

在人工智能辅助下构建的 一体式多普勒效应演示系统

All-in-one Doppler effect demonstration system
based on AI support

主 讲：魏 霞

指导教师：杨 英

制作团队：科普未来队

CONTENTES

目录

#01 背景与意义

Background and Significance

#02 方法及过程

Methods and Processes

#03 多普勒现象展示及应用

Display and Application

#04 改进构思及方案

Improvement ideas and proposals



#01

背景与意义

Background and Significance

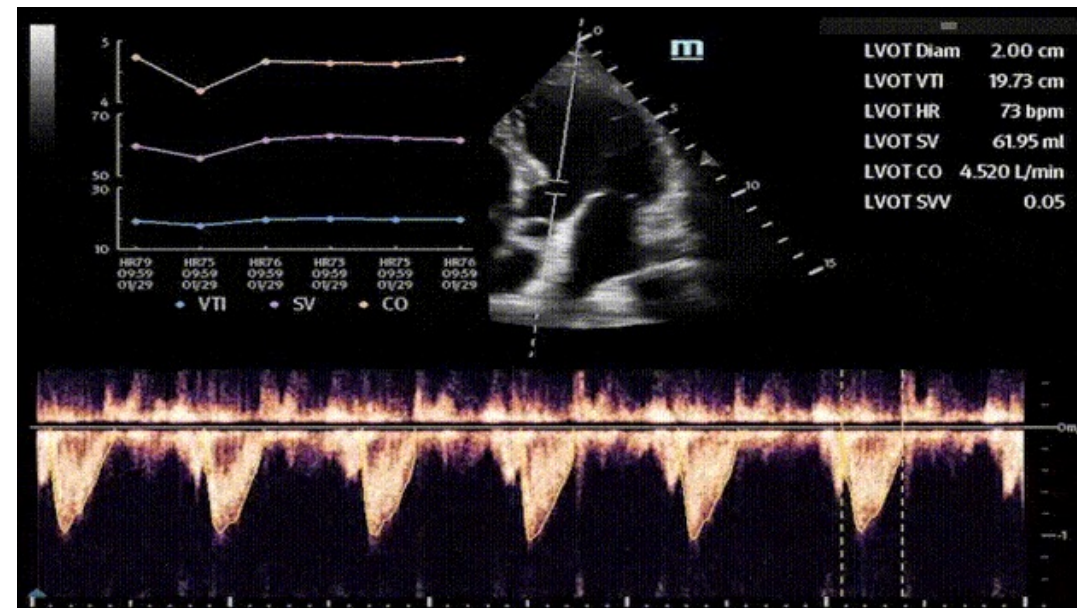
#1 背景与意义

多普勒效应是指波源与观测者之间存在相对运动时，观测到的波的频率发生变化的现象，其原理在于当波源向观测者靠近时，波的波长被压缩、频率升高；而当波源远离观测者时，波的波长被拉长、频率降低。



#1 背景与意义

在医学中，多普勒效应被广泛应用于超声诊断技术中，通过分析血液流动引起的超声波频率变化，医生可以评估血管狭窄、心脏功能异常等情况。



在工程领域，它的应用涵盖雷达测速、移动通信和声呐技术，例如雷达利用频率变化来测量物体速度，而移动通信中通过多普勒频移优化信号质量。这种效应因其简单却强大的原理成为多个科学与技术领域的重要工具。



#1 背景与意义

根据多普勒效应公式：

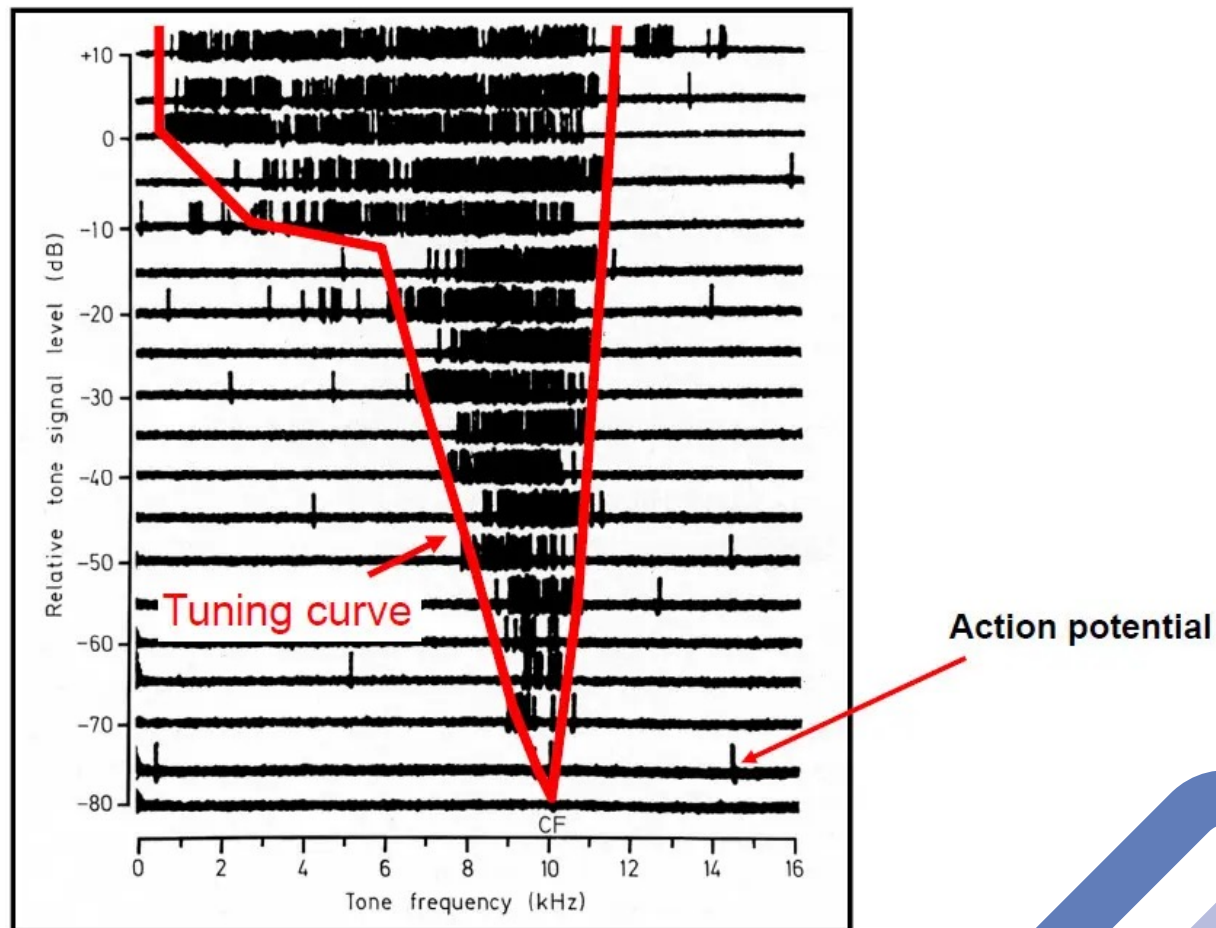
$$f' = \left(\frac{v \pm v_0}{v \mp v_s} \right) f$$

可以知道，观察者所接收到的频率变化与其相对于声源的速度有关，相对速度越快，多普勒效应越明显；并且声源发出的频率越高，多普勒效应也更明显。



#1 背景与意义

但是我们在实际演示的过程中发现，我们对高频声音的变化感知并不敏感，仅靠人耳感知不到多普勒效应。



如何让多普勒效应展现更明显？





#02

方法及过程

Methods and Processes

#2 方法及过程

因为计算机对声音处理具有客观性，并且具有良好的演示特性，因此我们团队决定使用计算机多媒体技术来展示多普勒效应。

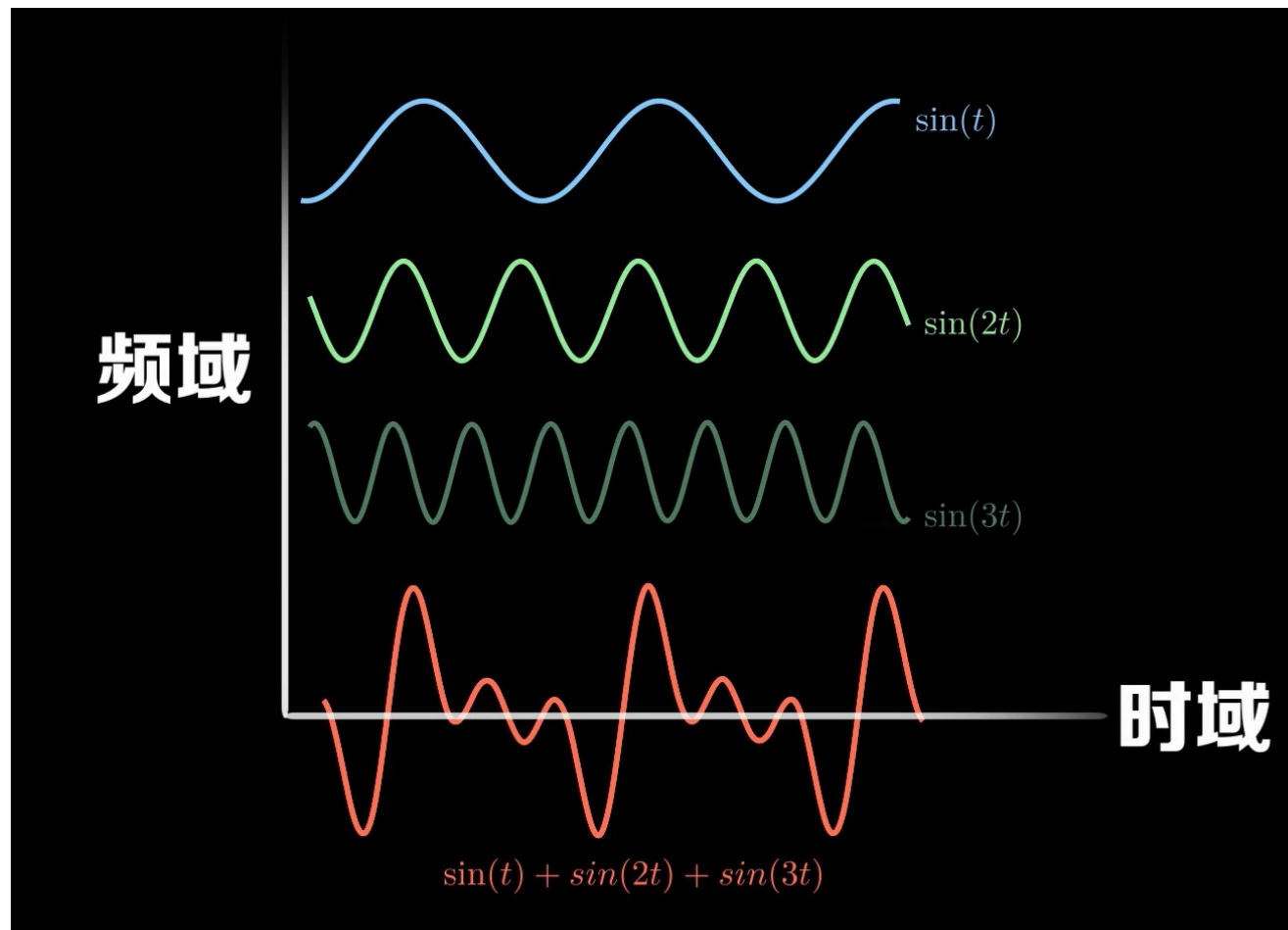




#2 方法及过程

核心理论

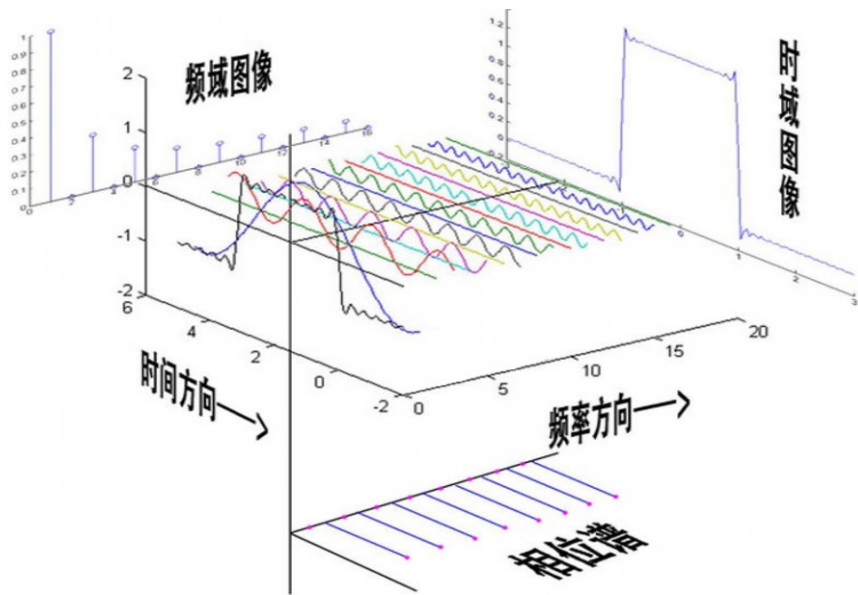
傅里叶变换（Fourier Transform）是一种将信号从时域转换到频域的数学工具。它的核心思想是，任何复杂的信号（例如声音、图像、振动等）都可以通过一系列正弦波（不同频率和振幅的波）来表示。傅里叶变换揭示了信号中的各个频率分量，并将其分解为频率的集合。



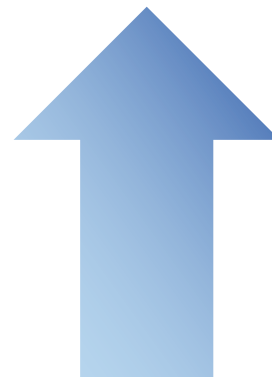
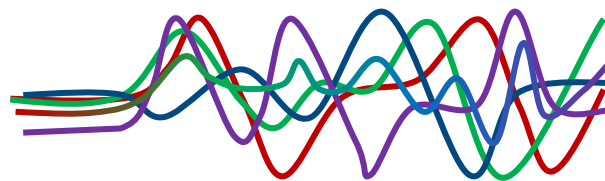
#2 方法及过程

核心理论

简单来说，傅里叶认为所有事物都可以被“拆分”和“重组”。



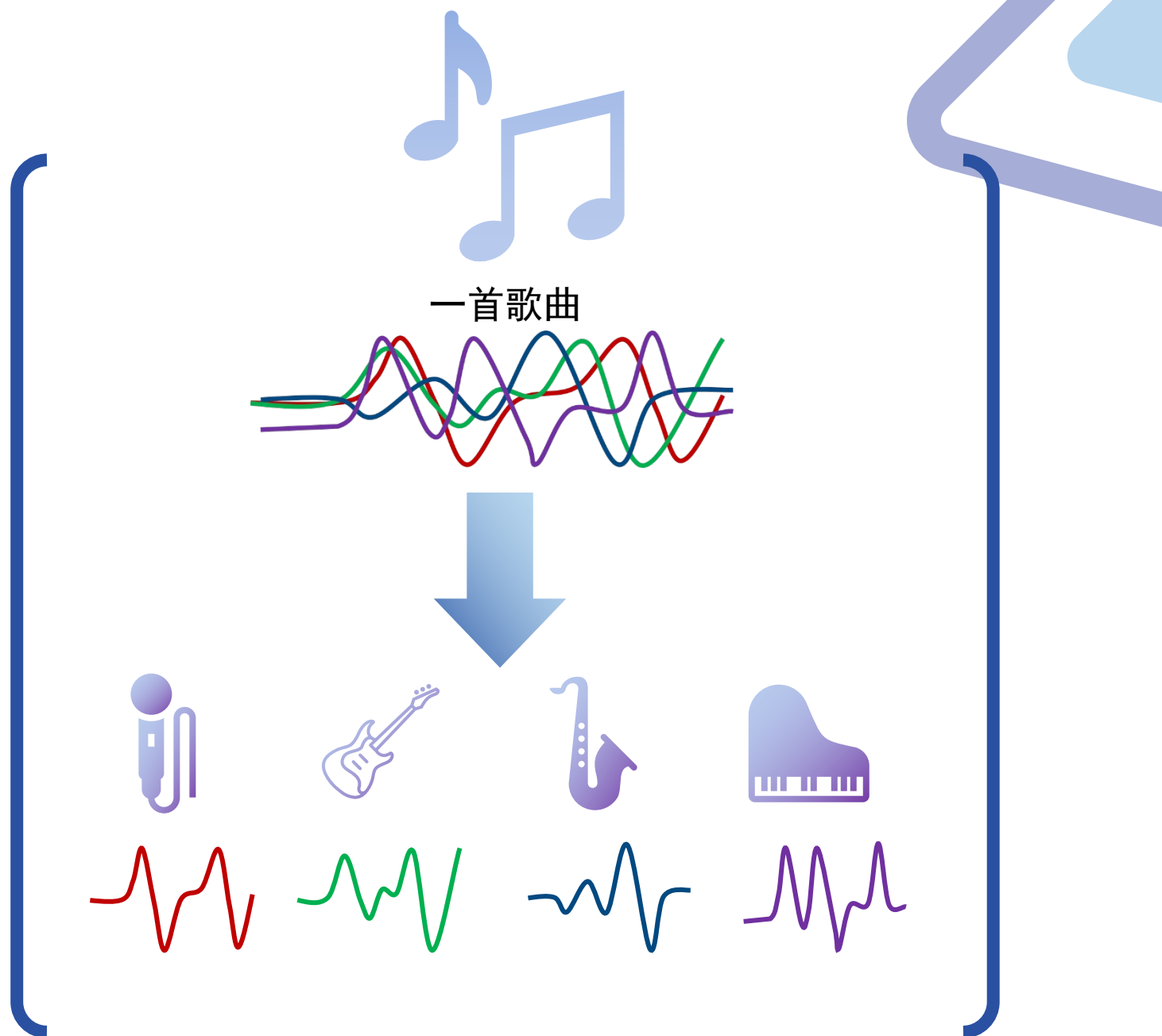
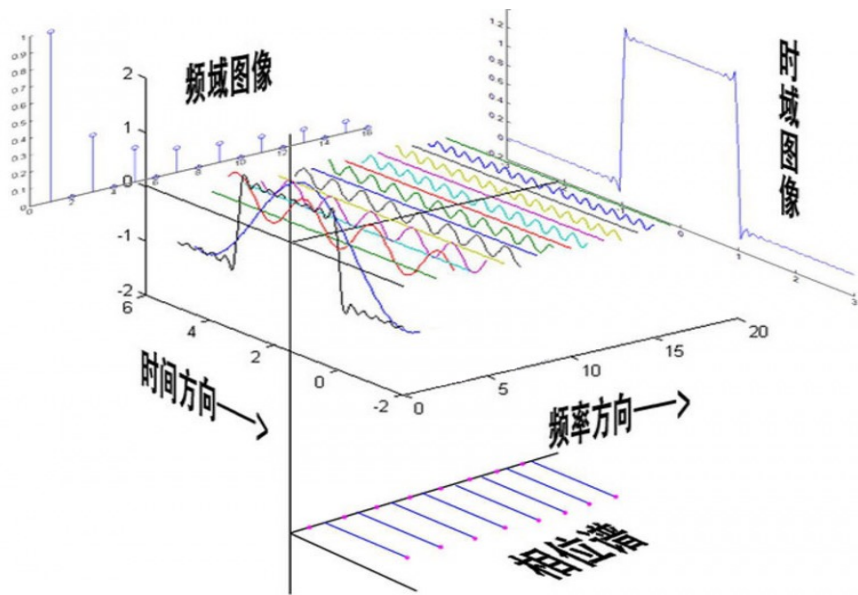
一首歌曲



#2 方法及过程

核心理论

简单来说，傅里叶认为所有事物都可以被“拆分”和“重组”。





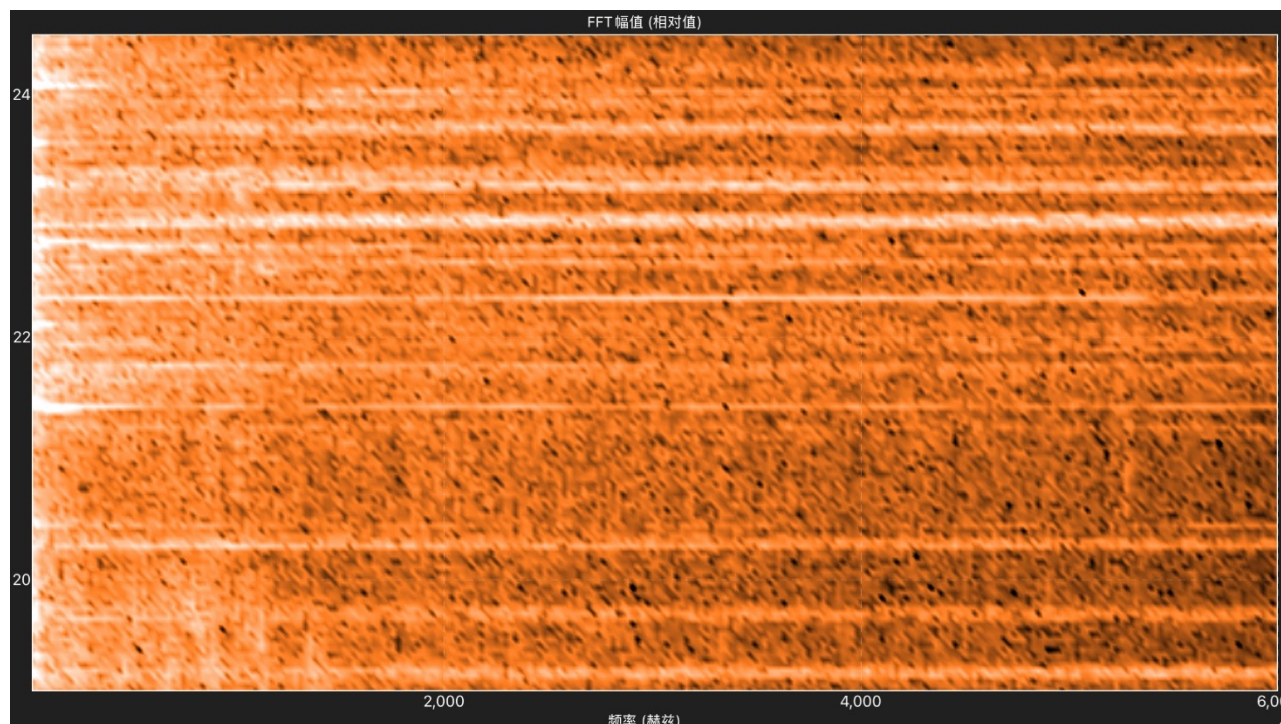
#2 方法及过程

计算方法

在数字信号处理中，信号由有限的离散数据点表示。为了将时域信号转换为频域信号，需要对离散数据进行傅里叶变换，通常使用**快速傅里叶变换**

(FFT) 算法。FFT是一种高效计算离散傅里叶变换（DFT）的方法，可以将大规模数据的处理速度大大提升：

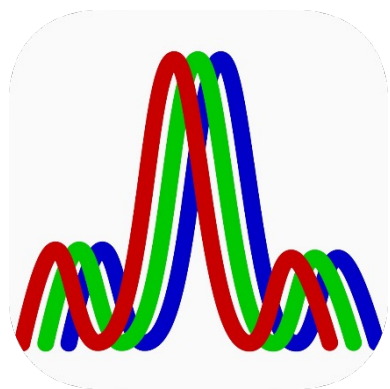
$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn}, \quad k = 0, 1, \dots, N-1$$



FFT频谱图

#2 方法及过程

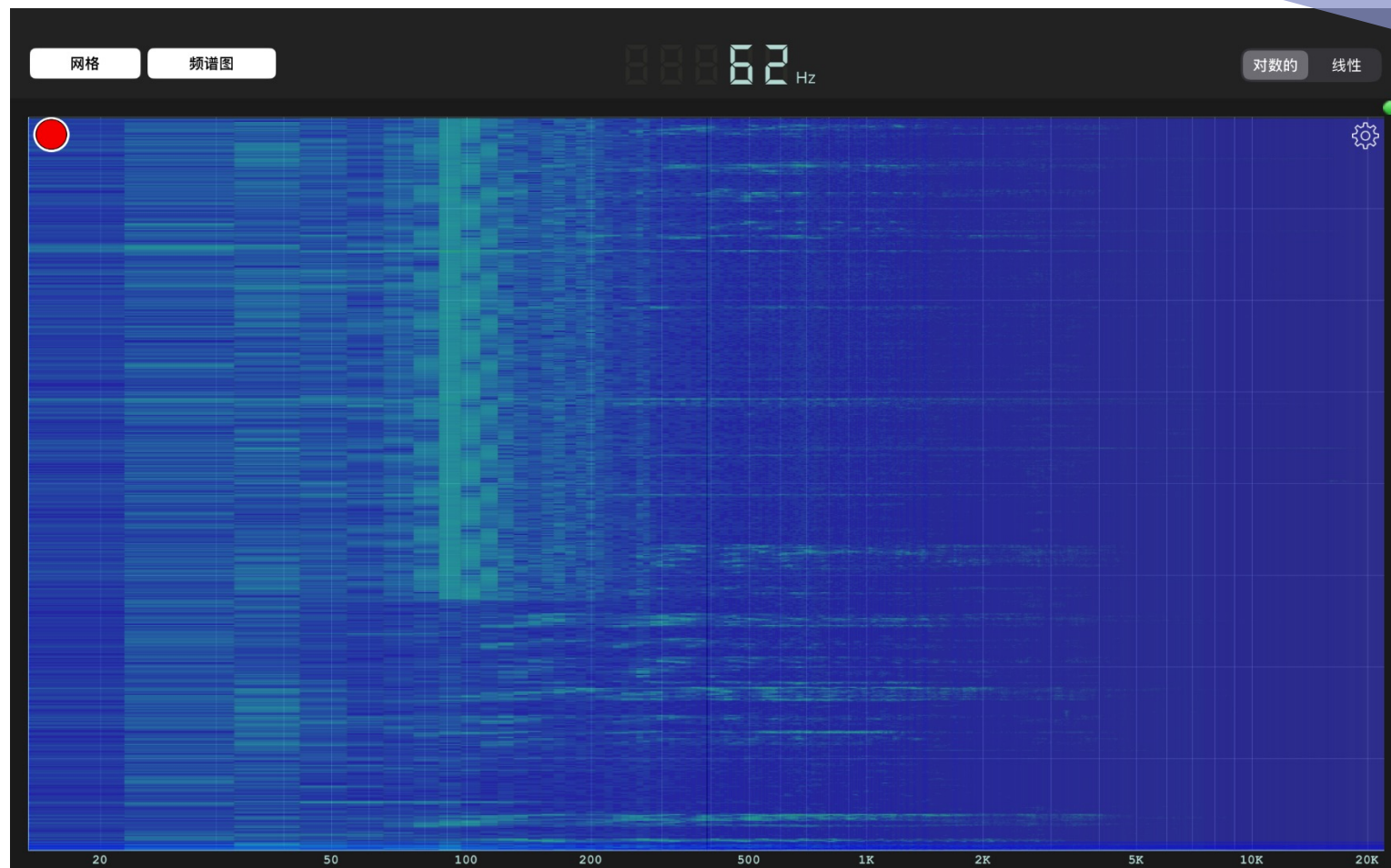
演示方案1 频谱图演示法



Sonic Tools
接收处理音频



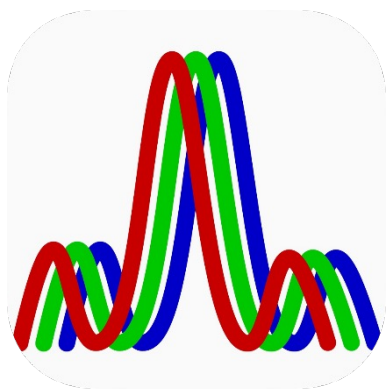
phyphox
作为发声源



频谱图

#2 方法及过程

演示方案1 频谱图演示法



Sonic Tools
接收处理音频



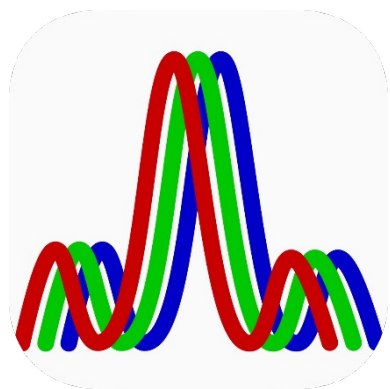
phyphox
作为发声源

优点：能够尽可能完整的还原声音频谱，适用于专业人员分析。

缺点：图形过于复杂，并且难以观察到多普勒效应；违背了科普的通俗性和趣味性。

#2 方法及过程

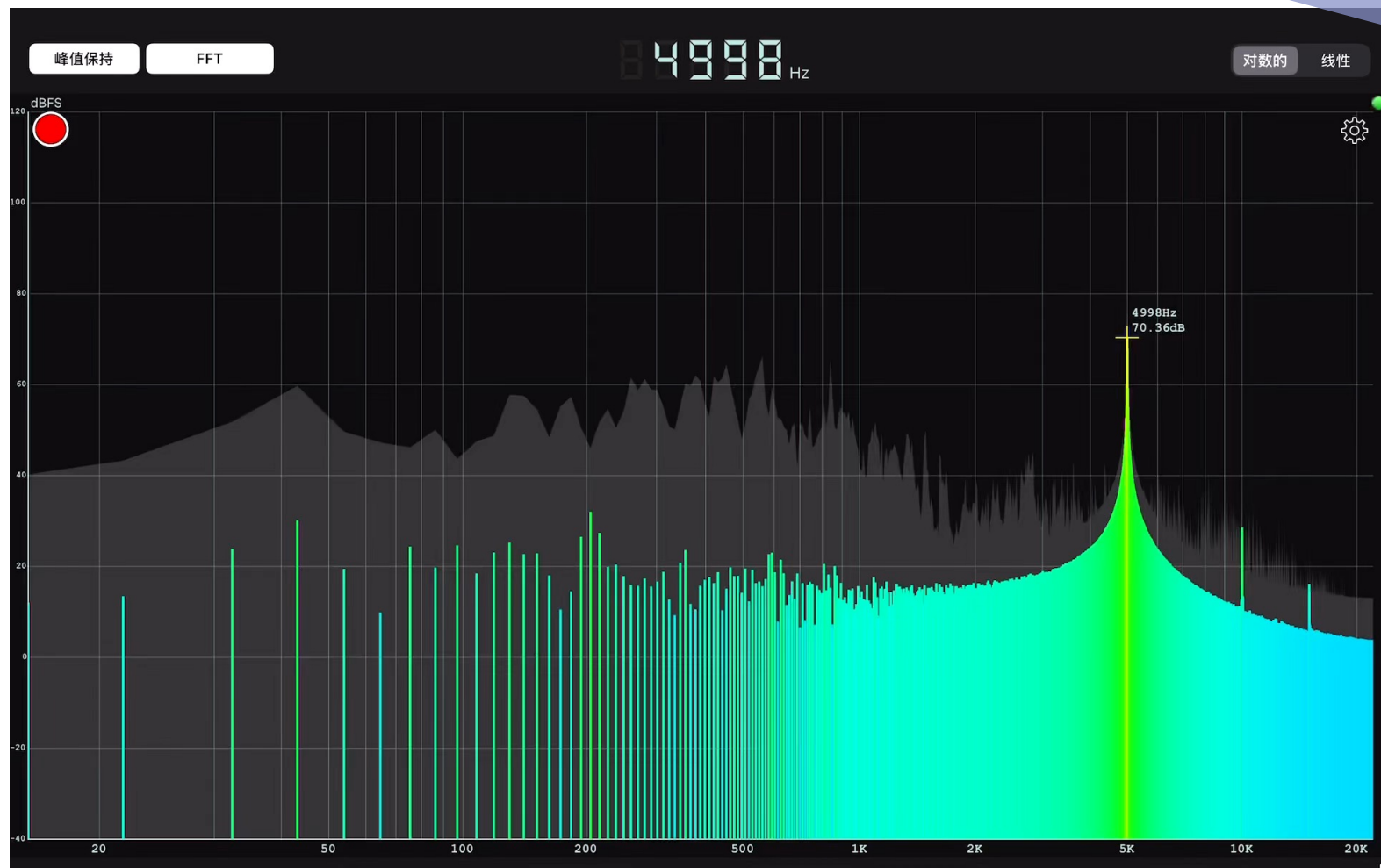
演示方案2 FFT图像演示法



Sonic Tools
接收处理音频



phyphox
作为发声源

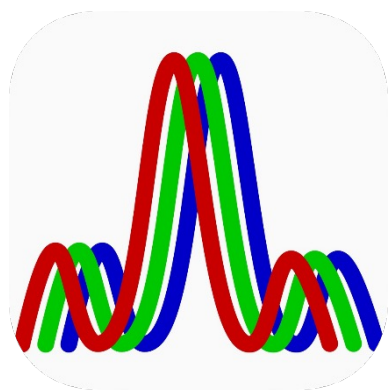


FFT图像

#2 方法及过程

演示方案2

FFT图像演示法



Sonic Tools
接收处理音频



phyphox
作为发声源

优点：能够比较简单直观的反映出所接收到音频频率的变化。

缺点：无法观察到变化的趋势以及峰值等数据，缺乏连续性。

#2 方法及过程

演示方案3 phyphox演示法



phyphox
多普勒效应功能



phyphox
作为发声源



“多普勒效应” 功能

#2 方法及过程

演示方案3 phyphox演示法



phyphox
多普勒效应功能



phyphox
作为发声源

优点：能够连续的显示频率的变化，能够记录峰值等数据。

缺点：图像较曲折，并且易受到环境声干扰；需要两个设备，并且需要进行参数设置，设置错误可能出现运行错误的情况。

如何让演示效果更好?

#2 方法及过程

目标定位

基本功能实现

能够实现如phyphox软件中的多普勒图形效果，并实现一体化实验。

01



02



03



04



改进演示效果

通过加窗等技术处理，让图像更加平滑，从而让演示效果更好。

减少环境声影响

通过阈值等方式让程序能够尽可能地避免环境杂音的影响。

添加动画演示

通过程序设计，展示多普勒效应的传播形式，与实验结合，加深受众的理解。

开发过程



Step·01

文献搜集

确定功能目标，在网络上查找相关文献；整合资料后理出整体思路。



Step·02

程序编写

使用Matlab进行功能函数的编写，为了便于操作添加GUI图形操作页面。



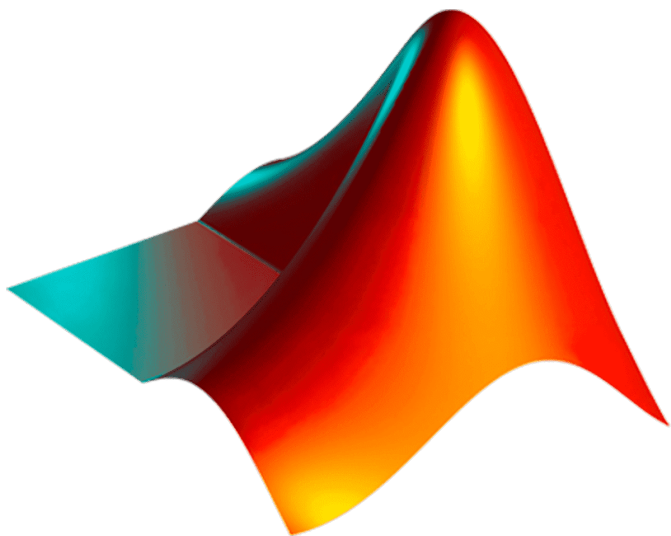
Step·03

功能测试 & 改进

对程序的主体功能进行测试，并对出现的一些问题进行针对性的改进。

#2 方法及过程

基于MATLAB R2024b



MATLAB®

&



deepseek



#2 方法及过程

功能测试 & 改进（基于MATLAB R2024b）



MATLAB多普勒效应演示程序开发



已深度思考 (用时 51 秒)

以下是一个基于MATLAB的多普勒效应演示程序，包含图形界面、声音生成、频率分析和实时绘图功能。请按照以下步骤操作：

```
matlab 复制
function DopplerDemo
% 创建主界面
fig = figure('Name','多普勒效应演示','NumberTitle','off',...
    'Position',[400 200 800 600],'Color',[0.9 0.9 0.9]);

% 初始化参数
fs = 44100;           % 采样率
soundSpeed = 343;     % 声速 (m/s)
maxDuration = 10;     % 最大录制时长 (秒)
bufferSize = 1024;    % 音频缓冲区大小

% 创建控件
uicontrol('Style','text','Position',[20 550 150 20],...
    'String','基频 (Hz):','HorizontalAlignment','left');
freqSlider = uicontrol('Style','slider','Position',[20 520 200 30],...
    'Min',200,'Max',2000,'Value',1000,'SliderStep',[1/1800 100/1800]);
startBtn = uicontrol('Style','pushbutton','Position',[20 450 90 30],...
    'String','开始','Callback',@startRecording);
stopBtn = uicontrol('Style','pushbutton','Position',[130 450 90 30],...
    'String','停止','Enable','off','Callback',@stopRecording);

% 创建绘图区域
ax = axes('Parent',fig,'Position',[0.3 0.1 0.65 0.8],...
    'FontSize',10,'Box','on','GridLineStyle','--');
```

给 DeepSeek 发送消息

深度思考 (R1)

联网搜索

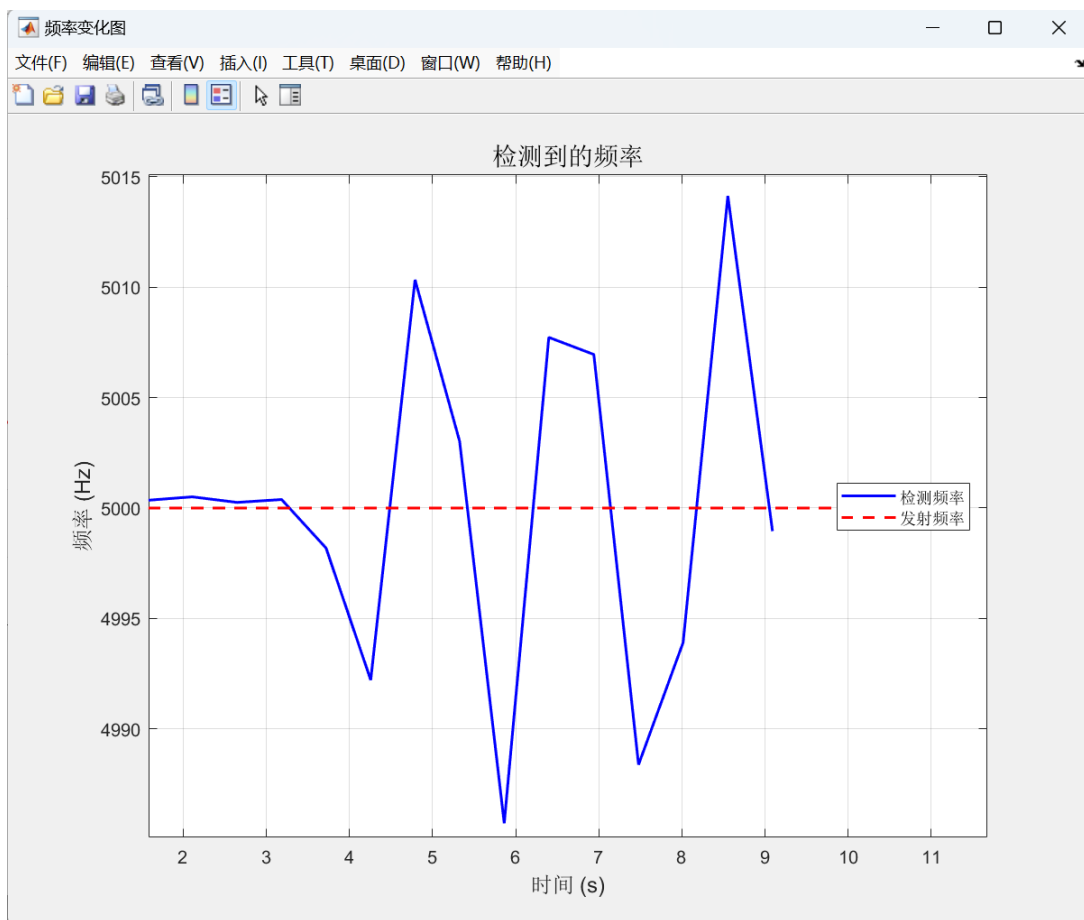


内容由 AI 生成，请仔细甄别



#2 方法及过程

功能测试 & 改进（基于MATLAB R2024b）



图形并不光滑

在检测开始时频率从0Hz开始

...

询问deepseek

给出方案以及对应代码

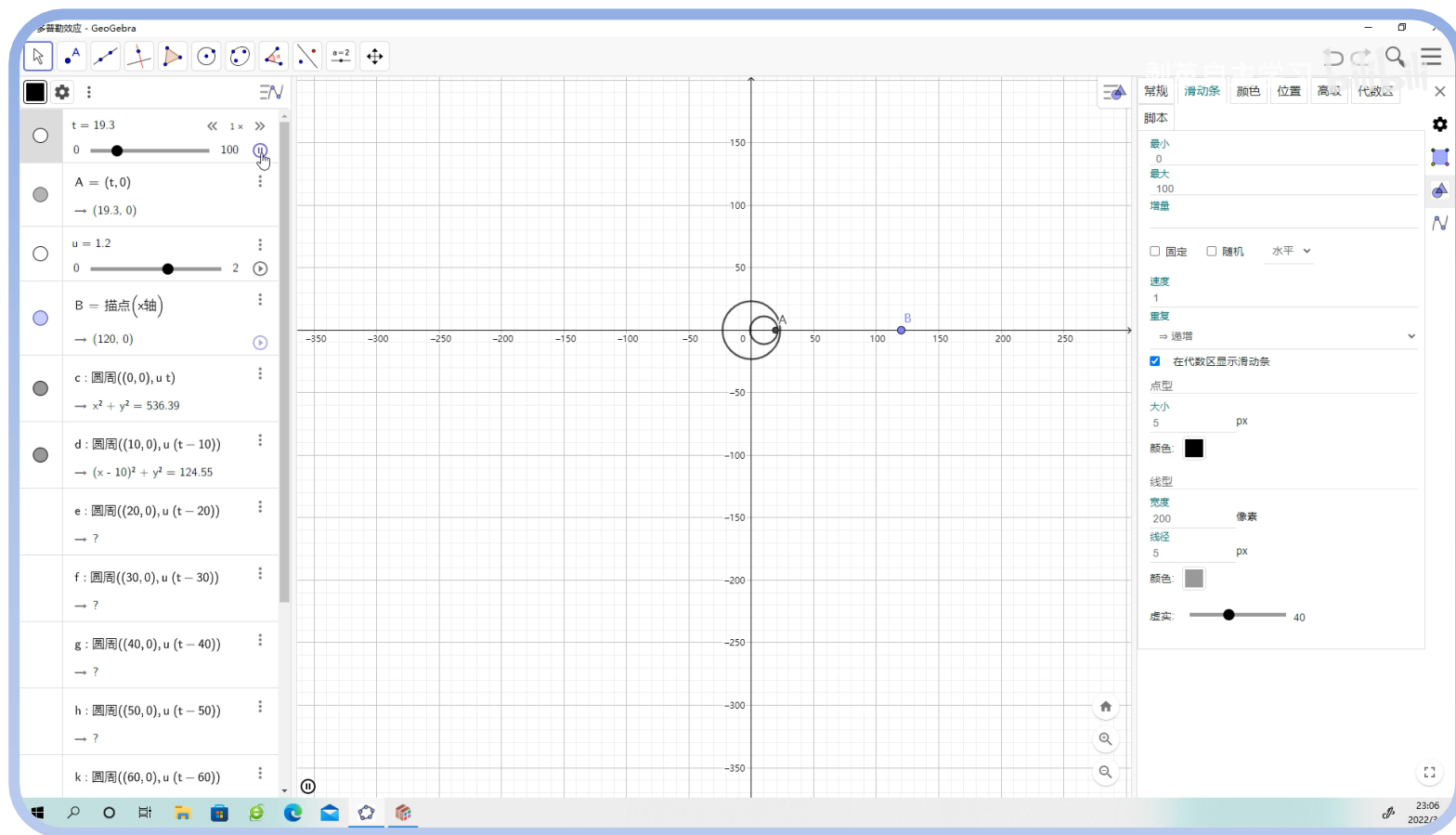
测试并发现问题

代码展示

```
1 function ceshi2
2     % 多普勒效应演示程序
3
4     % 清除工作区和关闭所有图形窗口
5     clearvars; close all; clc;
6
7     % 创建图形用户界面 (GUI)
8     fig = figure('Name', '多普勒效应演示', 'NumberTitle', 'off', ...
9         'Position', [100, 100, 800, 700], 'Resize', 'off');
10
11     % 默认发射频率 (Hz)
12     defaultFreq = 5000; % 增加默认频率至5000Hz
13
14     % 创建频率输入框和标签
15     uicontrol('Style', 'text', 'Position', [50, 650, 120, 25], 'String', '频率 (Hz):', ...
16         'HorizontalAlignment', 'left', 'FontSize', 12);
17     freqEdit = uicontrol('Style', 'edit', 'Position', [180, 650, 120, 30], ...
18         'String', num2str(defaultFreq), 'FontSize', 12);
19
20     % 创建“开始”和“停止”按钮
21     startButton = uicontrol('Style', 'pushbutton', 'Position', [50, 570, 150, 50], ...
22         'String', '开始', 'FontSize', 14, 'Callback', @startCallback);
23     stopButton = uicontrol('Style', 'pushbutton', 'Position', [220, 570, 150, 50], ...
24         'String', '停止', 'FontSize', 14, 'Callback', @stopCallback, 'Enable', 'off');
```

#2 方法及过程

多普勒动画演示-GeoGebra

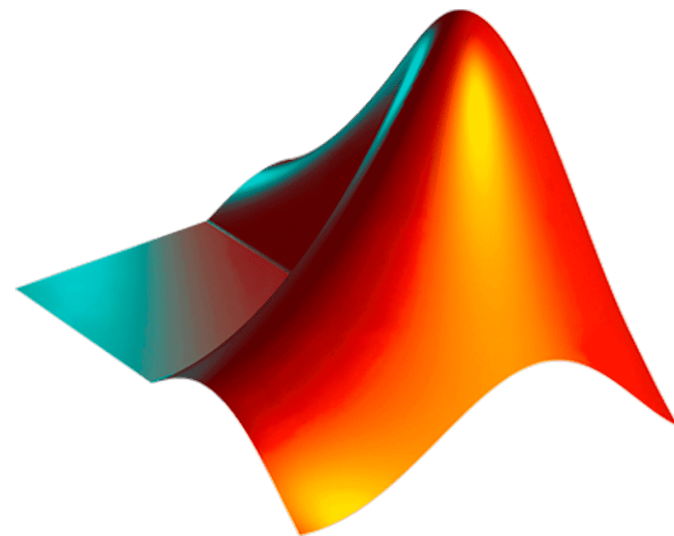
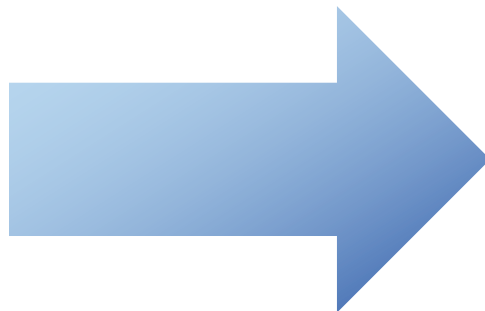


手机扫码观看/分享
可扫码观看完整视频

视频来源于网络

#2 方法及过程

GeoGebra

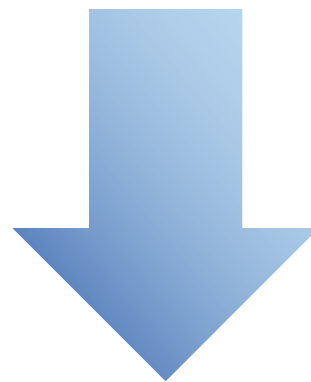


MATLAB®

#2 方法及过程



想法

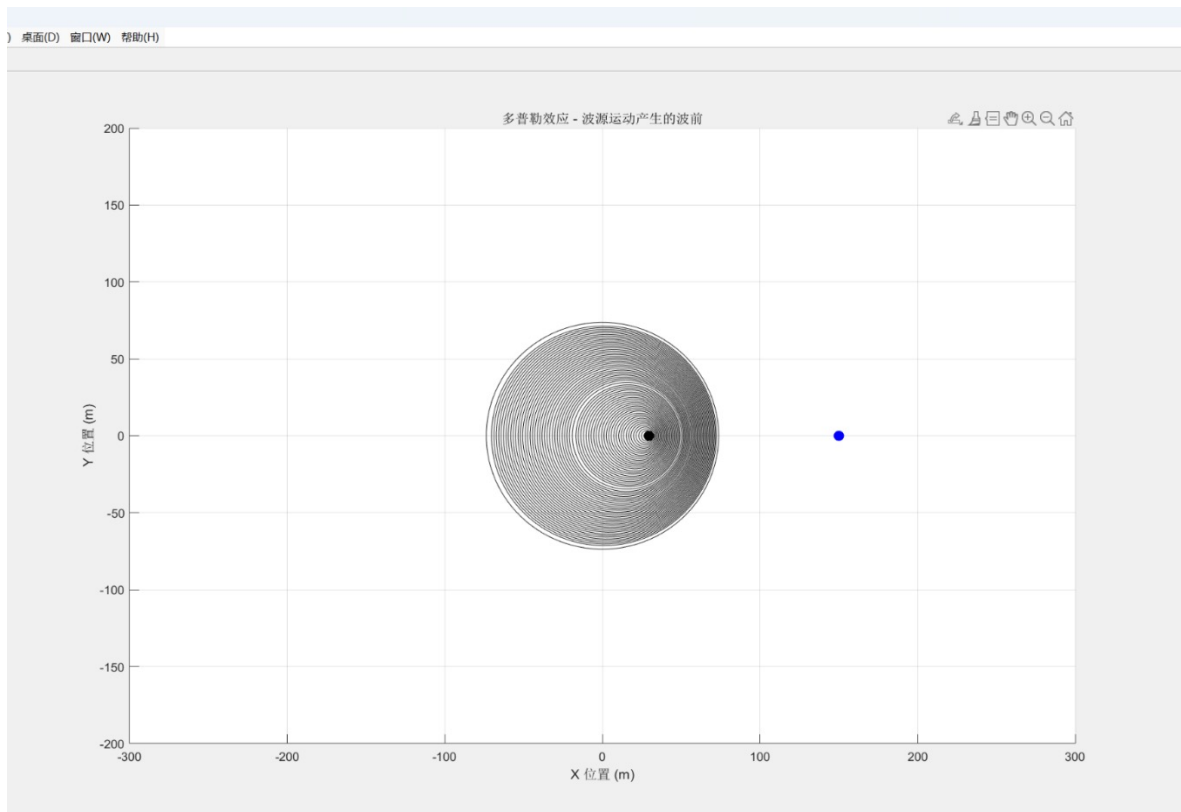


落地



#2 方法及过程

功能测试 & 改进（基于MATLAB R2024b）



图形过于密集
效果不明显
...

询问deepseek

给出方案以及对应代码

测试并发现问题

代码展示

```
1 function tuxingyanshi()
2     % 多普勒效应模拟
3     % 情况1: 声源运动, 两个静止的观察者
4     % 情况2: 声源静止, 一个运动的观察者
5
6     % 清除变量和命令窗口
7     clc;
8     clear;
9
10    %% 参数设置
11    c = 5;                % 波速 (m/s)
12    v_initial = 2;        % 初始速度 (m/s)
13    duration = 20;        % 模拟持续时间 (秒)
14    dt = 0.02;            % 时间步长 (秒)
15    epsilon = 1e-3;       % 防止除以零的小值
16    max_frequency = 1000; % 最大频率显示值 (Hz)
17
18    %% 初始位置
19    source_pos = [0, 0];  % 声源初始位置
20    observer1_pos = [40, 0]; % 观察者1初始位置 (情况1)
21    observer2_pos = [-40, 0]; % 观察者2初始位置 (情况1)
22    observer3_pos = [40, 0]; % 观察者3初始位置 (情况2)
23
24    %% 创建图形窗口
25    fig = figure('Name', '多普勒效应可视化', 'NumberTitle', 'off', ...
```



#03

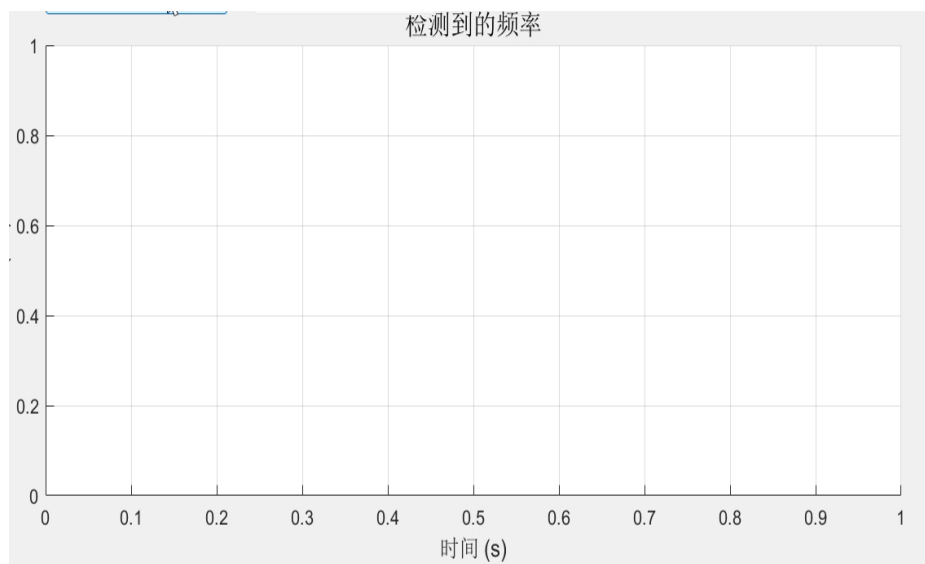
成果展示及应用

Display and Application



#3 成果展示及应用

效果对比



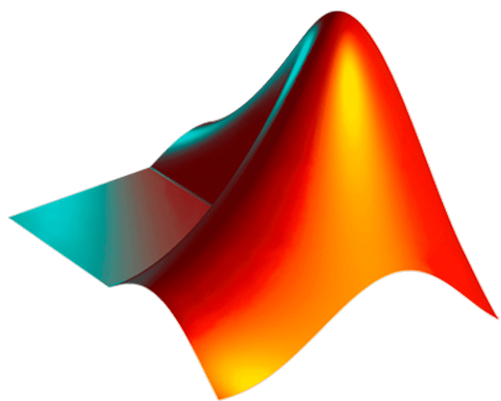
MATLAB®



phyphox

#3 成果展示及应用

效果对比



MATLAB[®]
多普勒演示程序



phyphox
多普勒效应功能

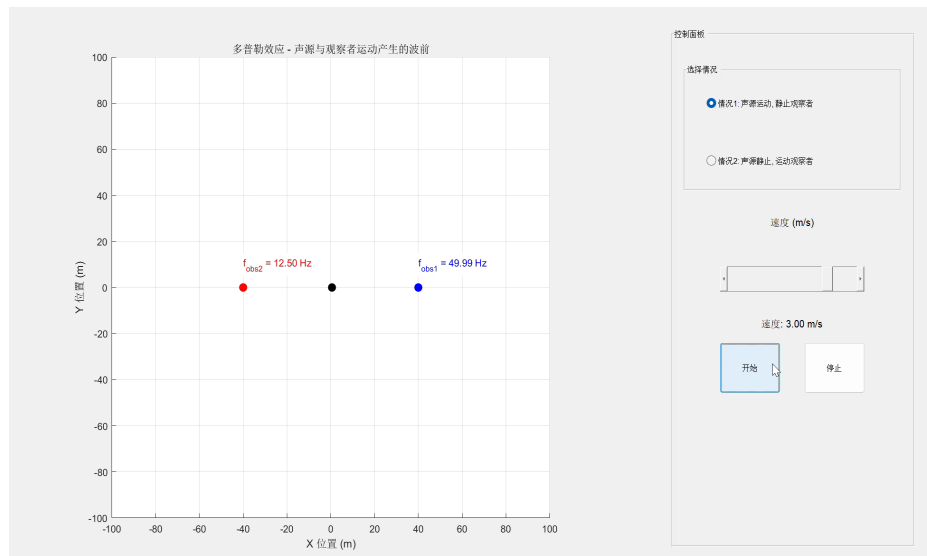
创新点

1. 图像更加平滑，更易于观察
2. 受环境声的影响大大减小
3. 可以自动进行参数设置及计算
4. 只需要一个设备就可以完成实验
5. 为图像添加了基准线，方便对照
6. 图像可保存到电脑上，方便研究

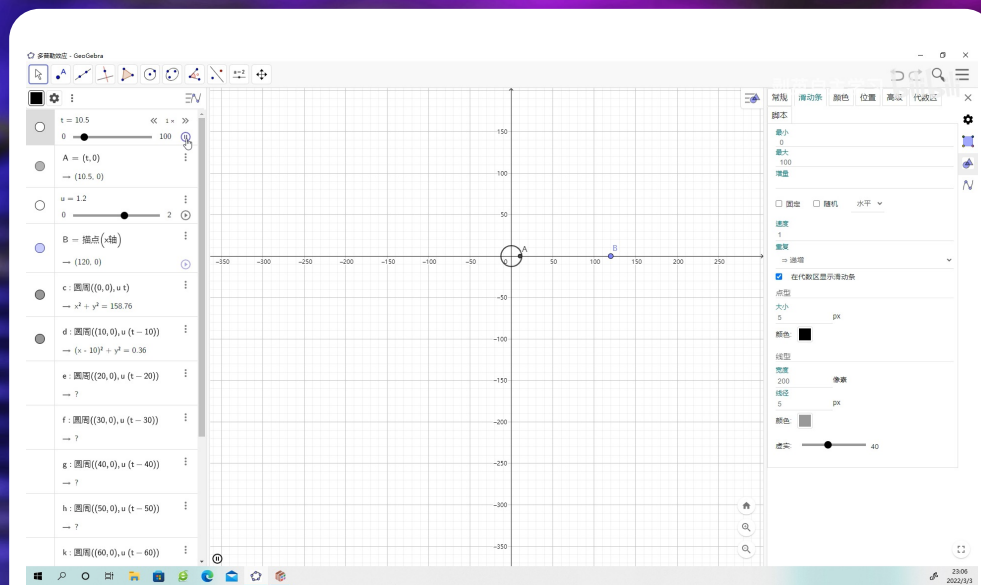


#3 成果展示及应用

效果对比



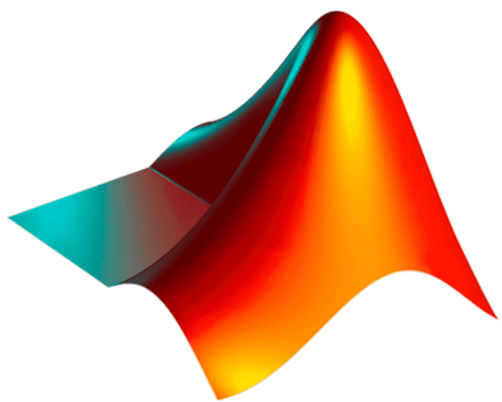
MATLAB®



几何画板

#3 成果展示及应用

效果对比



MATLAB®

多普勒效应动画演示程序

GeoGebra

几何画板

多普勒效应动画演示函数

创新点

1. 分别设计了两种情况（即观察者运动和声源运动）
2. 为观察者设计了频率显示，能够显示接收到的频率，更加直观
3. 以滑块形式改变移动速度，更加方便
4. 增加操作按钮，更易于操作



#3 成果展示及应用



成果应用

01

工程应用与设计测试

在声学工程和产品开发中，例如声纳系统或风噪检测，模拟多普勒效应可以帮助设备在不同条件下的性能，从而优化系统设计。

02

科普用途

用于科普展览和演讲中，向公众展示声音传播过程，通过实时的频率图变化，通过直观的可视化帮助观众理解多普勒效应。

03

科学实验与研究

在科学实验中，可以作为基础的工具来测量频率的变化，帮助研究人员研究运动对声波频率的影响，并测量移动速度等数据。

04

物理课堂教学

用于中学或大学的物理课堂上，作为演示工具帮助学生理解多普勒效应的概念，包括声源、观察者、波前的关系等。



#04

改进构思及方案

Improvement ideas and proposals

改进构思及方案

1. 进一步提高准确率以及演示效果
2. 制作为.exe软件方便使用
3. 设计更精美的操作面板
4. 为软件加入使用方法注释



```
max: function(e, t, n) {  
    var r, i = 0;  
    o = e.length;  
    n = n(e);  
    if (n) {  
        if (a) {  
            for (; o > i; i++)  
                if (r = t.apply(e[i], n), r >= i) break;  
        } else {  
            for (i in e)  
                if (r = t.apply(e[i], n), r >= i) break;  
        }  
    } else if (a) {  
        for (; o > i; i++)  
            if (r = t.call(e[i], i, e[i]), r >= i) break;  
    } else {  
        for (i in e)  
            if (r = t.call(e[i], i, e[i]), r >= i) break;  
    }  
    return e  
},  
trim: b && !b.call("\uffff\u00a0") ? function(e) {  
    return null == e ? "" : b.call(e)  
} : function(e) {  
    return null == e ? "" : (e + "").replace(C, "")  
},  
makeArray: function(e, t) {  
    var n = t || [];  
    return null != e && (N(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : b.call(n, e)), n  
},  
isArray: function(e, t, n) {  
    if (!t) {  
        if (a) return e.call(t, e, n);  
        for (r = t.length, n = n ? 0 > n ? Math.max(0, r + n) : n : 0; r > n; n++)  
            if (n in t && t[n] == e) return n  
    }  
    return !t || e[n] == e ? n : n  
}
```

THANKS

For Listening

