System zur Simulation von kognitivem Feedback im Kontext auditiver Szenenanalyse und auditiver Qualitätsbeurteilung

Thomas Walther[1], Jens Blauert[1], Alexander Raake[2]

[1] *Institut für Kommunikationsakustik, Ruhr-Universität Bochum, Deutschland, Email: thomas.walther@rub.de*

[2] *Institut für Medientechnik, Technische Universität Ilmenau, Deutschland, Email: alexander.raake@tu-ilmenau.de*

Einleitung

Das Two!Ears-Projekt zielt darauf ab, bekannte Ingenieur-Modelle des binauralen Hörens durch einen systemisch-kognitiven Ansatz zu erweitern. Die zu modellierenden Personen werden hierbei als multimodale Agenten aufgefasst, die im Zuge explorativer Interaktion mit ihrem Umfeld interne Weltmodelle entwickeln. Folgerichtig werden in Two!Ears kognitiv motivierte Modelle der aktiven auditiven Wahrnehmung entwickelt. Anwendungsgebiete für solche Modelle liegen zum Beispiel im Bereich der dynamischen auditiven Szenen-Analyse (DASA) oder bei der instrumentellen Schätzung von Quality-of-Experience (QoE) in auditiv geprägten Szenarien [6], [7].

Dabei soll das Two!Ears-System einen strukturellen Bogen von der binauralen Wahrnehmung über die kognitive Beurteilung einer Szene hin zur aktiven Reaktion des robotischen Agenten schlagen. Für die Realisierung dieser Prozesskette sind audio-visuelle, motorische und sensorische Rückkopplungsmechanismen erforderlich. Diese werden in Form eines Expertensystems implementiert, welches Interaktionen von signalbasierten (bottom-up) und hypothesen-basierten (top-down) Modellkomponenten erlaubt.

In seiner aktuellen Ausbaustufe beinhaltet das Two!Ears-System u. a. eine einfache Blackboard-Architektur [4], innerhalb derer bereits einige relevante Rückkopplung realisiert wird. Der Fokus liegt hierbei auf der Integration multi-modaler Rückkopplungspfade und der Einbindung von Ansätzen zur aktiven Exploration audio-visueller Szenarien. Hierzu erweisen sich die Erstellung neuer Wissensquellen (*knowledge sources* [4] (KSs)) sowie die Einbindung von existierenden Experten-Subsystemen als wichtige Voraussetzungen.

In diesem Zusammenhang wurde eine Software-Erprobungsumgebung entwickelt, das *Bochum Experimental Feedback Testbed* (BEFT). Eine „abgespeckte“ Version davon, das *Lean Virtual Test Environment* (LVTE), wird im Folgenden etwas detaillierter vorgestellt. Das LVTE ermöglicht schnelle und verlässliche Überprüfungen grundlegender Rückkopplungsroutinen und verbindet sich nahtlos mit der aktuellen Blackboard-Architektur. Die Auralisierung gegebener Szenarien erfolgt im LVTE mit Hilfe des von der Universität Rostock bereitgestellten *SoundScape Renderer* (SSR) [5].

Das Lean-Virtual-Test-Environment

Im LVTE wurden im Vergleich zum mächtigeren BEFT u. a. folgende Vereinfachungen vorgenommen: Die visuelle Komponente verzichtet auf eine „high-end“-Darstellung der betrachteten Szene. Anstelle simulierter Kamera-Daten treten emulierte visuelle Stimuli, welche aus degradierten „ground-truth“-Szeneninformationen abgeleitet werden [1]. Das auf MATLAB® basierende LVTE dient, wie oben angeführt, der Durchführung einfacher Experimente zur multi-modalen Rückkopplung und aktiven Exploration. Für komplexere Tests kann das LVTE entweder durch das BVTE oder durch eine reale Roboter-Plattform ersetzt werden.

Systemüberblick

Das LVTE verbindet sich mit dem *robotConnect*-Interface der Blackboard-Architektur wie in Abb. 1 angedeutet. Über diese Schnittstelle können einzelne Wissensquellen auf die Antriebseinheiten des virtuellen Roboters zugreifen und so die Steuerung der robotischen Plattform sowie des darauf aufgesetzten beweglichen Kun-

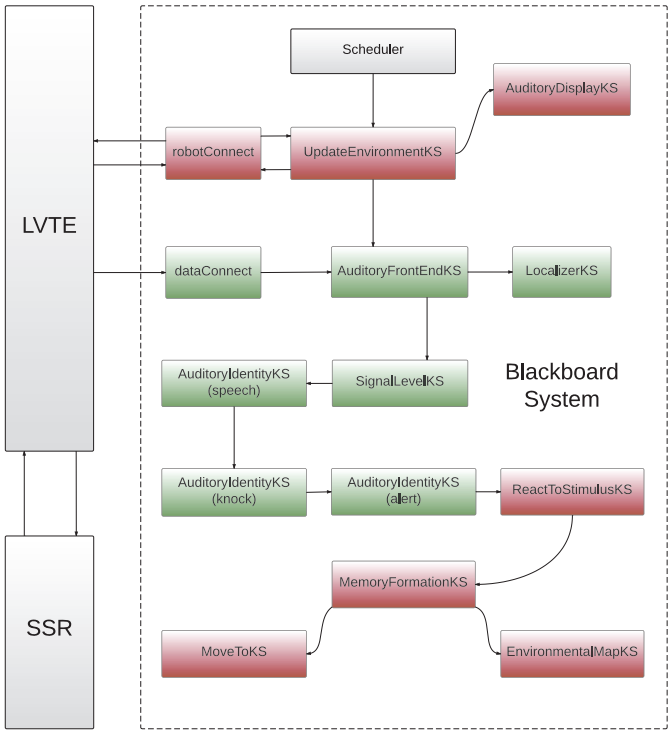

Abbildung 1: Das Blackboard-System für sequentielle Mehrquellen-Triangulation (vgl. Fließtext).

stkopfes übernehmen. Der Rückfluss von Umgebungsinformationen an die Wissensquellen erfolgt ebenfalls über das *robotConnect*-Interface und ermöglicht es den Expertensystemen, umweltbezogene Hypothesen zu generieren und Entscheidungen zu treffen. Die Synchronizität zwischen Blackboard-System und LVTE wird durch Einsatz einer speziellen Wissensquelle (*UpdateEnvironmentKS*) sicher gestellt.

Die direkte Kontrolle des SSR ermöglicht es dem LVTE darüber hinaus, die Ohrsignale des virtuellen Roboters zu generieren und diese zur weiteren Verarbeitung an die peripheren Elemente des Two!Ears-Systems zu senden – und zwar über die *AuditoryFrontEndKS*. Diese übernimmt die Vorverarbeitung der Signale und extrahiert relevante auditive Merkmale, welche dann an das Blackboard weitergereicht werden.

Abschließend sei angemerkt, dass die rein MATLAB®-basierte Struktur des LVTE nicht nur dessen einfachen Einbindung in die Two!Ears-Architektur erleichtert, sondern auch eine wichtige Voraussetzung für plattform-übergreifende Kompatibilität des Gesamtsystems darstellt. Im Folgenden wird näher auf die Klassenstruktur des LVTE eingegangen. Abb. 1 dient dabei als Diskussionsgrundlage.

‘Environment’-Klasse

Die *Environment*-Klasse stellt die Basis des LVTE dar. Sie erlaubt die programmatische Definition von Szenarien einfacher bis mittlerer Komplexität. Zur Zeit wird mit Szenarien in Rechteckräumen (*shoe-box* – Umgebung) experimentiert, wobei der Nachhall noch nicht berücksichtigt wird. Der Experimentator kann die Raumabmessungen manuell festlegen und den virtuellen Roboter frei im simulierten Umfeld platzieren. Die Dauer eines Szenarios lässt sich im Intervall von $[0 \dots T_S]$ einstellen. Alle im Auralisierungsprozess zur Zeit verwendeten Testschalle sind jeweils einer bestimmten *auditiven Kategorie* aus der IEEE sound database *AASP* [3] entnommen.

Angemerkt sei, dass die in den unten angeführten Experimenten verwendeten Testschalle aus dem kompletten zur Verfügung stehenden Korpus entnommen wurden. Dabei wurde zunächst wegen des „Proof-of-Concept“-Charakters der Experimente noch keine Rücksicht darauf genommen, ob eine Test-Datei bereits für das Training eines Quellenidentifikators verwendet worden war. Diese stark vereinfachende Vorgehensweise wird in Folgeexperimenten durch eine strikte Trennung zwischen Trainings- und Testdaten ersetzt werden. Ebenso werden in weiterführenden Tests umfangreichere Korpora zur Validierung der betrachteten Rückkopplungsmechanismen herangezogen werden. Mit diesen Ausführungen sei angenommen, dass $\mathcal{S}_A[j]$ Zugriff auf das $j^{\text{-te}}$ Element der Menge aller auditiven Kategorien $\mathcal{S}_A = \{\textbf{‘speech’}, \textbf{‘alert’}, \textbf{‘knock’}\}$ erlaubt. Dabei soll $C_j^A = \mathcal{S}_A[j]$ die $j^{\text{-te}}$ Kategorie aus der Menge extrahieren.

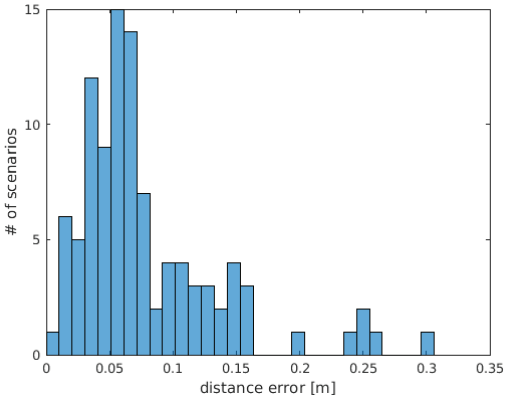

Abbildung 2: Experimentelle Ergebnisse der Triangulations-Experimente. Die Distanz-Fehler akkumulieren bei $0,0793\,\text{m}$ (vgl. Erläuterungen im Fließtext).

Für jede Kategorie in $\mathcal{S}_A$ kann eine Untermenge konkreter *Stimulus-Instanzen* definiert werden, wie das folgende Beispiel für die Kategorie „Sprache“ andeutet: $\mathcal{I}_{\textbf{‘speech’}} = \{\text{‘speech01.wav’}, \text{‘speech02.wav’}, ...\}$. Die zur Verfügung stehende Anzahl von Stimulus/Instanz-Mengen wird in Abhängigkeit von den zur Verfügung stehenden Testschallen kontinuierlich vergrößert werden. Zusätzlich zu den auditiven Kategorien/Instanzen verwaltet die *Environment*-Klasse eine Menge $\mathcal{S}_V$ von *visuellen Kategorien*, welche für durchzuführende audio-visuelle Experimente essentiell sind. Um eine Übereinstimmung zwischen auditiven und visuellen Stimuli auf der kognitiven Ebene zu erreichen, sei $\mathcal{S}_V = \{\textbf{‘person’}, \textbf{‘siren’}, \textbf{‘door’}\}$. Die Menge der visuellen Kategorien wächst dabei simultan mit $|\mathcal{S}_A|$.

Eine weitere Aufgabe der *Environment*-Klasse besteht in der Verwaltung der Menge aller beobachtbaren audio-visuellen Quellen. Diese können vom Experimentator frei im Szenario verteilt werden. Die Daten jeder platzierten Quelle werden zur Erstellung der auditiven Kulisse direkt an den SSR weitergereicht. Jede Quelle ist definiert durch eine Instanz der *Source*-Klasse, auf die im Folgenden eingegangen wird.

‘Source’-Klasse

Die *Source*-Klasse speichert grundlegende Informationen zu jeder im Szenario erstellten audio-visuellen Quelle. Die Quellen-Positionen werden zusammen mit den Namen der Quellen (ihrer jeweiligen *Identität*) und den emittierten Testschallen abgelegt. Die Testschalle können während der Dauer des Szenarios gewechselt werden, was die Simulation von Quellen mit mehreren *utterances* ermöglicht. Jede audio-visuelle Quelle, i, verwaltet eine *time-line*, $\mathcal{T}_i$, welche den onset und offset der emittierten Testschalle kontrolliert. Die Emissionen der Quelle werden dabei aus $\mathcal{I}_{C_i^A(t)}$ gewählt, wobei $C_i^A(t)$ die auditive Kategorie der Quelle i zum Zeitpunkt t darstellt.

Der resultierende *auditive Ablaufplan* wird von der *Environment*-Klasse verwendet, um die SSR-basierte Au-

ralisierung in Szenen einfacher und mittlerer Komplexität zu steuern. Die Ablaufpläne werden dabei entweder manuell generiert oder durch aufgabenspezifische Hilfsprogramme automatisch erzeugt. Allerdings bewegt sich die Erstellung visueller Ablaufpläne in engerem Rahmen als die der auditiven: Visuelle Kategorien aus $\mathcal{S}_V$ werden den einzelnen Quellen zu Beginn der Simulation zugewiesen, sie können während der Szenendauer dann allerdings nicht variiert werden. Sollte sich herausstellen, dass dynamische Änderungen der visuellen Kategorien während der Szenendauer relevant sind, so wird dies in späteren Ausbaustufen des Two!Ears-Systems berücksichtigt werden. Zusätzlich zur Liste aller audio-visuellen Quellen verwaltet die *Environment*-Klasse das unten beschriebene *RobotController*-Interface, welches Zugriff auf den virtuellen Roboter ermöglicht.

'RobotController'-Klasse

Die *RobotController*-Klasse stellt die virtuelle Repräsentanz des realen Roboters dar und erlaubt dem LVTE Zugriff auf die für das Two!Ears-Projekt relevanten Eigenschaften/Fähigkeiten der realen Platt form. Momentan verwaltet die *RobotController*-Klasse lediglich die Position des Roboters und stellt eine Schnittstelle zum aufmontierten Kunstkopf bereit. Jede Positionsänderung der robotischen Plattform wird an den SSR weitergeleitet, um die Synchronizität zwischen Auralisierung und der aktuellen Umweltsituation zu gewährleisten. In späteren Ausbaustufen des Systems könnte die *RobotController*-Klasse beispielsweise Informationen von Laser-Scannern oder anderen Sensoren integrieren, um so weitere Daten aus der virtuellen Umwelt zur Verfügung zu stellen.

'KemarHead'-Klasse

Die *KemarHead*-Klasse ermöglicht dem LVTE den Zugriff auf die Rotations-Steuerung des Kunstkopfes vom Typ KEMAR und verwaltet die für die SSR-Auralisierung benötigten *head-related transfer functions* (HRTFs). Momentan werden von der TU Berlin zur Verfügung gestellte HRTFs (azimutale Auflösung: 1 °, Distanz: 3 m) verwendet, andere HRTF-Datensätze können bei Bedarf ergänzt werden.

'Visualizer'-Klasse

Zur Darstellung des aktuellen Simulationszustandes in moderat-komplexen audio-visuellen Szenarien verwendet das LVTE eine spezielle *Visualizer*-Klasse. Diese stellt dem Experimentator eine vereinfachte 3D-Ansicht des gegebenen Szenarios zur Verfügung. Der Fokus liegt dabei nicht auf Geschwindigkeit und fotorealistischer Darstellung, sondern auf plattform-übergreifender Kompatibilität und einfacher Integrierbarkeit in das Two!Ears-Gesamtsystem. Zu diesem Zweck wird eine 3D-Darstellung erzeugt, welche vollständig auf MATLAB®-Routinen basiert. Dadurch kann auf komplexe 3D-Engines (wie z.B. OGRE [2]) verzichtete werden. Diese MATLAB®-basierten 3D-Repräsentationen stellen ungeachtet ihrer Einfachheit ein Werkzeuzg dar, mit dem man in grundlegenden Experimenten das Verhalten des Roboters während der Szenendauer gut nachverfolgen kann.

Aktive Exploration – Umgebungskarten

Die Architektur des Two!Ears-Systems soll neben der Formierung audio-visueller Objekte und der Erprobung multi-modaler Rückkopplungsschleifen die aktive Erforschung der gegebenen Szenarien ermöglichen. Ein fundamantale Experiment in der letztgenannten *active exploration*-Domäne stellt das folgende Triangulations-Szenario dar, in dem der Roboter durch aktive Bewegung im Umfeld die Positionen aller ihn umgebenden Schallquellen in der x/y-Ebene bestimmen soll.

Es sei angenommen, dass $\mathcal{Q} = [\mathbf{q}_1, ..., \mathbf{q}_{N_Q}]$ die Menge aller aktiven, statischen Schallquellen im LVTE darstellt. Weiterhin sei $\mathbf{p}_i^{GT}$ die „ground-truth"-Position der Quelle i in der azimutalen Ebene. Es existiere eine bijektive Korrelation zwischen den physikalischen Schallquellen und den von diesen Quellen emittierten Signalen wie folgt: Wenn $\mathbf{q}_i$ einen Stimulus der Kategory C_i^A aussendet, stammen die Stimuli aller anderen Quellen ausschließlich aus der Menge $\mathcal{S}_A \setminus C_i^A$. Schließlich sei $\mathbf{r}_t$ die Position des Roboters zur Zeit t.

Die Quellen in $\mathcal{Q}$ werden sequentiell aktiviert, nämlich gemäß $\mathbf{q}_1 \rightarrow \mathbf{q}_2 \rightarrow \ldots \mathbf{q}_{N_Q} \rightarrow \mathbf{q}_1 \rightarrow \ldots$. Die Intervalle zwischen aufeinander folgenden Aktivierungen entsprechen 0,2 s. In folgenden Systemversionen werden verbesserte Lokalisierungs-/Segmentierungschemata evaluiert werden, die eine gleichzeitige Aktivierung mehrerer Quellen erlauben.

Experimente zur Mehrfach-Kreuzpeilung

Zur Bestimmung der Peilgenauigkeit der vorgeschlagenen sequentiellen Mehr-Quellen-Triangulation wurden $N_T = 100$ einfache Testszenarien (ohne Nachhall, Abmessungen 10 m x 10 m) erstellt. Es sei angenommen, dass $N_Q = 3$ Quellen, $\mathbf{q}_{1...N_Q}$, in jeder generierten Szene zufällig räumlich verteilt sind. Dabei werden die gewählten Zufallspositonen $\mathbf{p}_{1,...,N_Q}^{GT}$ auf einen konzentrischen Ring mit einem inneren Radius von $r_i = 2\,\text{m}$ und einem äußeren Radius von $r_o = 4\,\text{m}$ beschränkt. Die minimale azimutale Differenz zwischen zwei benachbarten Quellen sei Δ_ϕ Grad. Zu Beginn jeder Simulation befindet sich der Roboter im Zentrum des Annulus bei $\mathbf{r} = [5, 5]^T$. Diese Szenarien-Definition erlaubt es dem Roboter, sich frei innerhalb des Perimeters r_i zu bewegen und stellt sicher, dass die einzelnen Quellen nicht zu nahe am Rand der Szene platziert werden.

Man beachte, dass die aktuelle Bewegungs-Strategie des Roboters für eine einfache „one-step"–Triangulation aus-

gelegt ist. Dabei lokalisiert der robotische Agent zunächst alle akustisch erfassbaren Schallquellen. Basierend auf den Azimut-Winkeln der erfassten Quellen errechnet das System danach einen linearen Pfad, welcher die Plattform orthogonal zu den gemessenen Quellenrichtungen verfährt. Anschließend wiederholt sich die Quellenlokalisierung und das Triangulations-System bestimmt die Lage der einzelnen Schallquellen in der x/y-Ebene. Indem Δ_ϕ auf 30 ° gesetzt wird, ist sichergestellt, dass die Lokalisierungkomponente des Systems die azimutale Position jeder Schallquelle korrekt disambiguiert. Die Aktivierung der einzelnen Quellen erfolgt sequentiell.

Das Ergebnis des Triangulations-Verfahrens für die geschätzten Positionen der einzelnen Quellen sei $\mathbf{p}^E_{1\ldots N_Q}$. Der Positionsschätzfehler ist damit gegeben durch

$$Err_i^{Pos} = \frac{1}{N_Q} \sum_{j=1}^{N_Q} \left|\left|\mathbf{p}_j^E - \mathbf{p}_j^{GT}\right|\right| . \qquad (1)$$

Abb. 2 zeigt die Verteilung von Err_i^{Pos} für alle N_T Szenarien.

Die Positionsfehler akkumulieren um 0, 0793 m, mit einer Standardabweichung von 0, 0578 m. Diese Werte deuten auf eine akzeptable globale Präzision der vorgeschlagenen Mehrquellen-Triangulationsmethode. Allerdings legt die verwendete Fehlermetrik in (1) keine Ausreißer in Bezug auf einzelne Quellen offen. Tatsächlich existieren solche Ausreißer. Diese können allerdings durch teilweise geminderte Signalqualitäten oder Quellenkonfigurationen erklärt werden, für welche das „one-step"-Triangulationsverfahren nicht geeignet ist.

Das Bochum-Virtual-Test-Environment

Das BVTE setzt die Entwicklungslinie fort, die im Rahmen des BEFT begonnen wurde. Es ersetzt dabei die einfache Visualisierungs-Komponente des LVTE durch eine frei verfügbare 3D-Engine. In der resultierenden virtuellen Umgebung kann der simulierte Roboter sich frei bewegen, so wie in Abb. 3 skizziert. Die verwendete 3D-Engine setzt sich zusammen aus der *Blender*–Visualisierungsumgebung [8] und dem Roboter-Simulationssystem MORSE [9]. Die robotische Einheit

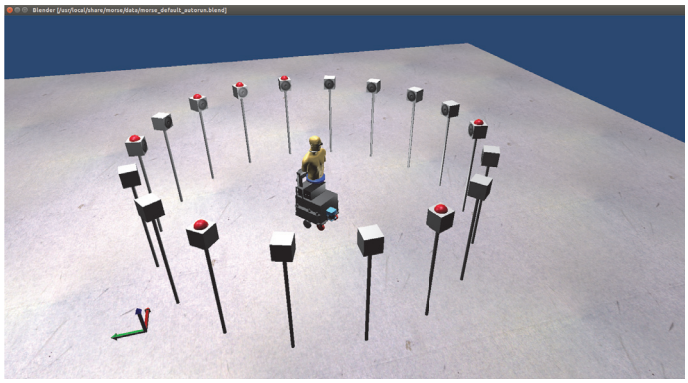

Abbildung 3: BVTE – exemplarisches Szenario

wird in der Simulation durch eine virtuelle Replik einer JIDO Plattform realisiert, welche Bewegungen und Rotationen in der x/y-Ebene ermöglicht. Auf der Plattform ist eine KEMAR–*Head-and-Torso–Struktur* montiert, die eine z–Achsen-Rotation des Kunstkopfes erlaubt. Die virtuelle Umgebung lässt den Einsatz eines artifiziellen Stereo-Kamerasystems zu, welches mittels einer maßgefertigten Brille am Kunstkopf befestigt wurde. Mit diesem virtuellen Kamerapaar können visuelle Merkmale aus den gegebenen Szenaren extrahiert und vom System verarbeitet werden. Erste Experimente in diesem audiovisuellen Simulationssystem zeitigen vielversprechende Ergebnisse.

Danksagung

Die Arbeiten zu diesem Artikel wurden im Rahmen des EU-Projekts TWO!EARS durchgeführt (ICT-618075, `www.twoears.eu`)

Literatur

[1] Walther, Th., Cohen-L'hyver, B.: Multimodal feedback in auditory-based active scene exploration, Proc. Forum Acusticum 2014, Krakow, 2014.

[2] OGRE - Open Source 3D Graphics Engine, `http://www.ogre3d.org/`, 2014

[3] IEEE AASP Challenge, University of London – School of Electronic Engineering and Computer Science, `http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/`, 2015

[4] Schymura, C., Walther, Th., Kolossa, D., Ma, N., Brown, G.J.: Interaural sound source localisation using a Bayesian-network-based blackboard system and hypothesis-driven feedback, Proc. Forum Acusticum 2014, Krakow, 2014.

[5] Geier, M., Spors, S., Ahrens, J.: The SoundScape Renderer: a unified spatial audio reproduction framework for arbitrary rendering methods, 124th AES Convention, Amsterdam, 2008.

[6] Blauert, J., Kolossa, D., Obermayer, K., Adiloğlu, K.: Further challenges and the road ahead. In J. Blauert (ed.), The technology of binaural listening, Springer, Berlin–Heidelberg and ASA Press, New York NY, 2013

[7] Raake, A., Blauert, J., Braasch, J., Brown, G., Danés, P., Dau, T., Gas, B., Argentieri, S., Kohlrausch, A., Kolossa, D., Le Goff, N., May, T., Obermayer, K., Spors, S. TWO!EARS integral interactive model of auditory perception and experience, DAGA 2014, Oldenburg, 2014

[8] Blender Foundation: Blender – 3D open source animation suite, `http://www.blender.org/`, 2014

[9] LAAS-CNRS: MORSE, the Modular Open Robots Simulation Engine, `https://www.openrobots.org/wiki/morse/`, 2014