



TÉCNICO
LISBOA



Human-Robot greeting: A model based on social studies and Hidden Markov Models

Manuel Picão Fernandes Campos de Carvalho

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. Plinio Moreno López
Prof. José Alberto Rosado dos Santos Vitor

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira

Supervisor: Prof. Plinio Moreno López

Member of the Committee: Prof. Rodrigo Martins de Matos Ventura

January 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Firstly, I would like to thank my two thesis supervisors, Professor Plinio Moreno López and PhD student João Avelino. Both of them showed a 24/7 availability to solve my doubts and concerns during this period. Even with all the constraints of online working, I am really thankful for the challenge given, for all the knowledge they shared with me and for turning our weekly meetings productive, but also entertaining times.

I also want to thank my family, all my friends, and essentially everyone that showed interest in this work, by helping me, giving advice, or support.

To finish, I want to mention everyone else that made this thesis possible. From the VisLab researchers who managed to develop an amazing robot as Vizzy, to all the IST professors who taught me through these five years, my thank you.

Resumo

Robôs sociais móveis devem ser capazes de iniciar uma interação com pessoas de forma eficaz. No entanto, cumprimentar alguém é uma tarefa complexa. Adam Kendon criou um modelo para cumprimentos, composto por seis fases: *Initiation of Approach*, *Distance Salutation*, *Head Dip*, *Approach*, *Final Approach*, e *Close Salutation*. Estas podem ser bastante úteis para um robô social compreender uma pessoa que pretende cumprimentá-lo e agir devidamente.

Neste trabalho propõe-se um sistema para robôs sociais móveis que estima a fase do cumprimento usando um HMM (*Hidden Markov Model*), através de características observáveis, e replica-a com movimentos apropriados usando BTs (*Behavior Trees*).

Usaram-se *datasets* públicos para treinar o HMM através do algoritmo EM (*Expectation-Maximization*), extraíndo e classificando, para tal, as características observáveis do cumprimento necessárias. De seguida, testou-se a previsão de estados com um conjunto de teste, obtendo 80.9% dos estados corretos.

Para testar o sistema de cumprimento, usou-se um robô humanóide e móvel do Instituto de Sistemas e Robótica, o Vizzy. Conduziram-se experiências num simulador, prevendo corretamente cerca de 92% dos estados em tempo real, dadas várias situações de cumprimentos. Quando se juntaram as BTs à previsão dos estados, confirmou-se que cada uma das seis fases era devidamente replicada. Finalmente, foi possível reproduzir cumprimentos de forma natural, confirmando a aplicabilidade do sistema para HRI (*Human-Robot Interaction*).

Palavras-chave: robôs sociais, cumprimentos, Hidden Markov Model, Behavior Trees.

Abstract

Social mobile robots should be capable of effectively open interaction with people. However, greeting someone is a complex task. Adam Kendon modeled greetings as a set of six phases: *Initiation of Approach*, *Distance Salutation*, *Head Dip*, *Approach*, *Final Approach*, and *Close Salutation*. These are valuable for a social robot to infer people's greeting intentions and comply with them.

This work proposes a system for mobile social robots that estimates the greeting phase through an HMM (*Hidden Markov Model*) by extracting observable features, and follows it with the appropriate behaviors using BTs (*Behavior Trees*).

We used publicly available datasets to train the HMM, through the EM (*Expectation-Maximization*) algorithm, extracting and labeling the necessary observable greeting features. Later, we tested the state estimation with sequences from the same datasets, and obtained an average accuracy of 80.9%.

To test the system we used a humanoid robot from the Institute for Systems and Robotics, Vizzy. We conducted experiments on a simulator, obtaining an accuracy around 92% while predicting states seen by the robot, in different greeting situations. When connecting BTs to the state prediction, we confirmed that every state was properly replicated and natural greetings were achieved, confirming the system's applicability for HRI (*Human-Robot Interaction*).

Keywords: social robots, greetings, Hidden Markov Model, Behavior Trees

Contents

Acknowledgments	vii
Resumo	ix
Abstract	xi
List of Tables	xv
List of Figures	xvii
Nomenclature	xix
Glossary	xix
1 Introduction	1
1.1 Motivation	1
1.2 Topic Overview	2
1.3 Objectives	2
1.4 Thesis Outline	3
2 Background and State of the Art	5
2.1 Adam Kendon's Greeting Model	5
2.1.1 Initiation of Approach	5
2.1.2 Distance Salutation	6
2.1.3 Head Dip	6
2.1.4 Approach	7
2.1.5 Final Approach	7
2.1.6 Close Salutation	8
2.2 Hidden Markov Models	8
2.3 Behavior Trees	12
2.4 State of the art	14
3 Greeting Model using a Hidden Markov Model	21
3.1 Global Overview of the model	21
3.2 Robot Operating System	22
3.3 HMM Observations	23
3.3.1 Distance	24
3.3.2 Speed	25

3.3.3	Gaze	26
3.3.4	Smile	30
3.3.5	Movements	31
3.4	Kendon Model as a Hidden Markov Model	32
3.5	Training a Hidden Markov Model with the EM algorithm	37
3.5.1	Extracting video information	38
3.5.2	Limitations of the videos	42
3.5.3	Training and Test sets	42
3.5.4	Comparison with Kendon Model	44
3.6	Testing the model	45
3.6.1	Using Test Sequences	45
3.6.2	Vizzy	49
3.6.3	Simulator	50
3.6.4	Testing on Real-Time Situations	51
4	Greeting Model using Behavior Trees	55
4.1	Global Overview	55
4.1.1	Global Behavior Tree	55
4.1.2	Behavior Tree of each phase	56
4.2	Behavior Tree Testing	63
5	Conclusions	67
5.1	Achievements	67
5.2	Future Work	68
	Bibliography	69

List of Tables

2.1	Greetings in Kendon's birthday party	5
2.2	Flow control nodes on Behavior Trees	14
2.3	Common characteristics of <i>Initiation of Approach</i> phase developed on other projects	16
2.4	Common characteristics of <i>Distance Salutation</i> phase developed on other projects	16
2.5	Common characteristics of <i>Head Dip</i> phase developed on other projects	17
2.6	Common characteristics of <i>Approach</i> phase developed on other projects	17
2.7	Common characteristics of <i>Final Approach</i> phase developed on other projects	18
2.8	Common characteristics of <i>Close Salutation</i> phase developed on other projects	19
3.1	Comparison of the observations' extraction approaches for the 2 Datasets	42
3.2	Summary of greetings' phases in the used Datasets	43
3.3	Accuracy of the two models on the test set	47
3.4	Confusion Matrix of the chosen model on the test set	47
3.5	Accuracy of the two models on 15 different train and test sets	48
3.6	Confusion Matrix of one low-accuracy model	49
3.7	States of the model present (Y) or not present (N) for each experiment	52
3.8	Accuracy of the model on the different sequences	54

List of Figures

2.1	Phases of greeting: a) Sighting (<i>Initiation of Approach</i>); b) <i>Distance Salutation</i> ; c) <i>Head Dip</i>	7
2.2	Behavior Tree with a task consisting of finding a target person, turning to him/her, and waving	12
3.1	Global diagram of the greeting model implemented as a Hidden Markov Model	22
3.2	Representation of 4 greeting features which were chosen as HMM observations: distance, speed, gaze and movements	24
3.3	Representation of the base_footprint (x,y,z) and world (X,Y,Z) coordinate frames	25
3.4	Example of the speed calculation for every observation on the HMM	26
3.5	Output of OpenFace: left image represents face landmarks, face orientation and gaze direction; right image contains the list of detected AUs and their intensities	27
3.6	Face landmarks labeling for OpenFace	28
3.7	Gaze Detector 1 situation: the eye gaze direction forms a cone around the robot, considering there is a direct gaze	29
3.8	Gaze Detector 2 situation: the head orientation direction forms a cone which does not include the center of the robot's head, considering there is not gaze	30
3.9	Action Units visual example and description	31
3.10	Sample of the environment in AVDIAR dataset's greetings	38
3.11	Sample of UoL's Dataset environment for greetings	41
3.12	First 4 test sequences predicted with the Viterbi algorithm	48
3.13	Last 4 test sequences predicted with the Viterbi algorithm	49
3.14	Left: Vizzy waving; Right: Vizzy's size comparing with a 1.75 m person	50
3.15	Map of the simulation environment in Gazebo	51
4.1	Global diagram of the model with the addition of the Behavior Tree branch (dotted)	56
4.2	Global Behavior Tree of the model	57
4.3	Behavior Tree of the <i>Initiation of Approach</i> phase	58
4.4	Example of the action of turning to a target individual	59
4.5	Behavior Tree of the <i>Distance Salutation</i> phase	60
4.6	Behavior Tree of the <i>Head Dip</i> phase	60
4.7	Behavior Tree of the <i>Approach</i> phase	61

4.8	Robot approach movement. Left: setting target position; Right: setting target orientation .	62
4.9	Behavior Tree of the <i>Final Approach</i> phase	63
4.10	Behavior Tree of the <i>Close Salutation</i> phase	64
4.11	Simulation: Initial positions	65
4.12	Simulation: Approaching movements	65
4.13	Simulation: <i>Close Salutation</i>	65

Acronyms

AI Artificial Intelligence.

APP Approach.

AU Action Unit.

BT Behavior Tree.

CA Control Architecture.

CS Close Salutation.

CV Computer Vision.

DS Distance Salutation.

FA Final Approach.

FACS Facial Action Coding System.

FSM Finite-State Machine.

HD Head Dip.

HMM Hidden Markov Model.

HRI Human-Robot Interaction.

IA Initiation of Approach.

ISR Institute for Systems and Robotics.

IST Instituto Superior Técnico.

ROS Robot Operating System.

YARP Yet Another Robot Platform.

Chapter 1

Introduction

1.1 Motivation

Although most of the time people do not realize, at the beginning of every interaction between humans, the two parties tend to follow a greeting ritual. This ritual is composed of several steps, starting on the moment people sight each other, finishing, generally, with a salutation.

Usually, greeting is a difficult and unnatural task. Even humans who have done it for years sometimes struggle to understand if the other person is expecting us to greet, if we are supposed to greet using kisses, handshakes, or no physical contact at all.

This struggle is due to greeting being a set of actions that can be very different according to the social relationship of the two parties. Culturally, there are also plenty of different approaches, and these can even change with education, for example. Unexpected behaviors can be, for instance, using cheek kisses if the person is expecting just a handshake, or opting for only one kiss while the other is expecting two - a very common situation in Portugal. These situations, in the act of greeting, can lead to an awkward or uncomfortable situation, which is not a positive start to a conversation.

If this process is difficult in human interactions, we should expect it to be even harder to model and implement in robots. However, it has as much of hardness as of importance.

Social robots have been used more and more in the last few years. Technology companies have been developing robots that may already serve in several job positions where they need to interact with people. This is the example of receptionists Pepper [1], Nadine [2], and Olivia [3], or companion robots S.A.M [4] and Buddy [5].

The greeting action emerges with major importance for social robots, as it is what triggers every Human-Robot Interaction (HRI). Performing proper and natural greetings can be the beginning of a well-succeeded social robot. On the other hand, unnatural and strange greeting behaviors may lead to people being uncomfortable around them.

In this work we will use a greeting model based on human greeting [6] and aim to implement it in HRI. For some of the experiments, we will use a humanoid robot named Vizzy [7], which belongs to Institute for Systems and Robotics (ISR), an affiliation of Instituto Superior Técnico (IST).

1.2 Topic Overview

Focusing on the western and, in particular, American ways of greeting, Kendon [6] created a model that people tend to follow when meeting others, through observation of several greetings at a birthday party. This model consists of six distinct phases, which may not always happen, nor necessarily by this order: i) Initiation of Approach (IA), which contains the sighting of a person, the decision of greeting, and the preparation for the approach; ii) Distance Salutation (DS), where people display a long-distance salutation, which may be subtle (tossing of the head, for example) or not (waving, for example); iii) Head Dip (HD), one head lower movement which commonly follows the DS phase; iv) Approach (APP), which is most of the movement (usually walking) toward the other person, starting at the first approaching moment and ending when the greeters begin to prepare for the final salutation; v) Final Approach (FA), which follows the APP phase and finishes the approach movement, also being characterized by several changes in the person's behavior, considered as the preparation for the close interaction. vi) Close Salutation (CS), usually the last phase of a greeting. It contains a salutation, which can have various forms, from non-contact to contact actions.

In order to provide a robot an estimation of these six phases, we will use a Hidden Markov Model (HMM) [8]. An HMM is one of many statistical models used to represent systems defined by a series of states, which in our case are the six phases of greeting. The particularity of this model is that the states are not entirely observable (they are called hidden states). Instead, there is another set of observable events (observations) that depend on these hidden states and allow the model to predict their evolution. In our greeting model, these observations can be seen as noticeable features that characterize some of the phases, and allow the model to identify them. For the prediction of these states, an HMM bases its ideas on the Markov property, which assumes that the probabilities for the next state depend entirely on the present one.

To control the robot's reactions to a phase estimated by the HMM, our robot will use a Behavior Tree (BT) [9]. BTs consist of flexible sequences of tasks (actions or movements, for example). A BT can be divided into several smaller trees, which allows us to create several specific sequences of tasks, where each one would be the robot's reaction for each greeting phase. This flexibility and the facility on the visualization and changing are some of the qualities that make BTs highly used in Artificial Intelligence (AI) and Robotics, comparing to other traditional approaches, such as the Finite-State Machines (FSM) [10].

1.3 Objectives

The main objective of this thesis is to contribute to the HRI skills of social mobile robots, by constructing a system that estimates the phases identified by Kendon using social signals of a target person's behavior, and responds with the appropriate actions.

This objective can be divided into two sub-objectives to achieve:

- Prediction of the greeting phases from a person that intends to greet, designing and creating an

HMM based on public greeting sequences obtained from external datasets. The model should estimate its parameters from sets of observations extracted and be able to predict phases given similar observations.

- Implementation of a robot's reactions to state prediction. For this, we intend to apply the HMM on the robot, extracting the necessary observations through Computer Vision techniques. The robot would then select the most adequate actions and execute them using BTs.

1.4 Thesis Outline

This dissertation is composed of five main chapters, as described below.

Chapter 2 is divided into two parts: the background and the State of the Art. The background includes an explanation about some fundamental topics for this dissertation and whose knowledge about a few key features is essential for understanding this work. Background topics include Adam Kendon's greeting model, interpersonal distancing in conversations, Hidden Markov Models (HMM), and the Robot Operating System (ROS). In the State of the Art section, we summarize all previous works related to our objectives within this dissertation. These contributed to establish our approach, detect possible obstacles, and define our objectives. On the related works, we included projects with social robots that had the ability to detect parts of greeting and/or to perform some phases of the greeting model.

On chapter 3 we describe the design of our HMM and the parameter estimation. We talk about the hidden states and the observations used, we bring details to how we trained an HMM based on real greeting data and we compare it to a reliable model based on the greeting model from Kendon. Then, we present the tests and results on different metrics and within different situations, using a simulator of our humanoid robot, Vizzy.

Chapter 4 contains the implementation of the robot's reactions to the HMM state estimation, using BTs. We provide a general explanation of BTs and how we implemented all the greeting phases on Vizzy using them. We later give brief information about Vizzy and its skills. Finishing, we grant information about testing approaches and results on the usage of BTs complementing the HMM.

Chapter 5 includes all achievements, contributions and conclusions accomplished with this dissertation. It also identifies all future effort which could be made and would, by some means, bring enhancements to this project.

Chapter 2

Background and State of the Art

2.1 Adam Kendon's Greeting Model

Kendon [6] filmed an entire family birthday party in New York, in order to take detailed notes about greetings between humans. He observed 63 greetings, including people with different social relationships, people from several ages and well divided in gender, as it is shown in Table 2.1. Based on the observation of these videotapes, he constructed a global greeting model, which people would normally follow when they meet others. According to this model, one greeting may consist of up to six phases, as follows.

Total number of greetings	63
Host - Guest	31
Guest - Guest	32
Adult - Adult	56
Adult - Child	7
Male - Male	18
Male - Female	26
Female - Female	19
Members of the same family	20
Close friends	15
Acquaintances	28

Table 2.1: Greetings in Kendon's birthday party

2.1.1 Initiation of Approach

A greeting interaction is initiated, according to Adam Kendon, at the point where the person turns toward the other, after sighting him or her. For this, the person in question needs to sight another individual and decide to initiate a greeting interaction. This decision may depend on some factors, such as the urgency of greeting and if any of the interactors is busy at the sighting time.

The *Initiation of Approach* (IA) phase is, then, the starting point of greeting, ending when both parties

have indicated, through mutual gaze, that they have observed each other and are ready to begin their approach to greet the other person.

Before this gaze, people orient only their head at the other while, sometimes, avoid looking directly for a few moments, until they are certain that the intention of greeting is reciprocal. To be rebuffed or unrecognized can be very embarrassing and people rarely risk it. When this risk disappears, the mutual gaze tends to begin and both greeters orient their entire body to the target they want to greet, preparing to begin their approach. Naturally, the most common phase to follow this is the *Approach*, however, Kendon described some situations in which a *Distance Salutation* follows the sighting almost directly.

2.1.2 Distance Salutation

The *Distance Salutation* (DS) phase occurs in every greeting analyzed by Kendon that also has close interaction. At one point during their approach to the target, people use one of several different forms of an explicit display, performing a long-distance salutation. The most common display observed by Kendon was a head toss movement, in which people suddenly tilt the head back and bring it forward after it, usually combining it with a verbal greeting, such as "hi". Another instance is to tilt the head forward, hold it and then raise it again, what is called a head lower movement and it is commonly seen as a response to a head toss. Other typical *Distance Salutation* displays are the nod (usually combined with verbal greetings), and the waving movement.

The distance at which this display occurs can vary due to the greeting urgency and environmental conditions. In fact, Kendon states examples of greeters who perform this display just after orienting to the target; and also examples in which a DS movement is seen in the later stages of the approach, being followed by a *Close Salutation* almost directly. Independently of the distance and the display performed, a *Distance Salutation* movement commonly contains a direct look from both greeters and smiles.

Greeting actions may end in this phase, if both people agree it is not necessary to have further interaction. If it does not, the *Approach* is the most likely following phase, even though a *Head Dip* may happen in between.

2.1.3 Head Dip

Following the *Distance Salutation*, there is, many times, one head lower movement by a forward bend of the neck, which Kendon calls the *Head Dip* (HD). This movement was clearly detected in 50% of the DS phases observed in Kendon's birthday party videos, though, usually, only one of the greeters did it.

The HD phase was mostly seen by Kendon in DSs which had occurred just after the person started an approach toward the target, while no DS in the final moments of the approach was followed by this movement. Kendon's conclusion is that this phase can be a way of beginning to fully attend to the other person, ending any other involvements, which justifies why it was found mostly in the first moments of the greeting approach.

In Figure 2.1 we can see a sketch, present in [6], representing an IA phase, by the sighting of a target person (a), followed by a head toss display, which is an example of a DS (b) and the described

HD movement (c).

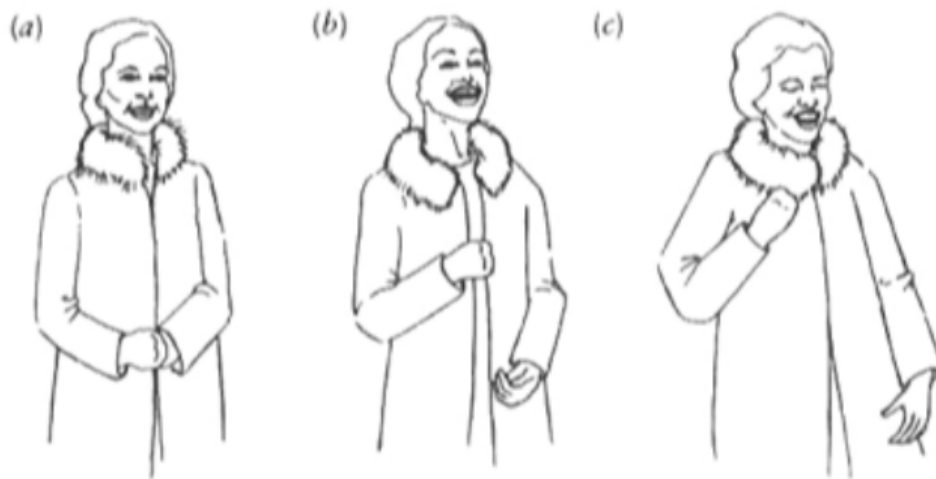


Figure 2.1: Phases of greeting: a) Sighting (*Initiation of Approach*); b) *Distance Salutation*; c) *Head Dip*

2.1.4 Approach

The first part of the movement of both parties to a position close to the other is considered the *Approach* (APP) phase. During the APP, it is usual that at least one person of the pair stops gazing at the other, looking away until he or she is very close to the other. In fact, the person who moves the most in the approach (seen as "entering the other's territory") seems to be more likely to look away.

Other usual behaviors in this phase are to bring one or both arms in front, crossing the upper part of the body, what is called "body cross", and grooming, with people adjusting their clothes, hair, or accessories.

In the videotapes, Kendon observed that the party's host usually moved less than the guest arriving, however, this difference tended to soften in important arrivals. In the referred birthday party, people who did not gaze at the other, made the "body cross" or walked bigger distances were usually people "entering the other's territory", for instance, guests, children, or people with less confidence with the host. These actions are also related to non-aggressive behavior.

At the end of the *Approach*, if no unusual situation occur, people start to prepare for an interaction, entering into the *Final Approach* phase.

2.1.5 Final Approach

The *Final Approach* (FA) is the last part of the approach movement, usually starting when the pair is less than 3 meters away. It is mainly characterized by a re-establishment of mutual gaze and smiles, as they generally both fade or disappear during the APP phase. There can also be some verbal greeting.

There is usually a change of the head position, comparing to the way they have been holding it during the approach. In the FA phase, five different positions of the head were observed by Kendon in the videotapes, where tilting the head forward was the most frequent movement.

Sometimes, people orient their palms to the other and extend their arms, known as the "palm presentation", in association with a waving movement or beginning the *Close Salutation*, which is, naturally, the following phase.

2.1.6 Close Salutation

The *Close Salutation* (CS) is the final phase of greeting. The participants stop in front of each other, looking directly at the other, and start the interaction after saluting the other, with a salutation movement.

Salutations in this phase can have a lot of forms, depending on local cultures and social relationships. The physical contact is frequent but optional, as some instances do not include it. In fact, Kendon divides the salutations observed in the birthday party into *with body contact* and *without body contact*.

Without body contact salutations in the birthday party normally consisted of the greeters coming to a halt in front of each other and maintaining the eye and head orientation they had a few meters away. There were also instances of this phase with head movements, such as head toss movements, nods, or bows, similar to what occurs in the DS phase.

With body contact salutations on this phase included handshakes (usually composed of three up-down movements), cheek kiss or kisses, embracing, and others.

Kendon also acknowledged that in female-female pairs, no contact greeting was the most familiar; in male-female pairs, the embracing; and in male-male, the handshake.

After the greeting, both participants change their positions and orientations, even if they continue to interact with each other.

2.2 Hidden Markov Models

Kendon's greeting model will be implemented in our work as a Hidden Markov Model (HMM), as it was introduced before. This type of model was mainly chosen because it deals with systems containing several phases which are not directly identified, but, instead, can be understood through a series of characteristics. Our implemented greeting model happens to be in this situation, since each phase of the model is estimated entirely through the observable behavior of the greeter.

An HMM is a probabilistic model created to represent stochastic systems and was first introduced by Baum and Petrie [11]. Since then, it has had several usages, including in the areas of speech recognition [12], bioinformatics [13], and finances [14].

The HMM is based on an augmented version of a Markov Chain. The Markov Chain [15] is a model that represents a set of N possible variables - states - and the transitions between them, including the respective probabilities of these transitions. Using a Markov Chain, we can predict the state that follows a given sequence of states. For this, the model strongly assumes that the current state is the only impacting this predicting calculation. This assumption is called Markov property and is satisfied in all Markov processes, in which the Markov Chains and the HMM are included.

More formally, the Markov property consists on (2.1), considering a set of possible states, M and a

sequence of states, q_1, \dots, q_n .

$$P(q_{n+1} \in M | q_n, \dots, q_1) = P(q_{n+1} \in M | q_n) \quad (2.1)$$

In an HMM there is also a sequence of observable events that are causal factors of our model. However, in this case, the events that interest us are, regularly, not observable. They are called hidden states. These states are predicted through the observable events and the previous hidden state, as in Markov Chains.

An HMM is defined by the following components:

- A set of N possible hidden states, s_1, s_2, \dots, s_N ;
- A sequence of V observations, o_1, o_2, \dots, o_V , where each belongs to a set of M possible observations.
- A transition probability matrix $\mathbf{A}_{(N \times N)}$ (with the matrix's dimensions below the variable), and where $a_{i,j}$ is the probability of moving from state i to state j , i.e., $P(s_j | s_i)$.
- An emission probability matrix $\mathbf{B}_{(N \times M)}$, where field $b_i(o_t)$ is the probability of having observation t in state i , $P(o_t | s_i)$;
- An initial probability distribution vector $\boldsymbol{\pi}_{(N)}$, where π_i is the probability that the HMM will start in state i .

Given an HMM, $\lambda = (\mathbf{A}, \mathbf{B})$, the present observation o_t and the previous state, s_i , the present state s is given by (2.2). In a situation where s is the first state of a sequence, (2.3) applies.

$$s = \operatorname{argmax}_j a_{ij} b_j(o_t) \quad (2.2)$$

$$s = \operatorname{argmax}_j \pi_j b_j(o_t) \quad (2.3)$$

In this thesis, we assume that our observations follow a Gaussian distribution, which leads to using a more specific type of HMM, a Gaussian Hidden Markov Model [16].

In these models, it is assumed that each possible observation $f_1, f_2, \dots, f_M \in F$ exists and has a specified value which follows its own Gaussian distribution. Thus, each observation o_i is a vector with length M , with the values of the M observations .

Also, the emission probability matrix \mathbf{B} is split into:

- A matrix $\mathbf{M}_{N \times M}$, where m_{ij} is the mean value of observation j , if the model is on state i .
- A 3-dimensional array $\mathbf{C}_{N \times M \times M}$, containing a covariance matrix with size $M \times M$ for each of the N states.

HMMs can be used to solve three fundamental problems in sequential data, first introduced by Rabiner [17].

Likelihood Problem The first of these problems is to compute the likelihood of a particular observation sequence. Given a model $\lambda = (\mathbf{A}, \mathbf{B})$, and an observation sequence \mathbf{o} , the objective is to determine $P(\mathbf{o}|\lambda)$. For solving this problem, the forward algorithm [8] is commonly used.

The forward algorithm is a type of dynamic programming algorithm, that is, an algorithm that uses an array variable to store intermediate values as it calculates the probability of the observation sequence.

Each intermediate value of the algorithm, $\alpha_t(j)$, represents the probability of being in state j after seeing the first t observations, given the model λ . The value of each cell $\alpha_t(j)$ is computed by summing over the probabilities of every path that could lead to that cell. Given t observations, the probabilities for each state are calculated iteratively through (2.5), where:

- α_{t-1} is the forward probability array from the previous observation, where $\alpha_{t-1}(i) = P(o_1, o_2, \dots, o_{t-1}, q_{t-1} = i | \lambda)$
- a_{ij} is the transition probability from state i to state j ;
- $b_j(o_t)$ is the observation likelihood from state j , given the observation t .

The goal probability $P(\mathbf{o}|\lambda)$ is then calculated in (2.6), by summing over the probabilities of all possible hidden state paths that could generate the given observation sequence.

We may summarize this algorithm by the three following steps:

1. Initialization

$$\alpha_1(j) = \pi_j b_j(o_1), 1 \leq j \leq N \quad (2.4)$$

2. Recursion

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t), 1 \leq j \leq N, 1 < t \leq T \quad (2.5)$$

3. Termination

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (2.6)$$

Decoding Problem The second problem is the Decoding Problem. In a given model $\lambda = (\mathbf{A}, \mathbf{B})$, solving this consists of determining the most probable sequence of states $Q = q_1, q_2, \dots, q_T$ that corresponds to a sequence of observations $\mathbf{o} = o_1, o_2, \dots, o_T$. This problem is usually solved by the Viterbi Algorithm [18], which is another dynamic programming algorithm, saving a variable from iteration to iteration.

In this algorithm, each variable, $v_t(j)$, represents the probability that the HMM is in state j after seeing the first t observations and passing through the most probable state sequence q_1, \dots, q_{t-1} , given a model λ . Instead of summing all the paths that could lead to this situation, such as in the forward algorithm, the value of $v_t(j)$ is here computed by recursively taking the most probable path that could lead to this cell, which is formally represented in (2.9). The factors being multiplied on this step are:

- v_{t-1} , the Viterbi path probability variable from the previous observation, where $v_{t-1}(i) = \max_{q_1, \dots, q_{t-2}} P(q_1, \dots, q_{t-2}, o_1, o_2, \dots, o_{t-1}, q_{t-1} = i | \lambda)$
- a_{ij} , the transition probability from state i to state j ;

- $b_j(o_t)$, the observation likelihood from state j , given observation o_t .

As opposed to the forward algorithm, the Viterbi algorithm has another component to be computed in every iteration, since it must compute the most probable state sequence. We compute this state sequence by keeping track of the path of the hidden state that led to each situation, using the bt matrix in (2.10). In the end of the recursive process, the algorithm traces the best path to the beginning, in what is called the Viterbi backtrace. The backtrace starts with (2.12) and, using the previously saved bt , returns the state sequence that best applies to the given observations.

1. Initialization

$$v_1(j) = \pi_j b_j(o_1), 1 \leq j \leq N \quad (2.7)$$

$$bt_1(j) = 0, 1 \leq j \leq N \quad (2.8)$$

2. Recursion

$$v_t(j) = \max_i v_{t-1}(i) a_{ij} b_j(o_t), 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq t \leq T \quad (2.9)$$

$$bt_t(j) = \operatorname{argmax}_i v_{t-1}(i) a_{ij} b_j(o_t), 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq t \leq T \quad (2.10)$$

3. Termination

$$P^* = \max_i v_T(i), 1 \leq i \leq N \quad (2.11)$$

$$q_T^* = \operatorname{argmax}_i v_T(i), 1 \leq i \leq N \quad (2.12)$$

Learning Problem The third problem is the Learning Problem, which is applied to models whose A and B parameters are unknown. Given a sequence of observations o and the set of possible states in the HMM, the model should learn the A and B matrices. There is more than one possible algorithm to solve this problem, however, to learn the HMM parameters for this work, we will use the Expectation-Maximization (EM) algorithm [8].

This algorithm starts with an initial estimation of parameters A and B . Then, there are two iteratively ran steps. In the Expectation-step (E-step), there is a computation of the expected state occupancy for every state j at every time t , $\gamma_t(j)$, through (2.13). Here, $\alpha_t(j)$ is the forward probability for this state and time (just as in the forward algorithm), while $\beta_t(j)$ is called the backward probability, which is the probability of seeing the observations from time $t+1$ to the end, given that we are in state j at time t . The field $\alpha_T(q_F)$ is the forward probability for the final state at the final observation.

The E-Step also computes the expected state transition count, $\xi_t(i, j)$, for every time t , and between every state i and j . For this, it uses (2.14), where the used A and B parameters come from the previous iteration, or the initial estimation.

In the Maximization-step (M-step), new A and B probabilities are recomputed by using the expected values from the E-step. To compute the probability that state j follows state i , a_{ij} , we divide the expected amount of transitions between state i and j , through the entire sequence, $\xi_t(i, j)$, by the expected total of transitions that happen from state i , as seen in (2.15).

For the estimation of an emission probability matrix value, $b_j(v_k)$, that is, the probability of having observation v_k on state j , we divide the expected amount of times the state happens with the respective observation, given by the notation $\sum_{t=1s.t.O_t=v_k}^T \gamma_t(j)$ in (2.16), by the total amount of times the state is expected to happen.

This iteration process is repeated until the algorithm reaches a convergence point.

E-Step

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\alpha_T(q_F)} \quad (2.13)$$

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(q_F)} \quad (2.14)$$

M-Step

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \sum_{k=1}^N \xi_t(i, k)} \quad (2.15)$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1s.t.O_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (2.16)$$

2.3 Behavior Trees

Behavior Trees (BTs) are a Control Architecture (CA), whose function is to structure the switching between different tasks in an autonomous agent, such as a robot [9]. An example of a BT performing a sequence of robotic tasks can be seen in Figure 2.2. The properties of a BT are useful in many applications, which has led to the spread of BTs from computer game programming to many branches of AI and Robotics.

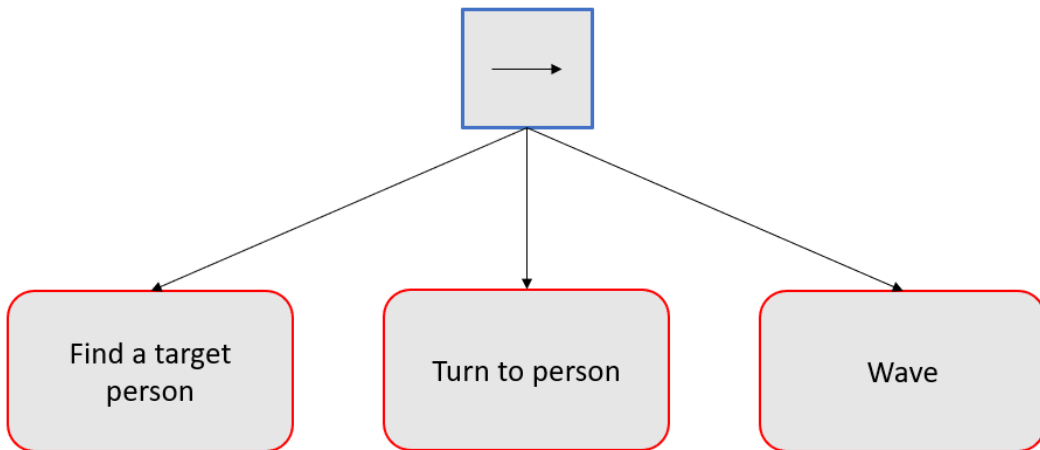


Figure 2.2: Behavior Tree with a task consisting of finding a target person, turning to him/her, and waving

BTs were developed by the computer game industry, as a tool to increase modularity in the CAs of Non-Player Characters, as an alternative to the previous control structures, which were composed

by Finite-State Machines (FSMs) [10]. Another alternative to FSMs are Petri Nets [19], though these present an alternative designed for the implementation of concurrent systems, instead of modular systems.

Two main characteristics put a BT ahead of a FSM for most of the Robotics systems [20, 21] and were quite relevant for the decision to choose them for our purpose. Firstly, their reactivity is very pursued in this area. A BT allows good handling of unexpected changes and errors by being able to check every condition and roll back to a previous task of the sequence, reacting quickly and efficiently. Since there are countless behaviors a person can have while greeting, unexpected events management is crucial for our work. The second property which BTs provide is the already mentioned modularity. Modularity, in the software industry, is the ability to decompose a system into smaller components that can be later recombined. This allows the different components to be developed and tested separately, which is highly beneficial in complex systems. As our system is precisely modular, composed of six distinct states which may run separately, it was very important for our testing to be able to implement a BT containing six smaller trees.

More formally speaking, a BT is a directed rooted tree where the internal nodes are called control flow nodes (blue in Figure 2.2) and the leaf nodes are called execution nodes (red). The root is the node without parents, and the leaves are the nodes without children, as in the common terminology of children and parents.

A BT starts its execution from the root node that generates signals (or ticks), which are sent to its children, that allow the execution of these nodes with a given frequency. A node is executed if and only if it receives ticks. The child receiving ticks will immediately return *Running* to the parent, changing to *Success* if it has achieved its goal, or *Failure* if not.

There exist three standard categories of control flow nodes: Sequence, Fallback, and Parallel.

The Sequence node is represented by a box with an arrow (\rightarrow) symbol, such as in the example tree. It follows an algorithm that consists of routing the ticks to its children from the left to the right until it finds a child that returns either *Failure* or *Running*, returning *Failure* or *Running* respectively to its own parent. It returns *Success* if and only if all its children return *Success*. When a child returns *Running* or *Failure*, the Sequence node does not route the ticks to the next child.

The Fallback node is represented by a box with a question mark ("?"). Its procedure corresponds to sending the ticks to its children from the left to the right until it finds a child that returns either *Success* or *Running*, then it returns *Success* or *Running* accordingly, to its own parent. It returns *Failure* if and only if all its children return *Failure*. Just like the Sequence node, when a child returns *Running* or *Success*, the Fallback node does not send the ticks to the next child.

Finally, the Parallel node is represented by a box with 2 arrows and it is defined by a parameter chosen by the user, M , where $M \leq N$, being N the number of children of the node. Its algorithm corresponds to routing the ticks to all its children, returning *Success* if M children return *Success*. It returns *Failure* if $N - M + 1$ children return *Failure*, and it returns *Running* otherwise.

The above information can be found summarized in table 2.2.

When an execution node receives ticks from its parent, it is programmed to execute a command or

verify a certain condition. It returns *Success* if the action is succeeded (or returned a True value) or *Failure* if the action has failed (False value). While the action is ongoing it returns *Running*.

Flow control type	Symbol	Success	Failure	Running
Sequence	→	All children succeed	One children fails	One child is running
Fallback	?	One child succeeds	All children fail	One child is running
Parallel	⇒	$\geq M$ children succeed	$> N - M$ children fail	Else

Table 2.2: Flow control nodes on Behavior Trees

To provide reactivity to the tree, the above-mentioned Sequence and Fallback control flow nodes keep sending ticks to the children to the left of a running child, in order to verify whether a child has to be re-executed and the current one has to be stopped, which is why they are called reactive nodes.

However, sometimes the user knows that a child, once executed, does not need to be re-executed. For these cases, we can use control flow nodes with memory, or non-reactive nodes. These specific nodes always remember whether a child has returned *Success* or *Failure*, avoiding the re-execution of the child until the whole Sequence or Fallback finishes (*Success* or *Failure*).

In this work, we will further use the terminology Reactive Sequence and Reactive Fallback, and (Normal) Sequence and (Normal) Fallback.

2.4 State of the art

Heenan et al. [22] built the only solution to directly predict Kendon’s phases and replicate them with a social robot, however, the authors opted for a Finite-State Machine (FSM) to control the state changing. This approach was not very reactive, since it could only change state when the present action had ended and was not particularly dependent on the target person’s behavior to do it. The model acquired simple information from an external sensing system, such as the detection of people and their pose, in order to change into the following state, but ignored many of the social signals involved, including their uncertainty. The model’s flexibility was also a problem, as the robot always performed the sequence of states in the same order: IA, DS, APP, FA, and CS, missing the *Head Dip* phase, which was not implemented. A similar implementation using an HMM is suggested by the authors, as it is more robust in the above mentioned topics.

To the best of our knowledge, so far, no other works have implemented a social robot that was able to estimate phases or parts of a complete greeting model.

Other social robots [23–25] which replicated more than one greeting phase sequentially opted for changing-phase rules simpler than our model and the one above referred to. These rules usually were not related to social signals, but mostly to the position of the person and the conditions of the environment, such as their availability to be greeted or if the path was blocked. Also, these were usually used to change between their own greeting phases, and never to estimate them in a person that is greeting.

Regarding the implementation greeting for mobile social robots, directly or indirectly, there are already instances of experiments which implemented some of Adam Kendon’s phases for Human-Robot

Interaction. Yet, no projects have implemented the entire greeting model, with all the phase characteristics.

For the rest of this section, we analyze some of these experiments, where parts of greeting were implemented using a robot, and we split them into their diverse contributions for each phase of Kendon's greeting model, as well as the achievements of their models. Avelino et al. [26] previously organized some of this information.

Initiation of Approach

On what concerns the first phase of a greeting sequence, the *Initiation of Approach*, most social robots are programmed to replicate it. A common detection of a person by the robot, followed by the respective orientation and/or gaze may be an example of this phase, and this happens, most of the times, without knowledge of Kendon's model.

For instance, Shi et al. [24] projected a robot to distribute flyers, using two distinct procedures for this phase, as the authors tested two delivery methods. The first was a wait-and-handing method, i.e., the robot waits for pedestrians to come through and only extends its arm (containing the flyer) when a person is near it. Here, they opted to identify the person, but only gazing directly when he or she was less than 3 meters away, without doing any change of orientation. On the second method, the robot chooses a target person, approaches and extends the arm. In this case, the *Initiation of Approach* phase consisted of a direct gaze and orientation as soon as the robot had chosen its target. Regarding this second case, the robot could identify several people simultaneously and choosing one target, using a complex algorithm to determine the best target person. For this, the expected gain from distribution was maximized, i.e., the number of people it would be able to handle a flyer by unit of time, and minimized a disturbance factor, increased by being close to people and disturbing their walking.

Satake et al. [25] used the same robot to approach people in a shopping mall to give them useful information, creating a model based on anticipating pedestrians' movements. It calculated the approach plan for each person on its sight of view and opted for the one which maximized the likelihood of a successful approach and the awareness from the target. As the best plan is chosen, the robot started its approach, without any other gesture.

On the project which aimed to directly recreate Kendon's greeting model for HRI [22], the authors used a small robot named *Nao* and followed a sequence of steps described by Kendon through an FSM, as referred before. In the first phase, the robot's behavior is simple. It displays an idle behavior until it detects the presence of a human, where it starts to look at him or her.

Examples such as [27] chose a simpler approach here, as the social robot used only selects a target person from the ones available. This was adequate given that the robot's function was to simulate a host greeting guests entering a building, and the robot was already pointed at the entrance. This host robot was implemented with three levels of enthusiasm and behaves differently on each one.

Similar to the previous case, [23, 28] are also instances of projects which could develop a social robot to select a target person in public spaces and started a direct gaze with the intention of beginning an interaction.

Our procedure in this phase will follow [22], however, only partially. Our robot will first change its orientation to the target’s direction, before fixing the gaze, as Kendon described the phase.

Table 2.3 summarizes the above information, by the several contributions made to this phase of the greeting model. From the six analyzed reproductions, only four contained a change of orientation and gaze direction to the target, as Kendon had described humans do after sighting someone. To replicate this is essential for an HRI, providing the human the robot’s intention of greeting.

	Selecting a person	Frontal orientation	Direct gaze
Initiation of Approach	[22–25, 27, 28]	[22–24, 28]	[22–24, 28]

Table 2.3: Common characteristics of *Initiation of Approach* phase developed on other projects

Distance Salutation

Saad et al. [27] developed an instance of a *Distance Salutation*, following every time the robot chose a guest to greet. The host robot waved at the target to draw attention and initiate interaction, and combined it with a verbal greeting if put in a moderate or high level of enthusiasm.

Heenan et al. [22] implemented a *Distance Salutation* based on Kendon which always followed the *Initiation of Approach*, unless it noticed the person had not responded. Here, the robot oriented its body to the person and performed one movement, which could be a wave, a head toss, head lower, or head dip.

Our method for this phase will follow the latter project, performing a salutation while looking straight at the target. None of the two referred examples implemented a smile on this phase, as Kendon described. This work will not include it as well, due to implementation limitations.

In Table 2.4 we present the contributions on this area, which are very scarce regarding social robots. The DS phase was found by Kendon in every human greeting, however only the two mentioned works implemented it. From these two, [22] reproduced it more similarly to human greeting, since a direct looking is a main characteristic of these long-distance salutations.

	Direct gaze	Wave or other distinct movement	Smiling
Distance Salutation	[22]	[22, 27]	No records

Table 2.4: Common characteristics of *Distance Salutation* phase developed on other projects

Head Dip

Even though Kendon refers to be common to observe a head dip movement following the DS phase, the only mention of this type of movement replicated on HRI is in [22]. However, the authors did not develop it as a separate phase. Instead, the movement was included as one of the possibilities for a DS movement, and it does not appear to be very used during the study.

Contemporary social robots do not use this greeting detail, likely due to its subtlety on human greetings. Hence, only projects based on a deeply detailed greeting analysis like Kendon’s could opt to use

it. Our approach will be, as mentioned, to replicate the *Head Dip* as a separate phase which sometimes follows the DS, following what Kendon described for human greetings.

Head dip movement following a salutation	
Head Dip	<i>No records</i>

Table 2.5: Common characteristics of *Head Dip* phase developed on other projects

Approach

Several experiments have tried to approach people with a robot. For example, the distributing flyers robot [24] started its approach movement as soon as it chose the target, using a behavior controller, which set the approach to be done from the front left/right side. It also maintained a direct gaze through the entire approach.

There are other approaching procedures, for instance, [25], which opted for a frontal approach and the robot was programmed to only stop the approach when it detected the person had also stopped and was ready to interact.

Kendon's model replication [22] also opted for a frontal approach. This model was the only one to add more details about this phase: the aversion of eye contact and, in some experiences, distinct movements, such as the "body cross" and grooming, as can be seen in Table 2.6.

Pepper host robot [27] produced a small approach movement, if put on the high enthusiasm level. This approach consisted of simply approximating 0.3 meters in the direction of the target, which the authors considered would draw more attention.

Brscic et al. [23] also managed to have the robot approaching a selected target, however, did it maintaining the gaze assumed on the first phase.

Our *Approach* phase will consider a partial aversion of gaze, mentioned by Kendon and will be mostly frontal, thus, similar to [22]. This gaze aversion is highly present in human greetings where people have the lower social position in the meeting. To avoid displaying an over-confident behavior or bring discomfort to the person, we believe a partial aversion of gaze is the best approach for this feature. The distinct movements mentioned are also relevant for bringing naturalness to the sequence.

	Movement toward person	Total/partial aversion of gaze	Distinct movements ("body cross" or grooming)
Approach	[22–25, 27]	[22]	[22]

Table 2.6: Common characteristics of Approach phase developed on other projects

Final Approach

From all robots which approached people, only a few examples could clearly divide the approach, adding a *Final Approach* phase.

Shi et al. [24] changed slightly its approach during it, by starting to extend the distributing arm and reducing its velocity as the person was closer, which may be understood as a *Final Approach* phase, since the robot is getting ready for interaction. Here, the robot also says "Please have a flyer".

Satake et al. [25] started a different approaching process as soon as the robot entered the social zone - less than 3.5 meters. Here, the robot prepared to start a conversation with the person, and it aimed to clearly show that the robot's intention was an interaction. On this behavior, the robot quickly oriented its body direction toward the approaching person, unless it detected that the person was already accepting to interact or that was leaving.

In [22], the authors clearly separated the two phases, following what Kendon had described. Whenever the robot entered a person's personal distance (less than 1.2 meters of distance), the robot oriented its head, simulating eye contact. As can be seen in Table 2.7, this was the only project to change according to the most evident difference stated by Kendon between the two approaching phases. This gaze and orientation adjustment is seen by Kendon as a common way to prepare for the interactions and provide the others that idea. For this phase, we will implement a procedure containing this adjustment.

	Change of gaze	Smiling
Final Approach	[22]	<i>No records</i>

Table 2.7: Common characteristics of Final Approach phase developed on other projects

Close Salutation

A reproduction of a *Close Salutation* phase can be found in most social robots that approach people. However, in the majority of these cases, the CS phase is limited to a stop in the movement and a verbal greeting.

The already mentioned robot distributing flyers [24] stopped near the targets and kept a frontal orientation and a direct gaze until they picked the flyer or the robot considered they had passed it.

In the experiment in which the same robot approaches people to talk to them [25], after the approach, the robot maintains a position in front of the target and only starts a conversation.

In [23] the robot also produced a verbal greeting after approaching, while [28] simplified this by waiting for people to come closer to the robot to greet them.

A stretch of the arm, asking for a handshake from the person, ending a greeting sequence was only found on the project that directly replicated Kendon's model [22]. Here, the handshake is accompanied by a verbal greeting and a continuity of the head position from the FA phase. The behavior to implement on our robot for this phase is expected to be similar to this project, following Kendon's description.

The contributions for the CS phase are found in Table 2.8. We believe that using contact salutations is more appropriate to our type of environment, since most people are accustomed to it and these tend to start more comfortable interactions, comparing to verbal-only greetings. Gazing directly was seen by Kendon as a main feature of this phase on the American and Western type of greeting. Hence, the projects that implemented it seem to have reproduced human behavior better.

	Verbal greeting	Direct gaze	Smiling	Handshake or other movement
Close Salutation	[22, 23, 25, 28]	[22, 24]	<i>No records</i>	[22]

Table 2.8: Common characteristics of *Close Salutation* phase developed on other projects

Greeting achievements on HRI

Shi et al. [24] had a superior performance at delivering flyers than the experiment previously made with human distributors. 18% of people passing accepted a flyer offered by the robot, while only 10% accepted from humans. However, the authors concluded that one of the main reasons for the success was that people tend to find robots interesting and rare. Hence, a share of people only interacted with the delivering robot for curiosity. The authors also stated that they could have had greater results if the robot performed actions described by Kendon, such as smiling, waving, or greeting the person, since it would appear to be friendlier.

Satake et al. [25] demonstrated a success rate of 55,9% for the approaching technique, which the authors consider "reasonably high". The targets in this study were people going through a shopping mall, which might explain some reluctance to interact with the robot.

The already mentioned enthusiastic host robot [27] achieved an attentiveness score of 84% for the mild level of enthusiasm, 77% for the moderate, and 95% for the high, even though attentiveness was considered whenever an entering guest at least looked at the robot.

Kendon's greeting model's replication work [22] faced some problems with their implementation. The robot used was 58 cm tall and somewhat fragile, which caused several limitations. They had issues in all kinds of physical contact, with people avoiding it, as they were afraid to damage the robot. Also, its movements were too slow, causing that people, sometimes, had to wait for points of the interaction that should be barely noticeable, which seriously limited the greetings' naturalness. Given the predicted enhancements in our phase-changing model comparing to this project, stated before, and using a robot more appropriate for these experiences we expect more satisfactory results, mainly on the model flexibility and naturalness of greetings.

Chapter 3

Greeting Model using a Hidden Markov Model

3.1 Global Overview of the model

A Hidden Markov Model (HMM), as introduced before, is a probabilistic model that alternates between a set of phases, called hidden states. To operate properly, the model only needs to receive a previous state and a set of features, called observations, being extracted at a time t . Using this, it can estimate the most likely state at this time and switch the present state to it.

As presented in the global diagram of this section (Figure 3.1), our set of hidden states will be the six distinct phases Adam Kendon identified in his greeting analysis [6]. To have the model predicting phases accurately, there was a need to choose observable features that characterized only some phases or could allow to distinguish them. The chosen observations were the person's distance, speed, gaze direction, smile intensity, and head or arms movements. In section 3.3 there is a detailed explanation about these five features.

To develop the HMM, two sources of information were used. Firstly, videos with human greetings' data were obtained from external datasets, serving as the base for our model to learn. These videos, after going through a process of data extraction and labeling (detailed in section 3.5.1) were used as the input for HMM's learning problem, where the Expectation-Maximization algorithm estimates the parameters of the model. This model will onward be mentioned as the "Data-Driven Model", and will be the HMM to use for our experiments. The other source of information were Kendon's notes. These provided detailed information about the phases, used to manually estimate the 4 probability matrices which characterize a Gaussian HMM. This model, further mentioned as the "Kendon Model", will be mostly used as a comparison, to ensure that the Data-Driven Model accurately represents Kendon's greetings.

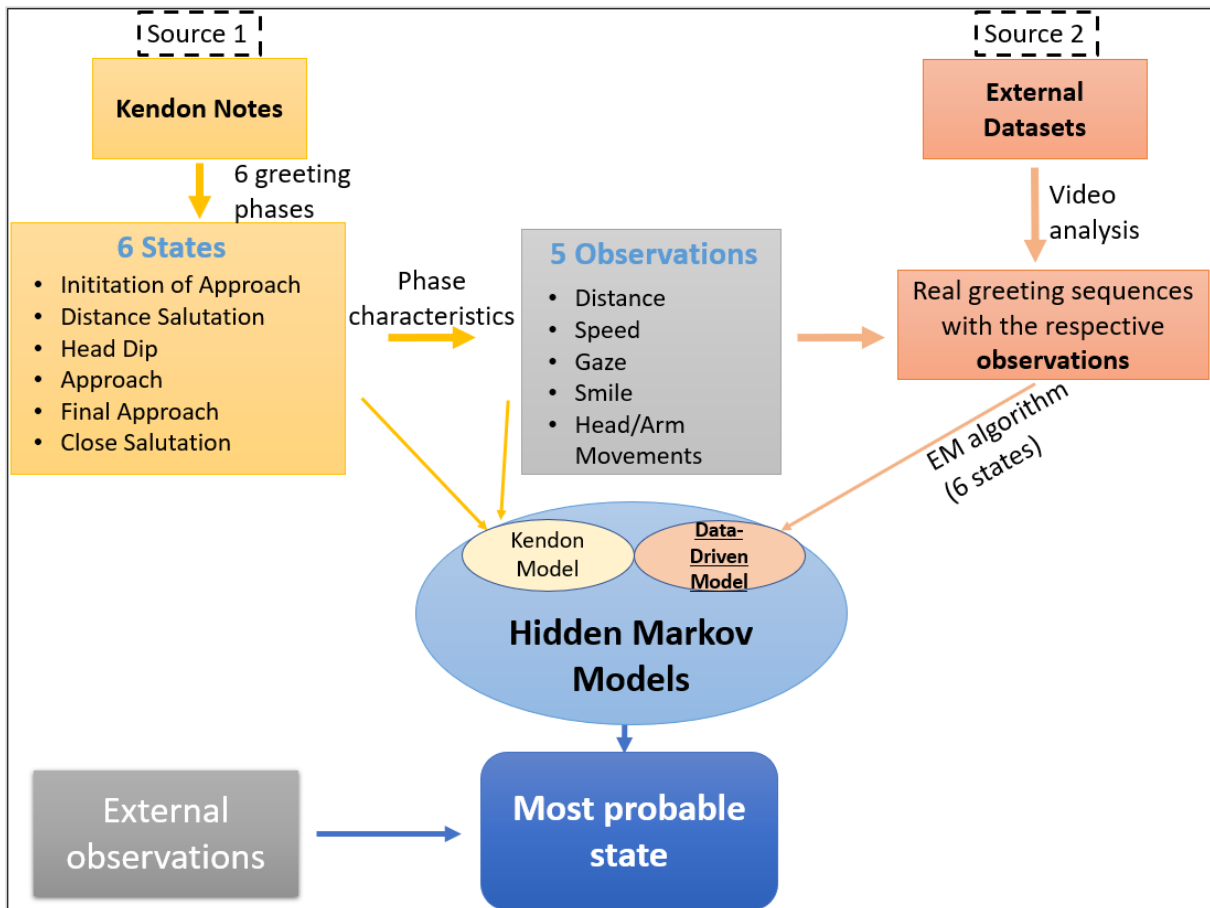


Figure 3.1: Global diagram of the greeting model implemented as a Hidden Markov Model

3.2 Robot Operating System

The Robot Operating System (ROS) [29] is a collection of software frameworks for developing robotics software, used in this work. It is an open source project containing several tools, libraries, and conventions created to simplify robotics development. ROS allows the creation of complex, yet robust robot behavior across a wide variety of platforms.

Here, we will present some of the elementary concepts of ROS, which are essential for a good comprehension of this chapter.

A ROS system is often composed of several computation processes, called ROS nodes. Nodes are combined into a graph and communicate with others mainly through ROS topics or services. A node usually commands a fine part of a robot control system. For instance, a moving robot like Vizzy must have at least one node related to the wheels' motors, other dealing with the robot's localization, other performing path planning, and many others.

The use of nodes allows superior failure management, as flaws are isolated individually. The complexity of the code is also reduced, while the nodes also support several alternate implementations and flexibility with programming languages. A ROS node always has a unique name identifying it for the system and can be written using ROS client libraries, such as rospy (Python) or roscpp (C++).

The first mean of communication inside a node graph is a ROS topic. A topic can be seen as a distinct

subject to which nodes can publish information (being a publisher node) or receive all the transmitted information on that topic (being a subscriber node). Every ROS topic has a specific message type, to which every publication must comply.

For instance, a path planning node could publish information to a topic containing the linear and angular velocity the robot should reach. This topic would have a subscriber node related to the wheels' motors, which obtains the velocity information and moves accordingly. A single node can both publish and subscribe to more than one topic, if needed.

Communication can also be made using ROS Services. These allow a two-way synchronous transmission, instead of the publish/subscribe model, given through a request and reply interaction. A ROS Service can be seen as a server, which receives requests with a message from client nodes and synchronously returns another message, which may or may not be different. Both request and response sides have a specific message format that must be respected. ROS Services are normally used for simple computations and quick actions, since the requesting node is blocked until it receives the response message.

3.3 HMM Observations

As described in Figure 3.1, the Hidden Markov Model will depend on a series of observable events called observations, which it will use, together with the present state, to predict the most probable state at each moment.

Figure 3.2 illustrates four of the five observation features used, on an example of the robot observing a person: his/her distance, the respective speed, the direction of gaze, and characteristic head and arm movements. The fifth feature used was the intensity of target person's smile.

Regarding the observations, this process starts when the robot detects a person, and from there it delivers observation vectors, one every 0.2 seconds. The value of 5 observations per second for the sampling rate was chosen to balance being high enough for the model to be very reactive, predicting the states without noticeable time gaps, but also low enough to not exceed any other robot or ROS connection rate, provoking errors. This leads to the Hidden Markov Model receiving new external observations and determining the most probable state in each time interval of 0.2 seconds, where the external observation vector contains information about the five features already mentioned.

For this, a ROS node named *Face Extraction* uses a Computer Vision (CV) software (OpenFace [30]) features to obtain the necessary information to extract the features. When there is a detection of a target face, the node publishes on a ROS topic named `"/faces"` a message of type *FaceExtraction*, which contains the following information:

- A 3D vector representing gaze direction of the left eye in the robot's frame; [31]
- A 3D vector representing gaze direction of the right eye in the robot's frame;
- A vector containing the 3D position of the face in robot coordinates;

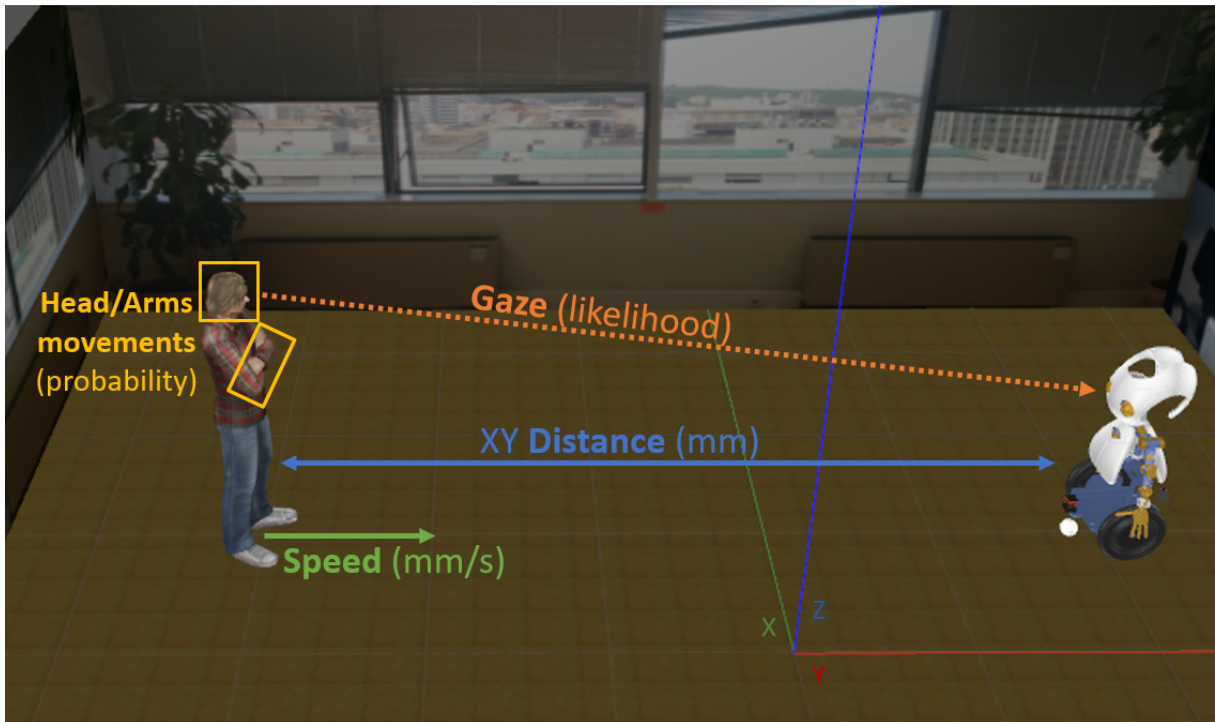


Figure 3.2: Representation of 4 greeting features which were chosen as HMM observations: distance, speed, gaze and movements

- A vector which contains the rotation of the face around the 3 axes, in robot coordinates;
- A vector containing the 2D position of the borders of the left eye, in robot coordinates; [32, 33]
- A vector with the 2D position of the borders of the right eye, in robot coordinates;
- A vector with the intensity of Action Unit 06 and Action Unit 12. [34]

From the list above, the four items using robot coordinates were first obtained from OpenFace in its own coordinate frame, with the origin being the camera, which, in the case of Vizzy, is present in its eyes. Later, these values were transformed to a coordinate frame whose origin is the base of the robot, named "base_footprint". In Figure 3.3 we can see this coordinate frame, together with the world frame, which has the origin in the center of the room.

3.3.1 Distance

To identify phases in a greeting sequence, in which people start far away from the robot and end up close, the distance between both is a key factor.

The value used for this observation will represent the distance from the center of the robot's base to the face of the target person, projected on the ground plane, so that it is not influenced by the height of the person, as illustrated in Figure 3.2.

The distance is, then, given by the norm of the point on the XY plane, as suggested by the mentioned

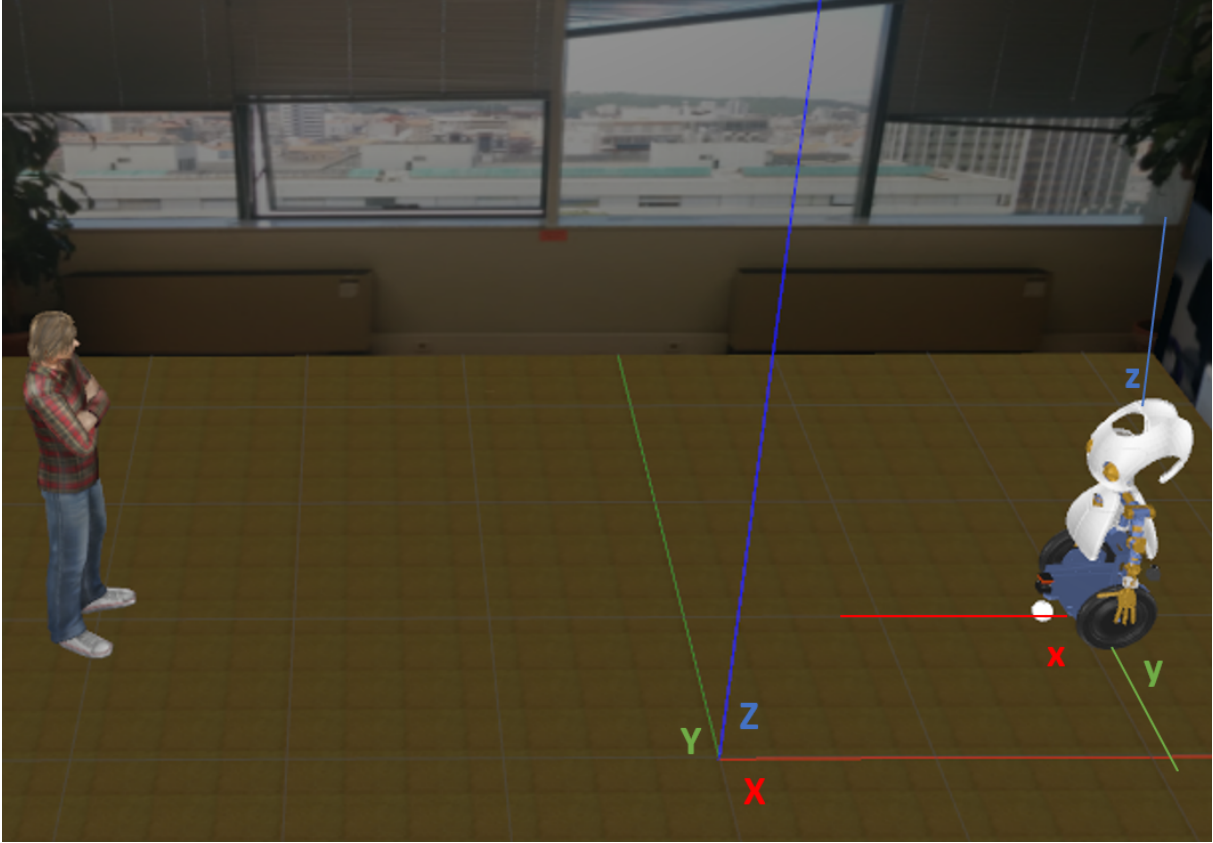


Figure 3.3: Representation of the base_footprint (x,y,z) and world (X,Y,Z) coordinate frames

figure. Ignoring the height coordinate, we consider $p'(x, y)$, and the distance in the following equation:

$$Distance = \|p'\| \quad (3.1)$$

All distances used on this model are given in millimeters (mm).

3.3.2 Speed

The speed of the greeting individual is another very important factor, as it provides information to separate between the static part of the model (before and after the approach) and the moving part.

To calculate speed values with a simple method, the robot detects how much distance the target has moved in its direction in the respective 0.2 seconds interval, and divides it by the time interval, calculating the average speed on that time frame.

More formally, saying we want to calculate the speed feature at time t . Firstly, we save the robot's position, p'_R , on the previous observation, $(t-0.2)$. Then, we obtain the distance from the previous observation, $PreviousDistance$, and the distance between p'_R and the person's position at time t , p_P . Having these 2 values, we can compute how much a target has moved during a given time interval by (3.2), and we use (3.3) to calculate the average speed. This situation is represented in Figure 3.4 for clearer comprehension.

$$Distance' = \|p''_R - p'_P\| \quad (3.2)$$

$$Speed = \frac{(PreviousDistance - Distance')}{0.2} \quad (3.3)$$

In the above equations, $p''_R = (x, y)$ and $p'_P = (x', y')$ are the equivalent of p'_R and p_P on the XY plane.

Identically to the distance observation, all speed values in the model were used in millimeters per second (mm/s).

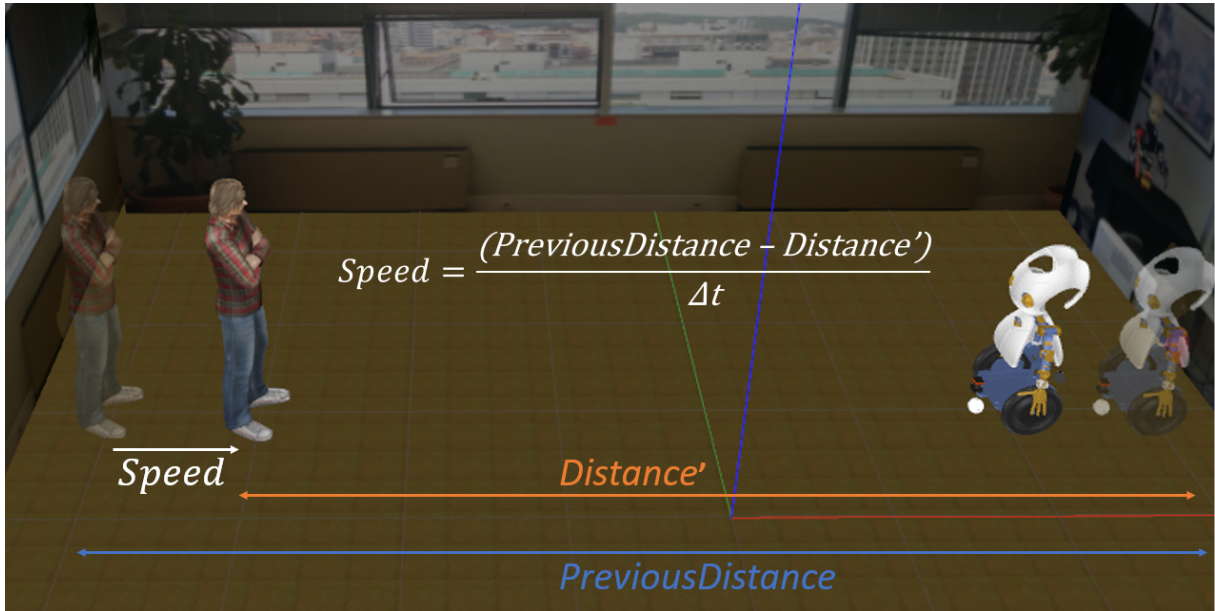


Figure 3.4: Example of the speed calculation for every observation on the HMM

3.3.3 Gaze

As stated many times in section 2.1, the direction of gaze is also a meaningful characteristic of Kendon's greeting phases. It is more likely to be found during both salutation phases and in the *Final Approach*.

To compute a target person's gaze direction, we used two features which can be output from the OpenFace software: the head orientation (blue cube in Figure 3.5) and eye gaze direction vectors (green lines coming from the eyes in Figure 3.5).

These two features were used to build two different gaze detectors. Firstly, we used the eye direction vectors to build *Gaze Detector 1*, however, it lacked in accuracy as the camera moved away from the person. Although it is theoretically more accurate to estimate gaze through the direction the eye is pointing, Kendon [6] stated that the head's orientation revealed, in most cases, the direction where the person was looking. Thus, according to the difficulties in estimating eye gaze direction at further distances, we also built *Gaze Detector 2*, using head orientation.

Both detector models will estimate a gaze point, that is, a point projected in the robot's YZ plane to which the gaze direction of the person is pointing. This point will be used to compute the likelihood of direct gaze, by analyzing its distance to the robot's face.

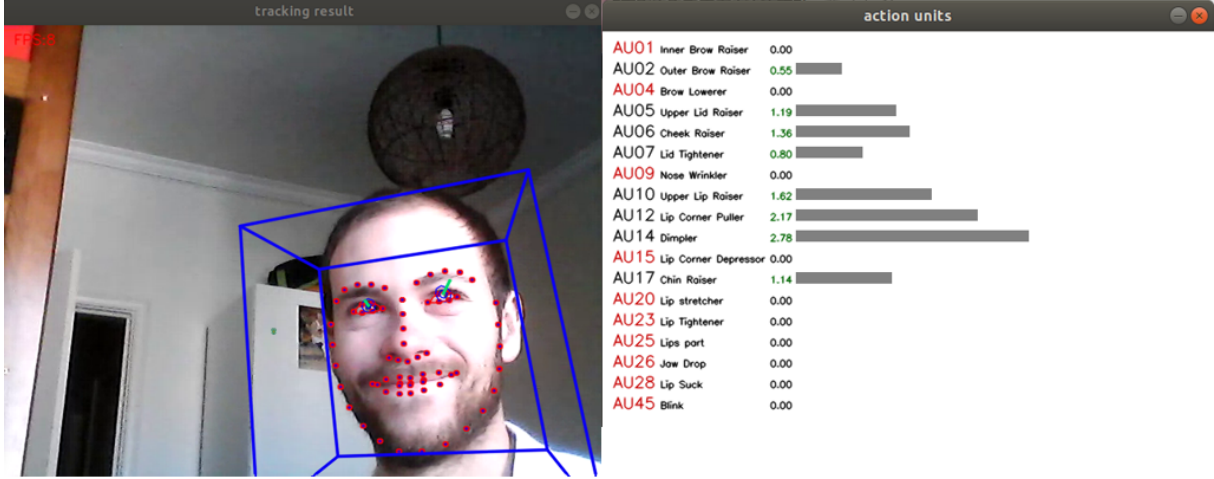


Figure 3.5: Output of OpenFace: left image represents face landmarks, face orientation and gaze direction; right image contains the list of detected AUs and their intensities

Gaze Detector 1

As stated before, OpenFace's output contains two 3D direction eye gaze vectors: g_0 , which represents the left eye in the image and g_1 the right eye, as in Figure 3.5.

After extracting the mentioned information, the following step was to obtain the 3D position of the center of the eyes. For this, we used landmark detection by the OpenFace and extracted, for the left and right eye, landmarks 36 and 39 and landmarks 42 and 45, respectively, labeled in Figure 3.6. These landmarks represent the left and right border of each eye and, averaging these points, we got the information needed.

Having the position of the eyes and the gaze direction vectors, we could get the points at which each eye is gazing, projected in the YZ plane of the robot's base coordinate frame. For this, we used the two following equations, where gp_0 and gp_1 are the gaze points from the left and right eye, respectively, and e_0 and e_1 are the center of both eyes, with everything being computed in the base_footprint referential:

$$gp_0 = g_0 + e_0 \quad (3.4)$$

$$gp_1 = g_1 + e_1 \quad (3.5)$$

Assuming the gaze point gp is the average of the left eye and right eye gaze, we computed the following equation:

$$gp = \frac{(gp_0 + gp_1)}{2} \quad (3.6)$$

Finally, we needed to decide a way to understand how likely are people to be looking, according to how far their gaze point is from the center of the camera. And, more importantly, where to start considering people are gazing.

For the likelihood problem, we used the cone model for the field of view of a human [35, 36], which is based on picturing a cone, where the vertex is located in the person's eyes and the base is centered in the point to which we are looking. Larger opening angles of the cone are associated with images

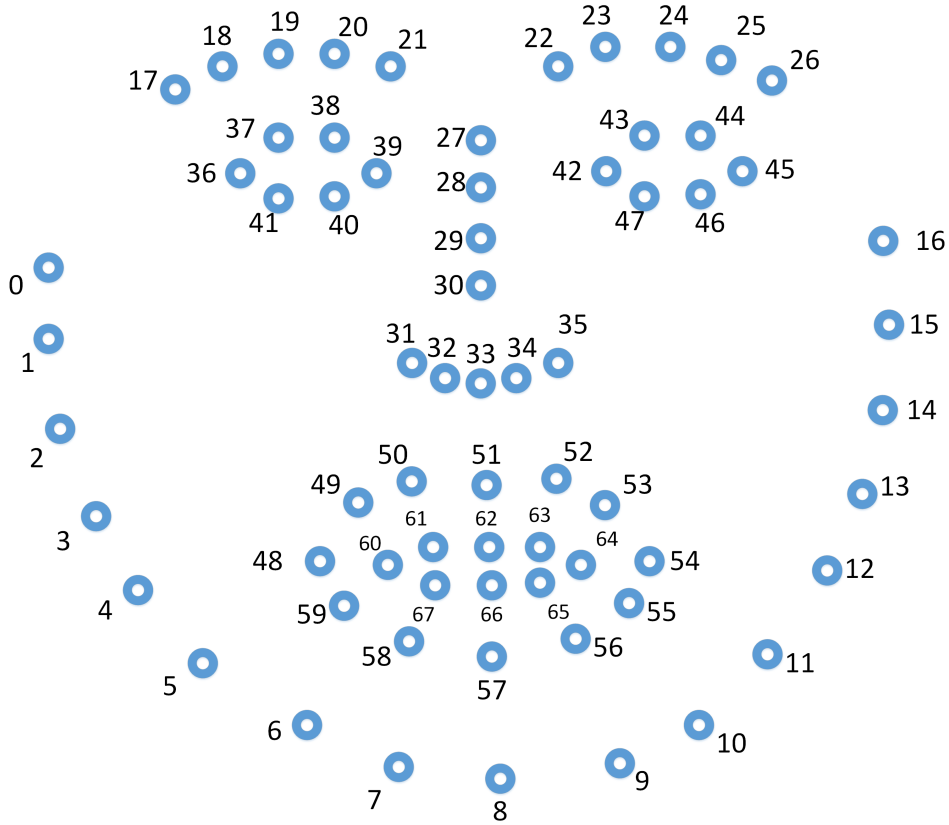


Figure 3.6: Face landmarks labeling for OpenFace

progressively more blurred, while smaller angles bring more details in the view. Our visual attention is commonly considered to begin with angles smaller than 60° , which signifies that a person can already distinguish colors, shapes, and people, when these are inside this angle limit, horizontal and vertically.

According to this model, we built a cone using a 60° angle and a base centered on the gaze point calculated before. Here, we consider that if the robot's face is inside the circle formed by the cone, such as Figure 3.7 indicates, then the person is likely to be gazing at it. In other words, the distance between the gaze point and the center of the robot's face is less than the cone base's radius.

Having the center of the robot's face, f , in base.footprint coordinates, we can, therefore, express this gaze situation as $\|gp - f\| < radius$. Following this line of thought, a non-gazing situation would be represented by $\|gp - f\| > radius$ and a 50% gaze likelihood would be a case in which the camera is at the circumference, i.e., $\|gp - f\| = radius$.

To calculate the radius (r), we picture a right-angled triangle on the upper side of the cone, formulating the following equation:

$$r = \tan\left(\frac{\theta}{2}\right) d \quad (3.7)$$

Here, θ is the opening angle, which is 60° , in our case, and d is the distance between person's eyes and the gaze point, represented with a dashed-dotted line in Figure 3.7.

To estimate the likelihood of direct gazing from the target, we created a ratio between the gaze-to-face distance and the size of the cone. $Gaze = 1$ would correspond to a 50% likelihood, with higher

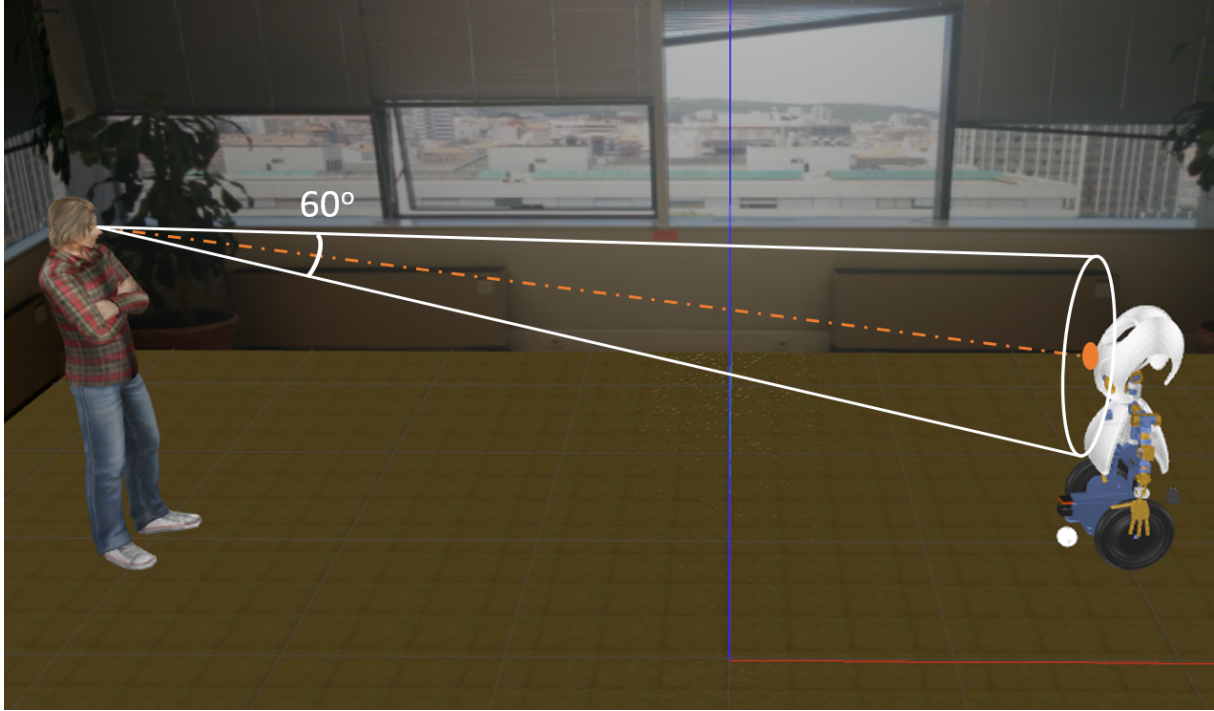


Figure 3.7: Gaze Detector 1 situation: the eye gaze direction forms a cone around the robot, considering there is a direct gaze

values of $Gaze$ indicating lower probabilities and lower values, higher probabilities. As stated, $Gaze$ is given by:

$$Gaze = \frac{\|gp - f\|}{r} \quad (3.8)$$

Gaze Detector 2

This second method uses the orientation of the head to estimate the point where the person concerned is looking at. Then, it uses an approach similar to *Gaze Detector 1*. It verifies if the face is inside the circle formed and computes a value which represents the gaze's likelihood.

The head orientation output from OpenFace contains 3 rotation values, r_x, r_y, r_z , each one around axis X, Y or Z, respectively.

Since we seek the point where target person is looking in the YZ robot plane, a 2D approach for the gaze point was used. The 2 following equations represent an estimation for the gaze point, $gp = (gp_y, gp_z)$:

$$gp_y = |t_x| \tan(r_z) - t_y \quad (3.9)$$

$$gp_z = |t_x| \tan(r_y) + t_z \quad (3.10)$$

Where $T = [t_x \ t_y \ t_z]$ is a vector with the position of the face along the three axes.

Having the gaze point calculated, the setup is, from now, similar to *Gaze Detector 1*, following (3.7) and (3.8) in order to get similar gaze values using the two methods. A practical example of this method is sketched in Figure 3.8, with a situation where a direct looking would be less than 50% likely.

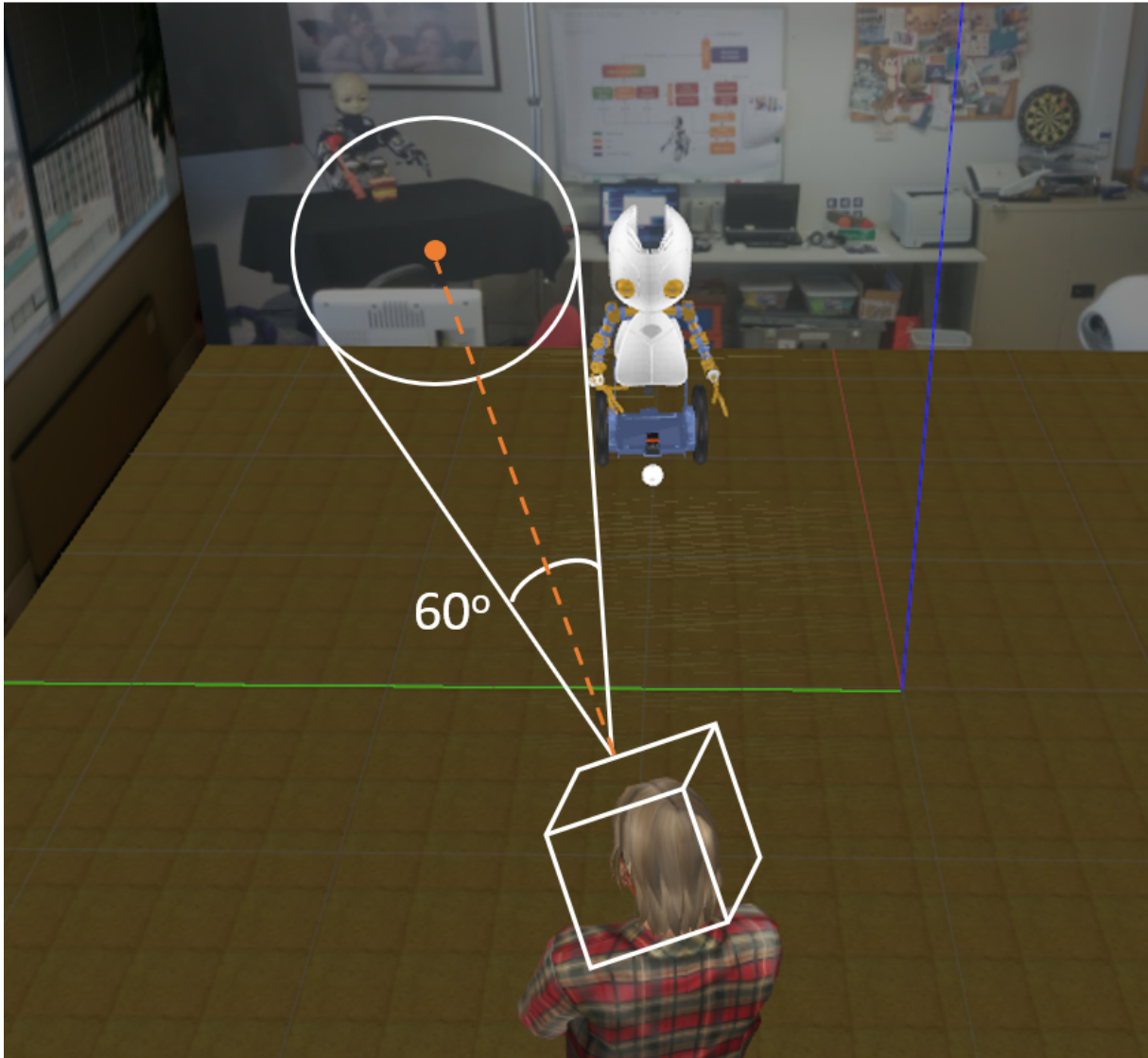


Figure 3.8: Gaze Detector 2 situation: the head orientation direction forms a cone which does not include the center of the robot's head, considering there is not gaze

3.3.4 Smile

Smiles are yet another very important characteristic of some phases of Kendon's greeting model and they will function as another observation feature. Smiles are very common on the interaction parts (eg. both salutations), though they tend to disappear during the *Approach* phase.

To create a reliable smile detector, we used the Action Units (AUs) detector from OpenFace. AUs are an approach to deconstruct nearly every facial expression existent invented by the Facial Action Coding System (FACS) [37]. The FACS describes AUs as the contraction or relaxation of one or more muscles. AUs are commonly used in various fields of study to detect emotions, such as fear, depression, or pain, as these are generally linked with one or a group of AUs. According to several studies, for instance, [38, 39], the feeling of happiness is physically displayed by the combination of AU6 and AU12, i.e., the combination of a raise of the cheeks and a pull of the lip corners (see Figure 3.9).

OpenFace outputs these AUs in an intensity scale from 0 to 5, where 0 is not present, 1 is present


Upper Face Action Units					
AU 1	AU 2	AU 4	AU 5	AU 6	AU 7
					
Inner Brow Raiser	Outer Brow Raiser	Brow Lowerer	Upper Lid Raiser	Cheek Raiser	Lid Tightener
*AU 41	*AU 42	*AU 43	AU 44	AU 45	AU 46
					
Lid Droop	Slit	Eyes Closed	Squint	Blink	Wink
Lower Face Action Units					
AU 9	AU 10	AU 11	AU 12	AU 13	AU 14
					
Nose Wrinkler	Upper Lip Raiser	Nasolabial Deepener	Lip Corner Puller	Cheek Puffer	Dimpler
AU 15	AU 16	AU 17	AU 18	AU 20	AU 22
					
Lip Corner Depressor	Lower Lip Depressor	Chin Raiser	Lip Puckerer	Lip Stretcher	Lip Funneler
AU 23	AU 24	*AU 25	*AU 26	*AU 27	AU 28
					
Lip Tightener	Lip Pressor	Lips Part	Jaw Drop	Mouth Stretch	Lip Suck

Figure 3.9: Action Units visual example and description

with minimum intensity, and 5 is present with maximum intensity. Figure 3.5 contains some values for the AUs on a real example.

Acknowledging that a smile could be represented by the combination of these AUs and that these have generally the same importance, a simple approach for the smile feature would be based on the average of these two values, as in the following expression:

$$Smile = \frac{AU6 + AU12}{2} \quad (3.11)$$

3.3.5 Movements

There are three phases of the model which include characteristic movements (i.e. gestures) of the person's arms or head. These are the *Distance Salutation* (waving and tossing or lowering the head), the *Close Salutation* (handshakes or DS movements but at a closer distance), and the *Head Dip*.

Due to its implementation complexity, automated detectors of these movements were not developed.

Here, the solution chosen rested on a non-automated process which depended on an external person pressing different keys on the keyboard, whether movements from different kinds were performed by the greeting person. Each key pressed returned a probability of 1 of detecting each kind.

To be viable to use these features on a Gaussian HMM, instead of using only one field of the observation vector to differentiate between the three types of movement, three more fields were added, *DSal*, *CSal*, and *HDip*. These fields contain the respective probabilities returned from the detector.

Therefore, the observation vector extracted in every time frame for the HMM consists of the following:

$$O = \left[\textit{Distance} \quad \textit{Speed} \quad \textit{Gaze} \quad \textit{Smile} \quad \textit{DSal} \quad \textit{CSal} \quad \textit{HDip} \right]$$

3.4 Kendon Model as a Hidden Markov Model

Based on Kendon's analysis of the several greeting phases, we created a Hidden Markov Model using the six states summarized in Section 2.1 and the five observation features described in Section 3.3. This HMM will be onward called the Kendon model.

In this subsection, we provide another description of Adam Kendon's greeting notes, however, more focused on the values necessary to estimate the HMM's parameters: the transition probabilities, the values of the observations for each state, and the initial state probabilities.

1. *Initiation of Approach* (IA)

Kendon does not mention any values about the duration of the IA phase. However, some examples of the observed greetings are detailed in his greeting analysis. From these, we assume that, on average, 0.5 seconds are sufficient to complete the short movements that compose this phase. Regarding this, as the transition's period is 0.2 seconds, we assume that, in average, there are 2 same-state transitions before changing to another, so, $P(IA|IA) = 2/3$.

According to Kendon, people tend to move first before performing a DS phase. We may assume 1 in every 10 greetings will salute without any approach first, so, there is a 10% probability that the last transition of this phase is to the DS. Therefore, $P(DS|IA) = 0.1/3 = 1/30$.

Kendon also states that the head dip movement always happens after a *Distance Salutation*, so it appears to be impossible to be found after the *Initiation of Approach*.

Following *Initiation of Approach* with an *Approach* is, by Kendon, the most likely scenario. Here, we assume 60% of sequences follow this. Following the line of thought of the other probabilities, $P(APP|IA) = 0.6/3 = 1/5$.

Although Kendon does not mention this, there is a possibility that the movement toward the person starts already at a close position. Thus, we assume 2 in every 10 greetings to go from sighting to the *Final Approach* phase. Thus, $P(FA|IA) = 0.2/3 = 1/15$.

To finish, the direct transition from this phase to a CS phase is not referred by Kendon, however, we assume 1 in every 10 greetings go from sighting to a *Close Salutation*, as a person could be approaching the other without being on their sight of view, for instance.

The above methods and assumptions will be similar for the following phases. Based on these, an estimation for the transition probabilities from the *Initiation of Approach* state for the Kendon model has the following values:

IA	DS	HD	APP	FA	CS
2/3	1/30	0	1/5	1/15	1/30

Regarding the observations' mean values for the *Initiation of Approach*, the average distance would have to be higher than the DS state, as this one always happens further in the sequence. Kendon gives this phase an average distance of around 5.5 meters, hence, we will consider 7 meters for the IA. Also, this phase is almost static, so the average speed would be around 0.

This phase includes orientating to the target person, where Kendon states it is not common to look directly, and the start of the approach, where the gaze is already common. So we consider direct gaze 50% of the time, if both half-phases have similar duration, which corresponds to $Gaze = 1$, by the section 3.3.3.

In what concerns to smiling, Kendon has no mentions on it, therefore the average is most likely a neutral face ($Smile = 1$), neither there are usual arms or head movements.

To sum up, the estimation for the mean values of the *Initiation of Approach* state is:

Distance	Speed	Gaze	Smile	DSal	CSal	HDip
7000	0	1	1	0	0	0

2. Distance Salutation

The average duration of this phase is, through Kendon's notes, around 0.3 seconds. Hence, on average, the model would only transit to the same state once and change to another state after it, which totals 2 transitions and a same-state transition probability of 0.5, following the previous methods.

The most probable following state is the *Head Dip*, as 25 out of the 50 DSs Kendon observed were followed by this movement.

Kendon states that, in some circumstances, a *Distance Salutation* occurs when the greeters "come into one another's presence", which describes a DS→FA transition, though it is more common to find an *Approach* phase after it. We consider that 80% of the DS phases happen during the APP. However, only 30% of these are not followed by an HD, as Kendon states it is much more common to find a head dip movement following a DS performed during the first stages of the approach. Considering these 2 factors and the usual 2 transitions from this phase, $P(APP|DS) = 0.8 * 0.3 * 0.5 = 3/25$

Going from the *Distance Salutation* to the *Close Salutation* phase with no approach in between is a very unlikely scenario, yet we assumed it to happen once in every 50 greetings. Also, a DS→IA transition is impossible in this model.

The probabilities for the transitions from the *Distance Salutation* state for this model, described in the last paragraphs, are the following:

IA	DS	HD	APP	FA	CS
0	1/2	1/4	3/25	3/25	1/100

Kendon analyzed that a DS is usually performed from 9 meters of distance until "few", which we considered 2, as it almost the end of the *Final Approach*. Hence, the average distance for this phase would be 5.5 meters. As a DS happens almost always during approaches, we believe the speed is similar to the approach, which appears to be around 1.2 meters per second, as this is a medium-to-slow average speed for a normal person walking.

According to Kendon, this is a phase where directly gazing tends to be very common, so, we estimated a value of $Gaze = 0.5$, which corresponds to a clear direct looking. Smiling is also very usual and one of the main characteristics. The given value, $Smile = 2$, can be described as a common smile.

There is, evidently, always a distance salutation movement in this phase, as this characterizes it.

The mean values for the observations in the *Distance Salutation* state were considered to be the following:

Distance	Speed	Gaze	Smile	DSal	CSal	HDip
5500	1200	0.5	2	1	0	0

3. *Head Dip*

Kendon describes a head dip movement as taking around 0.3 seconds, on average, which leads to the model changing to the same state once and switching to another one after it.

According to its description, the HD phase may only precede a DS if it happens during the first seconds of the approach movement. So, excluding the same-state, the HD→APP transition is the most likely.

Additionally, no HDs were observed by Kendon in the *Final Approach*, however, we may consider this transition to happen once in every 20 greeting sequences.

The other possible transitions from this state are not viable in this greeting scenario. Concluding, these are the transition probabilities from the *Head Dip* state, estimated for the Kendon model:

IA	DS	HD	APP	FA	CS
0	0	1/2	19/40	1/40	0

Given that the *Head Dip* happens mostly after *Distance Salutations* in the first moments of the approach, the average distance would have to be slightly smaller than the average distance of the IA phase.

Because of this phase happening during the approach movement, their average speed values were assumed to be similar.

By Kendon's analysis of the head movement which characterizes this phase, the head is not pointed towards the person, therefore there is no direct gaze during it. $Gaze = 1.5$ corresponds to this situation, as the gaze cone does not contain the robot's face. Smiling is also not a major characteristic of this phase, as there is no mention on the notes, so we chose the same neutral value of the IA phase.

In what respects to the movements features, the *Head Dip* phase is evidently characterized by its signature movement. These feature values are summarized in the following table:

Distance	Speed	Gaze	Smile	DSal	CSal	HDip
6500	1200	1.5	1	0	0	1

4. Approach

Taking into account the average distance walked and moving speed, as well as other phases that can happen during the approaching process, this phase lasts an average of around 3 seconds. Using the normal transition period, it would have to perform 15 transitions, where the last one would be to another state and, usually, there is another one to the *Distance Salutation* state, around 80% likely in a greeting sequence.

The last transition of this state is always to a *Final Approach* state, as Kendon assures this phase always happens before the end of the greeting (*Close Salutation* phase).

A transition to the IA is illogical, as this phase always precedes any approach, as well as for the CS transition and the HD transition.

Taking this into consideration, the transition probabilities for this state were estimated to be the following:

IA	DS	HD	APP	FA	CS
0	4/75	0	22/25	1/15	0

As stated before, this phase happens, usually, between 7 and 3 meters of distance to a target person, which gives an average distance of 5 meters. The speed feature was also already referred to be similar to the average speed for a medium-to-slow walker.

Kendon analyzed exhaustively mutual gaze in this phase and concluded that some people tend to look away as they approach the other, and they return to mutual gaze on the *Final Approach* phase. However, this normally occurs only in one of the greeters, as the other keeps the gaze. Therefore, Kendon considers a probability around 50% that each moment of the *Approach* phase has a direct looking, which corresponds to $Gaze = 1$.

In what concerns to the smiles, Kendon analyzes that, after the DS phase, smile rapidly fades in intensity, or even disappears completely, so we also considered a neutral face for this phase.

The relevant movements do not happen in this phase and, consequently, the estimation for the mean values on the observation features are:

Distance	Speed	Gaze	Smile	DSal	CSal	HDip
5000	1200	1	1	0	0	0

5. *Final Approach*

This phase tends to last around 0.9 seconds, meaning there is an average of 4 same-state transitions if there are no DS phases during it and 3 if there is one. This possible transition is given by a 20% likelihood of finding the salutation there, in opposition to the 80% of occurring during the *Approach* phase.

The last transition on this state is always to the *Close Salutation* state, since there is no possibility of transiting to any other.

The transition probabilities for this state of the model are, thus, the following:

IA	DS	HD	APP	FA	CS
0	1/25	0	0	19/25	1/5

According to Kendon, the *Final Approach* phase appears, usually, between 3 and 1.5 meters, which gives an average distance of 2.25 meters. The speed of the greeters appears to continue from the *Approach* phase.

On the other hand, gaze and smile are very common to be adjusted with the APP→FA transition, with people generally looking at each other and having smiles on their faces during this phase. In fact, only 8 out of 70 records showed no smile in Kendon's study. Therefore, the values for these two features were identical to the previously displayed *Distance Salutation*.

There are also no characteristic movements, opposing to the coming up phase. The observations' mean values for the *Final Approach* state are given by:

Distance	Speed	Gaze	Smile	Dsal	CSal	HDip
2250	1200	0.5	2	0	0	0

6. *Close Salutation*

After performing the *Close Salutation* phase, which has around 0.5 seconds of average duration, the sequence ends and the model is exited. Hence, in this phase, the same-state transition is the only possible transition, as stated in the following table:

IA	DS	HD	APP	FA	CS
0	0	0	0	0	1

Through Kendon's notes, a CS commonly takes place at a distance around 1.5 meters and it is a static phase, with both greeters standing face to face at a stable position.

People generally maintain the gaze and smile pose which they had previously, since the *Final Approach* is seen as a preparation for this phase.

The CS is also characterized by a close salutation movement, which always happens, though it may be as subtle as a small head lower.

The mean values for this state of the Kendon model are as follows:

Distance	Speed	Gaze	Smile	DSal	CSal	HDip
1500	0	0.5	2	0	1	0

The estimation of the overall state transition matrix (A) and the overall emissions' means matrix (M) for the Kendon model are, respectively, in the following matrices:

$$A = \begin{bmatrix} 2/3 & 1/30 & 0 & 1/5 & 1/15 & 1/30 \\ 0 & 1/2 & 1/4 & 3/25 & 3/25 & 1/100 \\ 0 & 0 & 1/2 & 19/40 & 1/40 & 0 \\ 0 & 4/75 & 0 & 22/25 & 1/15 & 0 \\ 0 & 1/25 & 0 & 0 & 19/25 & 1/5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 7000 & 0 & 1 & 1 & 0 & 0 & 0 \\ 5500 & 1200 & 0.5 & 2 & 1 & 0 & 0 \\ 6500 & 1200 & 1.5 & 1 & 0 & 0 & 1 \\ 5000 & 1200 & 1 & 1 & 0 & 0 & 0 \\ 2250 & 1200 & 0.5 & 2 & 0 & 0 & 0 \\ 1500 & 0 & 0.5 & 2 & 0 & 1 & 0 \end{bmatrix}$$

As for the initial state's probabilities, a greeting sequence of any individual must start by sighting the other, which, by Kendon's definitions, consists of the *Initiation of Approach* phase. Therefore:

$$\pi = [1 \ 0 \ 0 \ 0 \ 0 \ 0]$$

3.5 Training a Hidden Markov Model with the EM algorithm

After constructing an HMM by interpretation of Kendon's notes, we developed a model totally driven by data samples that provided the selected observations, which we named Data-Driven model. For this purpose, we used the Expectation-Maximization (EM) algorithm, mentioned in section 2.2, commonly used for HMM's third main problem, the Learning Problem. Firstly, we chose a set of greeting sequences and obtained manually the values of the observations for each sampling step, where each observation would be a vector of the form explained in section 3.3. Later, these sequences were joined, together

with the respective lengths, and served as inputs of the EM algorithm. This build, from them, the most likely Hidden Markov Model containing 6 states as requested.

3.5.1 Extracting video information

The datasets used were AVDIAR Dataset [40] and UoL 3D Social Interaction Dataset [41]. These contain several videos of humans greeting others in indoors environments, where it is possible to identify most of the phases of Kendon's greeting model.

The sequences extracted from the videos, helping to build this model are present in Table 3.2, as well as the greeting phases identified (Y) or not identified (N) in each one of them. From all the available videos, only a few of them were chosen, according mostly to two factors: length of the video (i.e., a good amount of observations extracted), and quantity of greeting phases identified. Each interaction resulted in two greeting sequences, one for each greeter (label 1 or 2 in the mentioned table).

AVDIAR Dataset

This dataset contained videos of people interacting with others in several environments, however, we opted to use videos from just one environment, as it suited best our objectives. The chosen environment was a small room (see Figure 3.10), where two people met and started a conversation, with the exception of one of the videos, where, in fact, four people met. Usually, before the conversation, these two people greeted each other, which is the section of the videos we were interested in.

These videos were all recorded by two parallel cameras, mounted 20 centimeters apart, delivering stereoscopic images that allow a 3D comprehension of the environment, almost representing a human-eye view of the scenes. The stereo calibration was performed and the results provided in a public file.



Figure 3.10: Sample of the environment in AVDIAR dataset's greetings

Apart from the stereo calibration, every video was also provided with a document, which had the

ground truth position data of 18 joints from each person present. The data was available for each frame the person appeared on the video, at a 25 frame per second rate.

However, this position data was given in a 2D format, in pixel values. Since we were interested in the real distance between the two people, the 3D information, in meter units, was necessary.

In order to get depth information, we needed to use the stereo information. Firstly, we extracted the left-camera and right-camera images of the frame we wanted to compute the depth. Having these, we acquired the disparity map, by using a specific function of the OpenCV Python library.

A disparity map for a pair of rectified stereo images is an image where each pixel designates the distance between the pixel in image one and its matching pixel in image two. We could move directly to this step since the images given were already rectified. There are several algorithms to obtain the disparity from two images. The OpenCV function used (`StereoSGBM.compute`) uses the Semi-Global Block Matching algorithm [42].

For transforming a disparity map so that we can obtain depth values, we needed the disparity-to-depth Q matrix, which had already been provided in the stereo calibration file.

To obtain the depth values, we used another OpenCV function (`reprojectImageTo3D`). This function takes the disparity map and the Q matrix, and outputs a 3D point for each pixel of the original image, which has 3D values in millimeters, according to the camera's coordinate frame. For each pixel (x,y) and the corresponding disparity, the function computes the two following equations:

$$\begin{bmatrix} X & Y & Z & W \end{bmatrix} = Q \begin{bmatrix} x & y & disparity(x,y) & 1 \end{bmatrix} \quad (3.12)$$

$$3D_image(x,y) = (X/W, Y/W, Z/W) \quad (3.13)$$

Where $3D_image(x,y)$ contains the 3D information of each (x,y) pixel.

Knowing the distance to the camera of each pixel in every frame of the videos, and the position of each face in the 2D image (through the ground truth document), we could compute the distance between the two greeters for each frame.

Assuming a Person 1 with face position $p_1 = (x_1, y_1, z_1)$ and a Person 2 at $p_2 = (x_2, y_2, z_2)$, the distance between both is given by (3.14), where $p'_1 = (x_1, z_1)$ and $p'_2 = (x_2, z_2)$, as we ignored the height (Y-axis value) difference, such as in section 3.3.1.

$$Distance = \|p'_1 - p'_2\| \quad (3.14)$$

Having the distance between people in all video frames and the time interval between frames (1/25 seconds), the speed observation could be computed for each person. Analogously to the method present on section 3.3.2, we assumed a Person 1, with a position $p'_1 = (x_1, z_1)$ and a previous frame position of $pp'_1 = (px_1, pz_1)$, ignoring the heights; and a Person 2, with a position $p'_2 = (x_2, z_2)$. Then, we computed (3.14) to (3.16), to execute the same logic described previously in Figure 3.4.

$$PrevDistance = \|pp'_1 - p'_2\| \quad (3.15)$$

$$Speed = \frac{PrevDistance - Distance}{\Delta t} = \frac{PrevDistance - Distance}{1/25} \quad (3.16)$$

As stated before, our HMM was developed to receive observation vectors 0.2 seconds apart. To train the model correctly for this, the training observations also needed to have this time gap, however, these had 0.04 seconds apart. To correct this, we grouped them in groups of 5.

The first approach was to average the distance and speed values in these groups of 5 observations, however, after a short analysis, we noted the features had a few outlier values. This was due to the lack of accuracy of the disparity map, which depended on several parameters of the Block-Matching algorithm. Hence, we opted by using the median value of each group of 5, also ensuring intervals of 0.2 seconds. To avoid unknown speed values on the first observation of the videos, we set the observations to begin at 0.1 seconds and maintained the desired interval from there.

After this, in order to eliminate most of the outliers still present in the data, we opted to perform data interpolations on these values of distance. Some speed values were, thus, changed according to the new distances.

Since no more information was provided, the rest of the features were observed and labeled manually. The following process was performed for gaze direction and smiling labeling:

1. Construction of an adequate scale for the observation values;
2. Extraction of video frames with an interval of 0.2 seconds, through a script;
3. Image analysis and labeling of the 2 features, according to the scale;
4. Changing to the next video.

For the head and arm movements, the task was simpler. The person labeling identified a distance salutation, head dip, or close salutation movement with 100% likelihood. Hence, the values on these three features were restricted to 0 or 1.

UoL 3D Social Interaction Dataset

The second dataset used contained more greetings than the first, however, most of them were not used. In each video, the same two people could greet more than one time, sometimes turning to each other at a very short distance (1.5 meters or less) and simply shook hands, which does not provide much information as a greeting sequence. A frame of one of this dataset's videos can be seen in Figure 3.11, for environmental context.

As a source of information, the videos came with a "skeleton file", which had the 3D position and orientation of several joints from each person, on the camera coordinate frame. This file also had the time stamp of each sample, according to the beginning of the video.

To compute the distance between the two greeters, the method was identical to the one used on the first dataset. We extracted the position of each person's head joint and used (3.14), saving the timestamps of all extractions.



Figure 3.11: Sample of UoLs Dataset environment for greetings

Speed computation was also similar, extracting both positions and one previous position, to calculate the speed of the desired sample, even though the Δt firstly used was not 0.2 seconds.

The next step was to ensure the 0.2 seconds difference between each observation. Opposing to the previous dataset, here there were no relevant cases of outlier values on the distance computation. Hence, intervals with the correct duration were created, and the distance and speed values computed by averaging the ones inside each interval.

Unlike in the first dataset, having orientation values of each face (in Quaternion format, on the camera coordinate frame) allowed us to automatically estimate gaze direction. This was performed with the assumption that head orientation is a good measure for the gaze direction, in a procedure similar to the second method described on section 3.3.3.

Having the orientation and position values for each person, the method used consisted on the following steps:

1. Convert orientation of Person 1 (camera coordinate frame) to the Person 2 coordinate frame;
2. Convert position of Person 1 (camera frame) to the Person 2 frame;
3. Use algorithm *Gaze Detector 2* from section 3.3.3 where Person 2 is the robot and Person 1 is the target person;
4. Repeat steps 1 to 3 switching Person 1 and 2.

After executing this method on all the necessary samples, the generated gaze values were reviewed and slightly corrected. We detected a few mistakes, which had been created due to some inconsistency on the orientation data, and swept them with labeled values.

Smile and head/arm characteristic movements were later manually labeled, following the same method and the same scales used on the first dataset.

In Table 3.1 we compare the approaches taken on both datasets which allowed to extract one sequence of observation vectors with 7 fields, for each greeting sequence, and respecting the norms previously defined for the HMM.

	AVDIAR	UoL
Available data format	2D	3D
2D-3D conversion using stereo images	YES	NO
Distance and Speed extraction	Algorithms using 3D position	Algorithms using 3D position
Value aggregation to ensure 0.2 seconds	Median	Average
Outlier interpolation	YES	NO
Gaze extraction	Labeling	Algorithm using head orientation
Smile and movements extraction	Labeling	Labeling

Table 3.1: Comparison of the observations' extraction approaches for the 2 Datasets

3.5.2 Limitations of the videos

As we can infer from Table 3.2, there were some limitations in the videos, in what concerns the content of the sequences.

First of all, it is possible to see that many sequences miss the *Initiation of Approach* phase, which should be the start to a normal greeting sequence. This happens due to some videos starting only when the two greeters are already approaching each other.

Another visible problem, comparing to Kendon's description of the phases is the lack of head dips, as less than 10% of people performed such a movement. As Kendon considers this phase to be a way of ending other involvements and paying full attention to the other, this may be the reason behind this limitation. In both datasets, in the greeting sequences filmed, people are already expecting the greeting and, though they try to act naturally, the surprise is an important feeling for an action as spontaneous as the *Head Dip*. Additionally, this movement tends to happen in the beginning of long approaches. Therefore, the short environment was not helpful as well.

A third limitation is found in the APP column. The indoors spaces utilized for filming were small and did not allow to have greeting sequences as complete as the ones studied and described by Kendon. Most sequences started with greeters at a distance from 2 to 3 meters away, where they were already preparing for the *Close Salutation*. For this reason, the approaching movements were shorter than expected, which led to a low share of sequences with the *Approach* phase. It occurred only in 10 out of the 33 sequences (30.3%).

3.5.3 Training and Test sets

Having 33 greeting sequences, we split them into a train and a test set. The train set was used as the input for the EM algorithm, while the test set served, posteriorly, to measure the model's quality.

Dataset	Greeting	IA	DS	HD	APP	FA	CS	Set
UoL 3D Social Interaction	Session 1 - 1st G (1)	Y	Y	N	N	Y	Y	Train
	Session 1 - 1st G (2)	Y	Y	N	N	Y	Y	Train
	Session 1 - 3rd G (1)	Y	Y	N	N	Y	Y	Train
	Session 1 - 3rd G (2)	Y	Y	N	N	Y	Y	Train
	Session 1 - 5th G (1)	Y	Y	N	N	Y	Y	Train
	Session 1 - 5th G (2)	Y	Y	N	N	Y	Y	Train
	Session 2 - 1st G (1)	Y	Y	N	N	Y	Y	Train
	Session 2- 1st G (2)	Y	N	N	N	Y	Y	Train
	Session 2 - 2nd G (1)	Y	N	N	N	Y	Y	Train
	Session 2 - 2nd G (2)	Y	N	N	N	Y	Y	Train
	Session 3 - 1st (1)	Y	Y	N	N	Y	Y	Train
	Session 3 - 3rd (1)	Y	Y	N	N	Y	Y	Train
	Session 3 - 3rd (2)	Y	Y	N	N	Y	Y	Train
	Session 4 - 1st (1)	Y	Y	N	N	Y	Y	Train
	Session 4-1st (2)	Y	Y	Y	N	Y	Y	Train
	Session 5 - 3rd (1)	Y	Y	Y	Y	Y	Y	Train
	Session 5 - 3rd (2)	Y	Y	N	Y	Y	Y	Train
	AVDIAR	S02-01 (1)	N	N	N	N	Y	Y
S02-01 (2)		N	N	N	N	Y	Y	Test
S02-02 (1)		N	Y	N	Y	Y	Y	Test
S02-02 (2)		N	N	N	Y	Y	Y	Test
S02-03 (1)		N	Y	N	N	Y	Y	Test
S02-03 (2)		N	N	N	N	Y	Y	Test
S04-01-left (1)		N	N	N	Y	Y	Y	Test
S04-01-left (2)		Y	N	N	N	Y	Y	Test
S04-01-right (1)		N	N	N	Y	Y	Y	Train
S04-01-right (2)		Y	N	N	N	Y	Y	Train
S05-01 (1)		Y	N	N	N	Y	Y	Train
S05-01 (2)		N	N	N	Y	Y	Y	Train
S05-02 (1)		Y	Y	Y	N	Y	Y	Train
S05-02 (2)		N	N	N	Y	Y	Y	Train
S05-05 (1)		Y	Y	N	Y	Y	Y	Train
S05-05 (2)		N	N	N	Y	Y	Y	Train
Total	33	22/33	18/33	3/33	10/33	33/33	33/33	

Table 3.2: Summary of greetings' phases in the used Datasets

Firstly, the greeting sequences were manually split, to obtain the most accurate model possible, according to Kendon's notes. The sequences which provided the most observations and more complete information, having more greeting phases, were chosen to be the train sequences, while the remaining ones would belong to the test. Each sequence's set is referred to in Table 3.2.

Thus, using the above-mentioned training set and the EM algorithm (using the Python library "hmm-learn"), we estimated the parameters for a model several times, given randomized initializations. For each of these models, the algorithm stopped the iteration process when it assumed convergence had been reached, given by the gain in log-likelihood dropping below 0.01; or divergence had been reached, by the number of iterations exceeding 100. In most cases, the model obtained was the same, as the algorithm kept finding the same minimum. This was also the model with the highest accuracy on the test set, so, we opted to use it for further testing.

The HMM generated is presented below with its transition matrix (A), means' emission matrix (M), and initial state probability matrix (π). As the EM is an unsupervised algorithm, the rows and columns in the matrices did not necessarily represent the 6 states observed by Kendon, but rather 6 states clustered

by the algorithm. Thus, these states had to be labeled depending on the similarity with the real ones. The matrices presented below were already organized, so that each state is positioned as its most similar state on the Kendon model's matrices. The following order was used: *Initiation of Approach* (IA), *Distance Salutation* (DS), *Head Dip* (HD), *Approach* (APP), *Final Approach* (FA), *Close Salutation* (CS).

$$M = \begin{bmatrix} 2068 & 99 & 1.47 & 1.33 & 0 & 0 & 0 \\ 1617 & 142 & 0.80 & 1.89 & 1 & 0 & 0 \\ 1635 & 256 & 1.26 & 1.40 & 0 & 0 & 1 \\ 2771 & 1896 & 0.97 & 1.77 & 0 & 0 & 0 \\ 1569 & 389 & 0.68 & 1.73 & 0 & 0 & 0 \\ 1083 & 131 & 0.57 & 1.91 & 0 & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.608 & 0.144 & 0 & 0.030 & 0.218 & 0 \\ 0 & 0.625 & 0.075 & 0 & 0.300 & 0 \\ 0 & 0 & 0.400 & 0.198 & 0.402 & 0 \\ 0.066 & 0 & 0 & 0.631 & 0.303 & 0 \\ 0 & 0.051 & 0 & 0 & 0.673 & 0.277 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\pi = [0.929 \quad 0 \quad 0 \quad 0.071 \quad 0 \quad 0]$$

3.5.4 Comparison with Kendon Model

Given that one of the main objectives of the Kendon model is to validate the Data-Driven Model obtained by analysis of real greetings, the matrices of both models must describe similar situations. This comparison is emphasized by the three below matrices, which contain the values from the Data-Driven Model in blue, and the respective ones from the Kendon model, in red.

$$\pi = [0.929/1 \quad 0/0 \quad 0/0 \quad 0.071/0 \quad 0/0 \quad 0/0]$$

$$A = \begin{bmatrix} 0.608/0.667 & 0.144/0.033 & 0/0 & 0.030/0.200 & 0.218/0.067 & 0/0.033 \\ 0/0 & 0.625/0.500 & 0.075/0.250 & 0/0.120 & 0.300/0.120 & 0/0.001 \\ 0/0 & 0/0 & 0.400/0.500 & 0.198/0.475 & 0.402/0.025 & 0/0 \\ 0.066/0 & 0/0.053 & 0/0 & 0.631/0.880 & 0.303/0.067 & 0/0 \\ 0/0 & 0.051/0.040 & 0/0 & 0/0 & 0.673/0.760 & 0.277/0.200 \\ 0/0 & 0/0 & 0/0 & 0/0 & 0/0 & 1/1 \end{bmatrix}$$

$$M = \begin{bmatrix} 2068/7000 & 99/0 & 1.47/1.00 & 1.33/1.00 & 0/0 & 0/0 & 0/0 \\ 1617/5500 & 142/1200 & 0.80/0.50 & 1.89/2.00 & 1/1 & 0/0 & 0/0 \\ 1635/6500 & 256/1200 & 1.26/1.50 & 1.40/1.00 & 0/0 & 0/0 & 1/1 \\ 2771/5000 & 1896/1200 & 0.97/1.00 & 1.77/1.00 & 0/0 & 0/0 & 0/0 \\ 1569/2250 & 389/1200 & 0.68/0.50 & 1.73/2.00 & 0/0 & 0/0 & 0/0 \\ 1083/1500 & 131/0 & 0.57/0.50 & 1.91/2.00 & 0/0 & 1/1 & 0/0 \end{bmatrix}$$

Most of the differences between the two models originated from the environmental differences between the two information sources, as explained in section 3.5.2. This provoked some discrepancies between the two models, mainly on the Distance and Speed observations. For instance, the average distance of the IA phase is around 7 meters on the Kendon model, while only around 2 meters on the Data-Driven Model. However, the rest of the features and the transition probability and initial probability matrices suggests that this state was well identified by the algorithm.

Another relevant difference between the models appears on the *Approach* and *Final Approach* states, on the transition probabilities to these two states. Again, this is a repercussion of the environmental differences and scarcity of *Approach* phases. For instance, the IA→FA transition is much more probable on the Data-Driven Model than in the Kendon model, while the IA→APP transition is less probable, due to not happening much in the videos. Despite this, these differences are expected characteristics for a model developed essentially for greetings in small environments.

The initial probability matrices π are both very similar as well. On Kendon's model, a greeting sequence starts with a probability of 100% on the *Initiation of Approach* state, while on the Data-Driven Model, this probability is around 93%. The other possible state to be the initial is the *Approach*, with a 7% probability. This is due to some videos starting already during the greeting sequences.

Despite the disparities on some values, the Data-Driven Model, in fact, predicted six phases very similar to the theoretical description of the six original ones. State 0 represented an almost static phase and more distant than most of the other states, where people are not usual to look directly, just as in the IA phase in Kendon's notes. State 1 had a distance salutation movement, together with direct gaze and smiles. State 2 also resembled the HD phase, with the characteristic movement, the aversion of gaze, and a distance similar to State 1. State 3 and 4 represent two phases with a high speed, differing by a smaller distance, and more likely direct gaze, just as in the APP and FA comparison. State 5 also represented well a CS phase, with the characteristic movement, likely direct gaze and smiling.

3.6 Testing the model

3.6.1 Using Test Sequences

After a superficial validation by comparing the matrices resulted from the model training with the ones inferred from Kendon's description of the sequence, our Hidden Markov Model was tested with the 8 greeting sequences, containing sequences of observations with intervals of 0.2 seconds, chosen for the test set. The two metrics used for testing were the accuracy of state labels, i.e., the percentage of states

the model predicted correctly in these sequences, and the confusion matrix. The accuracy measure was divided into 2 types of prediction: state labels predicted using the Forward algorithm and using the Viterbi algorithm.

The main difference between these two predictive algorithms, as introduced in section 2.2, is that while the forward algorithm receives a sequence of observations and calculates, iteratively, the probability of each state for each index on the sequence, the Viterbi algorithm computes directly the most probable state path that corresponds to a sequence of observations. This difference makes the forward algorithm the only one capable to predict the states in real-time situations, as we will not obtain entire sequences but, instead, one observation at a time.

For the Viterbi algorithm, Python library "hmmlearn" also provided a function for Gaussian HMMs, while the forward needed to be developed, as presented in Algorithm 1.

Algorithm 1 Forward algorithm for a Gaussian HMM

```

1: function FORWARDITERATION(observation, pi, alpha, model, iteration)
2:    $\mathbf{y} \leftarrow \text{ones}(\text{NumberOfStates})$ 
3:   for  $i < \text{NumberOfStates}$  do
4:     for  $j < \text{NumberOfFeatures}$  do
5:        $P \leftarrow \text{PDF}(\text{observation}(j), \text{model.means}(i, j), \sqrt{\text{model.covars}(i, j, j)})$   $\triangleright$  Probability of the
         observation value given the Gaussian with  $\mu$  and  $\sigma$  known
6:        $\mathbf{y}(i) \leftarrow \mathbf{y}(i) * P$ 
7:     end for
8:   end for
9:    $\mathbf{y} \leftarrow \frac{\mathbf{y}}{\sum \mathbf{y}}$ 
10:  if any  $\mathbf{y}(i) = 0$  then
11:     $\mathbf{y}(i) \leftarrow 10^{-100}$ 
12:  end if
13:   $\mathbf{y} \leftarrow \frac{\mathbf{y}}{\sum \mathbf{y}}$   $\triangleright$  variable which contains probability of each state given observation
14:   $\mathbf{D} \leftarrow \text{diag}(\mathbf{y})$ 
15:  if  $\text{iteration} = 1$  then
16:     $\alpha \leftarrow \mathbf{D} * \mathbf{pi}$ 
17:  else
18:     $\alpha \leftarrow \mathbf{D} * \text{model.transmat}^T * \alpha$ 
19:  end if
20:   $\alpha \leftarrow \frac{\alpha}{\sum \alpha}$ 
21:  return  $\alpha$ 
22: end function
23:  $\alpha \leftarrow \pi$ 
24:  $\text{iteration} \leftarrow 1$ 
25: while Sequence not finished do
26:    $\alpha \leftarrow \text{FORWARDITERATION}(\mathbf{O}, \pi, \alpha, \text{model}, \text{iteration})$ 
27:    $\text{PredictedState} \leftarrow \underset{i}{\text{argmax}} \alpha(i)$ 
28:    $\text{iteration} ++$ 
29: end while

```

Table 3.3 expresses the accuracy with both predicting methods, comparing to the values of the Kendon model on the same greeting sequences.

The results represent higher accuracy on predictions (with both algorithms) for the Data-Driven Model, when comparing it to the model formed based on Kendon's analysis, which would be presumable. Even though all sequences are different, the environment and the greeters from the train and

test sequences are the same, which is seriously relevant, considering the already stated environmental differences and some patterns which the same people keep from sequence to sequence.

On our model, the significant accuracy difference (5.38%) between the predictions of the two algorithms was also slightly expected. In the case of complete sequences, the Viterbi algorithm is, in most cases, a more viable option for predicting than the forward algorithm, since it analyzes the entire sequence of observations, as explained before.

	Forward Algorithm	Viterbi Algorithm
Data-Driven Model	0.839	0.893
Kendon Model	0.785	0.774

Table 3.3: Accuracy of the two models on the test set

To analyze deeper the predictions using the Viterbi algorithm on the 8 test sequences, Figure 3.12 and Figure 3.13 compare the predicted labels with the real labels for each sequence, while Table 3.4 is the confusion matrix obtained.

	IA	DS	HD	APP	FA	CS	Real Total
IA	1	0	0	0	0	0	1
DS	0	0	0	2	0	0	2
HD	0	0	0	0	0	0	0
APP	1	0	0	8	3	0	12
FA	2	0	0	2	47	0	51
CS	0	0	0	0	0	27	27
Predicted Total	4	0	0	12	50	27	93

Table 3.4: Confusion Matrix of the chosen model on the test set

The confusion matrix demonstrates that a relevant part (50%) of the incorrect labels was originated by some confounding between the *Approach* and *Final Approach* states (4 and 5 in the graphs). These mistakes were predictable, as we had already concluded the model had some problems identifying these two states as Kendon did. This was mostly due to the lack of information on the videos and the similarity of their characteristics, which sometimes causes difficulties identifying the transition points between the two, even for Kendon.

Despite this, the fact that the *Approach* state provided the worst results (excluding the DS state, which only had 2 instances on the testing, not containing very reliable results), but still had an accuracy of 66,7% is revealing of the good performance of the model.

The 8 graphs tracking the difference between prediction and real state during the entire sequences also provide positive information, since they demonstrate that one prediction error does not tend to escalate to more. From the 8 sequences, there was only one example where more than one error happened consecutively.

In order to check the robustness of our model and greeting sequences, we created 15 train-test splits, different from the split used before, though with the same train-test percentage: 76% train (25 sequences) and 24% test (8 sequences). Using these splits, we calculated the mean and the standard deviation of the accuracy calculated both with the forward algorithm and with the Viterbi algorithm on the

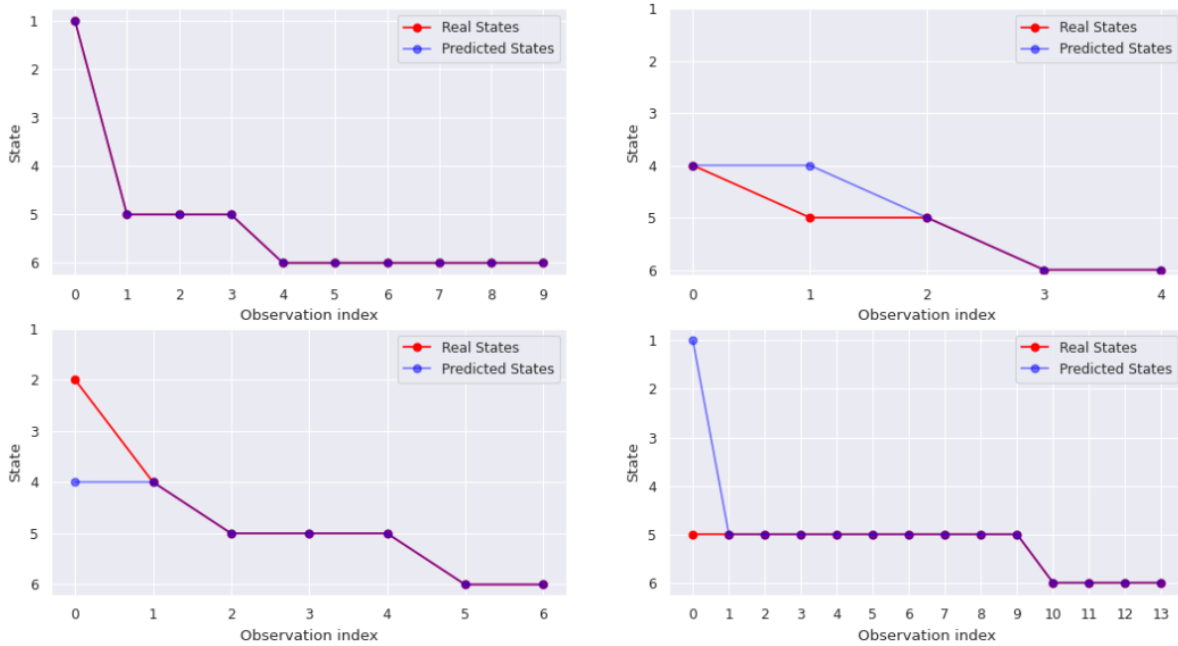


Figure 3.12: First 4 test sequences predicted with the Viterbi algorithm

test set. Additionally, these values were compared with the same values for the Kendon model on the same 15 test sets. The results are in Table 3.5 on the format mean +/- standard deviation.

	Forward Algorithm	Viterbi Algorithm
Data-Driven Model	0.780 +/- 0.132	0.801 +/- 0.115
Kendon Model	0.810 +/- 0.058	0.823 +/- 0.051

Table 3.5: Accuracy of the two models on 15 different train and test sets

Unlike the results on the initially trained model, using several train and test sets produced an accuracy (on average) slightly lower than Kendon’s model and with higher values of standard deviation.

These values were expected, as we are using different train sets and, consequently, different models in each of the 15 experiments. Some models represented accurately the six states, as the initial model. However, since the number of training sequences was not significantly high, a few of these models did not represent the greeting phases as Kendon described, and were not accurate in labeling the test sequences. This discrepancy between models that could find six states similarly to the original model, and those that could not, resulted in a slightly lower average accuracy and a high standard deviation. Also, to be consistent with the previous training, the parameter initialization was random, which may have not generated the best model for every case. To analyze an example of one low-accuracy model, Table 3.6 represents the confusion matrix of a model which had around 68,4% of accuracy, using the Viterbi algorithm.

By this confusion matrix, it is possible to conclude that the states considered most similar to the definition of the *Head Dip* and *Approach* states were not sufficiently similar, having 0 of 4 and 10 of 28 correct labels, respectively. This was, presumably, due to most of these states’ labels being on the test set, therefore, the train set did not have much information on these states and could not train a

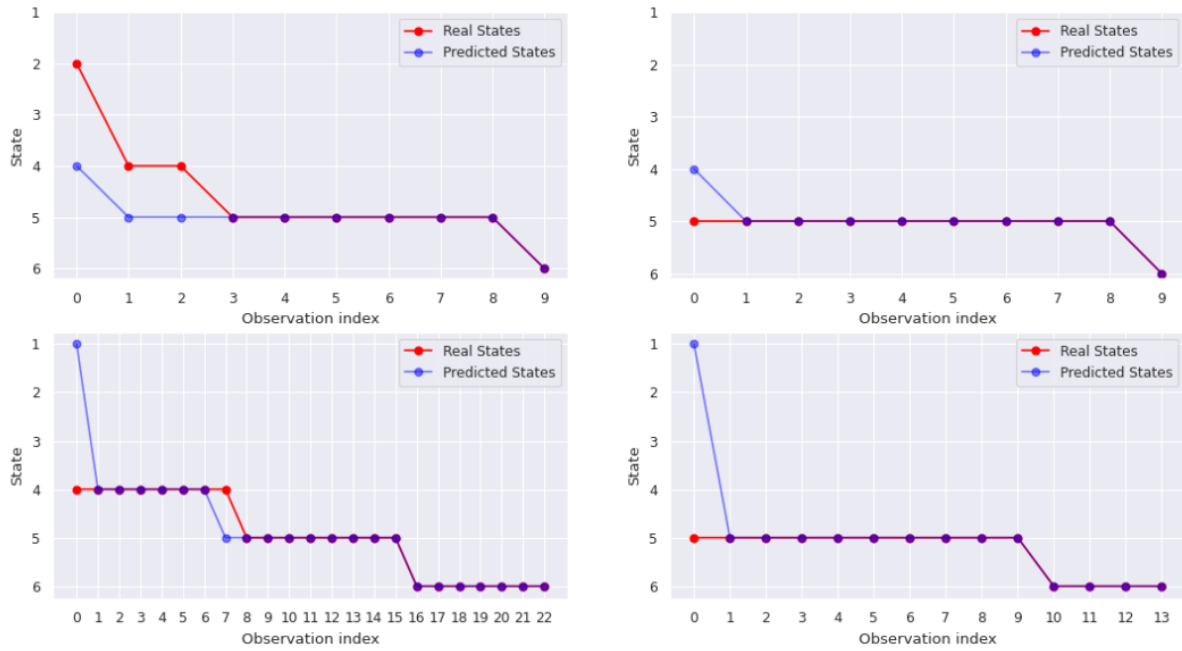


Figure 3.13: Last 4 test sequences predicted with the Viterbi algorithm

proper model. The IA and APP states were responsible for most of the low accuracy models, which was expected, as they are the two less present states in our sequences, as examined in Table 3.2.

	IA	DS	HD	APP	FA	CS	Real Total
IA	7	0	0	0	0	0	7
DS	0	8	0	0	0	0	8
HD	0	0	0	4	0	0	4
APP	17	0	0	10	1	0	28
FA	8	0	0	7	21	0	36
CS	0	0	0	0	0	34	34
Predicted Total	32	8	0	21	22	34	117

Table 3.6: Confusion Matrix of one low-accuracy model

To finish this testing section, we may conclude that the results were positive, even though our model would benefit considerably by having more training data. The latter results demonstrate a standard deviation much higher than the results from the Kendon model, given by the discrepancies between models trained with different training sets. With a higher quantity of greeting data, we predict the model would stabilize similarly to the first trained model, providing a smaller standard deviation and an average accuracy that should exceed the Kendon Model.

3.6.2 Vizzy

Vizzy [7] is a humanoid-like robot developed by the Institute for Systems and Robotics (ISR) for assistive robotics and was used for the testing of our system. It was designed combining a friendly and organic approach, with an upper humanoid-like torso and a large wheeled platform for locomotion. Its appearance can be seen in Figure 3.14.

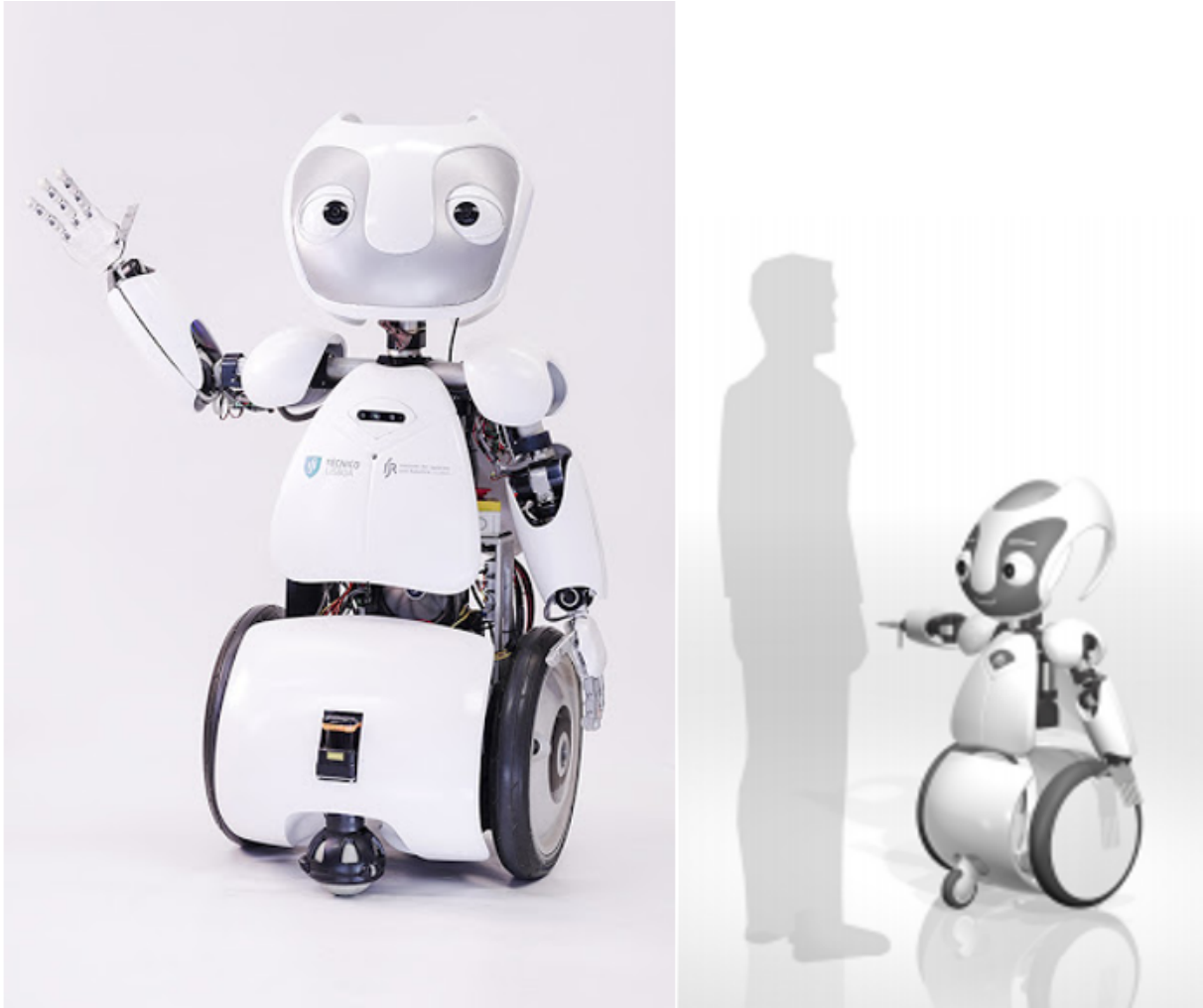


Figure 3.14: Left: Vizzy waving; Right: Vizzy's size comparing with a 1.75 m person

On the mechanical part, Vizzy has 30 degrees of freedom: 2 for the mobile base, 1 for the waist, 8 for each arm, 3 for each hand, and 5 for the head. Each dof has its own motor module. The battery, when fully charged, supports around 4 hours of continuous use of the PC and torso motors.

Several initiatives had already developed skills on Vizzy at the starting point of this work, using two different middlewares: ROS and Yet Another Robot Platform (YARP) [43]. These skills include: reaching and grasping for simple shape objects, 3D face detection (position and orientation), localization and autonomous navigation in a known map, arm gestures like handshake, waving, arm stretching, pick objects and drop objects, head control for a 3D fixation point (gaze) and speaking.

3.6.3 Simulator

As a means of testing parts of the work, we used Rviz [44] and Gazebo [45], 3D visualization and interaction tools for ROS. Previous works on Vizzy had already set a comfortable environment for testing the robot's movements and behaviors. This environment included a map of the 7th floor of the North Tower of IST, favorable for testing, as it represented specifically Vizzy's real environment, and a replica of the robot, at the same scale and functioning with a setup identical to the real one.

Rviz included an interactive interface that allowed to send various tasks to Vizzy, including navigation goals, arm movements, and gaze changing through a clickable image. Gazebo's environment was mostly used for superior visualization, as it contained the entire 3D rendered map (see Figure 3.15).



Figure 3.15: Map of the simulation environment in Gazebo

The environment set also included a previous work which created several Gazebo models to simulate people that were crucial for the simulations. The pose information from these was constantly available, through ground-truth data, from which was created a fake face detector for the robot.

3.6.4 Testing on Real-Time Situations

After testing with greeting sequences from the same source as the ones which trained the model, we moved the experiments to the simulation environment, described in the previous section. Since the previous testing was performed with sequences from the same sources as the testing, experimenting in other environments was necessary for robustness.

In addition, we also wanted to confirm the model's ability to change state correctly in real-time situations, opposing to the tests with entire sequences in previous sections. In other words, given real-time observations from the person simulated at each time stamp, the model would use the forward algorithm to calculate continuously a most probable state.

Using one person created for simulations and the model of Vizzy, we could approximate a greeting situation. The script computed to create the artificial people also generated their ground-truth information, publishing it to a ROS topic called `"/people"`, on the form of a `PoseArray` message, where each `Pose` had the following format:

- A position (of type `Point`), (X,Y,Z) ;
- An orientation (of type `Quaternion`), (x,y,z,w) .

Given this information, a ROS node called *PeopleExtractor* computed observation vectors of the target person every 0.2 seconds. For simplicity, the smiling and gaze features were considered to be constant inside a simulation. The detection of movements was replicated by a keyboard input detecting one of the 3 movements with 100% probability, depending on the key pressed.

In order to move a person toward the robot, we built an algorithm using a ROS service named `/gazebo/set_model_state`. This ROS service is used to move Gazebo models to certain positions, based on the name of the model given. By making one fictional person move little from one spot to another multiple times and wait a small amount of time in each spot, it would simulate a continuous walking. This method is explained in Algorithm 2

Algorithm 2 An algorithm for simulating a walking model in Gazebo

```

1: function MOVEPEOPLE( $xInitial, xFinal, yInitial, yFinal, Speed$ )
2:    $StateMsg \leftarrow ModelState()$  ▷ Using the correct message type
3:    $StateMsg.model\_name \leftarrow 'pessoa1'$  ▷ Using the correct Gazebo model
4:    $Distance \leftarrow \sqrt{(xFinal - xInitial)^2 + (yFinal - yInitial)^2}$ 
5:    $NumSteps \leftarrow 10 * (Distance/Speed)$  ▷ Number of steps given
6:    $\mathbf{x} \leftarrow linspace(xInitial, xFinal, NumSteps)$  ▷ Vector with  $NumSteps$  equispaced values, which
   begins in  $xInitial$  and ends in  $xFinal$ 
7:    $\mathbf{y} \leftarrow linspace(yInitial, yFinal, NumSteps)$ 
8:   for  $i \leftarrow 0; i < NumSteps; i++$  do
9:      $pause(0.1)$  ▷ Pausing for 0.1 seconds in each step
10:     $StateMsg.pose.position.x \leftarrow \mathbf{x}(i)$ 
11:     $StateMsg.pose.position.y \leftarrow \mathbf{y}(i)$ 
12:     $StateMsg.pose.position.z \leftarrow 0$ 
13:    call ROS service /gazebo/set_model_state ▷ Move the person to the position given in
    $StateMsg$ 
14:   end for
15: end function

```

Experiment	IA	DS	HD	APP	FA	CS
1	Y	Y	Y	Y	Y	Y
2	Y	N	N	Y	Y	Y
3	Y	Y	N	Y	Y	Y
4	Y	N	N	N	Y	Y
5	Y	Y	Y	Y	Y	Y
6	Y	Y	Y	Y	Y	Y

Table 3.7: States of the model present (Y) or not present (N) for each experiment

Regarding the experiments, six different greeting situations were planned, in order to confirm that our model was flexible enough to predict the same states correctly in different circumstances. The observation sequences were later saved and labeled, for further accuracy computation. In these experiments the person's (a woman) intention of greeting was assumed, even without the robot making any movement, for all cases and during the entire sequences.

Table 3.7 summarizes the six states simulated, that were present (Y) or not (N) in the labels of each of the six experiments.

For the first experiment, we intended to reproduce a common greeting, having the six states of the model. The woman started around 5.5 meters away from the robot, with a frontal orientation, simulating

a brief *Initiation of Approach* before starting the movement. After walking 1.5 meters, she displayed a *Distance Salutation* (simulated by pressing a key that added the respective observation feature), followed by a *Head Dip*. The movement continued until the person stopped, 1 meter away from Vizzy and performed the *Close Salutation* phase. As mentioned before, the gaze and smile values were kept constant during the entire sequence, with $Gaze = 1$ (average between the direct looking and not), and $Smile = 1.5$ (face with a slight smile) being the values from experiment 1 to 4. Retrieving the necessary information for the 5 features, the HMM predicted states, which were compared with the labels, achieving 95.7% of accuracy for this sequence (24 of 25 labels), as can be seen in Table 3.8. The only mistake was due to confounding APP and FA, as this transition becomes harder to detect without gaze and smile changing.

For the second experiment we used a simpler situation. The simulated woman also started 6 meters away and moved in the direction of Vizzy, as in the first test, without displaying any other movement, to perform a *Close Salutation* as she stopped 1 meter away. In this case, 88.9% of the labels were well predicted, with the HMM reacting well to a sequence with only 4 states. The mistakes were, again, originated from the two approach phases, which is not a major situation, since it is very related to the constant values of gaze and smile.

On the third experiment, we tested a greeting with the normal approach movement, but a DS movement displayed very close to the robot, which is, by Kendon, a situation where the HD is usually not found. The CS, then, finished the sequence. The HMM predicted this sequence perfectly (100%), as the model could detect the DS state in a position different from the other experiments and make a correct transition from it.

Experiment 4 reproduced another possible greeting situation. Here, the sighting was made at a shorter distance (around 2 meters), therefore, the approach movement was much smaller than usual and the DS was not present, neither was an HD. The CS ended the sequence, as in every case. As can be seen in the table, the HMM managed another perfect state prediction for this situation.

On experiment 5 and 6 we modified the values kept constant until then to ensure the predictions continue accurate. Firstly, we set $Smile = 2.5$, which corresponds to a clear smile that would be present the entire sequence and reproduced a situation similar to experiment 1. The HMM responded accordingly, predicting all the 32 states correctly. On the last experiment, we set a clear direct gaze ($Gaze = 0.4$) for the entire sequence, while the smile values were changed to the original ones. As can be seen in the table, this sequence had the lowest accuracy value of the 6 experiments, hitting only 73.3% of the labels. The mistakes, here, were mostly concentrated on the IA and APP phase, since these are two phases which are not characterized by a direct gaze and can be confounded with opposite circumstances. These results also suggest that the gaze feature is more relevant for state prediction than the smiling. However, it is not usual that a person is constantly looking directly at the face from the moment the robot sights him/her, therefore, this situation may be slightly exaggerated.

The six testing situations provided an average accuracy of nearly 92% (which rises to almost 97% if we exclude the least successful). These also included three perfectly predicted sequences. Despite the environment having changed and testing several situations, the obtained results were even higher

than in the previous testing, which was probably due to having less noise in the observations (ground truth position data and constant gaze and smile values). Consequently, these experiments demonstrate that, for reliable observations, the model's prediction is extremely accurate in a real-time observation extraction. Also, the HMM showed the desired flexibility, by predicting correctly different state sequences.

Experiment	Description	Correct labels	Total labels	Accuracy
1	Normal greeting with every state	24	25	0.960
2	Greeting without DS or HD	24	27	0.889
3	Greeting with DS only close to person	23	23	1
4	Greeting starting at short distance	10	10	1
5	Smiling greeting (smile=2.5)	32	32	1
6	Gazing greeting (gaze=0.4)	22	30	0.733
Total		135	147	0.918

Table 3.8: Accuracy of the model on the different sequences

Chapter 4

Greeting Model using Behavior Trees

4.1 Global Overview

We have previously detailed a model capable to detect target people intending to greet, analyze their behaviors and reflect them in the set of greeting phases described by Kendon, estimating which phase of the greeting the person is performing.

In this chapter, a new branch is included in this model: the reaction branch, represented by the dotted and green section on the updated global diagram, Figure 4.1. Using Behavior Trees to perform movements sequentially, a robot would now be able to react to the greeter's behavior, depending on the state predicted by the HMM, by replicating the same state. Each state will have a previously defined Behavior Tree, i.e., a series of actions the robot will perform as long as the model predicts it as the most probable state. A more detailed explanation of Behavior Trees was presented in section 2.3.

For constructing and visualizing our model's Behavior Trees, we used a software named Groot (<https://github.com/BehaviorTree/Groot>), a graphical editor with the purpose of BT creation. The BTs to be presented in the following subsections were constructed using this software.

4.1.1 Global Behavior Tree

As in Figure 4.1, the most probable state computed by the HMM at each observation is published on a ROS topic called `"/state"`, at the sampling rate of the observations. This topic is subscribed by the Behavior Tree of our model, using the `GetState` execution node, which extracts the data from the topic and writes it into a String variable called `"state"`.

Figure 4.2 presents the global structure of the Behavior Tree to be followed by our robot.

In order to map the state variable to the respective sequence of actions to be performed, a `Switch` block was used, as in the figure. A `Switch6` block indicates the variable can take six different values, which would, therefore, match six different child nodes. In this specific situation, each value of state from 1 to 6 is matched to the respective state's node. Each one of these nodes contains a sub-tree representing the state, on an example of a BT's modularity.

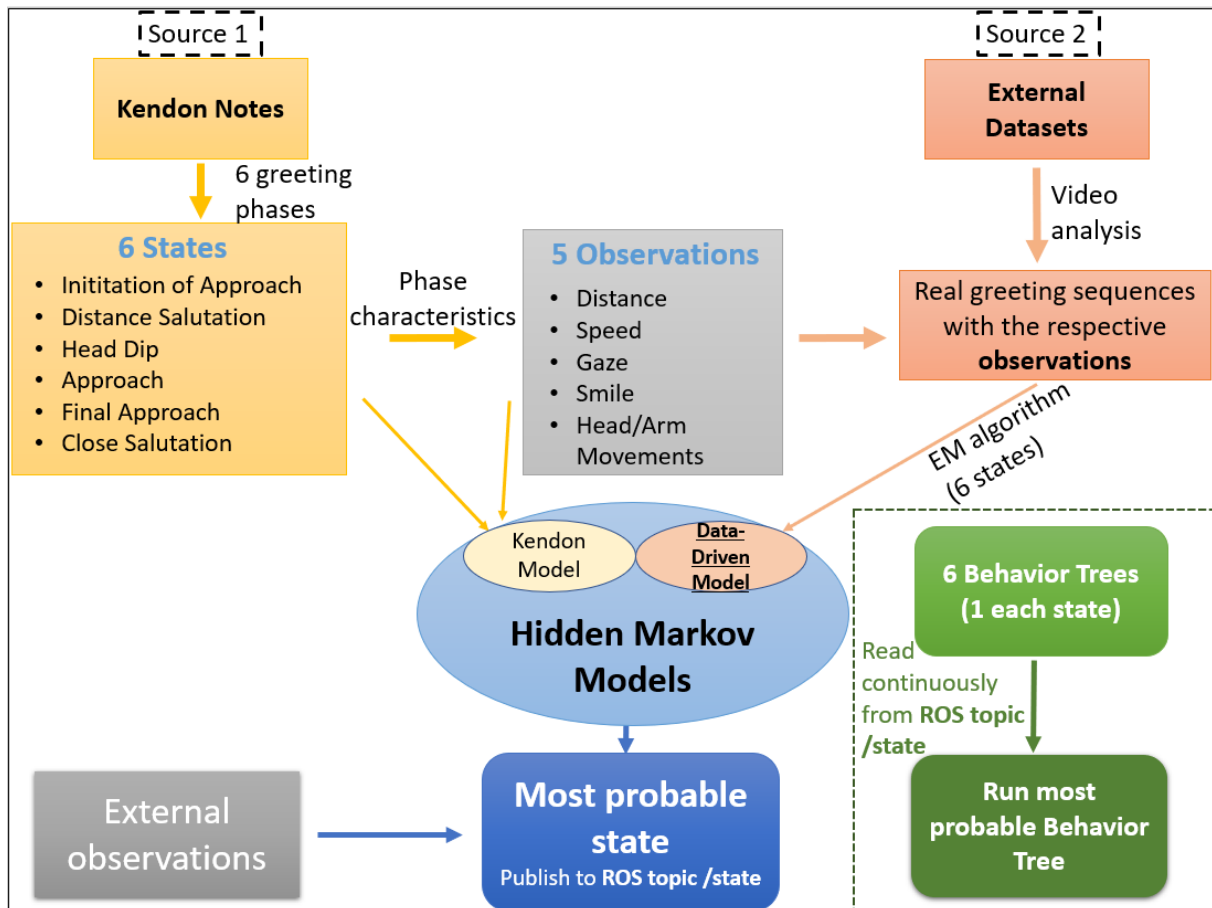


Figure 4.1: Global diagram of the model with the addition of the Behavior Tree branch (dotted)

Once the *GetState* node returns *Success*, the flow control node activates the Switch block, which will use the variable to send ticks to one of its children, allowing the predicted state to be immediately replicated.

The use of a Reactive Sequence flow control node provides flexibility to this BT, since it continues to send ticks to the *GetState* node after it returned *Success*. In this manner, every time that a ROS node (*HMM*) updates the value on the `"/state"` topic, the BT state variable will change and the Switch block will react to it, changing automatically to the correct node.

The parent-child tick routing was set to a rate of 10 times per second, meaning that a parent checks the state of its child node 10 instances in a second. This value was set with the requirement that it imperatively had to be higher than HMM's state changing rate, 5 per second. By using a higher rate, we assure no change of state can happen without the respective reaction on the BT.

4.1.2 Behavior Tree of each phase

Here we describe the six sub-trees present in the global BT, one for each of the six model states.

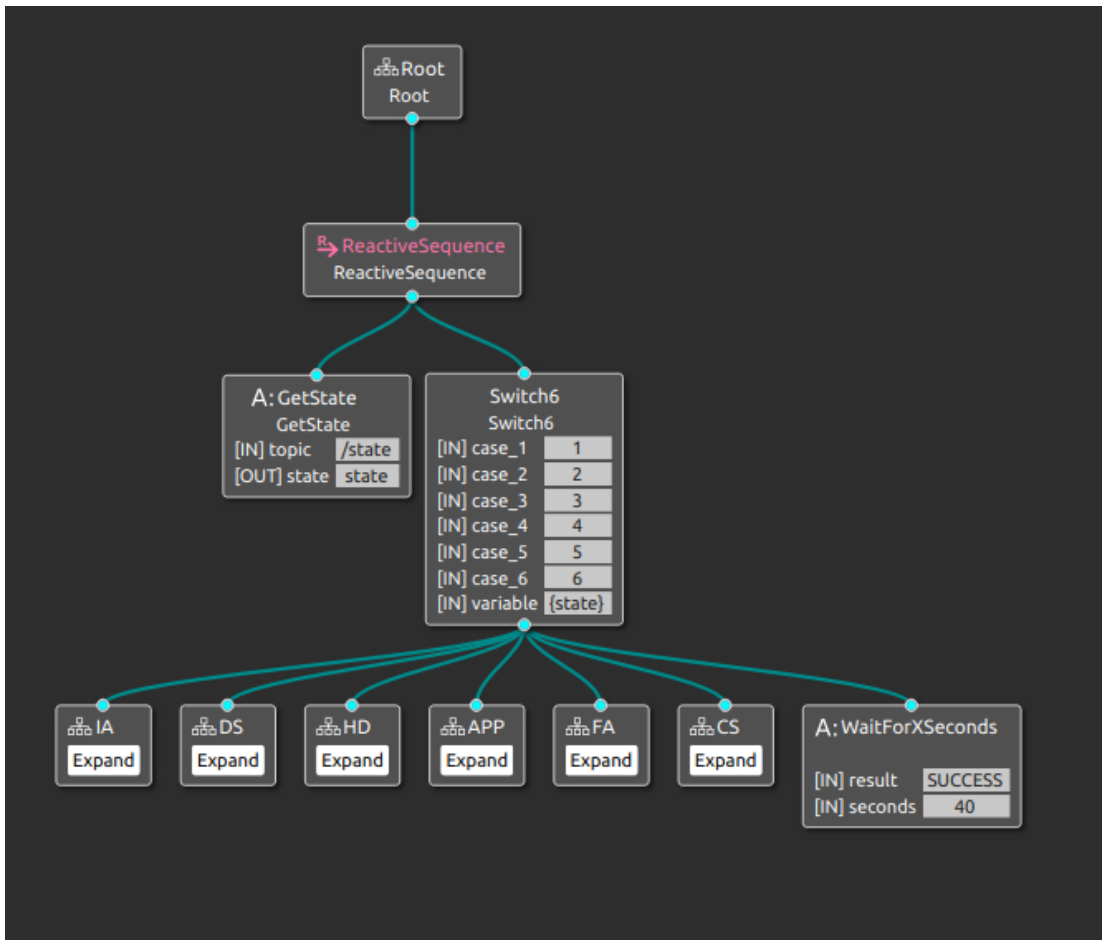


Figure 4.2: Global Behavior Tree of the model

Initiation of Approach

The first BT (Figure 4.3) runs every time that $state = 1$ verifies. Hence, as soon as the HMM identifies the *Initiation of Approach* phase, the robot tries to replicate it by performing the sequence of actions in the BT. These consist of an orientation changing and direct gaze.

The first leaf node, *GetPoseFromFace*, subscribes to the "/faces" ROS topic, which contains the face information extracted (referred to in section 3.3) and outputs a variable, "closest_face", which only has the position and orientation of the target person's face.

After this, the *TurnToPersonCoords* and *MoveBase* nodes will use the obtained variable to orientate the robot in the direction of the person, without changing its position. An example of this kind of movement can be seen in Figure 4.4, and Algorithm 3 describes the technique more formally. Here the "atan2" function used is the 2-argument arctangent, which returns an unambiguous angle value, using the x and y signs that are lost when $arctan(y/x)$ is made.

The last leaf node, *GazeAtTarget*, performs the gaze action, by orientating Vizzy's head to the position of the previously identified face.

These four nodes replicate Kendon's description of this phase, also complying with the generally followed social norm that people only look at the other after orientating their body first, with the fear of not being replied.

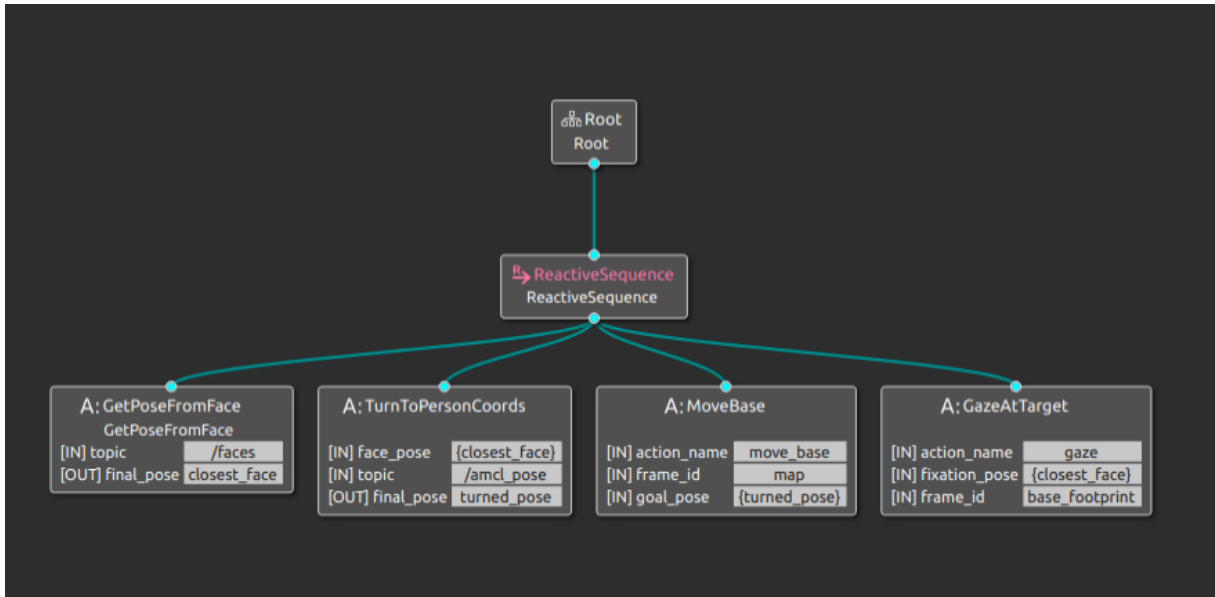


Figure 4.3: Behavior Tree of the *Initiation of Approach* phase

Once again, the Reactive Sequence node sends ticks to previous nodes, independently of these returning *Success*, which is important to keep the robot updated with the target person's position. If the target suddenly moves, the frontal orientation and direct gaze should be held.

Algorithm 3 An algorithm for turning the robot into target person's direction

Input: Target face's pose and ROS topic that contains robot pose

Output: Movement of the Robot

- 1: $P \leftarrow FacePose$ ▷ Person's pose in robot coordinate frame
 - 2: $R \leftarrow ROS\ topic\ /amcl_pose\ data$ ▷ Robot's pose in world coordinate frame
 - 3: $\alpha_1, \beta_1, \lambda_1 \leftarrow GETRPY(R(orientation))$ ▷ We further ignore Roll and Pitch as these are not relevant
 - 4: $R_Z \leftarrow Rotation\ Matrix\ around\ Z\ axis\ (\lambda = \lambda_1)$
 - 5: $PW \leftarrow R_Z * P(position) + R(position)$ ▷ Person's position in world coordinates
 - 6: $\lambda_F \leftarrow atan2(y = (PW_y - R(position)_y), x = (PW_x - R(position)_x))$
 - 7: $FinalOrientation \leftarrow (0, 0, sin(\lambda_F/2), cos(\lambda_F/2))$ ▷ Orientation in Quaternion format
 - 8: $FinalPosition \leftarrow R(position)$
 - 9: $MOVEBASE(FinalPosition, FinalOrientation)$ ▷ Moves the robot to the pose given
-

Distance Salutation

The above BT (Figure 4.5) is activated when the predicted state is the *Distance Salutation*, given by the value 2 published on the "/state" topic.

This sub-tree starts with an action node called *ArmRoutines* which performs a waving movement, by publishing the gesture "WAVE" to a ROS topic which will be subscribed by a node controlling Vizzy's arm motors.

When the gesture starts to be performed, the node returns *Success* and the Sequence starts to send ticks to the next nodes. The combination of the *GetPoseFromFace* and *GazeAtTarget* nodes was already seen on the first sub-tree and it ensures direct gaze while the waving is performed.

After the gaze is set and the respective node returns *Success*, there is a last node, which only

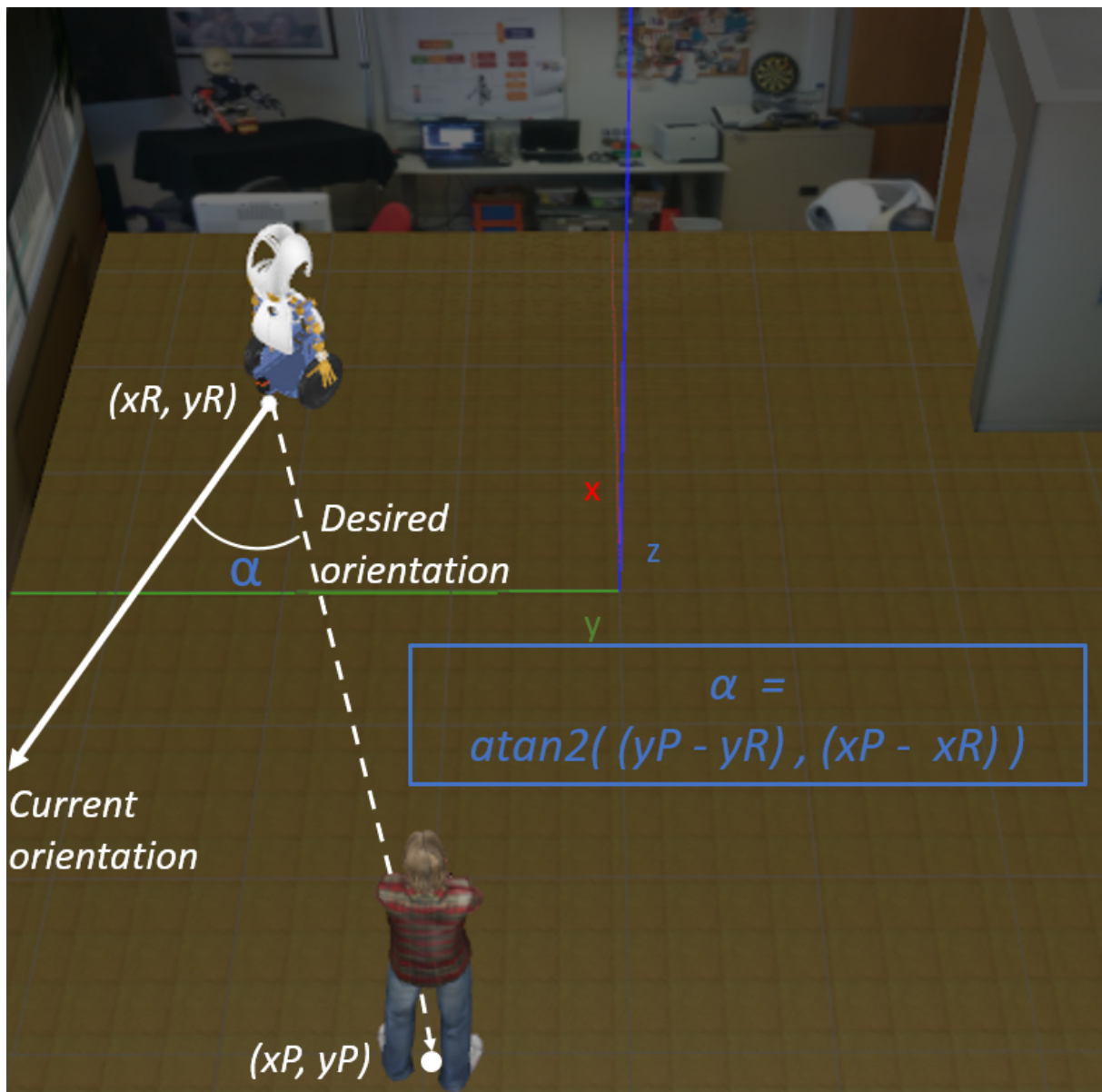


Figure 4.4: Example of the action of turning to a target individual

functions as a pause. This was inserted to guarantee that the sub-tree does not return *Success* and is performed again, which would cause the robot to perform multiple waving movements.

Head Dip

Figure 4.6 represents the sub-tree that will be ran in case the model is on state 3, the *Head Dip*.

To simulate this movement, we created a node called *GazeDownCoords*, which, given a target face, returns a target pose with the same coordinates of the face, but half of the height. After a distance salutation movement, which always includes direct gaze at the face, changing this gaze to a lower direction will replicate a head dip movement.

Before this head movement, it should be ensured that no other actions are being performed, therefore, we start this sub-tree with an *ArmRoutines* action node, which sets the robot's arms to the default

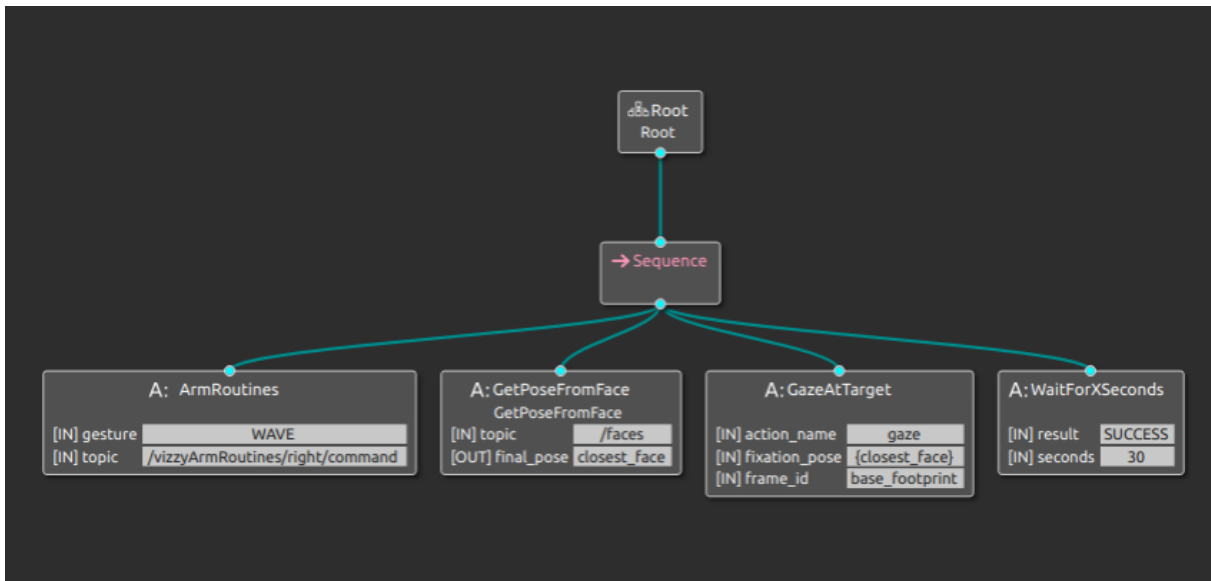


Figure 4.5: Behavior Tree of the *Distance Salutation* phase

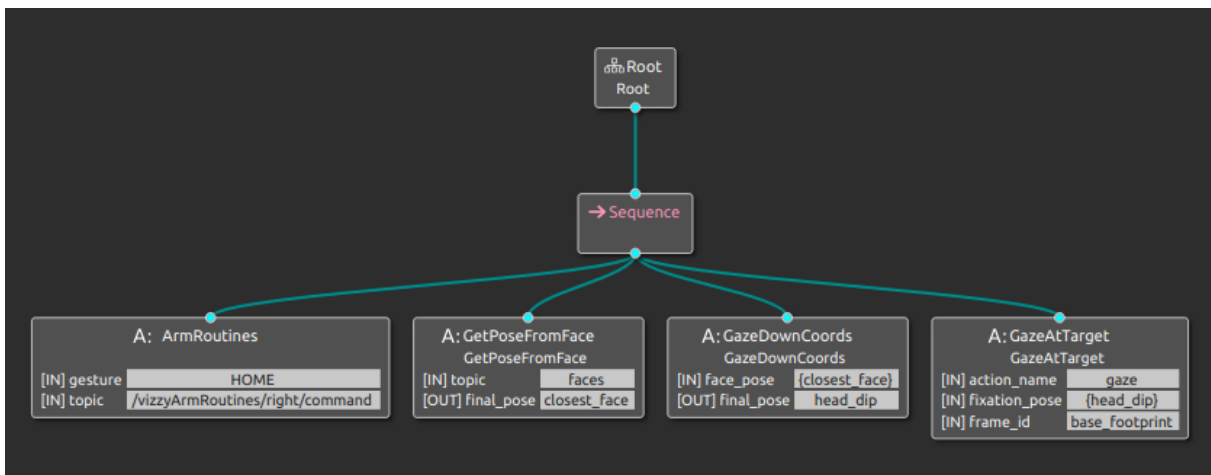


Figure 4.6: Behavior Tree of the *Head Dip* phase

position (HOME).

Approach

When the model detects an *Approach* phase (i.e., state 4 is published on the topic), the Switch block starts sending ticks to the sub-tree seen in Figure 4.7.

This tree has a Parallel node, which allows its 2 child nodes to be running simultaneously. The Parallel node was created with its variable $M = 2$, which means it will only return *Success* if both children return *Success*. If this value was 1, the Parallel node would return *Success* any time one of its children returned it, stopping the execution of the *Running* child. To properly replicate this phase we require both nodes, below described, to be completed.

The left child node runs a Reactive Sequence with 3 child nodes: a *GetPoseFromFace*, which, as already mentioned, returns the position and orientation of the target person's face; a *GetApproachCo-*

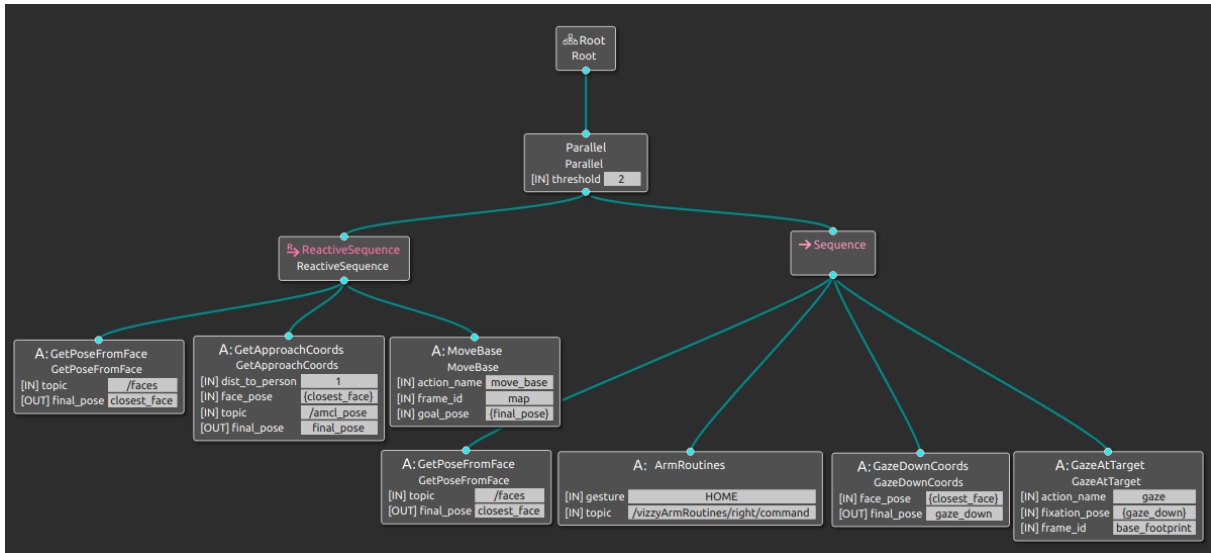


Figure 4.7: Behavior Tree of the *Approach* phase

ords, which calculates approach coordinates; and the *MoveBase*, which calculates a plan and moves to the coordinates given.

The *GetApproachCoords* node outputs these approach coordinates, i.e., the goal coordinates for the robot to approach the target person. To compute this, it takes as input a parameter which sets up the distance we want the robot to keep from the person (we used 1 meter in the example tree). Algorithm 4 was used to reach the goal coordinates, in a situation similar to the described in Figure 4.8. Here, the left figure shows the robot moving to its goal position, 1 meter away from the target, which is, by [46], an adequate distance for a Human-Robot-Interaction. The right figure represents the adjustment of its orientation, so that the person is greeted from the front.

The parent node (left child of the Parallel node) being a Reactive Sequence is essential for tracking the target’s movements. By constantly ticking the previous nodes, the position of the target person is updated in a variable every 0.1 seconds, which is the tick period. Consequently, new approach coordinates are calculated at this rate, allowing the robot to keep adjusting its movements, according to the circumstances. This ensures a proper approach even in unexpected situations, such as a change in the target’s velocity or in its approach direction, due to dodging an obstacle, for example.

On the right side of the parallel node, we ensure one major characteristic of this phase: the aversion of gaze. We opted for a partial aversion of gaze, using the *GazeDownCoords* and *GazeAtTarget* nodes, which set the gaze direction to half of the target face’s height, as in the previous state. With this gaze direction, at regular distances, a target face would be seen near the limits of the sight of view explained in section 3.3.3, which would translate into a near 50% probability of looking at the person, similarly to the values of gaze estimated by Kendon for this phase.

The *ArmRoutines* node is also present to ensure the arms are at their normal position, as done in the HD phase.

Algorithm 4 Algorithm to approach a target person and stop with frontal orientation at a chosen distance

Input: Target face's pose, ROS topic that contains robot pose and desired distance from person

Output: Approach movement

- 1: $P \leftarrow FacePose$ ▷ Person's pose in robot coordinate frame
 - 2: $R \leftarrow ROS\ topic / amcl_pose\ data$ ▷ Robot's pose in world coordinate frame
 - 3: $\alpha_1, \beta_1, \lambda_1 \leftarrow GETRPY(P(orientation))$
 - 4: $\alpha_2, \beta_2, \lambda_2 \leftarrow GETRPY(R(orientation))$ ▷ We further ignore Roll and Pitch as these are not relevant
 - 5: $R_Z \leftarrow$ Rotation Matrix around Z axis ($\lambda = \lambda_2$)
 - 6: $PW \leftarrow R_Z * P(position) + R(position)$ ▷ Person's position in world coordinates
 - 7: $\lambda_P \leftarrow \lambda_1 + \lambda_2$ ▷ Person's orientation in world coordinates
 - 8: $x_F \leftarrow PW_x + DistToPerson * \cos(\lambda_P)$
 - 9: $y_F \leftarrow PW_y + DistToPerson * \sin(\lambda_P)$
 - 10: $\lambda_F \leftarrow \lambda_P + \pi$ ▷ Robot's orientation must be the opposite of the person
 - 11: $FinalPosition \leftarrow (x_F, y_F)$
 - 12: $FinalOrientation \leftarrow (0, 0, \sin(\lambda_F/2), \cos(\lambda_F/2))$ ▷ Converting into Quaternion
 - 13: $MOVEBASE(FinalPosition, FinalOrientation)$ ▷ Moves the robot to the pose given
-

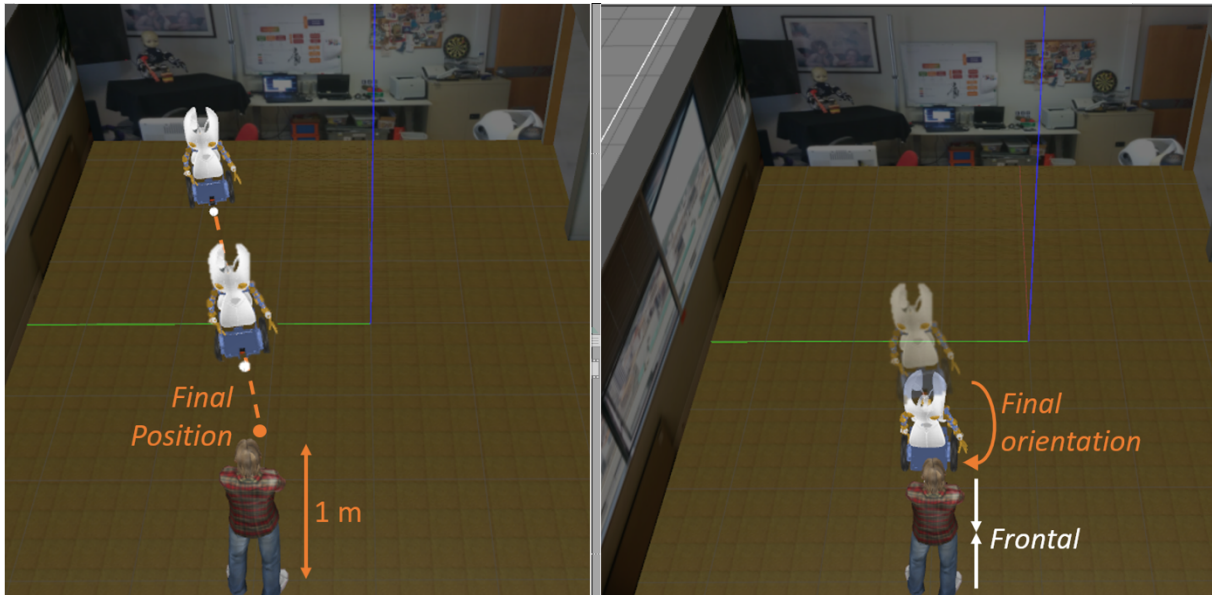


Figure 4.8: Robot approach movement. Left: setting target position; Right: setting target orientation

Final Approach

Similarly to the previous phase, the sub-tree related to the *Final Approach* phase, initiated when $state = 5$, is also composed by a Parallel node with 2 children and $M = 2$ (see Figure 4.9).

As the FA phase is a continuation of the *Approach*, where the robot should start to prepare itself for the *Close Salutation*, the two sub-trees are almost identical, with the only change happening on the gaze action. This phase is, by Kendon, usually characterized by a return of mutual gaze, and the robot responds accordingly, by looking at the target's face.

The change regarding the gazing behavior of the robot is found in the right child of the Parallel node, where the combination of the *GetPoseFromFace* and *GazeAtTarget* nodes ensure a direct looking at target face, opposing the aversion of gaze usually found previously.

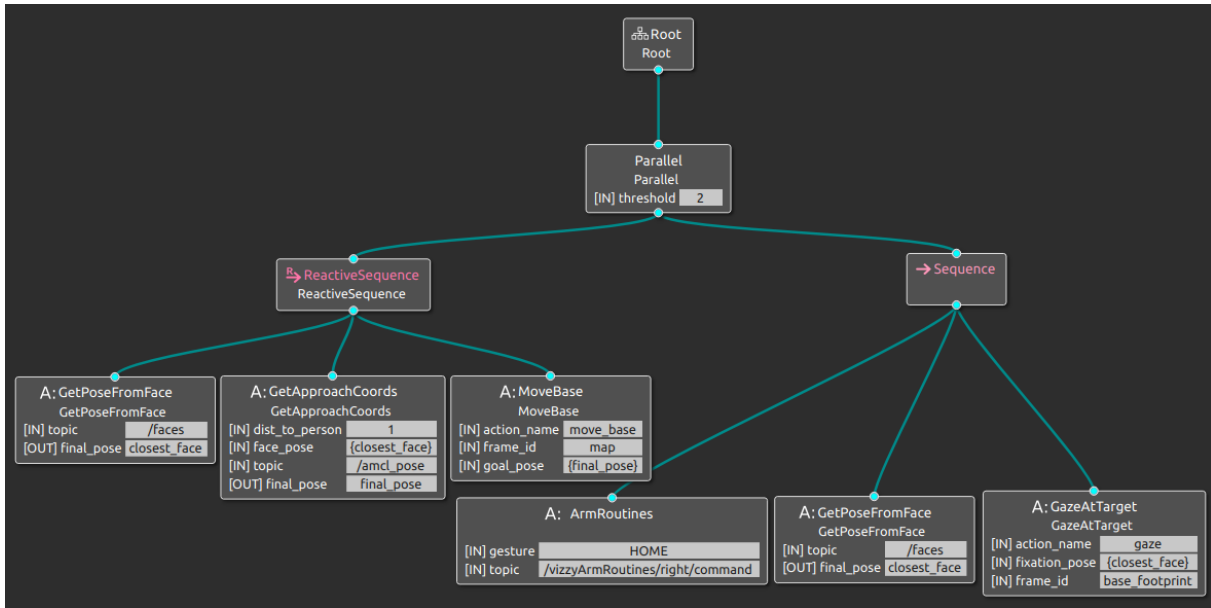


Figure 4.9: Behavior Tree of the *Final Approach* phase

Close Salutation

In Figure 4.10 we see the robot's response to a *Close Salutation* phase. This sub-tree is activated when the HMM section publishes the value 6 on the `"/state"` topic.

First of all, by using the *GazeAtTarget* node with the target face's information, we ensure that the robot is looking directly at the person, which is one essential aspect of the phase.

After gazing, the robot performs the salutation. A parallel node allows it to perform a handshake movement, which is done with the *ArmRoutines* action node, simultaneous with a verbal greeting, by saying *"Muito prazer"*, which translates to "Pleasure to meet you".

Once more, the use of a Reactive Sequence node guarantees that the `"closest_face"` variable is constantly updated with the information on the `"/faces"` ROS topic and the gaze direction changes if necessary (the person moves, for example).

4.2 Behavior Tree Testing

In order to confirm the efficiency of the integration between the Hidden Markov Model and the developed Behavior Trees, we made some tests in the Vizzy simulator, described in section 3.6.3.

We followed a testing approach similar to the previous simulator tests (section 3.6.4), where a fictional person stands in front of the robot and starts to perform a greeting sequence. However, in this case, the HMM will not only predict the most probable states, but also send them to the `"faces"` ROS topic, as mentioned in the previous sections of this chapter. This would activate the global Behavior Tree, and the respective sub-tree would be ran, reproducing the person's behavior and simulating a two-sided greeting sequence. To be able to reproduce this properly, all time-related variables were adapted due to the simulator's conditions, including the time gaps between observations, speed of the person and

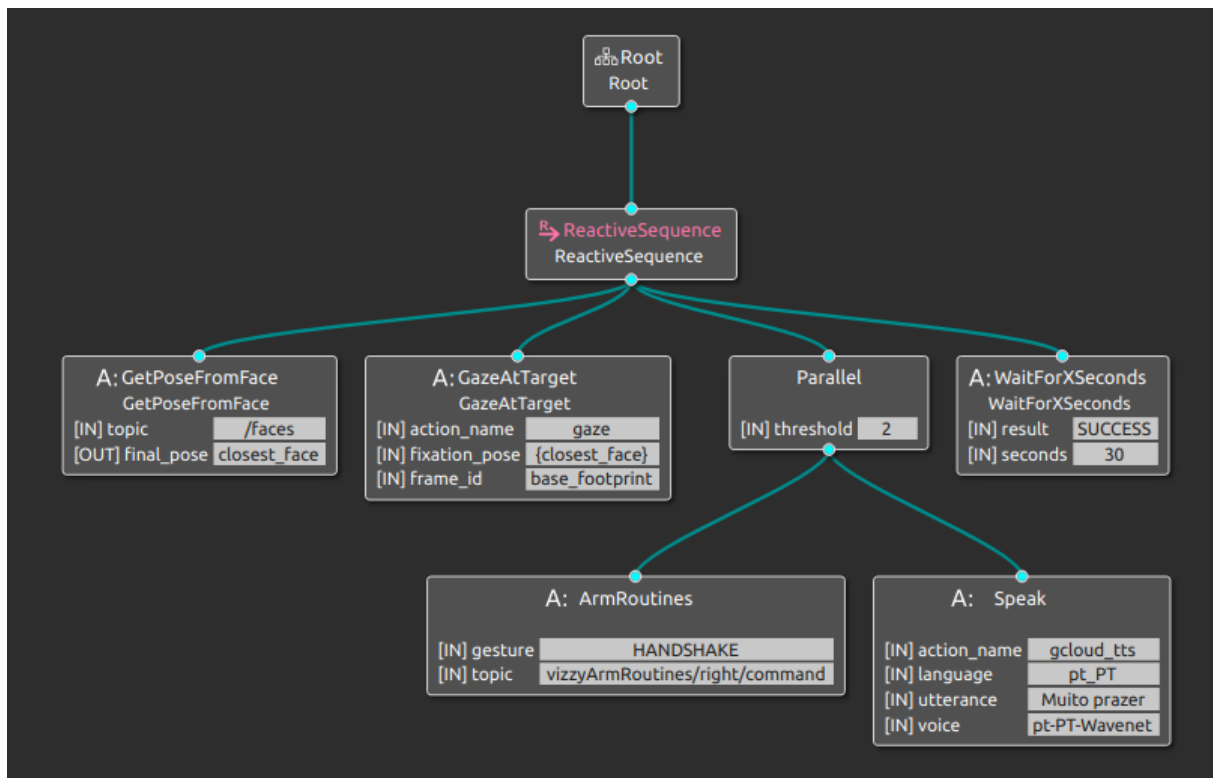


Figure 4.10: Behavior Tree of the *Close Salutation* phase

duration of gestures.

An example of a successful greeting sequence can be seen in Figure 4.11 to 4.13. Firstly, Vizzy sights a person (Figure 4.11) and changes its orientation and gaze display, as it estimates the person has also observed it (IA state). As the woman starts to approach it (APP state), the robot also starts to move (Figure 4.12), however it soon stops, to perform a right-arm waving, as it detected a similar movement (DS state). Following it, the robot slightly lowers its head (detected HD state) and continues the movement. As the two parts are approximating, Vizzy changes its gaze behavior to look directly (detected FA state), and displays a handshake when both stop in front of each other and the CS state was predicted (Figure 4.13).

After experimenting a few more greeting situations, similarly to what was done in section 3.6.4, our testing was concluded. Even though we were not able to experiment this system on a real robot, due to the circumstances, the results obtained in Chapter 3 predicting the greeting phases, and the respective robot movements in the simulator provide very promising results. We believe that an implementation based on this system having more time and less constraints can achieve more relevant results regarding complete human-robot greeting sequences.

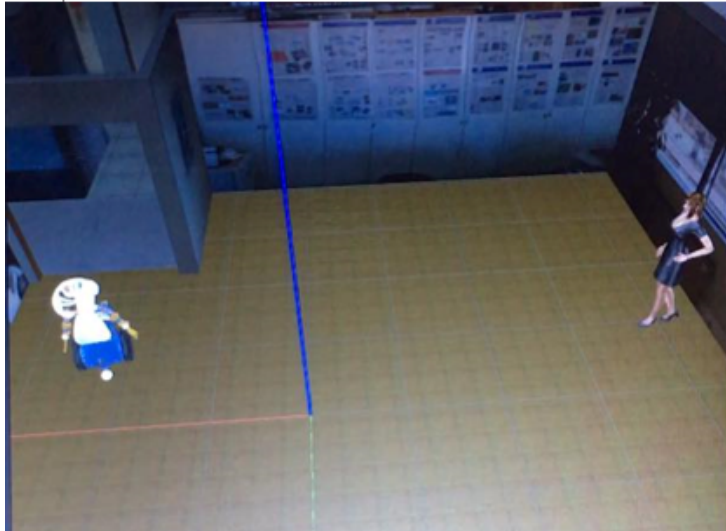


Figure 4.11: Simulation: Initial positions



Figure 4.12: Simulation: Approaching movements



Figure 4.13: Simulation: *Close Salutation*

Chapter 5

Conclusions

After a description of most of the aspects of this work, we will now present the conclusions and final notes taken. In the following sections, the work's successes will be discussed, as well as some aspects which could enhance the results obtained.

5.1 Achievements

To build a Hidden Markov Model trained with real data that could represent six phases similar to Kendon's greeting model was, undoubtedly, a major challenge in this work.

The extraction of data that would be proper to train our HMM was, occasionally, troubling. There were some datasets available online containing social experiences, however, most of them did not have the data necessary for the observations, or did not perform most of the greeting phases. In addition, external variables such as the indoors environments for training created different values for key observations such as distance and speed, comparing to Kendon's outdoors environments. Nonetheless, we could develop a data-driven model that was able to discover the six phases of Kendon's greeting model.

Apart from all the resemblances to the base model, this trained model also produced positive results when experimented on a test set. Predicting the states of the test sequences, using the Viterbi algorithm, got 89.3% of the labeling correct, and 83.9% using the forward algorithm. The analysis of the confusion matrix and the state-tracking plots also demonstrate the errors, beyond being few, were balanced through all classes and did not escalate.

When experimenting with several training and test sets, the average accuracy for the Data-Driven Model decreased slightly, with both predicting methods getting, on average, more than 78% of the labels correct. Although the model had a larger accuracy standard deviation (around 11%) than the Kendon model (around 5%), we believe that by gathering a larger dataset, the Data-Driven Model should improve the handcrafted Kendon Model by a larger margin. In addition, the results showed that the most of the state classification errors appear to be noisy results, which can be filtered, lowering the accuracy variance and improving the average.

In the context of a realistic greeting phase estimation, in an environment different from the training

ones, the application of the forward algorithm and the Data-Driven Model provided positive results, achieving an accuracy around 93%, which we believe can still be improved by gathering more greeting samples.

When testing in the simulator our robot reacting to the state prediction of the Hidden Markov Model, the implementation using Behavior Trees also responded as desired. The six states could be appropriately displayed, which is promising for future testing with a real robot.

5.2 Future Work

Although the last section presents relevant successes, there is room for improvement on some aspects of this work and still an uncountable number of developments that could enhance HRI on social robots.

Firstly, the training of our HMM would have highly benefited from having more greeting sequences available. Datasets containing greetings performed with reasonable quality and all the necessary information for, at the minimum, localizing the greeters in space are very scarce on the internet. Filming and posteriorly extracting the desired information on one or more big meetings as the one Kendon analyzed was not feasible in this work, but would have certainly improved the model. To include more than one training environment would also bring even more robustness, and approximate the model to the actual situation of greetings, perhaps even more than Kendon's greeting model, since this one is mainly centered in one experiment.

Regarding the real-time prediction of greeting phases, the implementation of movements detectors is crucial for a social robot to be human-independent, while socializing using this model. Due to the complexity of training an algorithm of this kind, this was not implemented in the present work, however, our manual implementation brings promising results for the functionality of an automated detector along this HMM.

On the robot's greeting behavior, even though we consider naturalness was achieved on our greetings, a few details can make the robot more similar to a greeting person. In our opinion, the implementation of smiling would make it appear friendlier and would bring more confidence to the person intending to greet. Other details mentioned by Kendon such as the grooming, or the "body-cross" are subtle movements, however, can give an important humanoid touch if properly implemented. Other salutations such as the nod can also bring flexibility to our greetings, since the manner as we salute people should change depending on several factors.

To finish, another valuable extension to this work would be an adaptation to group greeting, instead of solely individual. Despite a group greeting being composed of several individual ones, it would be necessary to implement the HMM for continuous greetings, without repeating any target.

Bibliography

- [1] A. K. Pandey and R. Gelin. A mass-produced sociable humanoid robot: Pepper: The first machine of its kind. *IEEE Robotics & Automation Magazine*, PP:1–1, 07 2018. doi: 10.1109/MRA.2018.2833157.
- [2] M. Ramanathan, N. Mishra, and N. Thalmann. *Nadine Humanoid Social Robotics Platform*, pages 490–496. 06 2019. ISBN 978-3-030-22513-1. doi: 10.1007/978-3-030-22514-8_49.
- [3] A. Niculescu, B. van Dijk, A. Nijholt, D. K. Limbu, S. L. See, and A. H. Y. Wong. Socializing with olivia, the youngest robot receptionist outside the lab. In S. S. Ge, H. Li, J.-J. Cabibihan, and Y. K. Tan, editors, *Social Robotics*, pages 50–62, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-17248-9.
- [4] J. Lebersfeld, C. Brasher, C. Clesi, C. Stevens Jr, F. Biasini, and M. Hopkins. 2157 the socially animated machine (sam) robot: A social skills intervention for children with autism spectrum disorder. *Journal of Clinical and Translational Science*, 2:49–49, 06 2018. doi: 10.1017/cts.2018.190.
- [5] G. Milliez. Buddy: A companion robot for the whole family. pages 40–40, 03 2018. ISBN 978-1-4503-5615-2. doi: 10.1145/3173386.3177839.
- [6] A. Kendon. *Conducting Interaction: Patterns of Behavior in Focused Encounters*. Cambridge University Press, 1990. ISBN:978-0521389389.
- [7] P. Moreno, R. Nunes, R. Figueiredo, R. Ferreira, A. Bernardino, J. Santos-Victor, R. Beira, L. Vargas, D. Aragão, and M. Aragão. Vizzy: A humanoid on wheels for assistive robotics. In *Robot 2015: Second Iberian Robotics Conference*, pages 17–28. Springer, Cham, 2015. doi: 10.1007/978-3-319-27146-0_2.
- [8] D. Jurafsky and J. H. Martin. Hidden markov models. In *Speech and Language Processing*, pages 548–563. Prentice Hall PTR, USA, 1st edition, 2000. ISBN 0130950696.
- [9] M. Colledanchise and P. Ogren. *Behavior Trees in Robotics and AI: An Introduction*. 07 2018. ISBN 9781138593732. doi: 10.1201/9780429489105.
- [10] J. Wang and W. Tepfenhart. *Formal Methods in Computer Science*. 06 2019. ISBN 9780429184185. doi: 10.1201/9780429184185.

- [11] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966. ISSN 00034851. URL <http://www.jstor.org/stable/2238772>.
- [12] J. Picone. Continuous speech recognition using hidden markov models. *IEEE ASSP Magazine*, 7(3):26–41, 1990.
- [13] J. Henderson, S. Salzberg, and K. Fasman. Finding genes in dna with a hidden markov model. *Journal of computational biology : a journal of computational molecular cell biology*, 4:127–41, 02 1997. doi: 10.1089/cmb.1997.4.127.
- [14] N. Nguyen. Hidden markov model for stock trading. *International Journal of Financial Studies*, 6:36, 03 2018. doi: 10.3390/ijfs6020036.
- [15] P. A. Gagniuc. *Markov Chains: From Theory to Implementation and Experimentation*. 07 2017. ISBN 978-1-119-38755-8.
- [16] R. Paroli and L. Spezia. Parameter estimation of gaussian hidden markov models when missing observations occur. *Metron - International Journal of Statistics*, LX:163–179, 02 2002.
- [17] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [18] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.
- [19] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989. doi: 10.1109/5.24143.
- [20] D. Hu, Y. Gong, B. Hannaford, and E. Seibel. Semi-autonomous simulated brain tumor ablation with ravenii surgical robot using behavior tree. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015:3868–3875, 06 2015. doi: 10.1109/ICRA.2015.7139738.
- [21] K. R. Guerin, C. Lea, C. Paxton, and G. D. Hager. A framework for end-user instruction of a robot assistant for manufacturing. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6167–6174, 2015. doi: 10.1109/ICRA.2015.7140065.
- [22] B. Heenan, S. Greenberg, S. A. Manesh, and E. Sharlin. Designing social greetings in human robot interaction, 2014. doi:10.1145/2598510.2598513.
- [23] D. Brscic, T. Ikeda, and T. Kanda. Do you need help? a robot providing information to people who behave atypically. *IEEE Transactions on Robotics*, PP:1–7, 01 2017. doi: 10.1109/TRO.2016.2645206.
- [24] C. Shi, S. Satake, T. Kanda, and H. Ishiguro. A robot that distributes flyers to pedestrians in a shopping mall. *International Journal of Social Robotics*, nov 2017. doi: 10.1007/s12369-017-0442-7.

- [25] S. Satake, T. Kanda, D. Glas, M. Imai, H. Ishiguro, and N. Hagita. A robot that approaches pedestrians. *Robotics, IEEE Transactions on*, 29:508–524, 04 2013. doi: 10.1109/TRO.2012.2226387.
- [26] J. Avelino, L. Garcia-Marques, R. Ventura, and A. Bernardino. Break the ice: a survey on socially aware engagement for human-robot first encounters. *International Journal of Social Robotics*, in press.
- [27] E. Saad, J. Broekens, M. A. Neerinx, and K. V. Hindriks. Enthusiastic robots make better contact. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1094–1100, 2019. doi: 10.1109/IROS40897.2019.8967950.
- [28] M. Foster, R. Alami, O. Gestranus, O. Lemon, M. Niemelä, J.-M. Odobez, and A. K. Pandey. The mummer project: Engaging human-robot interaction in real-world public spaces. volume 9979, pages 753–763, 11 2016. ISBN 978-3-319-47436-6. doi: 10.1007/978-3-319-47437-3_74.
- [29] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. URL <https://www.ros.org>.
- [30] T. Baltrušaitis, A. Zadeh, Y. C. Lim, and L. Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 59–66, 2018. doi: 10.1109/FG.2018.00019.
- [31] E. Wood, T. Baltruaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3756–3764, 2015. doi: 10.1109/ICCV.2015.428.
- [32] A. Zadeh, T. Baltrušaitis, and L. Morency. Deep constrained local models for facial landmark detection. *CoRR*, abs/1611.08657, 2016. URL <http://arxiv.org/abs/1611.08657>.
- [33] T. Baltrušaitis, P. Robinson, and L. Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 354–361, 2013. doi: 10.1109/ICCVW.2013.54.
- [34] T. Baltrušaitis, M. Mahmoud, and P. Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 06, pages 1–6, 2015. doi: 10.1109/FG.2015.7284869.
- [35] P. L. Alfano and G. F. Michel. Restricting the field of view: Perceptual and performance effects. *Perceptual and Motor Skills*, 70(1):35–45, 1990. doi: 10.2466/pms.1990.70.1.35. URL <https://doi.org/10.2466/pms.1990.70.1.35>. PMID: 2326136.
- [36] H. Strasburger, I. Rentschler, and M. Jüttner. Peripheral vision and pattern recognition: A review. *Journal of Vision*, 11(5):13–13, 12 2011. ISSN 1534-7362. doi: 10.1167/11.5.13. URL <https://doi.org/10.1167/11.5.13>.

- [37] P. Ekman and W. Friesen. Facial action coding system: a technique for the measurement of facial movement. 1978.
- [38] S. Park, K. Lee, J.-A. Lim, H. Ko, T. Kim, J.-I. Lee, H. Kim, S.-J. Han, J.-S. Kim, S. Park, J.-Y. Lee, and E. C. Lee. Differences in facial expressions between spontaneous and posed smiles: Automated method by action units and three-dimensional facial landmarks. *Sensors*, 20:1199, 02 2020. doi: 10.3390/s20041199.
- [39] K. Schmidt and J. Cohn. Dynamics of facial expression: Normative characteristics and individual differences. volume 0, 01 2001. doi: 10.1109/ICME.2001.1237778.
- [40] I. D. Gebru, S. Ba, X. Li, and R. Horaud. Audio-visual speaker diarization based on spatiotemporal bayesian fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1086–1099, 2018. doi: 10.1109/TPAMI.2017.2648793.
- [41] C. Coppola, S. Cosar, D. Faria, and N. Bellotto. Automatic detection of human interactions from rgb-d data for social activity classification. In *IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 871–876, 2017.
- [42] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814 vol. 2, 2005. doi: 10.1109/CVPR.2005.56.
- [43] G. Metta, P. Fitzpatrick, and L. Natale. Yarp: Yet another robot platform. *International Journal of Advanced Robotic Systems*, 3(1):8, 2006. doi: 10.5772/5761. URL <https://doi.org/10.5772/5761>.
- [44] H. R. Kam, S.-H. Lee, T. Park, and C.-H. Kim. Rviz: a toolkit for real domain data visualization. *Telecommunication Systems*, 60:337–345, 2015.
- [45] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, 2004. doi: 10.1109/IROS.2004.1389727.
- [46] L. Takayama and C. Pantofaru. Influences on proxemic behaviors in human-robot interaction. *IROS 2009*, 2009. doi:10.1109/IROS.2009.5354145.