

COMMAND BY SPEECH RECOGNITION

ARDIAN SYAH B MOHD YUSOF

This thesis is submitted as partial fulfillment of the requirements for the award of the
Bachelor of Electrical Engineering (Hons.) (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

MAY 2008

ACKNOWLEDGEMENT

With encouragement and determination, I would like to acknowledge those people who gave all of their effort in helping me to finish this project. This project will not come to the end without considerable ideas and support from them.

Firstly, I would like to thank my project supervisor, Mr. Raja Mohd Taufika Raja Ismail, for providing the guideline with continues advices and feedback throughout the duration of finishing this project.

Secondly, I would also like to give appreciation to all other University Malaysia Pahang staff members especially Faculty of Electrical and Electronics Engineering staffs that I may have called upon for assistance since the genesis of this project. Their opinions and suggestions have helped me in realizing this project. Also not to be forgotten, I would like to thank to all of my friends for their support, valuable opinions and ideas sharing during the progress of this project.

Finally, I would like to thank to all of my family members for their understanding, encouragement and support, towards the completion of my project. Thank you so much! May god bless you all.

ABSTRACT

Speech recognition is a topic that very useful in many applications and environments in our daily life. Generally, speech recognizer is a machine which understand humans and their spoken word in some way and can act thereafter. In daily usage, for example, it can be used in a car environment to voice control non-critical operation such as dialing a phone number to ensure a maximum control to the car and enhance the safety. A different aspect of speech recognition is to facilitate for people with functional disability or other kinds of handicap. The system develop for this speech recognizer will be done using MATLAB and for this project, the system will recognize discrete word only and not a sentences or a robust speech. There are two main operations in this speech recognizer which are generating the voiceprint and store it as template in the word bank and recognizing the word spoken by comparing it with the template stored in word bank. The voiceprint is create by extracting it's MEL Frequency Cepstral Coefficient (MFCC) which is the default number of coefficients need to be extracted are 12. To get more accurate result, 20 coefficients will be extracted. For recognizing purpose, Dynamic Time Warping (DTW) method is use. DTW will calculate the distance of two vectors which are the word spoken and the stored voiceprints and it will recognize the word as the same word if the the distance is the lowest or in other words, nearly zero.

ABSTRAK

Pengecaman suara adalah satu topik yang sangat berguna dalam pelbagai kegunaan dan keadaan dalam kehidupan seharian. Secara amnya, pengecam suara adalah satu mesin yang dapat memahami kata-kata manusia dan boleh bertindak balas melaluinya. Dalam penggunaan seharian sebagai contoh, ia boleh digunakan di dalam kereta bagi mengawal operasi yang tidak kritikal seperti mendail nombor telefon bimbit bagi memaksimumkan pengawalan memandu ke atas kereta. Pada aspek yang berlainan pula, teknologi pengecaman suara dpaat membantu orang-orang cacat. Sistem pengecaman suara bagi projek ini telah dibangunkan menggunakan perisian MATLAB dan untuk projek ini, sistem ini hanya mengecam perkataan diskret sahaja dan bukannya satu ayat penuh. Ada dua operasi utama untuk sistem pengecaman suar aini iaitu menghasilkan ‘cap suara’ dan menyimpannya sebagai templat di dalam bank perkataan dan yang kedua ialah mengecam suara yang disebut dengan membandingkan suara yang disebut dengan ‘cap suara’ yang disimpan di dalam bank perkataan. Penghasilan ‘cap suara’ dilakukan dengan mengekstrak Pemalar Kepstral Frekuensi Mel (MFCC) di mana pemalar yang perlu diekstrak pada asasnya adalah sebanyak 12. Untuk mendapatkan hasil yang lebih jitu, sebanyak 20 pemalar akan diekstrak. Bagi tujuan pengecaman pula, kaedah ‘Dynamic Time warping’ (DTW) akan digunakan. DTW akan mengira perbezaan antara dua vektor iaitu suara yang disebut dan ‘cap suara’ yang disimpan dan ia dapat membua tpengecaman jika perbezaan antara kedua-dua vektor tersebut adalah kecil atau menghampiri sifar.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION OF THESIS'S STATUS	
	DECLARATION OF SUPERVISOR	
	TITLE	
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENT	vii
	LIST OF TABLES	x
	LIST OF FIGURES	xi
	LIST OF SYMBOLS	xiv
	LIST OF APPENDICES	xv
1	INTRODUCTION	1
	1.1 Background	1
	1.2 Project Objective	3
	1.3 Project Scopes	4
	1.4 Problem Statement	4
	1.5 Thesis Outline	4
2	LITERATURE REVIEWS	6
	2.1 Introduction	6

2.2	MATLAB	6
2.2.1	The overview of the MATLAB	6
2.2.2	Programming in MATLAB	8
2.2.2.1	If, else, and elseif	8
2.2.2.2	Switch and case	9
2.2.2.3	Try	9
2.2.3	Creating graphical user interface using GUIDE	10
2.2.3.1	Laying out a GUI	11
2.2.3.2	Programming a GUI	12
2.3	SPEECH RECOGNITION	13
2.3.1	Speech recognition methodology	13
2.3.2	MEL Frequency Cepstral Coefficient	14
2.3.3	Dynamic Time Warping	21
3	GRAPHICAL USER INTERFACE DESIGN	23
3.1	Introduction	23
3.2	Graphical user interface (GUI) with MATLAB	23
3.2.1	Creating a GUI programmatically	24
3.2.1.1	Describing the GUI	25
3.2.1.2	Functions summary	26
3.2.1.3	Creating the GUI M-file	26
3.2.1.4	Laying out the GUI	28
3.3	Create GUI using GUIDE	38
3.3.1	Guide Overview	39
3.3.2	Laying out GUI using GUIDE	41
3.3.2.1	Adding the components	43
4	SOFTWARE DEVELOPMENT	48
4.1	Introduction	48
4.2	Programming the gui for recording voice	48
4.2.1	Programming the 'RECORD' push button	49

4.2.1.1	Initializing window soundcard	51
4.2.2	Programming the static text for slider	51
4.2.3	Programming the play and save push button	52
4.3	Programming the GUI for speech recognizer	55
4.3.1	Programming the 'RECOGNIZE' push button	55
4.3.2	Creating the template	56
4.3.3	Dynamic Time Warping	58
4.3.4	Interfacing MATLAB with hardware	59
5	RESULT AND DISCUSSION	60
5.1	Introduction	60
5.2	Voiceprint	61
5.3	Performance of the speech recognizer	64
5.4	Conclusion and future recommendation	68
5.5	Cost and commercialization value	69
	REFERENCES	70
	Appendices A-F	71 - 88

LIST OF TABLES

TABLE NO.	TITLE	PAGE
3.1	List of functions to create the GUI	27
3.2	Series of property/value pairs	33
3.3	Tools of layout editor	40
3.4	Components Description	44
4.1	Arguments for digitalio function	59
5.1	List of the Words in each template	61
5.2	Recognition rate for each template.	64
5.2	(a) English template	64
5.2	(b) Malay template	64
5.2	(c) Dictionary template	64
5.3	User effect to the recognition rate	65
5.4	Voiceprint effect to the recognition rate	67
5.5	Number of words in template effect to the recognition rate	67

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	Spectral representation of speech intensity	2
1.1	(a) Sound Spectrogram	2
1.1	(b) amplitude of the voice signal	2
2.1	MATLAB	7
2.2	If, else, and elseif statement	8
2.3	Switch and case statement	9
2.4	Try statement	10
2.5	Overview of GUIDE	11
2.5	(a) GUIDE quick start page	11
2.5	(b) Layout editor	11
2.6	Adding the Push Button	12
2.7	Callback template for a push button.	13
2.8	Speech recognition algorithm	14
2.9	Steps to extract MFCC from speech signal	15
2.10	Pre-emphasizing the word 'Backward' signal	16
2.10	(a) Original signal	16
2.10	(b) after pre-emphasizing	16
2.11	Hamming windows with different curves	17
2.12	FFT of speech signal	18
2.12	(a) Original signal	18
2.12	(b) FFT signal	18
2.13	Relationship between the mel and the linear frequencies	19
2.14	Triangular Bandpass Filters	20
2.15	Mapping the Dynamic Time Warping	22
3.1	Block diagram of basic operation for recording voice	25
3.2	Sketching a GUI	26

3.3	M-file editor	27
3.4	Command function	28
3.5	Creating figure	29
3.6	Physical view of figure	29
3.7	Statement to add axes	31
3.8	Drawing the axes	32
3.9	Statement to add push button	32
3.10	Effects of normalized unit when resize GUI	34
3.10	(a) Units for both axes are normalized	34
3.10	(b) Units for upper axes is not normalized	34
3.11	Adding slider to GUI	35
3.12	Indicate slider with static text	35
3.13	Slider	36
3.14	Statement for adding pop-up menu	36
3.15	Pop-up menu	37
3.16	GUI for recording voice	38
3.17	GUI for speech recognizer	39
3.18	GUIDE Layout Editor	40
3.19	GUIDE Quick Start Dialog Box	41
3.20	GUIDE Blank Template	42
3.21	Resizing the GUI	42
3.22	Components Palette	43
3.23	Property Inspector	45
3.24	GUI for Speech Recognizer in Layout Editor	46
3.25	M-file Automatically Generated by MATLAB	47
4.1	Pop-up Menu value	49
4.2	Process for recording data	50
4.3	Initializing sound card with MATLAB	51
4.4	Flow Chart for programming the static text	52
4.5	Playing and saving data	53
4.5	(a) Flowchart for playing data	53
4.5	(b) Flowchar for saving data	53
4.6	UIPUTFILE window	54
4.7	Speech Recognizer flowchart	56

4.8	Extracting MFCC from speech signal	57
4.9	Dynamic Time Warping flow chart	58
5.1	Complete GUI for Speech Recognizer	61
5.2	Voiceprints for the word spoken	62
5.2	(a) 'on'	62
5.2	(b) 'off'	62
5.2	(c) 'basikal'	63
5.2	(d) 'kucing'	63
5.3	The difference between basic and improved MFCC for word 'basikal'	66

LIST OF SYMBOLS

<i>cos</i>	-	cosine
DCT	-	Discrete Cosine Transform
DTW	-	Dynamic Time Warping
<i>f</i>	-	frequency
FFT	-	Fast Fourier Transform
GUI	-	Graphical User Interface
GUIDE	-	Graphical User Interface Development Environment
Hz	-	Hertz
kHz	-	kiloHertz
<i>ln</i>	-	Natural logarithm
<i>mel(f)</i>	-	Mel frequency
MFCC	-	Mel Frequency Cepstral Coefficient
ms	-	milliseconds
<i>s(n)</i>	-	Speech signal
<i>t</i>	-	Seconds
	-	Alpha
π	-	phi
Δ	-	delta

LIST OF APPENDIXES

APPENDIX	TITLE	PAGE
A	M-file coding for recording GUI	71
B	M-file coding for speech recognizer GUI	76
C	M-file coding for creating voiceprint	80
D	M-file coding for MFCC	81
E	M-file coding for DTW	85
F	M-file coding for initialize soundcard	88

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Speech recognition is a topic that very useful in many applications and environments in our daily life. Generally, speech recognizer is a machine which understand humans and their spoken word in some way and can act thereafter. In daily usage, for example, it can be used in a car environment to voice control non-critical operation such as dialing a phone number to ensure a maximum control to the car and enhance the safety. Applying voice control technology or speech recognition technology seems will help a lot in enhancing the safety for certain situation. A different aspect of speech recognition is to facilitate for people with functional disability or other kinds of handicap. To make their daily routine easier, voice control could be helpful. With their voice, they could operate the light switch, turn on/off the electrical appliances. Incredibly, this leads to the discussion about intelligent homes where these operations can be made available for common man as well as for the handicapped.

The speech signal and all its characteristics can be represented in two different domains, the time and frequency domain. Spectral representation of speech intensity over time is very popular and the most popular one is the sound spectrogram, see Figure 1.1.

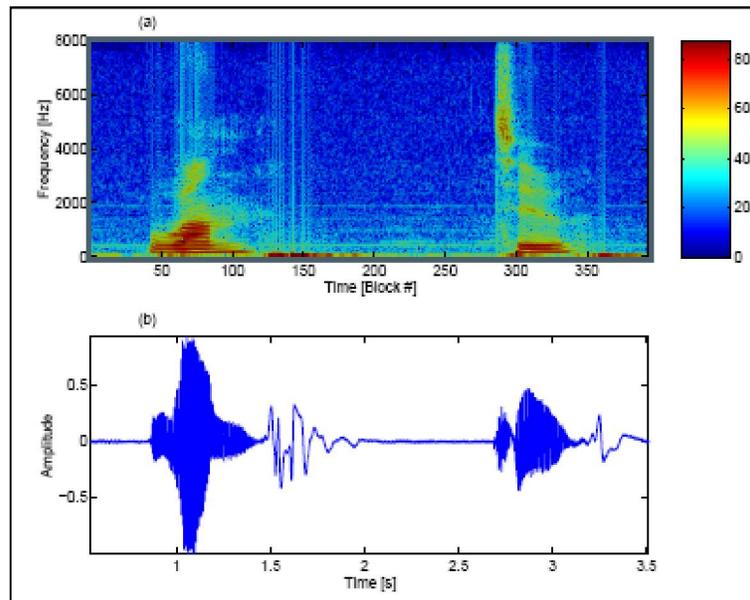


Figure 1.1: Spectral representation of speech intensity (a)Sound Spectrogram
(b) amplitude of the voice signal

The main objective of this final year project is to develop a speech recognition tool based on the MATLAB GUI. This tool will ease and help the researchers, lecturers and the students whom involved in speech recognition study. The implementation of MATLAB software is due to its features which are very useful and easy to use for this project since its already provided certain features that can be used directly for speech recognition analysis. Graphical User Interface (GUI) is create to ensure this tool is user friendly and help the user to perform the analysis faster and easier because they no longer need to write a complex program each time they want to perform some calculations or obtain the graph.

Basically, this speech recognition tool will help the user to record the voice for the analysis purposes. User can choose either to record it with sampling frequency of 8 kHz or 16 kHz. After recording the voice, two graphs will be executed and it will display the voice signal in time domain. One graph will shows the amplitude of the voice signal and the another one will shows its spectrogram.

User can play it back and save the file as .wav file. Besides that, user can also load other wave files which are stored in their computer. Fourier analysis can be done easily with this tool. With one click, user can obtain the voice signal graph in frequency domain which is vital in speech recognition analysis.

This tool offers many advantages to the user because it can help to improve the result of the analysis perform by the user. User no longer need to waste a lot of time just to write a simple program, for example, to obtain the fourier graph for the speech signal. The recording tool provided in this tool also enable the user to record the voice with sampling frequency of 8 kHz compare with the Windows recorder which only enable the user to record the voice with sampling frequency of 16 kHz.

This speech recognition tool is divided into two sections:-

1. Create a GUI to record the voice using microphone and play it back.
2. Create a GUI to perform speech recognition.

1.2 Objectives

1. To create a graphical user interface (GUI) with MATLAB
2. To create a voiceprint for the word spoken
3. To perform speech recognition for commanding electrical devices.

This is the main objective for this project. A GUI will be created as the speech recognizer and it will perform the speech recognition process. The process of recognizing the word spoken is done based on the Dynamic Time Warping (DTW) method/. By using DTW, it will calculate the difference or the distance between the word spoken and and the voiceprint that have been stored in the words bank. If the distance is small, that means the word spoken is equal to the voiceprint and the speech recognizer will recognize it.

1.3 Project scope

The main scope for this project is to create a user friendly GUI using MATLAB to perform the speech signal analysis for the speech recognition purpose. The GUI will consist of several functions and buttons of operation regarding to the speech recognition analysis. User just need to select one of the button to perform certain job. The GUI will display the desire result according to what its task is and have the ability to save or print the result.

1. Recording Tool with Playback Ability.
2. Saving file
3. Perform the speech recognition using MATLAB

1.4 Problem statement

In our daily life, electrical devices are very important in order to improve the quality of our life but most of them are not friendly for those handicapped user. For that reason, this project is propose in order to create a system that can be attached with electrical devices and command them by using our voice. Those handicapped user can easily operate the devices without touching all the operate button. But, this system is totally not suitable for those who are experience the mute problem.

1.5 Thesis outlines

The Speech Recognition Tool final thesis is a combination of 5 chapters that contains and elaborates specific topics such as the Introduction, Literature Review,

Software Design, Result, Discussion, Conclusion and Further Development that can be applied in this project.

Chapter 1 basically is an introduction of the project. In this chapter, the main idea about the background and objectives of the project will be discussed. The full design and basic concept of the project will be focused in this chapter. The overview of the entire project also will be discussed in this chapter to show proper development of the project.

Chapter 2 is about the literature review and the methodologies for the development of the Speech Recognition Tool. This includes the future project development that can be added in this project.

Chapter 3 will be discussed about the design of the graphical user interface (GUI) using MATLAB. In this chapter, it will explain how to create GUI programmatically or using the graphical user interface development environment (GUIDE) provided for MATLAB with version 7.0 and above.

Chapter 4 will discuss the software development. It will show and explain the flow chart that been used to write the coding, developing the process using the MATLAB.

Chapter 5 discusses all the results obtained and the limitation of the project. All discussions are concentrating on the result and performance of the speech recognizer. This chapter also discusses the problem and the recommendation for this project. Commercialization value also be discussed in this chapter.

CHAPTER 2

LITERATURE REVIEWS

2.1 Introduction

In this chapter, the basic knowledges and fundamental concept in creating the speech recognizer will be discussed. This speech recognition project is using the MATLAB as the main processor and dynamic programming for the recognition purpose.

2.2 MATLAB

MATLAB is an interactive, matrix-based system for scientific and engineering numeric computation and visualization. User can solve complex numerical problems in a fraction of the time required with a programming language such as Fortan or C. The name MATLAB is derived from MATrix LABoratory. When using MATLAB. The command *help functionname* will give information about a specific function. For example, the command *help fft* will give information about function fast fourier transform and how to use it.

2.2.1 The overview of the MATLAB

The MATLAB high-performance language for technical computing integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Figure 2.1 shows the looks of the MATLAB.

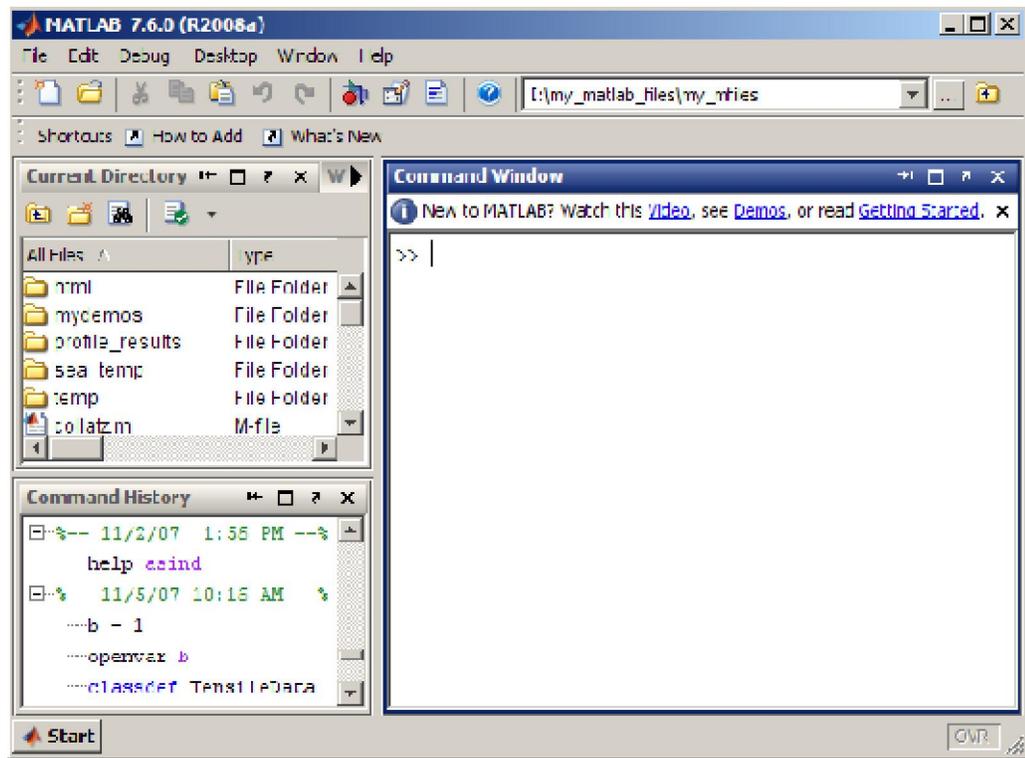


Figure 2.1: MATLAB

Typical uses include:

1. Math and computation
2. Algorithm development
3. Data acquisition
4. Modeling, simulation, and prototyping
5. Data analysis, exploration, and visualization
6. Scientific and engineering graphics
7. Application development, including graphical user interface building

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called *toolboxes*. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. You can add on toolboxes for signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many other areas.

2.2.2 Programming in MATLAB

This section covers those MATLAB product functions that provide conditional program control.

2.2.2.1 If, else, and elseif

The if statement evaluates a logical expression and executes a group of statements when the expression is *true*. The optional elseif and else keywords provide for the execution of alternate groups of statements. An end keyword, which matches the if, terminates the last group of statements. The groups of statements are delineated by the four keywords—no braces or brackets are involved. The MATLAB algorithm for generating a magic square of order n involves three different cases: when n is odd, when n is even but not divisible by 4, or when n is divisible by 4. This is described by Figure 2.2:

```
if rem(n,2) ~= 0
    M = odd_magic(n)
elseif rem(n,4) ~= 0
    M = single_even_magic(n)
else
    M = double_even_magic(n)
end
```

Figure 2.2: If, else, and elseif statement

2.2.2.2 Switch and case

The switch statement executes groups of statements based on the value of a variable or expression. The keywords case and otherwise delineate the groups. Only the first matching case is executed. There must always be an end to match the switch. The logic of the magic squares algorithm can also be described by Figure 2.3:

```
switch (rem(n,4)==0) + (rem(n,2)==0)
case 0
    M = odd_magic(n)
case 1
    M = single_even_magic(n)
case 2
    M = double_even_magic(n)
otherwise
    error('This is impossible')
end
```

Figure 2.3: Switch and case statement

Unlike the C language switch statement, the MATLAB switch does not fall through. If the first case statement is true, the other case statements do not execute. So, break statements are not required.

2.2.2.3 Try

The general form of a try-catch statement sequence is shown in Figure 2.4. In this sequence the statements between try and catch are executed until an error occurs. The statements between catch and end are then executed. Use `lasterr` to see the cause of the error. If an error occurs between catch and end, MATLAB terminates execution unless another try-catch sequence has been established.

```
try
    statement
    ...
    statement
catch
    statement
    ...
    statement
end
```

Figure 2.4: Try statement

2.2.3 Creating graphical user interface using GUIDE

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). We can also create GUIs programmatically. These tools greatly simplify the process of designing and building GUIs. We can use the GUIDE tools to:

1. Lay out the GUI.

Using the GUIDE Layout Editor, we can lay out a GUI easily by clicking and dragging GUI components—such as panels, buttons, text fields, sliders, menus, and so on—into the layout area. GUIDE stores the GUI layout in a FIG-file.

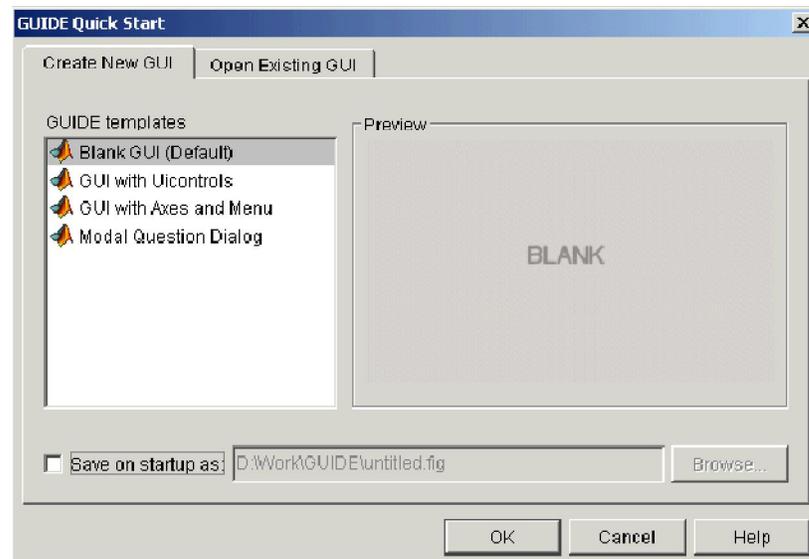
2. Program the GUI.

GUIDE automatically generates an M-file that controls how the GUI operates. The M-file initializes the GUI and contains a framework for the most commonly used callbacks for each component—the commands that execute when a user clicks a GUI component. Using the M-file editor, we can add code to the callbacks to perform the functions we want.

2.2.3.1 Laying out a GUI

Start GUIDE by typing `guide` at the MATLAB® command prompt. This displays the GUIDE Quick Start dialog box. When open a GUI in GUIDE, it is displayed in the Layout Editor, which is the control panel for all of the GUIDE tools.

Figure 2. 5 (b) shows the Layout Editor with a blank GUI template.



(a)