Ses Kaynağı Konumlandırma Simülasyonu Proje Raporu

# 1. Projenin Amacı ve Kapsamı

Bu proje, dört veya sekiz mikrofon kullanarak bir ses kaynağının konumunu tahmin etmeye yönelik bir simülasyon gerçekleştirmektedir.   
Amaç, ses kaynağının bilinen mikrofonlara olan mesafe ve zaman farklarını (TDOA - Time Difference of Arrival) kullanarak konumunu hesaplamaktır.   
Bu tür bir simülasyon, çeşitli akustik sistemlerde, güvenlik alanlarında ya da bilimsel deneylerde kullanılabilir.   
Ses kaynağının konumlandırılması, özellikle çok mikrofonlu sistemlerde, çevresel ses izleme ve konumlandırma sistemlerinde yaygın olarak kullanılır.

# 2. Projenin İşleyişi ve Arayüz

Proje, PyQt5 kütüphanesi kullanılarak geliştirilmiş bir grafik kullanıcı arayüzüne (GUI) sahiptir.   
Bu arayüzde, kullanıcı fare ile belirli bir noktaya tıklayarak ses kaynağını simüle edebilir.   
Tıkladığı bu nokta, mikrofonlar arasındaki zaman farkına göre yeşil bir tahmin noktasıyla eşleştirilmektedir.   
Matplotlib kütüphanesi ile mikrofonların, ses kaynağının ve tahmini ses kaynağının görsel gösterimi sağlanmaktadır.   
Kullanıcı ayrıca, mikrofonların yerleşimini sağ tıkla sürükleyerek değiştirebilir.

# 3. Mikrofonların Yerleşimi ve Konumlandırılması

Proje, 4 veya 8 mikrofon kullanarak ses kaynağının konumunu belirler.   
Mikrofonlar kare düzeninde yerleştirilmiş olup varsayılan konumlar şunlardır:  
- 4 mikrofonlu yapı: köşe mikrofonlar (0,0), (10,0), (0,10), (10,10) koordinatlarında.  
- 8 mikrofonlu yapı: köşe mikrofonlara ek olarak (5,0), (0,5), (10,5), (5,10) noktalarında mikrofonlar bulunur.  
  
Mikrofonlar kullanıcı tarafından ekranda sağ tıklanarak yeni pozisyonlara taşınabilir. Bu, dinamik ortamlar ve farklı düzenlerde mikrofonların yerleştirilmesi için esneklik sağlar.

# 4. Matematiksel Hesaplamalar ve Algoritmalar

Proje, Time Difference of Arrival (TDOA) algoritmasına dayanmaktadır.   
TDOA, ses kaynağının birden fazla mikrofona ulaşma süreleri arasındaki farkları kullanarak, ses kaynağının konumunu belirleyen bir yöntemdir.  
Her bir mikrofon ile ses kaynağı arasındaki mesafeyi hesaplamak için Öklid mesafesi formülü kullanılır:  
d = sqrt((x2 - x1)^2 + (y2 - y1)^2)  
  
Ses kaynağının her mikrofona ulaşma zamanı şu formülle hesaplanır:  
t = d / ses\_hızı  
  
Ardından, tüm mikrofon çiftleri arasındaki zaman farkları kullanılarak, ses kaynağının en uygun konumu Nelder-Mead optimizasyon yöntemi ile hesaplanır.  
Bu optimizasyon yöntemi, belirli bir kayıp fonksiyonunu minimize eder. Kayıp fonksiyonu, mikrofonlar arasındaki gerçek zaman farkları ile tahmini zaman farkları arasındaki farkları ifade eder.

# 5. Kullanılan Teknolojiler

Projede kullanılan başlıca teknolojiler şunlardır:  
- \*\*Python:\*\* Projenin temel dili olarak kullanılmıştır.  
- \*\*PyQt5:\*\* Grafik kullanıcı arayüzünün (GUI) oluşturulmasında kullanılmıştır.   
 PyQt5, kullanıcı tıklamaları, butonlar ve sürükleme işlemleri gibi etkileşimleri yönetir.  
- \*\*Matplotlib:\*\* Mikrofonların ve ses kaynağının görselleştirilmesi için kullanılmıştır.   
 Bu kütüphane ile ekranda grafiksel elemanlar çizilmiş ve etkileşimli olarak gösterilmiştir.  
- \*\*NumPy:\*\* Matematiksel işlemlerin, özellikle vektör ve matris işlemlerinin gerçekleştirilmesi için kullanılmıştır.  
- \*\*SciPy:\*\* Nelder-Mead optimizasyon algoritması SciPy kütüphanesi ile uygulanmıştır.   
 Bu kütüphane, mikrofonlar arasındaki zaman farklarını kullanarak en uygun ses kaynağı tahminini yapar.

# 6. Kodun İşleyişi ve Yapısı

Projedeki sınıfların ve fonksiyonların işleyişi şu şekildedir:  
  
1. \*\*SoundSourceLocalization Sınıfı:\*\* Projenin ana sınıfıdır ve GUI'nin tüm işlevlerini içerir. PyQt5 kullanılarak GUI oluşturulmuş ve kullanıcı etkileşimleri bu sınıf ile yönetilir.  
2. \*\*initUI():\*\* Kullanıcı arayüzünü başlatan fonksiyondur. Ana pencerede mikrofon yerleşimleri, butonlar ve sonuçların gösterileceği alanlar oluşturulur.  
3. \*\*calculate\_distance(mic\_pos, source\_pos):\*\* Mikrofon ve ses kaynağı arasındaki mesafeyi hesaplar. Öklid mesafesi formülüne dayanır.  
4. \*\*tdoa\_loss(source\_pos, mic\_positions, time\_stamps):\*\* Mikrofonlar arasındaki zaman farklarını kullanarak kayıp fonksiyonunu hesaplar.   
 Mikrofon çiftleri arasındaki tahmini ve gerçek zaman farklarını karşılaştırarak hatayı (residual) minimize eder.  
5. \*\*find\_sound\_source(mic\_positions, time\_stamps):\*\* Ses kaynağını optimize eden fonksiyondur.   
 Nelder-Mead optimizasyonu kullanılarak kayıp fonksiyonu minimize edilir ve en uygun ses kaynağı konumu hesaplanır.  
6. \*\*on\_click(event):\*\* Kullanıcı ekranda bir noktaya tıkladığında çalışan fonksiyondur.   
 Bu nokta ses kaynağı olarak kabul edilir ve ses kaynağının konumu hesaplanarak ekranda gösterilir.  
7. \*\*update\_plot():\*\* Ekrandaki grafikleri günceller. Mikrofonlar, ses kaynağı ve tahmini konumlar bu fonksiyonla güncellenerek ekranda çizilir.  
8. \*\*clear():\*\* Ses kaynağı ve tahmini konumları ekrandan silen fonksiyondur.  
9. \*\*reset\_mic\_positions():\*\* Mikrofonların varsayılan pozisyonlarına dönmesini sağlar.   
 Kullanıcı mikrofonları ekranda taşıdıktan sonra bu butona basarak mikrofonları sıfırlayabilir.

# 7. Görselleştirme ve Sonuçlar

Simülasyon, Matplotlib kullanılarak görselleştirilmiştir.   
Mikrofonlar mavi noktalarla, gerçek ses kaynağı kırmızı bir nokta ile ve tahmini ses kaynağı yeşil bir nokta ile gösterilir.  
Kullanıcı fare ile ekranda tıklama yaparak ses kaynağını belirleyebilir ve ardından sistem mikrofonlar arasındaki zaman farklarını kullanarak tahmini konumu hesaplar.  
Mikrofonların yerleri de sağ tıkla değiştirilebilir, bu da farklı konumlandırma senaryolarının test edilmesini sağlar.

5. \*\*find\_sound\_source(mic\_positions, time\_stamps):\*\* Ses kaynağını optimize eden fonksiyondur.   
 Nelder-Mead optimizasyonu kullanılarak kayıp fonksiyonu minimize edilir ve ses kaynağının en uygun tahmini yapılır.  
6. \*\*on\_click(event):\*\* Kullanıcı ekranda bir noktaya tıkladığında çalışan fonksiyondur. Bu nokta ses kaynağı olarak kabul edilir ve ses kaynağının konumu hesaplanarak ekranda gösterilir.  
7. \*\*update\_plot():\*\* Ekrandaki grafikleri günceller. Mikrofonlar, ses kaynağı ve tahmini konumlar bu fonksiyonla güncellenerek ekranda çizilir.  
8. \*\*clear():\*\* Ses kaynağı ve tahmini konumları ekrandan silen fonksiyondur.  
9. \*\*reset\_mic\_positions():\*\* Mikrofonların varsayılan pozisyonlarına dönmesini sağlar. Kullanıcı mikrofonları ekranda taşıdıktan sonra bu butona basarak mikrofonları sıfırlayabilir.

# 7. Görselleştirme ve Sonuçlar

Simülasyon, Matplotlib kullanılarak görselleştirilmiştir. Mikrofonlar mavi noktalarla, gerçek ses kaynağı kırmızı bir nokta ile ve tahmini ses kaynağı yeşil bir nokta ile gösterilir.  
Kullanıcı fare ile ekranda tıklama yaparak ses kaynağını belirleyebilir ve ardından sistem mikrofonlar arasındaki zaman farklarını kullanarak tahmini konumu hesaplar.  
Mikrofonların yerleri de sağ tıkla değiştirilebilir, bu da farklı konumlandırma senaryolarının test edilmesini sağlar.