

University of Sheffield

# Real-Time Speech Synthesis based on a Simulated Vocal Tract



Chenxi Liu

*Supervisor:* Prof. Roger K. Moore

A report submitted in fulfilment of the requirements  
for the degree of MSc in Advanced Computer Science

*in the*

Department of Computer Science

COM6905  
October 11, 2019

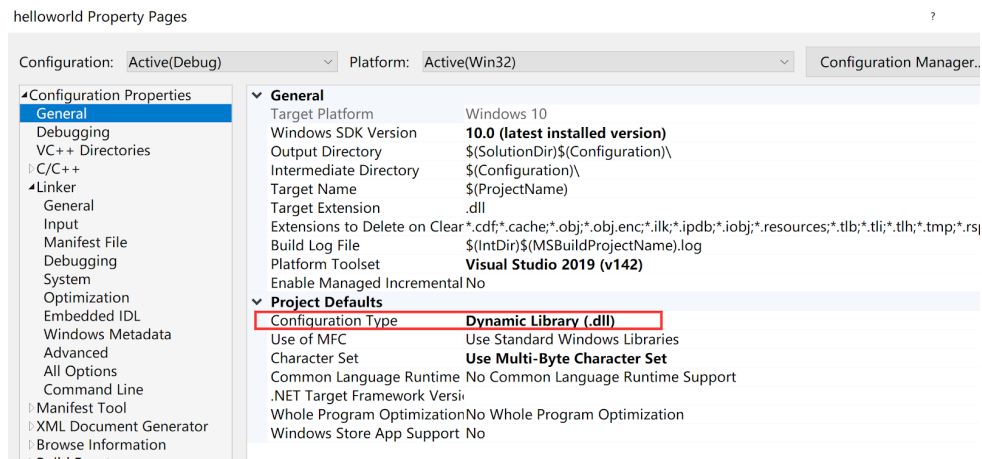
# Chapter 1

## Setup Guide and User Guide

### 1.1 Setup Guide

#### 1.1.1 IDE configuration

1. download Visual Studio 2019(Or any version of Visual Studio should work fine)
2. copy 'pd.lib' to the work folder
3. create a new project 'helloworld'
4. Set configuration properties:
  - a. Set configuration type to .dll(shown as Figure 1.1):
    1. Back in Visual C++, open the helloworld Property Pages window (Alt+F7).
    2. In the left pane, select Configuration Properties - General.
    3. In the right pane, under Project Defaults, click on Configuration Type, and select Dynamic Library (.dll)

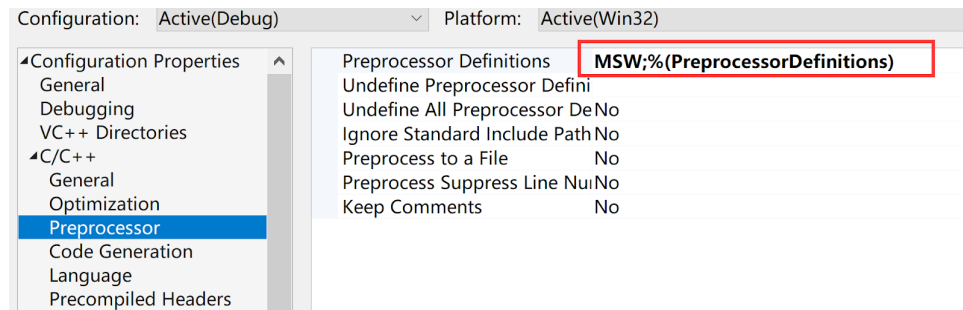


*Figure 1.1: Set configuration type to .dll*

- b. Add MSW to preprocessor definitions(shown as Figure 1.2):
  1. In the left pane, select

Configuration Properties - C/C++ - Preprocessor.

2. In the right pane, type MSW in the Preprocessor Definitions field.



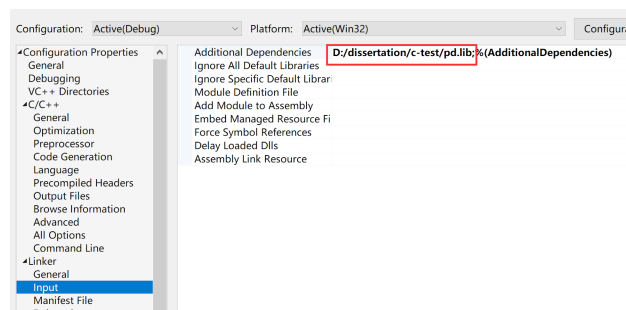
**Figure 1.2:** Add MSW to preprocessor definitions

- c. Tell compiler which language to use:

1. In the left pane, select Configuration Properties - C/C++ - Advanced.
2. In the right pane, select Compile As.
3. Select Compile as C Code (/TC)

- d. Tell linker where to find pd.lib:

1. In the left pane, select Configuration Properties - Linker - Input.
2. In the right pane, select Additional Dependencies and enter 'D:/workplace/pd.lib.'

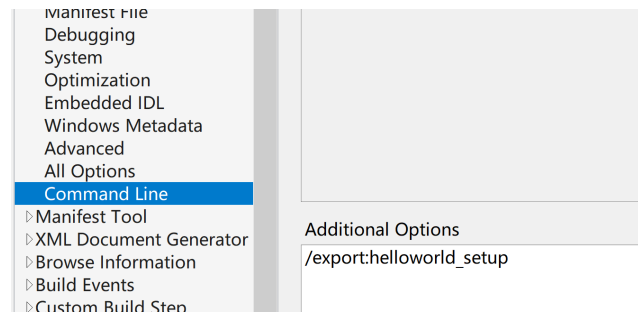


**Figure 1.3:** Tell linker where to find pd.lib

- e. Tell linker to export the setup function:

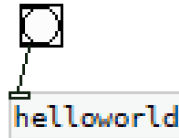
1. In the left pane, select Configuration Properties - Linker - Command Line.
2. In the right pane, type '/export:helloworld\_setup' in the Additional options field.
3. Click OK.

5. Compile and link: Use the Build Solution command (F7).
6. Copy the new helloworld.dll file into pd.



**Figure 1.4:** Tell linker to export the setup function

7.create an object and it worked!

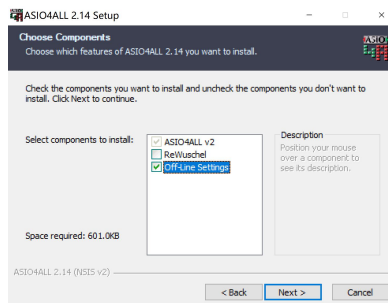


**Figure 1.5:** create object

### 1.1.2 ASIO4ALL configuration

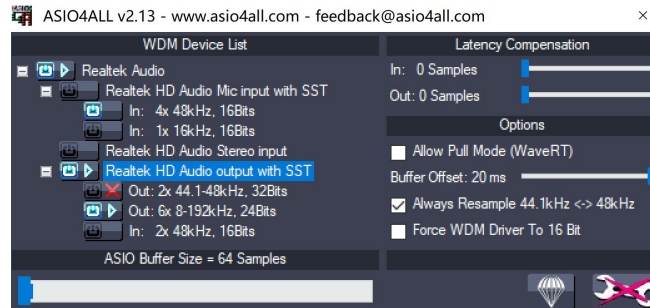
It is best to get a ASIO4ALL for a better sound output of the Pure Data. The official website of ASIO4ALL is <http://www.asio4all.org/>. Once downloaded, you can follow the steps below to configure your Pd:

1. download the ASIO4ALL and follow the install instructions. Note to check the off-line setting box as Figure 1.6 below



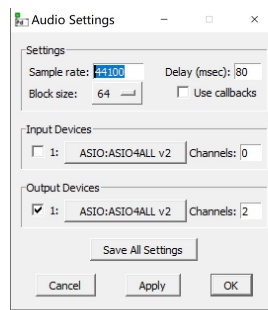
**Figure 1.6:** ASIO install

2. After it successfully installed, go to its root dictionary and click to open the .exe file and configure the ASIO4ALL as Figure 1.7 shows.



*Figure 1.7: ASIO configuration*

3. Then, go to the pd-vanilla or pd-extended, click the 'ASIO(via portaudio)' in the media menu.
4. Select ASIO in the 'Input Device' and 'Output Device' as shown in Figure 1.8.



*Figure 1.8: ASIO - Pure Data configuration*

Except for ASIO4ALL, Jack Route can also be used for the same purpose. If you are Mac user, sound-flower is also worth trying.

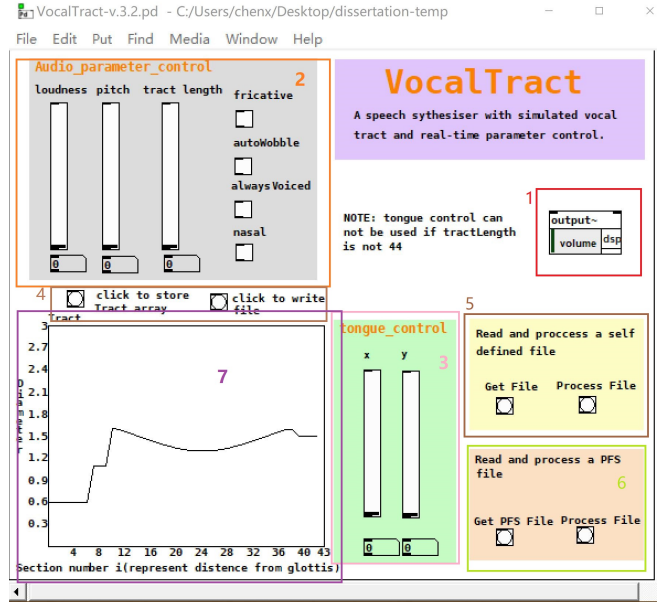
### 1.1.3 Pd-vanilla configuration

The external can be used in the pd-extended once you put the .dll file into the './pd/extra' dictionary. However, pd-extended and pd-vanilla is not always compatible. For this project, you can copy the pd.lib file in the pd-extended dictionary and paste it to substitute the pd.lib file in the pd-vanilla dictionary.

## 1.2 User guide

### 1.2.1 Interface guidance

The user interface is roughly divided into seven sections according to different functions as Figure 1.9 manifests.



*Figure 1.9: User interface guidance*

1. The first section in the figure above is the switch of DSP. The switch should be kept turned on when you are using other functions. You can adjust volume here as well.
2. The second section is the audio parameter control panel. you can try any number of any parameters you want to see what happened to the sound output. But note that the tongue control can not be used if the tract length is not set as the default length 44.
3. The Third section is the tongue control section and you can see the tract shape changed by changing the tongue coordinate.
4. The Fourth section can store your tract diameter array every time you click the store button and output it as a file named 'tract\_array' when you click the write button.
5. The fifth section is for self-defined file processing. The format of the self-defined file is 'pitch loudness fricative nasal autoWobble alwaysVoiced TractArray'. Note that the length of the tract array is depend on the section number of the tract length you defined.
6. The sixth section is for a PFS file processing. If you right click the interface and click open you can see the inner patch of the synthesizer. The PFS file is first processed in a

sub-patch named 'testPitch' and the output of the processed numbers are the input of the external 'FormantTransfer'. This is an external to convert the acoustic parameters to the model parameters. This function can generate the entire speech utterances through the stored parameters.

7. The seventh section is the graphic interface for the tract diameter array. You can define the tract shape by drawing the different line chart in the table. Please try different diameter values and you can the corresponding output.

### 1.2.2 Solutions for possible noise sound

If the synthesizer is not producing a clear and distinguishable sound, it might be the problem of wrong configuration. First please check the setup guide and download the ASIO4ALL if you haven't download it. If it still not working and there are continuous noise sound in the background, you may want to check the cpu and make sure you does not have your computer overwhelming. If you are using a laptop, you have to pay special attention to whether your device is charging and make sure it is.

## Chapter 2

# Base structural components

The value of the three fixed value arrays is shown as Figure 2.1. The index  $i$  is the section number from glottis to lip.



$i$	$\omega(i)$	$\phi_1(i)$	$\phi_2(i)$
1	0.636	0.018	-0.013
2	0.561	0.001	-0.007
3	0.561	-0.013	-0.029
4	0.550	-0.025	-0.059
5	0.598	-0.036	-0.088
6	0.895	-0.048	-0.108
7	1.187	-0.062	-0.120
8	1.417	-0.076	-0.123
9	1.380	-0.093	-0.118
10	1.273	-0.111	-0.107
11	1.340	-0.130	-0.092
12	1.399	-0.149	-0.075
13	1.433	-0.167	-0.056
14	1.506	-0.183	-0.035
15	1.493	-0.196	-0.014
16	1.473	-0.204	0.008
17	1.499	-0.207	0.032
18	1.529	-0.203	0.057
19	1.567	-0.193	0.084
20	1.601	-0.175	0.111
21	1.591	-0.151	0.138
22	1.547	-0.119	0.164
23	1.570	-0.082	0.188
24	1.546	-0.041	0.206
25	1.532	0.004	0.218
26	1.496	0.051	0.221
27	1.429	0.097	0.214
28	1.425	0.141	0.195
29	1.496	0.181	0.164
30	1.608	0.214	0.121
31	1.668	0.240	0.070
32	1.757	0.257	0.013
33	1.842	0.264	-0.046
34	1.983	0.260	-0.100
35	2.073	0.246	-0.143
36	2.123	0.224	-0.167
37	2.194	0.194	-0.165
38	2.175	0.159	-0.132
39	2.009	0.122	-0.066
40	1.785	0.087	0.031
41	1.675	0.057	0.148
42	1.539	0.038	0.264
43	1.405	0.034	0.346
44	1.312	0.048	0.338

*Figure 2.1: Base structural components*