Audio-driven upper-body motion synthesis on a humanoid robot

Jan Ondras
Trinity College

A dissertation submitted to the University of Cambridge
in partial fulfilment of the requirements for
Computer Science Tripos, Part III

University of Cambridge
Department of Computer Science and Technology
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom

Email: jo356@cam.ac.uk

May 31, 2018

Declaration

I Jan Ondras of Trinity College, being a candidate for Computer Science Tripos, Part III, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: 11,966

Signed:

Date:

Acknowledgements

I would like to thank to my project supervisor Dr Hatice Gunes for all their advice, help, feedback, and support during this project. I would like to also express my gratitude to all the participants in the web-surveys who helped me to evaluate my project.

Audio-driven upper-body motion synthesis on a humanoid robot

Abstract

Body language is an important aspect of human communication which an effective human-robot interaction should mimic well. However, the currently available robotic systems are limited in their ability to automatically generate behaviours that align with their speech. In this project I developed an automatic system in which audio input from a user generates upper-body movements of the user on the humanoid robot Pepper. To the best of my knowledge this system brings two novelties: (i) it performs whole upper-body motion synthesis including head, hand and hip movements; and (ii) it is targeted to a humanoid robot. From an implementation perspective, the system was developed using only single-view RGB videos and it supports an offline as well as online (real-time) synthesis mode.

Using audio-visual recordings of upper-body movements of 19 speakers, I extracted audio and pose features. For pose feature extraction, I compared four 3D pose estimation methods that estimate 3D joint positions of the human skeleton from a single-view RGB video. The estimated 3D joint positions were used to calculate angles between upper-body joints and the obtained angle time-series were then smoothed and constrained to the robot's operating limits. To learn the mapping between audio features and upper-body pose, I trained the multilayer perceptron (MLP) and long short-term memory (LSTM) neural network models in a subject-independent (SI) and subject-dependent (SD) manner. Comparison of the four 3D pose estimation methods showed that the method *Lifting from the deep* is best suited for the development of upper-body pose regression models, as it correctly handles videos with missing body joints and its pose estimates result in the least jerky movements.

The developed system was evaluated quantitatively and also qualitatively using web-surveys when driven by natural as well as synthetic speech. In particular: I compared the SI with SD model variant and the MLP with LSTM model type; I investigated the relationships between quantitative and qualitative evaluation metrics; and I also examined how the speaker's personality traits affect the synthesised movements.

My results show that the SD model variants outperform the SI variants, suggesting that it is reasonable to develop subject-specific models for this task. Next, the MLP model is better suited for real-time motion synthesis than the LSTM, as it performs the online synthesis approximately 5-times faster. On natural speech, the movements generated by the LSTM model were assessed as significantly more appropriate for the given audio than those generated by the MLP model, which generalises the same findings of previous speech-to-head-motion-synthesis works to the whole upper-body motion synthesis. On synthetic speech, however, the survey respondents preferred the MLP model over the LSTM. Relating the quantitative and qualitative results, I conclude that the synthesised movements that are more similar to the ground truth movements are perceived as more appropriate for the audio. Lastly, I found two significant relationships between the speaker's personality traits and the motion synthesised for the speaker. For speakers with high conscientiousness trait the SD variant of the MLP model generates movements that better match the ground truth movements than for speakers with low conscientiousness trait. The movements synthesised by this model for speakers with high conscientiousness trait are further perceived as more appropriate for the input audio.

Total word count: 11,966

Contents

List of abbreviations

Abbreviation	Meaning	Page
BLSTM	bidirectional long short-term memory	8
CCA	canonical correlation analysis	25
CRF	conditional random field	7
CVAE	conditional variational autoencoder	8
DBN	dynamic Bayesian network	7
FFT	Fast Fourier transform	18
GRU	gated recurrent unit	9
HFCRBM	hierarchical factored conditional restricted Boltzmann machine	8
HMM	hidden Markov model	7
HSMM	hidden semi markov model	8
LCCA	local canonical correlation analysis	25
LFTD	3D pose estimation method *Lifting from the deep* [1]	31
LogFB	log filter bank (audio features)	17
LogFB-26	audio feature set with LogFB feature vectors of size 26	29
LogFB-52	audio feature set with LogFB-based feature vectors of size 52	29
LogFB-78	audio feature set with LogFB-based feature vectors of size 78	29
LSTM	long short-term memory (model)	21
LSTM-SD	LSTM model trained in subject-dependent manner	38
LSTM-SI	LSTM model trained in subject-independent manner	38
MFCC	Mel frequency cepstral coefficient (audio features)	17
MFCC-13	audio feature set with MFCC feature vectors of size 13	29
MLP	multilayer perceptron (model)	19
MLP-SD	MLP model trained in subject-dependent manner	38
MLP-SI	MLP model trained in subject-independent manner	38
MSE	mean squared error	24
OP+A	3D pose estimation method based on OpenPose [2] and domain-specific assumptions	31
OP+M	3D pose estimation method based on OpenPose [2] and 2D→3D matching approach [3]	30
ReLU	rectified linear unit (activation function)	21
RMSE	root mean squared error	24
SAVC	short audio-visual clip	48
SBS	side-by-side (video)	48
SD	subject-dependent (model variant)	38
SI	subject-independent (model variant)	38
TTS	text-to-speech (system)	49
VNect	3D pose estimation method [4]	32

List of symbols

Symbol	Units	Meaning	Page
f_f	Hz	prediction frame rate	23, 29, 36
N_{fe}		number of audio features (audio feature vector size)	19, 29
N_{fr}		total number of prediction frames/time-steps	16, 19, 23
N_τ		number of frames/time-steps per sequence/gesture	21, 25, 42
τ	s	gesture duration window	25
ν		joint angle normalisation function	24
σ_S		sigmoid activation function	21,
θ_i	rad	joint angle i	16
${}_{min}\theta_i$	rad	lower bound on joint angle i	16
${}_{max}\theta_i$	rad	upper bound on joint angle i	16
$\boldsymbol{\theta}$	rad	pose vector of 11 employed joint angles	16
$\boldsymbol{\Theta}$	rad	$N_{fr} \times 11$ matrix of joint angles	16
${}^t\boldsymbol{\Theta}$	rad	$N_{fr} \times 11$ matrix of ground truth joint angles	23
${}^p\boldsymbol{\Theta}$	rad	$N_{fr} \times 11$ matrix of predicted joint angles	23
$\boldsymbol{\Theta}^{01}$		$N_{fr} \times 11$ matrix of normalised joint angles (pose feature set)	24, 30
f_s	Hz	audio sampling frequency	17, 29
α		pre-emphasis filter coefficient	17, 29
w	s	audio frame size	17, 29
s	s	audio frame stride	17, 29
N_{FFT}		FFT size	18, 29
N_{filt}		number of extracted filter bank energies	18, 29
N_Δ		differential audio features extraction parameter	19, 29
${}_i\boldsymbol{\omega}$		audio feature vector of i features	29
$\boldsymbol{\Omega}$		$N_{fr} \times N_{fe}$ matrix of audio features	19
${}_i\boldsymbol{\Omega}$		$N_{fr} \times i$ matrix of audio features (audio feature set)	29
RMSE	rad	root mean squared error	24
L		loss optimised by neural network models	23
		loss evaluation metric	24
$\text{LCCA}_{\boldsymbol{\Theta}}$		local correlation between true and predicted movements	25
ΔLCCA		local audio-with-movements correlation absolute difference	26
J	$\text{rad}^2\text{s}^{-5}$	angular jerkiness	26, 34
$\Delta\tilde{J}$	$\text{rad}^2\text{s}^{-5}$	averaged angular jerkiness absolute difference	26

List of Figures

List of Tables

Chapter 1

Introduction

This chapter first describes the motivation and context for the project. Next, it provides overview of the undertaken work and research along with the obtained results. Lastly, the structure of the dissertation is laid out.

1.1 Motivation

Body language plays an important role in human communication. While speaking, people use facial expressions, head motion and hand gestures to convey the same meaning as speech and to complement and enrich the message [14]. Head movements contribute to speech comprehension [15], increase the level of perceived naturalness [16, 17], warmth and competence [18], and also convey the emotional state of the speaker [19, 20, 21]. Likewise, hand and arm movements are significant for distinguishing between affective states [22]. It has been also shown that 90% of human gesticulation occurs while speaking [14] and that the speech and gestures originate in the same internal process and share the same semantic meaning [23, 14]. These results thus motivated the investigation of the correlation and synchrony between these two modalities. Voigt et al. [24] showed that there is a statistically significant correlation between prosodic features extracted from audio and raw body movements. Several studies [25, 14, 26] further confirmed the synchrony between gesture strokes and stressed syllables and also between gesture phrases and intermediate intonation phrases. All these findings led to attempts at cross-modal prediction, learning the mapping from audio to movements. Such a *many-to-many mapping*[1] problem is challenging, as humans are very sensitive to natural human movements [27] and to the consistency between speech and associated gestures [28]. Despite this, several works successfully developed audio-driven motion synthesis systems, for example, for synthesis

[1]Multiple different audio signals can be associated with the same motion sequence and vice-versa, multiple different motion sequences can be associated with the same audio signal.

of head motion [29, 30, 11] or hand movements [31, 32]. However, there was no previous attempt at *whole upper-body* motion synthesis driven by audio.

Previous research targeted the synthesis at animated conversational agents and talking avatars that can be used in several fields including business enterprises, healthcare, education and entertainment. More specific examples are: virtual agents for online shopping, technical support and customer service; visual aids for hearing impaired individuals and speech therapy; virtual storytellers for children; or automation of the animation process in the development of computer games and animated films. However, to the best of my knowledge there was no previous work that synthesised body movements driven by audio on a *humanoid robot*. Such an automatic system would allow more expressive and human-like communication using only audio input in areas such as robotic telepresence where a robot provides physical embodiment at a remote place. This would have applications in various settings including business meetings [33], remote education [34] or elderly care [35] which are currently emerging application domains of social robotics. Also, this system could be employed in public robotic assistants and guides to automate their behaviours. Moreover, an extension of this system could employ a text-to-speech (TTS) subsystem so that the system input would be a written text that the TTS subsystem converts to artificial speech used to drive robot movements.

1.2 Context

This project is based on the audio-visual dataset collected by Bremner et al. in a study where upper-body gesturing of 20 subjects was captured while talking [5]. The subjects were asked to describe one of their hobbies and tell a dramatic story based on a given picture. In total, the dataset consists of 40 single-person videos (two for each subject) recorded using a single RGB camera. The average duration of recordings per subject is below 2 min. The dataset also contains the subjects' self-reported personality traits. Sample video frames from this dataset are shown in Figure 1.1.

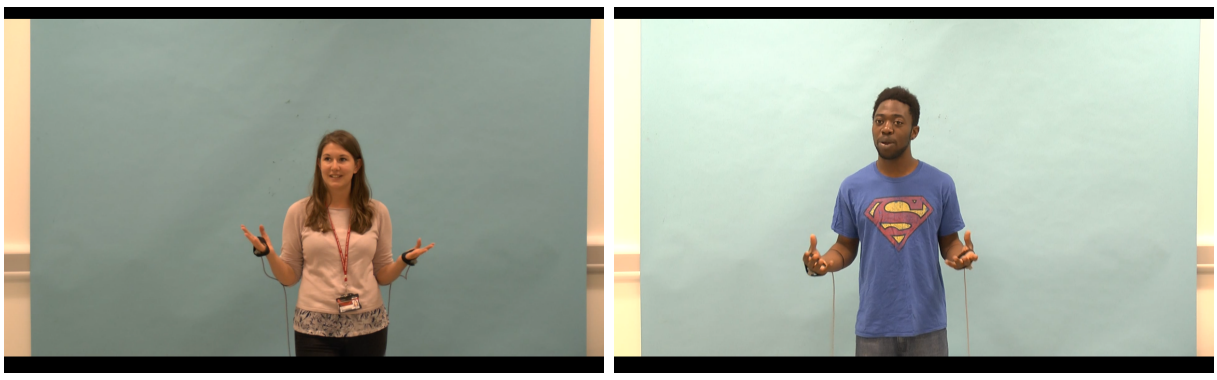


Figure 1.1: Sample video frames from the employed audio-visual dataset [5].

1.3 Project overview

In this project I developed an automatic audio-driven upper-body motion synthesis system targeted to the humanoid robot Pepper. Specifically, the system takes audio input from its user and uses the trained neural network to predict time-series of 11 angles between upper-body joints that are used to control the robot upper-body pose. The simplified operation of the system is illustrated in Figure 1.2.

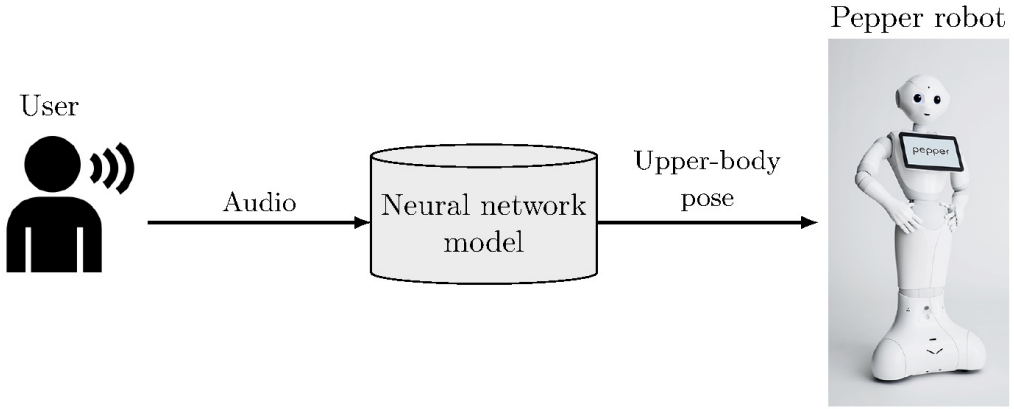


Figure 1.2: Simplified operation of the audio-driven upper-body motion synthesis system. Left and right pictures were taken from [6] and [7] respectively.

To the best of my knowledge, this system is first of its kind in two ways: (i) it performs whole upper-body motion synthesis including head, hand and hip movements, unlike the previous works that synthesised either head or hand movements; and (ii) it is targeted to a humanoid robot, unlike other existing works that focused on animated virtual characters or embodied agents. From an implementation perspective, the system was developed using only single-view RGB videos of upper-body movements of 19 speakers and it supports an offline as well as online (real-time) synthesis mode.

Using the audio-visual dataset of Bremner et al. [5], I first extracted audio and pose features. In particular, I extracted and compared four types (Section 4.2.2) of audio features. For pose feature extraction, I compared four 3D pose estimation methods (Section 4.2.3) that estimate 3D joint positions of the human skeleton from a single-view RGB video. The estimated 3D joint positions were used to calculate 11 angles between upper-body joints and the obtained angle time-series were then smoothed and constrained to the robot's operating limits. To learn the mapping between audio features and upper-body pose (in terms of the 11 angles), I trained the multilayer perceptron (MLP) and long short-term memory (LSTM) neural network models in subject-independent[2] (SI) and subject-dependent[3] (SD) manners. The MLP model had the advantage of being simpler but treated each training example independently, whereas LSTM, being more complex, could exploit the sequential structure in the data.

[2]General model trained and evaluated on data from multiple subjects.
[3]Specific model trained and evaluated on each subject separately.

My research during the system development brought the following findings.

- Quantitative and qualitative comparisons of the four 3D pose estimation methods showed that the method *Lifting from the deep* (LFTD) [1] is best suited for the development of upper-body pose regression models, because it correctly handles videos where only part of the human skeleton is present and it provides pose estimates that result in the least jerky movements.
- Comparison of the four audio feature sets did not provide convincing conclusions about which of these sets is best suited for the audio-to-upper-body-motion learning task.

The developed system was evaluated quantitatively (in terms of the loss, RMSE, ΔLCCA, $\text{LCCA}_{\boldsymbol{\Theta}}$, and $\Delta\tilde{J}$ metrics, Section 3.4) when driven by natural speech. The qualitative system evaluation involved two web-surveys[4] where the respondents watched audio-visual recordings of the Pepper robot generating upper-body movements of a speaker and assessed the appropriateness of the generated movements for the given audio. One survey was based on natural speech (20 responses) and the other on synthetic speech (43 responses). Overall, the system evaluation involved: comparison of the SI with SD model variant and the MLP with LSTM model type; investigation of the relationships between quantitative and qualitative evaluation metrics; and also an examination of how the speaker's personality traits affect the synthesised movements. My results and findings follow.

- Both quantitative and qualitative evaluations showed that the SD model variants perform better than SI for the MLP as well as LSTM model type, suggesting that it is best to develop subject-specific models for this task.
- Quantitative comparison of the MLP and LSTM model did not clearly show which of them generates better movements. However, the MLP model performs the online motion synthesis approximately 5-times faster than the LSTM model which makes it better suited for real-time motion synthesis.
- Qualitative comparison of the MLP and LSTM model on natural speech showed that the movements generated by the LSTM model are assessed as significantly more appropriate for the given audio than those generated by the MLP model and this was the case for both SI and SD model variants. This result generalises the findings of previous speech-to-head-motion works [30, 11] to the whole upper-body motion synthesis.
- Qualitative comparison of the MLP and LSTM model on synthetic speech resulted in exactly the opposite model preference than on the natural speech, namely, the

[4]The Ethics Committee approval for both surveys is provided in Appendix C.1.

movements generated by the MLP model were assessed as significantly more appropriate for the given audio than those generated by the LSTM model. This reflects the fact that the more machine-like movements generated by the MLP model better match the more machine-like synthetic speech.

- Relating the quantitative and qualitative results, it can be concluded that the synthesised movements that are closer to the ground truth movements (lower loss and RMSE measures) or that are more correlated with the ground truth movements (higher $\text{LCCA}_{\mathbf{\Theta}}$ measure) are perceived as more appropriate to the audio. Furthermore, the obtained results clearly indicate that the ΔLCCA metric (evaluating how well the local correlation between audio and predicted movements matches the local correlation between audio and ground truth movements) is not suitable for evaluation of the upper-body movements driven by audio, as it is in significant disagreement with other metrics.
- The investigation of the relationship between the speaker's personality traits and the motion synthesised for the speaker revealed the following two associations. For speakers with high conscientiousness trait the MLP-SD model generates movements that
  (i) are significantly more correlated with the ground truth movements, and
  (ii) are perceived as significantly more appropriate for the input audio
  than for speakers with low conscientiousness trait.

The project source code is available at
`https://github.com/jancio/Audio-driven-upper-body-motion-synthesis`.

1.4 Structure of the dissertation

This dissertation is organised as follows:

1. **Introduction**.
2. **Related work** details the previous studies in the field and summarises the differences and contributions of this project.
3. **Background** presents the target robotic platform and underlying theories employed in the development.
4. **Audio-driven upper-body motion synthesis system** provides the whole system overview, development methodology and implementation details.
5. **Evaluation** describes evaluation methods and discusses the obtained results.
6. **Conclusion** gives a summary of the completed work and results and also highlights future research directions.

Chapter 2

Related work

In this chapter I relate my work to previous studies and summarise its contributions.

2.1 Audio-driven motion synthesis

Previous works in the field of audio-driven motion synthesis can be compared considering the following six aspects.

1. **Target domain**
   All the below-reviewed studies targeted and applied their motion synthesis systems to animated talking avatars or embodied conversational agents. However, to the best of my knowledge, there was no previous work that synthesised audio-driven movements on a humanoid robot.

2. **Model type**
   Early studies [36, 37, 19, 38] usually relied on hard-coded rules deciding which motion pattern to synthesise. The major limitation of such rule-based systems is the repetitiveness of movements as there is only a fixed set of rules. Also, it is problematic to ensure synchronisation between verbal and non-verbal events using rules. Data-driven methods address these problems by capturing the variability of movements from the training data and by implicitly learning the synchronisation between audio and motion. Earlier works employed probabilistic models such as dynamic Bayesian networks (DBNs) [17], hidden Markov models (HMMs) [16, 39, 40, 41, 31], conditional random fields (CRFs) [32] or Gaussian mixture models [42]. Recently, the deep neural network models attained popularity also in this field. Ding et al. [43] showed that deep neural networks generate better head motion sequences than HMMs. In particular, they used a multilayer perceptron (MLP) model and in their later work [44] they further improved the system with unsupervised pre-

training using deep belief networks. The MLP models are however limited in modelling temporal data. Ding et al. [11] and Haag and Shimodaira [30] thus compared the MLP with bidirectional long short-term memory (BLSTM) model in the head motion synthesis task. Both works reported improvement of the BLSTM-based system over the MLP-based one. Other studies experimented with generative models, for instance, Chiu et al. [45] used hierarchical factored conditional restricted Boltzmann machines (HFCRBMs) to generate hand movements and Greenwood et al. [46] introduced a generative head motion model based on the conditional variational autoencoder (CVAE) that allows prediction of several motion trajectories for the same audio. One major challenge of the data-driven methods is the lack of meaning: even if the generated movements are well synchronised with the speech they may lack or even contradict the meaning of the communicated message. Several studies [29, 47] thus developed hybrid approaches, for example, Sadoughi et al. [29] constrained their DBN model on several discourse functions (affirmation, negation, question, and backchannel).

3. **Body parts**
   Most studies focused only on head movements [44, 29, 46, 30] and some further included facial expressions [48, 49, 50]. Other works [31, 32, 38, 45] synthesised only hand movements, for example, Bozkurt et al. [31] used hidden semi Markov models (HSMMs) relying on the hierarchical model of hand gestures [25]. Only Sadoughi et al. [47] attempted to synthesise both head and hand movements simultaneously. They focused on 3 prototypical hand gestures and 2 head gestures using two hybrid DBN models (one for head and one for hand gestures) which however required manual annotation of motion sequences. To the best of my knowledge, there was no previous work that performed whole upper-body motion synthesis, combining the synthesis of head, hand and hip movements in a fully automatic system.

4. **Source of 3D pose**
   The development of motion synthesis models requires estimation of the speaker's 3D pose. Majority of works relied on motion capture devices that directly provide 3D locations of body joints [29, 30, 11, 47, 31, 32, 38, 45] or alternatively, they used multiple cameras [46, 31]. Only the studies [44, 48, 49, 50] employed single-view RGB videos to estimate the *head pose*. However, to the extent of my knowledge, no previous work has used single-view RGB videos to estimate the *whole upper-body* pose in the context of audio-driven motion synthesis.

5. **Synthesis mode**
   Most studies [44, 29, 46, 30, 11, 49, 47, 31, 38, 45] developed offline systems that require the whole input audio upfront. In such cases there is no need for low-latency predictions and synthesis so that more complex models such as BLSTM

can be used. Online (or real-time) systems are much less researched. For instance, Pham et al. [48, 50] used LSTM and also the gated recurrent unit (GRU) model for real-time head motion synthesis and facial animation, while Levine et al. [32] generated hand movements from live speech.

6. **Evaluation measures**
   Due to the *many-to-many mapping*[1] nature of the problem, the evaluation of audio-driven motion synthesis systems is challenging and various methods have been proposed. Quantitative (objective) measures include: (root) mean squared error ((R)MSE) (between true and predicted trajectories), canonical correlation analysis (CCA) (between true and predicted trajectories, and between audio features and predicted trajectories), kinetic energy (KE) difference (between true and predicted trajectories) and jerkiness (J) of the predicted trajectories.[2] Equally important is the qualitative (subjective) evaluation, as humans are very sensitive to natural human movements [27] and to the consistency between speech and gestures [28]. Typically, the assessed qualities are appropriateness (A) of synthesised movements for the audio and naturalness (N) of the generated motion in general, both measured on a Likert scale. However, the comparison of numerical results between works in the field is problematic due to the lack of standardised multimodal corpora, as they often collected and used their own data.

Comparison

The most closely related studies are compared in Table 2.1, according to the above-described six aspects and also in terms of the average duration of recordings per speaker. Compared to almost all other works, the problem addressed in this work is more challenging due to the small amount of recordings per speaker.

[1] Multiple different audio signals can be associated with the same motion sequence and vice-versa, multiple different motion sequences can be associated with the same audio signal.

[2] For definitions of the employed quantitative measures see Section 3.4.

Table 2.1: Comparison of audio-driven motion synthesis studies in terms of the six aspects (Section 2.1) and average duration of recordings per speaker. *mocap* stands for motion capture device. For abbreviations of model types and evaluation measures see Section 2.1. * denotes that the durations of recordings were estimated based on the utterance duration of 12.5 seconds (typically 10-15 s).

	Aspect						
Study	Target domain	Model type	Body parts	3D pose source	Synthesis mode	Evaluation measures	Recordings per speaker (min)
[44]	talking avatar	MLP	head	single RGB camera	offline	MSE, CCA	100
[29]	talking avatar	hybrid DBN	head	mocap	offline	CCA, A, N	27
[46]	talking avatar	BLSTM, CVAE	head	3 RGB cameras	offline	MSE	180
[30]	talking avatar	BLSTM	head	mocap	offline	CCA, N	16
[11]	talking avatar	BLSTM	head	mocap	offline	RMSE, CCA, N	263*
[49]	talking avatar	BLSTM	head + face	single RGB camera	offline	RMSE, N	146*
[47, 51]	talking avatar	hybrid DBN	head + hands	mocap	offline	CCA, A, N	30, 66
[50, 48]	talking avatar	LSTM, GRU	head + face	single RGB camera	*online*	RMSE	6, 6
[31]	talking avatar	HSMM	hands	mocap, 4 RGB cameras	offline	CCA, KE, J, N	20
[32]	talking avatar	CRF	hands	mocap	*online*	N	12
[38]	talking avatar	rule-based	hands	mocap	offline	N	6
[45]	talking avatar	HFCRBM	hands	mocap	offline	A	1
This work	humanoid robot	MLP, LSTM	head + hands + hip	single RGB camera	*online*	CCA, RMSE, loss, J, A	2

2.2 Contributions

In the context of audio-driven motion synthesis my work has the following contributions:

- First of its kind system for whole upper-body motion synthesis, combining head, hand and hip movements.
- Also, first of its kind system targeted to a humanoid robot.
- Comparison of four 3D pose estimation methods based on a single-view RGB video (Section 4.2.3).
- Comparison of four types of audio features (Section 4.2.2).
- Quantitative and qualitative comparison of the MLP with LSTM model and also between their subject-independent and subject-dependent variants.
- Comparison of the qualitative preference for the MLP/LSTM model between natural speech and synthetic speech driven system.
- Examination of relationships between quantitative and qualitative evaluation metrics for audio-driven robot motion synthesis.
- Investigation of the influence of the speaker's personality traits on the synthesised robot movements.

Chapter 3

Background

This chapter first describes the Pepper robot and the associated implications for the project. Later, it provides the necessary underlying theories of audio features and neural networks. The chapter concludes with definitions of the employed quantitative evaluation metrics.

3.1 Robotic platform – Pepper

As a robotic platform for upper-body motion synthesis I used the humanoid robot Pepper (version 1.7 and body type V16) developed by SoftBank Robotics [7]. The Pepper robot is 1.20 m tall wheeled humanoid robot with 17 joints and three omni-directional wheels, as shown in Figure 3.1. This robot is commercially available and widely used in human-robot interaction studies [52, 53, 54].

Figure 3.1: Pepper robot [7].

3.1.1 Hardware

The Pepper robot can move its body by setting the joint angles, as shown by Figures 3.2–3.3.

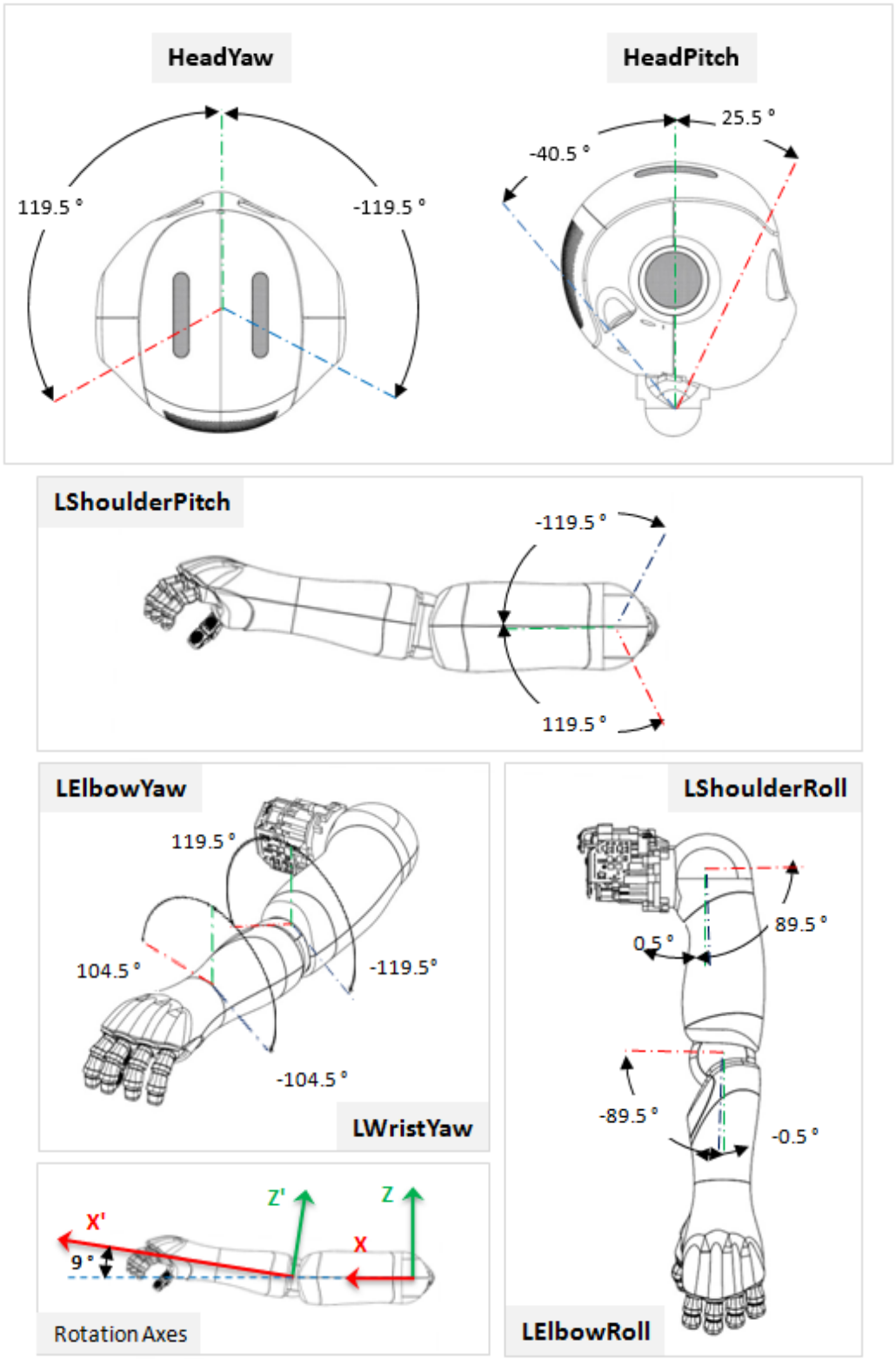


Figure 3.2: Possible movements of Pepper robot: head (top) and left arm (bottom) [8].

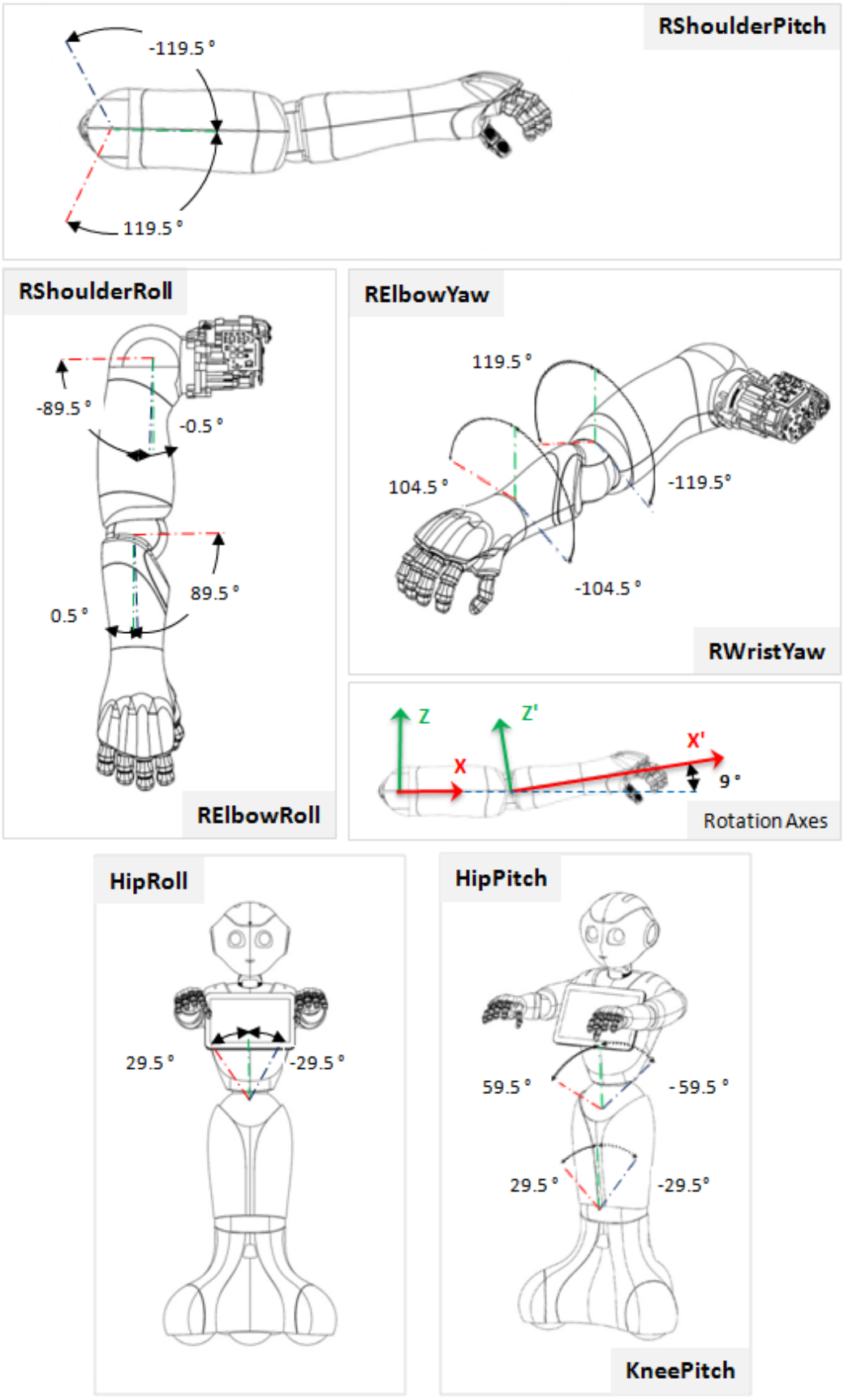


Figure 3.3: Possible movements of Pepper robot: right arm (top) and trunk (bottom) [8].

Out of all possible rotations demonstrated in Figures 3.2–3.3 I controlled the following 11 joint angles of the robot's upper body:

HeadPitch (θ_1), HeadYaw (θ_2),
LShoulderRoll (θ_3), LShoulderPitch (θ_4), LElbowRoll (θ_5), LElbowYaw (θ_6),
RShoulderRoll (θ_7), RShoulderPitch (θ_8), RElbowRoll (θ_9), RElbowYaw (θ_{10}),
HipRoll (θ_{11}).

The hip and knee pitch angles were always kept at their default values to ensure standing robot pose. Both wrist yaw angles were also set to their default values as these angles do not convey much gesturing and it would be extremely difficult to estimate these angles from videos.

The pose of the robot is then given by the pose vector $\boldsymbol{\theta} \in \mathbb{R}^{11}$. Each angle $\theta_i \in \boldsymbol{\theta}$ has its lower bound ${}_{min}\theta_i$ and upper bound ${}_{max}\theta_i$ such that ${}_{min}\theta_i \leq \theta_i \leq {}_{max}\theta_i$. These bounds are summarised in Table 3.1.

Table 3.1: Pepper robot's lower and upper bounds on 11 controlled joint angles [8].

i	Joint angle (θ_i)	Lower bound ${}_{min}\theta_i$ (rad)	Upper bound ${}_{max}\theta_i$ (rad)
1	HeadPitch (θ_1)	-0.7068	0.4451
2	HeadYaw (θ_2)	-2.0857	2.0857
3	LShoulderRoll (θ_3)	0.0087	1.5620
4	LShoulderPitch (θ_4)	-2.0857	2.0857
5	LElbowRoll (θ_5)	-1.5620	-0.0087
6	LElbowYaw (θ_6)	-2.0857	2.0857
7	RShoulderRoll (θ_7)	-1.5620	-0.0087
8	RShoulderPitch (θ_8)	-2.0857	2.0857
9	RElbowRoll (θ_9)	0.0087	1.5620
10	RElbowYaw (θ_{10})	-2.0857	2.0857
11	HipRoll (θ_{11})	-0.5149	0.5149

3.1.2 Software

For software development I used the NaoQi framework (version 2.5.5) and Python SDK provided by SoftBank Robotics [7]. To speedup the development and testing of the proposed system I simulated the virtual robot in the Choregraphe environment [9], as shown in Figure 3.4.

The robot was controlled by sending commands specifying the joint angles $\boldsymbol{\theta}$ at desired times. The upper-body movements over N_{fr} time-steps were then given by the matrix $\boldsymbol{\Theta} \in \mathbb{R}^{N_{fr} \times 11}$ of joint angles.

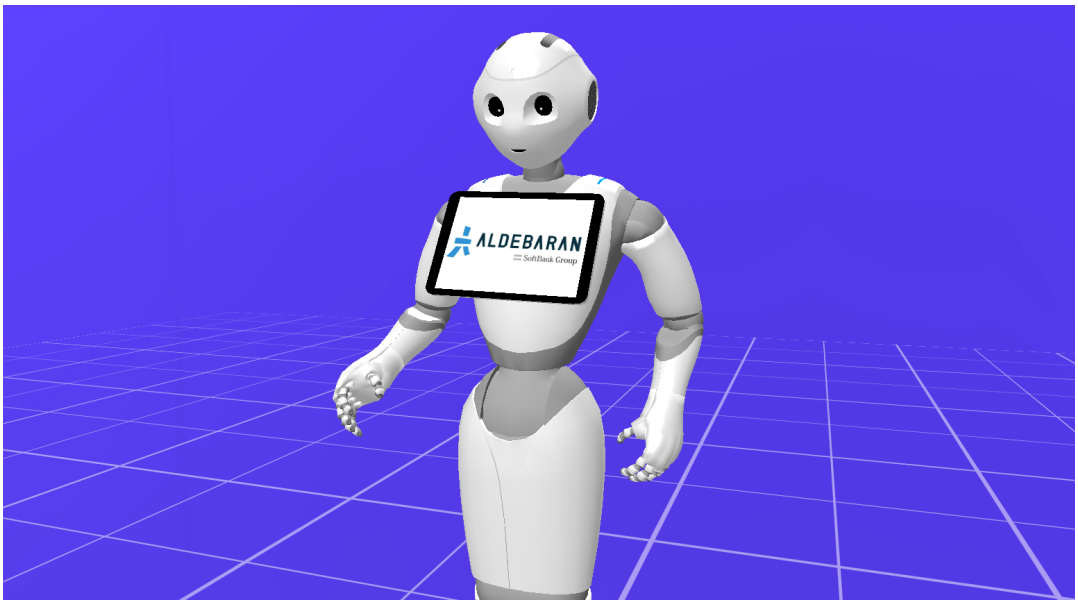


Figure 3.4: Pepper robot simulated in the Choregraphe environment [9].

3.2 Audio features

The state-of-the-art audio features used for audio-based learning are Mel frequency cepstral coefficients (MFCCs), introduced by Davis and Mermelstein [55]. More recently, log filter banks (LogFBs) are becoming popular [44, 46, 11]. The computation of MFCCs includes the calculation of LogFBs and involves the following steps [56, 57].

1. **Pre-emphasis filtering**
   Pre-emphasis filtering is used to balance the frequency spectrum since high frequencies usually have smaller magnitudes than lower frequencies. It further avoids numerical problems during the Fast Fourier transform (FFT) [58] in a later step and may also improve the signal-to-noise ratio. In particular, the emphasised signal $y(t)$ in time domain is calculated as

   $$y(t) = x(t) - \alpha x(t - f_s^{-1}) \tag{3.1}$$

   where $x(t)$ is the original signal, f_s is the sampling frequency, and α is the filter coefficient with typical values of 0.95 or 0.97 [56].

2. **Framing**
   Assuming that on short time scales the statistics of the audio signal does not vary a lot, the signal is split into short overlapping time frames by specifying the frame size w and frame stride s. Commonly used values are $w = 25$ ms and $s = 10$ ms [44, 30, 11, 49, 59].

3. **Windowing**
   To counteract the assumption made by FFT that the signal is infinite and to reduce spectral leakage, a window function is applied to each frame. For example, the frequently used Hamming window [60] is given by

$$h[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N_w - 1}\right) \tag{3.2}$$

   where $n \in \{0, 1, \ldots, N_w - 1\}$ and $N_w = f_s w$ is the frame size in samples.

4. **Fast Fourier transform and power spectrum**
   On each windowed frame y_i an N_{FFT}-point FFT [58] is performed and the corresponding power spectrum P_i is calculated as

$$P_i = \frac{|FFT(y_i)|^2}{N_{FFT}} \tag{3.3}$$

   Higher N_{FFT} values provide higher spectral resolution but take longer to compute. For audio feature extraction, N_{FFT} is usually set to 256 or 512 [56].

5. **Filter banks**
   To extract frequency bands and obtain filter banks, N_{filt} triangular filters ($N_{filt} = 26$ is standard) are applied on a Mel scale to the power spectrum of each frame. The Mel frequency scale mimics the non-linear human ear perception of sound, being more discriminative at lower frequencies and less discriminative at higher frequencies. The conversion between the frequency m in Mel and frequency f in Hertz is given by

$$m = 1125 \ln(1 + f/700) \qquad f = 700(\exp(m/1125) - 1) \tag{3.4}$$

   For mathematical details of the calculation of the N_{filt} filter bank energies see [57]. Since humans do not perceive loudness on a linear scale, logarithm of filter banks is computed, resulting in the LogFB feature vector $\boldsymbol{\omega}_i \in \mathbb{R}^{N_{filt}}$ for each audio frame i.

6. **MFCCs**
   Since the log filter bank coefficients may be correlated, the Discrete Cosine Transform [61] is applied to obtain a more compressed representation of the filter banks. Typically, only the lower 13 (out of N_{filt}) cepstral coefficients are retained, resulting in the MFCC feature vector $\boldsymbol{\omega}_i \in \mathbb{R}^{13}$ for each audio frame i.

7. **Differential features**
   To represent the dynamic nature of the audio signal, the first and second order time derivatives of the MFCC/LogFB coefficients are often [44, 30, 11] appended to the feature vectors. Specifically, the first-order differential coefficient of the feature j at

frame i is calculated as

$$\Delta\boldsymbol{\omega}_{i,j} = \frac{\sum_{n=1}^{N_\Delta} n(\boldsymbol{\omega}_{i+n,j} - \boldsymbol{\omega}_{i-n,j})}{2\sum_{n=1}^{N_\Delta} n^2} \tag{3.5}$$

where N_Δ is usually set to 2 [57]. The second-order differential coefficients $\Delta^2\boldsymbol{\omega}_{i,j}$ are then calculated from the first-order differential coefficients using the equation of the same form.

8. **Normalisation**
   The final step for both feature types (MFCC/LogFB) is the normalisation of each feature. Typically, the z-normalisation per speaker is performed [29], so that data of each speaker are standardised by removing the mean and scaling to unit variance along each feature dimension independently.

The whole audio feature extraction process results in the audio feature set $\boldsymbol{\Omega} \in \mathbb{R}^{N_{fr} \times N_{fe}}$ where N_{fr} is the number of audio feature frames and N_{fe} is the size of each feature vector.

An alternative and recently researched approach to audio feature extraction uses convolutional neural networks [50]. However, it does not consistently outperform the hand-crafted MFCC/LogFB features and achieves similar performance [59, 62, 63]. Therefore, this approach is not considered in this dissertation.

3.3 Neural network models

In this section I first describe two neural network models used for prediction of joint angles from audio features and then outline the training procedure. Since I aim to synthesise continuous movements, I specifically focus on the regression task where the outputs take continuous values as compared to the classification that assigns discrete class labels.

3.3.1 Multilayer perceptron (MLP) model

Given an input feature vector $\mathbf{x} \in \mathbb{R}^{N_{fe}}$ of N_{fe} features the multilayer perceptron (MLP) model is trained to predict the output vector $\mathbf{y} \in \mathbb{R}^{N_o}$ of N_o predictions [10]. The MLP model consists of several layers of units so that each unit is typically connected with all previous-layer units and applies a specific transformation to its inputs. In general, the MLP model is specified by the following attributes.

- Number of input-layer units: equivalent to the number N_{fe} of features.
- Number of hidden layers.

- Number of units per each hidden layer.
- Number of output-layer units: equivalent to the number N_o of predictions.
- Activation function for each hidden-layer and output-layer unit.

An example of the MLP network is shown in Figure 3.5.

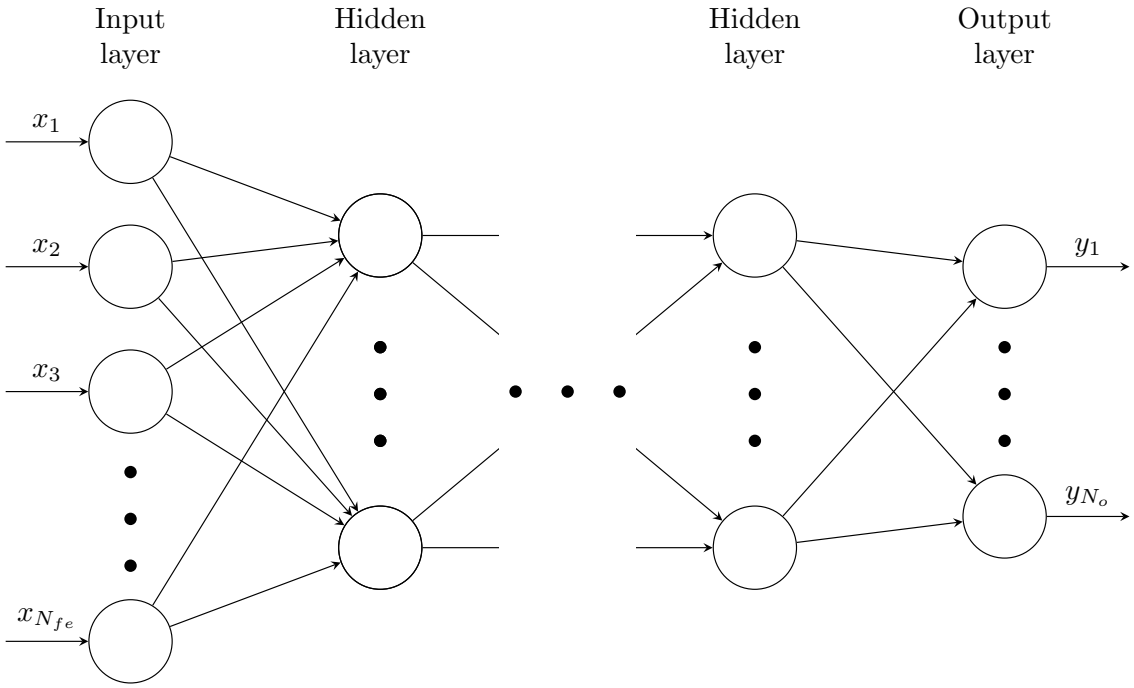


Figure 3.5: Multilayer perceptron (MLP) neural network with N_{fe} input features and N_o outputs. Diagram inspired by [10].

The input-layer units simply forward the input features to the next layer. Each hidden-layer and output-layer unit then inputs all n outputs h_i from the previous layer units and transforms them with a weighted linear summation $\sum_i w_i h_i + b$, applying the trained weights w_i and bias term b, and consequently with the activation function σ. These per-unit operations are illustrated in Figure 3.6.

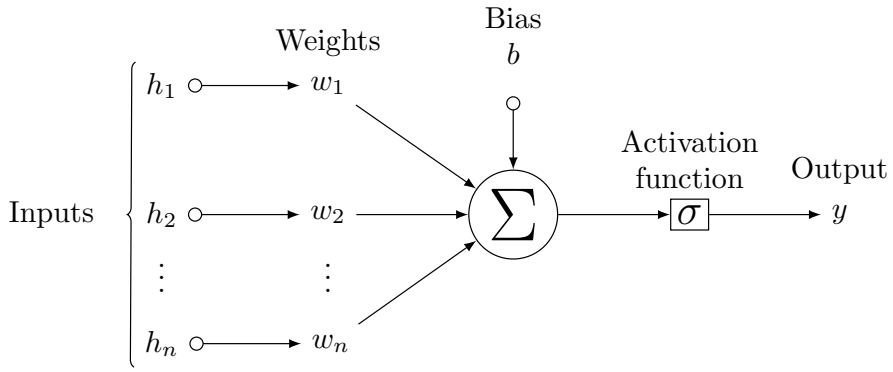


Figure 3.6: Operations within a single unit of the MLP network. The unit inputs n previous-layer outputs. Diagram inspired by [10].

For regression tasks the most commonly used activation functions are:

- **Rectified linear unit (ReLU)**: usually applied to hidden-layer units. It avoids vanishing gradient problem during training of deep (multiple hidden layers) networks [64, 65].

$$\sigma_R(x) = \max(0, x) \tag{3.6}$$

- **Logistic (sigmoid)**: scales the output to the $[0, 1]$ range.

$$\sigma_S(x) = (1 + e^{-x})^{-1} \tag{3.7}$$

- **Hyperbolic tangent**: scales the output to the $[-1, 1]$ range.

$$\sigma_T(x) = (e^x - e^{-x})(e^x + e^{-x})^{-1} \tag{3.8}$$

The MLP treats each training example $(\mathbf{x}, \mathbf{y})$ independently, and so if it is applied to sequential data it does not exploit relations between adjacent items within a sequence.

3.3.2 Long short-term memory (LSTM) model

In contrast to MLP, the long short-term memory (LSTM) [66, 67] model can directly model sequential data, incorporating past contexts using internal memory. I will particularly focus on the sequence labeling problem [68] where given an input sequence $\mathbf{X} \in \mathbb{R}^{N_\tau \times N_{fe}}$ of N_τ time-steps with N_{fe} features per time-step, the task is to output the sequence $\mathbf{Y} \in \mathbb{R}^{N_\tau \times N_o}$ of the same length with N_o predictions per time-step.

The main building block of the LSTM network is the LSTM unit shown in Figure 3.7. For each time-step $t = 1..N_\tau$, it contains the memory cell c_t that stores the current context and input i_t, output o_t and forget f_t gates representing write, read, and reset actions on the memory cell respectively. Given an input feature vector x_t the LSTM unit computes the output vector h_t as follows:

$$\begin{aligned}
i_t &= \sigma_S(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma_S(W_{xf}x_t + W_{hf}h_{t-1} + W_{ci}c_{t-1} + b_f) \\
a_t &= \sigma_T(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
c_t &= f_t c_{t-1} + i_t a_t \\
o_t &= \sigma_S(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \sigma_T(c_t)
\end{aligned} \tag{3.9}$$

where σ_S, σ_T are activation functions defined in Section 3.3.1 and a_t is the cell input activation function. All weight matrices W and bias vectors b are estimated during training.

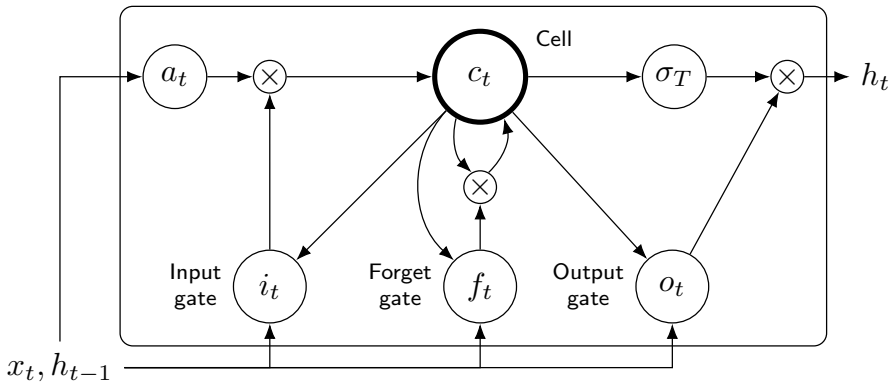


Figure 3.7: Long short-term memory (LSTM) unit. Diagram inspired by [11].

Several LSTM units then form an LSTM layer and the whole LSTM network is generally composed of any combination of fully-connected[1] and LSTM layers. An example of the LSTM network with one LSTM hidden layer and fully-connected output layer is shown in Figure 3.8.

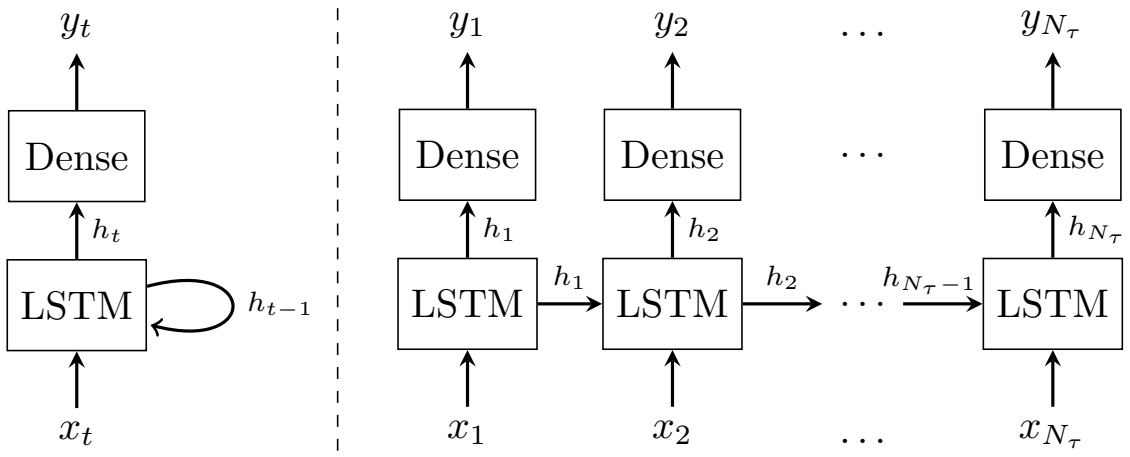


Figure 3.8: Long short-term memory (LSTM) network with one LSTM hidden layer and fully-connected (*Dense*) output layer: non-unrolled (left) and unrolled along time dimension (right). x_t is the input feature vector, y_t is the output vector of predictions, and h_t denotes all outputs from the LSTM layer, at time-step t. Diagram inspired by [12].

In case of variable-length sequences, the input sequence is first padded (usually with zero feature vectors) to the maximum sequence length N_τ. Then, a masking technique is applied so that if all feature values at a time-step are equal to the mask value (in this case zero) then the time-step is masked meaning that all computations for this time-step are skipped in all network layers.

[1]Layer with MLP units each connected to all previous-layer units.

3.3.3 Training

The neural network models are trained by minimising a loss function. For regression problems the typical choice is the squared error loss function:

$$L = \frac{1}{N_o} \sum_{i=1}^{N_o} (\hat{y}_i - y_i)^2 \tag{3.10}$$

where $\hat{\mathbf{y}}, \mathbf{y} \in \mathbb{R}^{N_o}$ are the true and predicted output vectors respectively. More specifically, the backpropagation algorithm [69, 70] (and backpropagation through time [67] for LSTM) first computes gradients that are then used to perform updates of weights and bias terms. There are several update techniques such as stochastic gradient descent [71] or recently popular Adam [72]. The updates are made each time a batch of training examples was shown to the network. Larger batch size means more accurate updates, but this is often constrained by the available memory resources. The whole training phase consists of several epochs so that at each epoch all training examples are revisited. The training is stopped after a specific number of epochs or when the loss computed on validation dataset stops to improve.

To prevent models from overfitting the training data the dropout regularisation [73] is commonly used. During training it randomly disables each unit with probability p_d to prevent the unit from over-specialising.

3.4 Quantitative evaluation measures

In this section I provide definitions of the employed quantitative evaluation measures. As indicated in Section 2.1, it is challenging to evaluate the audio-to-motion predictions due to the *many-to-many mapping*[2] nature of the problem and so there is no single quantitative metric that could be relied on [46]. Therefore, inspired by previous works (Section 2.1) I used a combination of several measures including the root mean squared error (RMSE), canonical correlation analysis (CCA) and jerkiness (J) and I further defined the loss measure (L).

I will denote the input matrix of audio features by $\mathbf{\Omega} \in \mathbb{R}^{N_{fr} \times N_{fe}}$, the matrix of ground truth joint angles by ${}^{t}\mathbf{\Theta} \in \mathbb{R}^{N_{fr} \times 11}$, and the output matrix of predicted joint angles by ${}^{p}\mathbf{\Theta} \in \mathbb{R}^{N_{fr} \times 11}$, where N_{fr} is the number of frames the predictions were made on and let the prediction frame rate be f_f.

[2] Multiple different audio signals can be associated with the same motion sequence and vice-versa, multiple different motion sequences can be associated with the same audio signal.

3.4.1 Root mean squared error (RMSE)

The root mean squared error of joint angle θ_j is given by

$$\text{RMSE}(\theta_j) = \sqrt{\frac{1}{N_{fr}} \sum_{i=1}^{N_{fr}} \left({}^{t}\boldsymbol{\Theta}_{i,j} - {}^{p}\boldsymbol{\Theta}_{i,j}\right)^2} \tag{3.11}$$

and the overall RMSE is calculated as

$$\text{RMSE} = \sqrt{\frac{1}{11N_{fr}} \sum_{j=1}^{11} \sum_{i=1}^{N_{fr}} \left({}^{t}\boldsymbol{\Theta}_{i,j} - {}^{p}\boldsymbol{\Theta}_{i,j}\right)^2} \tag{3.12}$$

However, the RMSE measure does not take into account the angle ranges (Table 3.1) and so the error in angle with small range has equal weight as the same error in angle with large range.

3.4.2 Loss (L)

To address the above-described limitation of the RMSE metric I defined the loss measure L as the MSE normalised by angles' ranges. Firstly, let $\boldsymbol{\Theta}^{01}$ be the matrix of normalised joint angles (also referred to as pose feature set) such that

$$\forall i \in \{1, 2, \ldots, N_{fr}\}.\forall j \in \{1, 2, \ldots, 11\}.\boldsymbol{\Theta}^{01}_{i,j} = \nu(\boldsymbol{\Theta}_{i,j}) \tag{3.13}$$

where the function ν normalises the i^{th} prediction of the j^{th} joint angle $\boldsymbol{\Theta}_{i,j}$ to the range $[0, 1]$ as

$$\nu(\boldsymbol{\Theta}_{i,j}) = \frac{\boldsymbol{\Theta}_{i,j} - {}_{min}\theta_j}{{}_{max}\theta_j - {}_{min}\theta_j} \tag{3.14}$$

The loss L is then defined as

$$L = \frac{1}{11N_{fr}} \sum_{j=1}^{11} \sum_{i=1}^{N_{fr}} \left({}^{t}\boldsymbol{\Theta}^{01}_{i,j} - {}^{p}\boldsymbol{\Theta}^{01}_{i,j}\right)^2 \tag{3.15}$$

Training the neural network models to predict the *normalised* joint angles (given by $\boldsymbol{\Theta}^{01}$), then ensures that the loss measure L directly corresponds to the loss optimised by the neural network (Section 3.3.3). Furthermore, if the sigmoid activation functions σ_S are used at the output layer (Section 3.3.1), the predictions are implicitly constrained to the robot's operating limits. The predicted joint angles ${}^{p}\boldsymbol{\Theta}$ in original ranges are then recovered using ν^{-1}.

3.4.3 Canonical correlation analysis (CCA)

The movements much different from the ground truths can still be relevant for the given audio, even though they have large RMSE and loss values. This limitation of the RMSE and loss metrics can be addressed using the canonical correlation analysis (CCA) [74] that, contrary to standard correlation, can operate on multi-dimensional data rather than on single column vectors. Given two zero-mean datasets $X \in \mathbb{R}^{d \times n}$ and $Y \in \mathbb{R}^{d \times m}$, CCA finds a pair of basis vectors $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$ that maximise the correlation ρ between the projections $U = X\mathbf{a}$ and $V = Y\mathbf{b}$ onto a canonical coordinate space.

$$\rho = \max_{\mathbf{a},\mathbf{b}} \operatorname{corr}(X\mathbf{a}, Y\mathbf{b}) \tag{3.16}$$

The projections U and V form the first pair of canonical components and the subsequent canonical components can be iteratively computed constraining the next pair of canonical components to be orthogonal to the preceding one. As the final CCA value between the datasets X and Y the correlation of the first canonical component is used.

$$\mathrm{CCA}(X, Y) = \rho \tag{3.17}$$

Since the evaluated motion sequences consist of several shorter gestures, the linear correlations evaluated by CCA are unlikely to hold at longer time-scales (e.g. the whole video). In such cases the CCA is calculated using sliding-window approach with the window size of N_τ datapoints and 1 datapoint stride. This is also known as the *local CCA* (LCCA) [30, 75].

$$\mathrm{LCCA}(X, Y) = \frac{1}{d - N_\tau + 1} \sum_{i=1}^{d-N_\tau+1} \mathrm{CCA}(X_{i:i+N_\tau-1}, Y_{i:i+N_\tau-1}) \tag{3.18}$$

Furthermore, LCCA allows comparisons of movements with different lengths (i.e. long vs short videos). Considering the hand gesture durations reported in previous works (0.3-5 s [76] and 2.49 s [77]) and also the choice of 3 s windows to capture distinct head movements in the related speech-driven head motion synthesis study [30], I set the time window to $\tau = 3$ s which corresponds to $N_\tau = \tau f_f$ frames.

Specifically, I used LCCA to evaluate two kinds of correlations.

- **Correlation between true and predicted movements**

$$\mathrm{LCCA}_{\mathbf{\Theta}} = \mathrm{LCCA}({}^{t}\mathbf{\Theta}, {}^{p}\mathbf{\Theta}) \tag{3.19}$$

- **Correlation difference between audio and true/predicted movements**
  Firstly, the LCCAs between audio features and true angles, and between audio

features and predicted angles are calculated. Their absolute difference is then used as the evaluation metric

$$\Delta\text{LCCA} = \left|\text{LCCA}(\mathbf{\Omega}, {}^{t}\mathbf{\Theta}) - \text{LCCA}(\mathbf{\Omega}, {}^{p}\mathbf{\Theta})\right| \tag{3.20}$$

This metric describes how well the correlation between audio and predicted movements matches the correlation between audio and ground truth movements.

Even the CCA-based metrics are limited for the audio-to-motion prediction task. For example, head shaking and head nodding at the same phase and frequency would show strong correlations even though their meanings are opposite.

3.4.4 Jerkiness (J)

Another quality of the synthesised motion is its smoothness which is usually[3] evaluated by its angular jerkiness [78]. When dealing with discrete time steps the overall angular jerkiness J of the motion given by $\mathbf{\Theta}$ is defined as

$$J(\mathbf{\Theta}) = \frac{1}{2f_f(N_{fr}-3)} \sum_{i=1}^{N_{fr}-3} \sum_{j=1}^{11} \left(\Delta^3 \mathbf{\Theta}_{i,j}\right)^2 \tag{3.21}$$

where $\Delta^3\mathbf{\Theta}_{i,j} = (\mathbf{\Theta}_{i+3,j} - 3\mathbf{\Theta}_{i+2,j} + 3\mathbf{\Theta}_{i+1,j} - \mathbf{\Theta}_{i,j})f_f^3$ is the 3$^{\text{rd}}$-order forward difference of angle θ_j.

It has been shown that the voluntary human movements obey minimum jerk trajectories, which are also the smoothest trajectories [79, 80]. However, if the jerkiness of the ground truth motion is not known the direct minimisation of jerkiness may lead to overly smooth motion or even no motion at all, and so the comparisons based on jerkiness have to be done with care. If the jerkiness $J({}^{t}\mathbf{\Theta})$ of the ground truth motion is available, it is thus better to minimise the absolute difference $\Delta J = |J({}^{t}\mathbf{\Theta}) - J({}^{p}\mathbf{\Theta})|$, in order to avoid overly smooth movements.

Since the jerkiness defined by Equation (3.21) is normalised by trajectory length, the jerkiness of n motion sequences can be assessed by calculating the means

$$\tilde{J} = \frac{1}{n}\sum_{i=1}^{n} J_i \qquad\qquad \Delta\tilde{J} = \frac{1}{n}\sum_{i=1}^{n} \Delta J_i \tag{3.22}$$

The jerkiness metric does not however evaluate the generated motion with respect to the audio.

[3] For example, also used by Bozkurt et al. [31] for evaluation of their speech-driven motion synthesis system.

Chapter 4

Audio-driven upper-body motion synthesis system

This chapter first provides an overview of the developed audio-driven upper-body motion synthesis system and consequently, it describes the individual system stages and the associated methodologies.

4.1 System overview

The architecture of the audio-driven upper-body motion synthesis system is shown in Figure 4.1. There are two main processing phases:

1. **Analysis phase:** the dataset of audio-visual recordings is used to train the pose regression model, learning the mapping between the extracted audio feature set $\mathbf{\Omega}$ and pose feature set $\mathbf{\Theta}^{01}$.

2. **Synthesis phase:** the trained pose regression model is applied to the audio features $\mathbf{\Omega}$ extracted from the audio input to predict the pose features $\mathbf{\Theta}^{01}$. The post-processing stage then
   - inverts the predicted pose features to joint angles $\mathbf{\Theta}$ using ν^{-1} (Section 3.4.2),
   - smooths the obtained angle trajectories (Section 4.2.3), and
   - sends the commands specifying the pose vectors $\boldsymbol{\theta}$ to the robot using adaptive sleep-times to ensure synchrosnisation with the audio playback (Section 4.4).

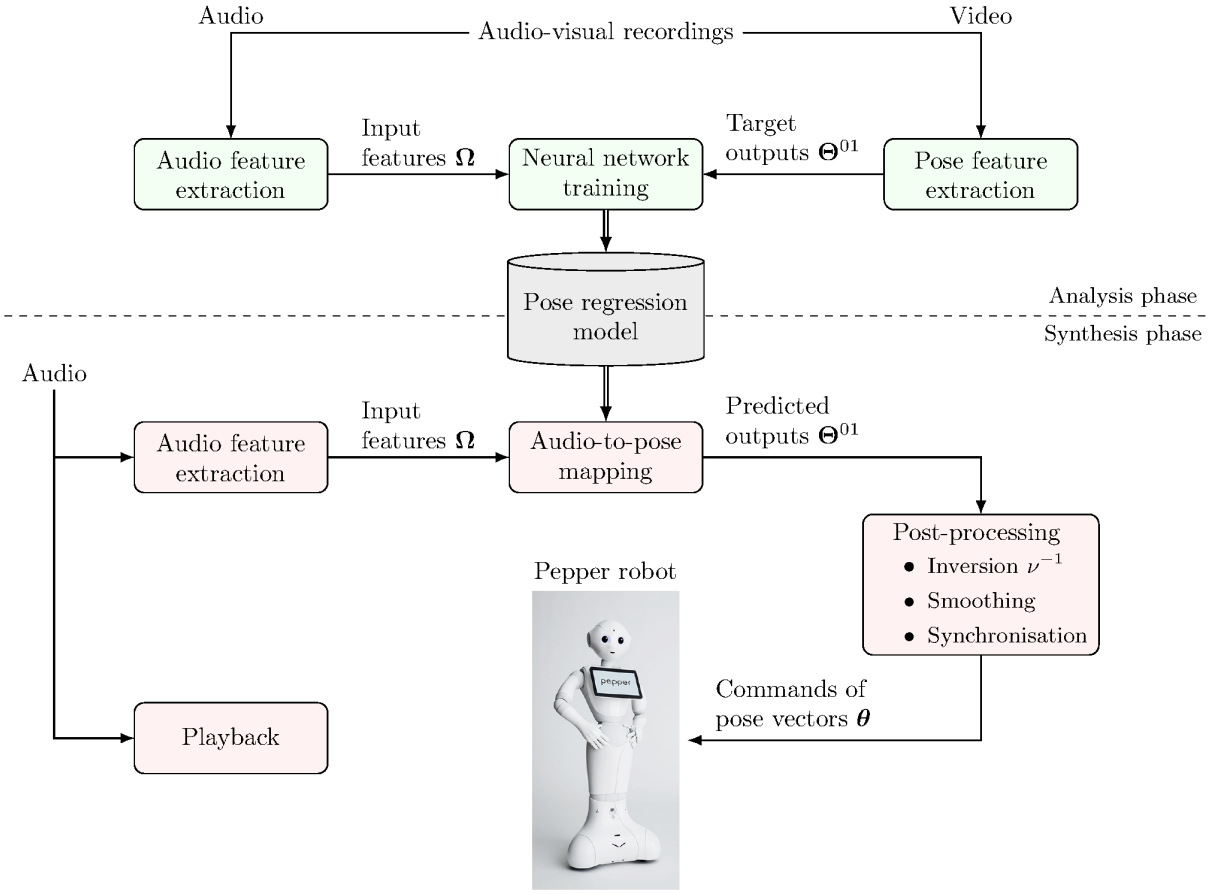


Figure 4.1: Diagram of the audio-driven upper-body motion synthesis system.

4.2 Feature extraction

This section describes the employed dataset as well as the audio and pose feature extraction stages.

4.2.1 Dataset

The audio-visual dataset used in this project was collected in the study [5] where upper-body gesturing of 20 subjects was captured while talking (Figure 1.1). Each subject was asked to perform two tasks: (i) describe one of their hobbies and (ii) tell a dramatic story based on a given picture. In total, 40 single-person videos (one for each task) were recorded using an RGB camera with the video frame rate $f_v = 50$ Hz and stereo audio sampled at frequency $f_a = 44.1$ kHz. Durations of all videos are shown in Figure 4.2.

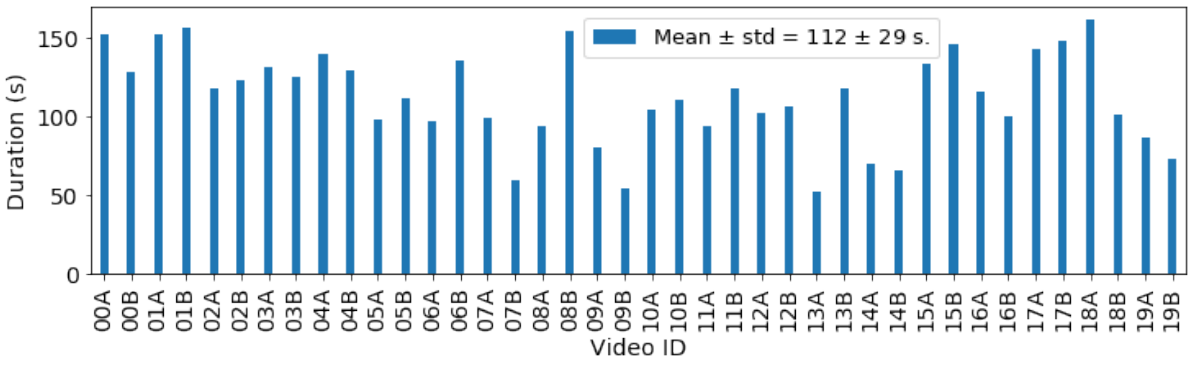


Figure 4.2: Durations of all videos from the employed dataset [5]. Suffixes A and B of video IDs denote task (i) and task (ii) videos of the same subject respectively. Durations range 52-162 s, corresponding to 2603-8081 frames at frame rate $f_v = 50$ Hz.

For each subject the dataset further contains their self-reported personality along the Big Five personality traits [81]: *Extroversion*, *Agreeableness*, *Conscientiousness*, *Neuroticism* and *Openness*. Each personality trait was measured on 1–10 Likert scale using the BFI-10 test [82].

4.2.2 Audio feature extraction

The stereo audio signal was first averaged to a single channel and downsampled to $f_s = 16$ kHz, similarly to [44, 46, 30, 11]. For each video I then extracted MFCC, LogFB and differential audio features as described in Section 3.2 using the Python toolbox *python-speech-features* [83]. In particular, I created the following four audio feature sets:

- **MFCC-13** $({}_{13}\boldsymbol{\Omega})$ of feature vectors ${}_{13}\boldsymbol{\omega} \in \mathbb{R}^{13}$ of 13 MFCC features,
- **LogFB-26** $({}_{26}\boldsymbol{\Omega})$ of feature vectors ${}_{26}\boldsymbol{\omega} \in \mathbb{R}^{26}$ of 26 LogFB features,
- **LogFB-52** $({}_{52}\boldsymbol{\Omega})$ of feature vectors ${}_{52}\boldsymbol{\omega} \in \mathbb{R}^{52}$ of 26 LogFB features with 26 first-order differential features appended,
- **LogFB-78** $({}_{78}\boldsymbol{\Omega})$ of feature vectors ${}_{78}\boldsymbol{\omega} \in \mathbb{R}^{78}$ of 26 LogFB features with 26 first-order and 26 second-order differential features appended.

In all four cases I used the standard settings of $\alpha = 0.97$, $w = 25$ ms, $s = 10$ ms, $N_{FFT} = 512$, $N_{filt} = 26$, $N_{\Delta} = 2$ and per-subject z-normalisation, as previously noted in Section 3.2. The resulting rate of audio features was $f_f = 100$ Hz, dictated by $s = 10$ ms.

4.2.3 Pose feature extraction

The extraction of pose features $\mathbf{\Theta}^{01}$ involved the steps illustrated in Figure 4.3 and detailed in the following subsections.

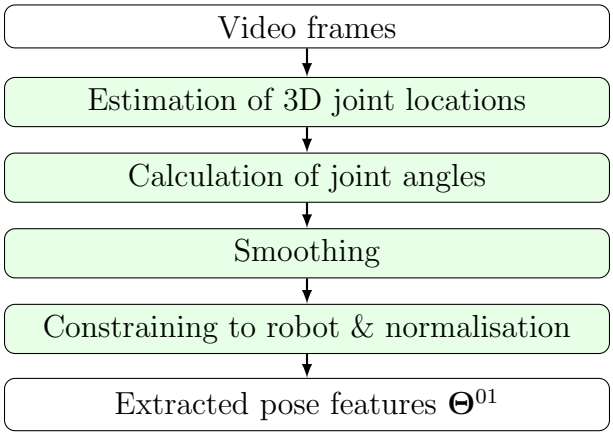


Figure 4.3: Pose feature extraction pipeline.

3D pose estimation from single-view RGB videos

Given the single-view RGB videos the first step was to locate the upper-body joints in 3D. This is an ill-posed problem in the Hadamard sense [84] as the solution may not be unique and is thus a very active research area with various methods being proposed. One approach is to first estimate 2D joint locations and consequently, lift them to 3D. For example, a very popular 2D pose estimation framework OpenPose [2] can be combined with the 2D→3D matching approach of Chen and Ramanan [3]. An alternative way is to develop models that directly predict 3D joint positions. For instance, the very recent real-time VNect [4] system or the *Lifting from the deep* (LFTD) method of Tome et al. [1].

To choose the most appropriate method for the problem addressed in this project, I compared the following four approaches:

1. **OP+M**: OpenPose [2] for 2D pose estimation combined with the 2D→3D matching approach of Chen and Ramanan [3].[1] The upper-body joints extracted by this method are shown in Figure 4.4.

[1] OpenPose model is available at: `https://github.com/CMU-Perceptual-Computing-Lab/openpose`. 2D→3D matching model is available at: `https://github.com/flyawaychase/3DHumanPose`.

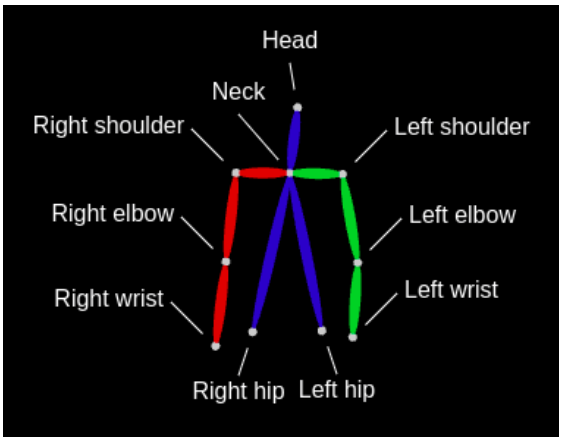


Figure 4.4: Upper-body joints extracted by OP+M or OP+A 3D pose estimation method.

2. **OP+A**: OpenPose [2] for 2D pose estimation combined with my own 2D→3D lifting method based on assumptions specific to the problem domain. In particular, I made the following assumptions to resolve ambiguities that the 2D→3D lifting involves:

   - The lengths of bones between joints are constant.
   - The first frame of the video contains neutral reference pose perpendicular to the viewpoint and is used to calculate bone lengths. Adaptive correction of bone lengths: if at any later frame a longer bone (calculated from 2D distances) is found, the original length is updated.
   - Neck, two shoulder and two hip joints always lie in the same plane P_B perpendicular to the viewpoint.
   - Head, two elbow and two wrist joints are always on the same side (closer to the camera) of the plane P_B.
   - Each wrist joint is always further from the plane P_B than the elbow joint of the same arm.

   The set of extracted joints was the same as in the case of OP+M (Figure 4.4).

3. **LFTD**: direct prediction of 3D joint positions by Tome et al. [1].[2] The upper-body joints extracted by this method are shown in Figure 4.5. Comparing to the OP+M or OP+A method, LFTD provides estimates of three more joints, namely, Pelvis, Spine and Nose.

[2]LFTD model is available at: `https://github.com/DenisTome/Lifting-from-the-Deep-release`.

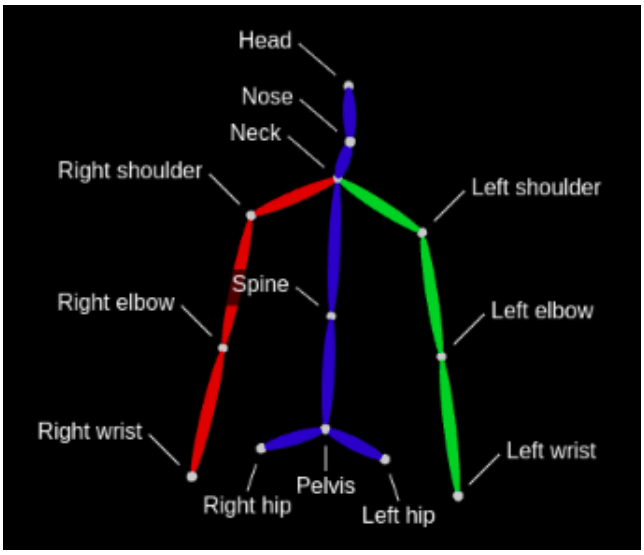


Figure 4.5: Upper-body joints extracted by LFTD or VNect 3D pose estimation method.

4. **VNect**: direct prediction of 3D joint positions by Mehta et al. [4].[3] The set of upper-body joints extracted by this method was the same as in the case of LFTD (Figure 4.5).

Table. 4.1 compares these four methods in terms of two qualities relevant for this project:

- *Temporal consistency*: whether the reconstructed 3D skeleton is temporally stable, i.e. pose reconstruction is *not* made independently for each video frame.
- *Robustness to missing joints*: whether the method can reconstruct an appropriate 3D skeleton when the video frames do not contain the whole human skeleton.

Table 4.1: Qualitative comparison of 3D pose estimation methods.

	Method			
Quality	OP+M	OP+A	LFTD	VNect
Temporal consistency	✘	✘	✘	✔
Robustness to missing joints	✘	✔	✔	✘

If the quality was not clear from the associated research papers, I explicitly contacted the authors. For their replies see Appendix B. The OpenPose framework itself is robust to missing joints, as also demonstrated by Figure 4.6 where the missing knee and toe joints are assigned zero reliability scores. However, the subsequent 2D→3D matching approach [3] was not designed to support such a feature and may result in deformed skeleton reconstructions, which was my main motivation to develop the OP+A method.

[3] VNect model is available at: `http://gvv.mpi-inf.mpg.de/projects/VNect/`.

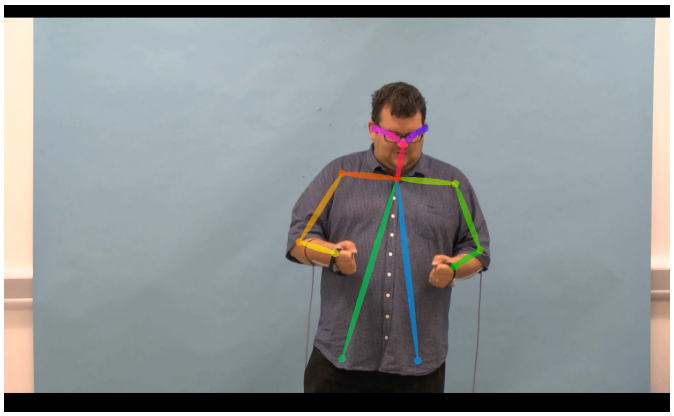


Figure 4.6: 2D pose estimation by OpenPose [2] on a sample video from dataset showing robustness to missing joints: knee and toe joints are assigned zero reliability scores.

Since none of the four methods satisfied both qualities, I further investigated which quality is more important in the context of upper-body recordings. For this I simulated the movements of 3D skeletons reconstructed by all four methods using the *VPython* library [13]. Table 4.2 shows a particular case where all four methods perform well. However, the simulation experiments with multiple videos showed that the skeletons reconstructed by VNect were significantly deformed for several videos (Table 4.3), implying that in the context of upper-body recordings the *robustness to missing joints* is more important than the *temporal consistency*. For this reason I decided to further compare only OP+A and LFTD methods.

Table 4.2: Sample skeleton reconstructions of the four 3D pose estimation methods. Simulated using *VPython* library [13]. Pictures taken at 0.3 s (15 frame) intervals.

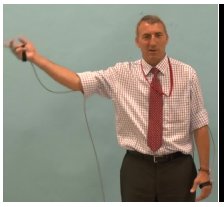
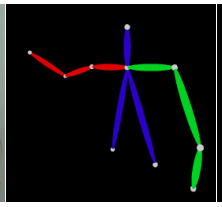
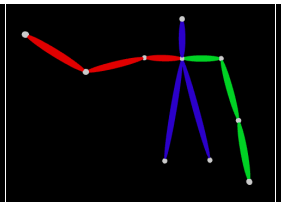
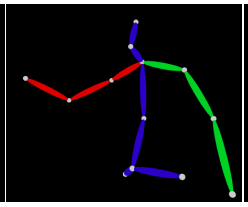
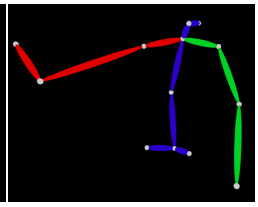
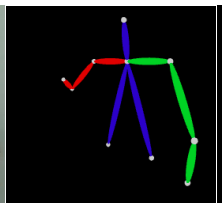
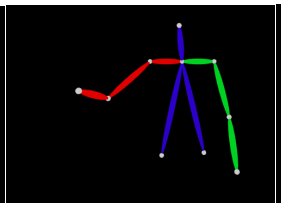
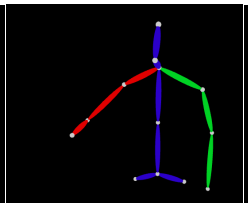
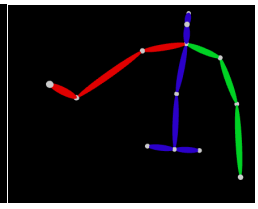
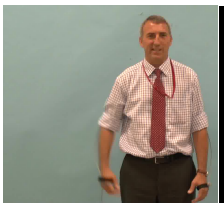
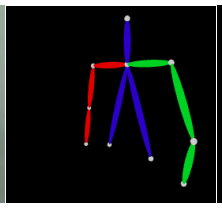
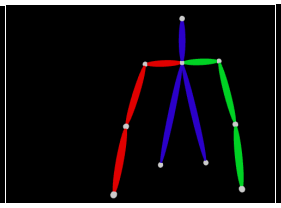
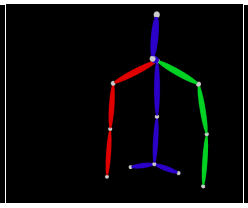
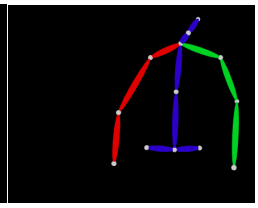
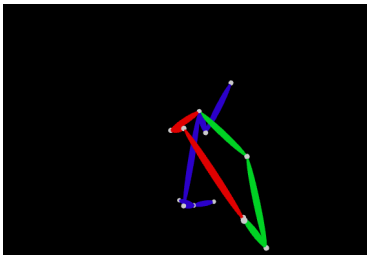
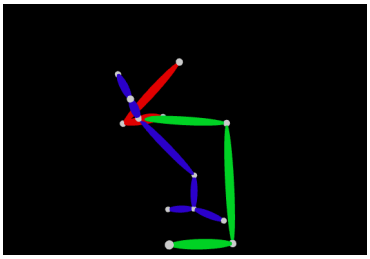
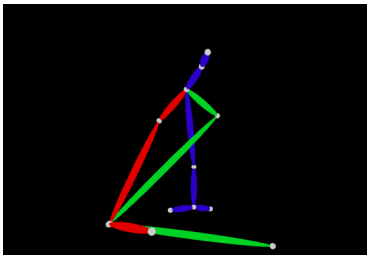
	Method			
Original video	OP+M	OP+A	LFTD	VNect
				
				
				

Table 4.3: Deformed skeleton reconstructions based on VNect 3D pose estimation method, for sample videos.

Original video	Skeleton reconstruction
	
	
	

I compared the OP+A with LFTD method in terms of the smoothness of the reconstructed upper-body motion, because these methods do not ensure *temporal consistency* while it is an important quality of natural-looking movements. Specifically, I used the angular jerkiness measure (Section 3.4.4). Firstly, for each method and for each video frame in every video, the 11 upper-body joint angles were calculated from the extracted 3D joint positions, using the package *geometry-simple* [85] for geometric computations. For each video $i \in \{1, 2, \ldots, 40\}$, this resulted in matrices ${}_{OP+A}\boldsymbol{\Theta}^i$ and ${}_{LFTD}\boldsymbol{\Theta}^i$ of joint angles for OP+A and LFTD method respectively. Using Equation (3.21) I then calculated the angular jerkiness $J({}_m\boldsymbol{\Theta}^i)$ for $\forall m \in \{\text{OP+A}, \text{LFTD}\}.\forall i \in \{1, 2, \ldots, 40\}$ with the rate f_f set to the video frame rate $f_v = 50$ Hz. The obtained results are shown in Figure 4.7.

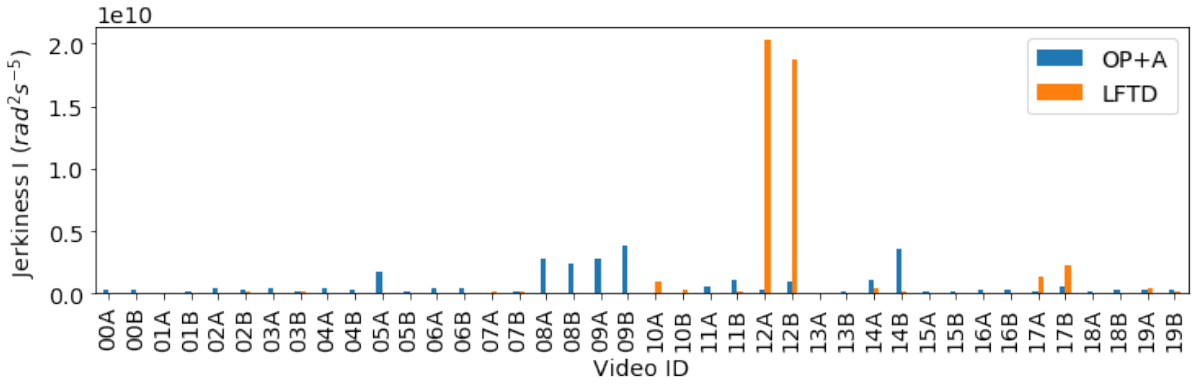


Figure 4.7: Angular jerkiness of reconstructed movements by OP+A and LFTD methods, **for each video** in the dataset, **summed over** all upper-body joint **angles**. Suffixes A and B of video IDs denote task (i) and task (ii) videos of the same subject respectively.

As can be seen from Figure 4.7, the reconstructed movements by LFTD for two videos with IDs 12A and 12B belonging to the same subject are extremely jerky. I also verified this using simulations that showed messy and unrealistic motion reconstructions in complete disagreement with the video recordings of this subject. This altogether shows that for some unknown reason the LFTD method failed to extract the 3D pose for this particular subject. Therefore, for final method choice I removed this subject's videos and reevaluated the overall jerkiness averaged over the remaining 38 videos. As shown by Figure 4.8, the LFTD method provides significantly less jerky movements (p-value < 0.001).

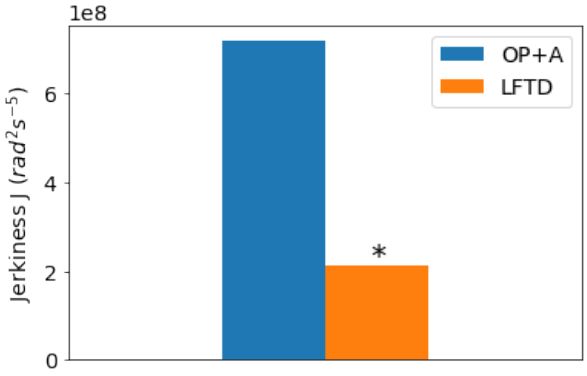


Figure 4.8: Angular jerkiness of reconstructed movements by OP+A and LFTD methods, **averaged over** all but the outlier subject's **videos** in the dataset, **summed over** all upper-body joint **angles**. * denotes significantly better performance (p-value < 0.001).

To further investigate on which joint angles the OP+A method performs worse than LFTD, I evaluated the jerkiness by angles when averaged over the 38 videos. Figure 4.9 shows that although the OP+A method performs better on majority of angles, it provides significantly more jerky trajectories for LElbowYaw and RElbowYaw angles. This indicates that the developed OP+A method is not well suited for reconstruction of movements involving elbow yaw angles and thus neither for the problem addressed in this project.

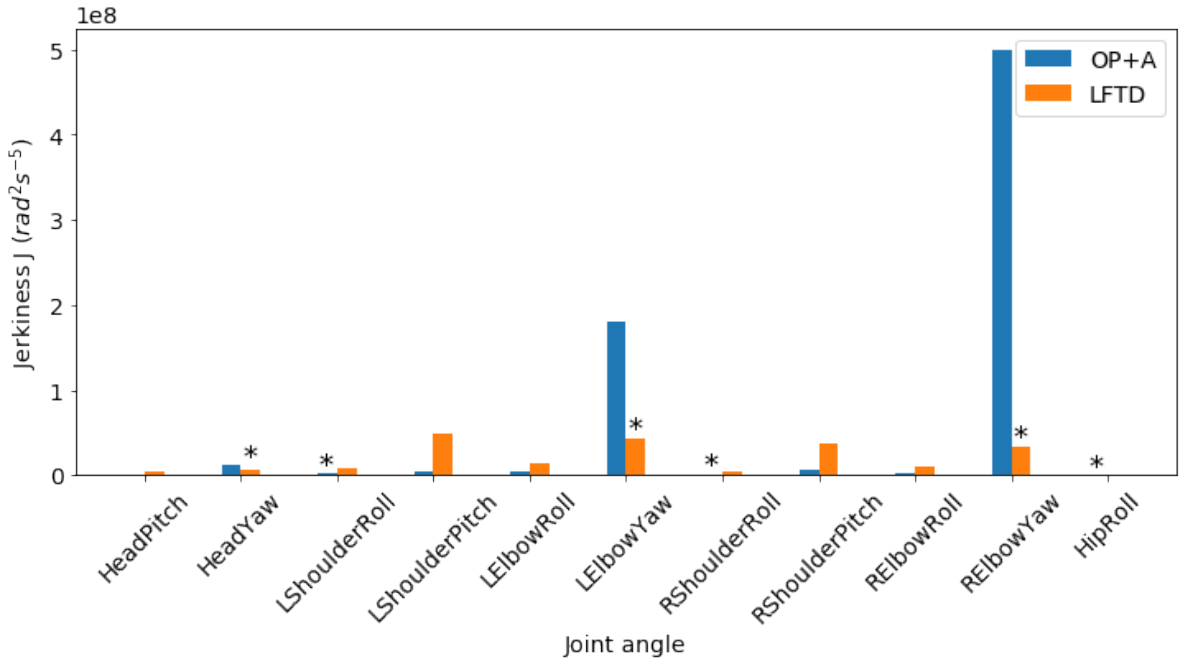


Figure 4.9: Angular jerkiness of reconstructed movements by OP+A and LFTD methods, **for each** upper-body joint **angle**, when **averaged over** all but the outlier subject's **videos** in the dataset. * denotes significantly better performance (p-value < 0.001).

Lastly, to ensure that the lower-jerkiness method does not provide overly smooth movements (as noted in Section 3.4.4), I simulated the pose reconstructions made by both methods for several videos and I did not observe any overly smooth movements. Therefore, for the rest of my project I chose the LFTD method and removed the above-mentioned subject from the dataset as this method failed on their videos.

Synchronisation with audio features

The joint angles were originally extracted at the frame rate, $f_v = 50$ Hz, of the videos while the audio features at rate $f_f = 100$ Hz. In order to train a model between these two data streams, they had to be time-synchronised. I thus upsampled the series of joint angles to $f_f = 100$ Hz using linear interpolation, similarly to [86]. Specifically, between each two consecutive original samples their average was included as a new sample point between them. This type of synchronisation has the advantage of smoother trajectories which is desirable, as compared to mere duplication of the joint angles.

Smoothing

To minimise the chance of learning noisy associations in the subsequent model training, I further smoothed the extracted series of joint angles, as the LFTD method does not

itself ensure temporal consistency. Similarly, smoothing of ground truth movements was performed in the head motion synthesis study [44]. In particular, I applied the 5^{th} order low-pass Butterworth filter with a cut-off frequency f_c using the *SciPy* library [87]. To set an appropriate cut-off frequency, I reviewed the relevant literature: Agarwal et al. [88] report the optimal cut-off frequency of 5 Hz for filtering head movements and Bartlett [89] the range 4-8 Hz as a common choice of cut-off frequencies for filtering measurements of human movements in general. I thus qualitatively compared the smoothed movements for $f_c \in \{4, 5, 6\}$ Hz. The frequency $f_c = 4$ Hz resulted in most natural movements and was thus chosen for smoothing of movements in the whole project. For this cut-off frequency the residual averaged over all joint angles was $R < 4.5°$ which presents a trade-off between the remaining noise and introduced distortion in the smoothed trajectory [90]. The frequency spectra of movements of all 11 joint angles from all videos before and after filtering are shown in Figure 4.10.

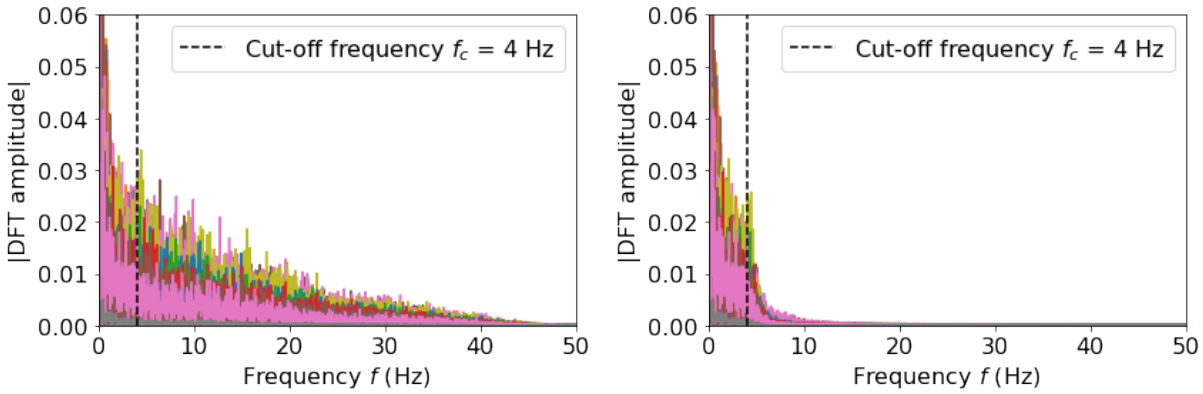


Figure 4.10: Frequency spectra of movements of all 11 joint angles from all videos before (left) and after (right) low-pass filtering with cut-off frequency $f_c = 4$ Hz. Each angle from each video is represented by a separate curve and colour.

Constraining to robot operating limits & normalisation

As a last step, the smoothed joint angles were constrained to the robot's operating limits specified in Table 3.1, and each joint angle was normalised to the range $[0, 1]$ as described in Section 3.4.2, resulting in the pose feature set $\mathbf{\Theta}^{01}$ ready for model training. These two operations ensured that all angles are treated equally while training and also that the predictions later made by the pose regression model (with the sigmoid activation functions at the output layer) will be implicitly constrained to the robot's limits. An example of smoothed and constrained right elbow yaw angle θ_{10} before $[0, 1]$ normalisation is shown in Figure 4.11.

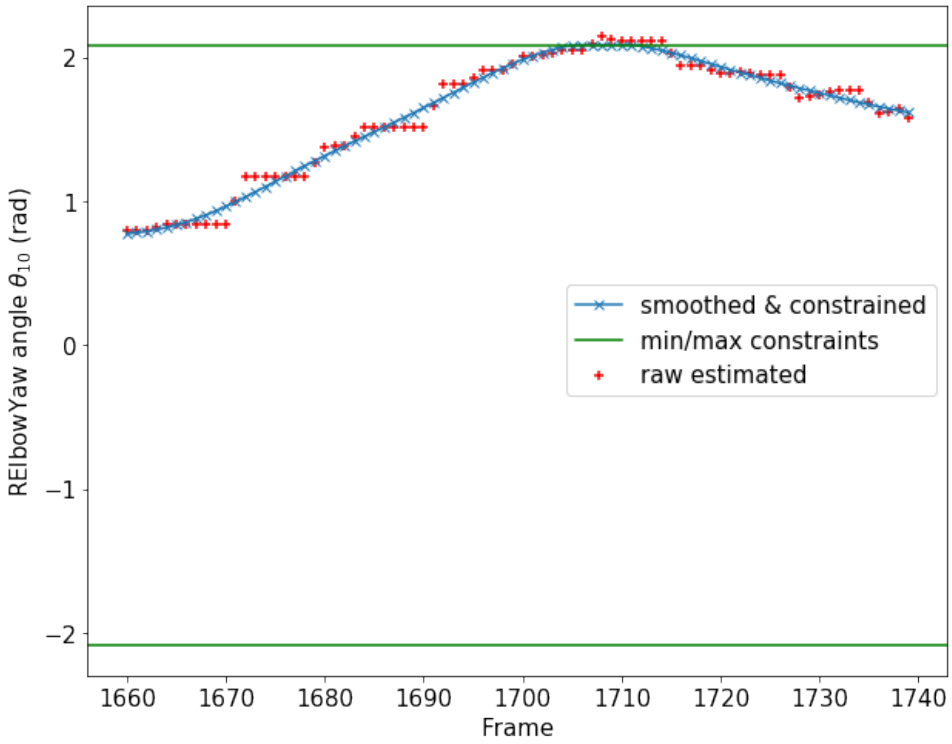


Figure 4.11: Smoothed and constrained right elbow yaw angle θ_{10} before $[0, 1]$ normalisation, when estimated from a sample video.

4.3 Pose regression models

This section describes the development of MLP and LSTM neural network models (Section 3.3) to learn the mapping from audio features $\mathbf{\Omega}$ to pose features $\mathbf{\Theta}^{01}$. For both model types the subject-independent[4] (SI) as well as subject-dependent[5] (SD) model variants were created. All these model combinations will be further referred to using *MLP-SI*, *LSTM-SI*, *MLP-SD* and *LSTM-SD*.

The development of models involved the following procedures: choice of model architecture, comparison of audio feature sets, and investigation of the influence of dropout regularisation (see Appendix A). To choose the best model architectures and to compare the four audio feature sets (Section 4.2.2), I used the SI variants as they are more generalisable. The audio feature sets were compared using only the MLP model, as the comparison using LSTM would be computationally very expensive.

[4]General model trained and evaluated on data from multiple subjects.
[5]Specific model trained and evaluated on each subject separately.

4.3.1 Implementation details and resources

I implemented both MLP and LSTM models in Python using the machine learning library Keras [91]. The training and validation experiments were performed using the cloud computing service AWS provided by Amazon. Specifically, I used the instance *p2.xlarge* with one NVIDIA K80 GPU (12 GB of GPU memory), 4 vCPUs and 61 GiB of RAM memory.

4.3.2 Dataset split

For both SI and SD variants, the dataset was split into training, validation and test partitions as specified in Table 4.4.

Table 4.4: Dataset split.

	Subject-independent (SI)		Subject-dependent (SD)
	# videos	# subjects	Fraction of subject's recordings duration
Training set	30	15	70%
Validation set	4	2	15%
Test set	4	2	15%

4.3.3 MLP

For each audio feature set ${}_{13i}\mathbf{\Omega}$ with $i \in \{1, 2, 4, 6\}$, I first optimised the model architecture on the single-split validation set. The feature sets were then compared via 10-fold subject-independent cross-validation, using the corresponding best model architectures.

In all experiments, I trained the MLP model with $13i$ input-layer units and N_l hidden layers containing N_u units each for a maximum of 10,000 epochs with early stopping with a window size of 10 (I stopped the training if the validation loss did not decrease for 10 consecutive epochs). The mean squared error loss was optimised using Adam optimiser [72] and the whole training partition was used for every iteration (every epoch had one step). For hidden layers I used the ReLU activation function (Section 3.3.1) and He initialisation of weights as suggested by [92]. The output layer with 11 units used the sigmoid activation function σ_S to target 11 angles in the normalised range $[0, 1]$.[6] For evaluation, the predictions were transformed to original angle ranges using ν^{-1} (Section 3.4.2) and smoothed by low-pass filter specified in Section 4.2.3, following other motion synthesis systems [44, 29, 30, 76, 45] that performed post-smoothing.

[6] Similarly to Pham et al. [50] who also used the sigmoid activation function to constrain the network ouputs in the motion synthesis task.

Choice of architecture

For each feature set ${}_{13i}\mathbf{\Omega}$, I searched for the best model architecture in terms of the number $N_l \in \{1, 2, 3, 5, 7\}$ of hidden layers and the number $N_u \in \{8, 16, 32, 64, 128, 256, 512\}$ of units per hidden layer (same for all hidden layers). The maximum values in these ranges were set considering the optimal architectures found by previous works [30, 11] that used larger datasets (Section 2.1). For each combination (N_l, N_u) of these hyperparameters a separate model was trained and evaluated on the validation set (single split specified in Table 4.4). The best hyperparameters found for each feature set are summarised in Table 4.5. We can see that the model capacity increases as the dimensionality of the feature vectors increases which is expected since more information provided on inputs allows the development of more complex models. Comparing to other MLP-based audio-driven motion synthesis systems [30, 11] with hundreds of units per hidden layer, the obtained optimal architectures have much lower capacity which correctly reflects the use of a smaller dataset.

Table 4.5: Optimal MLP-SI model architecture for each audio feature set. N_l denotes the number of hidden layers and N_u the number of units per hidden layer.

Feature set	Optimal N_l	Optimal N_u	Number of parameters
MFCC-13	5	8	499
LogFB-26	7	8	747
LogFB-52	5	16	2123
LogFB-78	7	16	3083

Choice of feature set

Using the best model architecture for each feature set, I performed 10-fold subject-independent cross-validation to compare the feature sets. In every fold the data of 2 subjects were in the validation set and of 15 subjects in the training set. The results averaged over subjects are summarised in Table 4.6, in terms of various evaluation metrics from Section 3.4.

Table 4.6: Comparison of audio feature sets using the MLP-SI model tuned for each feature set. Evaluated by 10-fold subject-independent cross-validation and averaged over subjects. * denotes significantly better performance (p-value < 0.05) relative to all other feature sets.

Feature set	Loss L	RMSE (°)	ΔLCCA	$\text{LCCA}_{\mathbf{\Theta}}$	$\Delta\tilde{J}$ ($10^5\text{rad}^2\text{s}^{-5}$)
MFCC-13	0.0148	13.66	0.027	0.9766	1.331
LogFB-26	0.0146	13.52	0.020	0.9765	1.332
LogFB-52	0.0144	13.44	0.017	0.9756	1.329
LogFB-78	0.0154	13.73	0.005*	0.9751	1.327

As we can see from Table 4.6, all feature sets perform very similarly in each evaluation measure and only the feature set LogFB-78 achieved significantly (p-value < 0.05) better values of ΔLCCA when compared to other three feature sets, suggesting that the motion synthetised using the LogFB-78 feature set is better correlated with the input audio. However, the significant difference only in one measure is not convincing to clearly decide, especially in the pose regression task where multiple metrics have to be considered. I thus further compared the feature sets in terms of the RMSE by angles, as shown in Figure 4.12. We can again see that all feature sets perform very similarly on each angle and moreover, the difference between any two feature sets for none of the angles is statistically significant (p-value > 0.05).

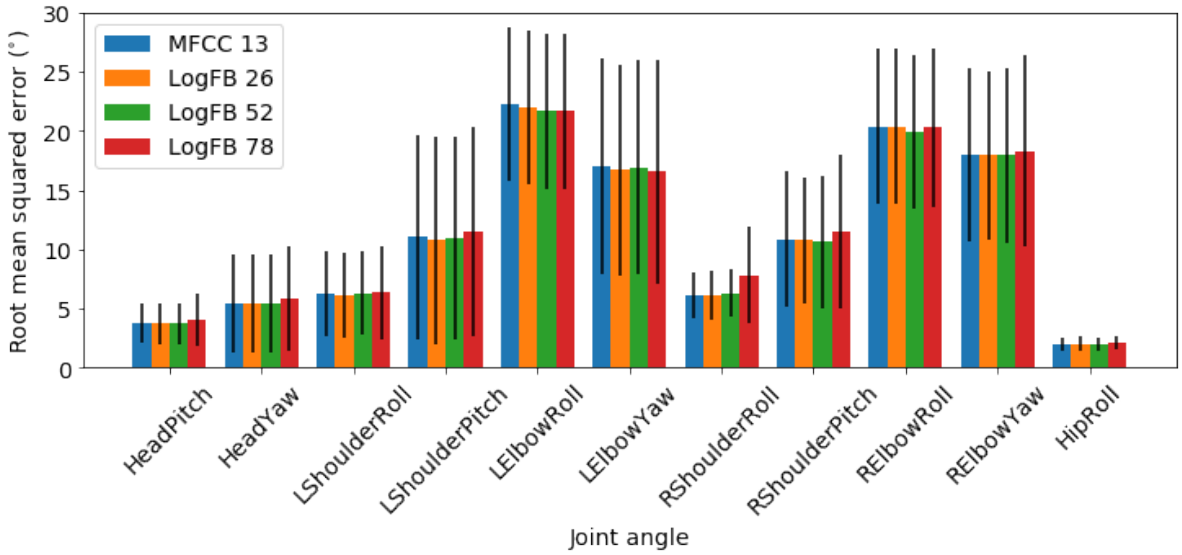


Figure 4.12: Root mean squared error (RMSE) for each joint angle when the MLP-SI model tuned for each feature set was evaluated using 10-fold subject-independent cross-validation and RMSE was averaged over subjects. Difference between any two feature sets for none of the angles is statistically significant (p-value > 0.05).

Additionally, I synthesised the movements predicted using each feature set and the consequent visual inspection also confirmed high similarity between all feature sets. These findings altogether indicate that the choice of feature set is not crucial in this task. Therefore, I based my decision on: (i) the previous studies [93, 94] reporting that LogFB features outperform MFCC and particularly in the speech-to-head-motion task [44] and (ii) the computational complexity involved in later training of LSTM models if high-dimensional feature vectors were used. I thus chose the LogFB-26 ($_{26}\mathbf{\Omega}$) audio feature set for the rest of this project.

Finally, the MLP-SD models were trained for each subject, using the feature set $_{26}\mathbf{\Omega}$ and the corresponding optimal architecture ($N_l = 7, N_u = 8$). The same training settings were used as for the MLP-SI.

4.3.4 LSTM

For the chosen audio feature set ${}_{26}\mathbf{\Omega}$, I first segmented the dataset (both audio ${}_{26}\mathbf{\Omega}$ and pose $\mathbf{\Theta}^{01}$ feature sets in the same way) into sequences of N_τ frames with the stride of 1 frame. According to the reviewed gesture durations in Section 3.4.3, I set $N_\tau = 300$ frames (corresponding to 3 s at frame rate $f_f = 100$ Hz) so that a whole gesture can be captured within one sequence used by LSTM.

Next, the LSTM network with one hidden layer[7] of $N_{u'}$ LSTM units was trained for a maximum of 100 epochs with early stopping with the window size of 10. The mean squared error loss was optimised using Adam optimiser [72] and the training batch size was set to 15,000 sequences (limited by the available memory). As for the MLP model, the output layer with 11 units applied the sigmoid activation function to target 11 angles in the normalised range $[0, 1]$ and the predictions were then transformed to original angle ranges and smoothed.

Choice of architecture

I searched for the best model architecture in terms of the number of LSTM units in the hidden layer, namely, $N_{u'} \in \{3, 6, 9, 12, 15, 18, 21, 24, 27\}$. This range was chosen based on the optimal network capacity found for MLP (Table 4.5) and the amount of training data. For each hyperparameter $N_{u'}$ a separate model was trained and evaluated on the validation set (single split specified in Table 4.4). The results in Figure 4.13 clearly indicate that the choice $N_{u'} = 12$ presents the best LSTM model achieving the minimum validation loss.

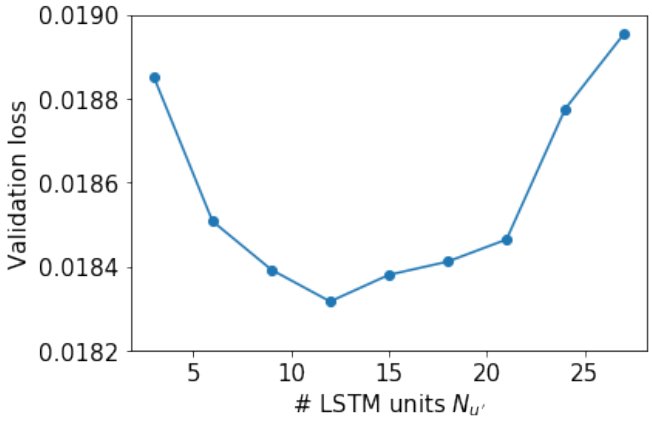


Figure 4.13: Validation loss of various LSTM-SI model architectures in terms of the number $N_{u'}$ of LSTM units in the hidden layer, using the chosen audio feature set ${}_{26}\mathbf{\Omega}$.

[7] I chose one hidden layer based on the optimal model architectures found for MLP (Table 4.5) and the amount of training data.

Finally, the LSTM-SD models were trained for each subject using the optimal architecture ($N_{u'} = 12$). The same training settings were used as for the LSTM-SI except the following: the maximum number of epochs was increased to 500 as the training of SD models did not converge as fast as for SI and each training batch now contained all training sequences.

4.4 Synthesis on the Pepper robot

This section details the synthesis phase of the system outlined in Figure 4.1. The following two developed synthesis modes are described.

- **Offline synthesis:** the whole audio input is available upfront, and the prediction and synthesis of movements are performed at any later time.
- **Online synthesis:** the movements are predicted and synthesised on-the-fly while the input audio is being captured.

4.4.1 Offline synthesis

Since the whole audio input is available in advance, the whole audio feature set ${}_{26}\boldsymbol{\Omega}$ is extracted using the procedure from Section 4.2.2 for one recording. The trained pose regression model is then applied to obtain the set of predictions $\boldsymbol{\Theta}^{01}$ corresponding to the whole audio input. After the inversion of the $[0, 1]$ normalisation via ν^{-1} (Section 3.4.2), the predicted angles are smoothed using the low-pass filter specified in Section 4.2.3, following other motion synthesis systems [44, 29, 30, 76, 45] that performed post-smoothing. Lastly, the commands specifying the pose vectors $\boldsymbol{\theta}$ are sent to the robot to perform the target movements. To ensure synchrosnisation with the audio playback running in parallel, these commands are clocked at the original feature frame rate $f_f = 100$ Hz using the adaptive sleep time $t_s(t, i)$ so that at time t from the synthesis start the sleep time before the synthesis of the i^{th} frame is set to

$$t_s(t,i) = \begin{cases} \eta((i-1)f_f^{-1} - t) & \text{if } (i-1)f_f^{-1} > t \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where the parameter $\eta = 0.95$ was determined empirically.

4.4.2 Online synthesis

In the online synthesis mode the audio is captured in real-time using the library *pyaudio* [95], with the same settings as the audio used for model training, namely, single

channel and $f_s = 16$ kHz sampling rate (Section 4.2.2). At each time-step (clocked at $f_f = 100$ Hz) the most recent $wf_s = 400$ audio samples (fitting into one audio feature extraction window w) are used to calculate a single audio feature vector ${}_{26}\boldsymbol{\omega}$ of 26 LogFB features. This feature vector is then z-normalised using the precomputed features' means and standard deviations (calculated over the whole dataset [5]). The normalised feature vector is in turn fed into the pose regression model and the output is inverted as in the offline mode. Differently from the offline mode, the subsequent smoothing is performed using the Kalman filter, as it is commonly applied in real-time motion synthesis systems [88, 96]. Lastly, the commands specifying the pose vector $\boldsymbol{\theta}$ are sent to the robot and the adaptive sleep time is applied as in the offline mode.

Chapter 5

Evaluation

The developed system was evaluated quantitatively and also qualitatively via web-surveys. All four model types (MLP-SI, LSTM-SI, MLP-SD, LSTM-SD) are compared quantitatively in Section 5.1 and qualitatively in Section 5.2. Additionally, Section 5.2.2 examines the relationships between evaluation metrics and Section 5.3 investigates how the speaker's personality traits affect the synthesised motion, using both quantitative and qualitative measures.

5.1 Quantitative evaluation

This section compares the models in terms of various evaluation metrics in Section 5.1.1 and real-time synthesis latency in Section 5.1.2.

5.1.1 Model comparison: SI vs SD and MLP vs LSTM

All four models (MLP/LSTM-SI/SD) were evaluated on the test partition of the employed dataset [5]. The SI models were trained and tested using 10-fold protocol so that the model was evaluated on each subject. The results for each of the four models were then averaged over subjects and compared as shown in Table 5.1 and Figure 5.1.

Table 5.1: Quantitative model comparison on test set, averaged over subjects. * denotes significantly better performance (p-value < 0.001) between MLP and LSTM model for each variant (SI/SD) separately. † denotes significantly better performance (p-value < 0.05) between the SI and SD variant of the same model type (MLP/LSTM).

Model	Loss L	RMSE (°)	ΔLCCA	LCCA$_{\boldsymbol{\Theta}}$	$\Delta\tilde{J}$ (10^5rad^{2}s^{-5})
MLP-SI	0.0150	13.71	$0.016^{\dagger *}$	0.9757	1.333
LSTM-SI	0.0151	13.70	$0.042^{\dagger}$	0.9817*	1.330
MLP-SD	$0.0068^{\dagger}$	$9.17^{\dagger}$	0.024*	0.9760	1.019
LSTM-SD	$0.0073^{\dagger}$	$10.23^{\dagger}$	0.049	0.9818*	1.016

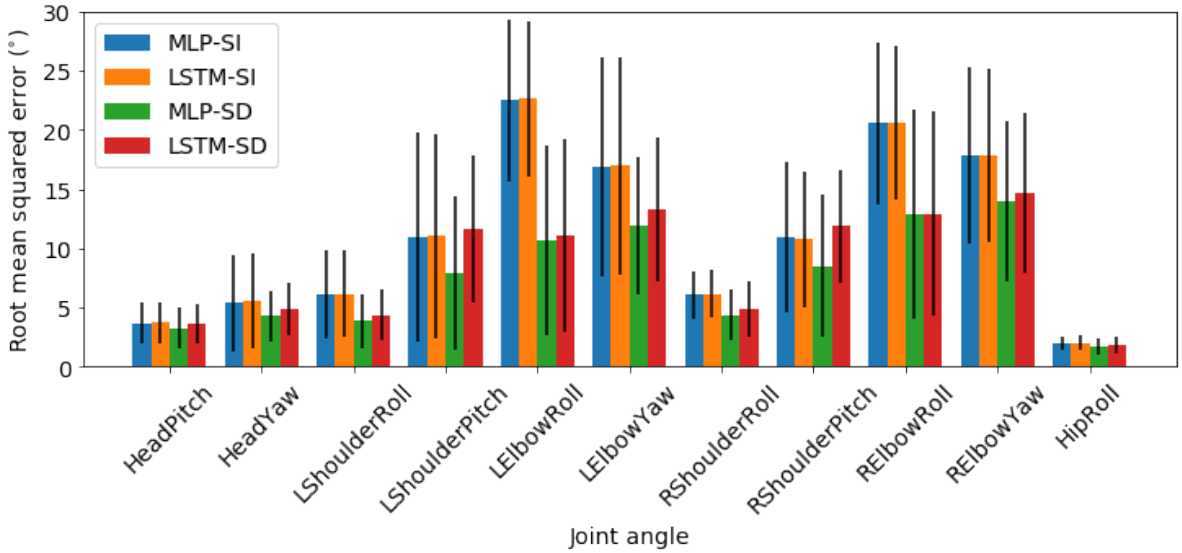


Figure 5.1: Root mean squared error (RMSE) for each joint angle, for each model evaluated on test set. RMSE was averaged over subjects.

In general, the SD variant is expected to perform better than the SI variant of the same model, which is confirmed by significant differences (p-value < 0.05 denoted by †) in loss and RMSE metrics in Table 5.1. However, the ΔLCCA measure directly and significantly contradicts such expectation which suggests that the ΔLCCA may not be suitable evaluation measure for this task.

Table 5.1 further shows that the difference between MLP and LSTM models (for both variants SI/SD) is relatively small. Specifically, for loss, RMSE and $\Delta\tilde{J}$ measures the differences between MLP and LSTM are not statistically significant (p-value > 0.05). However, according to $\text{LCCA}_{\mathbf{\Theta}}$ the LSTM is significantly better whereas ΔLCCA suggests the opposite (p-value < 0.001). Again, the ΔLCCA measure contradicts the expectation that the LSTM model would perform better than MLP.

To summarise, the quantitative evaluation showed that SD variants perform better than SI. However, the quantitative comparison of the MLP and LSTM model did not provide a convincing conclusion about which of them generates better movements. Therefore, the qualitative evaluation was essential to provide final conclusions. Furthermore, it allowed for better assessment of the suitability of the ΔLCCA evaluation metric.

5.1.2 Real-time synthesis latency: MLP vs LSTM

I further evaluated the latency of motion synthesis by MLP and LSTM models operating in the online synthesis mode (Section 4.4.2). In particular, I measured[1] the model inference latencies τ_{MLP} and τ_{LSTM} of the MLP-SI and LSTM-SI model respectively, over 10,000

[1]All the measurements were made on my laptop with quad-core 2.2GHz Intel i7 CPU and 8GB RAM, without a GPU support for Keras [91].

inferences. As shown by Figure 5.2, the MLP model makes predictions significantly faster than LSTM (p-value < 0.001), which reflects the fact that the LSTM model is much more complex. Comparing the obtained inference latencies $\tau_{MLP} = 1 \pm 1$ ms and $\tau_{LSTM} = 41 \pm 10$ ms to the frame period $\tau_f = f_f^{-1} = 10$ ms used to develop the models, it is evident that the LSTM model cannot make real-time predictions at such a rate, whereas the MLP seems to be able to. For the MLP model, I thus also measured the latency τ_{ops} of all other per-frame operations (audio stream read, z-normalisation, Kalman filtering, dispatch of commands to the robot) and the overall per-frame processing latency τ'_{MLP}, again over 10,000 inferences:

$$\tau_{ops} = 9 \pm 6 \text{ ms} \qquad \tau'_{MLP} = 10 \pm 6 \text{ ms} \tag{5.1}$$

As we can see, although the latency of other per-frame operations τ_{ops} is significantly higher than the MLP inference latency τ_{MLP} (p-value < 0.001), the overall MLP per-frame processing latency τ'_{MLP} is comparable with the frame period τ_f which suggests that the MLP model is suitable for real-time motion synthesis at the rate it was trained. If the LSTM model had to be used, the movements would have to be synthesised at about 5-times slower rate[2] ($\sim$ 20 Hz), which might still be acceptable but of a very low quality as the human vision begins to perceive series of images as a motion at around 15 images per second [97].

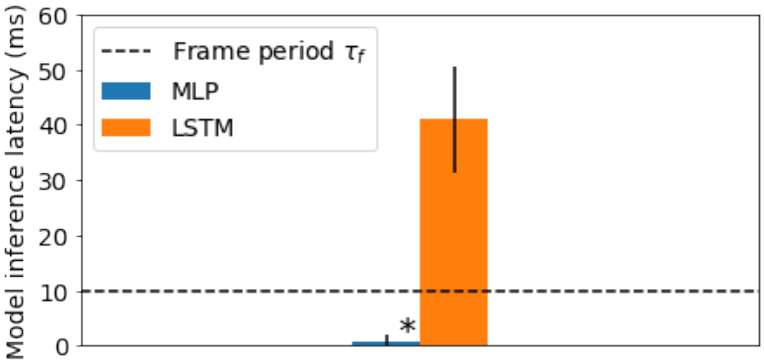


Figure 5.2: Model inference latency, averaged over 10,000 inferences. MLP model: $\tau_{MLP} = 1 \pm 1$ ms. LSTM model: $\tau_{LSTM} = 41 \pm 10$ ms. Frame period $\tau_f = f_f^{-1} = 10$ ms, used to develop the models. * denotes significantly better performance (p-value < 0.001).

Out of the related works in Section 2.1, only Pham et al. [48] reported latency measurements. Specifically, they measured the inference latency of 5 ms for their real-time LSTM-based head and facial movements synthesis sytem, which is considerably smaller than the latency $\tau_{LSTM} = 41 \pm 10$ ms. However, they made predictions using GPU which significantly lowers the inference latency. Due to the dependence on computational resources, I thus report only the relative comparison that the MLP model can perform the online motion synthesis approximately 5-times faster than LSTM.

[2] Assuming that the latency τ_{ops} of other per-frame operations is similar for both models.

5.2 Qualitative evaluation: web-surveys

The developed system was qualitatively evaluated via anonymous web-surveys with the following aims:

- To qualitatively compare MLP and LSTM models (Section 5.2.1).
- To examine relationships between qualitative and quantitative evaluation measures (Section 5.2.2).
- To compare the behaviour of the system driven by natural speech with the one driven by synthetic speech (Section 5.2.3).
- To investigate how the speaker's personality traits affect the perceived appropriateness of the synthesised movements for the speech (Section 5.3).

Using Google Forms I created the following two surveys:

1. **Natural speech based survey**
   For evaluation based on natural speech I used audio recordings from the original dataset [5]. Firstly, the movements of the Pepper robot generated by each of the four models (MLP-SI, LSTM-SI, MLP-SD, LSTM-SD), trained as in Section 5.1.1, were recorded[3] in the offline synthesis mode while predicting on the test partition of each subject's audio recording. This resulted in 4 (models) $\times$ 38 (videos) = 152 short audio-visual clips (SAVCs) of less than 15 s each. For each of the 38 original audio recordings, I then created two side-by-side (SBS) videos:

   (i) SAVC by MLP-SI & SAVC by LSTM-SI

   (ii) SAVC by MLP-SD & SAVC by LSTM-SD

   The position (left/right) of the SAVC by MLP was chosen randomly for each SBS video. All 76 SBS videos were then randomly shuffled and inserted into the survey. An example of the SBS video is shown in Figure 5.3.

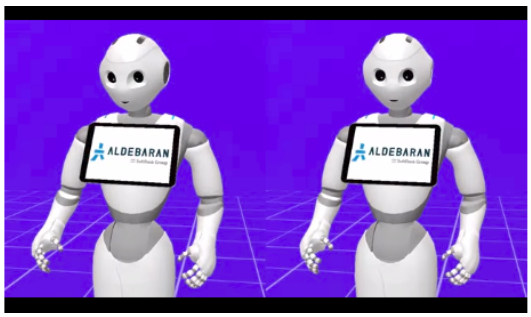


Figure 5.3: Sample side-by-side (SBS) video. Movements synthesised by the MLP and LSTM model on randomly chosen left/right side.

[3]To record the robot I used the program *SimpleScreenRecorder* [98] along with the *pyautogui* library [99] to automate the recording process.

The survey respondents were first given instructions and told which joints of the Pepper robot are controlled in order to prevent them from assessing movements not supported. They were then asked to watch each SBS video and assess the appropriateness of the generated movements for the given audio, on a Likert scale ranging from 1 = very inappropriate to 5 = very appropriate, for *each* side of the SBS video. A sample survey question is shown in Appendix C.2.

2. **Synthetic speech based survey**
   Firstly, I aggregated a set of short stories, namely, four of the Strange Stories [100]. Texts of these chosen stories are provided in Appendix C.3. I then used the open-source text-to-speech system MaryTTS 5.2 [101] to create synthetic audio recordings. The MaryTTS system provides four voice types each represented by one character with a specific personality type [102], namely,

   - Obadiah (male) is gloomy and depressed,
   - Spike (male) is angry and argumentative,
   - Prudence (female) is pragmatic and practical,
   - Poppy (female) is outgoing and optimistic.

   To capture larger variety of speaking styles I synthesised speech by each of these characters for every story, with further MaryTTS settings of HMM-based models and Great Britain English accent. This resulted in the set of 16 audio recordings of less than 40 s each.

   Analogously to the natural speech based survey, I created 16 SBS videos comparing the MLP-SI and LSTM-SI models.[4] The survey respondents were asked to perform the same task as in the natural speech based survey.

The Ethics Committee approval for both surveys is provided in Appendix C.1.

In total, I collected 21 and 44 responses for the natural and synthetic speech based survey respectively. The responses with

- the same appropriateness value for *all* left sides of SBS videos,
- or the same appropriateness value for *all* right sides of SBS videos

were considered as unreliable and were excluded, resulting in the final set of 20 and 43 responses for the natural and synthetic speech based survey respectively.

[4] In this case it was not possible to use the SD models as none of them was trained on none of the character's data.

5.2.1 Model comparison: SI vs SD and MLP vs LSTM

The comparison of models based on the results from the natural speech based survey is shown in Figure 5.4. We can see that even though the survey was not designed to compare SI with SD model variants, the movements synthesised by the SD variant were assessed as significantly more appropriate for the audio than those by the SI variant (p-value < 0.001 denoted by †). Also, some participants reported that they noticed better and more pronounced gestures referring to the SD model variants, even though they were not told which SBS videos were comparing SI and which SD models. This all confirms the quantitative results in Section 5.1.1 and altogether suggests that it is best to develop subject-specific models for this task.

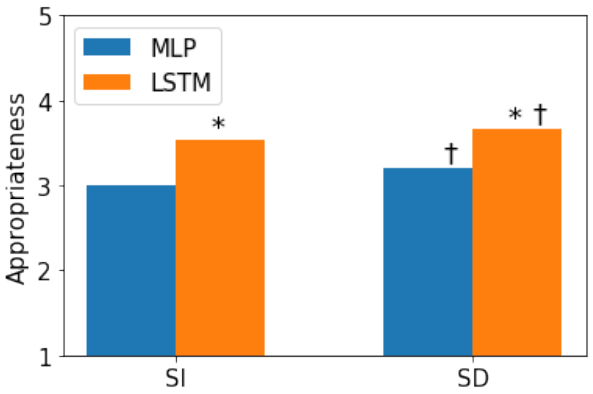


Figure 5.4: Appropriateness of the generated movements for the given audio, assessed in the natural speech based survey on a Likert scale ranging from 1 = very inappropriate to 5 = very appropriate. * denotes significantly better ratings (p-value < 0.001) between the MLP and LSTM model, for each model variant (SI/SD) separately. † denotes significantly better ratings (p-value < 0.001) between the SI and SD variant of the same model type (MLP/LSTM).

Most importantly, we can observe that for each variant (SI/SD) the movements generated by the LSTM model were considered significantly more appropriate for the audio than those by the MLP model (p-value < 0.001). Also, some respondents explicitly noted that one side (although not always the same) of the SBS video always showed smoother movements, most likely referring to the LSTM-based models. These results show that even though there were no clear quantitative differences between the two models, humans perceive considerable differences between them rating LSTM as the better model. This agrees with other works [30, 11] reporting that LSTM-based models outperform MLP models in the head motion synthesis tasks. Moreover, the obtained result generalises these findings to the whole upper-body motion synthesis.

5.2.2 Relationships between evaluation metrics

Relating the obtained qualitative results to the quantitative results (Table 5.1), it can be concluded that:

- Comparing the SI with SD model variants, significantly higher appropriateness is associated with significantly lower loss and RMSE but with significantly higher ΔLCCA. The negative correlation between appropriateness and each of loss and RMSE can be explained by the fact that the generated movements that are closer to the ground truth movements (lower loss and RMSE) are perceived as more appropriate to the audio input.
- Comparing the MLP with LSTM models, significantly higher appropriateness is associated with significantly higher $\text{LCCA}_{\mathbf{\Theta}}$ as well as ΔLCCA. The positive correlation between appropriateness and $\text{LCCA}_{\mathbf{\Theta}}$ can be explained by the fact that the generated movements that are more correlated with the ground truth movements (higher $\text{LCCA}_{\mathbf{\Theta}}$) are assessed as more appropriate to the audio input.

In both cases the positive correlation between appropriateness and ΔLCCA means that the generated movements whose audio-with-generated-movements correlation better matches the audio-with-ground-truth-movements correlation are *not* perceived as more appropriate to the audio. This is a somewhat contradictory finding indicating that the ΔLCCA measure is probably not suitable for evaluation of movements driven by audio. However, to make such a general conclusion more experiments need to be performed.

5.2.3 Natural vs synthetic speech

For the SI model variant, the preference for the LSTM model (demonstrated by Figure 5.4) was compared with the results from the synthetic speech based survey. As can be seen from Figure 5.5, in the case of synthetic speech the movements generated by the MLP model were assessed as more appropriate than those by the LSTM, which directly contradicts the results based on the natural speech. This can be explained by the combination of the following three facts:

(i) the synthetic speech is more machine-like and choppy than the natural speech as the changes in intonation and stress patterns are not yet fully understood [103], it further lacks the natural phonetic variability [104, 105] and it is also less intelligible than the natural speech [105];

(ii) the MLP model generates more machine-like and less smooth movements than LSTM as each frame is treated independently (Section 3.3.1) which can be also supported by the results from Section 5.2.1 and previous works [30, 11];

(iii) the respondents were asked to assess how appropriate the movements are for the speech which meant that they rated the *machine-like speech* to match the *machine-like movements* better than the more natural movements generated by LSTM.

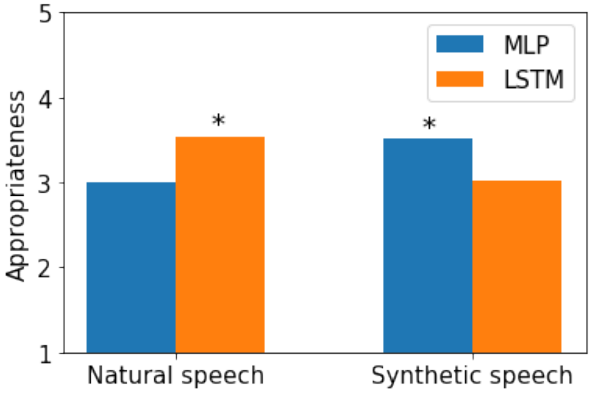


Figure 5.5: Appropriateness of the generated movements for the given audio, assessed in natural and synthetic speech based surveys on a Likert scale ranging from 1 = very inappropriate to 5 = very appropriate, comparing the SI variants of the MLP and LSTM models. * denotes significantly better ratings (p-value < 0.001) between the MLP and LSTM model, separately for natural and synthetic speech based evaluation.

5.3 Relationship to personality

In this section I investigate the relationship between speakers' personality traits and the generated movements (in terms of quantitative and qualitative evaluation metrics), for each model type. This investigation is performed for the natural speech based motion synthesis using the subjects from the employed dataset (Section 5.3.1) and for the synthetic speech based motion synthesis using the four characters from MaryTTS (Section 5.3.2).

5.3.1 Natural speech based motion synthesis

Given the subjects' five personality traits on the 1-10 scale (Section 4.2.1), for each trait I first calculated the mean over all subjects and then categorised all 19 subjects into low and high classes, thresholding on that mean.[5] For each evaluation metric (loss, RMSE, $\text{LCCA}_{\mathbf{\Theta}}$, ΔLCCA, $\Delta\tilde{J}$, and appropriateness from the natural speech based survey), I compared the measures obtained for subjects from the low personality trait category with those from the high category. This was performed for each personality trait and for each of the four model types. Out of all 120 comparisons[6] only the relationships shown in Table 5.2 were statistically significant (p-value < 0.05). We can see that even

[5] The same procedure was also used in the work [5].

[6] 5 (personality traits) $\times$ 6 (evaluation measures) $\times$ 4 (model types) = 120 (comparisons)

though none of the models was developed with explicitly providing the information about subjects' personalities, the MLP models have implicitly learned four relationships.

Table 5.2: Statistically significant (p-value < 0.05) relationships between personality traits and evaluation metrics when the movements were synthesised by a particular model.

Relationship	Model
(R1) High openness and lower ΔLCCA	MLP-SI
(R2) High openness and lower ΔLCCA	MLP-SD
(R3) High conscientiousness and higher $\text{LCCA}_{\mathbf{\Theta}}$	MLP-SD
(R4) High conscientiousness and higher appropriateness	MLP-SD

The relationships R1 and R2 suggest that for individuals with high openness (more imaginative, curious, open-minded) the MLP models predict movements that are better correlated (closer to the ground truth correlation) with the input audio than for individuals with low openness. However, as the ΔLCCA metric is likely to be unreliable (Section 5.2.2) the meaning of these two relationships is questionable. According to relationships R3 and R4, the MLP-SD model generates movements that are more correlated with the ground truth movements and are also assessed as more appropriate to the input audio for more conscientious people (better organised, more reliable) than for less conscientious. We can also notice that these two relationships reflect the positive correlation between appropriateness and $\text{LCCA}_{\mathbf{\Theta}}$, described in Section 5.2.2.

According to the previous research [106], the openness and agreeableness personality traits are positively correlated with the smoothness of body movements and neurotiscism tends to be positively correlated with jerky and accelerated movements. However, the developed system does not seem to capture these relationships, namely, no significant relationships involving the $\Delta\tilde{J}$ measure were found. This indicates that in order to generate personality-specific behaviours, a separate model might need to be developed for each personality type (for further details see Section 6.2).

5.3.2 Synthetic speech based motion synthesis

In the case of synthetic speech, only the appropriateness measure can be compared between the four characters and thus between the four personality types they represent. Figure 5.6 shows such comparisons for the MLP-SI and LSTM-SI model. We can see that movements by all the characters were assessed similarly and there is no significant difference between neither pair of the characters for neither model (p-value > 0.1). These results indicate that the four personality types do not significantly affect the perceived appropriateness of movements for the synthetic speech. This can be explained by the fact

that the MLP-SI and LSTM-SI models were developed in SI manner (using data from multiple subjects) and without any personality inputs. Indeed, there may not even exist any differences in the perceived appropriateness of movements for the input audio between these four personality types. To investigate this further, personality-specific models could be used, as suggested in Section 5.3.1.

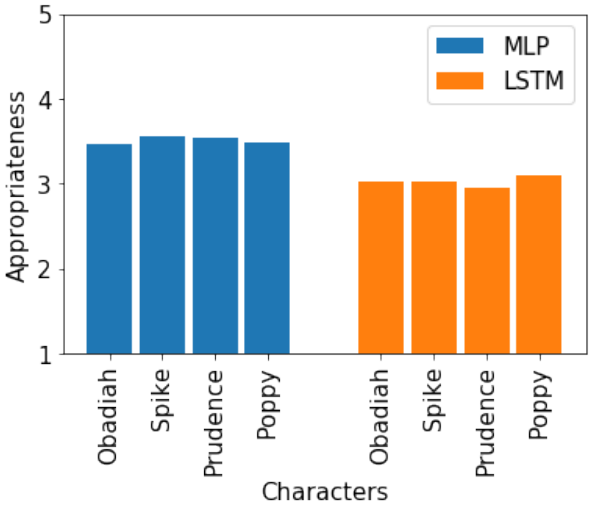


Figure 5.6: Appropriateness of the generated movements for the given audio, comparing four characters representing four personality types (Section 5.2). Assessed in the synthetic speech based web-survey on a Likert scale ranging from 1 = very inappropriate to 5 = very appropriate. Difference between neither pair of the characters for neither model is statistically significant (p-value > 0.1).

Chapter 6

Conclusion

This chapter summarises the accomplished work and results in Section 6.1, and proposes further ideas, possible improvements and extensions in Section 6.2.

6.1 Summary

In this project I developed an automatic audio-driven upper-body motion synthesis system targeted to the humanoid robot Pepper. In particular, the system takes audio input from its user and uses the trained neural network to predict time-series of 11 angles between upper-body joints that are used to control the robot upper-body pose. This system is first of its kind in two ways: (i) it performs whole upper-body motion synthesis including head, hand and hip movements, unlike the previous works that synthesised either head or hand movements; and (ii) it is targeted to a humanoid robot, unlike other existing works that focused on animated talking avatars. The system was developed using only single-view RGB videos of upper-body movements of 19 speakers and it supports an offline as well as online (real-time) synthesis mode.

Using the audio-visual dataset of Bremner et al. [5], I first extracted audio and pose features. Specifically, I extracted and compared four types (Section 4.2.2) of audio features. For pose feature extraction, I compared four 3D pose estimation methods (Section 4.2.3) that estimate 3D joint positions of the human skeleton from a single-view RGB video. The estimated 3D joint positions were used to calculate 11 angles between upper-body joints and the obtained angle time-series were then smoothed and constrained to the robot's operating limits. To learn the mapping between audio features and upper-body pose, I trained the MLP and LSTM neural network models in subject-independent (SI) and subject-dependent (SD) manner.

The results of my research during the system development follow.

- Comparisons of the four 3D pose estimation methods showed that the method *Lifting from the deep* (LFTD) [1] is best suited for the development of upper-body pose regression models, because it correctly handles videos with missing body joints and results in the least jerky reconstructed movements.

- The comparison of the four audio feature sets did not provide convincing conclusions about which of these sets is best suited for the audio-to-upper-body-motion learning task.

The developed system was evaluated by several quantitative measures when driven by natural speech and qualitatively using web-surveys when driven by natural as well as synthetic speech. In particular: I compared the SI with SD model variant and the MLP with LSTM model type; I investigated the relationships between quantitative and qualitative evaluation metrics; and I also examined how the speaker's personality traits affect the synthesised movements. The results and findings follow.

- The SD model variants outperform SI for both MLP and LSTM model types, suggesting that it is best to develop subject-specific models for this task.

- Quantitative comparison of the MLP and LSTM model did not clearly show which of them generates better movements. However, the MLP model is better suited for real-time motion synthesis than LSTM, as it allows approximately 5-times faster online synthesis.

- On natural speech, the movements generated by the LSTM model were assessed as significantly more appropriate for the given audio than those generated by the MLP model and this was the case for both SI and SD model variants. This result generalises the findings of previous speech-to-head-motion works to the whole upper-body motion synthesis.

- On synthetic speech, the survey respondents preferred the MLP model over LSTM, which reflects the fact that the more machine-like movements generated by the MLP model better match the more machine-like synthetic speech.

- Relating the quantitative and qualitative results, I conclude that the synthesised movements that are closer to the ground truth movements (lower loss and RMSE measures) or that are more correlated with the ground truth movements (higher $\text{LCCA}_{\boldsymbol{\Theta}}$ measure) are perceived as more appropriate to the audio. Furthermore, the obtained results clearly indicate that the ΔLCCA metric is not suitable for evaluation of the upper-body movements driven by audio, as it is in significant disagreement with other metrics.

- I found two significant relationships between the speaker's personality traits and the motion synthesised for the speaker. For speakers with high conscientiousness trait the MLP-SD model generates movements that are significantly more correlated with the ground truth movements than for speakers with low conscientiousness trait. The movements synthesised by this model for speakers with high conscientiousness trait are further perceived as more appropriate for the input audio. However, to generate behaviours reflecting a particular personality type, personality-specific models need to be developed.

6.2 Future work

This work opens up several directions for further research.

- **Articulatory features:** Previous research has shown that articulatory features are more related to the head motion than acoustic features [107]. It might be thus beneficial to extract and incorporate such features into the system and investigate their impact on system performance. For example, they can be estimated using the acoustic-to-articulatory inversion technique [107].

- **Neural network models:** *Loss function:* Since there is no single evaluation metric to assess the predicted movements, various neural network loss functions could be compared and new ones developed, possibly combining multiple evaluation metrics. This would lead to novel conclusions about which loss function is best suited for the audio-to-motion learning task. *Model types:* Another research direction could experiment with generative models such as CVAE [46] that allow the prediction of several motion trajectories for the same audio input. This would well reflect the many-many mapping nature of the problem and enrich the system capabilities. Also, the suitability of attention mechanisms that are gaining popularity in various deep learning areas [108, 109, 110] could be researched for this task.

- **Text analysis:** Even though the synthesised movements may be well synchronised with the speech, they may be uncorrelated with the meaning of the message being communicated or even contradict the meaning (e.g. head nodding for disagreement). Therefore, besides the analysis of raw audio signal, the system could also analyse the text of the speech to extract the meaning of the message and thus generate more meaningful behaviours. Inspired by the hybrid speech-to-head-motion models [29, 47], a set of dialog acts [111] could be extracted from the text using supervised classifiers and then used to constrain the models to generate movements from a particular category of motion patterns.

- **Emotional channel:** It was demonstrated that the incorporation of affect information into the speech-to-hand-gestures synthesis system improves the gesture modelling [76]. An interesting extension would thus be to modulate the motion synthesis by continuous affect information (e.g. adding valence and arousal as input channels) to generate more expressive gestures and to make the system more customisable. For instance, the user could choose what kind of emotions they want to express via the synthesised movements.
- **Personality-specific models:** Body motion is an important indicator of personality type [112, 113] and moreover, people can assign personalities to robots according to their movements [114, 115]. Furthermore, matching the personality of the robot with the personality of the interacting human has a positive effect on the interaction [116]. Therefore, as also suggested in Section 5.3, the system could have several models each trained on data of a particular personality type, so that the user could then choose which personality they want to embody. Evaluation of such a system would then involve a personality perception study, where the robot appearance would have to be carefully taken into account as explained by Bremner et al. [115].
- **Evaluation and training resources:** The fact that the system was developed using single-view RGB videos allows it to be evaluated or further trained on various kinds of data such as weather forecasts, TV news or discussions. This would enable the development of a wide range of task-specific, subject-dependent or personality-specific models.
- **Movements in listening mode:** To capture all parts of the human-robot interaction, the system could be extended with the listening mode, so that the robot also generates movements while listening to the human involved in the human-robot communication. This is a very promising extension since the previous study [46] demonstrated that the audio signal can be used to predict not just the head motion of the speaker but also the head motion of the listener.
- **Whole body gesticulation:** The system could be further extended to generate gestures using the whole body, so that the robot would not be constrained to a fixed position but could slightly move around to express more natural behaviour during conversations. This would require a new dataset with audio-visual recordings of the whole body.

Bibliography

[1] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[3] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *CVPR*, volume 2, page 6, 2017.

[4] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. volume 36, July 2017.

[5] Paul Bremner, Oya Celiktutan, and Hatice Gunes. Personality perception of robot avatar tele-operators. In *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*, pages 141–148. IEEE, 2016.

[6] Lucian Dinu. Icons for everything. `https://thenounproject.com/term/speak/1064313/`. [Online; accessed 25/05/2018].

[7] Softbank robotics. `https://www.ald.softbankrobotics.com/en`. [Online; accessed 27/04/2018].

[8] Pepper robot documentation. `http://doc.aldebaran.com/2-4/family/pepper_technical/joints_pep.html`. [Online; accessed 27/04/2018].

[9] Choregraphe simulation environment. `http://doc.aldebaran.com/2-5/software/choregraphe/index.html`. [Online; accessed 27/04/2018].

[10] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[11] Chuang Ding, Pengcheng Zhu, and Lei Xie. Blstm neural networks for speech driven head motion synthesis. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[12] Christopher Olah. Understanding lstm networks. `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`. [Online; accessed 26/05/2018].

[13] BA Sherwood and Ruth Chabay. VPython – 3D programming for ordinary mortals. `http://vpython.org/`. [Online; accessed 07/05/2018].

[14] David McNeill. *Hand and mind: What gestures reveal about thought*. University of Chicago press, 1992.

[15] Kevin G Munhall, Jeffery A Jones, Daniel E Callan, Takaaki Kuratate, and Eric Vatikiotis-Bateson. Visual prosody and speech intelligibility: Head movement improves auditory speech perception. *Psychological science*, 15(2):133–137, 2004.

[16] Carlos Busso, Zhigang Deng, Michael Grimm, Ulrich Neumann, and Shrikanth Narayanan. Rigid head motion in expressive speech animation: Analysis and synthesis. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1075–1086, 2007.

[17] Soroosh Mariooryad and Carlos Busso. Generating human-like behaviors using joint, speech-driven models for conversational agents. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(8):2329–2340, 2012.

[18] Herwin Van Welbergen, Yu Ding, Kai Sattler, Catherine Pelachaud, and Stefan Kopp. Real-time visual prosody for interactive virtual agents. In *International Conference on Intelligent Virtual Agents*, pages 139–151. Springer, 2015.

[19] Catherine Pelachaud, Norman I Badler, and Mark Steedman. Generating facial expressions for speech. *Cognitive science*, 20(1):1–46, 1996.

[20] Carlos Busso and Shrikanth S Narayanan. Interrelation between speech and facial gestures in emotional utterances: a single subject study. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2331–2347, 2007.

[21] Carlos Busso, Zhigang Deng, Ulrich Neumann, and Shrikanth Narayanan. Learning expressive human-like head motion sequences from speech. In *Data-Driven 3D Facial Animation*, pages 113–131. Springer, 2008.

[22] Zhaojun Yang, Angeliki Metallinou, Engin Erzin, and Shrikanth Narayanan. Analysis of interaction attitudes using data-driven hand gesture phrases. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 699–703. IEEE, 2014.

[23] Justine Cassell, David McNeill, and Karl-Erik McCullough. Speech-gesture mismatches: Evidence for one underlying representation of linguistic and nonlinguistic information. *Pragmatics & cognition*, 7(1):1–34, 1999.

[24] Rob Voigt, Robert J Podesva, and Dan Jurafsky. Speaker movement correlates with prosodic indicators of engagement. In *Speech Prosody*, volume 7, 2014.

[25] Adam Kendon. Gesticulation and speech: Two aspects of the process of utterance. *The relationship of verbal and nonverbal communication*, 25(1980):207–227, 1980.

[26] Daniel P Loehr. Temporal, structural, and pragmatic synchrony between intonation and gesture. *Laboratory Phonology*, 3(1):71–89, 2012.

[27] Masahiro Mori. The uncanny valley. *Energy*, 7(4):33–35, 1970.

[28] Cathy Ennis, Rachel McDonnell, and Carol O'Sullivan. Seeing is believing: body motion dominates in multisensory conversations. In *ACM Transactions on Graphics (TOG)*, volume 29, page 91. ACM, 2010.

[29] Najmeh Sadoughi, Yang Liu, and Carlos Busso. Meaningful head movements driven by emotional synthetic speech. *Speech Communication*, 95:87–99, 2017.

[30] Kathrin Haag and Hiroshi Shimodaira. Bidirectional lstm networks employing stacked bottleneck features for expressive speech-driven head motion synthesis. In *International Conference on Intelligent Virtual Agents*, pages 198–207. Springer, 2016.

[31] Elif Bozkurt, Yücel Yemez, and Engin Erzin. Multimodal analysis of speech and arm motion for prosody-driven synthesis of beat gestures. *Speech Communication*, 85:29–42, 2016.

[32] Sergey Levine, Philipp Krähenbühl, Sebastian Thrun, and Vladlen Koltun. Gesture controllers. In *ACM Transactions on Graphics (TOG)*, volume 29, page 124. ACM, 2010.

[33] W. Standaert, S. Muylle, and A. Basu. Assessing the effectiveness of telepresence for business meetings. *2014 47th Hawaii International Conference on System Sciences*, 0:549–558, 2013.

[34] F. Tanaka, T. Takahashi, S. Matsuzoe, N. Tazawa, and M. Morita. Telepresence robot helps children in communicating with teachers who speak a different language. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI '14, pages 399–406, 2014.

[35] Y.-S. Chen, J.-M. Lu, and Y.-L. Hsu. Design and evaluation of a telepresence robot for interpersonal communication with older adults. In *Proceedings of the 11th International Conference on Smart Homes and Health Telematics, ICOST 2013, Singapore*, pages 298–303. Springer, 2013.

[36] Justine Cassell, Timothy Bickmore, Mark Billinghurst, Lee Campbell, K Chang, Hannes Hgni Vilhjlmsson, and Hao Yan. Embodiment in conversational interfaces: Rea. In *SIGCHI*, pages 520–527, 01 1999.

[37] Doug DeCarlo, Matthew Stone, Corey Revilla, and Jennifer J Venditti. Specifying and animating facial signals for discourse in embodied conversational agents. *Computer animation and virtual worlds*, 15(1):27–38, 2004.

[38] Adso Fernández-Baena, Raúl Montaño, Marc Antonijoan, Arturo Roversi, David Miralles, and Francesc Alías. Gesture synthesis adapted to speech emphasis. *Speech Communication*, 57:331–350, 2014.

[39] Sergey Levine, Christian Theobalt, and Vladlen Koltun. Real-time prosody-driven synthesis of body language. In *ACM Transactions on Graphics (TOG)*, volume 28, page 172. ACM, 2009.

[40] Yu Ding, Catherine Pelachaud, and Thierry Artieres. Modeling multimodal behaviors from speech prosody. In *International Workshop on Intelligent Virtual Agents*, pages 217–228. Springer, 2013.

[41] Gregor Hofer and Hiroshi Shimodaira. Automatic head motion prediction from speech data. 2007.

[42] Binh H Le, Xiaohan Ma, and Zhigang Deng. Live speech driven head-and-eye motion generators. *IEEE transactions on visualization and computer graphics*, 18(11):1902–1914, 2012.

[43] Chuang Ding, Pengcheng Zhu, Lei Xie, Dongmei Jiang, and Zhong-Hua Fu. Speech-driven head motion synthesis using neural networks. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[44] Chuang Ding, Lei Xie, and Pengcheng Zhu. Head motion synthesis from speech using deep neural networks. *Multimedia Tools and Applications*, 74(22):9871–9888, 2015.

[45] Chung-Cheng Chiu and Stacy Marsella. How to train your avatar: A data driven approach to gesture generation. In *International Workshop on Intelligent Virtual Agents*, pages 127–140. Springer, 2011.

[46] David Greenwood, Stephen Laycock, and Iain Matthews. Predicting head pose in dyadic conversation. In *International Conference on Intelligent Virtual Agents*, pages 160–169. Springer, 2017.

[47] Najmeh Sadoughi and Carlos Busso. Speech-driven animation with meaningful behaviors. *arXiv preprint arXiv:1708.01640*, 2017.

[48] Hai X Pham, Samuel Cheung, and Vladimir Pavlovic. Speech-driven 3d facial animation with implicit emotional awareness: a deep learning approach. In *The 1st DALCOM workshop, CVPR*, 2017.

[49] Xinyu Lan, Xu Li, Yishuang Ning, Zhiyong Wu, Helen Meng, Jia Jia, and Lianhong Cai. Low level descriptors based dblstm bottleneck feature for speech driven talking avatar. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5550–5554. IEEE, 2016.

[50] Hai X Pham, Yuting Wang, and Vladimir Pavlovic. End-to-end learning for 3d facial animation from raw waveforms of speech. *arXiv preprint arXiv:1710.00920*, 2017.

[51] Najmeh Sadoughi and Carlos Busso. Retrieving target gestures toward speech driven animation with meaningful behaviors. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 115–122. ACM, 2015.

[52] Fumihide Tanaka, Kyosuke Isshiki, Fumiki Takahashi, Manabu Uekusa, Rumiko Sei, and Kaname Hayashi. Pepper learns together with children: Development of an educational application. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 270–275. IEEE, 2015.

[53] Vittorio Perera, Tiago Pereira, Jonathan Connell, and Manuela Veloso. Setting up pepper for autonomous navigation and personalized interaction with users. *arXiv preprint arXiv:1704.04797*, 2017.

[54] Iina Aaltonen, Anne Arvola, Päivi Heikkilä, and Hanna Lammi. Hello pepper, may i tickle you?: Children's and adults' responses to an entertainment robot at a shopping mall. In *HRI*, 2017.

[55] Steven B Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *Readings in speech recognition*, pages 65–74. Elsevier, 1990.

[56] Haytham Fayek. Speech processing for machine learning: Filter banks, mel-frequency cepstral coefficients (mfccs) and what's in-between. `http://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html`. [Online; accessed 15/05/2018].

[57] Mel frequency cepstral coefficient (mfcc) tutorial. `http://www.practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/`. [Online; accessed 15/05/2018].

[58] Henri J Nussbaumer. *Fast Fourier transform and convolution algorithms*, volume 2. Springer Science & Business Media, 2012.

[59] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*, 2016.

[60] Ralph Beebe Blackman and John W Tukey. The measurement of power spectra. 1958.

[61] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

[62] William Song and Jim Cai. End-to-end deep neural network for automatic speech recognition. 2015.

[63] Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic. End-to-end audiovisual speech recognition. *arXiv preprint arXiv:1802.06424*, 2018.

[64] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[66] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[67] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2017.

[68] Hakan Erdogan. Sequence labeling: Generative and discriminative approaches. In *Tutorial at International Conference on Machine Learning and Applications*, 2010.

[69] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[70] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 1998.

[71] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[72] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[73] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[74] Bruce Thompson. Canonical correlation analysis. *Encyclopedia of statistics in behavioral science*, 2005.

[75] Mattias P Heinrich, Bartłomiej W Papież, Julia A Schnabel, and Heinz Handels. Multispectral image registration based on local canonical correlation analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 202–209. Springer, 2014.

[76] Elif Bozkurt, Engin Erzin, and Yücel Yemez. Affect-expressive hand gestures synthesis and animation. In *Multimedia and Expo (ICME), 2015 IEEE International Conference on*, pages 1–6. IEEE, 2015.

[77] Robert M Krauss, Palmer Morrel-Samuels, and Christina Colasante. Do conversational hand gestures communicate? *Journal of personality and social psychology*, 61(5):743, 1991.

[78] Rubin Wang and Fanji Gu. *Advances in cognitive neurodynamics (II): proceedings of the second International Conference on Cognitive Neurodynamics-2009*. Springer Science & Business Media, 2011.

[79] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985.

[80] Reza Shadmehr and Steven P Wise. *The computational neurobiology of reaching and pointing: a foundation for motor learning*. MIT press, 2005.

[81] Murray R Barrick and Michael K Mount. The big five personality dimensions and job performance: a meta-analysis. *Personnel psychology*, 44(1):1–26, 1991.

[82] Beatrice Rammstedt and Oliver P John. Measuring personality in one minute or less: A 10-item short version of the big five inventory in english and german. *Journal of research in Personality*, 41(1):203–212, 2007.

[83] James Lyons. Python speech features. `https://github.com/jameslyons/python_speech_features`, 2013. [Online; accessed 04/05/2018].

[84] Jacques Hadamard. Sur les problemes aux derive espartielles et leur signification physique. *Bulletin of Princeton University*, 13:1–20, 1902.

[85] Geometry-simple: 3D geometry library for Python, howpublished=`https://github.com/sbliven/geometry-simple`, note = [Online; accessed 07/05/2018].

[86] Zhihong Zeng, Zhenqiu Zhang, Brian Pianfetti, Jilin Tu, and Thomas S Huang. Audio-visual affect recognition in activation-evaluation space. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 4–pp. IEEE, 2005.

[87] Eric Jones, Travis Oliphant, and Pearu Peterson. SciPy: open source scientific tools for Python. 2014. http://www.scipy.org/ [Online; accessed 10/01/2018].

[88] Priyanshu Agarwal, Samer Al Moubayed, Alexander Alspach, Joohyung Kim, Elizabeth J Carter, Jill Fain Lehman, and Katsu Yamane. Imitating human movement with teleoperated robotic head. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, pages 630–637. IEEE, 2016.

[89] Roger Bartlett. *Introduction to sports biomechanics: Analysing human movement patterns*. Routledge, 2007.

[90] David A Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[91] François Chollet et al. Keras. `https://github.com/keras-team/keras`, 2015.

[92] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[93] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, et al. Recent advances in deep learning for speech research at microsoft. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8604–8608. IEEE, 2013.

[94] Abdel-rahman Mohamed, Geoffrey Hinton, and Gerald Penn. Understanding how deep belief networks perform acoustic modelling. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4273–4276. IEEE, 2012.

[95] Hubert Pham. Pyaudio: Portaudio v19 Python bindings. `https://people.csail.mit.edu/hubert/pyaudio/`, 2006. [Online; accessed 09/05/2018].

[96] Jan Ondras, Oya Celiktutan, Evangelos Sariyanidi, and Hatice Gunes. Automatic replication of teleoperator head movements and facial expressions on a humanoid robot. In *Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE International Symposium on*, pages 745–750. IEEE, 2017.

[97] Paul Read and Mark-Paul Meyer. *Restoration of motion picture film*. Butterworth-Heinemann, 2000.

[98] Maarten Baert. SimpleScreenRecorder: screen recorder for Linux. `https://github.com/MaartenBaert/ssr`. [Online; accessed 22/05/2018].

[99] PyAutoGUI: Python module for programmatically controlling the mouse and keyboard. `https://pyautogui.readthedocs.io/en/latest/`. [Online; accessed 22/05/2018].

[100] Therese Jolliffe and Simon Baron-Cohen. The strange stories test: A replication with high-functioning adults with autism or asperger syndrome. *Journal of autism and developmental disorders*, 29(5):395–406, 1999.

[101] The MARY Text-to-Speech System. `http://mary.dfki.de/`. Version 5.2. [Online; accessed 17/05/2018].

[102] Margaret McRorie, Ian Sneddon, Etienne de Sevin, Elisabetta Bevacqua, and Catherine Pelachaud. A model of personality and emotional traits. In *International Workshop on Intelligent Virtual Agents*, pages 27–33. Springer, 2009.

[103] Victoria Fromkin, Robert Rodman, and Nina Hyams. *An introduction to language*. Cengage Learning, 2018.

[104] Roy W Roring, Franklin G Hines, and Neil Charness. Age differences in identifying words in synthetic speech. *Human factors*, 49(1):25–31, 2007.

[105] Stephen J Winters and David B Pisoni. Perception and comprehension of synthetic speech. *Research on spoken language processing report*, (26):95–138, 2004.

[106] Geoff Luck, Suvi Saarikallio, and Petri Toiviainen. Personality traits correlate with characteristics of music-induced movement. In *ESCOM 2009: 7th Triennial Conference of European Society for the Cognitive Sciences of Music*, 2009.

[107] Atef Ben-Youssef, Hiroshi Shimodaira, and David A Braude. Speech driven talking head from estimated articulatory features. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4573–4577. IEEE, 2014.

[108] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015.

[109] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[110] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*, 2016.

[111] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373, 2000.

[112] Gene Ball and Jack Breese. Emotion and personality in a conversational agent. *Embodied conversational agents*, pages 189–219, 2000.

[113] Gary Collier and Gary James Collier. *Emotional expression.* Psychology Press, 2014.

[114] Heeyoung Kim, Sonya S Kwak, and Myungsuk Kim. Personality design of sociable robots by control of gesture design factors. In *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, pages 494–499. IEEE, 2008.

[115] Paul Adam Bremner, Oya Celiktutan, and Hatice Gunes. Personality perception of robot avatar teleoperators in solo and dyadic tasks. *Frontiers in Robotics and AI*, 4:16, 2017.

[116] Eunil Park, Dallae Jin, and Angel P del Pobil. The law of attraction in human-robot interaction. *International Journal of Advanced Robotic Systems*, 9(2):35, 2012.

[117] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[118] Theodore Bluche, Christopher Kermorvant, and Jérôme Louradour. Where to apply dropout in recurrent neural networks for handwriting recognition? In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 681–685. IEEE, 2015.

Appendix A

Effect of dropout regularisation

Using the chosen audio feature set ${}_{26}\mathbf{\Omega}$ (Section 4.3.3) and the MLP-SI and LSTM-SI models (with optimised architectures from Section 4.3.3 and Section 4.3.4 respectively), I examined the influence of dropout regularisation (Section 3.3.3) on model performance. On the single-split validation set (specified in Table 4.4), I compared the models trained with dropout probabilities $p_d \in \{0, 0.25, 0.5\}$ applied to all input-layer and hidden-layer units. For the LSTM-SI model only the feed-forward dropout was used (i.e. the recurrent dropout was not set), as recommended by [117, 118]. The results for both models are summarised in Tables A.1-A.2 and Figure A.1. As we can see, for both models all the quantitative measures are very similar for each p_d. I thus further compared the predicted movements for each p_d by visual inspection of various joint angles. Figure A.2 shows the movements predicted by MLP-SI in terms of the left shoulder roll angle θ_3 for each p_d. We can clearly see that the dropout regularisation resulted in flat predicted motion trajectories which was also the case for other angles and for the LSTM-SI model. This suggests that the noise already present in the dataset, due to the *many-to-many mapping*[1] nature of the problem, provides sufficient regularisation to prevent from overfitting. Further regularisation by dropout thus rather deteriorates than improves the performance in the pose regression task. This investigation also confirms that the qualitative evaluation plays an important role in assessing the audio-to-motion-synthesis system performance.

Table A.1: Effect of dropout regularisation on performance of the MLP-SI model. Evaluated on single-split validation set.

Dropout probability p_d	Loss L	RMSE (°)	ΔLCCA	LCCA$_{\mathbf{\Theta}}$	$\Delta\tilde{J}$ (10^4rad^{2}s^{-5})
0	0.0181	13.78	0.020	0.9795	1.195
0.25	0.0180	13.46	0.001	0.9747	1.191
0.5	0.0182	13.62	0.042	0.9552	1.192

[1]Multiple different audio signals can be associated with the same motion sequence and vice-versa, multiple different motion sequences can be associated with the same audio signal.

Table A.2: Effect of dropout regularisation on performance of the LSTM-SI model. Evaluated on single-split validation set.

Dropout probability p_d	Loss L	RMSE (°)	ΔLCCA	$\text{LCCA}_{\boldsymbol{\Theta}}$	$\Delta\tilde{J}$ ($10^4\text{rad}^2\text{s}^{-5}$)
0	0.0181	13.63	0.041	0.9841	1.163
0.25	0.0181	13.79	0.044	0.9846	1.180
0.5	0.0180	13.77	0.042	0.9852	1.178

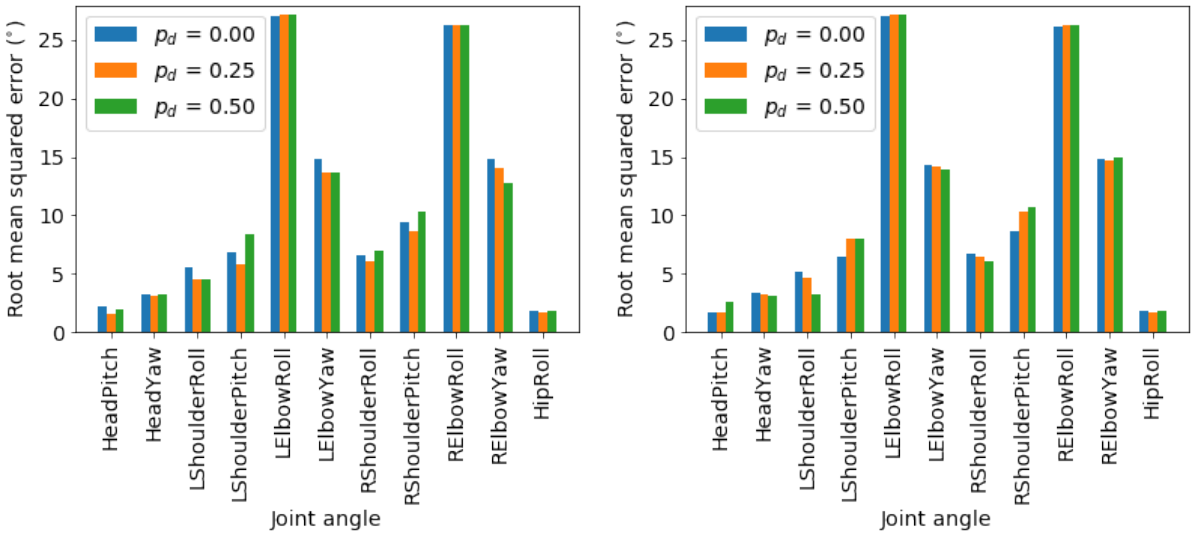


Figure A.1: Root mean squared error (RMSE) of each joint angle for various dropout probabilities p_d, for MLP-SI model (left) and LSTM-SI model (right). Evaluated on single-split validation set.

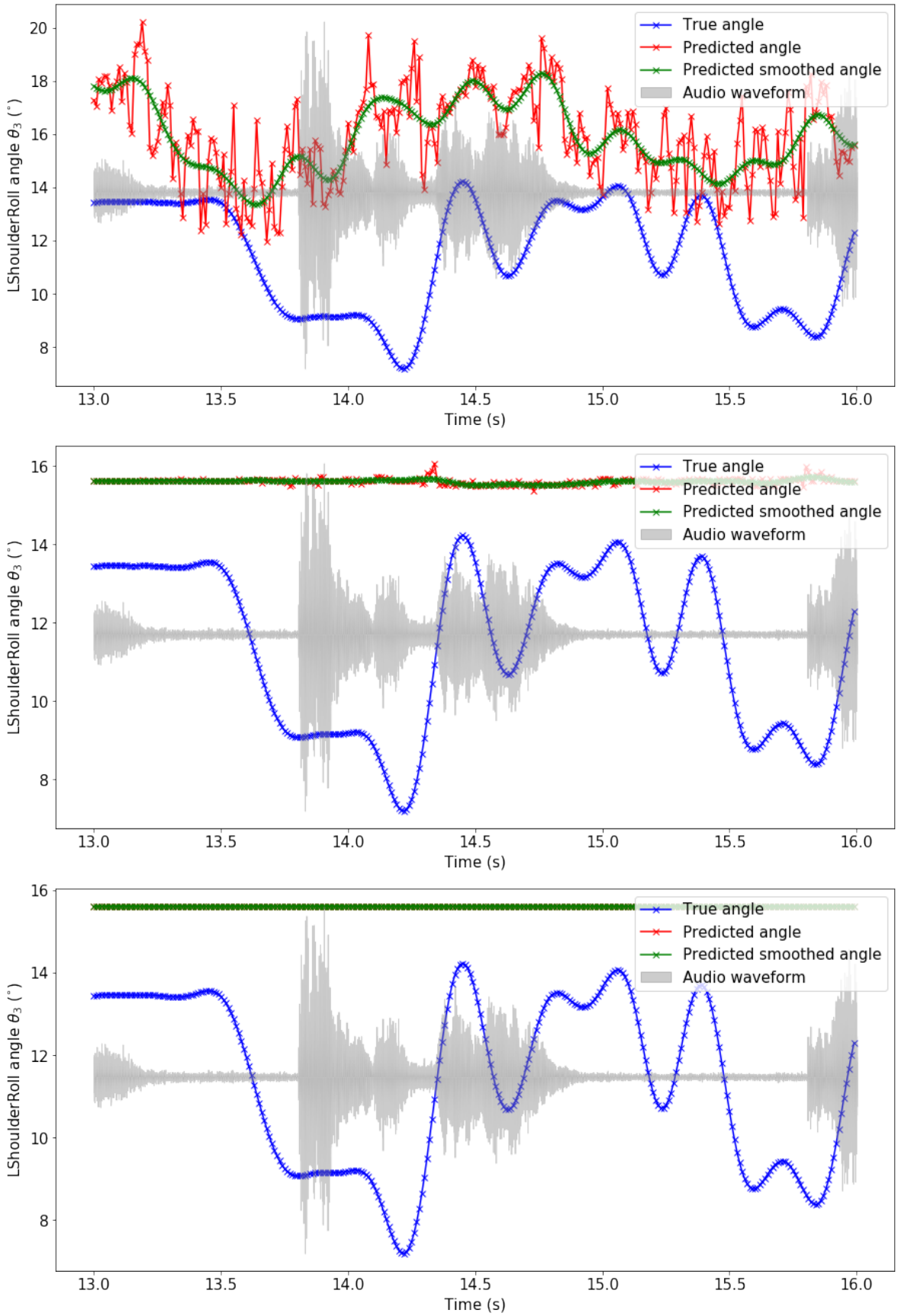


Figure A.2: Predicted left shoulder roll angle θ_3 for dropout probabilities $p_d = 0,\ 0.25,\ 0.5$ from top to bottom, using MLP-SI model.

Appendix B

Replies from authors of the 3D pose estimation methods

I asked the authors of the 2D→3D matching approach [3], LFTD [1] and VNect [4] whether their 3D pose estimation algorithms are robust to missing joints. I received the following replies.

- **2D→3D matching approach [3]**, reply from Ching-Hang Chen:
  "In this work, I didn't deal with partial body matching explicitly. "
- **LFTD [1]**, reply from Denis Tome:
  "The approach handles missing joints and tries to always reconstruct the 3D pose using only the joints that are visible. In that case the upper body should not be affected, whereas the lower body will be a mean pose coming from the dataset. "
- **VNect [4]**, reply from Dr Dushyant Mehta:
  "The method is not robust to missing joints, and needs the person to be in view completely. Some of our follow up work addresses that, but it is still under development. "

Appendix C

Web-surveys

C.1 Ethics Committee approval

Ethics Review #554
TITLE: Audio-driven upper-body motion synthesis on a humanoid robot
APPLICANTS: Jan Ondras
EMAIL: jo356@cam.ac.uk
DATES: 09/05/2018–28/05/2018
STUDY TYPE: survey methods
FUNDING BODY: –

DESCRIPTION

The proposed study consists of two anonymous online surveys in order to evaluate the developed automatic system that analyses the input audio of a person talking and generates corresponding upper-body movements on the humanoid robot Pepper. In both surveys the respondents will be asked to listen to a person's narration and at the same time they will be shown a side-by-side video of the Pepper robot generating the upper-body movements of the speaker. Each side of the video will show movements generated by different synthesis model. Consequently, for each side of the audio-visual recording, they will be asked to assess the appropriateness of the synthesised movements for the given audio on a Likert scale ranging from 1 = very inappropriate to 5 = very appropriate. In the first survey, the respondents will listen to natural speech and there will be 38 side-by-side video clips with durations around 10 seconds each. In the second survey, the respondents will listen to synthetic speech (artificially generated speech from an open-source virtual agent framework MaryTTS, `http://mary.dfki.de/`) and there will be 16 side-by-side video clips with durations around 30 seconds each. I aim to collect a minimum of 10 responses per survey.

PRECAUTIONS

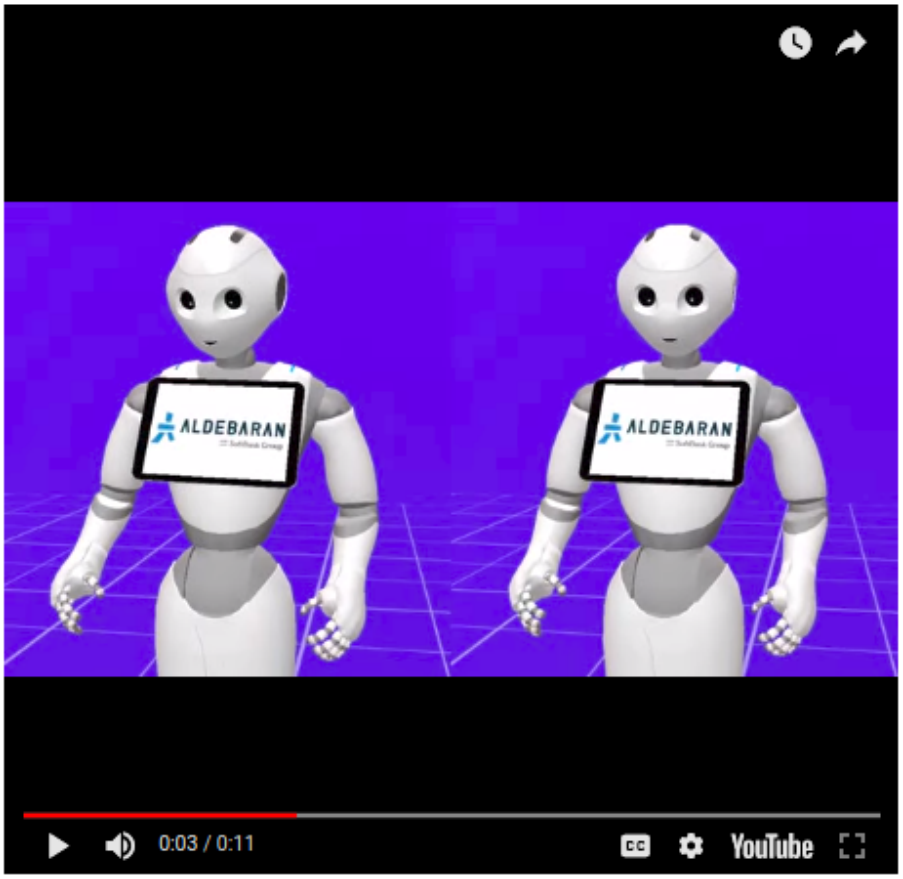
Consent:

All the audio recordings that will be played to the respondents in the first survey are anonymised and they were gathered in the previous study (Paul Bremner, Oya Celiktutan, and Hatice Gunes. Personality perception of robot avatar tele-operators. In *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*, pages 141-148. IEEE, 2016.) where the participants signed the consent form. Prior to taking the online surveys, the respondents will be given a description that explains the goal of the study, and for the survey with natural speech they will be also asked to declare a tick-box with the text: "I understand that I will be listening to recordings of private individuals. I promise that I will not copy or otherwise reuse these private recordings." Both surveys will be anonymous and no demographic data will be collected.

Participant Recruitment:

All the participants will be recruited online by forwarding the surveys to adult university students, staff, friends and relatives. All participation will be on voluntary basis.

C.2 Sample survey question



Please, compare the two sides (your left/right) of the video by rating how appropriate the generated movements are for the audio. *

	Very inappropriate	Inappropriate	Neutral	Appropriate	Very appropriate
Left side	○	○	○	○	○
Right side	○	○	○	○	○

Figure C.1: Sample web-survey question.

C.3 Texts for synthetic speech

Four stories chosen from the Strange Stories [100].

Banana

Katie and Emma are playing in the house. Emma picks up a banana from the fruit bowl and holds it up to her car. She says to Katie, "Look! This banana is a telephone!"

Picnic

Sarah and Tom are going on a picnic. It is Tom's idea, he says it is going to be a lovely sunny day for a picnic. But just as they are unpacking the food, it starts to rain, and soon they are both soaked to the skin. Sarah is cross. She says, "Oh yes, a lovely day for a picnic alright!"

Army

Two enemy powers have been at war for a very long time. Each army has won several battles, but now the outcome could go either way. The forces are equally matched. However, the Blue army is stronger than the Yellow army in foot soldiers and artillery. But the Yellow army is stronger than the Blue army in air power. On the day of the final battle, which will decide the outcome of the war, there is a heavy fog over the mountains where the fighting is about to occur. Low-lying clouds hang above the soldiers. By the end of the day the Blue army have won.

Glasses

Sarah is very long-sighted. She has only one pair of glasses, which she keeps losing. Today she has lost her glasses again and she needs to find them. She had them yesterday evening when she looked up the television programmes. She must have left them somewhere that she has been today. She asks Ted to find her glasses. She tells him that today she went to her regular early morning keep fit class, then to the post office, and last to the flower shop. Ted goes straight to the post office.