POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Degree Thesis

Integration of facial expressions recognition and navigation output styles into a fuzzy logic emotional model for a mobile robot assistant

Supervisors

Prof. Marcello CHIABERGE

Prof. Paloma DE LA PUENTE

Candidate

Carlotta PIFFERI

December 2025

Abstract

The pursuit of a more natural and empathetic human–robot interaction represents one of the key challenges in contemporary social robotics. This thesis presents the design and implementation of an emotional control system for the TIAGo robot, developed as an extension of previously proposed fuzzy-logic emotional frameworks and carried out under the supervision of a psychologist. The system integrates real-time facial expression recognition with a fuzzy-logic emotional model that dynamically modulates the robot's navigation parameters. The goal is to enable the robot to exhibit adaptive and lifelike behaviour by adjusting its motion according to the emotions perceived in the user and its own internal affective state.

An initial version of the emotional control system was first implemented and tested in a pilot experiment with 8 participants, which produced preliminary evaluations and revealed some technical limitations (e.g., latency in the real-time image acquisition and processing pipeline, and navigation parameter updates that were too smooth to be clearly perceivable). Based on the results obtained, targeted improvements were introduced to develop an optimized system, which was then validated on a larger sample of 28 participants, ensuring higher statistical robustness and reliability of the results.

The proposed system defines two internal emotional dimensions for the robot, mood and alertness, which evolve in response to the emotions detected on the user's face. These internal states, processed through two fuzzy state tables, influence two corresponding navigation outputs: speed (linked to mood) and tuning (linked to alertness), the latter regulating angular velocity and proximity to obstacles. In this way, the robot's movement becomes an external manifestation of its internal affective state, producing behaviour that more closely resembles that of a living being.

Two distinct robot personalities, "shy" and "intense", were implemented by differentiating the fuzzy rules and the amplitude of the emotional reactions. The shy configuration is characterised by smoother and slower responses, while the intense one reacts with faster and more marked behavioural changes.

The entire system was developed within the ROS framework and integrated on the TIAGo platform at the Universidad Politécnica de Madrid. Real-time emotion detection was achieved using a convolutional neural network (CNN) trained on facial datasets, interfaced with the robot through dedicated Python and ROS nodes.

The experimental validation involved human participants who interacted with the robot under two conditions: neutral mode (without emotional modulation) and personalised mode (with the emotional system active). Participants then

completed the Godspeed Questionnaire (Animacy and Likeability scales) and the EVEA self-report emotion scale. Statistical analyses using mixed ANOVA showed significant improvements in perceived animacy and likeability for the personalised mode, confirming that integrating emotional dynamics enhances the perceived naturalness and pleasantness of TIAGo. Furthermore, qualitative data revealed that participants perceived the shy robot as more delicate and empathetic, while the intense one appeared more expressive and engaging.
The proposed system allows for intuitive and modular adaptation of robot's personality traits, laying the groundwork for future developments in emotionally adaptive and socially intelligent robotics.

Acknowledgements

I would like to express my deepest gratitude to Professor Paloma de la Puente for her guidance and precious advice. Her experience and continuous support have been fundamental to the development of this project and to my professional as well as personal growth.

I would also like to thank Professor Marcello Chiaberge for his support and feedback.

A special thanks also goes to Zhang and Jiaxi, with whom I had the pleasure of sharing the lab.

Finally, I extend my gratitude to all the people who have supported me throughout this academic and personal journey.
I am deeply grateful to my parents, my family, my boyfriend and my friends, thank you for the love you surround me with every day.

Table of Contents

List of Tables

List of Figures

Acronyms

HRI
: Human-Robot Interaction

FA
: Fuzzy Atmosfield

AEI
: Artificial Emotional Intelligence

SAR
: Socially Assistive Robot

DoF
: Degrees of Freedom

IMU
: Inertial Measurement Unit

OGM
: Occupancy Grid Map

ROS
: Robot Operating System

SLAM
: Simultaneous Localization And Mapping

AMCL
: Adaptive Monte Carlo Localization

DWA
Dynamic Window Approach

CNN
Convolutional Neural Network

GUI
Graphical User Interface

HRIES
Human–Robot Interaction Evaluation Scale

EVEA
Escalas Visuales Analógicas de Estado de Ánimo

VAS
Visual Analog Scale

ANOVA
Analysis Of Variance

SD
Standard Deviation

Chapter 1

Introduction

Human–robot interaction (HRI) is progressively moving beyond the traditional view of robots as purely functional machines, evolving instead toward systems capable of engaging with humans in a social, adaptive and emotionally meaningful manner.

In daily interactions between humans and robots, the ability to interpret non-verbal and emotional signals stands as a frontier of rising importance. On one hand, humans are naturally inclined to read emotional states through visual, auditory, and behavioral cues; on the other, robots must acquire tools that enable them to recognize and respond to such stimuli coherently in order to live and collaborate effectively in social environments.

In recent years, the concept of social robotics has gained growing attention, with the goal of making robots not just functional tools, but also partners capable of interacting empathetically and adaptively with humans. In this context, equipping robots with an emotional system capable of influencing their operational behavior represents a fundamental step toward more "human" robotics.

The idea of artificial emotions aims to create computational models capable of translating sensory signals into dynamic internal states, which in turn can modulate the robot's actions, adapting them to the context and improving the quality of interaction. One of the most promising methodologies for managing this complexity is fuzzy logic, as it allows for modeling the uncertainty and ambiguity typical of emotions and human behavior. Unlike rule-based or threshold-based systems, it supports smooth transitions across emotional and behavioural states, allowing robots to react to stimuli in a human-like, continuous manner. By combining fuzzy inference with real-time perception modules—such as in this project facial-expression recognition—robots can build internal affective representations and translate them into coherent behavioural responses, including changes in speed, trajectory, or proximity management.

The present thesis contributes to this line of research by designing, implementing,

and experimentally validating an emotional control system for the TIAGo robot, developed at the Centro de Automática y Robótica (CAR) of the Universidad Politécnica de Madrid. Building on existing fuzzy-logic emotional frameworks and supervised by a psychologist, the system enables TIAGo to modulate its navigation behaviour dynamically based on emotions recognised on the user's face. Through this mechanism, the robot can adjust its linear velocity and its navigation "style", exhibiting behaviour that is more expressive, lifelike, and personalised. A key characteristic of the proposed system is the definition of two internal affective dimensions, mood and alertness, which evolve in real time according to the emotional cues detected in the user throught a real-time face expression detection algorithm that was integrated on the robot. These dimensions are processed through a hierarchy of fuzzy controllers that influence TIAGo's navigation parameters: speed (modulated by mood) and tuning, a composite parameter regulating angular velocity and obstacle-handling behaviour (modulated by alertness). This framework allows the robot's movement to become an external manifestation of its internal emotional state, bridging the gap between perception and behaviour.
To explore how personality traits can further enrich robot behaviour, two distinct robot personalities, shy and intense, were implemented by modifying the fuzzy rules and adjusting the sensitivity of the control system. These profiles differ in reaction amplitude, speed modulation, and tolerance to proximity. The shy personality exhibits cautious, reserved behaviour, whereas the intense one responds more rapidly and dynamically to perceived emotional states.
All the software modules developed, including the fuzzy emotional controller, the hierarchical affective state manager and the ROS interfaces, have been released in an open-source GitHub repository.
The complete codebase is available at: https://github.com/carlottapifferi2002/Real-time-fuzzy-emotional-control-system-for-social-robotics
The system was tested and validated through extensive experimental sessions involving human participants. After an initial pilot experiment, technical optimisations were introduced to increase responsiveness, reduce latency and ensure that behavioural changes were clearly perceivable. The final system was evaluated with 28 participants who interacted with both a neutral robot (without emotional modulation) and a personalised robot (with the emotional system active). Participants then completed standardised HRI questionnaires—Godspeed (Animacy and Likeability scales) and EVEA—together with open-ended questions. Statistical analyses (mixed ANOVA) showed significant improvements in perceived animacy and likeability when the emotional system was active, confirming the effectiveness of emotional modulation in enhancing user perception. Qualitative data further supported these findings, describing the personalised robot as more attentive, more responsive and more human-like.

1.1 Thesis objectives

This thesis addresses the challenge of integrating a real-time facial expression recognition algorithm and an emotional system based on fuzzy logic into the TIAGo robot, exploring how emotions and perceived stimuli—such as user emotions—can influence key parameters of its navigation behavior. This approach aims to make human-robot interaction more natural and adaptive, enhancing the robot's capabilities in social and assistive contexts.
The objective is twofold: on one hand, to study the impact of artificial emotions on the quality of human-robot interaction; on the other, to propose a modular and scalable approach that can be easily extended to other contexts or robotic platforms.

1.2 Intelligent Control Group

For this thesis project, I had the honor of working with the Intelligent Control Group (ICG), a research group within the Centro de Automática y Robótica (CAR) of the Universidad Politécnica de Madrid (UPM) and the Consejo Superior de Investigaciones Científicas (CSIC).
CAR is dedicated to advanced research in the fields of control engineering, artificial perception, and robotics. More information about it can be found on their official website https://www.car.upm-csic.es/.
ICG focuses on the development of intelligent control techniques for complex systems, integrating approaches such as fuzzy logic, expert systems and machine learning. Its activities include the design of algorithms for adaptive and predictive control, the analysis of nonlinear systems, and the development of applications in areas such as mobile robotics, industrial automation, and autonomous vehicles. For more details, the link to their website is https://blogs.upm.es/controlinteligente/welcome/.

1.3 Thesis Outline

The thesis is structured as follows:

- Chapter 2 – State of the Art:
  reviews relevant literature in social and emotional robotics, autonomy and emotional navigation, fuzzy logic applied to robotic systems, evaluation and statistical methods in HRI.

- Chapter 3 – TIAGo and its Navigation:
  describes the TIAGo robot used in this project, its hardware and software

architecture, sensor suite and the navigation stack employed for autonomous operation.

- Chapter 4 – Real-Time User Expression Recognition:
  details the deep-learning-based facial-expression recognition system developed and integrated into ROS.

- Chapter 5 – Emotional System Based on Fuzzy Logic and Personality Modulation:
  presents the architecture of the fuzzy emotional system, configuration files, controllers and ROS integration.

- Chapter 6 – Experiments:
  describes the experimental setup, procedure, statistical analysis and results from both pilot and final studies.

- Chapter 7 – Conclusions and Future Works:
  summarises the findings and outlines potential extensions of the emotional system.

Chapter 2

State of the Art

2.1 Social and Emotional Robotics

In recent years, the field of Human-Robot Interaction (HRI) has seen a growing emphasis on the development of robots capable of interacting with humans in a more natural and socially intelligent way. This evolution is closely linked to the recognized importance of emotions in cognitive processes and human interaction. The goal is to create artificial agents that not only perform tasks but can also understand and respond to the social and emotional nuances of human interaction. A crucial part of this progress stems from the adoption of psychological and neuroscientific models that explain how emotions influence cognitive processes, decision-making and social relationships, models that are now being translated into computational architectures capable of guiding the robot's behaviour.

2.1.1 The Role of Emotions and Social Interaction in Robotics

Emotions are inherently present in human language and play a crucial role in interactions. Rosalind Picard defined the concept of Artificial Emotional Intelligence (AEI), which recognises how the presence of an emotional model makes the interaction more natural, intuitive and trustworthy, highlighting the role of emotions in HRI. [1]
Over the past twenty years, scientific interest in this topic has grown exponentially—with a growing trend in publications and citations—in parallel with the development of increasingly sophisticated computational techniques for recognising and generating emotion in artificial agents. The evolution of AEI has made it possible to extend the notion of robotic emotion towards the construction of fully fledged artificial personalities. Indeed, one of the most significant achievements in

contemporary social robotics is the ability to endow a robot not only with emotions, but also with a genuine personality. Robotic personality—understood as the set of coherent and recognisable behavioural patterns expressed over time—plays a crucial role in shaping user trust, perceived predictability and overall interaction comfort. [2]
The relevance of personality also emerges clearly in Breazeal's studies on sociable robots: robots such as Kismet integrate a genuine motivational system that governs emotions, attention and expressive behaviour. This approach is still regarded today as one of the most advanced models of socio-emotional architecture. [3]

Moreover, the author distinguishes between different categories of robotic sociality:

- Socially evocative robots: designed to stimulate anthropomorphism for playful interaction, such as toys that use a caregiving model. However, the robot does not reciprocate social responsiveness;

- Robots with a social interface: use social signals and communication modes similar to those of humans to facilitate interaction with people (e.g., robotic avatars or robot guides in museums). Their social model of the user is often superficial, and their social behavior may be pre-programmed;

- Socially responsive robots: benefit from interaction by learning from humans through demonstration or other social signals, such as acquiring motor skills or elements of a proto-language. They are more perceptive but socially passive, responding to human initiatives without engaging proactively;

- Sociable robots: active social participants, equipped with their own goals and internal motivations. They proactively engage with people for mutual benefit and model individuals on a social and cognitive level. Their social behavior derives from a computational social "psychology".

The ultimate goal is to create robots capable of interacting with empathy, recognized as a key factor in HRI, improving quality of life and social connection. Empathic agents are defined as those capable of understanding the emotional state of a user or another agent and responding appropriately. The study of empathy can be approached from two perspectives: agents designed to evoke empathy in humans (agents as targets of empathy) and agents equipped to feel empathy toward others (empathic agents as observers).
Robot empathy is not limited to recognising the user's emotional state; it also encompasses broader cognitive processes such as response modulation and the ability to adopt another's perspective. The distinction between empathic mechanisms, empathy modulation, and empathic responses clarifies how a robot can move from a purely expressive reaction (for example, imitating a smile) to a more complex

form of emotional participation that takes into account context, interaction history, and social goals. [2] [4]

The potential applications of social and emotional robots are diverse and include: interactive toys, mediated communication through robotic avatars, guidance systems in museums or shopping centers, healthcare (e.g., nurse robots, robotic pets, emotional support for Alzheimer's patients), support for astronauts, family companionship, public services, and education. In particular, Socially Assistive Robots (SARs) are being studied for assisting children with autism spectrum disorders and personal assistant robots for accompanying isolated elderly individuals and young diabetic patients. [5] [6]

2.1.2 Emotional Systems and Computational Models

The primary goal of AEI is to enable machines to interpret human emotional states and adapt their behavior accordingly, generating emotionally informed and context-appropriate responses. This field is divided into two main research areas:

- Robot emotion: explores how robots can be endowed with the ability to recognize human emotions and, based on that recognition, generate their own emotional responses.
- Emotional robots: emphasizes the development of robots that not only understand human emotional signals but also express emotions through physical and behavioral modes.

Psychological theories on the generation of emotions are divided into three main frameworks: physiological theories (attribute emotions to bodily responses), neurological theories (link emotional experiences to brain activity) and cognitive theories (consider thoughts and mental evaluations as central to the formation of emotions). Although human emotions are subjective and often ambiguous, machines operate deterministically and can be programmed to simulate emotional responses based on algorithmic logic, allowing emotional processes to be designed in a structured and predictable way.
The development of emotional robots focuses on five main functional areas for practical implementation:

- speech recognition: enables robots to interpret human speech and derive emotional content;
- verbal interaction: facilitates two-way dialogue that responds to emotional tone;
- facial identification: recognizes emotional expressions based on facial patterns;

- emotional expression: allows the robot to exhibit its own emotional state;
- action coordination: regulates physical behavior based on perceived or internal emotional states.

Numerous robots developed over the past two decades embody these principles. [2]
For example, Aibo has evolved to include advanced emotion-related features such as facial recognition and the expression of over 60 emotional states. [2]
Paro, a therapeutic robot in the shape of a seal, uses sensors to respond to environmental stimuli and provide emotional support, particularly for Alzheimer's patients. [2]
Saya, a humanoid robot, employs facial muscle simulation to display a range of expressions, enhancing engagement. [2]
Kismet and Probo use multidimensional emotional spaces based on psychological theories. [2] [3]

Emotional space models serve as fundamental tools for both recognizing human emotions and synthesizing or regulating the robot's emotions. These models use geometric and topological frameworks to represent emotions in multidimensional spaces. The Valence–Arousal (VA) model, an important 2D representation, organizes emotions into quadrants based on their pleasantness and intensity. This framework has been further elaborated through models such as Positive Activation–Negative Activation (PANA) and Pleasure–Arousal–Dominance (PAD), the latter introducing a third dimension (dominance) to differentiate emotions. [2]
In all these models, emotion is understood as a continuum, allowing gradual transitions and nuanced changes in emotional expression. These models provide a theoretical basis for architectures that represent emotion as the result of a combination of event appraisal, memory, personality, and adaptation. In educational contexts, the effectiveness of multimodal models for estimating the user's emotional state has been clearly demonstrated. In their scenario, the iCat robot integrates information from facial expressions, posture, gaze direction, and task performance to infer the child's emotional valence and respond accordingly. Such integration is essential for creating genuinely responsive robots capable of recognising subtle emotional cues that would be difficult to infer from a single sensory channel. [5]

Finally, several specific computational models have been developed to simulate the emotional process in artificial agents: FLAME (Fuzzy Logic Adaptive Model of Emotions) based on fuzzy logic, FearNot! based on appraisal theories, ALMA (A Layered Model of Affect) which integrates the OCC model and Mehrabian's three-dimensional model, EMA (Emotion and Adaptation), and EP-Bot, a chatbot focused on conversational empathy. [7]
Fuzzy logic, in particular, is used to handle vague concepts and approximate

reasoning to represent emotions in a robot.

2.2 Autonomy and Emotional Navigation

On the one hand, autonomy enables the robot to act in a flexible and adaptive manner; on the other, it can generate concerns related to loss of control, particularly when it is not accompanied by transparent communication of the robot's intentions. Robotic emotions therefore play a fundamental role: they allow the robot to express internal states and modulate its navigation style, making the interaction more predictable and understandable, thus reducing potential perceptions of threat.

2.2.1 Autonomy, Social Acceptance and Perceived Threats

Recent studies show that the perception of robotic autonomy is closely tied to trust. [8] Research has examined in depth how perceived autonomy influences the acceptance of social robots. Findings indicate that seeing robots capable of making fully autonomous decisions—even to the point of ignoring a human command — can generate feelings of threat, both in realistic terms (safety, resource competition) and in identity - related terms (human uniqueness). This dynamic is reinforced by Western cultural narratives, rich in dystopian depictions of rebellious robots, which unconsciously shape user expectations. [8]
However, the effect is not unidirectional. Studies conducted in real HRI scenarios show that these concerns can be mitigated when the robot is transparent in its decision-making, when it explains the reasons behind its actions, or when it displays understandable emotional reactions that make its behaviour predictable. When a robot clearly explains why it acts in a certain way, or when it demonstrates awareness and respect for the user's emotional state, the perceived threat decreases significantly.
This aligns with Breazeal's observations [3], according to which a robot's social capabilities, especially those supported by expressive and behavioural cues, promote anthropomorphisation and enable users to interpret autonomy not as a risk, but as a relational capability. Similarly, anthropomorphisation, as shown in museum-based experiments with the Pepper robot, helps create a sense of familiarity and comfort, reducing perceived strangeness. [9]

2.2.2 Emotional Control and Emotion-Aware Navigations

From an engineering perspective, a robotic system endowed with emotions must be capable of translating its internal emotional state into observable physical behaviours. Fuzzy logic, already employed in several affective models, provides an especially effective mechanism for achieving a smooth transition between emotional

states and motor responses. [1]
Extending an emotional model to navigation means enabling the robot to modulate speed, interpersonal distance, trajectory and movement style according to both its internal state and the user's state. For instance, if a robot experienced an emotional condition analogous to fear, it might slow down, maintain larger distances from obstacles, or avoid sudden turns. This perspective is consistent with studies on expressive motion in socially interactive robots.
Emotion-aware navigation thus represents a natural extension of multidimensional affective models such as PAD. [9]
In these models, dominance and arousal can directly influence control parameters such as acceleration, turning radius and safety distance. A robot with low dominance or high uncertainty will tend to move cautiously; conversely, a positive emotional state with high dominance may result in more fluid and direct movements.
The integration of fuzzy logic, internal emotional state and navigation therefore enables the robot to evolve from a purely reactive system into an agent capable of expressing personality through movement. In the case of the TIAGo robot, employing a fuzzy controller that regulates not only emotions but also motion-planning parameters leads to coherent behaviour, in which affective expression and trajectory are aligned. This paradigm is particularly promising in social and assistive robotics, where the ability to convey safety, calmness or enthusiasm through movement is crucial for effective interaction with groups of people or in crowded environments.

2.3 Theoretical Foundations of the Real-Time Expression Detection Algorithm

Methods for the automatic recognition of emotions from facial images can be divided into two main categories: traditional computer vision approaches and deep learning-based approaches. Each category has distinct characteristics, with its advantages and limitations depending on the use case.

2.3.1 Traditional Computer Vision Approaches

Traditional computer vision approaches n manual feature extraction and require preprocessing of the images to identify relevant facial traits. Common techniques include:

- Histogram of Oriented Gradients (HOG): a method used to extract visual features from image contours, useful for detecting the shape and orientation of objects.

- Local Binary Patterns (LBP): an approach that identifies facial textures, useful for classifying emotions based on local patterns.
- Facial Landmark Geometry: analysis of key facial points (such as eyes, mouth, eyebrows) to measure and classify expressions.

These approaches have been used for decades in computer vision, with a strong emphasis on traditional classification through algorithms such as Support Vector Machines (SVM), Random Forests, and K-Nearest Neighbors (KNN). [16]
These methods have been effective in controlled environments but face limitations in more complex situations, such as partial expressions, variable lighting, or non-frontal face orientations.
The main drawbacks of these methods are the dependence on manual feature extraction and the difficulty in adapting to new environmental conditions, such as variable lighting and imperfectly defined facial expressions. However, they continue to be used in environments with limited image resolution or applications that require fewer computational resources.

2.3.2 Deep Learning-Based Approaches

In recent years, the adoption of Deep Learning has revolutionized the field of facial emotion recognition. Convolutional Neural Networks (CNNs) have emerged as the dominant method for facial image analysis due to their ability to automatically extract discriminative features without the need for manual feature engineering. [16]
CNNs can learn to recognize complex patterns and detailed features from large facial image datasets. These models are trained to identify emotions such as happiness, sadness, surprise, anger, fear, disgust and neutral. The adoption of CNNs has led to superior performance compared to traditional methods, especially under challenging conditions such as variable lighting, partial expressions or non-frontal face orientations.
Their effectiveness is particularly evident when trained on large and balanced datasets, making them the current standard for real-time applications.

2.4 Fuzzy Logic in the Robotic Context

In the field of assistive and social robotics, the ability to dynamically and adaptively modulate a robot's behaviour is a key requirement for achieving natural, effective, and empathetic interactions with human users. While recognizing the user's emotions forms the foundation for understanding the emotional state of the interlocutor, the real challenge lies in translating this perception into a coherent

and "human-like" behavioural response from the robot.
This need is particularly pronounced in personal assistance systems, where the robot is not only required to perform physical actions but also to communicate an internal state through its behaviour that is perceived as credible and context-appropriate.
To address this challenge, the work described in this project has chosen to adopt fuzzy logic as a central tool for regulating the internal emotional state of the TIAGo robot and modulating navigation parameters.
Fuzzy logic is particularly suited for handling uncertain, vague and qualitative information, allowing it to overcome the limitations of traditional systems based on rigid thresholds or binary decisions. It is especially ideal for modelling concepts that, in reality, do not manifest as clear-cut values but instead vary continuously and gradually, depending on a variety of sensory, contextual and psychological factors.

2.4.1 Theoretical Foundations of Fuzzy Logic

Fuzzy logic, introduced by Lotfi Zadeh in the 1960s, represents one of the main extensions of classical logic, specifically designed to handle uncertainty and the graduality typical of natural and cognitive phenomena. In contrast to traditional Boolean logic, which operates on discrete values (true/false, 0/1), fuzzy logic allows each element to have a degree of membership to a set, expressed as a continuous value between 0 and 1. This approach is particularly effective for modelling fuzzy concepts such as "happiness", "shyness", "attention", or "risk", which cannot be adequately described by rigid boundaries.
In social interactions, most emotional and behavioural states cannot be neatly classified into rigid categories but evolve fluidly based on subtle signals, contextual variations, and implicit intentions. For instance, modulating movement speed based on the user's emotional state, adjusting the safety distance from obstacles based on alertness levels or varying voice tone and gestures in relation to perceived emotion, these are all scenarios where fuzzy logic overcomes the limitations of traditional threshold-based systems.
From an operational perspective, a fuzzy system is composed of three key components:

- Fuzzification, which transforms numerical signals from sensors into degrees of membership to linguistic sets (e.g., "low", "medium", "high" emotional level);
- Inference engine, based on linguistic IF–THEN rules, which allows complex decisions to be made from expert knowledge;
- Defuzzification, which converts the result of the fuzzy inference into a numerical value usable by actuators or control modules of the robotic system.

The rule-based structure also provides considerable transparency in the decision-making process: each robot behaviour can be directly traced back to the linguistic rules that generated it, which is particularly important in assistive contexts where interpretability and safety are crucial requirements. [11] [12]

2.4.2 Significant Robotic Applications and Hybrid Approaches

The application areas in which fuzzy logic has demonstrated practical value include autonomous navigation and obstacle avoidance, manipulator control in the presence of non-linearities and dynamic uncertainties, sensor fusion and the modulation of social and emotional behaviours in interactive robots. In navigation, the use of fuzzy rules to merge heterogeneous information (sonar, LiDAR, vision) enables fluid maneuvering decisions that are tolerant to the imprecision characteristic of unstructured environments. Recent studies show how fuzzy and hybrid approaches are effective for collision avoidance and planning in dynamic scenarios. [13]
Regarding social robotics and neurorobotics, fuzzy logic is particularly suited for modelling emotional and behavioural descriptors (e.g., comfort levels, expressive intensity), allowing the robot to modulate motor and interaction parameters in a way that is perceived as natural by users. [14]
The literature describes both "pure" fuzzy controllers, successfully used for regulation and stabilization tasks, and hybrid architectures that combine fuzzy logic with machine learning to achieve adaptability and automatic tuning. Specifically, Takagi–Sugeno models are easier to integrate into optimization and adaptive control schemes, while Mamdani models provide greater semantic transparency in the rules. To reduce dependence on manual tuning, neuro-fuzzy approaches (which learn membership functions and rules from data) and evolutionary algorithms are used for rule base optimization. The ANFIS (Adaptive Network-based Fuzzy Inference System) architecture is a classic and widely cited example of the fusion between neural networks and fuzzy inference, useful for supervised learning and improving system adaptability in the presence of experimental data. [15]

2.4.3 Advantages of Fuzzy Logic

The integration of fuzzy systems into robotic control offers multiple advantages, which have contributed to their widespread use in areas such as mobile robotics, social robotics and adaptive systems:

- Management of Uncertainty: fuzzy logic allows for the processing of noisy, incomplete or variable data, typical of robotic sensory perception, without causing abrupt changes or incoherent behaviour;

- Adaptability and Flexibility: the robot's response can vary gradually and be customized based on changes in the user's emotional state, improving the quality of the interactive experience and avoiding robotic behaviours perceived as "rigid" or "artificial";
- Consistency with Natural Human Communication: linguistic concepts such as "move a bit slower", "maintain a comfortable distance" or "be more energetic" have a direct translation into fuzzy sets, making the robot's behaviour more intuitive for the user;
- Compatibility with Neural Models: fuzzy logic can be easily integrated with systems based on deep learning, as demonstrated in this project, where recognized emotions are used as inputs for modulating behavioural parameters;
- Ease of Customization: the structure of the rules and membership functions can be modified through simple configuration files, allowing the robot's behaviour to be tailored to different personalities or application scenarios.

These characteristics make fuzzy systems suitable for assistive scenarios where transparency and predictability are crucial. However, practical limitations remain: scalability of the rule base (which grows rapidly with the number of variables), the cost of manual tuning of membership functions, and the difficulty of providing formal safety guarantees without dedicated verification tools.

2.5 Evaluation Methods in HRI

This section presents the evaluation tools used in the study to measure both the social perception of the robot and the variations in the participants' emotional state.

Over the years, several psychometric instruments have been developed to assess dimensions such as anthropomorphism, social competence, emotional reactions, and global attitudes toward robots. These tools allow researchers to translate subjective impressions into quantitative data, enabling rigorous comparisons across conditions, robot behaviours, and studies in the broader HRI literature.

The approach adopted combines standardised questionnaires, widely used in the literature, with an analogue–numerical scale designed to capture short-term emotional changes. Among the various methods available, two instruments were selected as particularly suitable for the objectives of the present work: the Godspeed Questionnaire Series, which measures how participants perceive the robot's social and affective characteristics, and the EVEA scale, a validated tool for assessing immediate emotional states. Together, they offer a complementary perspective on the interaction, capturing both how the robot is perceived and how users feel after interacting with it.

2.5.1 Godspeed Questionnaire Series

To investigate participants' perception of the robot's characteristics, items were selected from the Godspeed Questionnaire Series, a standardized and validated psychometric instrument widely adopted in the HRI research community. Developed by Bartneck et al. (2009), it represents one of the main references for assessing perceptions and attitudes toward social robots. The questionnaire consists of five principal dimensions aimed at exploring complex psychological aspects: the tendency to attribute human-like qualities to the robot (Anthropomorphism), its perceived "liveliness" (Animacy), the degree of sympathy or pleasantness it evokes (Likeability), the perception of intelligence (Perceived Intelligence), and the perception of safety (Perceived Safety). The items in these scales are organized in pairs of opposing adjectives (for example, "Dead–Alive", "Mechanical–Organic", "Dislike–Like"), evaluated on a 5-point semantic differential scale.
The use of bipolar semantic scales allows for an intuitive and immediate assessment of users' perceptions, translating qualitative impressions into comparable quantitative data. [19]
In the present study, the Animacy and Likeability scales were selected, as they are particularly sensitive to variations in the robot's emotional behavior and personality, key aspects for the experimental objectives.

Although the Godspeed Questionnaire thus remains an effective tool for the quantitative assessment of social perception of robots, while ensuring comparability of results with those reported in the international literature, it is important to acknowledge more recent approaches that broaden and strengthen this analysis. In particular, the Human–Robot Interaction Evaluation Scale (HRIES; Spatola, Kühnlenz and Cheng, 2021), developed to overcome some psychometric limitations of earlier scales, adopts a multicomponent perspective grounded in theories of dehumanization and social cognition. It identifies four main dimensions — Sociability, Agency, Animacy, and Disturbance — capable of capturing both the attribution and deprivation of human traits in robots, including aspects such as intentionality attribution and discomfort. Although it was not employed in the present study, the HRIES represents a valuable methodological reference for future research aiming to extend the analysis of human–robot perception beyond the sole dimensions of liveliness and likeability. [20]

2.5.2 EVEA

To analyse the impact of the interaction on participants' emotional state, an adapted version of the EVEA (Escalas Visuales Analógicas de Estado de Ánimo) scale was employed.
This instrument is a validated psychometric tool for assessing current emotional

states and is widely used in both clinical and experimental settings. In its original format, the scale employs 10-cm visual analog scales (VAS) for each emotion; however, for greater practicality and ease of digital administration, this study used a numerical scale ranging from 0 to 10, which retains equivalent psychometric validity.
The EVEA scale (Sanz et al., 2003) is designed to provide a quantitative, immediate and multidimensional measure of an individual's emotional state at a specific point in time. [21] The choice of a visual analog or equivalent numerical scale allows for the detection of even slight variations in emotional changes, making this instrument particularly suitable for experimental studies involving shifts in emotional state in response to specific stimuli. This is especially relevant in the context of interactions with social robots, where emotional variations may emerge within a single session.

2.6 Statistical Models of ANOVA and mixed ANOVA

The analysis of Human–Robot Interaction data often requires statistical methods capable of detecting differences between experimental conditions and of modelling how participants' responses vary across within-subject and between-subject factors. For this reason, the present study employs Analysis of Variance (ANOVA) techniques, which represent the standard inferential framework for comparing group means in controlled experiments. In particular, the mixed-design ANOVA is well suited for studies in which each participant is exposed to multiple robot behaviours while also belonging to distinct personality categories.

ANOVA is a family of statistical techniques widely used in experimental studies to compare means obtained across different groups or experimental conditions. [22]
Historically introduced within the field of experimental statistics by R. A. Fisher, it allows one to assess whether the observed differences between means are too large to be attributed solely to chance, while simultaneously accounting for the variability within groups. In practice, it evaluates whether the variability observed between group means is significantly greater than the variability observed within groups. [23]

Conceptually, ANOVA compares two types of variability:

1. Between-groups variability, the extent to which the means of the different groups deviate from the overall mean;

2. Within-groups variability, the extent to which individual data points differ from their own group mean.

The ratio between these two quantities is expressed by the F statistic, which lies at the heart of the analysis:

$$F = \frac{MS_{between}}{MS_{within}}$$

Where $MS_{between}$ and MS_{within} are the mean squares, i.e., sums of squares normalised by their respective degrees of freedom.

- If $F \approx 1$, the differences between groups are comparable to random variability $\rightarrow$ no significant effect;
- if F is large, the differences between groups exceed random noise $\rightarrow$ a likely real effect.

To interpret the results, three key indices are considered.

- F statistic: the ratio between the variability explained by the experimental factor (e.g., "Condition") and the unexplained variability (error). The higher the value of F, the more likely it is that a real difference exists between the group means.
- *p-value*: the probability, under the null hypothesis, of observing a value of the F statistic at least as large as the one computed. In this work, we adopt the conventional significance threshold $\alpha = 0.05$: if $p < \alpha$ the observed difference is considered statistically significant. However, the p-value does not measure the practical importance of the effect, it only reflects its improbability under chance. Very large samples can render even small effects "significant", whereas small samples may fail to detect real differences.
- Partial eta squared, η_{p}^2: an effect-size index that quantifies how much a factor contributes to the total variation observed in the data. This index ranges from 0 to 1 and can be interpreted as a "percentage of influence" of the effect. Practical guidelines (Cohen) suggest: 0.01 = small effect, 0.06 = medium effect, 0.14 = large effect. [24]

In the present study, a mixed experimental design was adopted, combining two types of factors. [25]

1. Within-subjects, conditions in which the same participants are measured repeatedly (in this study, the Condition factor with two levels: Neutral and Personalised);
2. Between-subjects, where different participants belong to distinct levels (in this study: the two independent subgroups A and B, or the labels shy/intense).

To conclude, mixed ANOVA is particularly appropriate when one wishes to test simultaneously:

- the main effect of Condition, namely the average difference between Neutral and Personalised;
- the main effect of Group, namely the average differences between the independent subgroups A and B;
- the Group × Condition interaction, indicating whether the difference between Neutral and Personalised varies as a function of group membership.

As stated earlier, a conventional significance threshold of $\alpha = 0.05$ was adopted for all analyses.
In Section 6.4 (Results), the main results will be reported in terms of F values, *p-value* and η^2_{p} for each effect considered (Group, Condition, Group × Condition).

Chapter 3

TIAGo and its Navigation

TIAGo (Take It And Go), developed by PAL Robotics, is a mobile–manipulator robotic platform designed for advanced research in fields such as human–robot interaction, autonomous navigation, and manipulation in shared environments. The design philosophy behind TIAGo is inherently modular. The structure can be configured with one or two arms, different end-effectors (such as a parallel gripper or the five-finger humanoid hand named Hey5), alternative locomotion bases, and computing units of varying performance, while maintaining deep integration with the open-source ecosystem and with ROS, the framework that facilitates its use and extensibility within both academic and industrial communities [26].

3.1 TIAGo One Arm

The model available at the Centro de Automática y Robótica (CAR) of the Universidad Politécnica de Madrid (UPM), and used in this project, is the TIAGo One Arm version, equipped with a parallel gripper.
In this single-arm configuration, the platform combines mobility and manipulation in a compact body designed to operate within human-populated spaces, offering researchers a solid foundation for experimenting with perception, planning and control algorithms.

Figure 3.1: Model of the PAL Robotics TIAGo one-arm platform used for the project

3.1.1 Mechanical Structure and Sensor Suite

The structure of TIAGo One Arm is organised into four interconnected modules: the mobile base, the torso with lifter, the manipulator arm with and the sensory head.
The base (known as PMB2) is compact yet robust, designed for indoor use, and equipped with a differential-drive locomotion system that enables stable movement and effective manoeuvrability in confined spaces. It houses the locomotion motors, odometry sensors and power electronics; according to official documentation, the base has a diameter of approximately 54 cm and the entire system weighs around 72 kg.
Above the base is the torso, which incorporates a vertical lifting mechanism (lifter) providing a stroke of about 35 cm, yielding a total height range between 110 and 145 cm. This enables adjustment of the robot's operational height, improving perceptual coverage and reachability during manipulation tasks [27].
Attached to the torso is the manipulator arm, with a maximum reach of approximately 87 cm and a payload capacity of up to 3 kg (excluding the end-effector). It features a combination of 4 degrees of freedom (DoF) in the arm and 3 DoF in the wrist, for a total of 7 DoF. As mentioned earlier, in the configuration used for this project the robot is equipped with a parallel gripper, a reliable, precise and lightweight tool suitable for grasping and interacting with everyday objects, making it appropriate for assistive tasks, collaborative environments and domestic

scenarios. For more sophisticated interactions, optional modules such as compliant actuators and force/torque sensors at the wrist can be integrated to increase safety in physical interactions.

The platform includes a heterogeneous sensor suite, distributed between the base and the robot's head, designed to meet the primary needs of navigation and manipulation. It features a 2D LIDAR mounted on the front of the mobile base, which measures distances in a horizontal plane and allows the robot to acquire a planar scan of the environment for mapping and obstacle avoidance. Supported configurations include Hokuyo or SICK sensors. To maximise the effectiveness of the RGB-D sensor, the robot can adopt specific postures (slight torso lifting and head tilt) that extend the frontal field of view and improve scene coverage. Additionally, in scenarios where sensors may produce false readings or fail to detect hazards (e.g., glass panels or transparent surfaces), the Map Editor allows insertion of virtual obstacles, protecting critical areas of the map independently of sensor data. Complementing this, three ultrasonic sensors mounted on the rear of the base provide short to mid range collision avoidance ($0.03 \div 1$ m), and a centrally mounted IMU (Inertial Measurement Unit) supports odometry.
The head hosts an RGB-D camera (providing colour and depth information), essential for 3D scene perception and detecting obstacles that lie outside the LIDAR plane, as well as two stereo microphones for audio acquisition and applications such as speech recognition.
This sensor combination is designed to provide complementary information: the 2D LIDAR excels in planar scanning and 2D map construction for navigation, while the RGB-D camera is fundamental for detecting off-plane obstacles (tables, chairs, low objects) and for tasks requiring 3D perception, such as object recognition and manipulation [28].

3.1.2 Software Architecture and Control Stack

From a software perspective, TIAGo runs on Linux Ubuntu LTS with a Real-Time (RT) kernel, while the robotic framework underpinning its software environment is ROS (Robot Operating System). The typical pipeline governing autonomous locomotion involves the acquisition and preprocessing of sensory data, map construction or updating (mapping/SLAM), probabilistic pose estimation (localization), global path planning to the target, and the generation of local velocity commands that comply with the platform's kinematic and safety constraints.
PAL Robotics provides a set of nodes, configuration files and additional tools that simplify the use of this stack, including an integrated map editor within RViz, a system for managing points and areas of interest, and dedicated action servers for navigation primitives. The entire system is designed to be easily extensible, enabling

researchers to intervene at any stage of the pipeline, from raw data processing to motion execution.

3.2 Mapping, Localization and Path Planning

The autonomous navigation of TIAGo is managed through the robust ROS 2D navigation stack, a software architecture that allows the robot to orient itself, build maps, and plan paths in dynamic environments by exploiting data from the front LIDAR and other onboard sensors. The environment is typically represented as a 2D Occupancy Grid Map (OGM), built from laser scans and updated using odometry. Mapping can be performed in teleoperated mode, driving the robot through the environment with a joystick to acquire scans, or in online SLAM mode, with algorithms that progressively integrate observations during exploration to build and update the map.
Once the map is constructed, localization is addressed through a probabilistic approach. The most widely used algorithm, Adaptive Monte Carlo Localization (AMCL), represents the robot's pose as a distribution of particles that is continuously updated by fusing odometry and laser observations. This mechanism provides robustness against sensor noise and enables the robot to correct the inevitable odometric drift accumulated during navigation.
With a map available and the pose estimated, path planning takes place at two levels. The global planner operates on the static or quasi-static map and computes an optimal or suboptimal path to the goal, typically using grid-based search algorithms such as A*, whose classical formulation and simplicity guarantee solutions consistent with the defined cost function [29]. However, the generated path is only a high-level indication: its real execution in dynamic environments requires a local planner capable of translating the trajectory into velocity commands while respecting kinematic limits, platform inertia, and the presence of sudden obstacles. Methods such as the Dynamic Window Approach (DWA), or optimisation-based planners, are typically used to generate commands that are sent to the base controller. Supporting this, the costmap structure (global and local costmaps) aggregates sensory information and defines forbidden or penalised areas, simplifying the planner's task in evaluating collisions and safety margins.
From an operational perspective, TIAGo integrates functionalities designed to ensure safety and long-term autonomy. For use around people, the combination of multiple sensors and the careful tuning of costmaps and local planners is crucial to ensure behaviour that is socially aware, maintains appropriate interpersonal distances, and reacts effectively to human movements.

3.3 Operational use of the PAL Robotics stack and the RViz tool (experimental procedure)

In the project described in this thesis, the navigation stack provided by PAL Robotics and the RViz visualization interface were used to interact with the system. The adopted operational procedure began with the selection of the map to be used through the map-management service. Specifically, to load the map of the environment used during the experiments (described in more detail in Chapter 6), named “hall_map”, the following command was executed:

```
rosservice call /pal_map_manager/change_map "input:  'hall_map'"
```

After selecting the correct map, RViz allows the robot’s localization to be initialised using the “2D Pose Estimate” tool whenever the robot’s estimated heading differs from the real one. This tool publishes an initial pose estimate (x, y, θ), helping AMCL converge to the correct region of the distribution. Once the robot is properly localised on the map, the “2D Nav Goal” tool (the arrow in RViz) can be used to specify the target pose toward which the robot must move. The navigation stack receives the goal, computes the global path, and the local planner executes the required velocity commands to reach it.

Below is a screenshot from RViz showing the CAR laboratory map where TIAGo operates, the laser scan, and the trajectories. It is provided as an example to illustrate a typical visualization during experimental navigation.

Figure 3.2: Screenshot of the RViz interface with the CAR laboratory map

Chapter 4

Real-Time User Expression Recognition

Facial expression recognition represents a fundamental step in enabling the robot to adapt its emotional state to that of its interlocutor. Indeed, facial expressions are one of the primary channels through which humans convey emotions, and embedding this perceptual capability into the robot allows for more natural and empathetic interaction.
After an overview of the fundamental techniques, this chapter provides a detailed account of the process followed for the integration of a real-time facial expression recognition system, based on advanced deep learning and computer vision methods. The system used for this integration was sourced from the GitHub repository https://github.com/Harshxth/Real-Time-Expression-Detection.
This system is the result of the integration of emotion classification models with tools for the acquisition and analysis of static images, enabling the automatic interpretation of an individual's emotional state through the simple input of a webcam, initially the one embedded in the laptop used for integrate the algorithm, and subsequently the camera integrated into the TIAGo robot.

4.1 Deep Learning Approach for Real-Time Facial Expression Recognition

As outlined in Chapter 2 section 2.3, the recognition of facial expressions is rooted in fundamental principles of computer vision and machine learning For this project, a deep learning approach was adopted, as it provides greater robustness and flexibility, particularly when applied with the robot's camera, ensuring reliable accuracy even under varying lighting conditions and facial orientations.

CNNs are designed to efficiently detect and classify visual features within images. In the specific context of facial expressions, these networks are trained on large datasets of human faces, each associated with a particular emotion: angry, disgust, fear, happy, sad, surprise and neutral. Unlike pre-trained models such as VGG16 or ResNet, the architecture developed for this project is a tailored CNN, specifically designed to meet the needs of facial emotion classification.
The model consists of several convolutional layers, which progressively extract features from the input images, starting with basic patterns like edges and progressing to more complex textures and shapes. The convolutional layers are followed by max-pooling layers, which reduce the spatial dimensions of the feature maps, enhancing computational efficiency and improving the model's robustness against variations in image orientation and lighting. To prevent overfitting, dropout is applied at each convolutional and fully connected layer. This helps the model generalize better, avoiding excessive adaptation to the training data. After feature extraction, the model passes through fully connected layers, which map the learned features to the output classes. The final output layer uses a softmax activation function to classify the facial expression into one of seven categories: angry, disgust, fear, happy, sad, surprise or neutral. The entire system is built using deep learning frameworks like Keras and TensorFlow, ensuring flexibility and ease of integration with real-time processing pipelines.
In the context of image and video processing, the system leverages the OpenCV library, a widely adopted framework in computer vision, which plays a central role in the pipeline, as it enables the acquisition of frames directly from the webcam, supports the preprocessing of images, and performs the detection of faces within each frame.

4.2 Integration Procedure

The integration of the recognition system was developed starting from the GitHub repository: https://github.com/Harshxth/Real-Time-Expression-Detection.
The first stage of the work focused on the preparation and training of the emotion classification model.
For this phase, an interactive notebook named `ModelTraining.ipynb` was used, within which the dataset, consisting of facial images categorized by emotional class, was collected and organized.
The images were subjected to several preprocessing operations, including grayscale conversion, resizing, and value normalization. These steps ensured consistency across the dataset and enhanced the model's ability to learn relevant features. Subsequently, the Convolutional Neural Network was structured and trained, while accuracy and loss metrics were continuously monitored to evaluate its performance

on both validation and test data.

Figure 4.1: Sad expression

Figure 4.2: Disgust expression

Figure 4.3: Surprise expression

Figure 4.4: Happy expression

Once the training process was completed, to enable the use of the model on devices with limited resources, such as microcontrollers or embedded systems, the trained network was saved and subsequently converted into a compatible format. The trained model can be transformed into a header file (.h), which stores the parameters and network structure in binary form. This conversion, carried out through dedicated tools, allows the model to be integrated directly into the source code of applications written in C or C++.
However, in the specific case of this project, the model was converted to HDF5 (.h5) format, generating the file `emotiondetector.h5`. This format is particularly well suited to the Python environment, as it efficiently stores both the structure and the weights of the neural network, making it possible to load and use the model through libraries such as Keras and TensorFlow. This solution ensured a more straightforward management of the model while also facilitating its integration within the Python-based inference and visualization pipeline.

The development of the pipeline required the coordination of several fundamental steps, each designed to ensure efficient and real-time operation of the system:

1. Integration of the converted model into the final project;
2. Real-time acquisition of frames through the webcam, providing a continuous stream of images to be analyzed;
3. Detection and extraction of the face within each frame, isolating the region of interest from the image;
4. Local preprocessing of the inputs, required to ensure compatibility between the input format and the model;
5. Analysis of the detected face by the deep learning model, which outputs the recognized emotional category;
6. Immediate visualization of the detected expression, achieved by overlaying a simple label directly onto the video stream.

This process is continuously repeated for each acquired frame, thus ensuring emotion recognition that is constantly updated and responsive.

4.3 Technical Considerations

From a technical standpoint, the main challenges concerned, on the one hand, ensuring genuine real-time processing by optimizing inference times and minimizing the latency between frame acquisition and result visualization, and on the other hand, adapting the algorithm to operate correctly with the robot's integrated camera.

In the early phase, the facial expression recognition system was designed to operate exclusively with the computer's integrated webcam, using the OpenCV library for real-time frame acquisition. Figures 4.5 and 4.6 show some of the results obtained during the preliminary tests carried out with the laptop's webcam.

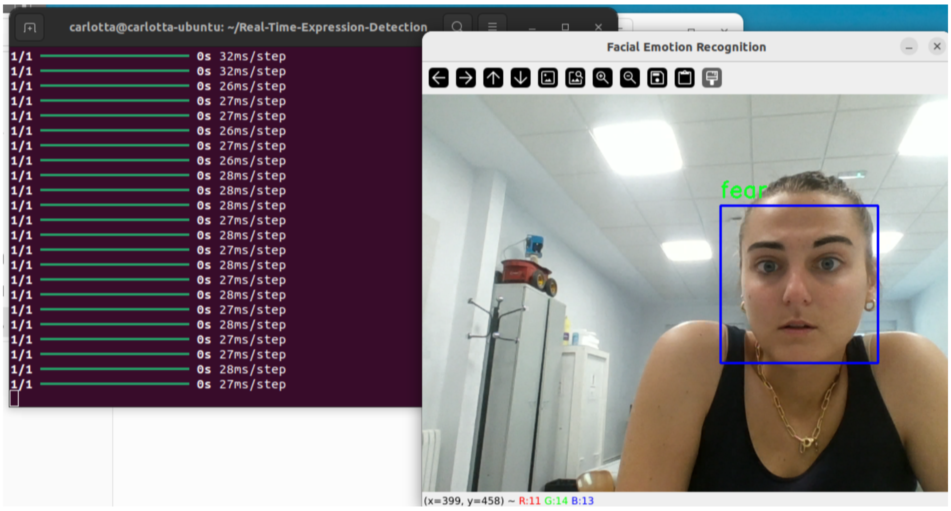

Figure 4.5: Interface of the facial expression recognition system during preliminary tests performed with the laptop's webcam – expression: fear

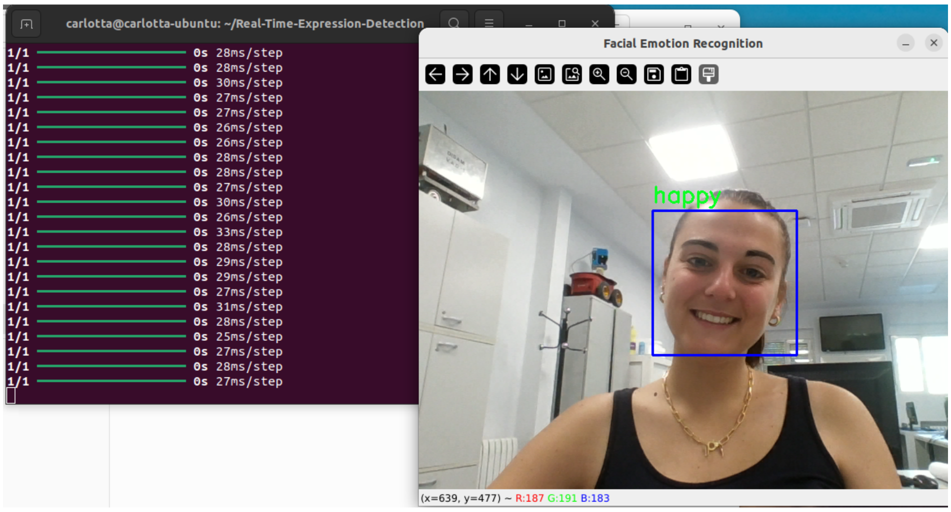

Figure 4.6: Interface of the facial expression recognition system during preliminary tests performed with the laptop's webcam – expression: happy

Subsequently, in order to enable the deployment of the emotion detection pipeline on the TIAGo robot, it was necessary to adapt the system to the new video source, the camera installed on the robot. This adaptation was implemented through the creation of the file *ros.py*, which allows the reception and processing of images transmitted by the ROS topic associated with the robot's camera. This step represented a crucial phase, as it allowed the algorithm to be tested under conditions closer to real-world usage, moving beyond the controlled context of preliminary trials.
The technical challenges encountered during this stage included:

- Lighting variations: TIAGo operates in different environments, where illumination is not uniform.
- Perspective and distance: the robot's camera is positioned at a different height compared to the laptop's webcam, and it often captures users' faces from non-frontal angles. This required verifying the model's ability to recognize emotions despite slight head rotations or changes in distance.
- Robot movement: during navigation tests, TIAGo does not remain stationary but performs movements and rotations. This introduces difficulties due to rapid changes in framing, which may affect recognition accuracy.

To address these issues, additional tests were conducted in diverse scenarios, with users placed at varying distances and under different lighting conditions. The results showed that, although there was a slight decrease in accuracy compared to the preliminary tests, the system maintained adequate performance, confirming

the feasibility of emotion recognition through the TIAGo's camera. In particular, the algorithm demonstrated a good ability to recognize more distinct facial expressions, such as happiness and surprise, which are characterized by clear and easily identifiable features. The neutral and disgust classes also exhibited generally stable performance, with medium-to-high confidence levels. In contrast, emotions with subtler visual characteristics, such as fear and sadness, achieved lower accuracy. In some cases, the algorithm tended to misclassify these expressions as neutral, highlighting the need for further training phases or the use of larger and more balanced datasets.
Figures 4.7 and 4.8 present some examples of facial expression detection involving multiple individuals within the field of view of TIAGo's integrated camera.

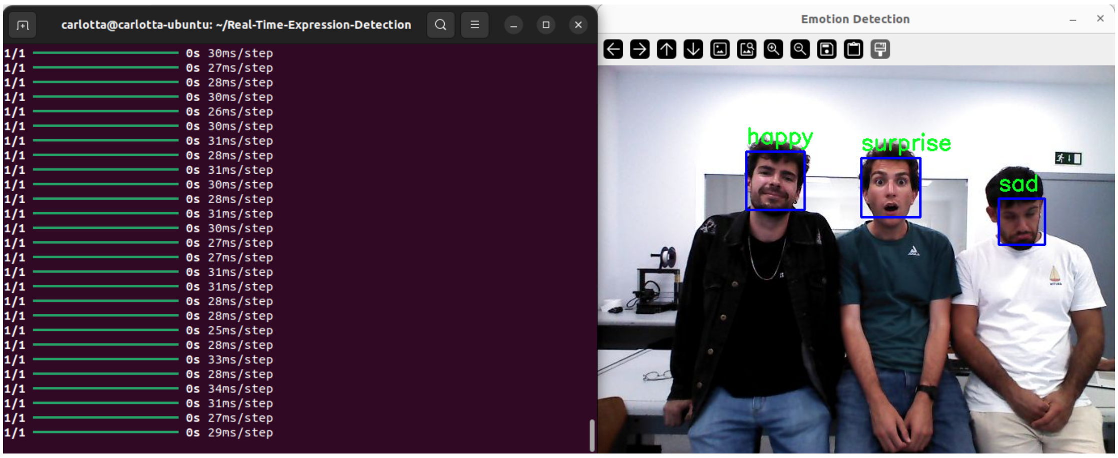

Figure 4.7: Interface of the facial expression recognition system during tests performed with the TIAGo robot's integrated camera – expressions: happy, surprise and sad

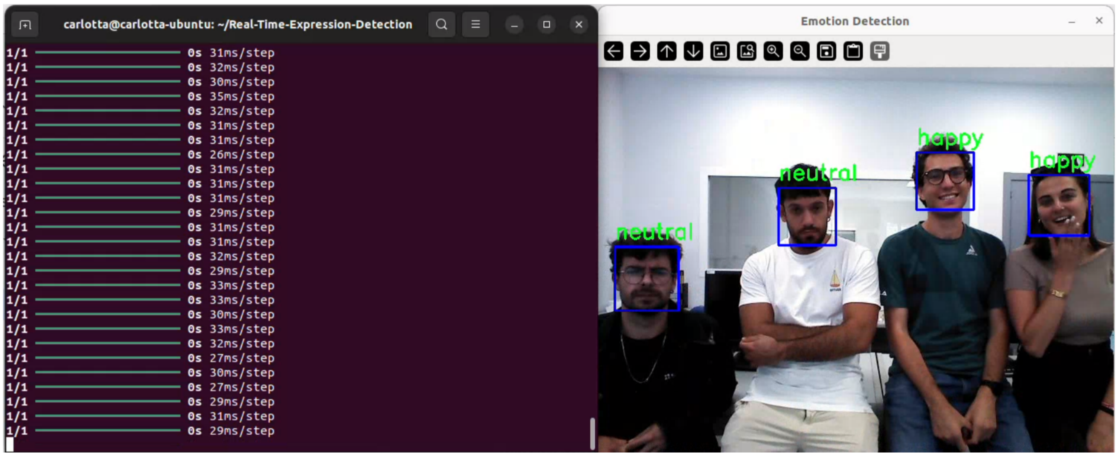

Figure 4.8: Interface of the facial expression recognition system during tests performed with the TIAGo robot's integrated camera – expressions: happy and neutral

To properly manage dependencies and different execution environments, a dedicated *venv* ("virtual environment") folder was created within the project directory, which must be activated before launching the various ROS nodes connected to the vision system. It represents a Python virtual environment, that is, an isolated space within the filesystem where the packages and libraries required for the project are installed independently.
The use of a virtual environment prevents conflicts between different versions of libraries or dependencies required by other projects running on the same machine. In practice, by activating the virtual environment (through the command `source venv/bin/activate`), the code is executed exclusively with the libraries installed within that specific folder, without interfering with the system-wide ones.
This solution is particularly useful in software development and in the management of complex projects, such as those involving the interaction with ROS and deep learning models. In this case, it allowed the maintenance of a clean, stable, and reproducible working environment, while facilitating the execution of the various scripts and ROS nodes without incurring errors due to dependency incompatibilities.

Chapter 5

Emotional System Based on Fuzzy Logic and Personality Modulation

After developing and validating the facial-expression recognition system described in the previous chapter, the next step was the integration of this perceptual module into a fuzzy-logic-based emotion-regulation system capable of translating recognised stimuli into observable behavioural variations in the robot. Integrating an affective model into social robots is a crucial step towards making human–robot interaction more natural and effective. Emotions act as a bridge between external stimuli and the robot's behaviours, modulating its responses according to the context and the user's perceived emotional state.

Among the various computational approaches available for modelling emotions, fuzzy logic, extensively explored in the literature for representing complex and uncertain phenomena, has proved particularly suitable thanks to its ability to handle uncertainty, express rules in natural language, and ensure gradual transitions between emotional states. This allows one to describe intuitively how stimuli of different kinds (internal, environmental or social) influence the robot's emotions and, consequently, its behavioural outputs. This feature was essential for involving a psychology student in the process, who reviewed the fuzzy rules that modulate the robot's emotions in response to incoming stimuli.

For the theoretical foundations of fuzzy logic, see Chapter 2 section 2.4.

The fuzzy system was designed for the management and simulation of the robot agent's emotional states, endowing it with flexible and adaptive affective processing so that it can respond naturally to interactions and to stimuli arising from the surrounding environment.

The system was developed with reference to the framework proposed by Fernández-Blanco et al. (2023) [1] and to the code structure available in the Github repository Grupo-de-Control-Inteligente/potato-v2 (https://github.com/Grupo-de-Control-Inteligente/potato-v2 , in particular the folder files/software/modules/emotional_manager), which were used in a research study on the "Potato" robot at the CAR of the Universidad Politécnica de Madrid. This repository proposes a modular architecture for managing emotional states via fuzzy logic, characterised by a clear separation between the definition of emotional variables, the inference rules, and the management of the robot's internal state. This system was extended and adapted to the specific needs of the project, while preserving its modular and flexible design philosophy.

This chapter presents the design principles and implementation of fuzzy logic within the robotic system. The developed software and code architecture, from the configuration files of the fuzzy controllers through to the centralised management of the emotional state and its final integration into the main ROS node is described. Through this pipeline, the robot adapts its speed, obstacle-distance handling, and other navigation parameters in a manner consistent with the selected personality and with the user's perceived emotion, thereby making the interaction more realistic and personalised.

To ensure full transparency, reproducibility and ease of reuse, all the codes developed for this project, including the fuzzy controllers, the emotional state manager, the ROS nodes and the navigation–modulation modules, are publicly available in a dedicated GitHub repository:
https://github.com/carlottapifferi2002/Real-time-fuzzy-emotional-control-system-for-social-robotics
The repository contains the complete source code, configuration files, and documentation required to run the entire emotional control architecture on the TIAGo platform or to adapt it to other robotic systems. Moreover, for users who intend to deploy or experiment with the proposed system, Annex A includes a detailed user manual describing the required dependencies, the execution pipeline and the operational steps necessary for interacting with the robot through the implemented emotional-control framework.

5.1 Architecture of the Fuzzy System

To implement the emotion regulation model described in the previous section, a modular software architecture based on fuzzy logic was developed. The primary objective is to translate the affective information detected by the facial-expression

recognition system into behavioural, and subsequently navigation, parameters that can be used by the TIAGo robot's planner and motor controllers, ensuring a response consistent with the robot's internal state.
The adopted model relies on a dimensional representation of emotions. In its full formulation, it comprises six principal dimensions (mood, sensory, affective, alertness, interest and expectancy). However, to make the implementation on TIAGo clearer and more manageable, while maintaining a focus on navigation, this project considers only two dimensions, which are sufficient to generate meaningful behavioural variations:

- Mood: happiness–sadness axis,
- Alertness: fear–calm axis.

The outputs of the affective system translate into behavioural modifications, specifically:

- Linear navigation speed,
- Navigation stack tuning, which accounts for obstacle distance (via the inflation radius parameter) and manoeuvre aggressiveness (angular velocity to control sharper or gentler turns).

The operation of the model is schematised in Figure 5.1. The input stimuli, namely the user's facial-expression recognition output and the previous Alertness and Mood states, are processed by transition tables (state tables) that update the values of the two emotional dimensions. These updated values are then translated, via fuzzy output tables, into the robot's navigation parameters related to its motion (speed and tuning).

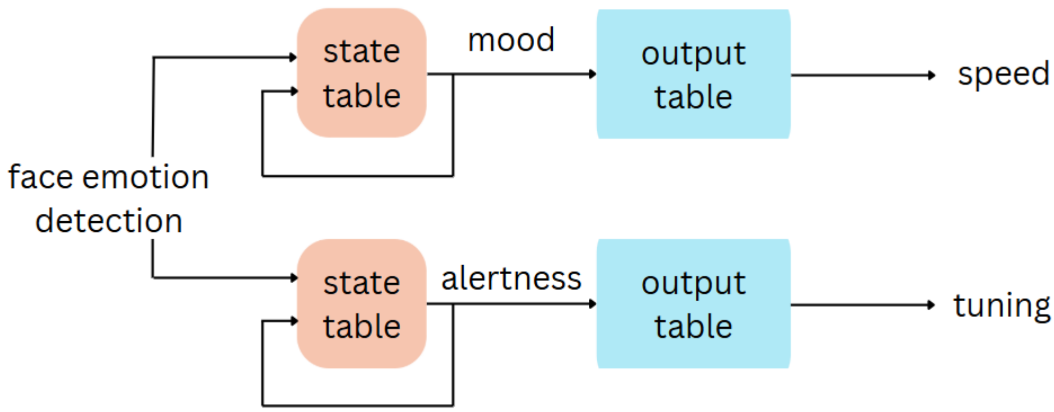

Figure 5.1: Architecture of the emotional fuzzy system: face emotion detection → state tables that update the internal emotional dimensions (mood and alertness) → output tables that update the navigation parameters (speed and tuning).

Fuzzy logic enables the robot's behaviour to be expressed through intuitive linguistic rules such as:

- If the user appears angry and the robot's mood is "standard", then reduce the mood value to "sad".
- If alertness is calm and the robot has "low" tuning, then increase tuning to "medium" (slightly reducing the distance from obstacles and slightly increasing the angular velocity).

The linguistic variables (e.g., low, standard, happy, slow, calm) are described by triangular and trapezoidal membership functions. The inference process is of the Mamdani type, with defuzzification via the centroid method, ensuring a continuous and gradual response.
A central aspect of the model is the definition of robotic personalities that modulate the mapping between stimuli and outputs. These were encoded through distinct sets of fuzzy tables, examples of which are provided in the following sections.

In this thesis, two variants were implemented:

1. Shy personality: characterised by slower, cautious movements, conservative responses in the presence of negative emotions, and a greater safety distance from obstacles. This configuration exhibits lower behavioural variability, prioritising stability and prudence;
2. Intense personality: characterised by dynamic navigation, with higher linear speeds and rapid, pronounced responses to stimuli. This configuration tends to approach obstacles more closely and react more decisively to changes in emotional state.

To ensure clarity and maintainability, the entire system was implemented in Python and integrated into the ROS (Robot Operating System) infrastructure. Modules communicate via ROS topics and services, following a clear and repeatable processing pipeline.

The overall structure comprises five main components, each with well-defined responsibilities:

- JSON configuration files — encode the fuzzy logic by defining membership functions, inference rules, and linguistic labels;
- `fuzzy_core.py` — interprets the generic fuzzy logic and performs inference according to the rules defined in the configuration files. It is responsible for creating the fuzzy systems for each variable, setting membership functions and linguistic rules, and computing the fuzzy output based on the inputs received;

- `emotional_manager.py` — coordinates the various fuzzy controllers and keeps the robot's overall emotional state up to date;
- `fuzzy_node.py` — the main ROS node responsible for communication between the perception system and the rest of the robotic platform (motor control modules). This node receives data from the emotion-recognition system, invokes the emotional_manager for fuzzy modulation, publishes the resulting variables on ROS topics, and dynamically updates navigation parameters;
- `parameters.py` — provides the mapping from the computed emotional state to navigation parameters (speed, inflation radius, angular gain).

This separation between logic (JSON files) and code makes the system highly flexible: new personalities or rules can be trialled simply by modifying the configuration files (membership functions or rules), without direct changes to the source code. With this approach, the software component of the fuzzy system is not only an experimental tool but also a reusable platform for comparing behaviours, measuring the impact of emotion regulation on navigation, and evaluating how different personalities affect the acceptability of human–robot interaction.

To clarify the relationships between the conceptual components described above and their software implementation, figure 5.2 summarises the high-level organisation of the emotional-control architecture. Before introducing the implementation details of each module, this diagram highlights the logical dependencies between components and the flow of information across the system.

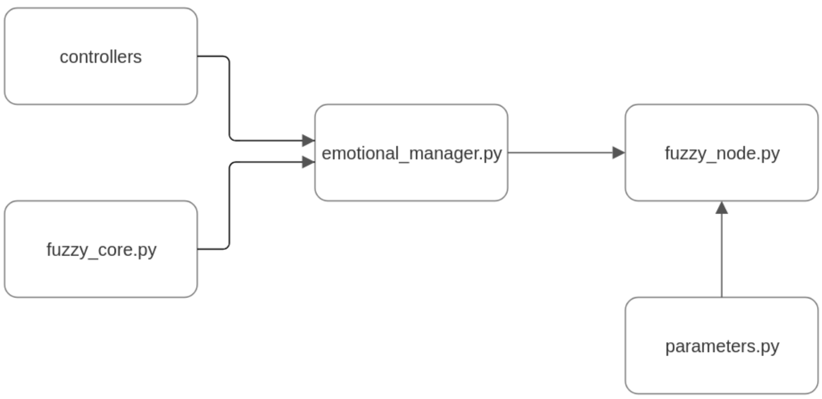

Figure 5.2: Functional overview of the emotional-control system, illustrating the dependencies between the main components and the flow of information.

5.2 Software Implementation of the Emotional-Control Architecture

The following subsections present the software modules that implement the emotional-control architecture. Each file is described in terms of its role, internal logic, and interaction with the other components in the pipeline, reflecting how the conceptual architecture introduced earlier is translated into code.

5.2.1 Fuzzy configuration files

The first layer of the software architecture devoted to the robot's emotional and behavioural regulation consists of the fuzzy-controller configuration files, stored in JSON format within the "controllers" folder. These files constitute the foundation of the entire fuzzy system: they define the input and output variables with their numerical ranges, the membership functions, the inference rules, the mapping between numerical values and linguistic labels, and optionally the temporal-delay parameters that govern the system's responsiveness.
Each JSON file describes the behaviour of a single emotional dimension of the fuzzy model, specifically:

- mood — encodes the robot's overall affective state along the happiness–sadness axis;
- alertness — quantifies the system's tension or calm in response to perceived stimuli (fear–calm axis);
- body_speed — regulates the robot's linear speed as a function of the mood state;
- tuning — modulates navigation parameters (inflation radius, safety distance and angular velocity) as a function of the alertness state.

Each JSON configuration file is organised according to a precise structure:

- inputs: list of input variables for the fuzzy system, each characterised by a value range, a set of membership functions (sets) and a default value used at initialisation.
- output: definition of the output variable with its range and corresponding membership functions. For example, the variables mood and tuning have a continuous domain from 0 to 120, partitioned into linguistic labels such as very_sad, sad, standard, happy, very_happy or low, medium, high. The variable body_speed instead uses a physical interval in metres per second

(0.1÷0.9) m/s with labels very_slow, slow, standard, fast, very_fast. Finally, alertness defines the robot's level of tranquillity through the labels calm, standard, and fear, and spans a range from 0 to 100.

- rules: the set of inference rules expressed in natural language. Each rule specifies a combination of conditions on the inputs and the corresponding output value. The rules are expressed in tabular form and allow one to encode both dependence on the previous state (memory) and the immediate reaction to perceptual stimuli (facial labels).

- mapping: the association between numerical output intervals and qualitative labels, used to translate fuzzy values into behavioural parameters.

- temporal stabilisation parameters (delay_const and use_delay): regulate the inertia with which the emotional state varies. For the intense personality, these parameters are tuned to balance reactivity and stability (for example, mood has a very low delay_const of 3 to respond more rapidly to visual cues, whereas alertness uses delay_const = 20 to smooth sharper variations). In the tuning module, for both personalities, use_delay = true with delay_const = 10, to avoid sudden changes in navigation parameters. By contrast, body_speed is configured with use_delay = false because the body's linear speed is updated without additional temporal integration.

Given that the system supports multiple robotic personalities, specifically shy and intense, it is possible to define personality-specific controller files that differentially modulate affective and navigation behaviour simply by modifying the rules and membership functions in the JSON files. Each personality includes four files (one for each fuzzy dimension: mood, alertness, body_speed, tuning), for a total of eight configuration files. The differences between the two personalities lie in the definition of the fuzzy rules, which determine the intensity and direction of affective variations.
The fuzzy rules on which the implemented control system is based are represented through two types of tables:

- state tables, which describe the transition of TIAGo's internal emotional state as a function of stimuli (facial expression and previous internal state);

- output tables, which map the updated internal emotional state to the behavioural parameters used during the navigation (speed and tuning).

Tables 5.1, 5.2, 5.3 and 5.4 report the fuzzy rules for the shy personality, while Tables 5.5, 5.6, 5.7 and 5.8 (presented later) illustrate the equivalent version for the intense personality.

FACE → MOOD ↓	**neutral**	**happy**	**sad**	**surprise**	**disgust**	**fear**	**angry**
very sad	very sad	very sad	very sad	very sad	very sad	very sad	very sad
sad	sad	sad	very sad	sad	very sad	very sad	very sad
standard	standard	standard	sad	standard	sad	sad	sad
happy	happy	happy	sad	standard	sad	standard	sad
very happy	very happy	very happy	standard	happy	standard	standard	standard

Table 5.1: State table – MOOD Emotional dimension (Shy personality)

MOOD → SPEED ↓	**very sad**	**sad**	**standard**	**happy**	**very happy**
very slow	very slow	very slow	very slow	slow	slow
slow	very slow	very slow	slow	slow	standard
standard	very slow	slow	standard	standard	standard
fast	slow	slow	fast	fast	fast
very fast	slow	slow	fast	very fast	very fast

Table 5.2: Output table – SPEED Emotional dimension (Shy personality)

FACE → ALERTNESS ↓	**neutral**	**happy**	**sad**	**surprise**	**disgust**	**fear**	**angry**
calm	calm	calm	calm	standard	standard	standard	fear
standard	standard	standard	standard	fear	standard	fear	fear
fear	fear	fear	fear	fear	fear	fear	fear

Table 5.3: State table – ALERTNESS Emotional dimension (Shy personality)

ALERTNESS → TUNING ↓	**calm**	**standard**	**fear**
low	low	low	low
medium	medium	low	low
high	medium	medium	low

Table 5.4: Output table – TUNING Emotional dimension (Shy personality)

The four tables presented above describe the behaviour of the fuzzy system for the Shy personality. In this configuration, the rules are designed to generate conservative transitions and limited variations in navigation parameters. The emotional state evolves gradually, with speed and tuning remaining within contained values even in the presence of intense stimuli. The result is slower and more cautious navigation, with a greater distance from obstacles, in line with the shy character of the profile. The tables that follow report the version of the fuzzy system for the Intense personality. While preserving the same formal structure (identical inputs, linguistic labels, and the same separation between state and output tables), the rules are modified to produce a more reactive and dynamic behaviour. Transitions between emotional states are sharper, especially in response to strongly positive or negative facial expressions—while speed and tuning values span a wider range, enabling the robot to adopt higher speeds and more decisive manoeuvres.

FACE → MOOD ↓	**neutral**	**happy**	**sad**	**surprise**	**disgust**	**fear**	**angry**
very sad	very sad	standard	very sad	very sad	very sad	very sad	very sad
sad	sad	standard	very sad	sad	very sad	very sad	very sad
standard	standard	happy	sad	standard	sad	sad	sad
happy	happy	very happy	standard	happy	sad	standard	sad
very happy	very happy	very happy	standard	very happy	standard	standard	standard

Table 5.5: State table – MOOD Emotional dimension (Intense personality)

MOOD → SPEED ↓	**very sad**	**sad**	**standard**	**happy**	**very happy**
very slow	very slow	very slow	very slow	slow	slow
slow	very slow	slow	slow	standard	standard
standard	very slow	slow	standard	standard	fast
fast	slow	slow	fast	fast	fast
very fast	slow	standard	very fast	very fast	very fast

Table 5.6: Output table – SPEED Emotional dimension (Intense personality)

FACE → ALERTNESS ↓	**neutral**	**happy**	**sad**	**surprise**	**disgust**	**fear**	**angry**
calm	calm	calm	calm	calm	standard	standard	standard
standard	standard	calm	standard	standard	standard	fear	standard
fear	fear	standard	fear	fear	fear	fear	fear

Table 5.7: State table – ALERTNESS Emotional dimension (Intense personality)

ALERTNESS → TUNING ↓	**calm**	**standard**	**fear**
low	medium	low	low
medium	medium	medium	low
high	high	high	medium

Table 5.8: Output table – TUNING Emotional dimension (Intense personality)

The two sets of tables enable a direct comparison and show how modifications to the rules produce tangible differences in the robot's navigation and behaviour.
After defining the rules for both personalities, the state and output tables were subjected to a qualitative review by a psychology student. This comparison allowed for the verification of consistency between the emotional transitions modelled and the way in which an "human-like" internal state would be expected to evolve in response to varying stimuli. The same review also confirmed the consistency between changes in the internal state and the corresponding behavioural changes (increases or reductions in speed, approaching or moving away from obstacles), strengthening the psychological plausibility of the implemented model.

To support the design and validation of the fuzzy controllers, simulation models were developed in MATLAB/Simulink, as reported in Figures 5.3, 5.8, 5.5 and 5.6. In these schematics, each input variable is visualised together with its membership functions, the central block implements a Mamdani Type-1 inference system with the full rule base, and the system output is likewise represented through its own membership functions. These images are mainly intended to illustrate the shape of the membership functions and the coverage of the input–output space. Triangular and trapezoidal membership functions are predominantly adopted, and defuzzification is performed using the centroid (centre of gravity) method.
For brevity, only the screenshots corresponding to the intense personality are reported. The shy personality shares the same structure and shapes of the membership functions for mood, alertness and tuning, and therefore would not provide

additional insight at the graphical level. The only relevant difference concerns the body speed controller: in the shy configuration, the output variable is defined in a reduced range (0.1÷0.5) m/s instead of (0.1÷0.9) m/s, so as to limit the maximum linear velocity and prevent the robot from reaching excessively high speeds.

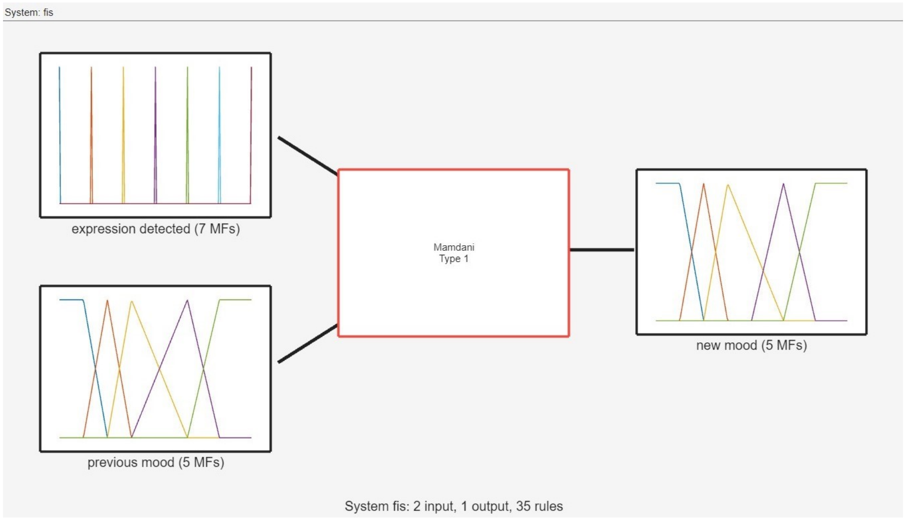

Figure 5.3: Example graphical representation of the Mamdani controller for mood produced in MATLAB (inputs: detected expression and previous mood → output: new mood).

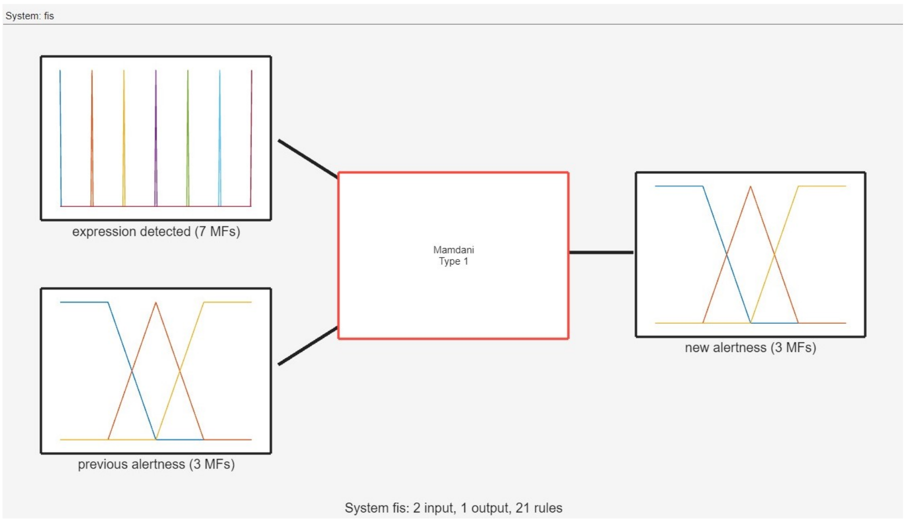

Figure 5.4: Example graphical representation of the Mamdani controller for alertness produced in MATLAB (inputs: detected expression and previous alertness → output: new alertness).

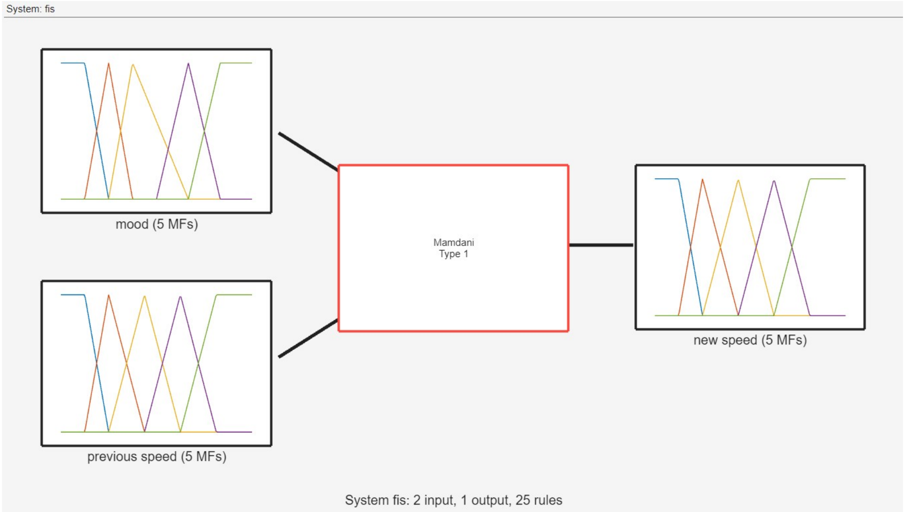

Figure 5.5: Example graphical representation of the Mamdani controller for speed produced in MATLAB (inputs: mood and previous speed → output: new speed).

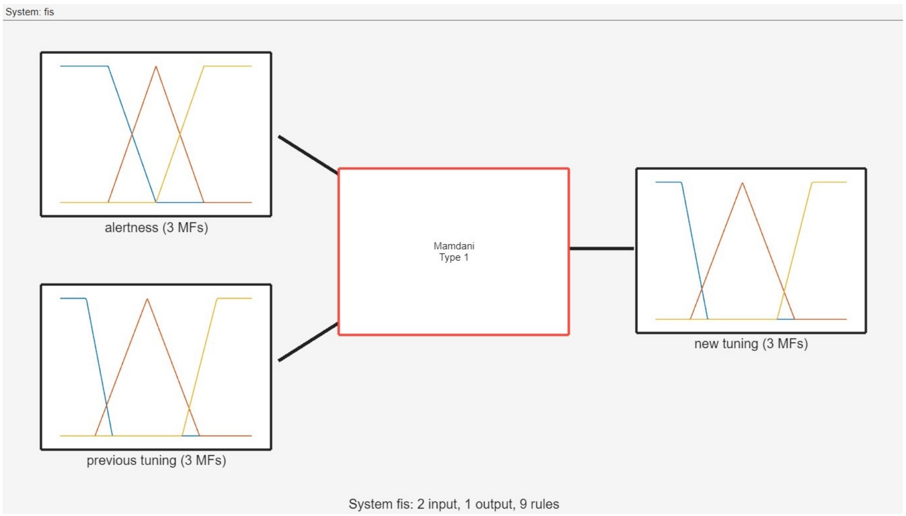

Figure 5.6: Example graphical representation of the Mamdani controller for tuning produced in MATLAB (inputs: alertness and previous tuning → output: new tuning).

5.2.2 Base module: `fuzzy_core.py`

The `fuzzy_core.py` module acts as the "inference engine" of the fuzzy pipeline, representing the component responsible for transforming the declarative specifications defined in the configuration files (membership, domains, rules, and mappings)

into executable and queryable fuzzy systems for the rest of the architecture.
In practical terms, within it, the FuzzyControllerBase class reads the JSON files and, using the scikit-fuzzy library, constructs the universe of discourse variables, membership functions, and the inference engine (Mamdani model). This level also handles defuzzification and provides the option to apply a temporal smoothing mechanism through the `use_delay` and `delay_const` parameters defined in the JSON files, in order to prevent sudden transitions caused by perceptual noise or abrupt changes in the emotional state.
From the pipeline's perspective, this module represents the level that isolates the knowledge (comprising the rules and membership functions) from the application logic: it provides a simple and uniform interface that accepts numerical input data, performs inference based on the defined rules, and returns numerical output (or, when necessary, qualitative labels). This output is then used by the upper layer (`emotional_manager.py`) to update the internal emotional state and, consequently, the behavioural parameters.
Thanks to this separation between application logic and declarative knowledge, it offers great flexibility in modifying the robot's behaviour. For example, introducing new personalities or redefining membership functions (MFs) can be done by modifying only the JSON configuration files, without requiring changes to the orchestration code.

Thus, this python script represents the most logical point to introduce future improvements related to the fuzzy representation itself (e.g., rule syntax extensions, alternative temporal smoothing strategies or caching to optimise performance), while keeping the interface to the emotional manager and the ROS node intact, facilitating modular and easily extendable management.

5.2.3 Emotional manager: `emotional_manager.py`

The emotional state is centrally managed by `emotional_manager.py`, which acts as the orchestrator of the entire fuzzy pipeline. This component coordinates the set of controllers, maintains the internal variables (*mood*, *alertness*, *body_speed*, *tuning*), and implements the temporal logic required to integrate past information into the current emotional state (for example, using the previous value of mood or alertness as input for computing the new emotional state). In addition, this module manages the dynamic loading of personality profiles, the controller update frequency, and update policies designed to limit undesired oscillations.
From an architectural perspective, the *EmotionalManager* forms the intermediate layer between the inference engine (`fuzzy_core.py`) and the ROS node (`fuzzy_node.py`): it receives formatted sensory inputs, invokes the appropriate controllers, combines the resulting outputs, and provides a set of values that can

be used by the control modules and the planner.
At startup, the manager loads the four controllers defined in the JSON configuration files corresponding to the selected personality and initialises the state with predefined values for *mood*, *body_speed*, *alertness* and *tuning*. It also maintains the current facial label (numerically encoded) and a mapping between facial-expression codes and their semantic string representations. This organisation enables the system to handle, in a uniform way, both perception-derived data (facial labels, current speed) and the internal values produced by the fuzzy components (mood, alertness, tuning).
The central operational method is *compute*: given an input dictionary (for example containing the facial label and the current mood value), the manager updates the internal state by following a defined processing sequence. For each controller, it constructs a subset of inputs by selecting only the relevant variables (thus isolating the controller from unexpected fields), invokes fuzzy inference to obtain new numerical values, and, when appropriate, applies additional transformations to the outputs before updating the state. The order of processing reflects the logical dependencies of the model: first, *mood* is updated (based on the previous mood and the facial label); next, *body_speed* is computed (as a function of mood and the actual velocity); the *alertness* level is then updated; and finally, *tuning* is calculated (based on alertness and the previous tuning).
To make the responses more natural and more perceptible to the user during interaction, this script also integrates a mechanism for transient speed modulation: when a change in facial expression is detected, a boost on *body_speed* is activated, defined as an additive delta with a duration expressed in iterations. It decays progressively over a predefined number of steps and is added to the speed value computed by the controller, and finally bounded within the allowed operational range of the platform. The boost parameters (amplitude and duration) are personality-dependent, enabling distinct behaviours (stronger for the intense personality, more contained for the shy one).
This functionality was not part of the initial version of the system but - as detailed in the Section 6.1.2 dedicated to the experiments - was added following a pilot test that revealed that, without transient modulation, the robot's motor changes were insufficiently perceptible to users. The purpose of the boost is therefore to enhance the perceptibility of behavioural changes without altering the underlying logic of the system (the boost does not modify mood, alertness, or tuning, but only the resulting velocity, and only for a limited period of time).
At the end of the computation cycle, the manager constructs and returns an output dictionary containing the updated values of *mood*, *body_speed*, *alertness*, and *tuning*. These results are then used by the integration layer (`fuzzy_node`), which publishes them on the appropriate ROS topics.

In this way, the *EmotionalManager* performs a dual function: it maintains an operative memory of the emotional state and provides a coherent and stable interface between the configurable fuzzy controllers and the robot's control layer.

5.2.4 Main ROS node: `fuzzy_node.py`

The operational orchestration of the entire system is handled by the ROS node implemented in `fuzzy_node.py`. This component serves as the integration point between perception, decision-making, and control: on one side, it subscribes to perception topics (in particular the facial-expression label and odometry/base-velocity data) and forwards them to the emotional manager (*EmotionalManager*); on the other side, it publishes the outputs produced by the fuzzy model on dedicated topics (`/fuzzy_mood`, `/fuzzy_alertness`, `/fuzzy_body_speed`, `/fuzzy_tuning`), so that they can be used by the control modules and the navigation planner (PalLocalPlanner).
A further role of the node is the translation of the qualitative labels produced by the fuzzy system into actual numerical parameters for the local planner: at runtime it queries the parameter mapping defined in `parameters.py` (e.g., for tuning = low, medium, high) and updates the planner behaviour through dynamic-reconfiguration mechanisms (*dynamic_reconfigure*). This approach enables the modulation of safety distances, angular limits, and maximum velocities without interrupting planner execution, thereby facilitating comparative experiments and direct observation of behavioural effects.
At startup, the node instantiates the emotional manager, specifying the selected personality (which may be provided as a runtime argument; if no argument is given, the default configuration is shy). It then registers to the necessary communication channels, subscribing to the perception topics to receive facial labels and the platform's current velocity, while maintaining dedicated publishers to distribute the computed fuzzy values. The odometry velocity is used as an input to the *body_speed* controller, allowing the system to assess changes in locomotion dynamics as a function of the emotional state.

To avoid unnecessary recalculations and reduce computational load, the node applies a filtering criterion: when the incoming facial label matches the last processed one, the computation is skipped. This design choice stems from experimental considerations: facial recognition operates at high frequencies, and repeated readings of the same state would trigger continuous reconfigurations of the planner, potentially causing repeated increments in TIAGo's speed and tuning parameters, resulting in undesirable oscillations and exaggerated behaviours. The implemented filter limits the system's actions to perceptually meaningful changes, thereby improving navigation stability. When an actual change in expression is detected, the node constructs the input packet expected by the emotional manager (encoded facial label and current velocity) and requests its processing; it then receives the updated values for *mood*, *body_speed*, *alertness*, and *tuning*. For each numerical value, the node retrieves the corresponding qualitative label and publishes both the numerical and qualitative representations to the dedicated topics. This dual publication is intentionally designed to support both qualitative experimental analysis and the direct use of numerical values by control components.

To support experimentation and debugging, the node maintains a detailed log of the current state (latest detected expression, numerical values and labels from the controllers, actual base velocity), producing periodic log entries that facilitate data collection and offline analysis. Below are two illustrative screenshots of the recognition Graphical User Interface (GUI) and the execution terminal, showing the format of the logs generated during the experiments.

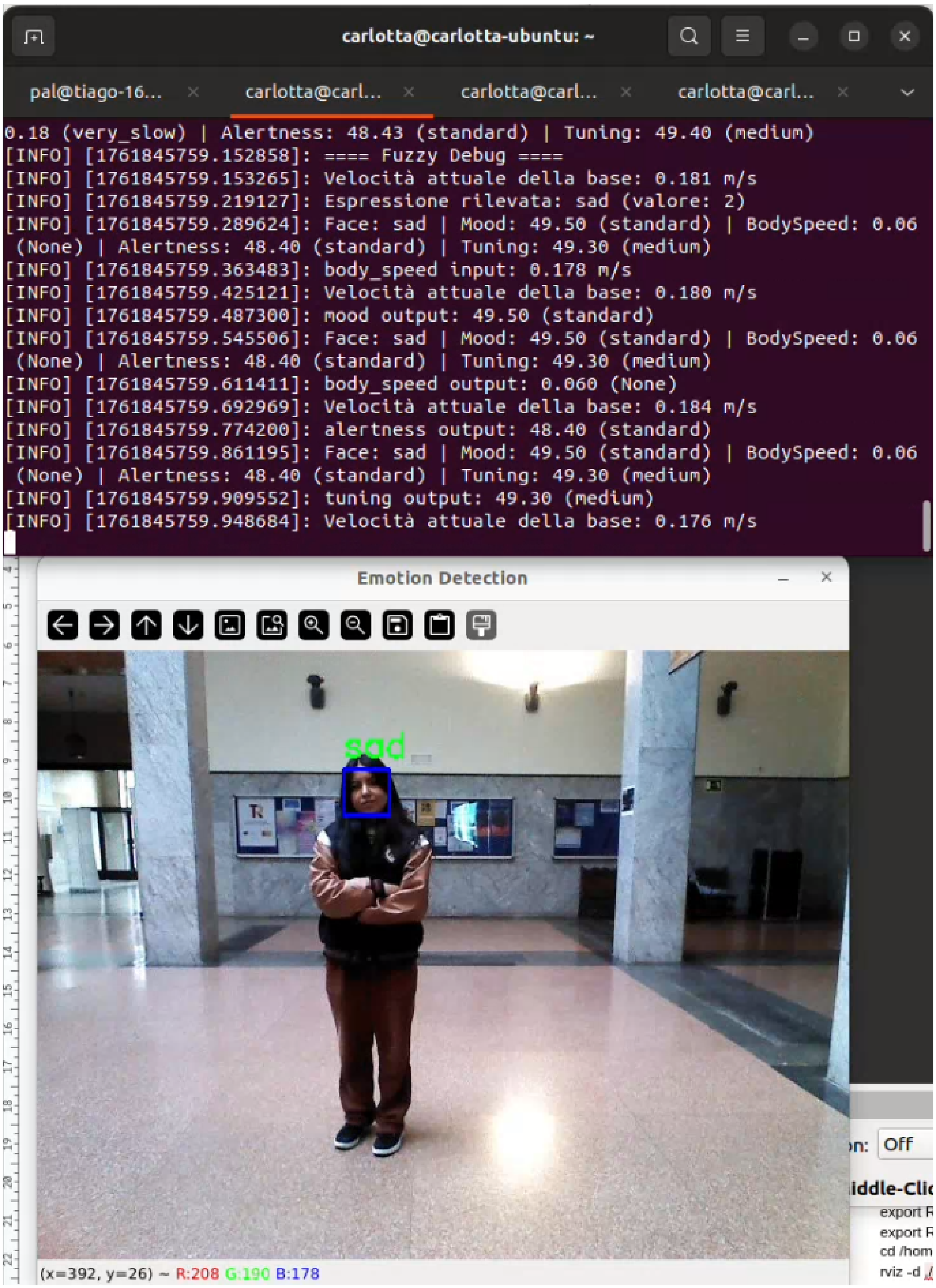

Figure 5.7: Terminal log and GUI (example "sad"): screenshot showing the log entries with face = "sad", the corresponding values of mood, body_speed, alertness and tuning, and the emotion-detection GUI window.

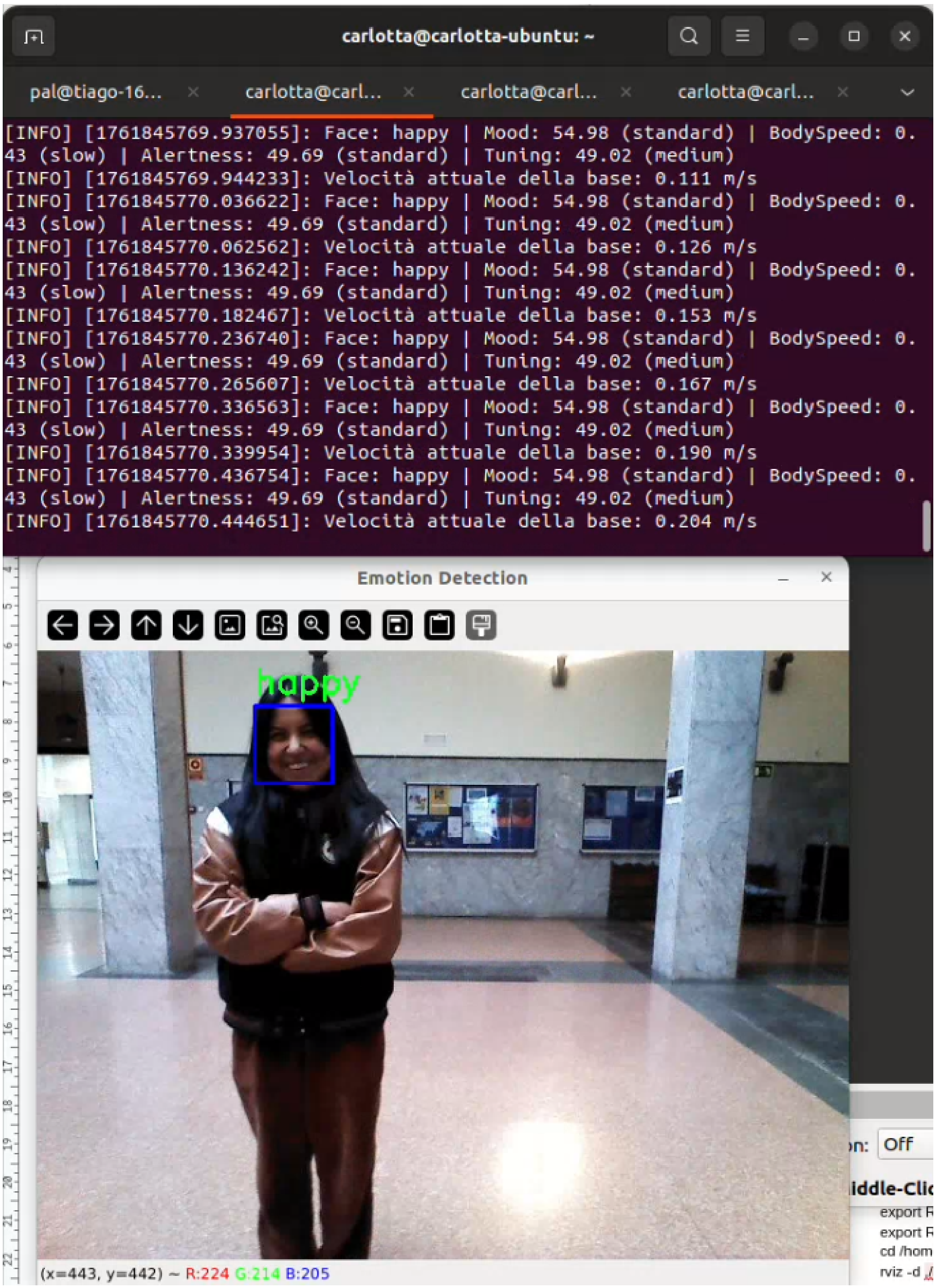

Figure 5.8: Terminal log and GUI (example "happy"): screenshot showing the log entries with face = "happy", the corresponding values of mood, body_speed, alertness and tuning, and the emotion-detection GUI window.

In the terminal, informative messages with timestamps are emitted, containing the following elements, repeated at each relevant update:

- the detected facial label (e.g., happy, sad), useful for visually linking the perceptual stimulus;
- the current numerical value of mood and its corresponding qualitative label in parentheses (e.g., *Mood: 54.98 (standard)*);
- the resulting numerical value for body_speed and its qualitative label (e.g., *BodySpeed: 0.43 (slow)*);
- the numerical value and label for alertness (e.g., *Alertness: 49.69 (standard)*);
- the numerical value and label for tuning (e.g., *Tuning: 49.02 (medium)*);
- auxiliary lines reporting the current base velocity estimated from odometry (e.g., *Current base velocity: 0.204 m/s*).

This textual representation in the terminal makes the correspondence between the stimulus (facial expression) and the system's reaction (numerical values and labels) immediately readable, facilitating debugging and the collection of experimental data synchronised with the images captured by the webcam.

In summary, `fuzzy_node.py` is the component that materialises the outcomes of the emotional model within the robot's navigation strategy: it coordinates incoming data flows, triggers fuzzy processing, translates the outputs into operational parameters, and applies dynamic updates to the planner, all while maintaining a clear separation between perception, inference, and control.

5.2.5 Parameters module: `parameters.py`

The `parameters.py` module serves as the final translation stage between the qualitative representation produced by the fuzzy system and the actual numerical parameters used by the robot's local planner. In practice, given a *tuning* category (e.g., low, medium, high), it provides a tuple of operational parameters, in the current implementation (*max_vel_theta*, *security_dist*), representing the planner's angular velocity limit and the minimum safety distance from obstacles, respectively. The predefined mapping adopted in the code is immediately interpretable (e.g., low → 0.5 rad/s and 0.5 m; medium → 0.75 rad/s and 0.3 m; high → 1.0 rad/s and 0.15 m) and includes a fallback behaviour (defaulting to medium in case of an unrecognised label), thereby ensuring robustness during operation.
Isolating this mapping within a dedicated module allows for a clear separation of responsibilities and abstraction levels: the fuzzy logic merely decides the qualitative

category, while `parameters.py` handles numerical translation and the engineering choices associated with the platform. In the operational flow, the integration node queries this function to obtain the numerical values corresponding to the computed tuning and transmits them to the planner (together with the *max_vel_x* value derived from *body_speed*) via *dynamic_reconfigure*, enabling real-time adaptation of the movement strategy.

Chapter 6

Experiments

6.1 Experiment Design

6.1.1 Experimental Setup

The experiments were conducted in the main hall of the Universidad Politécnica de Madrid, known as the "Sala de la Máquina". The choice of this environment was motivated by the need for a sufficiently large space, free of major obstacles, capable of ensuring safety and accessibility for both the robot and the participants. The area was prepared in advance to facilitate proper autonomous navigation of the robot and smooth interaction with the users.

The preliminary setup phase involved mapping the environment using Simultaneous Localization and Mapping (SLAM) techniques implemented onboard the TIAGo robot. The mapping procedure was performed through remote control of the robot using a joystick, allowing it to be guided along strategic paths within the hall and to acquire an accurate spatial representation of the environment. The SLAM process enabled the robot to build a digital map of the surroundings—an essential prerequisite for subsequent autonomous navigation and for executing the interactive behaviors defined by the experimental protocol. The result was visualized through the RViz interface, which allows real-time observation of the two-dimensional reconstruction of the environment along with the overlay of the sensory data acquired by the robot.
The robot used in the experiments was the TIAGo Base + Arm model, equipped with an integrated vision system, proximity sensors, and a single robotic arm.

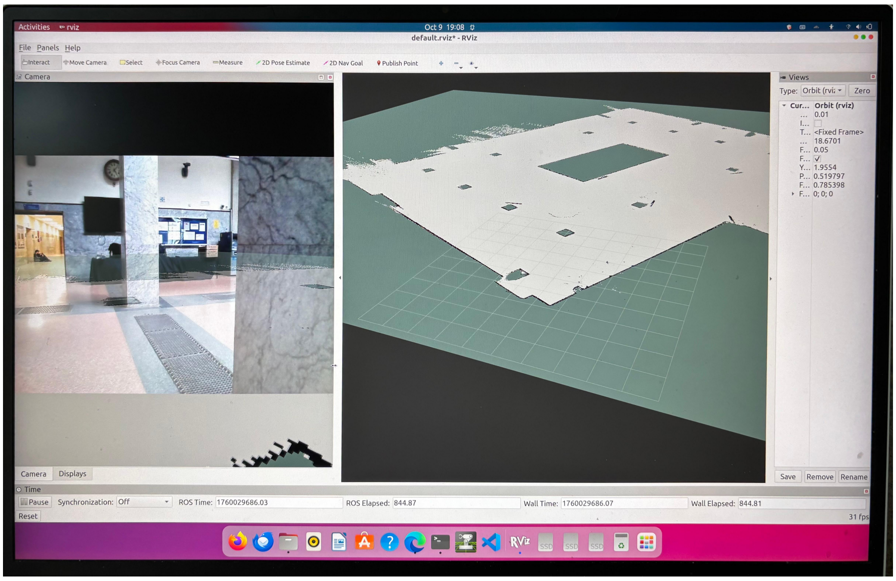

Figure 6.1: Visualization of the 2D map generated by the TIAGo robot in RViz during the SLAM phase, performed in the Sala de la Máquina at the Universidad Politécnica de Madrid.

The facial expression recognition system and the fuzzy-based emotion regulation model were integrated within the ROS1 framework, enabling real-time communication between the perception, processing, and navigation nodes. The correct operation of these modules was verified beforehand to ensure the full functionality of the system.
The experimental setup also included the definition of observation and interaction areas, as well as the preparation of the equipment required for data collection and for running and monitoring the various implemented codes and nodes. This configuration allowed the system to be evaluated under actual operating conditions, providing results that are representative of real human–robot interaction.

6.1.2 Experimental Procedure

The experimental procedure was structured to ensure the highest possible internal and external validity of the collected data, as well as the replicability of the experiment.
Before beginning the experiment, the testing procedure was explained in detail, and the experimental protocol was clarified, specifying the data collection methods and ensuring full protection of participants' privacy and anonymity. Participants were divided into two equivalent subgroups in order to minimize potential group effects and to guarantee a balanced distribution of the experimental conditions.

The experiments required participants to interact with the TIAGo robot under two different behavioral configurations, with the interaction procedure organized into the following phases:

1. Interaction with the robot in neutral mode.
   Each participant first interacted with the robot without the emotional system and without any predefined personality. In this phase, the robot's behavior was limited to standard responses, lacking emotional expression or behavioral variations related to personality traits.

2. Interaction with the robot in personalized mode.
   Subsequently, the two groups experienced different configurations:

   - the first group interacted with the robot configured with personality shy (A), characterized by moderate movements, greater interpersonal distance and more restrained emotional responses;
   - the second group interacted with the robot configured with personality intense (B), featuring faster movements and more pronounced emotional responses.

During each interaction session, participants were invited to interact freely with the robot, intentionally crossing its trajectory in order to observe potential variations in the robot's emotional and behavioral responses.
At the end of the experimental session, each participant was asked to complete a structured questionnaire designed to assess both their subjective perception of the robot under the different configurations and their own emotional state experienced during the interaction.
The procedure was carefully designed to ensure uniform testing conditions, control for potential biases, and enable the systematic collection of data required for both quantitative and qualitative analyses.

In order to verify the stability and reliability of the developed system, the experimental phase was organized into several distinct sessions.
The first testing session took place on October 13th, 2025, between 11:00 and 12:00 a.m., and involved a sample of eight students from the Universidad Politécnica de Madrid (UPM), who participated in the entire experimental protocol using the initial version of the system. Prior to the start of the experiment, participants were given a presentation of the project, illustrating the research objectives, the technical features of the TIAGo robot, and the nature of the planned interactions. Additionally, the participants were informed in advance that they would first interact with the robot in its neutral mode (without the emotional system) and then with the robot in its personalized mode, which would activate the emotional system based

on a predefined personality (shy or intense). This information was intentionally omitted during the final tests in order to objectively assess participants' perceptions without prior knowledge influencing their responses.
At the end of this first session, a preliminary analysis of the collected data was carried out, including the extraction of questionnaire responses and a statistical study based on a mixed ANOVA method, aimed at verifying the consistency of the measurements and identifying potential aspects of the experimental setup and interaction procedure that could be improved.

Figure 6.2: Experimental session held on October 13th 2025, in the Sala de la Máquina, Universidad Politécnica de Madrid.

Based on the observations that emerged, two main technical modifications were subsequently implemented to optimize the system, reducing latency and enhancing the visibility of the robot's emotional reactions.

- Video telemetry: the video acquisition mode was modified from continuous video streaming to a frame-based sampling approach, with frames captured at a frequency of 1 fps (frame per second). This modification was introduced to reduce the delay caused by the acquisition and transmission pipeline and to improve the temporal correspondence between the facial recognition event and the robot's subsequent reaction, thereby enhancing the perceived accuracy of the emotional expression.

- Reactive locomotion parameters: the `emotional_manager.py` module was updated by introducing a transient "boost" mechanism implemented as a layer superimposed on the existing fuzzy results. When a change in the user's facial expression is detected, an additive increase or decrease in `body_speed` (delta value) is triggered, applied immediately, and linearly decaying over N iterations (factor = steps_left / total_steps). The boost parameters (delta amplitude and duration in steps) differ between the two personalities (e.g., intense: happy +0.25 for 4 steps; shy: happy +0.12 for 3 steps). To ensure operational safety and dynamic consistency, the resulting speed is always clamped within an allowed interval, preventing invalid or unsafe values. This mechanism produces a visible speed peak immediately after the recognition of an expression, without altering the underlying fuzzy logic.
  In parallel, some adjustments were also made to the `.json` modules. Specifically, the `delay_const` value in the `mood_shy.json` and `mood_intense.json` files was reduced (from 10 to 3), resulting in a faster reaction to new facial inputs while maintaining slight temporal smoothing. Finally, the output range of `body_speed` was increased from $0.1 \div 0.5$ m/s to $0.1 \div 0.9$ m/s, allowing for a wider speed excursion that makes velocity variations more perceptible.

Following these modifications, which constituted the final setup, two additional experimental sessions were scheduled and conducted. The first took place on Monday October 20th 2025, between 4:00 p.m. and 7:00 p.m., involving twelve randomly selected participants recruited among passersby in the Sala de la Máquina. The second session was held on Thursday October 30th 2025, between 3:00 p.m. and 6:00 p.m., with an additional group of 16 participants.
These subsequent sessions followed the same experimental protocol described above (an initial interaction with the robot in neutral mode, followed by an interaction with Personality A or B depending on the group). The statistical analysis was conducted along the same lines, employing multi-condition comparison procedures (ANOVA) applied to the main scales, with the aim of identifying statistically significant differences between conditions.
The detailed methods and statistical results are presented in Section 6.3 (Statistical Analysis) and Section 6.4 (Results).

6.2 Questionnarie and Evaluation Metrics

To systematically evaluate the effectiveness of the emotion recognition and modulation system implemented on the TIAGo robot, a structured questionnaire was developed and administered to participants at the end of the experiment. The objective was dual: on the one hand, to collect data on the subjective perceptions of the participants about the robot and its emotional state; on the other hand, to assess the emotions actually experienced by the participants during the interaction. For the overall evaluation of the experimental sessions, a mixed-methods approach was adopted, combining validated psychometric instruments with specifically designed questionnaires. This strategy allows integration of standardized measurements, widely recognized in the scientific literature, with targeted questions that aim to capture specific aspects of the user–robot interaction experience.[17][18]

The first section of the questionnaire collects essential demographic and background information, such as age, gender, and participants' level of familiarity with technology in general and with robotics in particular. A question was also included to determine whether participants had any prior interactions with the TIAGo robot, in order to control for potential biases related to familiarity with the robotic platform. This information is useful for statistical purposes, as it allows for the analysis of possible correlations between participants' profiles and their evaluations.

To investigate the perception of the robot's characteristics, as introduced in Section 2.5.1, items from the Godspeed Questionnaire Series were selected. Specifically, the Animacy and Likeability scales were chosen, as they are particularly sensitive to changes in the robot's emotional behavior and personality, which are central to the experimental objectives. For the Animacy scale, all five items were selected: Dead-Alive, Stagnant-Lively, Mechanical-Organic, Artificial-Lifelike and Apathetic-Responsive, while for the Likeability scale, three items were included: Dislike-Like, Unfriendly-Friendly and Unpleasant-Pleasant.
These scales were administered twice, corresponding to the two experimental conditions: the first involving interaction with the robot without the emotional system, and the second with the robot equipped with the emotion recognition and modulation system. This design made it possible to analyze how the presence of the emotional system influenced the perception of the robot in terms of perceived liveliness and pleasantness. At the end of the Godspeed questions, participants were provided with a comparative section aimed at collecting their subjective preferences between the two robot versions. Specifically, they were asked which robot appeared more natural, more pleasant to interact with, more "alive" or expressive, and which one, overall, they would prefer to encounter again in a future interaction. In addition to these closed-ended questions, an open-ended item was included, allowing participants to freely describe the differences they perceived between the two robot versions and to provide qualitative observations or comments useful for interpreting the results.

Subsequently, to analyze the impact of the interaction on participants' emotional state, an adapted version of the EVEA (Escalas Visuales Analógicas de Estado de Ánimo) scale was employed (for theoretical background, refer to Section 2.5.2).

Participants were asked to rate, on a scale from 0 to 10, the emotions they experienced during their interaction with TIAGo (i.e., I felt happy, sad, calm, anxious, irritated and understood).
Finally, to complement this quantitative section, the questionnaire also included a few open-ended questions aimed at collecting spontaneous impressions not constrained by predefined scales. Specifically, participants were asked to indicate which emotions they perceived in the robot and to provide any comments or suggestions regarding their interaction experience, including possible improvements. These qualitative contributions enrich the overall analysis by offering additional interpretative insights beyond the numerical data alone. [17]

The questionnaire was created and administered through the Google Forms platform, chosen for its ease of use and its ability to automatically and anonymously collect participants' responses.
Since participants were divided into two subgroups—the first interacting with the robot without an emotional system and then with the robot featuring personality A (shy), and the second interacting first with the robot without an emotional system and then with the robot featuring personality B (intense)—two distinct but structurally identical questionnaires were prepared for practical and data management reasons. The two versions differed only in the coding of the collected data.
At the end of each experimental session, respondents were asked to complete the form by scanning a specifically generated QR code, which redirected them directly to the questionnaire link. Each version was associated with a dedicated QR code, ensuring that the responses of the two subgroups remained separated during the statistical analysis phase, while maintaining full consistency in both the administration procedure and the question structure.
This process ensured a fast, uniform and error-free data collection, while also enabling the immediate aggregation of responses for subsequent statistical analysis.
A complete copy of the questionnaire used in this study is provided in the Appendix section B.

6.3 Statistical Analysis

In this section, the statistical strategy adopted for the analysis of the data collected through the questionnaires is described (Godspeed: Animacy and Likeability scales; EVEA: emotions scale). The purpose of the analysis is to assess whether, and to what extent, the experimental conditions (interaction with the robot in Neutral vs Personalised mode) and membership of differential subgroups (shy vs intense personality) influence the measured variables of perception and emotional state.
The main questions addressed are:

- Does group (shy vs intense) influence the perception of Animacy and Likeability?
- Does condition (Neutral vs Personalised) influence perception?
- Do these effects combine in a specific way (i.e., is there an interaction)?

To address these questions, analyses were conducted using repeated-measures ANOVA (mixed ANOVA), following the theoretical framework outlined in Section 2.6.

6.3.1 Operational Procedure, Data Preparation and Software Tools

The data analysis was conducted following a reproducible pipeline structured into three main phases: preparation and aggregation of questionnaire responses; dataset checking and cleaning; execution of statistical analyses and production of supporting plots. The following subsection details the operations performed at each phase, the criteria adopted for data processing, and the software tools employed.

For the scales taken from the Godspeed Questionnaire, mean values were computed for each participant and for each experimental condition. Specifically:

- Animacy (Godspeed): each participant answered 5 items related to Animacy in both the Neutral and the Personalised robot conditions; for each condition, the arithmetic mean of the 5 items (Dead-Alive, Stagnant-Lively, Mechanical-Organic, Artificial-Lifelike, Apathetic-Responsive) was computed and stored as Animacy_mean_Neutral and Animacy_mean_Personalised;
- Likeability (Godspeed): each participant answered 3 items related to Likeability in both conditions; for each condition, the arithmetic mean of the 3 items (Dislike-Like, Unfriendly-Friendly, Unpleasant-Pleasant) was computed and stored as Likeability_mean_Neutral and Likeability_mean_Personalised.

These means formed the compact dataset for the ANOVA: in an Excel sheet, a table was created with one row per participant and the essential columns containing these values. This procedure and the creation of the Excel file were repeated for the two trials: in the first case, the file was named `godspeed_test.xlsx` and included the first eight participants who interacted with the first version; in the second case, it was named `godspeed_final.xlsx` and compiled the union of the data collected across the two experimental sessions, which involved 28 total participants who interacted with the final system.
The two Excel tables are reported below.

ID	Personality	Animacy _mean_ Neutral	Likeability _mean_ Neutral	Animacy _mean_ Personalised	Likeability _mean_ Personalised
1	shy	1.6	1.00	3.2	3.67
2	shy	2.4	3.33	3.0	3.33
3	shy	2.2	2.67	3.4	3.00
4	shy	2.2	2.33	3.8	4.00
5	intense	3.0	2.67	3.0	3.33
6	intense	4.0	4.00	3.6	3.67
7	intense	2.6	2.67	3.8	4.00
8	intense	2.6	2.33	4.0	4.00

Table 6.1: `godspeed_test.xlsx` : mean per-participant scores for the Animacy and Likeability scales under the Neutral and Personalised conditions - pilot test.

ID	Personality	Animacy _mean_ Neutral	Likeability _mean_ Neutral	Animacy _mean_ Personalised	Likeability _mean_ Personalised
1	shy	2.2	2.00	4.0	4.00
2	shy	3.2	2.67	3.6	3.00
3	shy	3.4	2.67	2.6	2.00
4	shy	3.4	3.67	2.6	3.67
5	shy	3.0	2.00	4.2	3.67
6	shy	2.0	3.67	2.2	3.33
7	intense	3.4	3.67	3.4	4.00
8	intense	1.4	3.67	2.0	4.00
9	intense	3.2	2.67	2.4	4.00
10	intense	1.4	1.67	4.0	3.67
11	intense	2.6	2.67	3.6	3.33
12	intense	1.6	2.67	4.2	4.33
13	shy	3.6	4.00	4.0	2.00
14	shy	3.6	1.67	1.8	2.00
15	shy	3.4	1.67	3.4	3.33
16	shy	3.0	3.00	3.4	3.33
17	shy	2.6	2.00	5.0	5.00
18	shy	3.2	2.33	1.0	2.67
19	shy	1.2	1.00	3.8	5.00
20	shy	2.8	2.33	4.0	3.33
21	intense	1.8	3.00	5.0	5.00
22	intense	2.0	2.00	4.0	4.00
23	intense	2.6	3.00	3.8	4.67
24	intense	2.0	3.00	4.4	4.67
25	intense	3.8	3.00	2.2	3.67
26	intense	4.2	3.00	3.0	4.33
27	intense	2.0	2.33	3.8	4.00
28	intense	2.6	2.33	3.8	4.00

Table 6.2: `godspeed_final.xlsx` : mean per-participant scores for the Animacy and Likeability scales under the Neutral and Personalised conditions - final tests.

Separately, for the EVEA scale, two additional Excel files were created for the two testing situations performed, named `evea_test.xlsx`, which contains the data of the eight participants who interacted with the prototype system, and `evea_final.xlsx`, which contains the data of 28 individuals who interacted with the final optimized system. Each file includes, for every participant, a unique ID, the personality with which they interacted in the second part of the experiment (shy or intense), and the scores on the six EVEA items (happy, sad, calm, anxious, irritated, understood).

Given the brevity of the interaction between the user and TIAGo, and the complexity of human emotions, it is difficult, and at times speculative, to describe participants' affective states. Nevertheless, for an indicative analysis these spreadsheets were used for descriptive summaries and for between-group comparisons of participants' emotional state.
These two Excel files are reported below in tabular form.

ID	Personality	happy	sad	calm	anxious	irritated	understood
1	shy	5	8	3	7	7	4
2	shy	7	0	7	2	0	3
3	shy	7	2	9	0	0	0
4	shy	8	8	8	2	2	6
5	intense	6	2	7	3	6	5
6	intense	7	2	4	6	1	6
7	intense	7	3	6	3	1	5
8	intense	7	1	7	7	0	6

Table 6.3: `evea_test.xlsx`: EVEA scores per participant — pilot test.

ID	Personality	happy	sad	calm	anxious	irritated	understood
1	shy	8	1	8	1	1	8
2	shy	3	0	0	7	5	0
3	shy	8	3	10	0	7	5
4	shy	8	0	8	1	3	6
5	shy	10	0	8	1	1	0
6	shy	5	0	6	7	1	2
7	intense	7	3	8	2	0	7
8	intense	6	4	8	2	3	5
9	intense	1	1	9	0	2	3
10	intense	5	0	9	3	2	7
11	intense	7	0	7	1	0	2
12	intense	8	2	9	0	0	6
13	shy	8	2	9	2	1	7
14	shy	7	1	10	0	4	3
15	shy	7	0	5	6	6	5
16	shy	8	3	6	7	8	8
17	shy	7	0	10	0	0	7
18	shy	9	3	8	1	1	4
19	shy	8	3	9	0	0	2
20	shy	8	1	7	1	0	3
21	intense	6	4	8	1	0	7
22	intense	8	1	7	1	1	7
23	intense	9	0	6	7	1	8
24	intense	6	1	8	2	1	8
25	intense	7	4	7	5	4	6
26	intense	8	0	8	2	0	5
27	intense	8	1	7	1	0	7
28	intense	8	1	6	1	1	7

Table 6.4: `evea_final.xlsx`: EVEA scores per participant — final tests.

The processing was automated using well-established open-source tools commonly adopted in experimental research:

- Python (version 3.10.12) as the scripting environment;
- pandas (version 2.2.3) for data handling and transformation;
- pingouin (version 0.5.5) for mixed ANOVA and statistical tests;
- seaborn (version 0.13.2) and matplotlib (version 3.8.2) for data visualization.

Two main Python scripts were used:

1. `godspeed_analysis.py`: takes as input the Excel file (`godspeed_test.xlsx` or `godspeed_final.xlsx`), runs the mixed ANOVA for Animacy and Likeability using pg.mixed_anova (pingouin), produces the Godspeed plots and prints the ANOVA tables to the terminal. The following figures are generated for both Animacy and Likeability:
   - Bar plot of means by condition and by group, with error bars representing the standard deviation (SD). This plot shows the observed means and sample variability; the error bars facilitate reading differences between groups and conditions;
   - Point plot (line plot with markers) of the Group × Condition interaction: it connects the means of the two conditions for each group, highlighting potential interaction patterns (parallel lines vs divergence/crossing).
2. `evea_analysis.py`: takes as input `evea_test.xlsx` or `evea_final.xlsx`, computes descriptive statistics for each emotion to allow rapid inspection of overall trends and between-group differences, prints numerical summaries, means (arithmetic mean of the collected scores for that EVEA item) and standard deviations SD (measure of score dispersion around the mean, high SD indicates wide variability among participants), and produces the following figures:
   - Bar plot of overall means (all participants) for each emotion;
   - Bar plot by group (shy vs intense) for each emotion.

The scripts were executed from the folder containing the Excel files using the commands: `python3 godspeed_analysis.py` and `python3 evea_analysis.py`.

Before each analysis, quality checks were performed on the dataset collected via the questionnaires, following the operational protocol below:

- verification of column names and data types, to ensure that the Python scripts found the expected columns;

- checking the uniqueness of identifiers (ID) and the correspondence between ID and group;
- identification of variables: subject ID (ID) as the participant identifier, Condition as the within-subjects factor (Neutral/Personalised), and group as the between-subjects factor (shy/intense personality);
- data aggregation and formatting: to apply mixed-ANOVA procedures, variables measured under two conditions (Neutral and Personalised) were reorganised into 'long' format (one row per participant $\times$ condition); this transformation was performed in the Python scripts via pandas.melt;
- preliminary inspection of distributions (histograms, extreme values) to identify clear outliers or transcription errors.

To avoid missing values (NaN) in the columns of interest, which would have led to the exclusion of the record, responses related to the Godspeed and EVEA scales were marked as mandatory in the administered questionnaire, unlike the open-ended questions.

Concluding, in both cases, pilot with the initial system and final analyses with the optimised system, the procedures for preparation, checking, transformation into long format, ANOVA analysis, and figure generation were identical, ensuring comparability of results across the experimental blocks.
To maximise reproducibility, the two Python scripts are stored and provided in Annex C of the project document; all numerical and graphical outputs generated by the scripts, including the ANOVA tables with F values, *p-value* and η_p^2 values, have been archived and incorporated into the present work in Section 6.4 Results.

6.4 Results

This section presents and discusses the statistical and descriptive analyses conducted on the data collected through the Godspeed questionnaires (Animacy and Likeability scales) and the EVEA scale. It is organised into three parts, corresponding to: the results of the first test conducted with a sample of eight participants; the results of the final tests with the optimised emotional system; and, finally, a comparison between the two datasets.

6.4.1 Results of the first test

The pilot session involved eight participants. From the questionnaire, their ages ranged from 21 to 23 years; 2 were female and 6 male, evenly distributed across

the two experimental personality conditions (shy and intense). Familiarity with technology, and, in particular, with robotics, was on average moderate–high (values between 3 and 4 on a 1–5 scale), while most participants (6 out of 8) had not previously had direct interaction with the TIAGo robot. The data analysed by the `godspeed_analysis.py` script (reported in Annex C) are the individual means computed on the Godspeed scales, Animacy (mean of 5 items) and Likeability (mean of 3 items), shown in Table 6.1. Conversely, the data analysed by the `evea_analysis.py` script, documented in Annex C, are the EVEA scores (0–10 scales for six emotions) collected at the end of the interaction, shown in Table 6.3
The objective of this first experimental test was to assess measurement consistency, validate the protocol, and obtain preliminary estimates of means and variability, identifying any criticalities to be corrected and optimised. It is important to acknowledge the limited sample size: with such a small number of participants, effect-size estimators can be unstable and susceptible to overestimation.
For this reason, the conclusions should be treated as exploratory.

For each of the Godspeed scales analysed (Animacy and Likeability), the Python script returns three rows of output from the mixed ANOVA, each corresponding to a different effect tested by the model. Specifically, the three values F, p and η_p^2 reported for each measure refer to:

- the main effect of the between-subjects factor (Personality): tests whether, averaging across the two conditions, there are overall differences between the two groups (shy vs intense);
- the main effect of the within-subjects factor (Condition): assesses whether the different experimental setting (Neutral vs Personalised) produces an average change in the measure under consideration;
- the interaction effect (Personality × Condition): tests whether the effect of Condition differs as a function of group.

For each measure (Animacy and Likeability), the results of the mixed ANOVA for the three model terms are reported in the table below: main effect of Personality (between), main effect of Condition (within), and Personality × Condition interaction.

Measure	**Effect**	**F**	**p**	η_p^2
Animacy	Personality	8.075	0.0295	0.574
Animacy	Condition	12.874	0.0115	0.682
Animacy	Interaction	1.947	0.2124	0.245
Likeability	Personality	2.159	0.1921	0.265
Likeability	Condition	6.952	0.0373	0.538
Likeability	Interaction	0.195	0.6742	0.031

Table 6.5: Results of ANOVA tests for Animacy and Likeability measures - firt test.

As described earlier, the F statistic represents the ratio between the variability attributable to the effect under consideration and the residual variability not explained by the model. A high F value indicates that the observed difference is large relative to sampling noise, implying that differences in Personality or Condition are meaningful and produce changes in users' responses; the associated p-value quantifies the probability of obtaining a value at least as extreme under the null hypothesis.
The values reported in the table show significant effects of Condition (Neutral vs Personalised) on both Godspeed scales (very high $F$and p<0.05) and, for Animacy, also a main effect of Personality. This indicates that the experimental manipulation (Neutral robot vs Personalised robot) was indeed perceived by participants and leads to an average increase in both Animacy and Likeability scores. In other words, the personalised version of the robot is, overall, judged as more 'alive' and more pleasant than the neutral version. The fact that the main effect of Personality is significant only for Animacy means that, averaging across the two conditions, the two experimental groups (shy vs intense) differ globally in their tendency to attribute lifelikeness/humanness to the robot, but do not show an overall difference in their evaluation of likeability. Concluding that the personality assigned to the robot stably shapes users' perception of it as 'lifelike', whereas perceived likeability depends primarily on whether the robot is personalised, rather than on the specific personality (shy vs intense).
It is also important to note that the absence of a significant interaction (Personality $\times$ Condition) implies that the effect of moving from Neutral to Personalised is generally similar for both groups: with the available data, there is no statistical evidence that personalisation has a different impact on participants who interacted with a shy robot with respect to the ones that interacted with an intense robot.

To complement the numerical results, figures were generated to make the collected data more immediately interpretable and comparable.

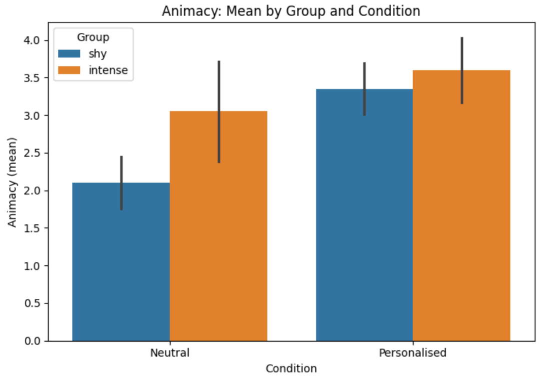

Figure 6.3: Bar plot of Animacy scale means by group and experimental condition - pilot test.

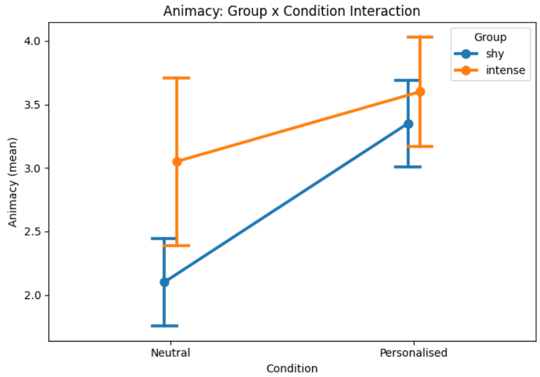

Figure 6.4: Point plot of the Group × Condition interaction for the Animacy scale - pilot test.

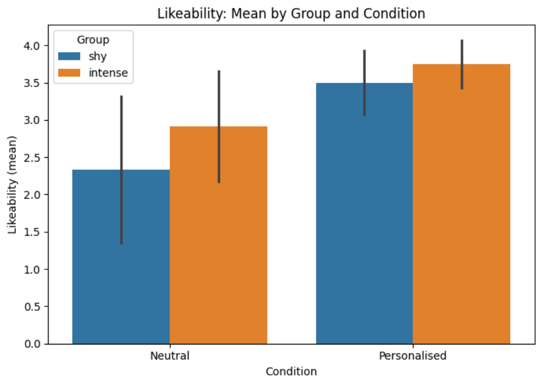

Figure 6.5: Bar plot of Likeability scale means by group and experimental condition - pilot test.

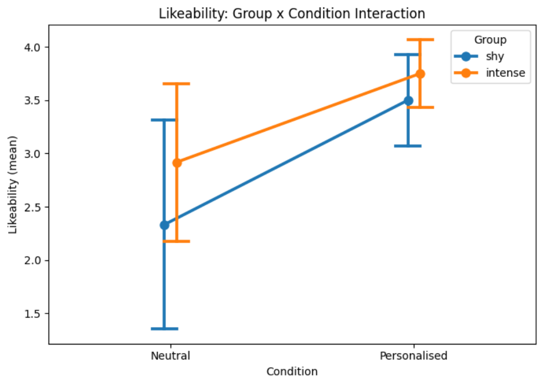

Figure 6.6: Point plot of the Group × Condition interaction for the Likeability scale - pilot test.

The four plots produced for the pilot experimental session provide an immediate visual summary of the trends reported in the table and facilitate the reading of differences between conditions and groups. The outcomes of the mixed ANOVA are consistent with the plots: the means increase when moving from Neutral to Personalised.

- Animacy — bar plot (Figure 6.9): shows Animacy means for each condition (Neutral vs Personalised), separated by group (shy, intense), with error bars representing the standard deviation. A marked average increase is observed when moving from Neutral to Personalised in both groups; in particular, the shy group exhibits a lower initial mean in Neutral and a larger increase in Personalised compared with the intense group.

- Animacy — interaction point plot (Figure 6.10): connects the Neutral and Personalised means for each group and highlights the slope of the lines. The different magnitude of increase between shy and intense (steeper line for shy) is clearly appreciable.

- Likeability — bar plot (Figure 6.11): for Likeability, an average increase is likewise observed in the Personalised condition for both groups. Means are higher in Personalised, and the error bars show greater variability in the Neutral condition.

- Likeability — interaction point plot (Figure 6.12): the near-parallel lines between shy and intense indicate that the increase in Likeability from the Neutral to the Personalised condition is similar for both groups, confirming the absence of a statistical interaction.

The quantitative and numerical results of the statistical study were complemented by a qualitative analysis of participants' open-ended responses.
In the group that interacted with the "shy" personality, the emotions perceived in the robot were primarily interpreted as shyness or discomfort; among these participants, three out of four reported preferring the version with the affective system (Personalised) over the neutral version. In the "intense" group, the emotions perceived were described as closeness and greater proximity; in this group, all participants (4/4) indicated a preference for the version with the emotional system.

Following the analysis of the Godspeed scales, the emotions reported by participants through the EVEA scale were examined to assess how the robot's personality influenced the users' emotional state. The `evea_analysis.py` script produced means and standard deviations for the six EVEA items calculated across all participants. The two resulting figures are shown below: the first displays the mean of each emotion experienced by users, and the second provides a direct comparison between shy and intense.

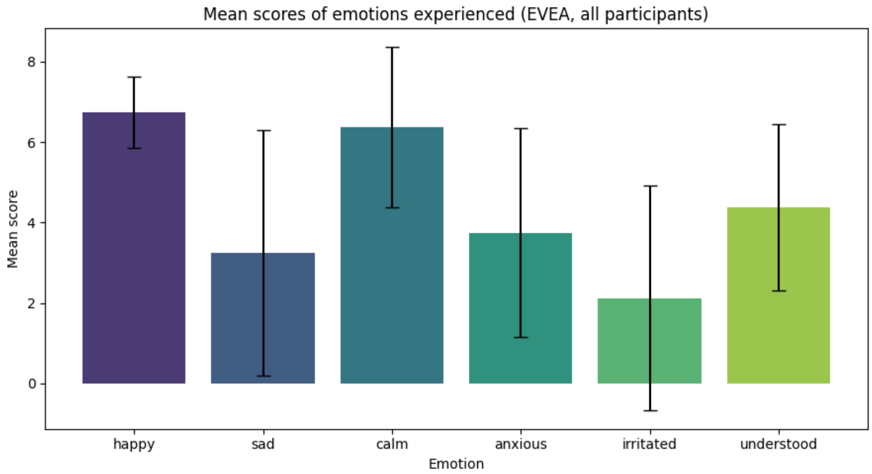

Figure 6.7: Bar plot of mean emotions measured via the EVEA scale (all participants) - pilot test.

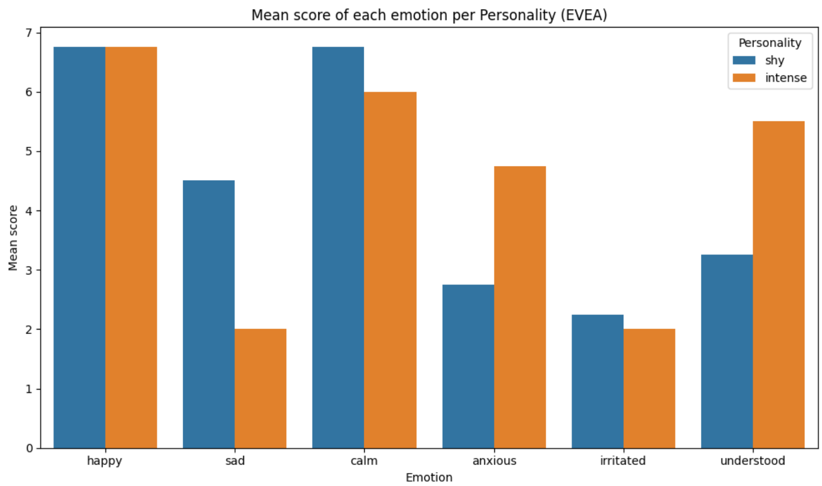

Figure 6.8: Bar plot of mean EVEA emotions by robot personality (shy vs intense) - pilot test.

The emotions "happy" and "calm" are overall high, suggesting that interaction with the robot generally elicited positive feelings and a state of calm in most participants. Conversely, emotions such as "sad" and "irritated" show lower means but large standard deviations, indicating substantial individual variability in responses.
The shy group reports a higher mean for "sad" (4.50 vs 2.00), whereas the intense group shows a much higher value for "anxious" (4.75 vs 2.75) and for "understood" (5.50 vs 3.25). This suggests that the robot's personality coherently modulates

users' emotional responses: the shy mode tends to evoke reactions oriented towards empathy or discomfort, while the intense mode appears to evoke greater activation (anxiety) but also a sense of being understood or recognised (higher "understood").

It should be emphasised that, given the brevity of the interaction and the inherently subjective nature of self-reported emotions, conducting robust statistical analyses on these measures is particularly challenging. In addition, the experimental system relies on facial recognition to modulate the robot's reactions: this implies that the actual interaction experience is not identical for all participants but highly personal. The elevated standard deviations highlight the strong heterogeneity of individual responses. Consequently, the EVEA results reported here should be regarded as exploratory and interpreted in light of these limitations.

6.4.2 Results of the final optimised system test

The optimised emotional system, obtained after the improvements described earlier, was validated in two experimental sessions held on 20th and 30th October, involving a total of 28 participants. The subjects were divided into two balanced groups ($n = 14$ per group): each interacted first with the Neutral version of the robot and subsequently with the Personalised version corresponding to the assigned personality (shy or intense).
The preliminary questionnaire revealed the following demographic and familiarity characteristics: participants' ages clustered around 24–25 years on average (range 18–43), with an overall gender composition of 6 females and 22 males. General familiarity with technology was, on average, medium–high (approximate means: shy ≈ 4.29, intense ≈ 3.64), while specific familiarity with robotic technologies was slightly lower (approximate means: shy ≈ 3.64, intense ≈ 3.29). Only a minority of participants (5 out of 28, $\approx 18\%$) had previously interacted with the TIAGo platform.

Following the same operational and data-preparation procedure, the Python script `godspeed_analysis.py` analysed the individual means computed on the Godspeed scales (Animacy: mean of 5 items; Likeability: mean of 3 items), reported in Table 6.2, while the data used by the Python script `evea_analysis.py` are the EVEA scores (0–10 scales for six emotions) collected at the end of the interaction, reported in Table 6.4
As in the pilot test, for each measure (Animacy and Likeability) the results of the mixed ANOVA are reported below (values of F, p and η_p^2) for the three model terms: main effect of Personality (between), main effect of Condition (within), and Personality $\times$ Condition interaction.

Measure	**Effect**	**F**	**p**	η_p^2
Animacy	Personality	0.656	0.4254	0.025
Animacy	Condition	9.027	0.0058	0.258
Animacy	Interaction	1.194	0.2846	0.044
Likeability	Personality	9.231	0.0054	0.262
Likeability	Condition	18.971	0.0002	0.422
Likeability	Interaction	0.319	0.5772	0.012

Table 6.6: Results of ANOVA tests for Animacy and Likeability measures — final tests.

The values reported in the table confirm that the experimental manipulation had clear and substantive effects on subjective evaluations.
For Animacy, the main effect of Condition (Neutral vs Personalised) is significant (p = 0.0058) with a medium-to-large effect size ($\eta_{\mathrm{p}}^2 = 0.258$). This means that participants perceived the personalised version of the robot as more "alive" than the neutral one. The main effect of Personality and the Personality $\times$ Condition interaction are not significant for Animacy, indicating that the increase in perceived lifelikeness due to personalisation occurred similarly for both the shy and intense personalities: participants noticed the difference between Personalised and Neutral, but did not rate the "liveliness" of the two personalities (shy vs intense) differently.
For Likeability, the picture is slightly different but consistent: both Condition (p = 0.0002, $\eta_{\mathrm{p}}^2 = 0.422$) and Personality (p = 0.0054, $\eta_{\mathrm{p}}^2 = 0.262$) show significant effects. This indicates that, beyond personalisation generally making the robot more pleasant, there is also an overall difference between the two personality profiles (shy vs intense) in the likability attributed to the robot (holding condition constant, the two groups tend to judge likeability differently). Here too, the interaction is not significant; thus, the increase in likeability from Neutral to Personalised is present for both personalities and does not appear to be differentially modulated in shy versus intense.
In summary: participants clearly perceive the personalised robot as more "alive" and more pleasant than the neutral robot; the choice of personality (shy vs intense) globally influences the likability attributed to the robot, but does not change the fact that personalisation improves perception for both personalities.

After the numerical analysis, the four plots generated by the Python script are presented below:

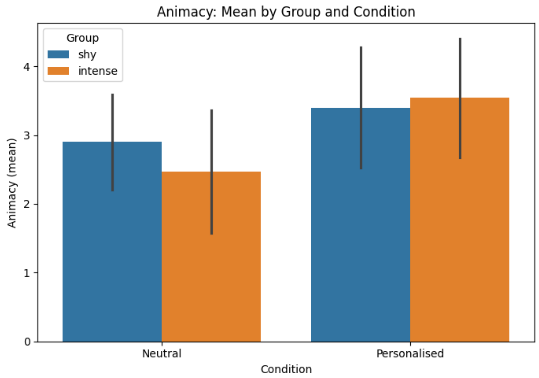

Figure 6.9: Bar plot of Animacy scale means by group and experimental condition - final tests.

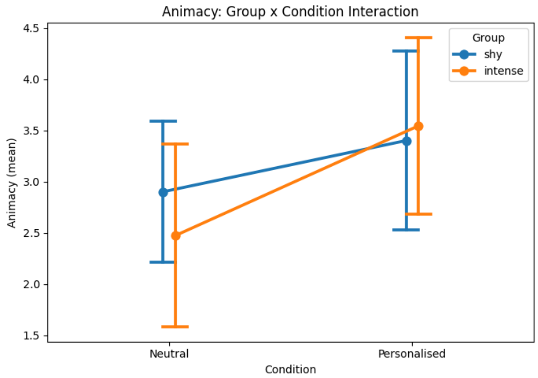

Figure 6.10: Point plot of the Group × Condition interaction for the Animacy scale - final tests.

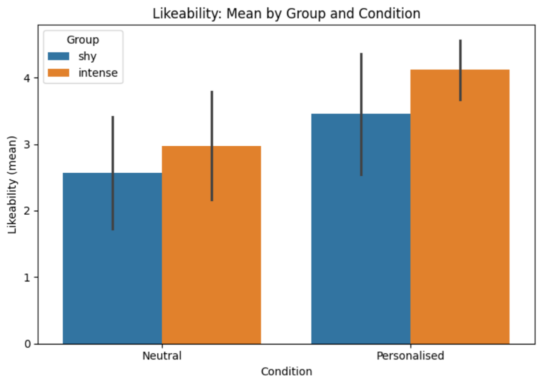

Figure 6.11: Bar plot of Likeability scale means by group and experimental condition - final tests.

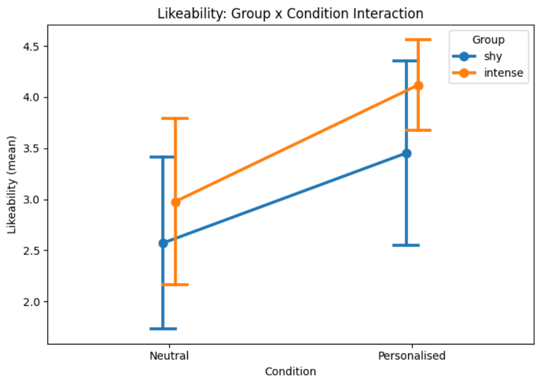

Figure 6.12: Point plot of the Group × Condition interaction for the Likeability scale - final tests.

The bar plots and point plots produced for the Animacy and Likeability scales visually show the results of the mixed ANOVA.
For Animacy, an average increase is observed when moving from Neutral to Personalised in both groups, consistent with a significant main effect of Condition. Although the curve for the intense group appears steeper in the point plot, the Personality × Condition interaction was not significant, meaning that, despite visible differences in slopes, there is no statistical evidence that personalisation has a different impact on shy versus intense personality. Participants therefore noticed the difference between Personalised and Neutral, but did not evaluate the "liveliness" of the two personalities in a substantially different way. The error bars

indicate individual variability that accounts for the non-significant interaction.
For Likeability, the plots clearly show both the effect of Condition (marked increase in Personalised) and an overall difference between groups (intense > shy). This pattern is confirmed by the ANOVA: a very robust Condition effect and a significant main effect of Personality, while the interaction remains non-significant. Practically, this means that activating the emotional system made the robot overall more pleasant for all participants, and that, regardless of condition, the two personality profiles receive different likeability ratings.

Open-ended responses collected in the questionnaires corroborate, and subjectively explain, the pattern that emerged from the Godspeed analyses.

Participants assigned to the shy personality repeatedly described a clear qualitative difference between the two behaviours. Behaviour 1 (Neutral robot, without the emotional system) was often perceived as rushed, indifferent, or "aggressive" (directly from the questionnaire responses *"At first, it felt aggressive and artificial"*, *"In robot's behaviour one, I perceived indifference, haste, and displeasure"*, *"At the beginning it did not seem very interested in my presence."* [1]), with faster movements and less attention to human interaction. By contrast, Behaviour 2 (Personalised robot, with the affective system integrated) was interpreted as more reserved yet also "more alive" and engaged: recurrent terms included shy/fearful, cautious, curious, and more alive/realistic (*"The second one was kind of shy, fearful, but at the same time felt more alive and realistic"*, *"The second robot seems to be more curious"*, *"The second one seemed more involved in the situation to me"*, *"Robot 2 tries to interact with sadness"*).
Some participants highlighted a trade-off: Behaviour 1 was more effective in task execution but potentially risky in crowded contexts, whereas Behaviour 2 was friendlier but sometimes too slow (*"Faster movements in behavior 1, more effective in task execution but dangerous if surrounded by humans"*; *"Behaviour 2 looks friendly but too slow"*).
Overall, the open-ended responses from the shy group explain why many of these participants ($\approx 57.1\%$) preferred Behaviour 2 (Personalised): it conveys signals of attentiveness and social responsiveness that strengthen impressions of "liveliness" and emotional closeness. It should nevertheless be noted that this preference is not particularly pronounced. This may be because the navigation-parameter changes introduced in the Personalised version were too gradual or too small in magnitude, and thus less perceptible to users.

[1] All open-ended questionnaire responses cited in this section have been translated into English and, where necessary, lightly edited for grammar and clarity.

In the intense group, analogous descriptions emerge with different shades of meaning: again, Behaviour 1 was often judged as detached or apathetic (*"The first robot was very apathetic, very uninterested in the situation it was in"*), whereas Behaviour 2 was perceived as more natural, curious, and human (*"Behaviour 2 felt much more natural, and less like a machine", "The second one interact more like a human"*, *"Very friendly and extroverty"*, *"Happiness to see person and curiosity"*, *"In the second personality I found a robot more similar to a human that senses what is in front of it, perceives the emotion, and acts accordingly"*).
Many observations emphasised that Behaviour 2 continually monitors the environment and responds more adaptively (*"The first one looks like takes information from the outside only once at the beginning. While the second one constantly checks its environment"*, *"Behaviour 2 is better because it stops when people get close"*), a feature appreciated as indicative of credible social behaviour. Comments in the intense group also underscored differences in speed and approach (*"First, it was fast but dismissive. Second, it was slow but more approachable"*).
The intense group's responses reinforce the idea that implementing emotional dynamics makes the robot more human-like and appealing to users: indeed, 13 out of 14 participants ($\approx 92.9\%$) indicated Behaviour 2 as their preferred option. This result suggests that, for the intense personality, the difference between the Neutral and Personalised versions was very clear and positive, providing strong evidence for the effectiveness of the emotional control system and indicating that affective modulation had a more evident subjective impact than in the shy profile.

Considering the complexity of the participants, it can therefore be argued that the majority indicated "Robot behaviour 2" as the most pleasant, the most alive, and the one with which they would prefer to interact a second time. This subjective preference corresponds to the quantitative results: the significant Condition effect on Animacy and Likeability (ANOVA) indicates that participants indeed perceived the difference between Neutral and Personalised, and the comments clarify which behavioural cues underlie this perception (greater visual orientation, pauses, adjustments in speed/proximity, "curious" or "attentive" behaviour).
The main Personality effect for Likeability is consistent with the tendency to evaluate the profiles' overall likability differently, while the absence of an interaction means that, despite these global differences, the increase due to personalisation occurred in parallel for both groups.

Following the Godspeed analysis, the two EVEA figures reported below (mean of all emotions for the entire sample and comparisons by personality shy vs intense) provide a coherent but multifaceted picture of participants' emotional experience.

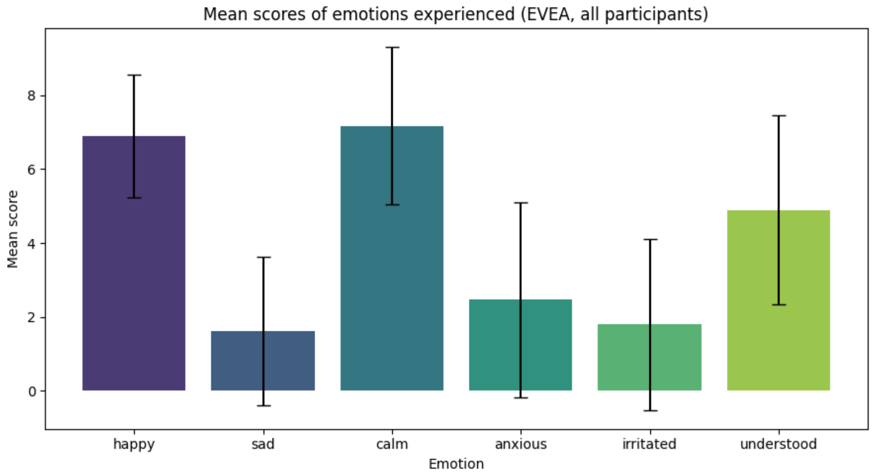

Figure 6.13: Bar plot of mean emotions measured via the EVEA scale (all participants) - final tests.

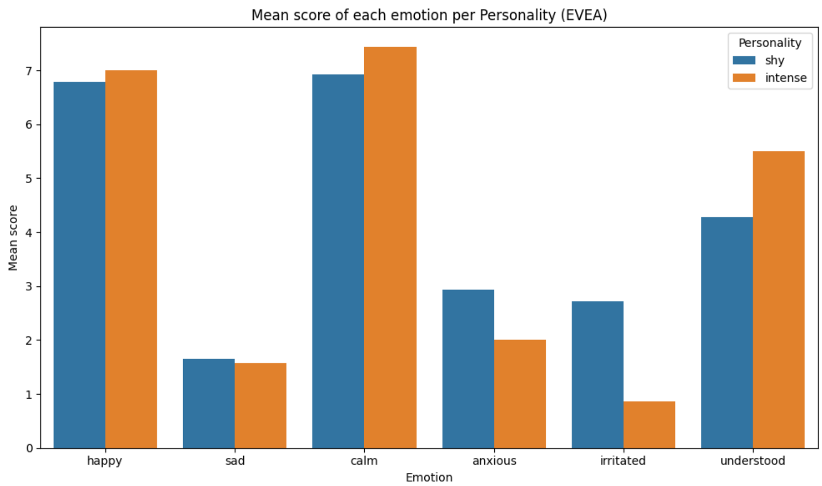

Figure 6.14: Bar plot of mean EVEA emotions by robot personality (shy vs intense) - final tests.

In overall terms, positive emotions predominate: the mean scores for “happy” and “calm” are the highest among the items, while “sad” and “irritated” show the lowest values. However, the standard deviations are relatively large for some items, indicating substantial individual variability in responses.
The comparison by personality (shy vs intense) highlights differences that are meaningful and interpretable in light of the Godspeed results: participants exposed to the intense personality report slightly higher scores for “happy” and “calm” and

a marked value for “understood”, whereas the shy group shows lower mean values for “understood” but tends to be higher for “anxious” and “irritated”. This suggests that the intense personality tends to evoke positive emotions and a stronger sense of being understood, consistent with the high likeability observed for the intense profile; by contrast, the shy personality may induce a more mixed response, with a subset of participants experiencing mild negative activation (anxiety/annoyance), likely linked to signals of shyness or discomfort perceived as harder to interpret at first glance or too slow.
Given the brevity of the interaction and the subjective nature of the responses, these results remain exploratory: the large standard deviations indicate that not all users had the same emotional experience and that the differences should be corroborated with targeted statistical tests. In any case, the EVEA figures support the general conclusion that activating the emotional system elicited predominantly positive reactions and that the two personalities modulate the emotional experience in distinct ways, an element that can inform future adjustments of the fuzzy rules to maximise both likeability and users’ perceived comfort.

6.4.3 Comparison between the two experimental campaigns

The comparison between the pilot test (N = 8) and the final test with the optimised affective system (N = 28) reveals both confirmation of the initial exploratory results and some noteworthy changes in their interpretation, attributable to the increased statistical power and to the technical modifications introduced.
In both cases, the main effect of Condition (Neutral vs Personalised) is present: participants consistently perceive the shift to the personalised version of the robot as an increase in animacy and pleasantness. In the pilot test, the Personalised condition already produced a significant increase in the Animacy and Likeability scales, but the main effect of Personality was limited to Animacy alone, with medium-to-small η^2_{p} values and substantial individual variability (limitations in significance and precision).
In the optimised version, by contrast, the effects become more stable and pronounced: for both scales, the main effect of Condition is highly significant, indicating that activating the emotional system consistently improves perceptions of the robot’s vitality and pleasantness. Moreover, the significance of the Personality effect for Likeability (not present in the pilot) suggests that the implemented modifications made the two behavioural profiles more distinguishable and recognisable, conferring coherence and a specific identity to the shy and intense personalities.
This “shift” of the Personality effect can be explained by two main factors:

1. the increase in sample size improved estimator precision and thus the ability to detect genuine effects across the different measures;

2. the technical enhancements (reduced latency in the acquisition/processing pipeline and greater perceptual salience of navigation-parameter updates) made aspects related to overall pleasantness (Likeability) more evident, while perceived differences in "liveliness" (Animacy) became more uniform across profiles.

From a qualitative standpoint, participants' open-ended responses reveal an equally significant evolution. In the pilot test, descriptions tended to emphasise perceived differences that were often vague or ambivalent, with comments alternating between curiosity and confusion about the robot's behaviour. In the final sessions, by contrast, impressions are more convergent: the personalised version is systematically described as more natural, attentive, and "alive", whereas the neutral mode is perceived as rigid or artificial.
With respect to the EVEA emotional measures, the comparison between the pilot and the final test shows consistency in the overall trend (high "happy" and "calm", low "sad" and "irritated"): in both studies the intense personality tends to elicit more positive responses and a stronger sense of "being understood", while the shy personality is associated with more mixed responses (greater variability, some signs of anxiety or discomfort for a subset of participants). The large standard deviations in both phases underline the subjective heterogeneity of the experience and the need to regard these results as indicative rather than definitively generalisable.
The pilot exhibited the typical limitations of a small sample (imprecise effect estimates, susceptibility to overestimation); it was therefore appropriate to treat those results as exploratory and as guidance for technical optimisations. The final test, while having greater power, still presents limitations to consider: a relatively young and gender-imbalanced sample, brief interactions, and possible variability in facial-recognition conditions across participants.

The transition from the pilot to the final test strengthened the evidence that integrating a fuzzy emotional control system, driven by real-time facial recognition, increases perceived Animacy and Likeability of the robot. The technical optimisations made these effects more evident and statistically robust, especially for Likeability. At the same time, differences between the shy and intense profiles emerge more clearly in overall likeability evaluations and subjective preferences (with intense receiving a more marked preference), indicating that personalising the affective dynamics is a promising lever, but one that requires further parameter tuning (magnitude and timing) according to the desired profile.

Chapter 7

Conclusions and Future Works

This thesis investigated how an affective control architecture, based on fuzzy logic and real-time emotion recognition, can enhance the naturalness and perceived quality of human–robot interaction in autonomous mobile robots.
The work focused on the development of an emotional navigation system capable of adapting its behaviour based on the user's facial expressions and this system was was tested on the TIAGo platform. The system integrates deep-learning-based emotion detection with two internal affective dimensions, mood and alertness, which influence the robot's linear velocity and navigation tuning parameters through a hierarchy of fuzzy controllers.

7.1 Summary of Contributions

During the course of this project, several contributions to the field of socially interactive robotics were provided, with a particular focus on affective navigation and the integration of emotional systems into autonomous platforms.
A first contribution concerns the integration of a real-time facial-expression recognition algorithm (https://github.com/Harshxth/Real-Time-Expression-Detection) directly on the TIAGo robot, using its onboard RGB camera and deploying the entire perception pipeline within the ROS framework. The resulting module enables the robot to infer the user's emotional state autonomously and without external hardware, ensuring real-time performance and seamless integration with the subsequent stages of the affective architecture.
Building on this perception layer, the thesis introduces a complete emotional control system that combines deep-learning-based emotion recognition with fuzzy-logic modelling of affective variables, developed by taking as reference and extending a

previous emotional fuzzy architecture designed within the Universidad Politécnica de Madrid. [1]
Two internal dimensions—mood and alertness—were defined and modelled through separate fuzzy controllers, enabling smooth and human-like transitions between affective states. These internal variables were then linked to measurable navigation parameters, specifically linear velocity and navigation tuning, allowing the robot's movement to serve as a direct expression of its internal emotional dynamics.
A further contribution of this work is the design and implementation of two distinct robot personalities, shy and intense, obtained by modifying the fuzzy-rule base and the sensitivity of the affective variables. This demonstrates how subtle variations in rule structures and membership-function parameters can produce qualitatively different behavioural profiles while maintaining safety and stability.
Finally, the thesis contributes an experimental validation of the proposed system through user studies involving 28 participants. Using standardised HRI instruments—Godspeed (Animacy and Likeability) and EVEA—combined with qualitative feedback, the experiments showed that affectively modulated navigation significantly improves users' perception of the robot. Participants described the emotional robot as more alive, attentive, and engaging, confirming the positive effect of affect modulation in autonomous behaviour.

Overall, this thesis offers a unified, ROS-integrated, and experimentally validated framework for emotional navigation, demonstrating that embedding affective dynamics into robot control can meaningfully enhance human–robot interaction.

7.2 Future Works

While the emotional navigation system developed in this thesis proved effective, several promising directions may further expand its capabilities and applicability. The following proposals outline potential future developments.

- Extension of emotional dimensions and personality models
  The system currently relies on two internal emotional variables, mood and alertness. Future work could incorporate additional psychological constructs, such as interest, affective or expectancy, enabling a richer behavioural repertoire. Moreover, adopting more sophisticated personality models inspired by psychological theories could allow robots to display more nuanced and stable long-term behavioural identities.

- Multimodal emotion perception
  The current approach relies exclusively on facial-expression detection. Integrating multimodal inputs, such as vocal cues, physiological signals, gestures,

predominant colors in the user's clothes or contextual information, would significantly improve emotional inference, especially in ambiguous or low-visibility situations.

- Adaptive learning of fuzzy rules
  The fuzzy-logic system has been manually designed to ensure interpretability and personality consistency. Future work may explore adaptive mechanisms that allow the robot to refine or expand its rule base autonomously. Promising directions include:

  – adaptive fuzzy systems that learn or fine-tune rules from user interaction data, gradually shaping behaviour to individual preferences;

  – reinforcement learning techniques to optimise affective responses through trial-and-error interaction;

  – hybrid neuro-fuzzy approaches, where neural networks support the automatic adjustment of membership functions or rule parameters.

  Current research is evolving along two main directions: refining fuzzy formalisms to handle more complex forms of uncertainty, such as type-2 fuzzy systems, which are particularly effective when dealing with ambiguous or noisy affective cues, and integrating fuzzy logic with learning-based methods to create adaptive architectures capable of autonomously updating their rule bases.

- User modelling and Long-Term interaction
  The experiments conducted focused on single-session interactions. A natural extension would be to study long-term effects. Implementing user-profile memory or long-term adaptation could support sustained, personalised assistance.

- Improved robot behavioural expressivity
  Motion expressivity could be expanded by modulating additional navigation parameters or non-verbal channels. For example, head and arm movements or the addition of a vocal response Combining navigation patterns with expressive cues could significantly enhance the clarity and communicative value of the robot's emotional state.

- Large-scale user studies
  Larger datasets would provide stronger statistical power and enable generalisable conclusions about affective robotics.

Appendix A

Manual

1. Environment setup

Before starting the system, ensure that the following conditions are met:

- ROS Noetic is installed and sourced;
- the TIAGo robot is powered on and connected to the laboratory Wi-Fi network;
- the local PC is connected to the TIAGo Wi-fi network;
- the PC's ROS workspace contains the required packages:
  - `camera_processor`
  - `fuzzy_emotion_control`
- the virtual environment (venv) is available and properly configured;
- the trained facial expression model `emotiondetector.h5` is present in the correct directory;
- he correct RViz configuration file (`rviz_TIAGo.rviz`) is available for visualization.

2. Connection with TIAGo

Open a terminal on the PC and establish remote access to the robot's onboard computer:

```
ssh pal@tiago-162c
```

Password: .....

This terminal will be referred to as TIAGo Terminal, and it will be used exclusively for operations that must run directly on the robot.

3. Change the map on the robot (TIAGo Terminal)

To load the desired navigation map (for example hall_map), execute:

```
cd .pal/maps/configurations/

rosservice call /pal_map_manager/change_map "input: 'hall_map'"
```

4. Verify the local IP address

Before launching any ROS node locally, check the IP address of your machine:

```
ifconfig
```

This step is essential because ROS requires explicit network configuration.
The IP address shown here will be used in the next step to ensure correct communication between the local PC and TIAGo.

5. Configuration of additional terminals

Every terminal that runs ROS nodes on the local PC must be configured with the following environment variables:

```
source /opt/ros/noetic/setup.bash

export ROS_MASTER_URI=http://tiago-162c:11311

export ROS_IP=10.68.0.129
```

Without this configuration, the nodes running on your computer will not be able to communicate with the robot.

6. Launching RViz

RViz provides a visual interface to monitor the robot state, camera view, transformations, and navigation:

```
cd /home/carlotta/catkin_ws/src/tiago

rviz -d ./rviz_TIAGo.rviz
```

7. Launching the real-time facial expression detection system

In this step, the perception pipeline running on the local PC is activated. This node processes the images from the robot's RGB camera and publishes a label describing the detected facial expression. This topic is consumed by the fuzzy emotional controller in the next step.

```
cd ~/Real-Time-Expression-Detection

source venv/bin/activate

export
PYTHONPATH=$PYTHONPATH:/opt/ros/noetic/lib/python3/dist-packages

source ~/catkin_ws/devel/setup.bash

rosrun camera_processor camera_listener.py
```

8. Launching the fuzzy controller (shy or intense personality)

This step activates the core of the emotional system.
The fuzzy controller receives: the detected facial expression and the robot's current linear velocity and generates the robot's internal emotional state (mood, alertness) and the navigation modulation parameters (speed, tuning).

```
cd catkin_ws

source /opt/ros/noetic/setup.bash

catkin_make

source ~/catkin_ws/devel/setup.bash
```

To launch the shy personality:

```
rosrun fuzzy_emotion_control fuzzy_node.py shy
```

To launch the intense personality:

```
rosrun fuzzy_emotion_control fuzzy_node.py intense
```

Appendix B

Questionnaire

Questionnaire on Emotional Interaction with the TIAGo Robot

Thank you for taking part in this experiment.

The aim of this questionnaire is to collect information about your perceptions and emotions during the interaction with the TIAGo robot.

The questionnaire takes about 5 minutes to complete.
There are no right or wrong answers; we are only interested in your personal experience.

By completing this questionnaire, you consent to the use of the collected data exclusively for research and statistical purposes.
All responses will be treated anonymously.

Thank you for your collaboration!

1 - General Information
(anonymous, for statistical purposes only)

- **Age:** ....
- **Gender:**
  ☐ M ☐ F ☐ Other
- **Familiarity with technology knowledge:**

Very little knowledge	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐	Very high knowledge

- **Familiarity with robotic technologies:**

Not at all familiar	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐	Highly familiar

- **Have you previously interacted with TIAGo?**

  ☐ Yes ☐ No

2 - Perception of the Robot

In the following section, you will be asked to rate your impression of the robot's general characteristics (e.g., how "alive", "organic", or "likeable" it appeared), using standardized scales from the Godspeed questionnaire.

For each statement, select the number that best reflects your perception for each statement from 1 to 5, indicating how strongly you perceived the presence of that item in the robot's behaviour.

Robot behaviour 1

Animacy scale:

	1	**2**	**3**	**4**	**5**	
Dead	☐	☐	☐	☐	☐	Alive
Stagnant	☐	☐	☐	☐	☐	Lively
Mechanical	☐	☐	☐	☐	☐	Organic
Artificial	☐	☐	☐	☐	☐	Lifelike
Apathetic	☐	☐	☐	☐	☐	Responsive

Likeability scale:

	1	**2**	**3**	**4**	**5**	
Dislike	☐	☐	☐	☐	☐	Like
Unfriendly	☐	☐	☐	☐	☐	Friendly
Unpleasant	☐	☐	☐	☐	☐	Pleasant

Robot behaviour 2

Animacy scale:

	1	2	3	4	5	
Dead	☐	☐	☐	☐	☐	Alive
Stagnant	☐	☐	☐	☐	☐	Lively
Mechanical	☐	☐	☐	☐	☐	Organic
Artificial	☐	☐	☐	☐	☐	Lifelike
Apathetic	☐	☐	☐	☐	☐	Responsive

Likeability scale:

	1	2	3	4	5	
Dislike	☐	☐	☐	☐	☐	Like
Unfriendly	☐	☐	☐	☐	☐	Friendly
Unpleasant	☐	☐	☐	☐	☐	Pleasant

Open question:

Which emotions did you perceive in the robot? ....

Preference between the two robots

- Which version of the robot did you find more natural?
  ☐ Robot behaviour 1 ☐ Robot behaviour 2
- Which version of the robot did you find more pleasant to interact with?
  ☐ Robot behaviour 1 ☐ Robot behaviour 2
- Which version of the robot seemed more alive or expressive?
  ☐ Robot behaviour 1 ☐ Robot behaviour 2
- In general, which robot would you prefer to interact with again?
  ☐ Robot behaviour 1 ☐ Robot behaviour 2

Open question:

Please describe briefly what differences you noticed between the two robots, if any. ....

3 - Personal Emotions of the Participant

Please indicate how you personally felt during the interaction with the robot. For each statement, use the scale from 0 to 10, where:

- 0 = not at all
- 10 = very much

Indicate the number that best represents your own emotional state.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
I felt happy	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
I felt sad	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
I felt calm	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
I felt anxious	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
I felt irritated	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
I felt understood	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Open question:

Do you have any comments or suggestions about the interaction with the robot? ....

Appendix C

Python Scripts for ANOVA

C.1 Script: godspeed_analysis.py

```python
import pandas as pd
import pingouin as pg
import seaborn as sns
import matplotlib.pyplot as plt

# Load Excel file
df = pd.read_excel('godspeed_test.xlsx')

# ========================================
# ANIMACY
# ========================================
# Prepare data in "long" format
df_animacy = pd.melt(df,
    id_vars=['ID', 'Personality'],
    value_vars=['Animacy_mean_Neutral', 'Animacy_mean_Personalised
    '],
    var_name='Condition',
    value_name='Animacy_mean'
)

df_animacy['Condition'] = df_animacy['Condition'].replace({
    'Animacy_mean_Neutral': 'Neutral',
    'Animacy_mean_Personalised': 'Personalised'
})

# Mixed ANOVA for Animacy
print("=== Mixed ANOVA Results for Animacy ===\n")
aov_animacy = pg.mixed_anova(
    dv='Animacy_mean',
    within='Condition',
```

```
    between='Personality',
    subject='ID',
    data=df_animacy
)
print(aov_animacy)

# BARPLOT Animacy
plt.figure(figsize=(7, 5))
sns.barplot(data=df_animacy, x='Condition', y='Animacy_mean', hue=
    'Personality', errorbar='sd')
plt.title('Animacy: Mean by Group and Condition')
plt.ylabel('Animacy (mean)')
plt.xlabel('Condition')
plt.legend(title='Group')
plt.tight_layout()
plt.show()

# POINTPLOT Animacy (interaction)
plt.figure(figsize=(7, 5))
sns.pointplot(
    data=df_animacy,
    x='Condition',
    y='Animacy_mean',
    hue='Personality',
    dodge=True,
    markers='o',
    capsize=.1,
    errorbar='sd'
)
plt.title('Animacy: Group x Condition Interaction')
plt.ylabel('Animacy (mean)')
plt.xlabel('Condition')
plt.legend(title='Group')
plt.tight_layout()
plt.show()

# ========================================
# LIKEABILITY
# ========================================
# Prepare data in "long" format
df_likeability = pd.melt(df,
    id_vars=['ID', 'Personality'],
    value_vars=['Likeability_mean_Neutral', '
    Likeability_mean_Personalised'],
    var_name='Condition',
    value_name='Likeability_mean'
)

```

```
df_likeability['Condition'] = df_likeability['Condition'].replace({
    'Likeability_mean_Neutral': 'Neutral',
    'Likeability_mean_Personalised': 'Personalised'
})

# Mixed ANOVA for Likeability
print("\n=== Mixed ANOVA Results for Likeability ===\n")
aov_likeability = pg.mixed_anova(
    dv='Likeability_mean',
    within='Condition',
    between='Personality',
    subject='ID',
    data=df_likeability
)
print(aov_likeability)

# BARPLOT Likeability
plt.figure(figsize=(7, 5))
sns.barplot(data=df_likeability, x='Condition', y='Likeability_mean', hue='Personality', errorbar='sd')
plt.title('Likeability: Mean by Group and Condition')
plt.ylabel('Likeability (mean)')
plt.xlabel('Condition')
plt.legend(title='Group')
plt.tight_layout()
plt.show()

# POINTPLOT Likeability (interaction)
plt.figure(figsize=(7, 5))
sns.pointplot(
    data=df_likeability,
    x='Condition',
    y='Likeability_mean',
    hue='Personality',
    dodge=True,
    markers='o',
    capsize=.1,
    errorbar='sd'
)
plt.title('Likeability: Group x Condition Interaction')
plt.ylabel('Likeability (mean)')
plt.xlabel('Condition')
plt.legend(title='Group')
plt.tight_layout()
plt.show()
```

Listing C.1: Python script used for the ANOVA analysis of Godspeed questionnaire responses.

C.2 Script: evea_analysis.py

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the EVEA data from Excel
df = pd.read_excel('evea_test.xlsx')

# Ensure Personality column exists and set order (shy first, then
    intense)
if 'Personality' in df.columns:
    df['Personality'] = df['Personality'].astype(str)
    df['Personality'] = pd.Categorical(df['Personality'],
    categories=['shy', 'intense'], ordered=True)

# Define the emotion columns
emotions = ['happy', 'sad', 'calm', 'anxious', 'irritated', '
    understood']

# ===========================
# GENERAL ANALYSIS (ALL PARTICIPANTS)
# ===========================

# Calculate mean and std for each emotion (all participants)
mean_emotions = df[emotions].mean()
std_emotions = df[emotions].std()

print("=== EVEA Questionnaire: All Participants ===")
print("Mean per emotion:\n", mean_emotions)
print("\nStandard deviation per emotion:\n", std_emotions)

# Barplot: mean per emotion (all participants)
plot_df = pd.DataFrame({'emotion': mean_emotions.index, 'mean':
    mean_emotions.values, 'std': std_emotions.values})
plt.figure(figsize=(9,5))
ax = sns.barplot(x='emotion', y='mean', data=plot_df, palette='
    viridis', errorbar=None)
# add manual errorbars (mean +/- std)
for i, row in plot_df.iterrows():
    bar = ax.patches[i]
    cx = bar.get_x() + bar.get_width()/2
    cy = row['mean']
    err = row['std'] if not pd.isna(row['std']) else 0
    ax.errorbar(cx, cy, yerr=err, fmt='none', c='black', capsize
    =5)
plt.title('Mean scores of emotions experienced (EVEA, all
    participants)')
```

```
plt.ylabel('Mean score')
plt.xlabel('Emotion')
plt.tight_layout()
plt.show()

# ===========================
# ANALYSIS BY PERSONALITY (shy vs intense)
# ===========================

# Calculate mean and std for each emotion per Personality
group_mean = df.groupby('Personality')[emotions].mean()
group_std = df.groupby('Personality')[emotions].std()

print("\n=== EVEA Questionnaire: Means per Personality ===")
print(group_mean)
print("\n=== EVEA Questionnaire: Standard deviations per Personality ===")
print(group_std)

# Prepare data for group barplot
group_mean_reset = group_mean.reset_index()
group_melt = group_mean_reset.melt(id_vars='Personality', var_name='emotion', value_name='mean')

hue_order = ['shy', 'intense']
palette_shy_intense = ['#1f77b4', '#ff7f0e']  # blue, orange

plt.figure(figsize=(10,6))
sns.barplot(data=group_melt, x='emotion', y='mean', hue='Personality',
            palette=palette_shy_intense, hue_order=hue_order, ci=None)
plt.title('Mean score of each emotion per Personality (EVEA)')
plt.ylabel('Mean score')
plt.xlabel('Emotion')
plt.legend(title='Personality')
plt.tight_layout()
plt.show()
```

Listing C.2: Python script used for the ANOVA analysis of EVEA questionnaire data.

Bibliography

[1] G. F.-B. Martín, F. Matía, L. G. Gómez-Escalonilla, D. Galan, M. G. Sánchez-Escribano, P. de la Puente, and M. Rodríguez-Cantelar, «An emotional model based on fuzzy logic and social psychology for a personal assistant robot», *Applied Sciences*, vol. 13, p. 3284, 2023. DOI: `https://doi.org/10.3390/app13053284`.

[2] F. Yan, A. M. Iliyasu, and K. Hirota, «Emotion space modelling for social robots», *Engineering Applications of Artificial Intelligence*, vol. 100, p. 104 188, 2021. DOI: `https://doi.org/10.1016/j.engappai.2021.104178`.

[3] C. Breazeal., «Toward sociable robots», *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 167–175, 2003. DOI: `https://doi.org/10.1016/S0921-8890(02)00373-1`.

[4] A. Paiva, I. Leite, H. Boukricha, and I. Wachsmuth., «Empathy in virtual agents and robots: A survey», *ACM Transactions on Interactive Intelligent Systems*, vol. 7, no. 3, 11:1–11:40, 2017. DOI: `https://doi.org/10.1145/2912150`.

[5] S. Park and M. Whang., «Empathy in human–robot interaction: Designing for social robots», *International Journal of Environmental Research and Public Health*, vol. 19, no. 3, p. 1889, 2022. DOI: `https://doi.org/10.3390/ijerph19031889`.

[6] I. Leite, A. Pereira, G. Castellano, S. Mascarenhas, C. Martinho, and A. Paiva, «Modelling empathy in social robotic companions», in *Advances in User Modeling*, ser. Lecture Notes in Computer Science, L. Ardissono and T. Kuflik, Eds., vol. 7138, Springer, 2012, pp. 135–147. DOI: `https://doi.org/10.1007/978-3-642-28509-7_14`.

[7] S. Cano, C. S. González, R. M. Gil-Iranzo, and S. Albiol-Pérez., «Affective communication for socially assistive robots (sars) for children with autism spectrum disorder: A systematic review», *Sensors*, vol. 21, no. 15, p. 5166, 2021. DOI: `https://doi.org/10.3390/s21155166`.

[8] J. Złotowski, K. Yogeeswaran, and C. Bartneck, «Can we control it? autonomous robots threaten human identity, uniqueness, safety, and resources», *International Journal of Human-Computer Studies*, vol. 100, pp. 48–54, 2017. DOI: `https://doi.org/10.1016/j.ijhcs.2016.12.008`.

[9] A. Andriella, C. Torras, and G. Alenyà, «Short-term human–robot interaction adaptability in real-world environments», *International Journal of Social Robotics*, vol. 12, pp. 639–657, 2020. DOI: `https://doi.org/10.1007/s12369-019-00606-y`.

[10] U. Maniscalco, A. Minutolo, P. Storniolo, and M. Esposito, «Towards a more anthropomorphic interaction with robots in museum settings: An experimental study», *Robotics and Autonomous Systems*, vol. 171, p. 104 561, 2024, ISSN: 0921-8890. DOI: `https://doi.org/10.1016/j.robot.2023.104561`.

[11] H. Mobahi and S. Ansari, «Fuzzy perception, emotion and expression for interactive robots», in *Proceedings of the 2003 IEEE International Conference on Systems, Man and Cybernetics (SMC'03)*, vol. 4, Washington, DC, USA: IEEE, 2003, pp. 3918–3923. DOI: `https://doi.org/10.1109/ICSMC.2003.1244500`.

[12] F. Leu, J. Liu, Y. Hsu, and Y. Huang, «The simulation of an emotional robot implemented with fuzzy logic», *Soft Comput*, vol. 18, pp. 1729–1743, 2014. DOI: `https://doi.org/10.1007/s00500-013-1217-1`.

[13] D. Dell'Anna and A. Jamshidnejad, «Evolving fuzzy logic systems for creative personalized socially assistive robots», *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105 064, 2022. DOI: `https://doi.org/10.1016/j.engappai.2022.105064`.

[14] W. Fang, F. Chao, C. Lin, L. Yang, C. Shang, and C. Zhou, «An improved fuzzy brain emotional learning model network controller for humanoid robots», *Front. Neurorobot*, vol. 13, 2019. DOI: `https://doi.org/10.3389/fnbot.2019.00002`.

[15] J. Jang, «Anfis: Adaptive-network-based fuzzy inference system», *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993. DOI: `https://doi.org/10.1109/21.256541`.

[16] M. Sajjad, F. Ullah, M. Ullah, G. Christodoulou, F. Cheikh, and M. Hijji, «A comprehensive survey on deep facial expression recognition: Challenges, applications, and future guidelines», *Alexandria Engineering Journal*, vol. 68, pp. 817–840, 2023. DOI: `https://doi.org/10.1016/j.aej.2023.01.017`.

[17] M. Maroto-Gómez, S. Álvarez-Arias, J. Rodríguez-Huelves, A. Segura-Bencomo, and M. Malfaz, «A robot companion with adaptive object preferences and emotional responses enhances naturalness in human–robot interaction», *Electronics*, vol. 14, p. 3711, 2025. DOI: `https://doi.org/10.3390/electronics14183711`.

[18] M. Bajones, D. Fischinger, A. Weiss, P. D. L. Puente, D. Wolf, and et al., «Results of field trials with a mobile service robot for older adults in 16 private households», *ACM Transactions on Human-Robot Interaction*, vol. 9, no. 2, pp. 1–27, 2019. DOI: `https://doi.org/10.1145/3368554`.

[19] C. Bartneck, D. Kulić, E. Croft, and S. Zoghbi, «Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots», *International Journal of Social Robotics*, vol. 1, pp. 71–81, 2009. DOI: `https://doi.org/10.1007/s12369-008-0001-3`.

[20] N. Spatola, B. Kühnlenz, and G. Cheng, «Perception and evaluation in human–robot interaction: The human–robot interaction evaluation scale (hries)—a multicomponent approach of anthropomorphism», *International Journal of Social Robotics*, vol. 13, pp. 1517–1539, 2021. DOI: `https://doi.org/10.1007/s12369-020-00667-4`.

[21] T. del Pino-Sedeño, W. Peñate, and J. M. Bethencourt, «La escala de valoración del estado de ánimo (evea): Análisis de la estructura factorial y de la capacidad para detectar cambios en estados de ánimo», *Análisis y Modificación de Conducta*, vol. 36, no. 153–154, pp. 19–32, 2010. DOI: `https://doi.org/10.33776/amc.v36i153-154.1058`.

[22] H.-Y. Kim, «Analysis of variance (anova) comparing means of more than two groups», *Restorative Dentistry and Endodontics*, vol. 39, no. 1, pp. 74–77, 2014. DOI: `https://doi.org/10.5395/rde.2014.39.1.74`.

[23] T. K. Kim, «Understanding one-way anova using conceptual figures», *Korean Journal of Anesthesiology*, vol. 70, no. 1, pp. 22–26, 2017. DOI: `https://doi.org/10.4097/kjae.2017.70.1.22`.

[24] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Second. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1988, ISBN: 978-0805802832.

[25] A. Kuznetsova, R. H. B. Christensen, C. Bavay, and P. B. Brockhoff, «Automated mixed anova modeling of sensory and consumer data», *Food Quality and Preference*, vol. 40, pp. 31–38, 2015. DOI: `https://doi.org/10.1016/j.foodqual.2014.08.004`.

[26] P. Robotics, *Tiago – mobile manipulator robot*, `https://pal-robotics.com/robot/tiago/`, Accessed: 2025-04-22.

[27] P. Robotics, *Tiago single (one arm) handbook*, `https://docs.pal-robotics.com/tiago-single/handbook.html#`, Accessed: 2025-04-22.

[28] P. Robotics, *Tiago base handbook — navigation*, `https://docs.pal-robotics.com/tiago-base/handbook.html#`, Accessed: 2025-04-22.

[29] P. E. Hart, N. J. Nilsson, and B. Raphael, «A formal basis for the heuristic determination of minimum cost paths», *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, 1968. DOI: `https://doi.org/10.1109/TSSC.1968.300136`.