Advanced Materials Research Vol. 980 (2014) pp 165-171
© (2014) Trans Tech Publications, Switzerland
doi:10.4028/www.scientific.net/AMR.980.165

Submitted: 2014-05-14
Accepted: 2014-05-14
Online: 2014-06-30

Implementation of a Voice-ControlSystem for Issuing Commands in a Virtual ManufacturingSimulation Process

Yun SuenPAI[1,a], Hwa JenYAP[2,b], and S. RAMESH[3,c]

[1,2,3]Department of Mechanical Engineering, Faculty of Engineering,

University of Malaya, 50603 Kuala Lumpur, Malaysia

[a]yspai1412@gmail.com, [b]hjyap737@um.edu.my, [c]ramesh79@um.edu.my

Key words: Speech Recognition, Voice Control, Virtual Manufacturing, Robotic Work Cell

Abstract: Speech recognition is a technology that attempts to involve audio cues during interaction with machines, instead of being limited to just visual and touch interfaces. However, a keyboard and mouse input is an archaic method of interaction, adding on to the fact that voice control is seemingly more natural. This study aims to implement speech recognition as a form of machine control to perform simple commands in a virtual simulation process. The simulation system is an in-house developed augmented reality robotic work cell which includes a robot arm, a conveyer belt, a computer numerical control (CNC) machine, and a pellet. Issuing commands are performed via the Windows Speech Recognition software built from the Microsoft Speech Application Programming Interface (SAPI). This software is advantageous because it can be fairly accurate once trained properly, is easily modifiable by anyone regardless of the operator's programming knowledge, and is free. A macros tool is used to support the additional features of the recognition software which includes directly programmable Extensible Markup Language (XML) codes.

Introduction

Voice recognition is a technology that plays a part in contributing towards a more efficient, flexible, expressive, and transparent means of human-machine interaction. It pushes the boundaries of interactivity to achieve a sense of immersion unlike any other, which is vital in today's trend of technological approach which favors wearable technology and computing. However, the technology still requires further refinements before it is able to completely replace the more traditional input methods of a keyboard and mouse. Precision is the main source of concern for speech recognition, as the human voice comes in a varying degree of intonation, language, and interpretation. Therefore, the main objective of this study is to apply speech recognition into virtual manufacturing as a form of control by using it as a replacement in issuing fundamental commands, instead of quantifiable functions like inputting depth of cut values, the final coordinate in inverse kinematics, and so on. This limitation ensures that voice control will not interfere with operations that require high precision, yet is able to deliver a higher sense of immersion to the user. Once issuing commands start to feel natural, a simulation operation can reduce the time required to perform them.

Related Studies

The concept of a natural human interface was first coined by Bill Gates, as he predicted that voice recognition would be a key technology in replacing the traditional input methods, alongside touch and vision-based systems[1]. Voice data goes through pretreatment, feature extraction and finally recognition algorithms in its processing cycle. Pretreatment is usually conducted behind the scenes to filter the noise. Algorithms like Mel Frequency Cepstrum Coefficients (MFCC), linear predictive coding (LPC), and Fast Fourier Transformation (FFT) fall under feature extraction. Finally, the Hidden Markov Models (HMM) creates the recognition system by calculating the output probability. Fractional Fourier transform (FrFT) which is a based on FFT has been used to extract features for signal processing [2]. In terms of efficiency, it is similar to that of the typical Fourier

transformation. This method showed an improvement when compared to the MFCC at a high signal-to-noise ratio. This means that the FrFT method can be applicable in other areas like synthesis or audio enhancement, though in the case of virtual simulations, the Discete Fourier method should prove sufficient.

Existing Issues with Speech Recognition. The main limitation present in commercial speech recognizers is the effects of ambient noise that causes recognition errors in the control system [3]. Therefore, it was found that interfacing the recognizer with a hybrid noise suppression filter can enhance the accuracy even under noisy conditions, though it only works for stationary noises. Furthermore, if the microphone comes with a built in surround noise cancellation, the enhancement would not be too drastic. In a manufacturing environment, ambient noise can be a major concern due to its direct effect on operators in the long term [4]. The Cave Automatic Virtual Environment (CAVE) is a recently developed system that is able to produce a virtual environment to represent the sound level present. Though no speech recognition system was used, the acoustic measurement that was utilized can prove useful in determining the optimum operation position without stalling production or the simulation.

With regards to user interface, consumer electronics is a suitable reference to generate guidelines on interfacing voice control applications so that it remains straightforward and robust [5]. It should give the user the freedom of choice for input, consistency, an appropriate feedback, consideration of the user's expectations, and avoid overloading the channel to maintain precision. These steps are naturally present in this proposed system, making it very accessible to even penetrate the mainstream market.

Microsoft SAPI and SDK. The Microsoft SAPI and software development kit (SDK) proved to be a versatile and efficient tool from numerous research utilizations. A recent study developed a voice control system in a robotized work cell, which is similar to the augmented reality case study for this proposed implementation [6]. Its main focus lies in the specific requirements of a manufacturing cell and considers the mutual influence between the semantic analysis, recognition, syntactic, and spontaneous effects. The underlying rules are that these mutual influences are taken into heavy consideration, the optimal solution for voice variation is by defining a sublanguage, some sort of mechanism for spontaneous speech is required coupled with an immediate reaction to commands, and an easy method for semantic analysis. Microsoft SAPI was used for the online-based aspect of the study as it covers most of the aforementioned factors.ViRbot, a control system for mobile robots that depends on a Microsoft SAPI-powered speech recognition engine, was also developed [7]. A conceptual dependency (CD) primitive is generated with each voice command, which then procedurally generates subtasks to fulfill the command. Since XML notations are supported, the recognition errors were drastically reduced, thus increasing the overall efficiency. Interestingly, command lines which are different but carry the same meaning was able to be recognized by the robot since the CD representation is the same.

Robotic related applications to this software do not end there, as it was implemented for commanding industrial robots as well [8]. It was proven that the technology is suitable for industrial use with reliable results, though it needs to be noted that when it comes to number commands, fixed rules should be avoided to maintain the flexibility. Another issue is that direct voice control on a robotic arm, or any manufacturing tools, must consider ambient noise that is present in the environment that may obstruct the recognition process. Another study combined speech recognition with a web-based control system for a robotic work cell [9]. This allows an operation to be remotely controlled purely through voice that is not limited to pre-defined voice commands. A quasi-natural language system based on the Microsoft SAPI as well allows the system components to be distributed among the client and server. However, since the scope of grammar has been expanded and not limited to voice commands, naturally false recognitions may happen more frequently, This is not to say that an error free function is impossible, but as of now, if machining is involved, it is best to limit the usage of voice recognition. Therefore, this shows that the Microsoft SAPI has low recognition time and high accuracy. For tele-operated robots, such as the Lego Mindstorm, web-based control via WiFi works in a similar fashion [10]. Limitations of such a system have been

tested with multiple case studies, and since both WLAN and IR are the primary method of communication with the robot, latency that exists in wireless communication needs to be considered.

Multimodal interface adoption that includes both gesture and speech is an interesting aspect as well, when applied in fields that can greatly benefit from them, such as architectural work or interior design [11]. A system dubbed as the Open Gesture Recognition Engine (OGRE) system uses both hand motion and speech recognition and comprises of various in-house developed modules, with Microsoft Speech SDK included for voice synthesizing. Most of the restrictions that exist is due to the gesture module, and as for the speech module, the results show that it is favorable when tested on subjects with minimal experience. Therefore, another study excludes the use of bodily gestures and focuses solely on spoken dialogue [12]. This exclusion negates the previously mentioned limitations, and due to the flexibility of Microsoft SAPI, it is able to couple with any API that is SAPI-compliant and dynamically define grammars with XML that can be understood by the robot. The bridge between language and vision exist due to the common memory structures since both of them use symbols for the robot to understand.

Robotics and design tasks aside, training is a ground that needs to be further analyzed for voice recognition to be implemented successfully into simulation systems. A recent study utilizes the Microsoft SAPI as the speech engine in their system for engineering training [13]. The training includes a virtual engine for system control simulation, which contains some resemblance to the virtual manufacturing in this study. The developed intelligent agent facility (IAF) was able to enable multimodal input such as natural spoken language analysis. In this aspect, the voice commands are able to interpret and provide two-way communication by converting the language into scripts. The data from the scripts are used for animation, arm motion, and facial expression.

Application of Other Speech Recognition Tools. Speech recognition has been used for math equations by having it produce equations in digital form. The main challenge is of course the large dictionary, and a recently developed system called Mathifier, places a limitation by following the characteristics of mathematical equations that requires specific grammar structures [14]. This is similar to limiting the engine into recognizing only machine commands instead of a wide array of operations. Systems like Mathifier has a great support for English language as well, but when it comes to support for other languages or those who do not speak English fluently, certain considerations must be taken. Through studies with the Dragon system, it was found that native speakers produce significantly higher accuracy scores, which concludes that single-speaker dependency still requires constant research [15]. Further use of voice recognition systems is highly encouraged to test these boundaries.

Methodology

The interface of the speech recognition allows the user to code for both speech recognition and text-to-speech, as well as offers two levels of access; high-level objects and low-level objects. For this study, the high-level access is utilized because it is more suited for minimalistic voice commands compared to the latter which is designed for sophisticated orders but runs the risk of lower precision, which is not an option in a machining simulation. Figure 1 shows the high-level API that is able to call low-level objects to perform the tasks.

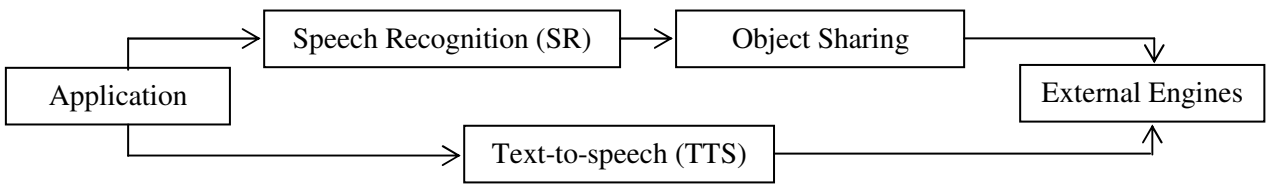

Fig. 1. Architecture of the high-level API

For the speech recognition section, initialization of the Object Linking and Embedding (OLE) is done by calling the "*CoInitialize*" function to create a Voice Command for the application. OLE is

automatically able to launch the specific dynamic-link library (DLL) that corresponds with the application. A notification sink is registered, which is a function used to callback notifications when something happens, and is also required to create a voice menu. Multiple voice menus are possible depending on the number of applications running on the PC. This means that two simultaneous simulations can be performed, or a simulation program coupled with data tabling programs, without any confusion occurring between the voice operations of the running applications. This is done via the "*gpIVoiceCommand*" function. Finally, once all operations are finalized, the OLE is released in the Windows handler. The full sequential functions are detailed in Figure 2, though at this point of time, no actions are taken yet since only recognition is conducted without TTS.

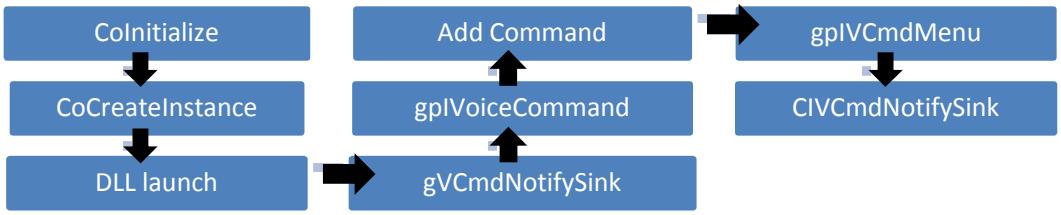

Fig. 2. Program sequence of the speech recognizer

To get the system to actually reply back in English to the operator as a form of feedback to confirm a successful operation, voice text is necessary. The TTS engine is able to output audio depending on the preceding notification sink.Fourier transformation is used to process the voice data, especially when the audio samples are extremely similar. In general, Fourier transformation uses sinusoidal curves to characterize a waveform. It is defined as

$$F\,(k) = \int_{-\infty}^{\infty} f\,(x) e^{-2\,\pi\,i\,k\,x}\,dx. \tag{1}$$

Since the waveform consists of sample signals, the discrete transformation is then defined as

$$X_k = \sum_{n=0}^{N-1} x_n\, e^{-\frac{2\pi i}{N} kn}\; where\; k = 0, \ldots., N-1. \tag{2}$$

The discrete Fourier transformation is able to transform a voice input and sample a continuous function so long as it has a finite duration. The default female cyber voice is embedded into the Microsoft Voice system, but for the general use of manufacturing simulation, it is not necessary to alter this since it will require a low-level API which is needlessly complex in this context.

The Windows Speech Recognition macros further extends the capabilities of the speech recognition tool by utilizing XML codes as shown in the figure below which is able to issue various English language commands triggered by voice commands. This eliminates the need for complex programming because XML grammar is able to interpret English accurately. Figure 3 shows the basic layout of an XML code which immediately allows the system to recognize any English word within the "*listenFor*" function and perform an immediate reply through the "speak" function. The priority function on the code tells the system to ensure the macros has maximum importance over any other Windows core functions, and the question mark at the "?computer" indicates that the word is optional when issuing the command, which is similar to CD primitives.

```
<?xml version="1.0" encoding="UTF-16"?>
<speechMacros><command priority="100">
<listenFor>Hello ?computer</listenFor>
<speak>Yes, sir?</speak>
</command></speechMacros>
```

Fig. 3. Layout of an XML code

This powerful set of macros is able to perform both individual tasks and multiple tasks in series. The virtual simulation processes that will be used runs on the Windows platform, and so it is necessary for the voice recognizer engine to know each keyboard and mouse function that can be dictated, and placed into the XML code in the way where the voice synthesizer can utilize the TTS engine to understand the commands. Additionally, the macros provides two forms of feedback; visual text via the speech widget docked on the screen, and audio confirmation programmed by the operator. Since one of the aims of this study is a higher immersion, audio feedback is favored as it allows the operator to focus on the visualization of the simulation and not the computer screen.

Voice Control in Robotic Work Cell Simulation. The simulation is an augmented reality-based robotic work cell which is created via Visual C++ programming, OpenGL library, and ARToolKit. A graphical user interface (GUI) is not included and the simulation runs purely from compilation of the codes involved in building it. Figure 4 depicts the work cell, which is rendered with OpenGL and tracked with ARToolKit. In this simulation, an offline programming method is implemented to teach the robot arm coordinates in space. The end effector of the arm is manipulated with a marker cube, and mouse operations will save the coordinate of the end effector. Inverse kinematics according to Denavit-Hartenberg's theorem is used to calculate each of the joint angles of the robot. Part of the simulation involves mouse operation to pick and place a virtual object around; if the user clicks the left button and holds it down, the virtual objects is picked, i.e. snaps to the manipulator, and letting it go places the object down, i.e. unsnaps the object. To achieve this in the XML code, the Autohotkey, which is a macro program, gives the additional function of "*mousedown*" and "*mouseup*" functions.

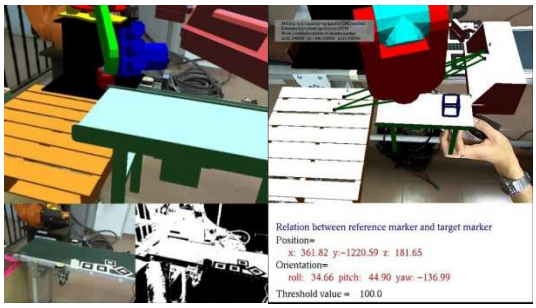

Fig. 4. Augmented reality robotic work cell simulation

The XML code for the simulation process covers the launching of the application until the closing, with a vocal reply assigned to each command, as shown in Table 1. Anytime throughout the simulation, the operator can choose to end the voice recognition simply by saying "stop listening" to pause the system if he or she wishes to switch to keyboard input, and then resume anytime by saying "start listening."

Table 1. List of voice command in the XML code

Voice Command	Function	Computer Reply
Run program	debugs program	initializing robotic work cell simulation
Create work cell	enters key to generate work cell	generating work cell
Save point	saves current coordinate of manipulator into an output file	saving current point
pick	snaps virtual object to manipulator	picking object
place	unsnaps virtual object	placing object
Remove work cell	removes the work cell	removing work cell
End program	ends the program	closing simulation

Discussion

An analysis was conducted on the sound wave of the user to determine the sound clarity and loudness of the input sound which is recorded from the microphone using Audacity, a digital audio

recording software. The user firstly records himself or herself saying “Hello, nice to meet you” in the same intonation, volume, and pronunciation when issuing commands, to collect the sound data in waveforms. Using the FFT algorithm, the region of data is converted into a power spectrum, which represents the energy present in each frequency. A higher vertical or decibel value represents a louder sound. The highest attained decibel is at a frequency of521Hz which lies between the rangeof469Hzto603Hz.This analysis allows us to inspect the degree of acceptance of sound data for the microphone and how much is being understood by the system. The results in Figure 5 show that each audio input is sufficiently loud and can easily be distinguished from the ambient noise present, boosting the signal-to-noise ratio.

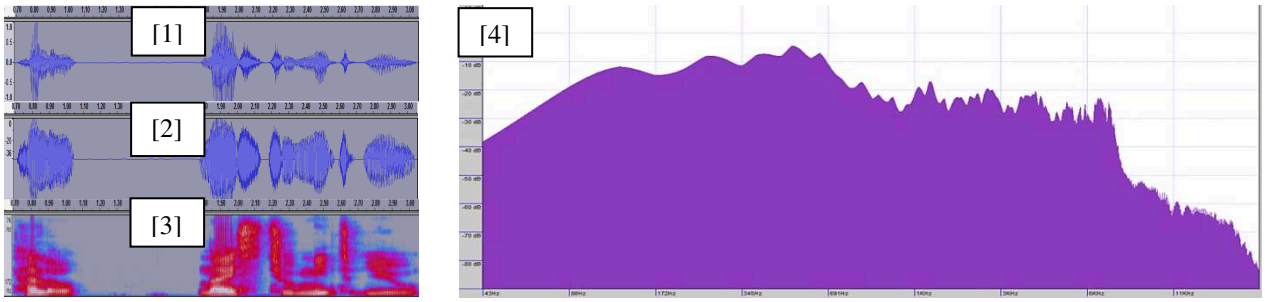

Fig. 5. Waveform of the recorded sound displayed in the unit of (1) Frequency, Hz, (2) Decibel, dB, (3) Spectrum colour, Hz. (4) Graph plotting the Decibel against Log Frequency interpolated using FFT algorithm

The word error rate (WER) method is used to determine the accuracy of the overall system since in general, a higher accuracy indicates a higher level of understanding. WER is calculated with the following equation:

$$WER = \frac{S+D+I}{N} = \frac{Wrong\ commands}{Total\ Commands}\ x\ 100. \quad (3)$$

Where *S* is the number of substitution, *D* is the number of deletion, *I*is the number of insertion, and *N* is the total number of words. This equals to the percentage of wrong commands over total commands[14].The case study is divided into a trained and untrained test for WER, and Table 2 summarizes the results. It is found that when 30 commands were issued, the trained system produces a WER of only 3.3%, while an untrained system produces a WER of 20.0%. This shows that if the SAPI is trained accurately, the voice command is reliable.

Table 2. Comparison result for robotic work cell simulation

	WER (%)	Correct Commands	Wrong Commands
Trained	3.3	29	1
Untrained	20.0	24	6

However, it needs to be understood that a higher degree of understanding a command does not entirely depends on the value of WER where lower WER indicates higher accuracy, since it has been demonstrated before that true understanding depends on more that high recognition accuracy[16]. As long as the experiment conducted matches the optimization objectives or the specific conditions, understanding becomes easier despite having a higher WER. Therefore, the WER value is a rough, though fairly dependable assumption on the degree of which the words are understood by the system, but is not indicative of the true accuracy.

Conclusion

The key point in speech recognition is that currently, it is still almost impossible to make it error-free. Therefore, utilizing the technology is only useful for automating common activities like

issuing commands.Due to the demand of occasional manual intervention, the transition between traditional and voice input needs to be seamless.

Acknowledgements

Thank you to the Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, for providing the necessary facilities to support this study. This work was supported by the Fundamental Research Grant Scheme (FRGS) under Grant Number: FP026-2013A.

References

[1] Lu, J.-n., et al., *Human-machine Interaction Based on Voice.* AASRI Procedia, 2012. **3**(0): p. 583-588.
[2] Sarikaya, R., G. Yuqing, and G. Saon. *Fractional Fourier transform features for speech recognition*. in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*. 2004.
[3] Chan, K.Y., et al., *A hybrid noise suppression filter for accuracy enhancement of commercial speech recognizers in varying noisy conditions.* Applied Soft Computing, 2014. **14, Part A**(0): p. 132-139.
[4] Aurich, J.C., et al., *Noise investigation in manufacturing systems: An acoustic simulation and virtual reality enhanced method.* CIRP Journal of Manufacturing Science and Technology, 2012. **5**(4): p. 337-347.
[5] Gamm, S. and R. Haeb-Umbach, *User interface design of voice controlled consumer electronics.* Philips Journal of Research, 1995. **49**(4): p. 439-454.
[6] Rogowski, A., *Industrially oriented voice control system.* Robotics and Computer-Integrated Manufacturing, 2012. **28**(3): p. 303-315.
[7] Savage, J., et al., *ViRbot: A System for the Operation of Mobile Robots*, in *RoboCup 2007: Robot Soccer World Cup XI*, U. Visser, et al., Editors. 2008, Springer Berlin Heidelberg. p. 512-519.
[8] Pires, J.N., *Robot-by-voice: experiments on commanding an industrial robot using the human voice.* Industrial Robot: An International Journal, 2005. **32**(6): p. 505 - 511.
[9] Rogowski, A., *Web-based remote voice control of robotized cells.* Robotics and Computer-Integrated Manufacturing, 2013. **29**(4): p. 77-89.
[10] Ayres, T. and B. Nolan, *Voice activated command and control with speech recognition over WiFi.* Sci. Comput. Program., 2006. **59**(1-2): p. 109-126.
[11] Sales Dias, M., et al., *Using Hand Gesture and Speech in a Multimodal Augmented Reality Environment*, in *Gesture-Based Human-Computer Interaction and Simulation*, M. Sales Dias, et al., Editors. 2009, Springer Berlin Heidelberg. p. 175-180.
[12] Kulyukin, V., *Human-Robot Interaction Through Gesture-Free Spoken Dialogue.* Autonomous Robots, 2004. **16**(3): p. 239-257.
[13] Wasfy, A., T. Wasfy, and A. Noor, *Intelligent virtual environment for process training.* Advances in Engineering Software, 2004. **35**(6): p. 337-355.
[14] Batlouni, S.N., et al. *Mathifier — Speech recognition of math equations*. in *Electronics, Circuits and Systems (ICECS), 2011 18th IEEE International Conference on*. 2011.
[15] Coniam, D., *Voice recognition software accuracy with second language speakers of English.* System, 1999. **27**(1): p. 49-64.
[16] Wang, Y., Acero, A., and Chelba, C., *Is Word Error Rate a Good Indicator for Spoken Language Understanding Accuracy*, in *IEEE Workshop on Automatic Speech Recognition and Understanding*2003: St. Thomas, US Virgin Islands.

Modern Technologies for Engineering, Applied Mechanics and Material Science
10.4028/www.scientific.net/AMR.980

Implementation of a Voice-ControlSystem for Issuing Commands in a Virtual Manufacturing Simulation Process
10.4028/www.scientific.net/AMR.980.165