工作，但是，在长时间的运行过程中，数据的错位异常是难以绝对避免的，而部分数据的错误可能会影响后继数据的正确性和连续性甚至导致整个系统卡死，因此系统设计了一套数据验证函数及相应的错误处理机制。

调试过程中，数据是一组为单位进行传输和交互的，而组与组之间的间隔都是毫秒级的，而同时系统要求对每一组的数据都进行验证，因此在验证函数的选择上既要考虑到验证成功率又要考虑到验证函数的时间消耗。经试验给出满足系统要求的验证函数如下：

$$\left\{ \begin{array}{l} Date(n/2) = \sum_{i=0}^{n/2} Date(i) \% 16 \\ Date(n) = Date(n/2) + \sum_{i=n/2}^n Date(i) \% 16 \end{array} \right.$$

对于总长为 n 位的数据状态码, 设置第 n 位与第 $n/2$ 位为校验位, 若第 $n/2$ 的值等于他前面所有值之和对 16 的余数, 同时第 n 位的值等于 $n/2$ 的值加上从 $n/2$ 到 n 所有值之和对 16 的余数, 则认为此组数据符合验证要求。若不满足则引入错误处理程序, 即取验证正确的上一组状态数据及本组操作命令, 将他们导入理论模拟函数中, 而后返回一个理论上的正确返回值, 并将数组错误位至 1, 同时, 将串口缓存中多余数据取出抛弃。通过这种方法, 对数据完成了验证, 防止了单组数据异常对整个调试过程连续性及稳定性的破坏, 保证了系统稳定。

为了方便笔记本电脑的使用, 我们还利用 CP2102 芯片开发了 USB 口的数据线, 并将其虚拟为串口以实现摩托车接口与电脑 USB 接口的通信^[3]。

2.2 客户端程序的封装

JRE 是 Java 程序运行的环境, 其最大的特点就是平台无关性, 无论是 windows、Linux 的操作系统不同还是手机、计算机、PDA 等的设备不同, 只要备有合适的 JRE, Java 程序都能顺利运行^[4]。

由于客户端环境差异较大为了更好的适应这些差异, 本系统的客户端采用系统自带 JRE 的方式, 首先根据软件功能优化定制 JRE, 而后用 FatJar 和 exe4j 对程序进行封装并与此优化过的 JRE 相关联, 从而达到平台无关性的要求。通过这种方式为将来客户端平台向手机、PDA 等设备上的延伸做好了准备。

3 服务端系统中的问题及方案

系统服务端主要包括两个部分, 一个是服务中心端, 一个是服务诊断人员端。中心端位于 FAI 网络内部, 常年处于开启状态, 采用虚拟服务器方式随时处理用户的连入。在功能上, 他要求为每一个用户连接创建一个单独线程以实现系统功能, 并且要求系统可以与公司内部数据库相连。为了更好的控制线程系统采用了线程池技术, 通过线程池来实现对线程的分配、开启和关闭。而在数据库连接方面使用了 Hibernate 构架读写数据库, 实现数据永久化要求。

服务诊断人员端是远程实时诊断中诊断人员使用

的部分。他包括数据接收端及数据分析工具集。他可以位于 FAI 网络内部, 也可以位于其他任何可以联网的地方, 通过这种方式, 诊断人员实现了坐在家也可以完成诊断工作的目标。

对于摩托车运行状态的分析是一个比较复杂的过程, 系统中服务诊断人员端接收数据并将其进行格式转化, 通过工具集中各种软件协助对车辆运行状态及故障情况加以诊断, 分析效果图如下:



图 3 运行分析效果图

4 故障诊断算法

摩托车主要有两种故障类型, 一种是传感器可接受到的故障, 一种是摩托车整体运行中才可发现的故障。第一种故障由传感器检测并保存于 ECU 故障码记录中, 而第二种故障无法用故障传感器直接检测到, 只能从摩托车的整车运行状态记录中去判断。

对于第一种故障, 本系统在自动故障检测部分给予了检测和解析, 客户端将已在 ECU 中记录的故障码发至服务器端解析并返回故障信息和解决方法^[1]。

整个故障码是由多个单元组成的, 每个单元分别记录着不同传感器的异常记录, 多数情况下, 一个故障码可以和一个具体的硬件故障相对应, 但是在有些时候一个传感器所能测到的数据异常即一段错误码并不一定就能唯一的指出故障所在, 例如, 尾气排放超标即尾气检测传感器报告异常, 造成此异常的原因可能是排气管内催化剂老化, 也有可能是电喷喷油过量等, 面对这种情况, 如何给予客户一个更加合理的故障信息呢? 本系统采用了下面的算法给予判定。

以尾气排放超标为例, 此尾气检测传感器超程引起的是导致报告此故障的原因, 而已知当前有排气管催化剂老化、动力喷嘴损坏及控制单元 ECU 错误这三种部件的损坏会导致此类异常。同时, 这些部件的损坏不仅只会引起一个传感器的数值变化, 也会是其他传感器发生变化, 如果变化偏移量超过一定范围就会

报告异常。例如，动力喷嘴如果出现问题导致喷油量出现偏差，每个单位的偏差将会导致油量传感器产生 3 个单位的偏差即 $a_1=3$ ，使氧气浓度传感器产生 2 个单位的偏差 $a_2=2$ ，使尾气检测传感器产生 4 个单位的偏差 $a_3=4$ ^[5]。同理，排气管催化剂老化和控制单元 ECU 错误也会引起传感器偏差，因此得可得到，故障与信号的关联因数矩阵：

$$A = \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} = \begin{pmatrix} 3 & 2 & 4 \\ 1 & 2 & 2 \\ 2 & 3 & 3 \end{pmatrix}$$

在没有故障下的传感器的取值称为基准值。

有氧气浓度传感器、油量传感器和尾气传感器基准值向量为：

$$\alpha_{ji} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 5 \end{pmatrix}$$

实际传感器取值称为测量值。

氧气浓度传感器、油量传感器和尾气传感器测量值向量为：

$$\beta_{ce} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 8 \end{pmatrix}$$

部件偏差向量为：

$$A \cdot \varphi = \beta_{ce} - \alpha_{ji}$$

$$\varphi = A^{-1} \cdot (\beta_{ce} - \alpha_{ji})$$

$$= \begin{pmatrix} 0 & -3 & 2 \\ -1/2 & -1/2 & 1 \\ 1/2 & 5/2 & -2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 3/2 \\ -1/2 \end{pmatrix}$$

向量中的值便为各部件产生偏差的大小，取偏差值超过额定范围的部件为故障部件。通过这种方法系

统实现了对于会产生多个传感器异常的故障的判别诊断^[6]。

而对于第二种故障，并没有那个传感器能够检测到异常，因此故障码中就没有适当的记录，例如，达到最佳油氧燃烧比例所需经过的时间等等。对于这类故障主要依靠检测技术人员对运行数据进行分析的方法加以诊断。为了协助诊断人员的工作，系统在车辆运行状态上传的诊断流程中引入了辅助决策系统^[7]。

由于上传检测中，用户需要按照指示来操作摩托车，系统将依据操作将特定的运行记录上传服务器端，而服务器端将会将记录解析为多条曲线，并分别取这些曲线与他们相对应的并早已存储于数据库中的典型故障曲线相匹配，批判函数如下：

$$\text{相似度 } P = \sum_{x=0}^n (y_{ji} - y_{ce})^2$$

系统将匹配度较高的前几个典型曲线信息反馈给诊断人员协助他们做出诊断判定^[8]。

5 结语

服务赢得客户，客户服务体系在日后的产品竞争中将起到越来越重要的作用，远程诊断作为智能客户的核心内容之一，也将受到更多企业的重视。下阶段，将在手机客户端的开发中，加大力度尽快完善手机部分功能调试，实现连上手机就能控制和诊断摩托车的目标。

参考文献

- 1 邹平吉. 基于 WEB 的远程控制系统市场前景分析. 信息技术, 2009, 4: 31-33.
- 2 陈庆, 李鹏飞, 姚彪国. 基于 Java 技术的远程控制实验系统开发. 西安工程大学学报, 2008, 22(1): 88-91.