信息技术与系统

智能显示器语音识别软件的自动化测试环境

钟　理，刘布麒，杨　颖
（中车株洲所电气技术与材料工程研究院，湖南 株洲　412001）

摘　要：智能显示器语音识别软件的软件测试需要组合多种外部环境噪声、语速、语调的指令语音，造成测试用例的组合数量巨大且执行成本高昂。对此，文章提出一种利用脚本自动建立多种要素组合的指令语音并同步播放、记录并判断识别结果的测试环境，介绍了测试环境的系统组成及核心功能，描述了测试环境的关键技术，包括多线程语音合成和相应时间性能的测试方法，最后列出了单个测试脚本的执行序列。该自动测试环境能覆盖所有语音指令的多种语音要素组合，兼顾环境噪声的测试需要，且扩展性与可重复性好。

关键词：语音识别；软件测试；测试环境；语音自动合成；Python；智能显示器

中图分类号：TP391　　**文献标识码：**A　　**文章编号：**2096–5427(2020)04–0035–04

doi:10.13889/j.issn.2096–5427.2020.04.007

Automatic Test Environment of Speech Recognition Software for Intelligent Display

ZHONG Li, LIU Buqi, YANG Ying
(CRRC ZIC Research Institute of Electrical Technology & Material Engineering, Zhuzhou, Hunan 412001, China)

Abstract: Software test of smart display speech recognition software involves a combination of multiple external environmental noise, speed of speech, and intonation of instruction voice, which causes a large number of test case combinations and high execution costs.This paper proposed a test environment which uses a script to automatically create and simultaneously play instruction voices with various types of elements combination, record and judge the recognition results. Core functions of the test environment are introduced and the system composition is described. Key technologies of the test environment are described, including multi-threaded speech synthesis and corresponding time performance test, and the execution sequence of a single test script is finally listed. The test environment can cover multiple combinations of speech elements of all voice instructions, and take into account the test requirements of environmental noise, good scalability and repeatability.

Keywords: speech recognition; software test; test environment; automatic speech synthesis; Python; intelligent display

0　引言

在智轨电车司机室使用的智能状态显示器，除提供基本的车辆控制状态显示与司机操控功能外，还具有智能语音处理与智能图像处理功能，支持语音识别、语音播报等人机交互方式[1]。在智能显示器的软件开发过程中，需对软件功能进行测试。根据标准 GB/T 21023-2007《中文语音识别系统通用技术规范》的要求，在语音识别系统的测试工作中，应采用基于语音识别标准库或者基于现场口呼的测试方法，以确保语音输入的语速、语调等有多种组合覆盖[2]。在对智轨电车智能显示器进行语音识别测试时，除了使用符合标准 GB/T 21023-2007 要求的语音指令进行测试以外，还需要考虑交通工具应用环境中各种现场噪声的影响[3]；对于测试过程中出现的问题，也需要进行全部用例的回归。

传统的测试软件库中没有针对智能显示器测试所需的现场噪声信号配置，需要建立新的测试语音组合，资源开销大且组合的可扩展性不足；口呼测试的方式可以

收稿日期：2019–08–28
作者简介：钟理（1981—），男，工程师，主要研究方向为软件测试技术与软件测试环境。

评估软件对口令的反应，但因为对应噪声的发生时间有先后次序，复现噪声环境中测试出现的问题有一定难度。鉴于此，针对显示器语音识别的技术特点，本文提出一种基于从文本到语言技术（text to speech,TTS）的语音识别自动化软件测试环境的具体实现方法，利用脚本自动建立多种要素组合的语音指令并同步播放，自动对识别结果进行评估。该测试环境可扩展性好，可以灵活扩充语音指令以及外部噪声，并配置噪声与指令的时序，且只需对测试配置与测试脚本稍加修改便可用于同类语音识别软件测试，具有一定通用性。

1 系统方案

1.1 系统核心功能

测试环境的系统核心功能包括语音指令合成、测试脚本框架及测试用例管理。

语音指令合成主要是根据标准 GB/T 21023-2007 的要求以及智能显示器的语音指令集，合成各种符合测试要求的语音指令，满足被测试智能显示器的测试用例输入要求。

测试脚本框架主要提供测试脚本自动化运行的支持，包括脚本的运行环境和操作工控机、外部程控电源接口、结果记录等。其不仅提供了测试用例输入的标准模式，同时也支持测试结果的判断，其使用为测试用例管理也提供了条件。

测试用例管理利用测试脚本来执行。测试用例在脚本中使用函数的形式进行维护并在函数中记录测试用例的输入，通过调用测试框架中提供的计时、语音合成及硬件端口管理等功能来实现测试用例管理，记录和分析测试结果，并将失败的测试用例统一与缺陷管理工作相关联。

1.2 测试环境系统组成

测试环境系统主要由硬件和软件两部分组成。硬件部分主要用于声音信号的播放，同时与被测软件通信；软件部分主要用于指令声音信号的合成与测试用例调度，并对识别结果进行判断。

1.2.1 系统硬件组成

在智能显示器语音测试过程中，驱动软件语音识别功能的数据主要是语音信息，而测试结果主要通过串口输出，因此测试环境的数据输入和输出需要与之匹配[4]。测试环境的数据输出使用扬声器来发送语音信号，串口通信主机与显示器间采用串口通信以满足测试环境的信息输入与输出需求。考虑到语音测试自动化的需要，系统硬件主要分为语音播放部分、工控机、通信部分和控制电源（图 1）。

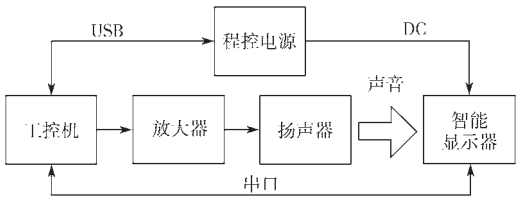

图 1 语音识别自动化测试环境硬件连接图

Fig. 1 Hardware connection diagram of speech recognition automated test environment

语音播放硬件主要包含 USB 声卡扩展、放大器和扬声器，用于实现测试工作的发声需求。

工控机用于提供测试脚本的运行环境，支持测试用例的自动化执行，进行测试管理相关工作。

通信部分用于建立被测件与工控机之间、控制电源与工控机之间的联系通道，包括 USB 连接方式以及串口连接方式，便于工控机与被测显示器、程控电源的通信以及结果的检查。

控制电源用于给被测显示器供电，利用 USB 接口进行通信，使用 VISA 协议控制设备状态。

1.2.2 测试环境软件架构

测试环境软件架构分为 3 层，分别是系统层、中间层和应用层（图 2），其中系统层与硬件直接关联。

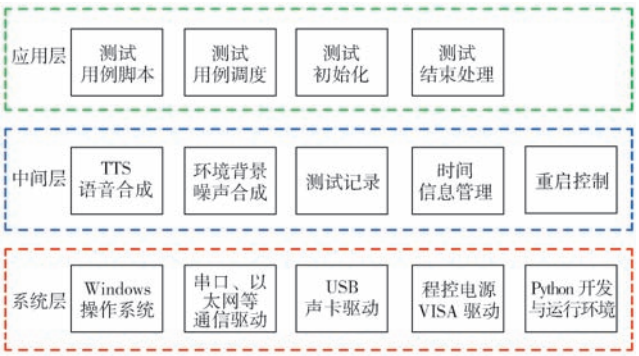

图 2 自动化测试软件架构

Fig. 2 Software structure diagram of the automatic test environment

系统层为测试工作开发与运行的基础，其包括工控机的操作系统软件、Python 脚本的开发与运行环境、与工控机直接关联的驱动硬件。中间层即测试框架，用于实现本测试环境所需的关键功能，其包含语音合成、噪声合成、时间信息管理及测试记录等模块。应用层主要是测试用例，为具体的测试用例实现模块，还有测试用例调度、测试初始化及测试结果数据处理等模块。应用层调用中间层的所有功能，与其进行数据交换；同时，中间层与系统层进行数据交换，驱动系统层执行对外指令输出与读取外部数据。

1.3 测试流程

测试过程中，先进行基本的语音指令测试，检查软件能否对每个指令做出正确的反应；接着对所有指令设置不同的语调、语速的组合，利用穷举法检查软件识别

的正确率以及响应时间；最后使用等价类划分法来确定典型的环境噪声并进行环境噪声合成指令的语音识别测试，检查识别正确率。在测试过程中，先记录软件的识别时间，最后分析时间性能。在组合各要素进行测试的过程中使用自动化测试手段，确保覆盖所有的指令要素；如果发现缺陷，软件进行修复以后，将所有的测试用例再执行一次。

2　关键技术

2.1　多线程语音合成

传统的语音识别测试环境使用语音库形式，配合不同的命令词、语速及语调的组合，数据量非常庞大，建设成本高昂且不易管理。在本测试环境中，在 Python 脚本中调用 Pyttsx 库来实现语音的合成。

Pyttsx 是一套基于 SAPI5 文语合成引擎的 Python 封装库。SAPI 全称为“The Microsoft Speech API”，是微软公司推出的语音应用编程接口。Pyttsx 支持将 txt 文本转化为语音在线播放，在播放语音时还支持设置发音的语速、语调及音量。在本测试环境中，利用 Pyttsx 的语音合成能力，将语音控制指令文本转化为语音命令进行播放。

测试用例设计时，为满足测试过程中语音指令应具备不同语速以及语调的要求，需借助 Pyttsx 的语速和语调设置能力。普通人发声的标准语速约为 150 字 /min，根据等价类划分的原则设计不同语速的等价类，包括 100 字 /min、125 字 /min、150 字 /min、175 字 /min 及 200 字 /min，对此 Pyttsx 都能够通过设置 rate 属性的具体数值来进行支持。语调方面，Pyttsx 支持通过设置 voices 属性，调用系统中安装了发音人声音，以切换不同的语调效果，包括男声和女声等。考虑到使用场合的特点，发音人没有使用童声效果。图 3 示出语音指令的音频波形，图中采样信号强度为其转换为数字量后的数值（如采样精度为 16 bit 时，其信号强度幅值为 2^{16}=65 536，即范围为 –32 768 ~ 32 768）。

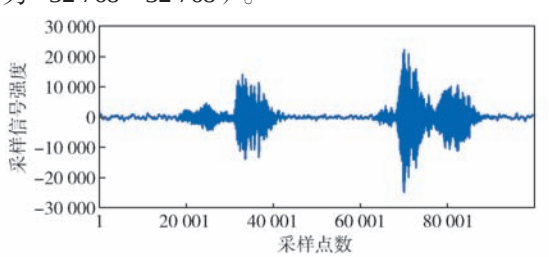

图 3　语音指令的音频波形

Fig. 3　Audio waveforms of voice command

测试过程中，外部噪声干扰对语音识别的影响也是测试设计时需考虑的重要因素之一。根据智能显示器的应用环境，测试工程师设计了常见现场噪声的等价类作为噪声影响的测试依据，主要包括手机在司机台上的来电振动声、鸣笛声、风噪声（20 km/h，60 km/h 及 100 km/h 风速下的）及雨点敲击玻璃的声音。测试工程师在现场录制了这几种声音素材，其为双声道，采样率为 44.1 kHz，采样精度为 16 bit，保存格式为 WAV。图 4 示出语音指令与风噪声同时播放时的音频波形。

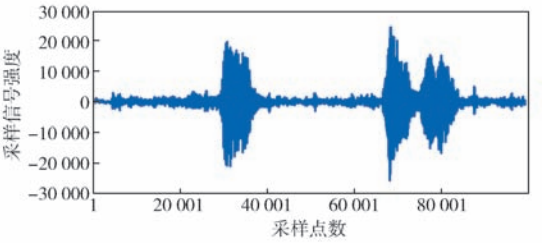

图 4　语音指令与风噪声同时播放时的音频波形

Fig. 4　Audio waveforms of voice command and noise playing simultaneously

在本测试环境中，应用了 Python 的多线程支持特性来进行语音指令与外部噪声的音效合成。测试脚本中设计了专门的背景噪声播放线程，可以控制其声音大小以达到信噪比的要求。当需要叠加时，启动背景噪声播放线程，即同时播放出语音指令和背景噪声，能够较好地达到测试语音识别功能的要求，且可重复性与可控性好，易于复现故障状态。

2.2　响应时间性能

根据标准 GB/T 21023-2007 的要求，系统响应时间使用实时系数来进行性能衡量。实时系数 r 定义如下：

$$r = (T_r - T_s) / (T_e - T_s) \tag{1}$$

式中：T_s——发音起始时间；T_e——发音结束时间；T_r——识别结束时间。

在本测试环境中，T_s 及 T_e 利用在测试脚本的时间管理线程中插入的时间戳进行时间点记录；T_r 是利用智能显示器上软件发出识别结果来判断的，该识别结果通过串口传输到测试环境中的工控机，当工控机接收到串口传输数据时进行时间点记录。串行通信传输线长度为 1 m，传输波特率为 9 600 kb/s，识别结果的数据长度为 4 字节且在实验室环境下受干扰的影响很小，因此可以近似地认为传输延时 T_t 为微秒级，与语音识别功能所需的秒级时间相比，其可以被忽略。在测试过程中，同一个用例需要反复执行多次，对计时结果取平均值作为测试结果 [5]。

根据测试环境调试过程中的经验，语音识别存在失败的可能，其会导致没有反馈信息，因此在时间管理线程中设计了超时保护机制，以防止测试脚本因一直等待而发生死锁；也有可能在没有语音指令时发生误报现象，这些都需要独立的串口通信线程进行通信结果的判

断与保护，防止影响正常测试过程的数据的产生。

2.3 单个测试用例的执行

测试用例执行过程中，TTS 语音合成、测试结果的判断与记录、线程调度等主要功能都利用测试用例函数实现（图 5）。测试用例脚本执行过程中，共有 3 个线程并行工作，分别是串口通信线程、时间管理线程和背景噪声播放线程。串口通信线程主要工作是监视串口接收缓冲区，接收到数据的时候，通知时间管理线程语音识别结束并保存记录的数据供测试用例函数查询。时间管理线程用于记录语音播放的开始时间以及结束时间，便于测试用例函数计算时间性能；同时进行超时检测，一旦发生超时，立刻通知关闭串口通信线程。根据测试用例设计，一部分用例需要使用背景噪声，在这些用例执行时，测试用例函数启动语音播放功能之前会启动噪声播放线程，同步播放选定的噪声信号，以确保合成所需要的音效。

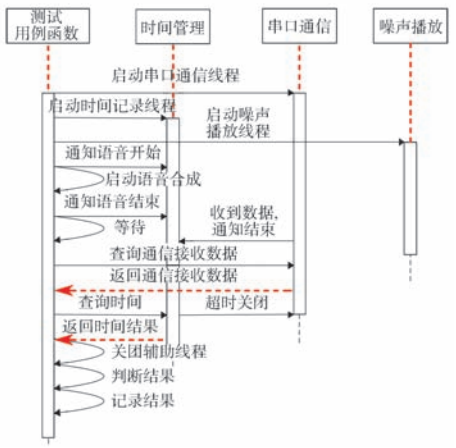

图 5 单个测试用例执行过程的序列图

Fig. 5 Sequence diagram of single testcase execution process

测试用例函数的主要工作步骤如下：

(1) 测试用例函数完成自身数据初始化后，首先启动串口通信线程，然后启动时间管理线程，根据测试脚本的设计来启动噪声生成线程。

(2) 各线程启动完毕后，测试用例函数并通知时间管理线程语音播放开始；启动测试用例函数的语音 TTS 功能，实时生成语音指令并进行播放，然后通知时间管理线程语音播放结束。

(3) 等待一段时间后，查询通信结果与计时信息；关闭时间管理进程，通过比对测试用例的语音指令输入与返回结果是否一致，来判断测试结果是否通过，并对测试结果进行记录。

3 应用情况

为了评估语音合成的效果，对合成语音指令与口呼语音指令音频进行了比对分析。分析过程中，首先对指令音频文件进行频域变换，提取语音指令的梅尔倒谱参数（MFCC）特征[6]，然后利用隐性马尔可夫模型（HMM）机器学习模型对口呼指令的特征值进行训练[7]，最后计算合成指令特征在训练集中的匹配程度，作为两者符合情况的分析结果。该评估方法基于人的听觉感受，通过一系列的频域和时频域分析来测量合成语音指令与口呼指令在听觉上的相似程度。此处使用语音合成的 3 个指令进行比较，对比口呼的相应指令的符合程度，同时加入了一个口呼指令作为参考，比较结果见表 1。表中的数值为对 HMM 模型计算的合成指令与口呼指令匹配概率进行对数换算后得到的结果，因此数值为负数。第一行中输入指令合成 A 与口呼指令 A 的匹配识别结果数值最大，表示两者匹配概率最大；同样的，指令合成 B/C 与对应口呼指令的匹配识别结果数值最大，高于其他口呼指令的匹配识别结果，说明合成指令与口呼指令一致性好[3]。

表 1 合成指令与口呼指令识别结果

Tab.1 Recognition results of synthetic command and spoken command

输入指令	口呼语音指令模型			
	指令 A	指令 B	指令 C	指令 D
合成 A	-32 646	-33 666	-33 204	-37 136
合成 B	-24 846	-23 743	-24 327	-26 217
合成 C	-32 233	-31 247	-30 079	-32 668

本测试环境在智轨电车智能显示器语音识别测试中已经得到了成功的应用，利用 Python 语言编写自动化测试脚本，实现了测试用例的设计意图。测试脚本覆盖了所有语音指令，并在每条指令执行过程中使用了 5 种语速、3 种语调、4 种噪声、3 种信噪比的组合模式，同时执行了指令的近音词语音生成与判断，完成了指令的多种语音要素的遍历；同时，利用脚本的特性，在测试设计过程中引入了随机测试的概念，主要测试语音指令发出时噪声随机发生时间对指令识别的影响。

经过测试后，被测软件的可靠性得到提高，功能、性能得到验证。传统的语音识别测试方法为口呼或者录音，口呼全部指令以及组合各种语素需 3 个人花费约 60 h 时间，且存在口呼时语速精度不高的问题。与口呼相比，本测试环境利用脚本批量生成标准的指令，在同样覆盖率的情况下，一个轮次测试人工时间成本降低 30% 左右；在重复执行用例以及回归测试过程中，脚本可以自动执行并判断结果，时间成本的降低更加明显。

4 结语

本文提出了一种用于语音识别软件的自动化测试环境，其满足标准 GB/T 21023-2007 的测试要求，使语音识别软件测试更加完善。（下转第 43 页）