

## AUFBAU EINES EINFACHEN SPRACHERKENNERS

### Versuch 4, Praktikum Technische Grundlagen der angewandten Informatik

In diesem Versuch wird ein einfacher Spracherkennung aufgebaut, der z.B. zur Steuerung eines Staplers in einem Hochregallager dienen könnte. Es reichen hierzu die Erkennung der vier einfachen Befehle "Hoch", "Tief", "Links" und "Rechts". Der Aufbau des Spracherkenners folgt dem in der Vorlesung beschriebenen Prinzip des Prototyp-Klassifikators. Die zugehörigen Spektren werden mit der Windowing-Methode berechnet. Wie immer sind die Vorüberlegungen, Berechnungen und Ergebnisse zum nächsten Praktikumstermin in Form eines Jupyter-Notebooks zu präsentieren. Bitte alle Programme in das Notebook einbauen und kommentieren.

Lernziele:

- Praktisches Verständnis der Fourierreihe und der Fourieranalyse
- Methoden der Frequenzmessung, insbesondere Windowing
- Mustererkennung mit Prototypen-Klassifikatoren

Für beide Aufgaben wird eine Mikrophon verwendet, das über einen Klinkestecker direkt mit der Soundkarte des Computers verbunden ist. Python stellt über das Paket **PyAudio** ein bequemes Interface zum Auslesen der Soundkarte zur Verfügung.

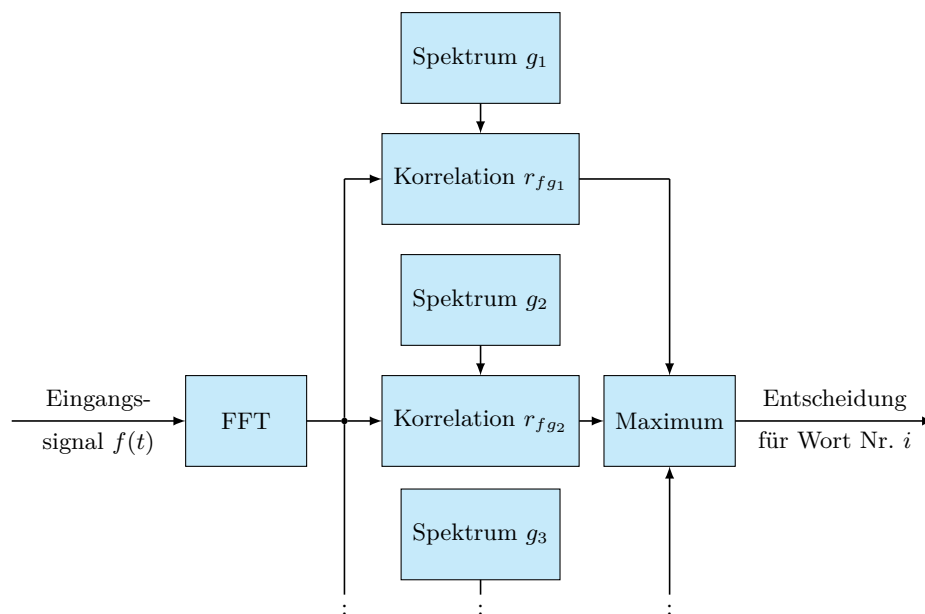
#### 1. Fourieranalyse lang andauernder Signale

- a) Schreiben Sie eine Python-Routine zur Aufnahme kurzer akustischer Signale über die Soundkarte mithilfe des Objektes **audiorecorder**. Orientieren Sie sich dabei an dem in Moodle zur Verfügung gestellten Codebeispiel (J.Kepler). Speichern Sie das Signal mit einem automatisch generierten eindeutigen Namen mithilfe des Befehls **numpy.save()**. Nehmen Sie als Beispiel eine beliebige Spracheingabe auf und stellen Sie diese mit korrekter Achsenbeschriftung als Diagramm dar.

- b) Ergänzen Sie Ihre Aufnahmeroutine um eine Triggerfunktion, die alle Samples nahe an 0 abschneidet, die vor dem eigentlichen Beginn des Signals aufgenommen wurden (s. Codebeispiel). Auf diese Weise stellen Sie sicher, dass alle Signale bei den späteren Vergleichsoperation zum gleichen Zeitpunkt beginnen. Schneiden Sie Ihr Signal ab dem Triggerzeitpunkt so aus, dass das ausgeschnittene Signal genau 1 s lang ist und füllen im Programm die möglicherweise fehlenden Samples mit Nullen auf. Überprüfen Sie die Korrektheit Ihrer Triggerfunktion durch erneute Darstellung des Signals als Diagramm.
- c) Bestimmen Sie mithilfe Ihres Codes aus Versuch 3 das Amplitudenspektrum des ausgeschnittenen Signals und stellen Sie dieses graphisch dar.
- d) Implementieren Sie die in der Vorlesung gezeigte Methode des Windowing. Zerlegen Sie dazu das Signal in Abschnitte mit einer Länge von 512 Samples, die sich jeweils zur Hälfte überlappen. Gewichten Sie die Samples in jedem Fenster mit einer Gaußschen Fensterfunktion, die so gewählt wird, dass die Fensterbreite 4 Standardabweichungen entspricht. Führen Sie in jedem Fenster eine lokale Fouriertransformation durch und mitteln Sie dann die Fouriertransformierte über alle Fenster. Berechnen Sie daraus wieder das Amplitudenspektrum und überprüfen Sie die Korrektheit durch Vergleich der graphischen Darstellung mit dem Spektrum aus der vorigen Teilaufgabe.

## 2. Spracherkennung

Architektur des Spracherkenners (aus der Vorlesung):



- a) Für den Spracherkennung brauchen wir zunächst die Referenzspektren für die vier Befehle "Hoch", "Tief", "Links" und "Rechts". Nehmen Sie dazu jeweils 5 Beispiele pro Befehl auf und berechnen Sie deren Spektren mit der Windowing-Methode aus Aufgabe 1. Achten

Sie darauf, dass die 5 Beispiele vom selben Sprecher stammen. Das Referenzspektrum erhalten Sie durch Mittelung über die 5 Spektren. Stellen Sie Ihre 4 Referenzspektren im Notebook graphisch dar.

- b) Nehmen Sie einen Testdatensatz mit jeweils 5 Beispielen für jeden Befehl vom selben Sprecher wie im Trainingsdatensatz auf. Achtung: hier dürfen Sie nicht dieselben Beispiele wie im Training verwenden, ansonsten überschätzen Sie im Test die Leistungsfähigkeit Ihres Spracherkenners. Nehmen Sie zum Vergleich einen weiteren Testdatensatz gleicher Größe auf, der aber von einem anderen Sprecher stammt.
  - c) Schreiben Sie eine Python-Routine zur Berechnung des Korrelationskoeffizienten nach Bravais-Pearson zum Vergleich zweier Eingabespektren (bitte nicht die bereits implementierte Lösung aus `scipy` o.ä. verwenden!). Testen Sie die Routine an ihren Referenzspektren: beim Vergleich identischer Spektren sollte die Korrelation 1 sein, bei verschiedenen Spektren nahe an 0.
  - d) Implementieren Sie den Spracherkenner nach der oben gezeigten Architektur und testen Sie ihn an beiden Datensätzen. Geben Sie jeweils an, wie viel Prozent der Testdatensätze korrekt bzw. falsch erkannt wurden (d.h. die *Detektions-* und die *Fehlerrate*) und diskutieren Sie das Ergebnis.
-