Echtzeit-Rahmenwerk zur Unterstützung der Evaluierung von Sprachkommunikationssystemen

Tobias Hübschen[1], Marco Gimm[1], Bastian Kaulen[1],
Gabriel Mittag[2], Sebastian Möller[2], Gerhard Schmidt[1]

[1] *Digitale Signalverarbeitung und Systemtheorie, 24143 Kiel, Deutschland, Email: thu@tf.uni-kiel.de*

[2] *Quality and Usability Lab, 10587 Berlin, Deutschland*

Einleitung

Bei der Evaluierung von Sprachkommunikationssystemen, insbesondere aber auch bei der Entwicklung der zugehörigen Einzelkomponenten, kommen hauptsächlich Signale zum Einsatz, welche entweder nicht die vorangegangene Signalverarbeitungskette berücksichtigen oder dies nur auf rein simulativer Basis tun. Um zumindest für einige Anwendungen realistischere Signale bereitzustellen, war das Hauptziel dieser Arbeit, die Beschaffung solcher Signale zu erleichtern. Das zu entwickelnde System sollte dabei echtzeitfähig, flexibel in der Anwendung sowie mobil einsetzbar sein, um so eine Unterstützung der Evaluierung von Sprachkommunikationssystemen zu ermöglichen.
Dieser Beitrag ist wie folgt strukturiert: Zunächst wird das von den Autoren entwickelte System in seiner Gesamtheit vorgestellt. Darauf folgt eine detailliertere Beschreibung des Echtzeit-Rahmenwerkes und der aktuell verwendeten Hardware. Abschließend werden einige Szenarien aufgezeigt, für welche die Verwendung des entwickelten Systems einen Vorteil bietet.

Gesmtsystem

Das System (Abb. 1) kann grundsätzlich in die folgenden vier Komponenten unterteilt werden: nahes/fernes Ende, Telefonkanal und Echtzeit-Rahmenwerk. Es wird verwendet, indem der Nutzer am nahen Ende mit seinem Telefonie-Endgerät den Telefonanschluss (fernes Ende) des Echtzeit-Rahmenwerkes anruft. Dieser Anruf wird vom Rahmenwerk automatisch entgegen genommen. Der Nutzer kann dann über die Endgerätetastatur auswählen, ob das am fernen Ende ankommende Audiosignal aufgenommen, oder am fernen Ende ein Audiosignal eingespeist werden soll. Die Kombination der beiden vorigen

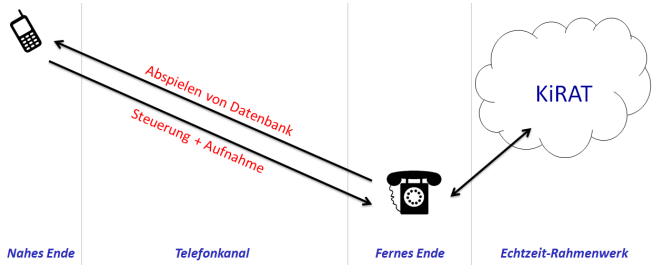

Abbildung 1: Grafische Übersicht über des Gesamtsystem mit nahem Ende, Telefonkanl, fernem Ende und Echtzeit-Rahmenwerk.

Optionen ist ebenfalls möglich. Sofern die Einspeisung eines Audiosignals angefordert wurde, muss das entsprechende Signal vom Nutzer aus einer hinterlegten Datenbank ausgewählt werden. Es ist zusätzlich möglich innerhalb des Echtzeit-Rahmenwerkes eine Vor- bzw. Nachverarbeitung der jeweiligen Signale zu aktivieren. Der Telefonkanal hingegen lässt sich nur durch die Position und Bewegung des Nutzers, sowie durch die Wahl der Netzbetreiber und der Endgeräte am nahen/fernen Ende beeinflussen.

Echtzeit-Rahmenwerk

Die notwendige Signalverarbeitung wurde in das *Kiel Real-Time Application Toolkit* (KiRAT) [1] eingebettet. Dieses Rahmenwerk wird seit 2010 am Lehrstuhl für Digitale Signalverarbeitung und Systemtheorie der Universität Kiel entwickelt. Die algorithmische Struktur ist modular aufgebaut, wobei die einzelnen Module in der Programmiersprache C geschrieben sind. Durch die Wahl dieser Sprache ist das Rahmenwerk auf verschiedenen Hardware-Plattformen echtzeitfähig. Die Grafische Nutzeroberfläche (GUI) ist in C++ unter der Verwendung der Qt Bibliothek [2] implementiert. In der Oberfläche ist zum einen der Signalfluss zwischen den Modulen ersichtlich, zum anderen lassen sich aber auch Signale in Echtzeit anzeigen oder Parameter ebenso in Echtzeit ändern. Aktuell beinhaltet des Rahmenwerk Signalverarbeitungsalgorithmen aus den Bereichen Audio, SONAR und Medizin und besitzt dafür eine Vielzahl an Schnittstellen für Sensoren/Aktoren.

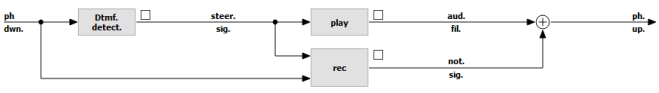

Abbildung 2: Grafische Nutzeroberfläche des neu implementierten Moduls im KiRAT.

Die neu implementierte Signalverarbeitung ist in drei Submodule unterteilt worden (Abb. 2): Die Abspiel- und Aufnahmefunktionalitäten wurden jeweils in eigenen Submodulen umgesetzt, das dritte Submodul fungiert als Steuerungseinheit. Die Grundlage für die Steuerungseinheit bildet ein Algorithmus zur Detektion von *Dual-Tone Multi-Frequency* (DTMF) Tönen [3]. Solche Töne werden standardmäßig bei Tastendruck von Telefonie-Endgeräten generiert und über den Telefonkanal übertragen. Jeder Taste ist dabei eine eindeutige Kombination zweier Sinustöne zugeordnet (Tabelle 1).

Tabelle 1: Definition der DTMF-Töne nach ITU-T Rec. Q.23

Frequenz	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	∗	0	#	D

Somit werden vom nahen Ende kodierte Befehle an die Steuerungseinheit im Rahmenwerk gesendet, welche von der Steuerungseinheit dekodiert und in Anweisungen an die anderen Submodule umgesetzt werden. Die Dekodierung der Befehle ist dabei unter Zuhilfenahme eines Zustandsautomaten implementiert. Darüber hinaus besitzt die Steuerungseinheit eine Querverbindung zur Steuerung der Hardwareschnittstelle.

Hardware

Die Hardware am fernen Ende kann prinzipiell flexibel nach den Ansprüchen der gewünschten Anwendung des Systems gewählt werden. Hierbei muss allerdings sichergestellt sein, dass eine entsprechende Hardware Schnittstelle existiert. Für den Probebetrieb des Systems wurde eine Prototypenhardware entwickelt, für die die Platine eines *Siemens Optiset E* Festnetztelefons als Basis diente. Die Audiosignale werden hier analog abgegriffen und über ein Audiointerface dem Echtzeit-Rahmenwerk zur Verfügung gestellt. Zur Steuerung des Telefons aus dem Echtzeit-Rahmenwerk heraus wird ein Mikrocontroller verwendet. Dieser greift zum einen die Information über einen eingehenden Anruf von der Platine ab und leitet diese Information per serieller Schnittstelle an das Echtzeit-Rahmenwerk weiter. Zum anderen simuliert der Mikrocontroller über ein entsprechendes Signal das Abheben des Telefonhörers, wodurch ein Anruf angenommen werden kann. Die automatische Annahme eines Anrufes ist allerdings nicht auf dem Mikrocontroller implementiert sondern wird vom Echtzeit-Rahmenwerk initialisiert.
Die im vorherigen Verlauf beschriebene Hardware ist in Abbildung 3 zu sehen. Da durch diese Hardware insbesondere die Bandbreite des Telefonkanals begrenzt ist, ist es angedacht weitere Hardwareschnittstellen zu entwickeln. Eine Möglichkeit ist hier beispielsweise die Anbindung eines Smartphones über Bluetooth. Sobald das ferne Ende mobil ist, ist es dann ebenfalls sinnvoll das Echtzeit-Rahmenwerk auf einen mobilen digitalen Signalprozessor (DSP) zu portieren um allgemeine Mobilität für das System zu erreichen. Dieser Schritt ist für das Rahmenwerk ohne großen Mehraufwand umsetzbar.

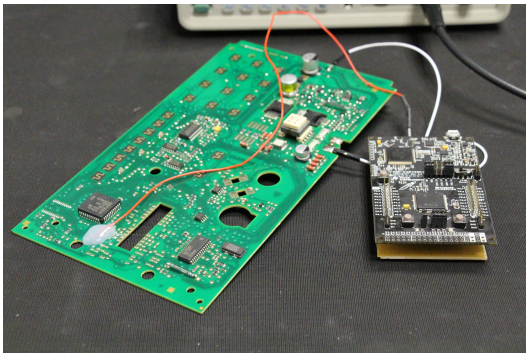

Abbildung 3: Modifiziertes *Siemens Optiset E* mit Mikrocontroller als Steuereinheit und Schnittstelle.

Anwendung

In den vorigen Abschnitten wurde das entwickelte System in seinen Einzelheiten beschrieben. In diesem Abschnitt soll nun aufgezeigt werden, für welche Anwendungen sich dieses System prinzipiell eignet.
Als grundlegendste Anwendung ist zunächst die Aufnahmefunktion an sich zu nennen. Die Aufnahme erfolgt hier unter Berücksichtigung des realen Telefonkanals und der Endgeräte am nahen/fernen Ende, wodurch die resultierenden Aufnahmen besonders für die Entwicklung empfangsseitiger Signalverarbeitungsalgorithmen interessant sind. Darüber hinaus unterstützt die Aufnahmefunktion noch Untersuchungen zum Echtzeit-Monitoring. Werden entsprechende Algorithmen in das Echtzeit-Rahmenwerk als Nachverarbeitung eingebunden, so können die Ergebnisse dieser Algorithmen mit den Ergebnissen klassischer Algorithmen, offline angewandt auf die aufgenommenen Signale, verglichen werden. Da der Nutzer des Systems mobil ist, lassen sich leicht verschiedene reale Geräuschszenarien in die Aufnahmen einbringen.
Die Abspielfunktion erlaubt das kontrollierte Einspeisen von Audiosignalen über den Telefonkanal in ein Endgerät. Diese Funktion bietet sich besonders zum Testen neuer experimenteller Algorithmen oder Komponenten an, welche hinter das Endgerät geschaltet sind. Dieses Szenario tritt besonders häufig in Kraftfahrzeugen auf, wo die Freisprecheinrichtung noch eine Vielzahl an Algorithmen zur Signalverbesserung enthält. Durch die Mobilität des nahen Endes lassen sich so Feldtests in wechselnder Umgebung mit variierender Verbindungsqualität durchführen. Die Signale lassen sich entweder direkt vor Ort subjektiv bewerten, können allerdings auch mittels eines Kunstkopfes für eine spätere Verwertung aufgenommen werden. Als relevante Algorithmen sollen hier beispielhaft die Bandbreitenerweiterung (BWE) [4] und das Near-End Listening Enhancement (NELE) [5] genannt sein. Insbesondere für NELE Algorithmen ist es wichtig Echtzeittests in der Zielumgebung durchzuführen, da diese Algorithmen eine umgebungsangepasste Signalverbesserung vornehmen und dementsprechend die resultierenden Signale auch in dieser Umgebung bewertet werden müssen. Da diese Tests nicht vollständig reproduzierbar sind, ersetzen sie natürlich nicht die standardisierten Testverfahren, welche unterschiedliche Systeme vergleichbar machen.
Die beiden grundlegenden Funktionen lassen sich ebenfalls kombinieren, sodass beispielsweise auch eine Echounterdrückung [6] analysiert werden kann. Wird das nahe Ende zusätzlich an eine Instanz des Echtzeit-Rahmenwerkes angeschlossen, so lassen sich darüber hinaus vollautomatisierte Testfolgen programmieren. Diese Tests berücksichtigen dann beide

Übertragungsrichtungen. Die Datenbanken, welche im System hinterlegt werden, sind grundsätzlich beliebig. Im Kontext der beschriebenen Anwendungen bieten sich beispielsweise auch unüblichere Testsignale wie Lombard-Sprache [7] an, was eine gestörte Umgebung am fernen Ende simulieren würde.

Zusammenfassung

In diesem Beitrag wurde ein System inklusive Echtzeit-Rahmenwerk beschrieben, welches die Evaluierung von Sprachkommunikationssystemen unterstützen kann. Durch die Hauptfunktionen der Aufnahme und des Abspielens von Audiosignalen lassen sich eine Vielzahl von Anwendungsszenarien definieren, für die so realitätsnahe Testbedingungen geschaffen werden. Da das verwendete Echtzeit-Rahmenwerk besonders flexibel bezüglich zusätzlicher Algorithmen und der Anbindung unterschiedlicher Hardware ist, ergibt sich ein Vorteil gegenüber anderer Ansätze wie beispielsweise Sprachserver. Obwohl das vorgestellte System reproduzierbare, standardisierte Testverfahren nicht ersetzen kann, verspricht der Einsatz des Systems einen direkten Mehrwert für die Entwicklung neuer Algorithmen.

Literatur

[1] KiRAT Overview, URL: `https://dss.tf.uni-kiel.de/index.php/research/realtime-framework/kirat-overview`

[2] Qt SDK Homepage, URL: `https://www.qt.io/`

[3] ITU-T Rec. Q.23, *Technical features of push-button telephone sets*, 1988

[4] Jax, P.: Enhancement of Bandlimited Speech Signals: Algorithms and Theoretical Bounds. Verlag Mainz, Aachen, 2002

[5] Sauert, B.: Near-End Listening Enhancement: Theory and Application. Wissenschaftsverlag Mainz, Aachen, 2014

[6] Vary, P. und Martin, R.: Digital Speech Transmission: Enhancement, Coding and Error Concealment. John Wiley & Sons, 2005, 505-561

[7] Lane, H. und Tranel, B.: The Lombard Sign and the Role of Hearing in Speech. Journal of Speech, Language and Hearing Research Vol. 14, 1971