Adaptative Social Navigation for Robots operating in Human Environments in real ecological use-cases

Philip Scales

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Informatique
Unité de recherche : Laboratoire d'Informatique de Grenoble

Navigation Sociale Adaptative du Robot en Environnement Humain dans une Ecologie Réelle d'Usages

Adaptative Social Navigation for Robots operating in Human Environments in real ecological use-cases

Présentée par :

Philip SCALES

Direction de thèse :

Olivier AYCARD PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES	Directeur de thèse
Véronique AUBERGE CHARGEE DE RECHERCHE, CNRS DELEGATION ALPES	Co-encadrante de thèse

Rapporteurs :

MOHAMED CHETOUANI
PROFESSEUR DES UNIVERSITES, SORBONNE UNIVERSITE
ADRIANA TAPUS
PROFESSEURE DES UNIVERSITES, ENSTA PARIS

Thèse soutenue publiquement le **10 janvier 2024**, devant le jury composé de :

PATRICK REIGNIER, PROFESSEUR DES UNIVERSITES, GRENOBLE INP	Président
OLIVIER AYCARD, PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES	Directeur de thèse
MOHAMED CHETOUANI, PROFESSEUR DES UNIVERSITES, SORBONNE UNIVERSITE	Rapporteur
ADRIANA TAPUS, PROFESSEURE DES UNIVERSITES, ENSTA PARIS	Rapporteure
IOANA OCNARESCU, CHERCHEURE, STRATE DESIGN	Examinatrice

Invités :

VERONIQUE AUBERGE
CHARGEE DE RECHERCHE, CNRS DELEGATION ALPES

Acknowledgement

Tout d'abord, je tiens à remercier les membres de mon jury. Merci aux rapporteurs, Adriana Tapus et Mohamed Chetouani, ainsi que les examinateurs, Patrick Reignier et Ioana Ocnarescu. Vos retours et vos questions lors de la soutenance ont permis d'avoir des discussions très intéressantes et enrichissantes! Merci particulièrement à Mohamed pour ses conseils et sa bienveillance dans le suivi de cette thèse.

Merci infiniment à mes directeurs de thèse, Olivier Aycard et Véronique Aubergé. Merci de m'avoir proposé de me lancer dans l'exploration de la robotique et de l'interaction humain-robot avec un sujet de recherche aussi passionnant que complexe, d'abord via le stage de master puis la poursuite en doctorat. Vous m'avez toujours soutenu à la fois scientifiquement et humainement, ce qui m'a permis de mener à bien cette thèse, ainsi que de me développer en tant que chercheur et en tant que personne. Merci pour tout.

Je tiens à remercier les membres du LIG que j'ai pu côtoyer, étudiants et permanents, avec qui il était toujours agréable d'échanger que ça soit sur des questions scientifiques ou non. Merci aux membres des équipes APTIKAL et GETALP qui m'ont accueilli, même si le sujet de mes recherches sortait un peu des sujets typiques de ces équipes. Merci également au personnel administratif du LIG pour leur réactivité et leur aide dans les nombreuses démarches.

Merci à tous les enseignants qui m'ont permis d'enseigner dans leurs UE au DLST, à l'UFR IM2AG, et à l'UFR LLASIC. La découverte du milieu de l'enseignement à été très enrichissante, et les conseils de tous les collègues avec qui j'ai pu enseigner m'ont grandement aidé. Merci également à tous les formateurs du label Recherche et Enseignement Supérieur d'avoir transmis leur passion pour l'enseignement et la pédagogie. Merci également à Maëlle Joveniaux de m'avoir accueilli pour mon intervention dans l'école primaire Gustave Rivet à Domène dans le cadre de l'atelier partenaires scientifiques pour la classe.

De par le sujet pluridisciplinaire et expérimental, plusieurs personnes ont apporté leur aide à divers étapes de la thèse. Merci à Jean-Philippe Guilbaud et Djamel Hadj pour leurs conseils et implication dans la construction du corpus vidéo BotEmoMove; à Véronique, Clarisse Bayol et Ambre Davat pour la création de l'outil de test en-ligne que j'ai étendu; à Nicolas Audibert et Solange Rossato pour leurs conseils sur les analyses statistiques; à Kévin Vézirian pour ses conseils sur l'analyse des entretiens semi-dirigés; à Raphaël Luffroy pour sa participation à la mise en place de l'expérimentation à l'UFR IM2AG, ainsi que Jonathan Bleuzen, Junaid Baber, et Robert Feres pour le soutien logistique et pratique. Merci à Manon et Mykhailo d'avoir relu des parties du manuscrit de thèse, ainsi que Thibaut pour ses retours sur la présentation de soutenance. Je tiens aussi à remercier les personnes ayant donné de leur temps pour participer aux expériences en-ligne ou en personne. Un grand merci aussi au Fabmstic, à tous ceux qui ont contribué au projet RobAIR, et en particulier à Germain Lemasson et Noé Boulaye pour tout le soutien technique sur la plateforme RobAIR, que ça soit pour mes travaux de thèse ou pour l'enseignement du cours d'introduction à la robotique.

Merci à mes collègues Ambre, Pu, Romain, Raphaël et Thibaut pour les discussions intéressantes autour de la robotique, les expériences d'interaction humain-robot, et pour votre

soutien et votre amitié. Merci aussi aux nouveaux collègues et amis au GIPSA, notamment Sébastien et Ian. Thanks also to all the friends at the LIG: Pu, Karim, Rouba, Pier, Manon, Fatima, Rami, Samandar, Daniela, Ophélie, Thibaut and Junaid. Thanks for creating a supportive and welcoming environment, and for the chats, meals, trips and laughs! Thanks also to all the wonderful volunteers from IntEGre for the fun times and their dedication to welcoming international students.

Un immense merci à mes amis Sébastien et Louis pour leur soutien continu, ainsi qu'Elsa et Estelle, que ça soit pour les sessions de JDR en-ligne ou les retrouvailles en personne, malgré le fait que je n'aie pas su me rendre aussi disponible que je l'aurais souhaité. Vous êtes supers! Merci également à mes parents et mon frère Julien pour tout ce qu'ils m'ont appris et donné comme opportunités, ainsi que leurs encouragements.

Enfin, mille mercis ma Clara. Merci pour ta compréhension et ta patience face à mes angoisses et doutes. Merci d'avoir été là, et de m'avoir soutenu tout au long de ce périple.

None of this would have been possible without all of these people and their invaluable contributions as scientists and as friends, so once again, thank you all from the bottom of my heart. To all those who are still on the path, I wish you all the best!

Abstract

Companies and research groups are increasingly aiming to deploy mobile robots to perform various navigation tasks in environments shared with humans. Early attempts used classical navigation algorithms which consider humans as mere obstacles to be avoided, leading to undesirable robot behaviour. The field of Social Navigation aims to design algorithms that account for various social factors such as personal space, social norms, or predictability and smoothness of the motion. Despite these improvements, there are still issues with the acceptance of current approaches, and the complete set of variables that a social navigation algorithm should account for is not known. Humans also tend to attribute social intentions, attitudes or affect to the way in which a robot moves.

In this thesis, we explore the following questions: Which features of the robot's motion elicit different attributions of social attitudes? How can we design a navigation algorithm that provides control over these features? What is the impact of the social perception of the robot on how people interact with it and evaluate it? To work towards answering these questions, we propose to design a social navigation algorithm that can adapt the robot's motion according to its impact on human social perception, based on an understanding of human perception of mobile robots acquired through experimental studies.

Our first contribution is a model describing the mapping between variations of robot motions and human social and physical perception of the robot. We start from the basic elements of locomotion and visual appearance, using analogies to vocal prosody to guide the selection of variables constituting our corpus of robot motions. Through a series of online and in-person perception experiments we find that each corpus variable significantly impacts participant's social and physical perception of the robot, providing us with a first definition of *movement prosody* which our algorithm must be able to control.

Our second contribution is the design, implementation, and validation of a social navigation algorithm that provides precise control over the corpus motion variables. We propose a local planning algorithm that uses specially designed constraints to ensure that the variations in the robot's velocity, acceleration, and their timing are controlled in accordance with our motion corpus variables, even when changes in the dynamic environment require re-planning the trajectory. The planner is integrated into a navigation architecture, and we show its ability to execute navigation tasks while maintaining the desired movement prosody.

Our third contribution is a real-world study of the impact of movement prosody on human perception, behaviour, and performance ratings of the robot. Unlike the initial perception tests, the robot is deployed fully autonomously with our algorithm in a campus hall, where it approaches people to hand out flyers. The robot uses different sets of prosody constraints that our model predicted to be associated with *confident* or *hesitant* attitudes. The results suggest that the effect of the variations of the robot's motion on participant perception differed from our perception experiments. Analysis of interviews provide us with insights into other aspects of the robot's behaviour and motion which impacted participants' perceptions, opening the path to future work to iteratively refine the understanding of movement prosody and the navigation algorithm.

Résumé

Les entreprises et les équipes de recherche cherchent à déployer des robots mobiles pour effectuer diverses tâches de navigation dans des environnements partagés avec des humains. Les premières approches étaient basées sur des algorithmes de navigation classiques qui considèrent les humains comme de simples obstacles, ce qui entraîne un comportement indésirable des robots. Le domaine de la navigation sociale vise à concevoir des algorithmes qui tiennent compte de facteurs sociaux tels que l'espace personnel, les normes sociales ou la prévisibilité et la fluidité du mouvement. Malgré ces améliorations, des problèmes subsistent quant à l'acceptation des robots, et l'ensemble des variables qui devraient être contrôlées par un algorithme de navigation sociale n'est pas connu. D'autre part, les humains ont tendance à attribuer des intentions sociales, attitudes ou affects à la manière dont un robot se déplace.

Cette thèse aborde les questions suivantes : Quelles caractéristiques du mouvement du robot suscitent différentes attributions d'attitudes sociales ? Comment pouvons-nous concevoir un algorithme de navigation qui permette de contrôler ces caractéristiques ? Quel est l'impact de la perception sociale du robot sur la façon dont les gens interagissent avec lui et l'évaluent ? Pour répondre à ces questions, nous proposons de concevoir un algorithme de navigation sociale capable d'adapter le mouvement du robot en fonction de son impact sur la perception sociale humaine, sur la base d'une compréhension de la perception des robots mobiles acquise grâce à des études expérimentales.

Notre première contribution est un modèle décrivant la correspondance entre les variations des mouvements du robot et la perception sociale et physique du robot par les humains. Nous procédons à partir des éléments de base de la locomotion et de l'apparence visuelle, et par analogie avec la prosodie vocale pour guider la sélection des variables constituant notre corpus de mouvements de robots. Une série d'expériences de perception en ligne et en personne montre que chaque variable du corpus a un impact significatif sur la perception sociale et physique, ce qui nous donne une première définition de la *prosodie du mouvement* que notre algorithme doit être capable de contrôler.

Notre deuxième contribution est la conception, l'implémentation et la validation d'un algorithme de navigation sociale permettant de contrôler précisément les variables de mouvement du corpus. Nous proposons un algorithme de planification locale qui utilise des contraintes spécialement conçues pour s'assurer que les variations de la vitesse et de l'accélération du robot sont contrôlées conformément aux variables de notre corpus, même en environnement dynamique nécessitant une re-planification de la trajectoire. Le planificateur est intégré dans une architecture de navigation, et nous montrons sa capacité à exécuter des tâches de navigation tout en maintenant la prosodie de mouvement souhaitée.

Notre troisième contribution est une étude en situation réelle de l'impact de la prosodie de mouvement sur la perception du robot par les participants, ainsi que leur comportement et leur évaluation des performances du robot. Le robot est déployé de manière autonome dans un bâtiment où il effectue une tâche de distribution de prospectus. Le robot utilise différents ensembles de contraintes prosodiques qui, selon notre modèle, sont associés à la perception d'attitude de *confiance* ou d'*hésitation*. Les résultats suggèrent que les variations du mouvement du robot n'ont pas eu le même effet sur la perception du robot que lors de nos expériences de perception. L'analyse des entretiens nous donne un aperçu des autres aspects du comportement et du mouvement du robot qui ont eu un impact sur les perceptions des participants, ce qui ouvre la voie à de futurs travaux visant à améliorer de

manière itérative la compréhension de la prosodie du mouvement ainsi que l'algorithme de navigation.

Contents

List of Figures

List of Tables

Introduction

1.1 Context

Mobile robots are being studied and developed as potential solutions to perform tasks in human populated environments, ranging from public spaces such as supermarkets and train stations as in Figure 1.1, to spaces with vulnerable people such as hospitals (Kivrak et al., 2020), care-homes, or individual homes (Coşar et al., 2020).

Mobile robots deployed during field tests tend to be met with mixed reactions from the humans with which they share the environment, who may accept or reject the robot through mechanisms which are not yet fully understood (Hebesberger et al., 2017; Mutlu & Forlizzi, 2008). On the one hand, the technical complexity and capabilities of a smart device and a robot are quite similar, so one may expect robots to be treated and perceived similarly to machines. On the other hand, some studies point to humans feeling some level of empathy towards robots (Menne & Schwab, 2018; Rosenthal-von der Pütten et al., 2014), and some soldiers have been burying their bomb disposal robots (Carpenter, 2013). At the same time, many researchers and companies seek to deploy mobile robots into human environments to accomplish useful tasks, but they often struggle to explain people's reactions to their robots' behavior. It seems that even when the robot simply navigates without any intention of the robot's designers for it to interact socially, people may interpret its actions in terms of attitudes or intentions. As discussed in a recent Social Navigation (SN) survey (Mavrogiannis et al., 2023), while progress has been made towards understanding some aspects of what makes a robot's navigation acceptable such as people's preference for robots that comply with personal space boundaries, we still lack a unified theory of social navigation. This lack may be due to the vastness of the domain and fragmented literature focusing on specific sub-problems and motion variables. This results in an inability to determine how a human's social perception and interpretation of a robot's actions is affected by different motion styles obtained through combinations of motion variables.

Social Navigation works tend to perform experiments where participants interact with a robot exhibiting a state of the art, complex robot behavior resulting from navigation al-

Figure 1.1: Left: Bossa Nova shelf scanning robot in a supermarket. Right: Hease robot in a train station.

gorithms combined with the rest of the robot's physical and software design (Carton et al., 2017; Gil et al., 2021; Kamezaki et al., 2019). Questionnaires and interviews allow researchers to perform an *evaluation* of the complex behavior in terms of its acceptability and ability to accomplish a specific task optimally (Mavrogiannis et al., 2019). This **top-down** approach enables the evaluation of a given algorithm in a given context, however it makes it difficult to determine which specific aspects of the robot's navigation were responsible for each aspect of the evaluation. Furthermore, most social navigation studies use comfort, naturalness, smoothness, perceived safety and, more recently, legibility (Kruse et al., 2013) as their social evaluation metrics, which is different from evaluating how people interpret robot motion as conveying social attitudes or intentions.

Other works in the field of Human-Robot Interaction (HRI) have explored other interaction modalities such as voice (McGinn & Torre, 2019), gestures (Augustine et al., 2020; Saldien et al., 2014; Zhou & Dragan, 2018), positioning (Brandl et al., 2016), as well as a few works on navigation parameters such as (Saerbeck & Bartneck, 2010). These studies have allowed researchers to determine that each of these modalities play a role in HRI, however most of them consider each modality separately, which makes it difficult to determine whether there are interactions between them when combined into a human's perception of a robot. Studies that consider *combinations* of modalities typically do not include navigation as one of them (Dautenhahn et al., 2009).

1.2 Research questions and methodology

It seems likely that people's social interpretations of robot motions as conveying intentions or attitudes may play a role in the acceptance and evaluation of the robot. Therefore, in order to properly integrate mobile robots into their various roles and tasks, we need to be able to control how they interact with people and how they are perceived. Current social navigation algorithms cannot be configured to generate motions which are perceived as conveying different social attitudes or intentions, and it is unclear which aspects of the robot's motion should be controlled to do so. Hence, in this thesis we address the following research question: **"How can we design a social navigation algorithm that enables the robot's motion to be adapted based on its impact on human social perception of the robot?"**. We approach this question by splitting it into three successive research goals.

When initiating the design of a social navigation algorithm, common approaches include adding an extra component to an existing navigation algorithm inspired by existing social sciences models of humans, such as avoiding personal space as defined by proxemics, or directly using machine learning approaches to imitate human navigation without modelling the underlying social mechanisms. **Our first research goal is to understand which aspects of the robot's motion are involved in shaping how people perceive social attitudes**. With these approaches, it is difficult to gain a principled understanding of the links between motion variables and social perceptions since the algorithms employed may limit which variables can be controlled, and to what extent they can be controlled. Furthermore, existing datasets of human navigation do not provide annotations of perceived social attitudes. **We propose to start by exploring the space of navigation variables**

independently from any existing algorithm. We start by defining a set of physical motion variables a priori, based on analogies to vocal prosody, which are combined to create a corpus of robot motions. Perception experiments conducted online (and in person once conditions allowed) enable us to establish how each of the variables relate to participant's social and physical perceptions of the robot, defining what we call "**movement prosody**".

Once we have determined which motion variables are relevant for altering a person's social perception of the robot, we use this knowledge to determine the requirements for our social navigation algorithm. Our second goal is to **propose a social navigation algorithm that explicitly provides control over the motion variables which were found to affect people's social perception of the robot**. Social navigation algorithms are generally built to generate a single, specific style of motion aiming to be comfortable, natural, smooth and legible while ensuring safety in dynamic environments. Expressive motion generation approaches focus on generating multiple styles of motion, usually to express emotion, however these approaches are either purely communicative, hard-coded motions and do not simultaneously accomplish a practical task, or the algorithms do not account for how to maintain a consistent style in dynamic environments. **We aim to develop an algorithm that can be configured to generate motions which accurately reproduce a specific movement prosody to elicit the corresponding perception of attitude, while operating safely in dynamic environments**. Studies on vocal prosody in human interaction show that small, subtle variations of the speech signal dynamics are responsible for altering perceptions of attitudes. Gesture and gaze dynamics have been shown to vary together with vocal prosody, which suggests that accurate and temporally consistent control over the robot's motion is also required. Usually, existing approaches incorporate mechanisms that bias the trajectory generation towards motions that exhibit the desired properties, while allowing trade-offs with the robot's practical task performance. In contrast, **we consider that the control over the robot's movement prosody should be given priority, even if it impacts task performance.**

Evaluating the navigation algorithm is essential to understand how the different types of movement prosody impact humans. Some social navigation algorithms are evaluated in simulations using metrics such as the number of collisions with pedestrians, infringement on personal spaces, as well as practical metrics such as path length. This does not allow to measure whether the algorithm successfully generated the desired social perception, and does not allow observation of possible reactions people may have to the robot. In-person experiments are also common, however they are often conducted in somewhat artificial lab conditions where participants are recruited in advance, and are aware that the goal of the study is to observe their motion. When compared with studies which involve tasks which are closer to real use cases, and pretext tasks to avoid participants suspecting the true goal of the study, different results have been observed. **We aim to evaluate the robot in an in-the-wild study, where participants spontaneously interact with the robot which is given a credible pretext task and role.** In order to capture participants' potentially subtle reactions, we require a method to capture synchronized data from the robot's onboard sensors, as well as externally placed sensors in the environment. We also require an interview design that is coherent with the pretext task in order to gather participants impressions of the robot without biasing their answers.

1.3 Contributions

Our first contribution is the construction of a model linking motion and appearance variables of our mobile robot to a set of perceptual scales using adjectives describing social attitudes and physical qualities. We propose a set of motion and appearance variables that cover the range of feasible motions and appearances of our mobile robot. The motion variables are derived from analogies with the vocal features known to be involved in voice prosody, and consist of robot acceleration and velocities, timing and sequences of accelerations and decelerations, as well as smooth, saccadic or incremental accelerations. The motion variables are systematically combined to form a corpus of motions which are performed on our mobile robot and recorded to create a video corpus[1]. The corpus is used in a series of three perception experiments where participants choose which adjective on either end of ten perceptual scales best describes their impression of the robot, for many combinations of robot motion variables. The results indicate that all of the corpus motion variables had significant effects on how people perceived the robot in terms of qualifiers such as aggressive or gentle, sturdy or frail, confident or hesitant. The corpus motions serve as a basis for our algorithm design.

Our second contribution is a local planning algorithm capable of generating robot motions that correspond to the desired human perception, while accomplishing a practical navigation task. We formulate the problem as a constrained trajectory optimization, where the robot should minimize distance to its goal while satisfying specially designed and highly restrictive constraints. These constraints force the motions to exhibit the desired movement prosody, in order to maintain a consistent social perception of the robot. The algorithm is designed to be able to maintain consistent prosody while dynamically re-planning its trajectory to deal with dynamic environments, and a prosody constraint hierarchy enables the robot to maintain partial satisfaction of the prosody constraints even in complex situations. The approach is implemented and validated on a real mobile robot, and shown to enable the robot to accurately reproduce the motion characteristics from our corpus.

Our final contribution is a human-robot interaction experiment where the robot interacts with naive participants in a university campus building. We implemented a complete software architecture around our proposed navigation algorithm, adding modules performing person detection and tracking, localization, and a high-level decision state machine to implement a flyer distribution robot. The experiment compares participant's reactions to the robot using our algorithm configured to convey a confident or hesitant attitude through its movement prosody. Through a semi-structured interview and observation of participants behaviour, we study whether the movement prosody employed by the robot while approaching a person can impact their behaviour, their rating of its performance, and how they perceive the robot. The statistical analysis of the results showed that although participants did not perceive the predicted hesitant motion as hesitant, there were some small differences in how participants behaved compared to the confident motion. Interestingly, participant responses during the interviews suggest that other aspects of the robot's behaviour may have had a greater effect on the hesitant/confident perception, suggesting

[1]Examples of the corpus videos: https://youtu.be/EiH8o1PjlOw. Full video corpus, and project page: https://osf.io/5csrg/

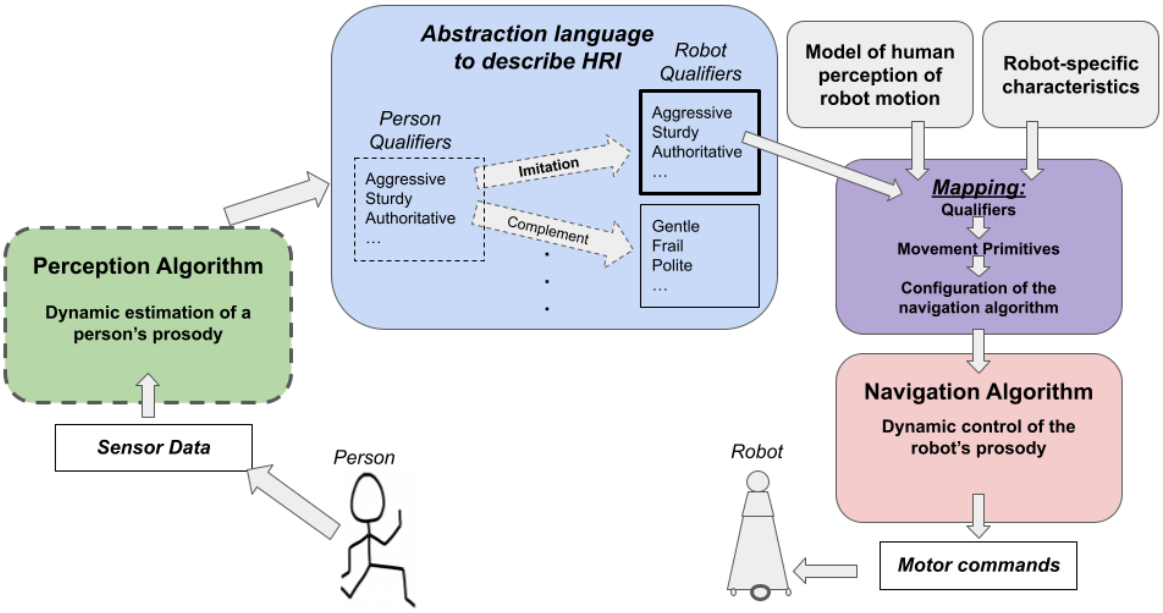


Figure 1.2: Proposed concept for a movement-prosody aware navigation architecture. Solid lines depict the parts of the architecture tackled in this PhD, dashed lines depict future work.

future directions for refining the model of movement prosody.

Together, these contributions make it possible for a robot designer or researcher to choose how the robot's motion should be generated based on the desired social and physical perception of the robot, and to understand how these perceptions may relate to people's physical reactions to the robot as well as their evaluation of its performance. While these contributions can be used as such, we also envision them as part of a broader project to work towards an architecture enabling co-adaptation between the robot and person's motion. This architecture concept is depicted in Figure 1.2. The purple and red boxes represent the first and second contributions of this thesis, i.e. the model allowing us to map from the adjectives used as qualifiers to describe the robot's movement prosody to physical motion variables, and the navigation algorithm which can be configured to produce motions which accurately follow the desired movement prosody in dynamic environments. In the envisioned architecture, rather than manually selecting the list of qualifiers which describe the impression generated by the robot, they would be determined based on some model describing how the robot should adapt to the person, for example by imitating the person's prosody, or adopting a complementary prosody (blue box). Naturally, this would require a perception algorithm to determine which type of prosody the person is displaying based on the robot's onboard sensor data (green box).

1.4 Outline

The thesis outline follows the order of the contributions.

- The second chapter presents relevant material from related works in the areas of social navigation algorithms, expressive motion generation algorithms, evaluation methods for robot motion, and aspects of human communication and interaction.

- The third chapter presents the robot platform and describes the construction of our robot motion corpus. It also presents the three perception experiments' design and results, thereby providing the design requirements for our algorithm. This chapter shares passages with our journal paper in Interaction Studies (Scales et al., 2023), which reported preliminary experiment results based on fewer participants. The thesis chapter presents more advanced statistical modelling of the effects of the corpus motion variables on participants' perceptions of the robot.

- The fourth chapter presents our approach to build the social navigation algorithm. It presents the general navigation problem formulation, followed by our approach to transform the fixed-length corpus motions into prosody constraints for the trajectory optimization. It then discusses how we ensure the algorithm can consistently apply the prosody constraints even while re-planning to account for dynamic environments, and lastly presents our use of a constraint hierarchy to ensure partial prosody compliance even in adverse navigation conditions.

- The fifth chapter begins by presenting our ROS implementation, and discussing important implementation details which render accurate control of the robot's motion feasible. It then presents a qualitative validation of our implementation, demonstrating our algorithm's ability to generate motions which comply with the prosody constraints in dynamic environments.

- The sixth chapter presents the ecological validation experiment. It discusses the reasoning behind the experiment design and the choice of the two robot attitudes to be tested. It presents the experimental results with respect to participants ratings of the robot's performance, how the confident and hesitant movement prosody settings were perceived, and how participants behaved depending on the prosody setting.

- The seventh and final chapter summarizes the contributions of the thesis, and presents directions for future works.

1.5 Publications

In this section we list our publications in chronological order, summarizing their relation to the thesis and their contributions.

Our first publication is a conference paper published at *ICRA 2020*, which presents a general navigation architecture in which the local planner's cost function can be altered at run-time to implement different navigation tasks. We use our algorithm to study how the parameterization of the robot's relative position with respect to the human impacts the human-robot interaction during a person-following task. In-person experiments were conducted in lab conditions and ecological conditions to compare the participant behaviour. Although there was no difference between robot positioning in the lab conditions, we found

that in ecological conditions participants interpreted the robot's attempt to position itself at their side as an error, as if the robot could no longer detect them. This work was conducted during the author's master's thesis. The algorithm presented in chapter 4 is a significant extension of the algorithm presented in the paper.

Scales, P., Aycard, O., & Aubergé, V. (2020). Studying navigation as a form of interaction: A design approach for social robot navigation methods. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 6965–6972. https://doi.org/10.1109/ICRA40945.2020.9197037

Our second publication is a conference paper which presents an algorithm dealing with global indoor localization of a mobile robot, which is often a prerequisite for HRI experiments such as those presented in chapter 6. We use images of the ceiling taken from the robot, since the ceiling features are mostly visible even when the robot is surrounded by people. We use convolutional neural networks to learn the correspondence between ceiling images and the robot's position and orientation. We obtained good localization performance with an average error of $0.21m$ and $0.1rad$, including in large open spaces where methods based on laser range finders tend to fail. This work was conducted as part of a short summer internship while the first two authors were completing their master's degree.

Scales, P., Rimel, M., & Aycard, O. (2021). Visual-based Global Localization from Ceiling Images using Convolutional Neural Networks. *16th International Conference on Computer Vision Theory and Applications*, 927–934. https://doi.org/10.5220/0010248409270934

Our third publication is a workshop paper presented at *Affective Robots for Wellbeing (ACII 2022)*. The paper presents early results from the perception experiments of chapter 3 which showed how the robot's motion could be interpreted as conveying social attitudes, as well as alter human's physical perceptions of it. We discuss the relevance of being able to influence a person's social and physical perception of the robot for applications in the field of care by discussing the concept of frail robots for tender care.

Scales, P., Aubergé, V., & Aycard, O. (2022). Socio-expressive robot navigation: How motion profiles can convey frailty and confidence. *2022 10th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, 1–8. https://doi.org/10.1109/ACIIW57231.2022.10086013

Our fourth publication is a long journal paper published in *Interaction Studies* which expands on the third publication. This paper gives an in-depth description our global methodology to study navigation variables from the ground-up by taking inspiration from the study of vocal prosody. It describes our concept of movement prosody, provides details about the motion corpus design and the perceptual scales, in addition to presenting the perception experiment results for the model linking motion variables to social and physical perceptions of the robot. Chapter 3 is largely based on this paper, with the addition of a new statistical analysis of the second online perception experiment with a larger participant pool.

Scales, P., Aubergé, V., & Aycard, O. (2023). From vocal prosody to movement prosody, from hri to understanding humans. *Interaction Studies*, *24*(1), 131–168. https://doi.org/https://doi.org/10.1075/is.22010.sca

1.6 Impact of covid-19 on the thesis methodology

In this section, we would like to point out the extent to which the covid-19 pandemic impacted the methodology of this thesis, particularly the work related to our first contribution of understanding and modeling movement prosody, which is the starting point for the whole thesis. Our original plan was to perform experiments following a similar methodology to our prior work conducted during the author's master's thesis, focused on in-person ecological experiments (the master's thesis methodology, algorithm, and experiments were published at ICRA 2020 (Scales et al., 2020)). We designed a large ecological experiment in a smart-home living lab in order to study how pairs of people interacted (either face to face or with one person using a teleoperated robot) while performing navigation tasks, according to their social relation. This would have allowed us to study whether motion patterns emerged, indicating which variables would be involved in social perception. The covid-19 pandemic began as we were finalising technical preparations after having spent several months on the experiment design, rendering all access to the living lab and robot impossible for months. A second, simpler in-person experiment was designed, but this experiment also had to be abandoned. Further details concerning these experiments are given in Appendix C. To avoid further setbacks, we proposed the online perception experiments presented in chapter 3. This is not simply a change of presentation modalities from in-person to online: it also affects what we are able to study. The person is not performing any kind of realistic task and they are not taking part in an embodied interaction with a robot or other person where their relation may evolve over time. Instead of studying a dynamic interaction process, we had to settle for studying one-sided social perception. This is why we spent a lot of time continuing to try to find a way to run an in-person study before switching to the online approach.

Chapter 2

State of the art

2.1 Introduction

This chapter begins by giving some background on relevant fields in the first three sections. The first section gives an overview of the problem of mobile robot navigation, and how the field of social navigation emerged in order to handle the presence of humans around the robot, remarking that the primary goal of social navigation is to work towards defining a natural, comfortable, legible navigation. In the second section, we first briefly discuss human interaction principles, highlighting that humans interpret non-verbal actions as conveying elements of intention or affect. We then discuss works in human robot interaction, discussing to what extent similar principles have been observed in modalities other than navigation. In the third section we present works in the field of expressive robot motion generation which seeks to explicitly induce certain perceptions of robots through their motion, pointing out that these works are aimed at expressing robot emotions or practical intentions rather than express socio-affects or attitudes that can be interpreted as being directed towards another person.

In the last three sections, we discuss specific aspects of how researchers approach the design and evaluation of social navigation and expressive motion algorithms. The fourth section discusses how researchers determine which aspects of the robot's motion they should control. The fifth section discusses how the algorithms perform the control of the motion according to the selected social or expressive features. The last section discusses which kinds of human perceptions researchers are usually interested in evaluating, and gives some background for our choice of evaluation method.

2.2 Background: from traditional to social mobile robot navigation

The problem of navigation can be summarized as determining how to drive a robot from its current position to some goal position, while avoiding collisions. Additionally, we aim to do this in a way that is efficient, by minimizing the path length, travel time, or energy. These objectives are those typically considered in traditional navigation approaches. When performing navigation in environments shared with humans, these objectives result in robot motion that people do not consider desirable, such as the robot passing very close to humans. In this section, we present an overview of traditional approaches to mobile robot navigation, and of how the problem and corresponding approaches have changed in order to handle deployment in environments which are shared with humans.

2.2.1 Traditional navigation

In this section, we give a quick overview of traditional approaches to solve the navigation problem, which are typically composed of two stages, global planning and local planning (Siciliano & Khatib, 2016). Global planning is a slower process that plans the entire path at a purely geometric level. Local planning uses the global path as guidance in order to

quickly compute a trajectory (consisting of path *and* timing) over a fixed time horizon in real-time that accounts for the current state of the environment based on sensor data, and that accurately reflects the robot's motion capabilities.

The global planning stage requires a representation of the robot's environment (e.g. a map), and computes a feasible path that connects the robot's initial position to the goal, typically making some simplifying assumptions on the robot's shape and motion characteristics in order to make the problem tractable. The output of such as step is usually a purely geometric path, defined as a sequence of robot configurations. One approach is to represent the static environment as an occupancy grid, over which A* (Hart et al., 1968) or other search-based algorithms can be run to find the shortest path. Such approaches are resolution-complete, and are easy to compute when the dimension of the search space is low. Another approach consists of sampling the configuration space, with algorithms such as Probabilistic Road Maps (PRM) (Kavraki et al., 1996) and Rapidly exploring Random Trees (RRT) (Karaman & Frazzoli, 2011), where a configuration (x, y, θ) is drawn at random, and checked for collision. If it is not in collision, it is added to the roadmap or tree, eventually finding a connection from start to goal, given enough samples. These were designed to avoid the exponential complexity of search as the number of dimensions grows, and they are probabilistically complete. In mobile robotics, global planning approaches are generally used to plan paths which only consider the robot's position and orientation, and assuming perfect knowledge of the environment. In practice, such paths may be infeasible due to the robot's kinematics, such as the differential constraint for non-holonomic mobile robots, or due to changes in the environment. Our interest is in studying and controlling the subtleties of mobile robot motion, rather than high-level path planning considerations, hence in this thesis we do not focus on global planning.

Local planning bridges the gap between the high-level global plan and the motor control commands sent to the mobile base. Local planners take sensor readings and the global path as input, and plan a trajectory that attempts to follow the global path, while avoiding obstacles and complying with the robot's kinematics constraints. The Dynamic Window Avoidance (DWA) (Fox et al., 1997) is a classical example, where the robot's dynamically feasible control input space is sampled in order to find a circular arc trajectory over some short (1 or 2 seconds) time horizon that is collision free, and minimizes a cost function driving the robot towards its local goal. Such early approaches to local planning typically only consider the immediate state of the robot, or a short prediction horizon. This tends to result in sub-optimal, short-sighted motion, which later approaches alleviate by leveraging more powerful computing hardware and more advanced algorithms in order to plan over longer horizons, such as in the Timed Elastic Band approach (Rosmann et al., 2015), where the online trajectory planning problem is formulated as an optimization problem, and solved within a Model Predictive Control (MPC) framework. Local planning methods which are tightly linked to the robot's low-level motor controller present themselves as a good way to explore and control the mobile robot's motion in order to study their impact on humans, hence this thesis aims at providing local planning algorithms with a way to modulate their trajectories according to how they will be perceived by humans.

2.2.2 Social Navigation

As robotics progressed, researchers began exploring applications where the robots are deployed around humans, and must navigate. This added two types of complications compared to traditional navigation applications and requirements.

Firstly, there were new challenges in terms of the complexity of the spatial and temporal problem of moving in such environments which can contain a large number of humans in motion, like when navigating through a crowd of people. The future motion of people is not known, and initial approaches with conservative estimates of the distribution of future motions tended to run into the Freezing Robot Problem (FRP) as described in (Trautman & Krause, 2010). They suggested that the missing element was to consider that people perform joint collision avoidance, rather than acting as individual agents, and proposed to model these interactions with interacting gaussian processes. Many works in the field of social navigation have a focus on this complex algorithmic problem of multi-agent cooperative navigation planning in real-time, with close ties to the field of human motion prediction.

Secondly, there were new so-called *social* challenges, given that treating humans as mere dynamic obstacles was found to result in negative effects on humans. Tasks that were seemingly *"solved"* by traditional navigation had to be reconsidered, as humans judged robot motions based on criteria other than their efficiency. For example, navigating through an environment could not be reduced to finding the shortest or fastest path from A to B, as researchers found that people preferred the robot to keep some distance away from them, leading to many works implementing a personal space around humans based on the concept of proxemics by Edward T. Hall (Hall et al., 1968). Another interesting aspect is the distinction between the actual safety of the robot navigation system and the *perceived* safety. Studies showed that certain robot motions could be perceived as unsafe, even though the roboticists knew the algorithm was safe (Pacchierotti et al., 2005), highlighting once again the necessity of considering human interpretations of the robot's actions. Other important aspects that have been considered include the legibility of the robot's motion, which impacts how accurately a person can infer where the robot will go in the near future based on its prior motion. Smoothness and naturalness of motion is also considered key, resulting in approaches that attempt to minimize jerk (the time derivative of acceleration), or more generally aim to replicate human motion characteristics.

Social navigation is often synonymous with the task of a robot navigating through a crowd, where its goal is not to engage in explicit interaction with people. However the definition of the term is still somewhat unclear, as discussed in a recent survey (Francis et al., 2023). Works which focus on other aspects of navigation such as approaching a person, joining a queue or group of people, following or guiding a person can also be considered as parts of social navigation, where the interaction with a person is part of the task. As pointed out in an early survey of social navigation works (Kruse et al., 2013), when the robot should navigate in a crowd, the main objective is to avoid discomfort for the surrounding humans, by making sure the navigation is perceived as comfortable, safe, natural and sociable (e.g. respects social norms such as staying on the right hand side of a corridor).

To summarize, the overall goal of social navigation research is to model the ideal or most acceptable form of navigation, and implement algorithms that are able to generate

such motions in complex dynamic, uncertain environments. Several factors are considered as contributing to the overall acceptability (comfort, naturalness, social norm compliance, legibility, perceived safety), however it is assumed that the desired outcome is to maximize all of these factors. Making an analogy to the vocal modality, we might say that this goal is similar to the goal of designing a speech synthesis system with a natural, clear, and maybe polite prosody, similar to smart assistants such as Amazon Echo. In our work, we instead aim to explore how subtle variations in motion may allow exploration of other dimensions of human perception of the robot, namely in terms of attitudes and physical characteristics. Instead of there being a single ideal style of motion, there may be different motion styles that are interpreted in various ways by humans, which are then considered as acceptable or not.

2.3 Background: communication and interaction

Simply because it is deployed in a human-populated environment, a mobile robot's actions and motion will be perceived, analysed, and judged by humans. In this section, we present an overview of studies of communication and interaction, first of all between humans, and secondly between humans and robots. We first discuss how humans form impressions of others through many different interaction modalities, with a particular focus on how vocal prosody has been shown to convey a person's attitudes towards others. We then discuss to what extent similar interactions have been studied in the field of human-robot interaction, and how the concept of subtle prosody could be extended to the motion and navigation modalities.

2.3.1 Human-human interaction

When interacting with each other, people perceive, interpret and generate behaviours across their whole body. Consciously or not, we use mechanisms such as eye contact (Argyle & Dean, 1965), body language (Scheflen & Scheflen, 1972), perception of interpersonal distance (White, 1975), proxemics (Hall et al., 1968), as well as vocal prosody (Gobl & Ní Chasaide, 2003) in order to communicate. Among these fields of study, proxemics has been widely used as inspiration in robot social navigation works since it describes how people maintain different distances with one another while interacting, depending on their culture and relationship. However, other aspects of locomotion and navigation such as the temporal dynamics of motion, and overall quality of body motion are not the focus of proxemics. We do not have models describing how a person's navigation dynamcis influences interpretations of their attitudes towards others.

Humans tend to interpret information across several modalities in a holistic manner, rather than treating each modality separately. People have been found to make strong associations between audio cues and shape cues (Magnani et al., 2017), as well as combining perceptual information from speech and lip movement when seeing and hearing a person speak (McGurk effect) (McGurk & MacDonald, 1976). It is therefore interesting to consider works which model the expression of attitudes in other communication modalities. The

vocal modality has received particular attention given that it's primary use is precisely to communicate (as opposed to gaze or navigation which are primarily of practical use). In speech, most of the communication channel's bandwidth is used in order to convey semantic meaning. But even when using exactly the same words, we can still modify *how* we say them by using the remaining bandwidth and degrees of freedom of the vocal signal such as changing pitch, rhythm, tone, or vocal effort. The variations of the *"how"* of speech are what constitute vocal prosody.

Studies have shown that vocal prosody variations could change the perceived affective state of the speaker (Schröder, 2003), and that different vocalizations of the same word could give it different meanings (Campbell, 2004) in an interaction. In addition to the aforementioned functions, vocal prosody can convey our *attitude towards other people* such as being polite or authoritative, aggressive or gentle, as well as give indications of our social role. Works by Aubergé and colleagues have studied how attitudes are expressed in vocal prosody (Shochi, 2008), as well as how people perceive them (Lu, 2015), finding that subtle variations in the dynamics of the voice are involved. The evolution of vocal prosody over time is hypothesized to be an essential part of how humans form relationships with one another creating a "socio-affective glu" (Sasa, 2018) between people, which captures the state of their relation.

In summary, humans use several modalities when interacting with others in order to communicate social and relational information. Although we do not have models linking kinematic and navigation parameters to the perception of attitudes, vocal prosody has received particular attention with studies showing that fairly subtle variations of the speech signal over time can convey one's attitude towards another person, and subsequently influence the interaction and their relation. In the following section we explore whether similar observations can be made within HRI, presenting studies which suggest that the notion of prosody could be extended to the whole body, not just voice.

2.3.2 Human-robot interaction

When people interact with robots, they tend to interpret the robot's actions as they would those of a social agent. This is reminiscent of the Computers Are Social Actors (CASA) paradigm (Reeves & Nass, 1996) which suggests that people treat computers and new media like social agents. Since the early days of deploying robots in human environments, researchers have explored whether or not similar verbal and non-verbal communication could be effectively employed on robots, leveraging human's tendency to already interpret robot actions as communicative. Studies have also been conducted on empathy towards robots (Rosenthal-von der Pütten et al., 2014). Many aspects of human interaction have been studied, and found to be important in human-robot interaction. Some examples of interaction modalities include intention (Sciutti et al., 2015), gaze (Mumm & Mutlu, 2011), engagement (Vaufreydaz et al., 2016), robot initiative (Munzer et al., 2017), reaction to touch stimuli (Shiomi et al., 2018). Typically, researchers motivate the incorporation of such communication as being beneficial for efficient human-robot collaboration, making the robot more interesting to interact with, more likeable. Researchers have also explored robot simulation of emotions as a tool to convey the robot's internal state and aid interaction.

Prior works of Aubergé and colleagues have extended the exploration of vocal prosody and the concept of socio-affective glu to human-robot interaction (Sasa & Aubergé, 2017; Tsvetanova et al., 2017), studying how vocal prosody can impact how the human behaves and interacts with a robot. In these works, robots are used as a means to explore what kind of impact robot usage of vocal prosody could have on a person's own vocal prosody, as well as on their overall behaviour during interactions with the robot. In addition to studying vocal prosody, (Sasa & Aubergé, 2017) observed that the changes in participant's vocal prosody over the course of an hour-long interaction with the robot were aligned with changes in their spatial behaviour, gaze, and voice quality (Tsvetanova et al., 2017). In another study (Girard-Rivier et al., 2016), participants were required to perform gestures directed towards a small mobile robot, and again their gestures were changed over the course of the interaction in a way that was consistent with the gradually more *glu-ing* vocal prosody emitted by the robot. In both works, participants are asked to self-annotate their interaction data after the experiment, and tend to use vocabulary to describe the robot that progressively moves from descriptions of an object to descriptions of an agent, or a subject, hinting again at the ties to the evolution of how the person perceives their relation with the robot. These works suggest that there could be a holistic prosody of the whole body, rather than only vocal prosody, which is what leads us to study what we call "prosody of movement" in mobile robots. In line with these prior works on vocal prosody, we are interested in determining how subtle variations in movement can be perceived as the robot having different attitudes towards people, as well as how they may be perceived as expressing different physical qualities of the robot, for example the expression of frailty or sturdiness.

Prosody has also been explored in HRI through studies on emotional music prosody (Savery et al., 2021) to generate emotionally expressive non-verbal audio, as well as gestures in (Savery et al., 2019). The influence on perceptions of a robot using voice and hand-over gestures with varying prosody was also studied in (Di Cesare et al., 2017), indicating that this concept may carry over to other modalities. Therefore, in addition to navigation factors, we also consider other dimensions that intervene in the perceptual experience of being near a mobile robot by studying visual appearance factors (presence and shape of eyes, head position, and stability of the robot base) as well as auditive factors through the presence or absence of motor noise. In a recent study parallel to our work (Lastrico et al., 2022), the authors studied how a robot arm's style of motion influenced a human participant's own motion. The robot picked up a glass in a "careful" or "not careful" manner, and performed a hand-over motion to give it to the participant. Participants were able to recognize whether the robot was being careful, and their own reaching motion mimicked the features of the robot's, i.e. slower and longer motions when the robot was careful. Once again this supports the idea that subtle motion variations of robot can be interpreted in terms of socio-affects.

In summary, human interaction takes place through multi-modal communication channels, within which subtle variations can express different attitudes towards others. When interacting with robots, humans consider similar factors, and perceive robots as social entities. Previous works have demonstrated that people's verbal and non-verbal behaviour with respect to a robot can be altered by the robot's own behaviour, more specifically its vocal prosody. In addition, these changes have been related back to participants' impression of their relation with the robot. While there have been HRI studies on proxemics and some basic movement dimensions with mobile robots, for the most part they do not study

how exactly different features of motion are tied to perceptions of attitudes, nor to how people consider their relation to the robot.

2.4 Background: expressive motion generation

In this section, we provide an overview of the area of expressive motion generation for robots. This field spans over various robot embodiments, and is not limited to mobile robots. In fact, its origins lie in computer graphics and animation rather than robotics, where the goal is to synthesize motion in different styles that can be interpreted in terms of emotions or intentions. For example, a character or robot could move in a way which is perceived as happy or sad. We first discuss works where the only reason for the robot to move is in order to express something. Second, we discuss recent works in expressive robot motion which aim to generate motions that accomplish a given practical task while also being expressive.

2.4.1 Expressivity as the only goal

Some works in robotics aim to use movement to convey emotions, affect, or style. A large part of such works are more focused on gestures, postures (Beck et al., 2012), or facial expressions, but some studies have been performed with robot motion. In many works, the sole purpose of the expressive motion is to convey the emotion (Chan et al., 2021; Mizoguchi et al., 1997; Song & Yamada, 2018; Yoshioka et al., 2015). Often, the display of emotions is motivated as a means for the robot to convey its internal state to the human, which has been shown to improve performance in human-robot collaboration settings (Breazeal et al., 2005). Other than emotions, people have also studied how to express practical, goal-directed intentions like where the robot will go (Szafir et al., 2014), as well as expression of affect (Saerbeck & Bartneck, 2010). In general, these motions are designed offline, and played back when the desired expression should be generated.

Our work's general goal is somewhat similar in the sense that we explore how movement can elicit various interpretations from humans, rather than simply being "natural". However, we are interested in combining this expressivity with useful, task-oriented robot navigation in dynamic environments, rather than fixed, hard-coded motions.

2.4.2 Functional expressive motion

In this section, we discuss a more recent development where instead of expressive motions being generated with the sole task of communicating an emotion or an intent, they are generated in order to accomplish a practical task on a robot simultaneously. For further reading, we refer the readers to a recent review (Venture & Kulić, 2019).

Within such works, an additional distinction can be made. In some works, the researchers study trajectories which are generated offline, or designed specifically in order to be evaluated in their study. In (Van Otterdijk et al., 2021), short trajectories were hand-

designed for a Pepper mobile robot, and combined with light displays and postures to convey sad, neutral or happy emotions while approaching a person to deliver a message. In (Vannucci et al., 2019), the authors explored whether a humanoid robot handing over an object could convey an aggressive or gentle attitude through its arm motion, or vocal prosody. The arm motion was designed by using motion capture to record a trained actor performing the handover action, and then re-mapping to the iCub robot and editing the motion to ensure it seemed to conserve similar characteristics to the acted motion. Participants were able to interpret the two styles when watching videos, as well as in an in-person interaction. These studies help to explore human perception of robot motion, but it is usually unclear which part of the motion captures the expressivity, and which part is purely tied to the task. It is unclear how one would design an algorithm to generate motions across different instances of the same tasks, or across different tasks.

More recently, researchers have started to explore ways of incorporating expressivity into algorithms which can plan and execution motions for arbitrary tasks. In some works, the features which make the trajectory expressive are explicitly known, such as (Hagane & Venture, 2022; Zhou & Dragan, 2018) for robot arms, and (Knight, 2016) for a mobile robot. In (Zhou & Dragan, 2018), an actress is asked to perform pick and place-like actions in a sad, happy, or hesitant style. The authors select features which seem to differ between the motions, and transfer them to a robot arm: average horizontal distance from end effector to base, average end effector height, average angle between vertical axis and end effector orientation, and speed along the path. Using humans performing expressive motions as an example is an interesting approach, however there are issues relating to the unclear mapping between human morphology and the robot's actuators. There may also be issues as to whether or not acted motions display identical prosody to spontaneous motions. In order to integrate these characteristics of motion into the robot arm's motion generation, the authors formulate the trajectory generation as an optimization problem where a function that captures the cost of a given trajectory should be minimized. The cost function is composed of terms relating to the robot's practical motion task (bring the end-effector of the arm to a given position in space) as well as terms expressing the style of motion, as described above. A learning based approach is used in (Sripathy et al., 2022) to train a generative deep neural network to generate trajectories for a bipedal mobile robot corresponding to emotions defined in the PAD (Pleasure Arousal Dominance) space. Due to the deep learning approach, the mapping between the desired expressivity and trajectory features which are manipulated is not explicitly available, and in order to gain insight into which specific features cause a trajectory to be perceived in a given way, further analysis is required. With such a learning-based approach, we do not directly obtain an understanding of which motion features map to which expressive dimensions.

These works are interesting because their goal is to build algorithms that can be configured differently to generate motions with different features, based on which emotion or affect should be expressed. However, only a few recent works are exploring the intersection of functional and expressive motion, and few of them are applied to mobile robots. Furthermore, our goal is to generate motions that express *attitudes towards other people*, as opposed to emotions which only describe a single person's internal state rather than the relation. In addition, these algorithms are usually only evaluated in static environments where dynamic re-planning of the motion is not necessary. Therefore, it remains unclear whether

deploying these approaches in the context of navigation in dynamic human environments would result in accurate representation of the desired expressivity.

2.5 Determining motion features to be explored

In this section, we discuss how researchers determine which characteristics are important in social navigation, and therefore which kind of motion the robot should perform.

In some social navigation approaches researchers use learning based techniques, either to mimic human navigation (Luber et al., 2012) or to learn strategies for navigation in crowded environments (Chen et al., 2017; Xu et al., 2023). In such cases, the motion features are not explicitly available. More importantly, the assumption when fitting algorithms to datasets is that the goal is to generate motions that represent the average human. In our case, we want to generate motions that convey certain attitudes, for which we would require labelling of attitudes in the datasets, making these approaches unsuitable for us.

The first method is to develop a full Social Navigation algorithm, either based on machine-learning methods which aim to imitate human navigation (Chen et al., 2017; Ramirez et al., 2016), or by manually observing and modeling human behaviour such as in (Kitagawa et al., 2021). Another approach is to implement existing models of human behaviour such as the Social Force Model (Shiomi et al., 2014). Spatial and proximity factors are the most commonly addressed in earlier works (Rios-Martinez et al., 2015), often being derived from the concept of proxemics (Hall et al., 1968). The algorithm can then be used to control a real or simulated robot, in order to conduct experiments. In (Honour et al., 2021) participants viewed top-down animations of robot trajectories generated via a Socially Aware Navigation (SAN) planner and a traditional planner, in order to compare them. This methodology enables the evaluation of one algorithm, or comparisons between algorithms, but it becomes difficult to determine how each aspect of the trajectories impacts the overall HRI and perception of the robot by humans. In addition, there aren't existing models of how humans perceive attitudes based on the whole-body locomotion. There are some works on posture, and some works on static distances with proxemics, but there aren't any models that can give us a relationship between the shape or velocity along a trajectory and perceptions of attitudes towards people.

In (Mavrogiannis et al., 2019), participants performed a navigation task while sharing the workspace with a mobile robot using one of three navigation methods (two algorithms, one teleoperated). The authors computed metrics on the robot's trajectories: average acceleration, average energy (defined as the integral of the squared velocity), minimum robot-human distance, path irregularity, efficiency, and topological complexity. Describing the navigation resulting from applying each algorithm may help to understand which variables are important, and what their impact is, but these metrics are relatively global values describing an average measure of the trajectory as a whole. When considering the inverse problem of how to generate a motion that induces a specific perception of the robot, this becomes an issue since the metrics might not uniquely control all of the robot's degrees of freedom.

Other works such as (Sorrentino et al., 2021) take an existing algorithm and alter some of its controllable variables. In this work, participants walked across a room, passing by a robot moving in the opposite direction. The robot avoided the collision using three different minimum obstacle distances and maximal velocities. This facilitates the understanding of the impact of a given parameter which varies systematically within this experiment, but comparisons with other algorithms will still be difficult if there are interaction effects between the controllable variables and non-controllable variables. When generated by an existing algorithm, can be hard if the algorithm approach in and as of itself already entails certain trajectory characteristics, for example with a longer or shorter planning horizon, different obstacle avoidance could be generated.

Lastly, some works opt to employ hand-crafted trajectories based on a small number of variables, such as curvature and acceleration (Saerbeck & Bartneck, 2010) or acceleration styles (Schulz et al., 2020), tested in basic navigation scenarios. This approach tends to provide a clearer idea of the impact of a given navigation variable, since only the variables of interest are directly manipulated. However each study only deals with one or two variables, once again lacking the power to thoroughly explore interactions between variables.

We propose to first assess which motion variables are important by using hand-crafted motions built through a systematic combination of the different variable values. To propose the set of variables to be studied, we make analogies to the variables known to impact voice prosody dynamics which are known to impact social interaction (Campbell & Mokhtari, 2003; Gobl & Ní Chasaide, 2003; Sasa & Aubergé, 2017; Tsvetanova et al., 2017). The selection and range of the variables reflect our robot's mechanical constraints and capabilities. Using systematically designed motions aids the understanding of the dimensions at play by using them in perception experiments. Among the initially proposed variables, only those that are found to have an impact on the person's perception of the robot will be kept, and used to guide the design of our social navigation algorithm.

2.6 Integrating social and expressive features into robot motion algorithms

In this section, we discuss in more detail how works in the fields of functional expressive motion generation and social navigation integrate the various trajectory features into the generation process. We also point out similarities in the choices of algorithmic frameworks between the two fields.

The goal of both of these fields is to generate motion that accomplishes a physical task, while taking into account a variety of metrics which account for human presence in the environment, as opposed to traditional navigation which aims to minimize time, or path length. This is reflected by recent works from social navigation and functional expressive motion generation turning towards similar methods by encoding the desired trajectory features into cost functions and/or constraints, followed by a search or optimization algorithm to generate trajectories which best match the social or expressive features. These objectives are often in conflict with each-other. The typical approach is to adopt a scalarization ap-

proach to treat the multi-objective problem as a single-objective problem, usually by computing a weighted average of cost terms (Mavrogiannis et al., 2023). This requires tuning the weights to obtain the desired behaviour, balancing the different social and expressive objectives, as well as the task objectives such as making progress towards a navigation goal.

In (Sisbot et al., 2007) the authors propose two costs, modelling visibility of the robot by the human, and a proxemics-inspired personal space. They propose either a weighted average, or taking the maximum out of the two costs. This choice depends on the task and balance between criteria, and the weights should be tuned according to the properties of the task. In a more recent work, (Khambhaita & Alami, 2020) present an approach which jointly plans cooperative trajectories for a single human and the robot, accounting for metrics such as the expected time to collision with the person, modulating robot velocity when near the person, and legibility of the trajectory. Other aspects have been modeled such as preferring deceleration rather than changing path shape to negotiate crossing a person (Kruse et al., 2012), maintaining a desired position and velocity while accompanying a person (Repiso et al., 2017), avoiding intrusion into group formations and the information processing space in front of people (Rios-Martinez et al., 2012). Similarly for expressive motion, (Zhou & Dragan, 2018) uses a weighted sum of costs, however they also explore the use of learned weights based on participant perception of emotion in the robot arm's trajectories, avoiding the manual tuning process. Some features are shared across several works, the most common being personal space around people derived from proxemics (Rios-Martinez et al., 2015), however most works consist precisely of proposing their own novel cost or constraint, leading to each work using different subsets of cost terms.

While trade-offs between traditional task performance metrics and social or expressive features are inevitable, the issue with these approaches is that there is limited control over *how* the trade-off is performed. In some works, this trade-off is enforced more explicitly, such as (Hagane & Venture, 2022) where the expressive features for a robot arm can only be expressed through degrees of freedom which have absolutely no effect on the practical task. On the contrary, in (Park, 2016), the authors first develop a smooth parameterized control law for their autonomous wheelchair such that it produces *graceful* motion. Their trajectory planner optimizes over the parameter space defined by the control law, thus enforcing a given style of motion, regardless of the impact on task performance.

Formulating the problem as a trajectory optimization provides a general framework consisting of cost functions and constraints that can be combined to model complex navigation styles. Given the flexibility of the method, we also adopt the trajectory optimization problem formulation. It is crucial that our algorithm generates motions which are very accurately matched to those which we will use to construct our model of human perception of robot motion. For this reason rather than modeling the desired motion through the cost function, which would make the desired prosody features subject to trade-offs with other cost terms modeling the functional task to be achieved, we propose to design specific prosody constraints to enforce the desired properties of motion. In this sense our approach is inspired by (Park, 2016), since we also restrict the valid trajectory space a-priori according to the desired style of motion. The constraints must take into account the consistency of the robot's motion style over time, requiring them to factor in the robot's past motion, as well as it's ability to plan a future trajectory with appropriate movement prosody.

2.7 Evaluation of robot motion

Social Navigation studies that deal with evaluating user's perceptions of robots employ a variety of methods, one of which is to use established questionnaires such as the Godspeed Questionnaire Series (GQS) (Bartneck et al., 2009) in (Mavrogiannis et al., 2019; Sorrentino et al., 2021), Negative Attitudes towards Robots Scale (NARS) (Nomura et al., 2006), Perceived Social Intelligence scale (PSI) (Barchard et al., 2020) in (Honour et al., 2021), or Robot Social Attributes Scale (RoSAS) (Carpinella et al., 2017). The items on such scales are often derived from existing theories and paradigms in social sciences.

In contrast, we base our selection of items on adjectives originating from previous studies investigating human vocal and gesture prosody generated during interactions with a robot (Guillaume et al., 2015; Sasa & Aubergé, 2016). The adjectives used in our scales are derived from the participant's self-annotations of their own interaction data. Each scale opposes two adjectives, some related to the physical impression of the motion, others related to perceptions of intentions or attitudes. The adjectives represent vernacular terms that a person may use in their everyday life, as opposed to terms derived from a scientific theory with a specific interpretation within its field. In a sense, these adjectives on the perceptual scales are the tools we give the participants in order for them to be able to describe the impression they have of the robot. This is inspired by the impressionistic paradigm, which has been used in prior works to study associations across modalities such as between sounds and shapes in the "kiki, bouba" experiment (Drumm, 2012), or between vocal prosody and colors in one of the series of works by Sagisaka et al. (Watanabe et al., 2014).

Some SN works focus on the efficiency of the method (often applied to navigating through dense crowds) rather than its impact on people's perceptions of robots (see (Mavrogiannis et al., 2023) for a recent survey). Other works evaluate the impact of their navigation algorithms using concepts such as acceptability, naturalness, comfort, likability or human-likeness (Kruse et al., 2013). These concepts are important in HRI, however they are general concepts that we believe may depend on more specific perceptions such as those we propose to study. Instead of a given style of navigation being inherently acceptable or unacceptable, we explore how a style induces perceptions of robot attitudes which may explain subsequent judgments on whether the navigation is acceptable within a given task and human-robot relation.

2.8 Summary and motivation for our approach

In this section, we summarise the various aspects discussed in the state of the art, and give our motivation for our overall approach and contributions. The general approach to design social navigation algorithms is to modify an existing navigation algorithm so that it improves people's evaluation of comfort, naturalness, legibility, and perceived safety. The assumption seems to be that there exists a single ideal navigation style that would maximize all of these criteria. Studies in HRI suggest that people often consider robots as subjects or agents, interpreting their actions in social terms by attributing intentions and attitudes to them. Prior studies have shown that subtle variations in vocal prosody can convey different

attitudes. Similar effects have been observed in a HRI context where a robot's vocal prosody altered the person's own behaviour and prosody, as well as their impression of the robot. Studies have also shown that human perception and production of social communicative cues is holistic, and that the dynamics of gestures evolves in a similar manner to vocal prosody during an interaction. These elements lead us to hypothesize that subtle variations in the robot's navigation might also be interpreted as conveying social attitudes towards the surrounding humans, which could in turn impact human's evaluation of the robot.

Unlike vocal prosody, we do not have existing models relating motion variables to perceptions of attitudes. Therefore, we cannot apply the typical approach of transferring knowledge from known human interaction concepts. This motivates our first contribution, which is to build a model relating robot motion variables to perceptions of attitudes. Navigation's impact on people is usually evaluated after the algorithm is developed, which may make it difficult to precisely understand which variables cause which perception. In contrast, we first study motion variables independently of any existing algorithm through perception experiments discussed in Chapter 3.

Only once we have established which variables are important in movement prosody do we turn towards constructing an algorithm that can control them. A common method for social navigation and expressive motion algorithms consists of formulating trajectory generation as a constrained optimization problem. Social or expressive aspects are typically encoded as terms in the cost function, alongside terms related to the robot's practical motion task, which can lead to trade-offs between the terms. Even if these trade-offs only result in small variations of the style of robot motion, studies on vocal and gesture prosody suggest that it is precisely in the small details that social and relational information is carried. We propose to prioritize the trajectory's compliance with the desired movement prosody over the objective navigation task performance so that the human perception of the robot stays consistent, including when operating in dynamic environments, as discussed in Chapter 4.

Many evaluations of social navigation algorithms are performed in lab studies, where participants are aware of the goal of the experiment, and placed in environments which do not resemble realistic deployment conditions. This can lead to different participant behaviour and impressions of the robot when compared with more ecological experimental conditions. We integrate our navigation algorithm into a complete architecture in order to run an in the wild study, where participants spontaneously interact with the mobile robot which is given a realistic pretext task and role. We combine observational measures of participants' reactions with semi-structured interviews to determine what impact different movement prosody had on their behaviour as well as impressions and performance rating of the robot. This experiment is presented in Chapter 6.

Chapter 3

Modelling human perception of socio-affects and attitudes in mobile robot locomotion

3.1 Introduction

Our first research goal is to generate a formal understanding of which properties of a mobile robot's navigation may be responsible for eliciting different social perceptions of the robot by humans. Rather than firstly implementing a full social navigation algorithm and only subsequently evaluating it, we propose to take a bottom-up approach by methodically constructing a corpus of robot motions to be used in perception experiments. In addition to motion variables, we also incorporate other aspects of the robot which change the person's perceptual experience of the robot when it navigates near them, motivated by the holistic nature of human perception and production of social cues. Once we have formalized the links between the physical characteristics of motion and appearance and human social perception, we may use this information to better understand how humans perceive robots,

In order to study the impact of the movement dynamics, appearance, and body motion of the robot on people's perceptions of it, we propose to design a robot motion corpus. The corpus consists of robot motions representing broad classes of motion and appearance characteristics, along with videos of the robot performing the motions. The motion and appearance parameters we include are velocities, accelerations, types of movements, head movements, chassis types, eye shapes as well as the presence or absence of motor noise. By designing a corpus of robot motion and appearance and conducting a perception experiment, we hope to help fill a gap in the literature in Social Navigation and Human-Robot Interaction research since, to the best of our knowledge, this kind of corpus and experiment aimed at holistically studying the impact of both low-level motion parameters *and* related visual and audio cues on HRI has yet to be published.

Designing a corpus of reference motions can help to avoid a common pitfall in Social Navigation which is the dependence on a specific robot platform or navigation algorithm, which makes comparisons between works difficult. To the best of our knowledge, this corpus is the first of its kind. Other researchers may choose to implement the same motions on their own robotic platform, which could help to further study the influence of different robot platforms on HRI. Furthermore, the video corpus allows researchers to conduct studies using exactly the same stimuli, which could help reproducibility of HRI studies (Irfan et al., 2018), and provide insights into the degree of cultural differences in HRI by running studies with participants from different countries. Our video corpus represents a wide range of robot motions, as well as various robot characteristics and motions which are not directly related to navigation. This allows us to study the relations between various perceptual stimuli and also to avoid ceiling effects, where one aspect of the robot's motion or appearance may dominate or nullify the effects of other aspects.

We believe the social navigation and HRI communities would benefit from holistically considering the appearance of the robot alongside its navigation style. By appearance, we refer to the robot's general aspect (mechanical, bio-evoking, human, animal, cultural references), as well as elements such as its size, color, texture, and structural appearance (sturdy vs. frail). In our motion corpus, we focus on the navigation variables for a given robot (RobAIR "RobAIR mobile robot, designed and built by FabMASTIC, Grenoble", 2021), on which we also have the ability to vary the frail-sturdy appearance dimension, the shape of its eyes, and orientation of its head. Therefore, these appearance variables are also included

in the corpus and combined with the navigation variables.

The structure of the chapter is as follows:

In section 3.2 we present the mobile robot platform which we use in this thesis. We give an overview of its physical design, hardware and sensors, and most crucially, its movement capabilities.

In section 3.3 we present our motion corpus construction which involves combining different values of variables affecting the motion and appearance of the robot. We start by presenting each of the motion variables and describing how each of them contributes to defining the robot's velocity profile. Then, we present the appearance variables and motivate their inclusion in the corpus.

In section 3.4 we discuss the creation of a video corpus, consisting of videos of the robot performing a simple straight line trajectory while subject to the different combination of corpus variable values. First, we discuss the relevance of online video studies in human robot interaction, pointing out the risk of losing some of the visual information depending on what kind of motions are filmed, as well as how the filming is performed. We then detail our process for ensuring that the videos capture the robot's subtle differences in motion and appearance as faithfully as possible.

In section 3.5 we present three perception experiments conducted using our corpus in order to determine the relationship between the corpus variables and people's perception of social attitudes or socio-affects of the robot, as well as physical qualities of the robot. We begin by presenting the perceptual scales which are used to evaluate participants' perceptions. Then, each experiment is presented, along with statistical analyses and interpretation of the results allowing us to establish which corpus variables had an impact on social and physical perception of the robot, as well as which perceptual scales were impacted. These results constitute a first model of the proposed concept of movement prosody, defining how the manner in which a robot performs a navigation task can convey social and relational meaning.

In section 3.6 we discuss the implications of the experiment results before concluding in section 3.7 by establishing which physical characteristics of the robot's velocity profile our navigation algorithm should be able to control in order to alter how it will be perceived by humans, leading into the next chapter.

The work presented in this chapter has been published as a journal paper in *Interaction Studies* (Scales et al., 2023), with a less detailed preliminary analysis of the results of the second online experiment.

The video corpus consisting of the 450 video files used in our perception experiments is made publicly available[1] so that other researchers may use it in their own studies. It was also the object of an APP deposit (French *Agence pour la Protection des Programmes*), under the name *BotEmoMove Database*[2].

[1]Examples of the corpus videos can be found at the following link: https://youtu.be/EiH8o1PjlOw. The full video corpus can be downloaded on the project page: https://osf.io/5csrg/

[2]Base de données BotEmoMove, APP#IDDN.FR.001.400027.000.S.A.2022.000.42000

In order to conduct the online experiments, we used a website developed internally at our university as part of previous students' work (Clarisse Bayol and Ambre Davat). This allowed us to have total control over how the stimuli were presented. In the context of this thesis, the author adapted the website to support video playback, as well as allowing the main study questions to be loaded from *csv* files so that non-programmers can easily change the questions without needing to modify code. This version of the online test tool was also deposited at the APP [3].

3.2 Robot platform

In this section, we present the RobAIR wheeled mobile robot platform which is used throughout the thesis, shown in Figure 3.1. The RobAIR platform ("RobAIR mobile robot, designed and built by FabMASTIC, Grenoble", 2021) is developed by the *FabMASTIC* fab lab at the Université Grenoble Alpes, where it serves both as a platform for teaching robotics, student projects, as well as for research.

3.2.1 Construction

The robot is $1.20m$ high, and has a diameter of $0.50m$ at its widest point, at the base. The body is constructed out of hard plastic flower-pot covers, providing the tapered shape to the body. The base deck is made out of plexiglass, and the whole structure is rigid. In the context of this thesis, the caster wheel mounting was slightly modified so that when tightened there would be absolutely no sway or give in the robot structure, making the base very stable. The mounting also enables to restore variable amounts of sway by loosening the caster wheels, making the robot base unstable. On its head, the robot has programmable LED light strips which can be programmed to display different eye shapes. Both of these features were used in the design of our motion and appearance corpus.

3.2.2 Hardware and sensors

The robot is equipped with two Hokuyo URG-04LX-UG01 2D Laser Range Finders (LRF) mounted horizontally, facing forward, with a range of $5.6m$ in ideal conditions and a 240 field of view. One sensor is mounted at ankle-height $0.10m$ above the ground and positioned $0.35m$ along the robot's forward axis, and another is mounted on the robot "head" at $1.20m$, centered. These sensors are used to detect obstacles as well as for detecting people based on the detection of legs and chests. The sensor acquisition frequency is of $10hz$, and the scanning time is $100ms$. The motor kit is a devantech RD03 24v, with an MD49 motor driver board. The MD49 is used to control the voltage sent to the EMG49 motors, and read the encoder counts to provide odometry information to estimate the robot's motion.

In addition to the LRFs used for autonomous navigation, the robot has a wide-angle fish-

[3] Online perception experiment tool: APP#IDDN.FR.001.400026.000.S.A.2022.000.42000

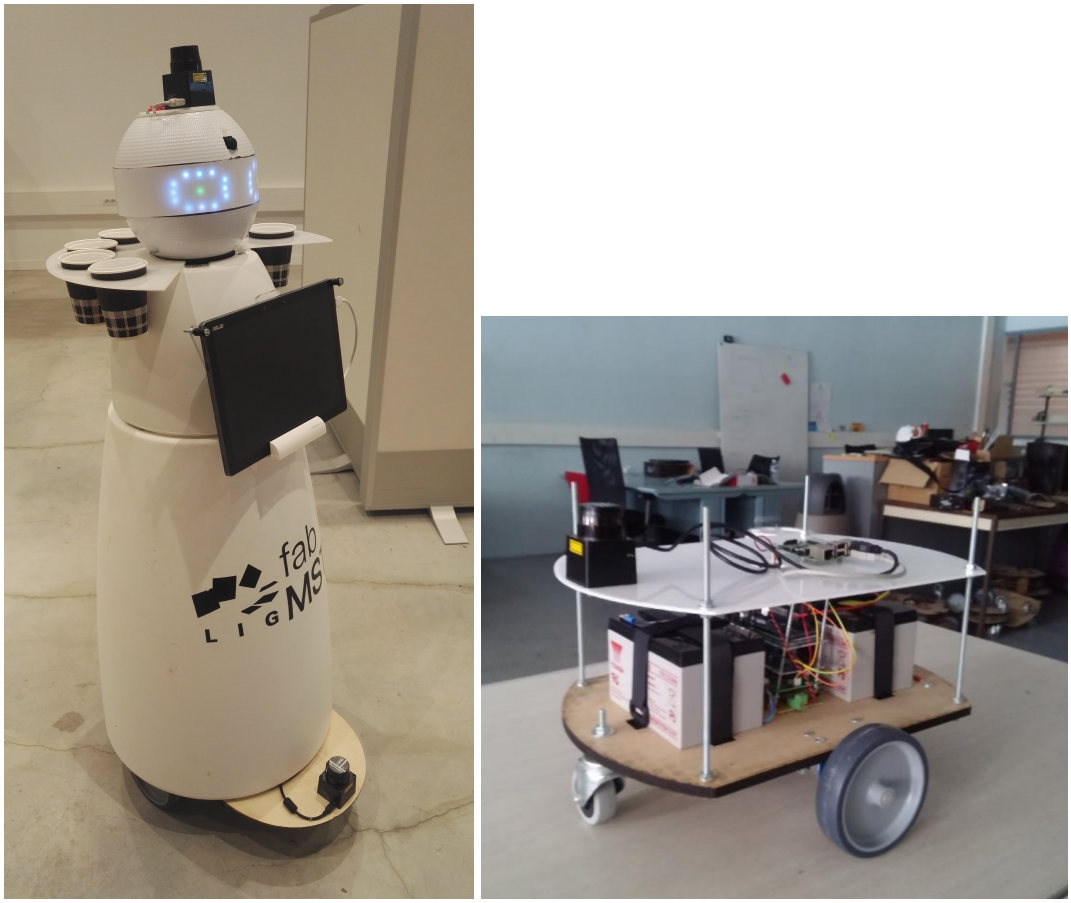


Figure 3.1: Left: RobAIR mobile robot. Right: RobAIR base.

eye camera positioned on its "forehead". Other sensors have been added in order to acquire higher quality and different types of data during human-robot interaction experiments. Two high quality Steinberg microphones are installed into the "head", allowing binaural capture of high-quality audio. In the context of this thesis, an Intel Realsense D415 color and depth camera was added to the head in order to capture higher quality images of the person's face and upper body when the person is near, or whole-body when they are far.

3.2.3 Motion capabilities

The robot has a differential drive configuration, with two driving wheels positioned at the center of the base, and two non-driven caster wheels at the front and back to keep the robot upright. This imposes constraints on the robot's motion, since it cannot instantaneously translate in a direction other than the one it is facing (so-called non-holonomic constraint (Siciliano & Khatib, 2016)). The EMG49 motors allow the robot to translate at a maximum velocity of $0.8m.s^{-1}$, and accelerate at $2.667m.s^{-2}$ in the highest acceleration setting, meaning it can reach its top speed in only $0.3s$. The MD49 only allows velocity control of the motors, and provides various settings to limit the robot's maximal acceleration. In the context of this thesis, the acceleration limit was set to the highest possible value to enable us to utilise the full range of the motor's capabilities. When implementing control algorithms in the acceleration space, we made sure the acceleration values were within the bounds achievable by the motors when translating the requested acceleration to the corresponding velocity command.

3.3 Robot motion corpus design

In this section, we present the seven variables of our robot motion corpus. In the first subsection, we describe the three variables used to define velocity profiles i.e. curves giving the velocity of the mobile robot over time. In the second subsection, we describe the four variables related to the visual appearance and audio aspects of the mobile robot.

3.3.1 Velocity profile design

The primary goal of the corpus is to enable the study of a robot's motion and kinematics parameters' impact on people. *Velocity profiles* specify the robot's movement, and are built by combining the values of three corpus variables: the *motion sequence*, *kinematics type* and *variant*. The motion sequence determines what we could call the general "shape" of the motion, in terms of speeding up, slowing down, maintaining speed. The kinematics type determines how abruptly the changes in velocity occur, and how fast the robot moves i.e. the slopes and maximum value of the *velocity profile*. These dimensions are related to the amount of kinetic energy required to perform the motion, hence the kinematics types represent different energy levels. The *variant* determines the fine details of the robot's motion, in order to add certain characteristics to it, such as smooth or saccadic.

In the following subsections, we present the variables which define the velocity profiles, and discuss the factors which had to be accounted for in the design process. These factors arose through five successive cycles of implementing and filming various motions on the robot and testing them on a few people in order to determine how the motions looked to a human bystander and on video.

Motion sequences

A *motion sequence* is a succession of motion phases, which can be acceleration phases, constant velocity phases, or deceleration phases. This essentially describes the type of motion performed by the robot. There are several choices for how to design the motion in each phase, mostly in terms of the shape of the velocity curve over time, which could be linear, exponential, logarithmic, sigmoidal, or other types of curves. The impact of using a given type of curve on people's perceptions of the robot has very rarely been studied (see (Schulz et al., 2020) for a comparison of linear and *slow-in, slow-out* velocity profiles). Ideally, we would compare the effect of different curve types, however in this study we limit ourselves to linear curves. We chose to use linear curves since they are the simplest type of curve, both to implement and to analyze.

The six motion sequences are illustrated in Fig. 3.2. Within the space of all feasible kinds of motions, we aimed to identify the basic building blocks that are representative of most mobile robot motions. The building blocks are:

1. accelerating from a standstill;
2. decelerating to a halt;
3. accelerating, constant velocity, then decelerating;
4. accelerating from a slower velocity;
5. decelerating to a slower velocity;
6. decelerating, constant velocity, then accelerating;
7. accelerating then immediately decelerating;
8. decelerating then immediately accelerating;
9. accelerating from a standstill, then constant velocity;
10. constant velocity, then decelerating to a halt.

Once we had established the building blocks, we determined six motion sequences (A, B, C, D, E, F) which contain the building blocks. Motion sequences A and C are designed to introduce a notion of "pausing" between acceleration and deceleration phases, by inserting small plateaus of constant velocity. The length of these plateaus ($300ms$) was chosen as an analogy to other communication aspects such as duration of a syllable, or of a sign in sign

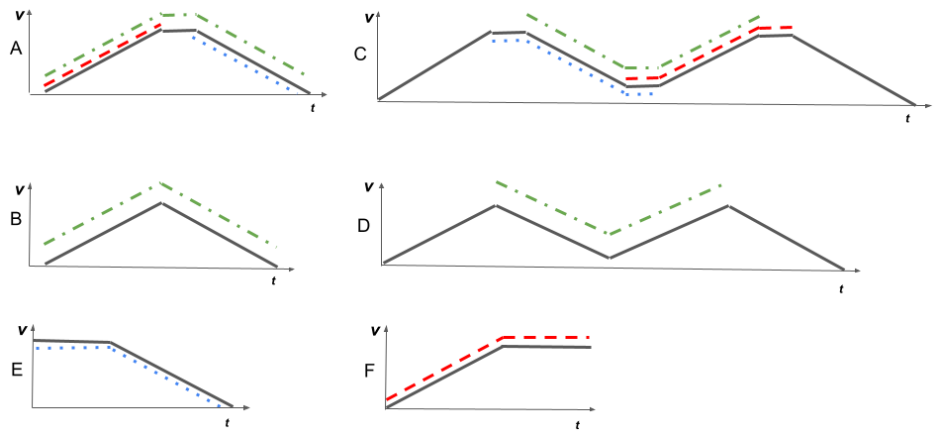


Figure 3.2: Illustrations of the effect of the six motion sequence values (solid lines), and ten building blocks (dashed lines) on the overall velocity profile. The motion sequence only determines the broad shape of the profile in terms of the ordering of accelerations and decelerations. The slope and maximum values of the profiles are determined by the *kinematics type*, and the fine details are determined by the *variant*.

language. We chose a value within this order of magnitude while also making sure that it was long enough to be perceptible when viewing the robot's motion. Motion sequences C and D incorporate a low-velocity phase half-way through the sequence, which may evoke a form of hesitation, as studied in the area of gestures in HRI (Moon et al., 2013).

Of course, choices have to be made regarding the duration of the acceleration and deceleration phases, which we will detail in the following part.

Kinematic types

We use the term *kinematic type* to refer to the bounds which are set on the robot's velocity and acceleration. Velocity and variations of velocity over time (acceleration) are the most basic descriptors of movement, that still allow us to capture a wide enough range of robot motion. Jerk (the time-derivative of acceleration) could be interesting to study, but it is a more advanced notion, and is typically non-trivial to account for in most current social navigation algorithms, so we do not explicitly control or study it in our work. The relevance of acceleration as a factor which influences people's perceptions of robots was stated in (Saerbeck & Bartneck, 2010), which studied the impact of acceleration and curvature on people's attributions of emotions and affects to two different robots.

Our goal was to choose three sets of values of the velocities (v_{min} and v_{max}) and accelerations a for the three values of the kinematic type variable. Given these kinematics types are intended to correspond to low, medium or high-energy motion styles, we chose to pair velocities and accelerations in a coherent way. For example, if we choose a *low* acceleration value, we combine it with *slow* v_{min} and v_{max}. Pairing a *low* acceleration value with a *fast* v_{max} could be feasible, but it introduces more subtleties when trying to compare two kinematics types in terms of the energy required to perform them.

The decision to use three kinematics types was made firstly for practical reasons, given that using more would imply a much larger corpus. Secondly, in this first study we aim to capture the extreme cases, which should provide us with the most contrast. The low and high energy types are the extreme values, and the medium type serves as a reference point between the extreme values.

In order to select the exact values to specify each kinematics type, there are several aspects one needs to balance and compromise on. Firstly, we have to account for the robot's physical capabilities. Naively using the robot's maximal velocity ($0.8m.s^{-1}$) and acceleration ($2.6m.s^{-2}$) as the high-energy type means the robot reaches its maximal speed within just $0.3s$, far too short to be properly perceived. Other aspects we had to balance were:

1. robot's physical motor limits;
2. minimum perceivable duration of acceleration;
3. duration of the whole motion (impact of exposure time of a stimuli on people's perception);
4. maintain similar duration of acceleration phase;
5. distance traveled (camera field of view and room limitations);
6. distinctness of minimal and maximal velocities;
7. distinctness of kinematic types.

Given all the constraints above, we selected an acceleration time from v_{min} to v_{max} of $1.0s$, so the acceleration phases to or from a zero velocity were between $1.25s$ and $1.5s$, depending on the kinematics type. This gives the motion sufficient length to be properly perceived by the viewer, while also allowing for a clear difference between the three kinematics types. The final values describing the kinematics types are shown in Table 3.1.

Table 3.1: Kinematics types parameters

Parameter	Low	Medium	High
a	$0.2m.s^{-2}$	$0.35m.s^{-2}$	$0.5m.s^{-2}$
v_{min}	$0.05m.s^{-1}$	$0.15m.s^{-1}$	$0.25m.s^{-1}$
v_{max}	$0.25m.s^{-1}$	$0.50m.s^{-1}$	$0.75m.s^{-1}$
0 to v_{max}	$1.25s$	$1.42s$	$1.5s$
v_{min} to v_{max}	$1.0s$	$1.0s$	$1.0s$

Variants

We designed two *variants* (incremental and saccade, illustrated in Fig.3.3), which modify the shape of the velocity profile locally. Once again, the design process for these variants has its roots in analogies to speech production variations which can be used intentionally by people, or originate from health issues or as a consequence of physical characteristics. When people manipulate their speech, it can become a means for them to alter the way other people perceive them.

In this study we aim to evoke two speech variations, hesitant speech and jittery speech. Hesitant speech could be observed when one is unsure of oneself, or not very confident, whereas jittery speech could be associated with frailness due to old age or health issues. Similar characteristics can be observed in people's gestures, and locomotion, which we aim to reproduce.

The *saccade* variant is an attempt to imitate jitter by introducing continuous shaking or stuttering of the robot by rapidly increasing and decreasing the velocity commands to the motors. There were two main conflicting aspects that had to be balanced: on the one hand, the resemblance to the type of dynamics observed in speech or human motion; and on the other hand, the reproducibility of the motion. Applying random perturbations to the velocity profile results in slightly more "natural" jitter, but it also means the motion sequence and kinematics type may become unrecognizable. In order to combat this effect, we perturbed the velocity commands in a deterministic, periodic fashion. Essentially, we determine a period and amplitude of the oscillations of the perturbed velocity profile around the value of the original profile. In our case, the period is $0.2s$, and the amplitude is fixed to different values according to the kinematics type (low: $0.044m.s^{-1}$, medium: $0.090m.s^{-1}$, high: $0.120m.s^{-1}$).

The *incremental* variant is an attempt to imitate hesitation by introducing increments into the acceleration and deceleration phases, meaning that an acceleration phase which is typically a single, constant acceleration applied for one second becomes a succession of three acceleration phases of a third of a second, interleaved by two constant velocity plateaus of $300ms$. These plateaus are a means of conveying pauses in the motion, in a similar fashion to motion sequences A and C, hence the same duration being used. Regarding the number of plateaus, we chose to use two plateaus simply because it allows each acceleration phase to be long enough to be perceptible, while provoking a big enough difference when compared to the original velocity profile.

When neither the increment nor saccade variants are applied to the profile, the motion phases conserve their original piecewise linear shape as in Figure 3.2, which corresponds to the *smooth* variant.

In this subsection, we presented the corpus variables which define the velocity of the robot during its movement. In the next subsection, we present the corpus variables which define the audio and visual aspects of the robot.

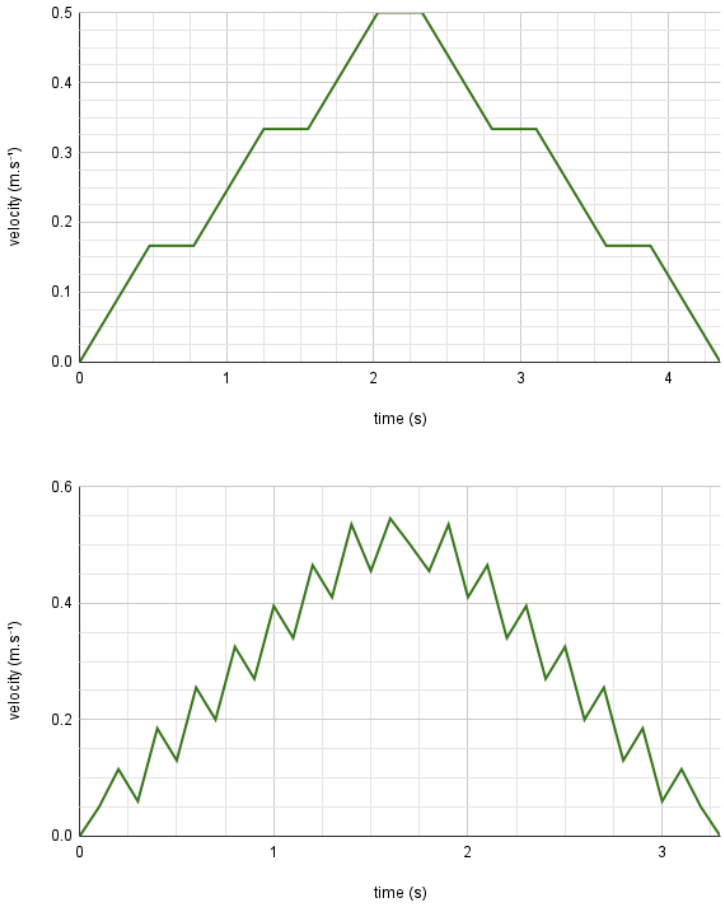


Figure 3.3: Velocity profiles resulting from combining the incremental (top) or saccade (bottom) variants with motion sequence A, and medium kinematics type.

3.3.2 Beyond velocity: robot appearance and body dynamics

Although our aim is to study the impact of motion parameters on human's perceptions, we need to keep in mind that other visual factors, as well as the sounds produced by the robot could also have an impact.

People communicate and perceive things holistically, by using and considering several modalities of expression. Therefore, if some variables other than the robot's kinematics also affect people's perceptions (which seems to be the case already for some of them studied independently like gaze (Fischer et al., 2016) or appearance (Magnani et al., 2017)), then their effect will be present in the results. This could become an issue if a variable has a *ceiling effect*, which would make it difficult or impossible to observe the effects of the motion variables we are interested in. We don't know if there are ceiling effects, because velocity, acceleration, hesitations, saccades, eyes, head motion, profile type, stability, and sound have not been studied together yet. Hence, we also manipulate other variables related to the robot's appearance, body motion, and sound to be able to detect and minimize any kind of bias or ceiling effect impacting people's perceptions.

Frail or robust robot

Typically, robots are designed with robustness in mind, which leads to robots which are stable, with well-mounted parts that do not shake even when the robot is under strain. The apparent physical stability and sturdiness of the mobile robot is an aspect which is rarely investigated, yet it may have an impact on how people perceive the robot's motion. We used two different robots to film our corpus, one typical stable robot and one modified, unstable robot. We modified an existing robot by loosening the front and back balancing wheel assemblies, and by loosely mounting its head on its body. The result is that the whole robot sways back and forth when changing speeds, especially when using the high-energy kinematics type, and its head shakes when using the saccade variant. This gives the robot body a different style of movement dynamics.

Eye shape and head movements

When robot designers include characteristics associated to living beings in their robot, it can impact how much people tend to anthropomorphize it, or how people interpret or perceive the robot's actions. Additionally, the exact shape of the eyes can also convey meaning. We use three eye variants for this corpus: switched off, round, and squinting. The round eye shape is part of the robot's design after a study where it was rated as the most "neutral" eye shape. The squinting eye shape was designed for this corpus in order to convey a colder, more unsettling feeling.

Gaze is a relevant means of interaction for living beings, and studies such as (Breazeal et al., 2005) have also shown it has an important role in HRI, due to its relation to attention, and its implicit signaling of the robot's perception capabilities. Gaze direction can also be tied to navigation and has been used to make motion more legible, as was studied in

(Fischer et al., 2016). For these reasons, we complemented the eye shape variants with four head settings: two settings where the head is stationary (facing straight, or facing the side, towards the camera) and two where the head moves during the robot's motion (from the straight position to the side, and vice-versa).

Audio recording

Most motors used in mobile robot locomotion are noisy, so any variation in the robot's motion also carries an audio signal. In some cases, navigation and control parameters do not cause a significant visual difference, but the change is still clearly audible through the motor noise. Other than motor noise, the chassis and other parts of the robot can also produce sounds which can give information about the structure of the robot. If the robot makes creaking and knocking sounds whenever it moves a bit too suddenly, we might deduce that the robot is not very well built. These types of sounds are called *consequential sounds* and have recently been the object of studies in HRI such as (Tennent et al., 2017) or (Robinson et al., 2021). In both studies, different sounds lead participants to perceive the robot's motion differently, highlighting the necessity of taking sound into account even when studying other dimensions of HRI.

The exposure to the sound produced by the mobile robot can therefore convey information, or be interpreted in various ways by the viewers, even if these sounds are simply direct consequences of the physical properties of the robot. Recording the sound of the mobile robot was therefore necessary, and could be used to contrast people's perception of the same motion when played back with or without the sound.

3.3.3 Summary of corpus variables

In Table 3.2, we summarize the variables we manipulated in order to obtain each video of the corpus, as well as the different values they can take. The sound variable is adjusted as a post-processing step: we record a given stimuli with sound, duplicate the video and mute one of them to obtain the silent version. In practice, we were unable to include all combinations of the values of these variables due to several limiting factors. Firstly, the stable robot's head was unable to rotate, meaning it was only filmed using the *straight* head setting. Secondly, due to time constraints on the corpus acquisition, we decided to remove certain combinations of variables: the saccades and incremental variants were not combined with round or squinting eye shapes. The resulting corpus contains 450 videos, for a total of 900 combinations of values of the seven corpus variables once we include the *sound* variable.

In the next subsection, we present the steps taken to ensure that we captured the robot's motion and appearance as faithfully as possible.

Table 3.2: Robot motion corpus variables.

	Corpus variable	Set of variable values
Motion variables	**Motion sequence**	$\{A, B, C, D, E, F\}$
	Kinematics type	$\{low, medium, high\}$
	Profile variant	$\{smooth, saccades, increment\}$
Appearance variables	**Base type**	$\{stable, unstable\}$
	Eye shape	$\{none, round, squint\}$
	Head setting	$\{straight, side, straight\_to\_side, side\_to\_straight\}$
	Sound	$\{with, without\}$

"Motion variables" affect the robot's velocity profile. "Appearance variables" encompasses variables related to visual and auditory perception of the robot.

3.4 Video corpus acquisition

In the previous subsection, we detailed the design of the robot's motions and appearance variables which constitute the corpus. In this subsection, we detail the considerations and precautions we took in order to produce a high-quality, exploitable corpus of videos. Various prior works have used video-based stimuli for HRI experiments involving moving robots (Carton et al., 2017; Chan et al., 2021; Knight et al., 2016; Torre et al., 2021), some of which validated their results on subsequent in-person experiments (Moon et al., 2013; Reinhardt et al., 2021). A recent study compared video and in-person experimental settings in the context of gestures with similar results, suggesting video studies may be appropriate (Honig & Oron-Gilad, 2020). Some of the variables explored in our corpus result in very slight visual differences in the robot's motion, such as the high-frequency stuttering and shaking induced by the saccade variant, or the swaying of the whole robot body when using the unstable base. For these reasons, we took extra precautions in order to capture the robot's movements as precisely as possible, and to make sure that they are well represented in the videos. We also aim to avoid any differences in the recording conditions that could introduce unwanted biases. All along the motion corpus design and recording, we consulted two experts (a professional videographer and a photographer) to discuss which parameters should be controlled in order to capture the robot's motion and appearance as precisely and truthfully as possible.

3.4.1 Robot movement consistency and framing

In order to minimize differences between two videos showing the same velocity profile, we implemented a method which allows us to execute a selected velocity profile automatically on the mobile robot. The profiles are represented as sequences of acceleration values, each associated to the corresponding duration over which it should be applied. This approach is described in more detail in our formalization of our navigation algorithm in chapter 4 (section 4.4). This also reduces the chance for errors during the corpus filming which is

already a tedious and time-consuming process. In addition, the whole control stack from the velocity profile control code down to the low-level motor controller was analyzed and modified when necessary in order to ensure the robot's motion was as faithful as possible to the velocity profile.

Regarding the camera framing, filming the robot moving towards the camera would give the impression of moving towards the viewer. In the end, we decided against it due to the unclear effect of the lack of depth perception resulting from the use of non-stereo video. Filming a robot motion parallel with the image plane should conserve as much information about the robot's motion as possible, unlike some prior video-based studies which required motion with components which are perpendicular to the image plane (Carton et al., 2017; Knight et al., 2016; Torre et al., 2021). The robot's initial position was also considered, as it may have a priming effect on people's anticipation and interpretation of the robot's motion. For example, if the robot starts very far on the right hand side of the frame, facing right, people could assume the robot will not travel very fast or far. In order to mitigate this, the robot starting point was selected such that the total motion to be performed was centered in the camera's framing. The exact position depends on the motion sequence, kinematics type, and variants; so in the interest of time, the initial positions were approximate and the motions were not always exactly centered.

3.4.2 Environment characteristics

Regarding the background, it is necessary to make sure that it is *mostly* uniform in order to avoid visual distractions, although when trying to capture movement it can help to have reference points such as vertical lines in order to better perceive the velocity of the robot. The background color should provide a high contrast with the robot's color. The type of ground on which the robot moves should also be considered in combination with the robot's drive assembly, given that any discontinuities in the ground could have repercussions on the robot's motion, and hence, visual appearance. Naturally, there should also be enough space in the environment to perform the longest velocity profile (six meters in our case). We also made sure there were no visible obstacles in the robot's direction of travel, since one could anticipate that the robot will start to slow down before reaching the obstacle.

The experts also highlighted the importance of lighting conditions to get the clearest possible picture of the robot, which is dependent both on the natural and artificial lighting of the room. In our case, strong natural light provided better lighting conditions than indoor artificial lighting, although this meant camera parameters had to be adjusted to compensate for the changing light throughout the day. Good lighting allows the details to be visible, and helps to clearly distinguish background from foreground.

3.4.3 Camera configuration and parameters

Initial filming tests revealed that smartphones are limited in terms of the field of view, and action cameras cause too much distortion with their fish-eye lenses. The experts informed us that the high-framerate recording action cameras provide is also not necessary for our

application. We also raised the question of shutter speed, which is tied to the amount of motion blur in an image, but given the speeds of the robot we were dealing with, standard shutter speed settings would suffice. The most important parameters were:

1. lighting conditions to avoid shadows;
2. high resolution to capture details;
3. stability of the camera.

One of the experts performed the final video recordings using a high-quality camera (Canon EOS 5D Mark IV) and advised us on the final framing, positioning, and lighting conditions. The camera and appropriate lenses and settings allowed us to frame wide enough to capture the full movements, while maintaining a good size of the robot in the frame, and high quality capture.

During the filming sessions, routine checks were made to ensure a consistent image over all 450 videos, despite the varying lighting conditions. One adjustment had to be made to the exposure settings of the camera in order for the robot's LED eyes to be clearly distinguishable. The resulting setting (under-exposure) was a compromise between image quality and visibility of the eyes. The exposure also had to be adjusted to compensate for the changing lighting conditions. We provide additional details regarding the camera, lenses, settings, and subsequent video post-processing, encoding, and formats in Appendix A.

3.5 Perception experiments

In order to analyze the effect of each variable of the corpus on the way in which a mobile robot is perceived we ran a series of three experiments; two online experiments where participants viewed videos of the robot and one embodied experiment, where the robot moved towards the participant, stopping at a pre-determined distance.

The goal of the online experiments was to collect participant's perceptions of the robot for the whole corpus, in order to establish a first baseline regarding which variables had significant influences on how participants perceived the robot. The goal of the embodied experiment was to attempt to replicate the findings of the online experiments for a subset of the corpus containing the variables which were found to be the most influential. All three experiments use similar methodologies.

3.5.1 Perceptual scales

In Table 3.3, we present the ten semantic differential scales which were used in all experiments in order to gather participants' social and physical impressions of the robot. Part of the scales represent *attitudes towards others*, such as Authoritative-Polite, Aggressive-Gentle, Inspires-Doesn't inspire confidence, Nice-Disagreeable, Tender-Insensitive. Evaluating these perceptions involves a directed attitude. Confident-Hesitant is more related to

the robot's own affective state. The remaining scales capture physical perceptions of the robot, with Sturdy-Frail, Strong-Weak, Smooth-Abrupt, Rigid-Supple. The scales were chosen based on words that participants in prior HRI studies had used to self-annotate their own recorded interaction data after a long experiment with a small butler robot (Guillaume et al., 2015; Sasa & Aubergé, 2016). The experiments were all conducted in French, hence the adjectives were translated to their closest vernacular equivalent in English for presentation here.

Table 3.3: Perceptual scales (original french wording in italic)

Adjective 1	**Adjective 2**
Aggressive *Agressif*	Gentle *Doux*
Authoritative *Autoritaire*	Polite *Poli*
Seems Confident *A l'air confiant en lui-même*	Doubtful, Hesitant *Doute, Hesitant*
Inspires confidence *Inspire confiance*	Doesn't inspire confidence *N'inspire pas confiance*
Nice *Sympathique*	Disagreeable *Antipathique*
Sturdy *Solide*	Frail *Fragile*
Strong *Fort*	Weak *Faible*
Smooth *Lisse*	Abrupt *Rude*
Rigid *Rigide*	Supple *Souple*
Tender *Tendre*	Insensitive *Insensible*

3.5.2 First online experiment: likert scale

The corpus of 900 stimuli was split into groups of 45 videos within which each value of each variable was represented. Thus, each participant viewed and rated all 45 videos of a given group, meaning they would see all values of all variables several times, but not all combinations. The order of the videos was randomized for each participant, and the number of participants for each video group was roughly balanced. Participants viewed each video once before rating it and moving on to the next. At the end of the experiment, participants could choose to leave a free-form comment. In the first online experiment, the rating scales were presented to participants as 5-point likert scales, where the middle

option signifies that neither of the adjectives correspond to their perception of the robot. Figure 3.4 shows how the videos and scales were presented on our website. The 900 stimuli were split into 20 groups, half with sound, and half without, meaning the sound variable was a between-subjects variable.

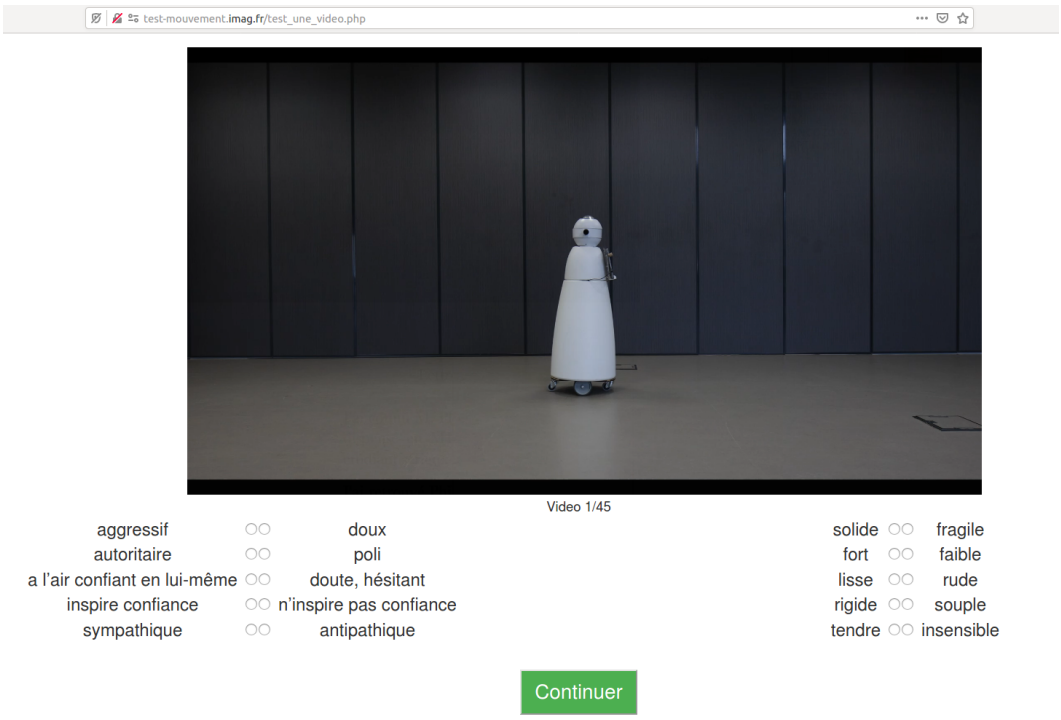


Figure 3.4: Online perception experiment website. This figure shows the second online experiment with binary choice. Presentation is identical for the first online experiment, except there are five response levels instead of two.

A total of $n = 42$ participants completed the first online perception experiment. Participants of all ages were recruited via university mailing lists, experiment recruiting lists, and social media. The first step in our analysis was to determine whether certain corpus variables or certain scales showed wide ranges of responses, or clear-cut bias to one of the opposing adjectives. For each value of each corpus variable, we computed the distribution of participants responses on all videos using that value, shown in Fig.3.5. With 42 participants, each of the 900 unique combinations of values of variables is only rated four times at most. However in the analysis of this first online experiment we do not study unique combinations of values, but rather all combinations that include a given value for a given variable such as all videos using the high kinematics type. In this arrangement, there are at least 300 ratings of videos using each value of each variable of the corpus. The responses followed normal distributions, generally with high mass around the center value as seen in Fig.3.5. For some scales, different values for certain corpus variables led to distributions that were shifted towards one of the adjectives such as the saccade variant shifting responses towards the doubtful/hesitant side of the confident-doubtful/hesitant scale. For all scales, a high percentage of the responses were on the neutral level (neither one nor the other end of a scale, column label 0), with half of the scales having 30 to 40% of neutral responses, and the other half having 40 to 45% of neutral responses. Distributions for participants who

had sound had an average of $15 \pm 5\%$ fewer neutral responses than those without sound.

We performed chi-square association tests to determine if there were dependencies between each motion variable and scale pairing. The resulting chi-square statistic and significance levels are reported in Table 3.4, with 46 out of the 70 pairings showing a significant dependence ($p < 0.05$). The significance levels are based on the adjusted p-values after Bonferroni correction, to account for multiple hypothesis testing.

Table 3.4: Chi-square association test results between corpus variables and perceptual scales, for the first online experiment. Reported as chi-square statistic, significance $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

	Aggressive Gentle	Authoritative Polite	Confident Doubtful	Inspires Conf Does not	Nice Disagreeable
Kinematics (2)	235 ***	198 ***	106 ***	n.s.	n.s.
Sequence (5)	n.s.	n.s.	221 ***	78 ***	n.s.
Variant (2)	47 ***	33 **	246 ***	178 ***	37 ***
Eyes (2)	53 ***	69 ***	145 ***	76 ***	118 ***
Head (3)	n.s.	n.s.	39 **	n.s.	n.s.
Base (1)	n.s.	n.s.	55 ***	47 ***	n.s.
Sound (1)	68 ***	60 ***	107 ***	76 ***	25 **

	Sturdy Frail	Strong Weak	Smooth Abrupt	Rigid Supple	Tender Insensitive
Kinematics (2)	50 ***	96 ***	59 ***	48 ***	44 ***
Sequence (5)	102 ***	90 ***	49 *	n.s.	n.s.
Variant (2)	306 ***	203 ***	97 ***	n.s.	n.s.
Eyes (2)	113 ***	107 ***	33 **	n.s.	75 ***
Head (3)	35 *	n.s.	n.s.	n.s.	n.s.
Base (1)	104 ***	53 ***	n.s.	n.s.	n.s.
Sound (1)	115 ***	46 ***	86 ***	82 ***	47 ***

The results of the association tests suggest that people's perceptions of a mobile robot along these ten perceptual scales may be dependent on several of the motion corpus variables, most notably the kinematics and variant ($\chi^2(2) \geq 33, p < 0.01$ for 8 out of 10 scales). The absence of sound leading to more neutral responses could indicate that the sound was informative, and did affect people's responses.

The high proportion of neutral responses across most scales and variables could indicate one of two situations:

1. neutral perception: the participant finds neither of the adjectives fitting to describe their perception of the robot;

		Aggressive (-) Gentle (+)					Authoritative (-) Polite (+)					Condifent (-) Hesitant (+)					Inspires conf. (-) Doesn't insp. (+)					Kind (-) Disagreeable (+)				
		-2	-1	0	1	2	-2	-1	0	1	2	-2	-1	0	1	2	-2	-1	0	1	2	-2	-1	0	1	2
Kinematics	high	12	25	42	17	4	10	26	43	18	3	18	29	23	19	10	6	22	34	23	15	6	19	46	20	9
	medium	3	15	45	26	11	3	15	47	27	8	7	25	30	24	14	5	25	36	20	14	7	25	43	16	8
	low	2	8	39	30	21	2	8	44	30	16	6	19	28	27	20	7	21	34	24	14	7	28	42	16	7
Motion Sequence	A	3	16	47	24	10	4	15	49	24	8	9	28	30	24	10	5	26	35	21	13	5	25	46	17	7
	B	6	18	40	24	13	4	16	47	24	9	6	25	30	22	16	4	22	36	23	15	8	22	43	18	9
	C	5	15	45	23	12	3	16	45	25	10	6	20	23	27	23	6	20	35	24	15	6	28	40	19	7
	D	9	16	41	22	12	5	14	47	27	8	5	12	24	33	27	3	14	30	31	22	6	19	41	22	12
	E	6	15	41	26	13	7	18	41	25	9	22	31	28	14	5	8	31	36	16	9	8	25	48	11	8
	F	5	16	39	27	13	8	19	38	25	10	15	30	28	20	8	9	23	35	21	12	7	24	44	18	6
Variant	smooth	5	17	40	24	14	6	17	41	26	10	14	30	30	19	7	9	27	35	19	10	8	25	42	16	8
	saccade	10	19	43	22	6	5	17	54	20	5	4	13	20	30	33	1	10	31	27	31	4	17	46	21	12
	increment	4	11	46	27	12	3	13	46	28	9	7	18	25	30	20	3	22	36	28	12	4	26	47	17	5
Eyes	none	6	15	45	24	11	4	15	49	25	8	7	19	26	27	20	4	20	35	24	17	4	22	49	17	8
	round	4	11	36	31	18	5	13	37	32	13	13	33	29	20	5	11	33	32	16	7	16	33	34	12	5
	squint	7	24	40	19	11	9	24	39	20	8	17	32	27	16	7	8	21	35	23	13	5	21	37	25	12
Head	straight	6	14	44	24	13	6	16	45	25	9	14	23	27	23	13	7	24	37	18	13	6	24	47	15	9
	side	5	15	40	25	14	4	16	42	27	10	8	24	25	23	19	4	24	31	26	15	7	27	42	17	7
	side to straight	4	18	44	24	10	4	18	46	25	7	6	26	25	27	16	4	20	36	26	14	6	23	42	22	6
	straight to side	7	19	39	23	12	7	17	43	24	10	9	27	30	20	14	7	21	32	24	17	10	21	39	19	10
Base	unstable	5	16	42	25	12	5	16	45	26	8	8	24	26	24	17	5	21	34	25	16	7	24	43	18	8
	stable	6	15	43	22	15	6	18	43	22	11	19	26	30	19	7	11	28	38	14	9	7	23	47	14	9
Sound	sound	7	19	34	24	16	6	19	37	27	12	11	27	18	24	20	8	23	27	23	19	7	24	39	20	10
	no sound	4	13	51	24	8	4	13	53	24	6	9	22	37	22	9	4	22	43	21	9	6	24	49	15	6

		Sturdy (-) Frail (+)					Strong (-) Weak (+)					Smooth (-) Abrupt (+)					Rigid (-) Supple(+)					Tender (-) Insensitive (+)				
		-2	-1	0	1	2	-2	-1	0	1	2	-2	-1	0	1	2	-2	-1	0	1	2	-2	-1	0	1	2
Kinematics	high	12	31	27	20	9	13	34	34	15	5	14	20	37	22	8	19	36	33	10	2	2	13	47	19	19
	medium	6	26	33	23	12	6	26	40	21	7	13	25	42	15	5	14	33	39	12	3	1	14	57	15	13
	low	5	24	34	28	10	3	21	38	27	10	15	27	44	13	1	11	26	43	16	4	3	22	48	13	13
Motion Sequence	A	6	30	31	24	9	7	28	37	21	7	13	25	45	16	2	16	28	42	12	2	1	19	49	17	15
	B	5	26	35	23	11	6	28	39	22	6	13	26	40	17	4	13	34	38	13	1	3	15	51	17	14
	C	4	27	29	28	12	4	26	38	21	10	14	22	41	18	5	12	33	39	12	4	2	20	48	15	16
	D	5	17	31	30	18	4	19	35	29	13	12	17	41	20	10	16	33	36	12	3	2	15	51	18	14
	E	15	33	29	16	7	13	33	36	13	5	17	27	40	12	4	16	32	35	15	2	3	15	49	14	18
	F	10	30	34	20	6	11	30	37	18	4	14	27	39	16	4	16	30	38	14	3	3	16	54	16	12
Variant	smooth	10	34	36	17	4	9	33	39	15	3	15	28	41	13	3	13	32	39	14	3	3	18	50	15	14
	saccade	3	13	21	34	29	3	15	29	33	19	11	14	39	24	12	20	33	34	10	3	1	12	49	21	17
	increment	6	21	29	33	10	7	21	39	25	8	14	23	41	20	2	14	30	40	14	2	1	16	52	15	15
Eyes	none	6	22	29	28	14	6	22	36	25	11	13	21	41	19	6	16	31	38	12	3	1	15	52	16	16
	round	9	34	36	17	5	9	30	42	16	3	16	30	40	12	2	12	29	39	16	3	5	26	49	9	10
	squint	11	35	35	16	4	10	39	36	12	3	13	27	40	14	6	14	36	36	12	2	2	13	47	21	17
Head	straight	11	28	31	21	8	10	27	37	20	6	15	26	40	15	4	15	32	38	13	3	2	14	52	16	16
	side	5	24	31	26	14	6	26	37	21	9	15	25	41	15	5	13	31	40	13	2	1	20	55	12	12
	side to straight	4	28	34	24	10	5	27	38	22	8	12	23	41	19	5	14	31	40	13	2	2	15	49	19	15
	straight to side	7	27	31	24	11	7	29	37	20	7	14	20	41	19	5	18	32	34	13	3	2	20	45	16	16
Base	unstable	6	25	30	26	12	6	26	37	23	9	13	23	41	17	5	14	32	38	13	2	2	17	51	15	15
	stable	15	34	37	12	2	12	34	39	12	3	17	28	38	14	3	16	31	37	13	12	2	14	50	19	15
Sound	sound	9	30	21	27	13	8	29	31	23	9	17	25	32	20	7	19	35	30	15	2	2	13	49	19	18
	no sound	6	25	43	19	7	7	25	44	18	5	11	23	51	12	3	10	28	47	11	4	3	21	52	13	11

<= 6.3%	<= 10%	< 15%	> 30%	>= 40%

Figure 3.5: Response distributions in percentages for the likert scale online experiment. Columns represent response levels for each perceptual scale. Rows represent corpus variable values by which the video ratings are grouped to compute the percentage of responses.

2. uncertainty: the participant feels unsure of their answer, and prefers to give a neutral response rather than answer in a way that *they perceive* as random.

This was reflected in several participants' comments, stating that they felt like it was difficult to answer, or that they answered randomly. While the association tests did report statistical significance of the dependencies, further investigation into the underlying cause of the neutral responses could give a better idea of the robustness of the associations. We designed the second perception experiment in order to explore this phenomena.

3.5.3 Second online experiment: binary choice

We designed a second online experiment which was similar to the first in most aspects, but with a few key changes aimed at extracting more useful responses from participants. In this experiment, the rating scales were presented to participants as simple binary choices between the two adjectives, rather than a 5-point likert scale. This was done so that participants could not answer with a neutral response, as many did in the first experiment. If participants actually have consistent inclinations towards certain perceptions for given variables, then forcing them to pick a side of the scale may result in higher mass of responses on one end of the scale. However if the perception of participants is truly not impacted by the variable, the responses may become more random, and the mass equally spread to either side of the scale. The forced choice design of this experiment allows us to determine which of these two situations occurs. Regarding the set of stimuli, we chose to use only the half of the stimuli which had sound for two reasons. Firstly, the presence of sound is more realistic since our robot cannot be made silent (and neither can most robots). Secondly, the first experiment results suggested that removing the sound only made participants' answers more neutral, losing part of the information about the robot's motion. This change reduced the total number of videos to 450, split into 10 groups.

Response distributions and scale correlations

A total of $n = 101$ participants completed the second online perception experiment. Participants of all ages were recruited via university mailing lists, experiment recruiting lists, and social media. When observing the distributions of responses, the trends of the first experiment tended to be confirmed. Distributions with an existing bias towards one side of a scale were shifted further towards that side, and distributions with little to no bias remained similar (see Fig.3.6[4]). For example, in the first experiment 51% of the stimuli using low kinematics were perceived as gentle, 10% as aggressive, and the remaining 39% as neither (neutral response), so the responses were biased towards the gentle perception. In the second experiment with the forced binary choice, a similar tendency is observed: 88% were perceived as gentle, and the remaining 12% as aggressive. If some participants were truly undecided and answered the forced choice question at random, we could have expected the response distribution to even out, rather than become more extreme.

[4]This table is available on our project page https://osf.io/5csrg/ which will be updated as the project continues.

		Aggressive (-) Gentle (+)		Authoritative (-) Polite (+)		Condifent (-) Hesitant (+)		Inspires conf. (-) Doesn't insp. (+)		Kind (-) Disagreeable (+)	
		-	+	-	+	-	+	-	+	-	+
Kinematics	high	59	41	56	44	63	37	39	61	43	57
	medium	33	67	34	66	50	50	43	57	53	47
	low	12	88	17	83	40	60	51	49	66	34
Motion Sequence	A	33	67	33	67	53	47	45	55	52	48
	B	33	67	30	70	47	53	45	55	57	43
	C	35	65	35	65	36	64	37	63	53	47
	D	34	66	34	66	29	71	33	67	52	48
	E	39	61	44	56	74	26	52	48	52	48
	F	34	66	39	61	67	33	53	47	56	44
Variant	smooth	32	68	36	64	62	38	53	47	58	42
	saccade	47	53	41	59	26	74	21	79	44	56
	increment	30	70	30	70	43	57	41	59	53	47
Eyes	none	36	64	34	66	43	57	38	62	51	49
	round	26	74	32	68	63	37	59	41	66	34
	squint	38	62	44	56	64	36	50	50	51	49
Head	straight	37	63	35	65	46	54	41	59	52	48
	side	35	65	34	66	48	52	44	56	53	47
	side to straight	35	65	36	64	51	49	43	57	55	45
	straight to side	34	66	36	64	49	51	44	56	54	46
Base	unstable	35	65	35	65	48	52	43	57	54	46
	stable	32	68	38	62	62	38	51	49	55	45

		Sturdy (-) Frail (+)		Strong (-) Weak (+)		Smooth (-) Abrupt (+)		Rigid (-) Supple(+)		Tender (-) Insensitive (+)	
		-	+	-	+	-	+	-	+	-	+
Kinematics	high	61	39	62	38	53	47	75	25	27	73
	medium	49	51	46	54	61	39	68	32	37	63
	low	42	58	34	66	75	25	55	45	47	53
Motion Sequence	A	53	47	50	50	66	34	64	36	38	62
	B	44	56	42	58	63	37	64	36	41	59
	C	44	56	41	59	60	40	64	36	36	64
	D	36	64	33	67	58	42	71	29	36	64
	E	66	34	61	39	64	36	68	32	33	67
	F	60	40	56	44	67	33	65	35	38	62
Variant	smooth	62	38	57	43	68	32	62	38	40	60
	saccade	22	78	25	75	47	53	78	22	29	71
	increment	44	56	41	59	62	38	66	34	36	64
Eyes	none	42	58	40	60	59	41	69	31	34	66
	round	62	38	56	44	74	26	58	42	51	49
	squint	65	35	61	39	64	36	65	35	34	66
Head	straight	44	56	43	57	61	39	66	34	36	64
	side	48	52	46	54	61	39	65	35	39	61
	side to straight	51	49	47	53	63	37	66	34	37	63
	straight to side	46	54	43	57	62	38	67	33	38	62
Base	unstable	47	53	45	55	62	38	66	34	38	62
	stable	65	35	57	43	68	32	66	34	36	64

<= 20%	<= 30%	< 40%		> 60%	>= 70%	>= 80%

Figure 3.6: Response distributions in percentages for the binary choice online experiment. Columns represent response levels for each perceptual scale. Rows represent corpus variable values by which the video ratings are grouped to compute the percentage of responses.

Some of the perceptual scale response distributions seem to be affected in a similar manner by the corpus variable levels, such as the impact of high and low kinematics on the aggressive-gentle and authoritative-polite scales. We computed the tetrachoric correlation (Kirk, 1973) between the responses for each perceptual scale, which is a type of correlation measure suited to binary variables. The correlations are shown in Fig.3.7. In addition, we performed hierarchical clustering on the scale correlation values in order to cluster scales with similar correlation patterns together. The scales in the correlation matrix are ordered according to these clusters, placing similar scales near each-other. The clusters can be more easily represented with the dendrogram in Fig.3.8. The rigid-supple, aggressive-gentle, and authoritative-polite scales all show similar patterns of moderate inverse correlation (between -0.61 and -0.35) with kind-disagreeable, inspires-doesn't inspire confidence, tender-insensitive, and smooth-abrupt. The highest correlations are sturdy-frail and strong-weak (0.86), aggressive-gentle and authoritative-polite (0.82), confident-hesitant and sturdy-frail (0.74), kind-disagreeable and tender-insensitive (0.73). On the other hand, some scales have very low correlation, such as strong-weak and tender-insensitive (-0.02).

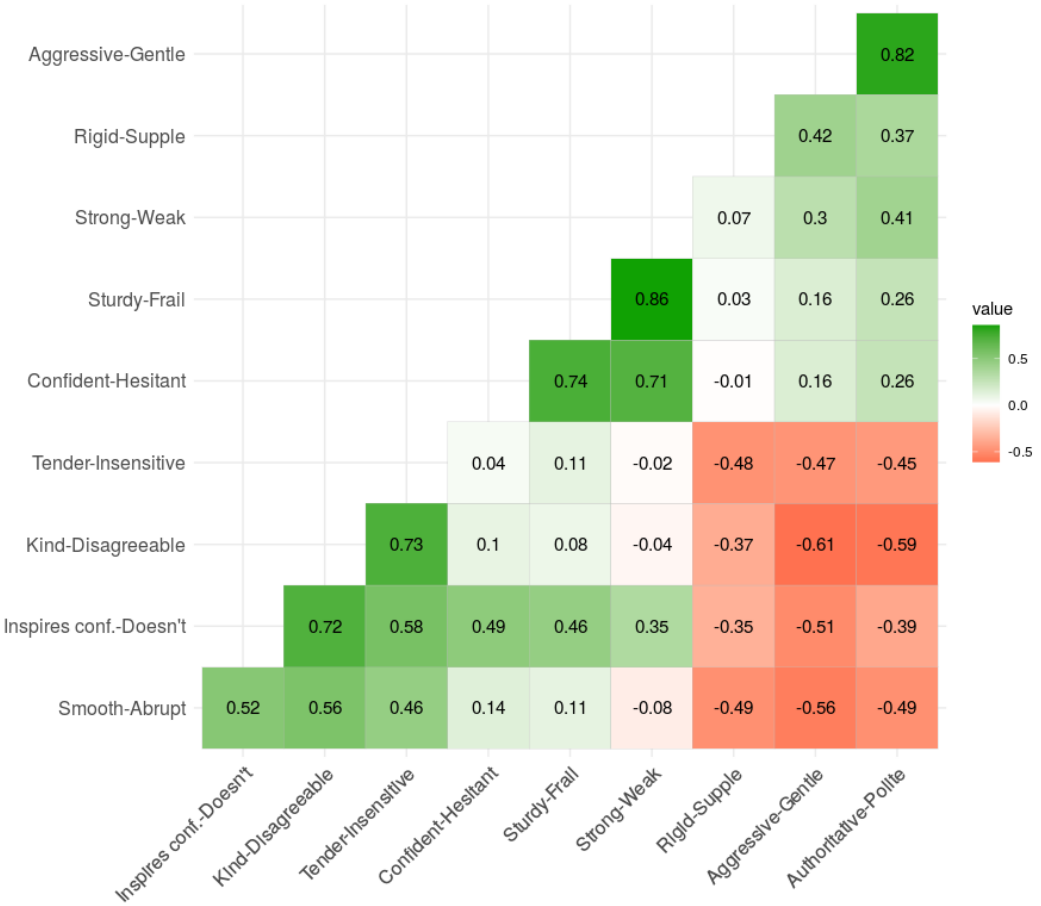


Figure 3.7: Correlations between the responses along each perceptual scale. Scales with similar correlation structures are grouped based on hierarchical clustering.

While the table of response distributions grouped by corpus variable values presented in Fig.3.6 is useful to uncover the more obvious trends in the data, it fails to represent the

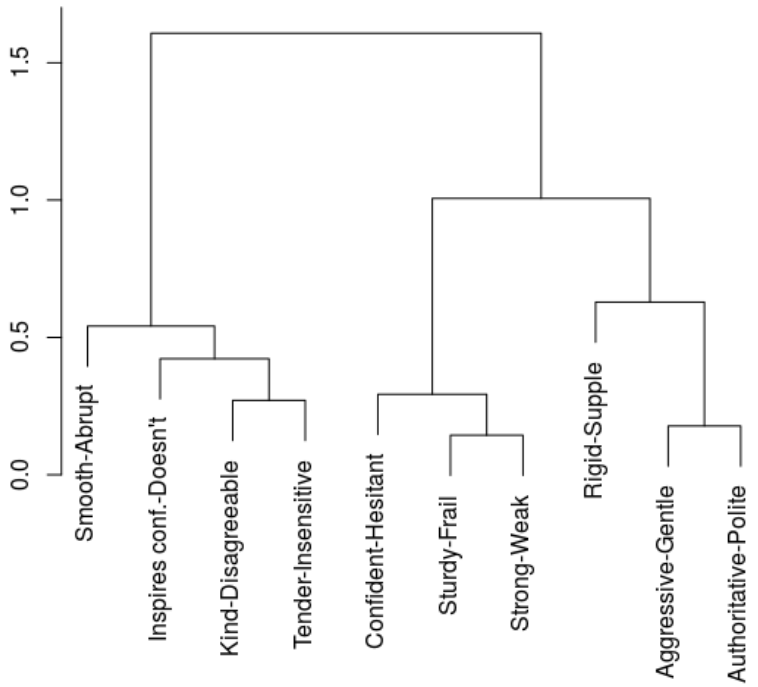


Figure 3.8: Dendrogram representing the hierarchical clustering of perceptual scales according to their correlations. The height between branching points indicates the dissimilarity between the clusters.

fact that the number of responses for each variable value is not balanced. Furthermore, it does not provide us with any information about possible interactions between the motion corpus variables. Similarly, the chi-square analysis performed for the first experiments only allows us to test the statistical significance of the scale dependencies on the corpus variables, without giving us an understanding of the magnitude of the effects of the variables on the perceptual scales. Lastly, when we consider each participant's average response on a given scale, we find that different participants have different baseline perceptions of the robot. As an example, we show the histogram of the participants' averaged responses on the aggressive-gentle scale in Fig.3.9. While a majority of participants perceived the robot as gentle with probability 0.6, the range of average values goes from 0.2 to 1.0. These issues could be addressed by using statistical methods which account for imbalanced data, as well as for the fact that responses from the same participant may be correlated. We now present our approach for the statistical modelling of the data.

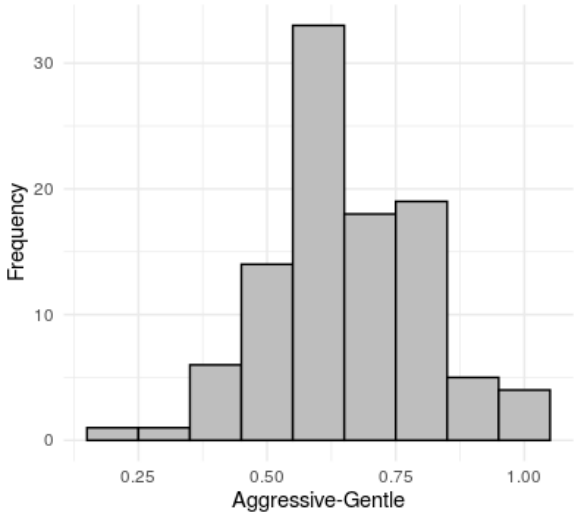


Figure 3.9: Histogram showing the frequency of per-participant averaged responses on the aggressive-gentle scale. The average is computed over all the participant's responses, showing their overall bias towards either side of the scale, regardless of the values of the motion variables.

Statistical model selection

We chose to fit a statistical model to the data, rather than only performing hypothesis tests. The model should capture how the different values of the six motion corpus variables affect participant's binary responses on the perceptual scales. We chose to use a logistic regression approach, fitting one model for each perceptual scale. More specifically, we chose to fit mixed effects logistic regression models (for details on mixed effect models, we refer the reader to (Winter, 2019)). Mixed models are able to account for the fact that the responses are not truly independent, since participants each provide 45 responses. A mixed effect model is defined by the choice of fixed effects as in classical logistic regression, as well as random effects, which capture dependencies or hierarchies within the data, such as our

repeated measures design. We used the same model structure for all 10 perceptual scales, where each of the corpus variables is treated as a categorical fixed effect, and the participant *id* is used as a random effect. Once the model is fitted, the regression coefficients for the fixed effects will allow us to interpret not only the direction of the effect of a given value of a corpus variable on the perceptual scale, but also its magnitude, in terms of how much the probability of responding on either end of the scale changes. In addition, the model estimates an intercept term for each participant accounting for the fact that each participant may have a different baseline perception of the robot. The models were implemented using the R programming language, using the $lme4$ package (Bates et al., 2015). The participants' responses on each scale were coded as 0 when they responded with the first (leftmost) adjective of the scale, and 1 for the second (rightmost) adjective, meaning the logistic model is modelling the probability of the participant choosing the second (rightmost) adjective of the scale. In R syntax, the model structure can be given as follows:

$$scale \sim \ kinematics + sequence + variant + eyes + base + head + (1|id) \tag{3.1}$$

The model in equation 3.1 models the effect of each variable individually, i.e. it is assumed that the effect of one variable does not depend on the value of another variable. We could also construct a model which accounts for this possibility by including interaction terms. In order to decide whether or not to include interaction terms, we fitted models with and without pairwise interaction terms on a training subset comprised of 80% of the response data, and evaluated the model's predictions on the remaining 20% comprising our test set. Following (Baayen, 2008), in order to assess the models we used the Area Under Curve (AUC) of the Receiver Operator Characteristic (ROC) curve (Fawcett, 2006). This value ranges from 0 to 1, and can be interpreted as the probability with which the logistic regression model will assign a higher probability to a randomly chosen positive instance than to a randomly chosen negative instance. $AUC = 0.5$ represents a random classifier, $AUC = 1$ a classifier which is always correct, and an AUC of 0.8 or higher is considered as an indication that the model has good predictive capability (Baayen, 2008). The resulting AUC values are shown in Table 3.5 for the predictions made using only the fixed effect coefficients and the global average intercept (without random effects), as well as for predictions made using each subject's intercept value (with random effects). In practice, in our human-robot interaction scenario we may encounter situations where the robot already knows the person, in which case we might have prior data on their perception of the robot, allowing us to leverage the random effect for that person to get a more accurate prediction. We might also encounter situations where the robot has never interacted with that person before, so we can only rely on the global average intercept value, meaning we do not leverage the random effect for prediction. When evaluating the models on the training data, the models with pairwise interactions had slightly better accuracy than the models without interactions, however this trend was reversed in the results on the test data shown in Table3.5, suggesting that the interaction models may be overfitting. We decided to perform the rest of our analysis using the models without interaction terms, as formulated in equation 3.1.

Table 3.5: Model comparison using the AUC metric computed on a test dataset made up of 20% of the observations, which were not used to fit the models. Comparisons are shown for each scale between different models (with/without interaction terms) and between prediction methods (with/without per-participant random effects (r.e.)).

	Pairwise interaction model		No interaction model	
	Without r.e.	With r.e.	Without r.e.	With r.e.
Aggressive-Gentle	0.772	0.809	0.783	0.819
Authoritative-Polite	0.750	0.812	0.746	0.814
Confident-Hesitant	0.795	0.823	0.792	0.819
Inspires conf.-Doesn't	0.657	0.730	0.693	0.755
Kind-Disagreeable	0.627	0.781	0.623	0.782
Sturdy-Frail	0.762	0.837	0.771	0.844
Strong-Weak	0.733	0.819	0.745	0.831
Smooth-Abrupt	0.658	0.785	0.654	0.788
Rigid-Supple	0.620	0.779	0.626	0.789
Tender-Insensitive	0.634	0.801	0.635	0.805
Average	0.701	0.798	0.707	0.805

Mixed effect logistic regression results

Results of the modelling for the aggressive-gentle scale are reported in Table 3.6. The first column lists the fixed effects along with their levels. The second column gives the regression coefficients which represent changes in the log odds:

$$\text{log odds} = log(\frac{p}{1-p}) \tag{3.2}$$

where p denotes the probability of an event occurring, which in our case is the probability of the participant selecting the second word of the perceptual scale. One may notice that there are no coefficients given for the first level of each corpus variable. This is because those levels are used as the reference levels with respect to which the changes in log odds are expressed. The intercept coefficient given in the first row corresponds to the log odds when all of the variables are set to their reference levels. In order to use the model to predict the log odds for other variable combinations, we add the log odds values for each variable that we change. A coefficient value of 0 indicates that the corresponding variable level has no effect on the log odds of the outcome. Positive (resp. negative) log odds correspond to higher (resp. lower) probabilities of the participant responding with the second word of the scale. For example, the reference level for the kinematics variable is *high*, and the positive coefficient for the *low* level indicates that changing from *high* to *low* increases the log odds of observing a *gentle* response on the aggressive-gentle scale by $+2.8$. If all other variables are at their reference levels, then the log odds for that combination of of variables is obtained by summing the intercept and the low kinematics coefficient: $0.07 + 2.8$.

Participant-specific predictions require adding the participant's random intercept to the computation. The third column gives 95% confidence intervals for the coefficients, and the fourth column gives p-values resulting from Wald tests against the null hypothesis that the coefficients are zero.

Table 3.6: GLMM logistic regression coefficients for the aggressive-gentle scale.

Variable	**log(OR)**[1]	**95% CI**[1]	**p-value**
Intercept	0.07	-0.28, 0.41	>0.9
Kinematics			
high	—	—	
low	2.8	2.6, 3.0	<0.001
medium	1.3	1.1, 1.5	<0.001
Sequence			
A	—	—	
B	0.02	-0.24, 0.27	>0.9
C	-0.09	-0.34, 0.16	>0.9
D	-0.07	-0.32, 0.19	>0.9
E	-0.37	-0.62, -0.12	0.042
F	-0.05	-0.30, 0.21	>0.9
Variant			
increment	—	—	
saccade	-1.0	-1.2, -0.78	<0.001
smooth	-0.10	-0.33, 0.13	>0.9
Eyes			
none	—	—	
round	0.36	0.13, 0.60	0.031
squint	-0.39	-0.62, -0.17	0.009
Base			
stable	—	—	
unstable	-0.26	-0.49, -0.03	0.2
Head			
straight	—	—	
side	0.10	-0.13, 0.33	>0.9
turn_side	0.16	-0.07, 0.39	>0.9
turn_straight	0.10	-0.13, 0.33	>0.9
Id.sd__(Intercept)	0.96		

[1]OR = Odds Ratio, CI = Confidence Interval

The model coefficients presented above can be used to compute predictions of the probability of a given response along the aggressive-gentle scale, given the combination of values selected for each corpus variable. These predictions can be made at the population level, or for individual participants by leveraging the random effects. In this thesis, we are mostly interested in interpreting the models to understand the relative effects of each corpus variable on participants' perceptions. For this reason we do not discuss the regression

coefficients themselves, and the nine other GLMM models are made available in Appendix D. The models are used in our next analysis step to generate more interpretable statistics. We propose to study the effect of each variable by computing marginal means using the model predictions, as discussed in the following section.

Contrasts between corpus variable values

The log odds scale is practical for fitting the models and computing predictions, however it is not a very intuitive scale to interpret the results. Instead, we would like to perform a transformation to the probability scale. Because the transformation from log odds ratios to probabilities is non-linear, the interpretation of the model coefficients on the probability scale becomes dependent on the per-participant random effects. Instead, we would like to be able to reason about the data at the population level rather than conditioned on each participant. One way of interpreting the results of such mixed effect logistic regression is to compute the estimated marginal means (EMM), based on the model predictions (S. R. Searle & Milliken, 1980). These means represent the average of predicted values of the response variable for each level of the corpus variables. Averaging is performed across all levels of all other variables. The difference with the observed means reported in Fig.3.6 is that the estimated means are computed using our mixed models, which account for the imbalance in the combinations of variables in our corpus. In order to establish the relative effects of the different levels of the corpus variables on each scale, we construct contrasts that compare each level's EMM with the average over all levels. We perform the EMM and contrast computation using the *emmeans* R package (Lenth, 2023).

Table 3.7 presents the marginal effects of each value of each corpus variable on the perceptual scales. The marginal effects are reported as percentage points, indicating the increase or decrease in the probability of participants selecting the second adjective of the scale when that level of a corpus variable is used, compared to the overall mean. For example using the high kinematics is estimated to decrease the probability of *gentle* being selected over *aggressive*, or equivalently, increase the chance of people perceiving the robot as aggressive by 28 percentage points compared to the overall mean. The statistical significance of the difference between the EMM for a given level and the average EMM over all levels was tested using z-tests. A Holm-Bonferroni correction (Holm, 1979) was applied to the p-values to adjust for multiple comparisons.

In order to interpret the results we will focus on the contrasts which have larger absolute values, indicating that the associated variable value causes large changes in the probability (expressed in the table as percentage points (pp)) with which a person will associate the robot with either adjective of the associated scale. The contrast values range from $-28pp$ (high kinematics effect on aggressive-gentle), to $+27pp$ (sequence D effect on confident-hesitant), with all values in between, including some null contrasts (sequence C effect on aggressive-gentle). While the statistical significance tests indicate which contrasts are unlikely to be due to random sampling, one could argue that *statistically* significant contrasts with small absolute values might not be of *practical* significance, such as the contrasts for the stable/unstable base variable's effect on the aggressive-gentle scale ($\pm 3pp$). In the following paragraphs, we will therefore only consider the contrasts which are both statistically

Table 3.7: Marginal effects of the corpus variables on the perceptual scales, in percentage points (pp). $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

- / +	Aggressive / Gentle	Authoritative / Polite	Confident / Hesitant	Inspires Conf. / Does not	Nice / Disagreeable
Kin. high	-28 ***	-24 ***	-17 ***	7 ***	15 ***
Kin. low	24 ***	22 ***	15 ***	-8 ***	-15 ***
Kin. medium	4 ***	3 *	2	1	1
Sequence A	2	4	-3	-1	2
Sequence B	2	7 **	5 *	-1	-4
Sequence C	0	2	19 ***	9 ***	1
Sequence D	1	2	27 ***	13 ***	1
Sequence E	-6 *	-10 ***	-28 ***	-9 ***	3
Sequence F	1	-5	-21 ***	-11 ***	-3
Var. increment	8 ***	6 ***	-1	-4 **	-3 *
Var. saccade	-14 ***	-8 ***	22 ***	20 ***	10 ***
Var. smooth	6 ***	3	-21 ***	-16 ***	-6 ***
Eyes none	1	5 **	4	3 *	3 *
Eyes round	7 ***	5 **	-1	-7 ***	-11 ***
Eyes squint	-8 ***	-10 ***	-3	4 *	8 ***
Stable	3 *	-2	-11 ***	-6 ***	-2
Unstable	-3 *	2	11 ***	6 ***	2
- / +	Sturdy / Frail	Strong / Weak	Smooth / Abrupt	Rigid / Supple	Tender / Insensitive
Kin. high	-15 ***	-20 ***	13 ***	-9 ***	13 ***
Kin. low	12 ***	17 ***	-14 ***	12 ***	-13 ***
Kin. medium	3 *	3 *	1	-2 *	0
Sequence A	-3	-3	-4	2	-1
Sequence B	9 ***	7 ***	0	3	-6 *
Sequence C	10 ***	9 ***	4	1	2
Sequence D	20 ***	18 ***	7 **	-5 *	2
Sequence E	-22 ***	-19 ***	-2	-2	6 *
Sequence F	-14 ***	-12 ***	-5 *	1	-2
Var. increment	-4 *	-2	-4 ***	3 *	-4
Var. saccade	27 ***	20 ***	15 ***	-9 ***	6 ***
Var. smooth	-24 ***	-18 ***	-10 ***	6 ***	-3
Eyes none	3	4 *	1	-1	6 ***
Eyes round	1	1	-7 ***	4 *	-14 ***
Eyes squint	-3	-6 **	6 ***	-3	8 ***
Stable	-16 ***	-11 ***	-4 ***	0	0
Unstable	16 ***	11 ***	4 ***	0	0

significant *and* have absolute values greater than $10pp$[5]. Before discussing the interpretation of the table, we note that the head rotation variable is not represented in the table given that the only significant effects were on the sturdy-frail scale, with contrasts $5pp$, and $-5pp$ for *straight* and *turn straight* values (both $p < 0.05$). All other contrasts for the head variable were smaller, mostly less than $3pp$.

Some general observations can be made with respect to the motion corpus variables and scales. Firstly, all of the corpus variables (rows) have effects greater than $10pp$ on several perceptual scales, suggesting that they are all relevant variables to be implemented in our navigation algorithm. Secondly, all of the scales (columns) have effects greater than $10pp$ related to several corpus variables, suggesting that the scales are relevant for studying the impact of robot motion and appearance. We propose to study the columns of Table 3.7 to outline which variables values have the most impact on each perceptual scale. For each scale we list the main variable contrasts in decreasing order of their absolute value. We discuss scales with similar contrast structures together.

The aggressive-gentle and authoritative-polite scales are mostly affected by the kinematics type and variant. Low kinematics and smooth or increment variants increase the probability of gentle and polite perception, whereas high kinematics and saccade variant increase the probability of aggressive and authoritative.

The confident-hesitant, sturdy-frail and strong-weak scales are mostly affected by the variant and the motion sequence, followed by the kinematics, and base stability. The saccade variant, motion sequences with hesitations (C and D), low kinematics and unstable base increase the probability of hesitant, frail, and weak perception. The smooth variant, longer motion sequences (E and F), high kinematics and stable base increase the probability of confident, sturdy, and strong perception.

The inspires confidence-doesn't inspire confidence scale is similar to the previous group of scales, although the absolute values of the contrasts are all smaller, suggesting the corpus variables do not have as much impact on this scale, or that it is more ambiguous for participants. Another difference is the inversion of the signs for the kinematics contrast.

The nice-disagreeable and tender-insensitive scales are mostly affected by the kinematics and eyes. Low kinematics and round eyes increase the probability of nice and tender perception, while high kinematics and squinting or absence of eyes increase the probability of disagreeable and insensitive.

The smooth-abrupt and rigid-supple scales are both mostly affected by kinematics and variant, and the contrasts are comparable especially if we invert either one of the scales so that smooth and supple have the same sign. Low kinematics and smooth variant increase the probability of smooth and supple perception, whereas high kinematics and saccade variant increase the probability of abrupt and rigid.

[5]This threshold on the percentage points is chosen arbitrarily to focus the discussion on the most impactful variables, it does not mean that the variable values with low contrasts can be ignored. To precisely determine how a full combination of corpus variables will be perceived, one should use the fitted logistic regression models to compute the appropriate prediction.

Summary

Similarly to the first online experiment, several participants commented about the difficulty in responding due to videos seeming similar or even identical, and mentioned answering at random on occasion. Despite this, all of the motion corpus variables except the head rotation had statistically significant effects on how participants perceived the robot. Many individual variables altered the average probability with which participants would choose either adjective of the perceptual scales by 10, 20, or even almost 30 percentage points. Our comparison of regression models both with and without interaction terms showed little difference between the models' performance, with the simpler model having slightly better performance.

When considering the influence of each corpus variable (see Table 3.7), some variables stand out. The kinematics and variant variables both have consistently large effects on every perceptual scale, mostly greater than $10pp$, and greater than $20pp$ for two to three scales. These are followed by the motion sequence variable with effects greater than $10pp$ on four scales, and the base stability and eyes with three and two scales greater than $10pp$, respectively. The head variable had little to no effect on any of the scales. These results strongly suggest that all of the motion corpus variables with significant effects on perception should be implemented into our navigation algorithm.

The correlation and clustering results suggest that some perceptions of socio-affects cannot easily be distinguished by using the corpus variables, such as aggressive-gentle and authoritative-polite. Given their high correlation, if our aim is to convey a gentle yet authoritative affect it seems that we would need to use other modalities than those explored in this study. On the other hand, these results also show us which scales seem to be independent with respect to changes in the corpus variables. Strong-weak and tender-insensitive show very little correlation and can be altered by using distinct corpus variables, suggesting it is possible to parameterize the robot for strong and tender perception, or weak and insensitive.

In addition to gaining an understanding of the direction and magnitude of the influence of the corpus variables on human perception, we were able to fit mixed effect logistic regression models which we can use to perform predictions for how a given combination of all the corpus variables may be perceived, either at the population level (averaged over participants) or for individual participants. One should however keep in mind that these models were based on online experiments exclusively. Despite our best efforts to film the motion corpus videos in a way that conserved as many details of the robot's motion as possible, it remains important to assess the impact of these motions in a physical, embodied experiment where participants are sharing their space with the mobile robot. We therefore designed a third experiment, which we present in the following section.

3.5.4 Embodied Experiment

The goal of the in-person experiment was to determine whether the effect of the motion corpus variables on participants' perception would be similar to the online experiments

when placed in an embodied interaction. Participants were informed of the general goal of the study prior to the experiment. Participants were asked to stand at a fixed position facing the robot as it moved towards them using one of the combinations of variables. The distance at which a mobile robot stops when approaching a person has been investigated, usually to determine what people consider as an acceptable distance (Brandl et al., 2016). In order to control for potential effects of different stopping distances, we ensured that the robot would always stop at a distance of $50cm$ from the person by using the same hard-coded velocity profiles as were used to film the video corpus. The experiment setup is shown in Figure 3.10.

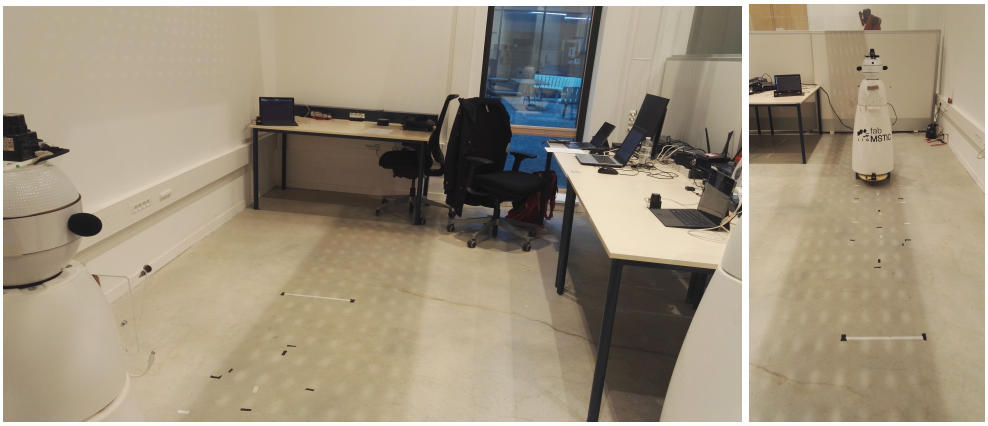


Figure 3.10: In-person lab condition experiment setup. Left: View from behind the robot. Right: view facing the robot. Participants stood with their feet on the white line.

We chose to focus on three variables, using two values for each: the kinematics type (low/high), variant (smooth/saccade), and head position (straight/side). The kinematics and variant variables were selected since they had the most influence on participant's responses in the second online experiment. The head position variable was selected since we aimed to test the hypothesis that gaze would be more influential in an embodied interaction. Each participant saw all combinations of values, resulting in 8 stimuli. The base variable was set to unstable, in order to make the saccade variant more visible; eyes were set to round in order to better indicate gaze direction, and the motion sequence was set to one of the shorter sequences (A) since it induced the least variability on the distance to the participant upon stopping ($\pm 5cm$). In addition, sequence A was the only one to have no significant impact on any of the scales in the second online experiment, hence being perceived as relatively neutral (see Table 3.7).

A total of $n = 22$ participants completed the embodied experiment, consisting of students recruited from our university, half of which were in the field of computer science and applied mathematics. The response distributions are shown in Figure 3.11. Once again, there are some strong contrasts for many of the scale and variable pairings, especially kinematics for Aggressive-Gentle and Authoritative-Polite, variant for Sturdy-Frail and Confident-Hesitant.

We adopt the same analysis method as for the first online experiment using chi-square association tests, and applying Bonferroni corrections to the p-values, reported in Table 3.8. Associations were found between most of the scales and the kinematics and variant

		Aggressive (-) Gentle (+)		Authoritative (-) Polite (+)		Condifent (-) Hesitant (+)		Inspires conf. (-) Doesn't insp. (+)		Kind (-) Disagreeable (+)	
		-	+	-	+	-	+	-	+	-	+
Kinematics	high	85	15	78	22	70	30	31	69	33	67
	low	15	85	14	86	39	61	52	48	84	16
Variant	smooth	42	58	45	55	83	17	60	40	66	34
	saccade	58	42	47	53	26	74	23	77	51	49
Head	straight	47	53	45	55	55	45	45	55	65	35
	side	53	47	47	53	55	45	38	63	52	48

		Sturdy (-) Frail (+)		Strong (-) Weak (+)		Smooth (-) Abrupt (+)		Rigid (-) Supple(+)		Tender (-) Insensitive (+)	
		-	+	-	+	-	+	-	+	-	+
Kinematics	high	61	39	76	24	33	67	80	20	25	75
	low	49	51	36	64	68	32	49	51	67	33
Variant	smooth	82	18	73	27	73	27	57	43	53	47
	saccade	28	72	40	60	28	72	72	28	39	61
Head	straight	56	44	59	41	53	47	59	41	52	48
	side	55	45	53	47	48	52	69	31	40	60

<= 20% <= 30% < 40% > 60% >= 70% >= 80%

Figure 3.11: Response distributions in percentages for the embodied, in-person experiment. Columns represent response levels for each perceptual scale. Rows represent corpus variable values by which the stimuli ratings are grouped to compute the percentage of responses.

variables ($\chi^2(1) \geq 18, p < 0.001$). As in the second online experiment, no association was found between the head variable and any of the scales. The four most significant dependencies were Aggressive-Gentle on kinematics ($\chi^2(1) = 87$), Sturdy-Frail on variant ($\chi^2(1) = 51$), Authoritative-Polite on kinematics ($\chi^2(1) = 74$), and Confident-Doubtful on variant ($\chi^2(1) = 57$), all with $p < 0.001$. Some differences can be observed with respect to the second experiment results for the kinematics and variant corpus variables. In the second online experiment, the kinematics variable had significant impact on all scales whereas in the embodied experiment, the effect on the Inspires-Doesn't inspire confidence scale is not significant. The variant also has fewer statistically significant associations. The values for which we observe changes seem to be those which did not have a large effect on the response probability in the second online experiment.

The hypothesis that gaze would have an influence in the in-person experiment is not supported. While the results of the embodied experiment are not directly comparable to the online experiments, the stronger associations between motion corpus variables are conserved across the online and in-person experiments for the most part.

Table 3.8: Chi-square association test results between a subset of corpus variables and all perceptual scales for the embodied, in-person experiment. Reported as chi-square statistic, significance $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

	Aggressive Gentle	Authoritative Polite	Confident Doubtful	Inspires Conf Does not	Nice Disagreeable
Kinematics (1)	87 ***	74 ***	18 ***	n.s.	47 ***
Variant (1)	n.s.	n.s.	57 ***	25 ***	n.s.
Head (1)	n.s.	n.s.	n.s.	n.s.	n.s.

	Sturdy Frail	Strong Weak	Smooth Abrupt	Rigid Supple	Tender Insensitive
Kinematics (1)	n.s.	28 ***	22 ***	18 ***	31 ***
Variant (1)	51 ***	19 ***	35 ***	n.s.	n.s.
Head (1)	n.s.	n.s.	n.s.	n.s.	n.s.

3.6 Discussion

3.6.1 Limitations

A first limitation of this study is that while the second online experiment analysis does provide us with estimations for how much various robot parameterizations can alter the probability of it being perceived in a certain way, we do not have any measure of the magnitude of the perception. The forced choice format of the second online experiment was chosen specifically to force people into selecting either end of the scale rather than responding at the neutral level as in the first experiment, and the results seem to indicate that for most scales and most corpus variables the response distributions showed clear effects. It may be worthwhile to provide more freedom of responses in future experiments, for example using continuous sliders for the scales rather than binary or likert responses.

Secondly, our corpus videos show a robot moving in an empty environment without any interaction with a person, whereas one could include a form of interaction with a person, given we are studying HRI. This decision was taken since our first goal is to isolate the physical navigation primitives from other factors that impact interaction. We hypothesize that in addition to the physical properties of motion, the *relation* between the robot and the person also plays a role in how a robot is perceived, and how we interact with it. If we impose a relation by framing a specific type of interaction in the experiments, it would be difficult to analyze whether a person's reaction to the robot was induced by motion and appearance primitives or by the relation.

3.6.2 Implications

Our results show that the impressionistic adjectives used in our perceptual scales are useful in aiding participants to distinguish and characterize various elements of movement prosody in robot navigation. Furthermore, the typology of variables proposed in the corpus can be used to establish socio-affective traits of expressive navigation which exhibit significant contrasts between one-another. As such, even if a given type of motion was chosen solely on the basis of practical considerations, our work suggests that it will be perceived and interpreted by humans as socio-affective expression, therefore impacting HRI. Thus, it is essential to take into account the fact that navigating intrinsically entails communicating with the human. To achieve this, we must understand and control what types of navigation profiles should be used to generate elements of interaction, whose communicative and ethical effects also require further rigorous study.

One of our perceptual scales is based on the concept of frailty, which has already been found to have significant impacts on the people interacting with a frail robot. In a prior work (Sasa & Aubergé, 2016), isolated elderly people interacted with a small butler robot by giving it voice commands. During the experiment the authors discovered by serendipity that when the robot showed signs of frailty by making mistakes (bumping into a wall while moving), participants became more attached and changed their attitude towards the robot by starting to *take care* of the robot. Similarly, in (Matsumoto, 2021) the authors compared a typical robot to a fragile robot which broke, requiring participants to fix it; finding that participants reported feeling more attached to the fragile robot, as well as finding it more pleasant, less boring, and more interesting. The interest of this is not so much that participants are attached, but rather that because they are attached, they tend to be more active in their interactions with the robot, often helping it, which may have positive effects on the person's physical and mental health when compared to passively receiving care (Takenaka, 2005; Tanaka, 1997). As such, the impact of the impressions generated by robot navigation variables seems to go far beyond the issues of user preference, usability or comfort.

3.7 Conclusion

In this chapter, we proposed an incremental bottom-up approach to the understanding of how fundamental properties of appearance and navigation impact a person's perception of a mobile robot. We constructed a novel holistic robot motion corpus in order to study the impact of navigation *and* audio-visual cues on people's perceptions of robots, in contrast to the more specialized studies of previous works. The variables contained in the corpus are hypothesized to be involved in what we define as *movement prosody*, a concept we derive by analogy with vocal prosody. The corpus was used in two online perception experiments ($n = 42$, $n = 101$) and one in-person experiment ($n = 22$). Participants rated a robot performing a navigation task along ten perceptual scales opposing adjectives describing physical aspects as well as perceived intentions and attitudes of the robot. A statistical analysis of the dependencies between each variable and scale showed that all scales had significant dependencies on several corpus variables, and most corpus variables impacted several scales. This includes variables related to the robot's navigation such as its max-

imal velocity, acceleration, smoothness, pauses and hesitations. These results show that this experimental methodology can bring some insights into people's perception of mobile robots, and more generally, how humans process cues from various modalities in order to build their perception of an agent.

The analysis of the effects of the motion corpus variables demonstrated that all of the variables directly involved in the robot's motion (kinematics type, variant, and motion sequence) impacted participants' perception of the robot. In addition, all of the values taken by these variables affected perceptions in different manners. This leads us to conclude that our navigation algorithm should be designed in such a way that it can produce motions which are representative of the velocity profiles resulting from the combinations of these variables. The algorithm should be able to produce motions at different velocities and acceleration rates to model the kinematics types, different temporal sequences of accelerations and decelerations, including pauses and hesitations for the motion sequences, and lastly different motion qualities such as smooth linear accelerations, jerky, saccadic motion, or incremental acceleration for the variants. It should enable the robot to produce motion that conserves the defining characteristics of all the values of the motion sequence, variant and kinematics corpus variables. The design and implementation of such a navigation algorithm is the topic of the following chapters.

Chapter 4

Designing a local navigation algorithm parameterized by movement prosody

4.1 Introduction

In the previous chapter we established that people perceive a mobile robot in different ways according to various physical motion characteristics. More specifically, a corpus of velocity profiles for executing straight-line motions was constructed by combining different values of three motion variables called the *kinematics type*, *motion sequence*, and *variant*. These variables all had statistically significant effects on people's attribution of social attitudes and physical qualities to the mobile robot. The different combinations of values of these corpus variables represent different styles of **movement prosody**, (by analogy with vocal prosody), in the sense that variations in the velocity profile induced by changing the corpus variables can convey social and relational meaning, rather than simply being functionally different ways of moving the robot from A to B. Each combination of variables defines a unique velocity profile in the corpus, meaning the distance covered by executing a given profile is fixed. Our goal is to use the corpus profiles as a starting point to enable arbitrary distances to be travelled in dynamic environments, *while* maintaining a consistent movement prosody.

We propose to design a social navigation algorithm which can be parameterized to enable the robot to plan and execute trajectories which have the same prosody properties as the various corpus profiles. The algorithm should provide accurate control over the robot's motion, even when operating in dynamic environments. By changing which combination of corpus variable values the algorithm uses, we change how the robot is perceived by people. This ability to change the robot's movement prosody while navigating autonomously can enable further experimentation to understand the impact of different movement prosody in more varied scenarios where pre-programmed motion would be inadequate. It may also help address issues with the acceptance of social navigation algorithms by enabling control over the social perception of the robot by humans.

In section 4.2, we review approaches to social navigation and functional expressive motion generation which share similar goals of altering the manner in which a robot motion task is performed. Although the motion characteristics and the aspects of people's perception they study are different to those we consider, we conclude that the trajectory optimization framework adopted by such works is also suitable for our problem, due to the flexibility provided by formulating specific constraints to shape the robot's trajectories.

In section 4.3, we briefly present the navigation algorithm proposed in the author's master's thesis (published at ICRA 2020 (Scales et al., 2020)). We discuss the difference in the goals and design of the algorithms, and describe how the algorithm presented in this chapter significantly extends the previous algorithm. Most notably, the control provided over the robot's motion was limited, for similar reasons to existing navigation approaches which also use cost functions to define the robot's motion characteristics.

In the following sections, we derive our algorithm by starting from a simple problem formulation consisting of reproducing the original fixed-length velocity profiles in static environments, and gradually adding complexity in order to arrive at our final algorithm.

In section 4.4 we describe a solution for executing the original corpus velocity profiles when filming the video corpus, which allows us to introduce the formalization of the profile

representation. We then explain how planning a profile can be formulated as a constrained trajectory optimization problem, and motivate the need for specific prosody constraints to ensure the planned trajectories retain the correct motion characteristics.

In section 4.5 we address the problem of planning and executing prosody-compliant trajectories with arbitrary distance in a static environment. We discuss which parts of the profiles should be constrained, and which should be allowed to change. We then formulate prosody constraints for each motion corpus variable which restrict the planned trajectories so that they exhibit the correct properties. We present our algorithm to plan the trajectory by solving the optimization problem subject to the prosody constraints, as well as the control algorithm to execute the plan.

In section 4.6 we extend our algorithm to handle dynamic environments which may cause the initial trajectory plan to become invalid due to the presence of dynamic obstacles, or if the goal position changes, for example when following or moving towards a person. We adopt a receding-horizon control approach which integrates planning into the control loop by re-planning the trajectory based on updated information from the environment. We discuss the impact of this approach on the ability to maintain consistent prosody in the robot's motion, and propose new prosody constraints that account for the continuous re-planning. The updated control algorithm which includes the planning step is presented.

Section 4.7 addresses cases where the planner is unable to find a solution that satisfies all of the constraints, which is a more common occurrence than in typical social navigation algorithms due to our very restrictive prosody constraints. We first discuss how accurate modelling of delays and latency in the robot architecture can eliminate some of such cases. We then present a constraint hierarchy approach that allows specific constraints to be relaxed to attempt to find a plan that partially satisfies the prosody constraints. We then present the final iteration of our control and planning algorithm.

In section 4.8 we discuss the limitations of our approach as well as possible extensions, before concluding in section 4.9.

4.2 Existing approaches for integrating expressivity or style in robot motion generation

The motion corpus variables evaluated in our perception studies alter the velocity profiles of the robot in ways which are sometimes quite subtle. In this section, we discuss how optimization-based approaches for social navigation and expressive motion shape the trajectories according to the features which are relevant for social compliance or expressivity. We then motivate our approach of formulating precise and very restrictive features forming a trajectory space which is as close as possible to our motion corpus profiles.

Works in the field of Social Navigation (Mavrogiannis et al., 2023) typically concern themselves with enabling mobile robots to navigate in complex environments (Vega et al., 2019), around many (potentially dynamic) pedestrians (Henderson & Ngo, 2021), and modelling uncertainty of surrounding pedestrian motion (Kollmitz et al., 2015). Many works

formulate social navigation in the framework of trajectory optimization due to the flexible specification of the robot's behaviour which can be achieved by carefully designing the cost function to be optimized and the constraints. In (Khambhaita & Alami, 2020) a joint optimization of the robot and human's trajectories is performed over a cost function encoding various social aspects, such as penalizing low time to collision values and penalizing high velocity near humans. Social navigation approaches tend to focus on ensuring the robot plans safe, comfortable and natural motion, following the definitions in (Kruse et al., 2013). The goal pursued by such approaches is different to ours since we aim to generate motions that induce different social perceptions off the robot's attitude towards others, as well as different physical perceptions.

Recent works in expressive motion generation for robots have increasingly been targeted at not only generating a movement that is perceived as manifesting various emotions, affects, or internal states of the robot, but also performing a practical task at the same time (see (Venture & Kulić, 2019) for a review). Some of the more recent approaches also make use of the trajectory optimization problem formulation, encoding the expressive features into the cost function, although these approaches are mostly applied to different robot embodiments such as humanoids or manipulator arms. In (Zhou & Dragan, 2018) the authors explore how to generate sad, happy, and hesitant motions on a robotic arm. The costs for each feature are linearly combined along with a cost encoding the robot arm's motion task to form the cost function. Although these more recent works use trajectory optimization approaches, they typically only study the capacity of the algorithms to generate appropriate motions in static environments.

Trajectory optimization approaches are common both in functional expressive motion generation works, as well as in social navigation works. In order toExisting approaches tend to either enforce that the primary task performance should not be altered by expressive or social factors (Hagane & Venture, 2022), or apply expressive or social features via the cost function (Khambhaita & Alami, 2020; Repiso et al., 2017; Zhou & Dragan, 2018). The cost functions are constructed from several cost terms, of which there can be many when the robot is supposed to address complex navigation scenarios (social norms, legibiltiy, proxemics, motion task specification). Using the cost function to encode social or expressive features in addition to the task turns the optimization problem to a multi-objective problem, however multi-objective optimization methods which determine the pareto front of solutions are still often too slow for real-time deployment. Instead, the problem is often handled by converting the problem to a single-objective problem by scalarizing the cost which is expressed as a weighted sum of cost terms. Tuning the weights adjusts the relative importance of costs, and requires trial and error to determine a weighting that provides the desired trade-off between the different cost terms. In dynamic environments, situations may arise where several trajectories have identical total cost distributed differently across each term. For example, the ideal path which struck the desired balance between task execution and social or expressive costs may become infeasible due to the environment's configuration, meaning the remaining space of feasible trajectories will sacrifice either task performance or social and expressive features.

In our work, the movement prosody features have been found to generate social and affective perceptions of the robot, hence we believe it is crucial that the robot's motion stays

consistent in its expression of prosody. We aim to generate robot motions which conserve the features of the corpus velocity profiles with as much detail as possible. For example the difference between different motion sequences can boil down to the presence or absence of a short (300ms) constant velocity phase placed between an acceleration and deceleration, and generating motions that capture the smooth variant relies on being able to produce stable velocity commands over time, despite being computed online, with noisy sensor data, and in dynamic environments where frequent re-planning of trajectories is necessary. We take the position that movement prosody should be kept consistent as much as possible given its role in human's perceptions of robots. In order to avoid ambiguous perceptions of the robot due to cost function tuning and trade-offs, we formulate the prosody features as hard constraints in the optimization problem, ensuring consistent prosody across the future planned trajectory **and** with respect to the robot's past motions. Using constraints will restrict the trajectory solution space, potentially degrading task performance, but our priority is for the motion to accurately correspond to the motions which were evaluated in the perception studies.

4.3 Previous work

In this section, we give an overview of the algorithm developed during the author's master's thesis, and discuss how the algorithm presented in this chapter extends it. For details, we refer the reader to the corresponding publication at ICRA (Scales et al., 2020).

4.3.1 Overview

The goal of the previous work was to design a navigation algorithm which would be flexible, in that it should be possible to parameterize it to perform different tasks, and also provide accurate control over the robot's motion during navigation. Such a flexible algorithm could enable studying the effect of various kinds of navigation on people. We focused on studying the navigation task of person following, since it inherently involves spatial interaction with a person. In existing works, we noticed that the robot's following position with respect to the person was usually chosen empirically, or left as a design decision. We chose to study the effect of different robot positioning on how humans behaved when navigating with the robot.

To design the algorithm, we took inspiration from two existing works, which both used cost functions to specify the robot's behaviour, each with their own advantages and drawbacks. In (Morales Saiki et al., 2012), the authors proposed to perform planning in the joint person-robot state space, simultaneously planning for what the person and robot should do together, as opposed to the more classical approach of first observing or predicting the person's position, and subsequently planning the robot's motion. The robot and person had their own cost functions, which the planner jointly optimized to determine which position in space the robot should move towards within a short time horizon of $2s$. This is interesting since it allows to computationally model the fact that the robot's actions may influence the human's actions. The cost function included terms which encoded a preferred velocity

and acceleration value for the robot, in addition to terms guiding the robot towards the specified position next to the person. A separate trajectory following algorithm was used in order to control the robot to move towards the optimal position resulting from the planning step. Instead, we take inspiration from the integrated planning and control approach of (Park, 2016), planning in the space defined by the robot's control inputs (linear and angular velocity), which allows us to directly use the output of the planner to control the robot. We also proposed to enable the weights of the cost function terms to be dynamically adjusted, which we used to specify the robot's following position.

This algorithm was implemented on the RobAIR robot, and deployed in experiments where the robot followed people side by side, or behind them as they were asked to go to a goal location. We conducted experiments in lab and ecological settings. In the lab setting experiment (Figure 4.1) participants were told how the robot would position itself, and were not affected by the robot's positioning. The algorithm enabled the robot to follow people accurately in both configurations. In the ecological experiment there were no issues when the robot was configured to follow behind people. However, when attempting to position itself to the person's right, participants would start to also move right, deviating from the path to the goal, and causing the robot to compensate and turn further right. When observing participant behaviour and from their comments after the experiment, it seems that they thought the robot had lost track of them, and they did not realise it was trying to follow by their side.

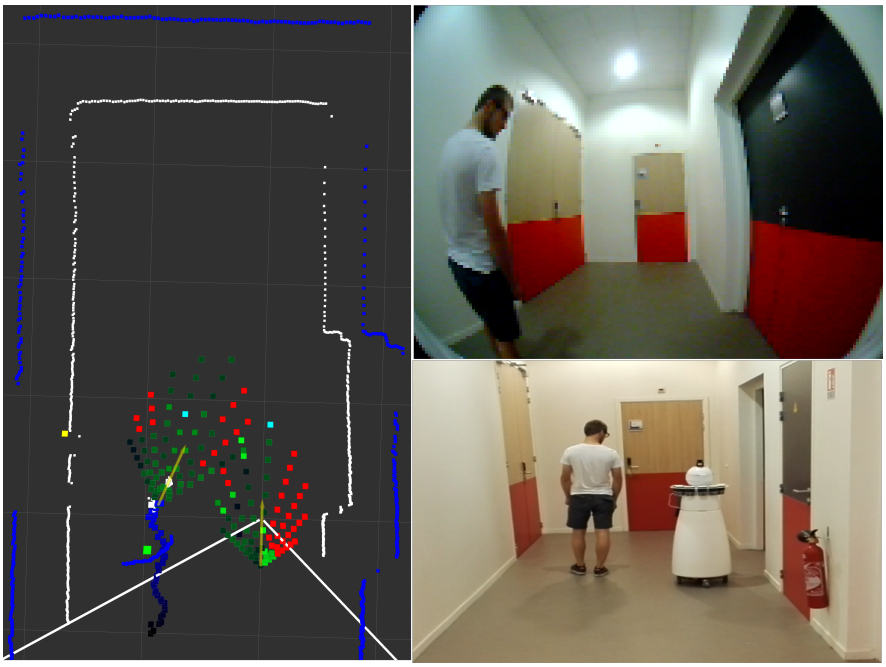


Figure 4.1: Snapshot from a lab-setting experiment from our person-following work (Scales et al., 2020). Left: visualisation of the joint person-robot planning. The position resulting from applying the optimal control is marked in light blue. Right: onboard and external views.

4.3.2 Differences with the algorithm developed during the PhD

The previous algorithm was designed to enable flexible control over the robot's motion characteristics, however it only used the cost function to control the relative position of the robot. In this thesis, we aim to control the overall style of movement in such a way that it is coherent through time, even when the robot is operating in dynamic environments. Furthermore, as with other navigation algorithms using several cost function objectives to shape the motion, our previous algorithm was difficult to tune, especially when objectives were conflicting. In this thesis, we propose a different control parameterization as well as many novel constraints that allow separation of the movement style from the task influence. We also propose a method to gracefully mitigate situations where the planner is unable to find a solution, which was lacking in our prior algorithm. Lastly, our previous algorithm only planned one control input, which was assumed to be held constant over the next two seconds. In this thesis, we require an algorithm that plans over several control inputs while considering the history of past motions, in order to replicate the motion characteristics studied in chapter 3.

4.4 Fixed distance corpus profiles and problem formulation

In this section, we first introduce the notation which will be used to describe the velocity profiles, and explain how the robot's motion is controlled when executing the fixed distance velocity profiles from the corpus. Then, we present a generic trajectory optimization problem formulation which would allow us to plan trajectories over arbitrary distances, and point out the necessity for prosody-specific constraints in order to ensure the resulting trajectory conserves the characteristics of the corpus motion variables.

4.4.1 Representation of corpus profiles

The corpus profiles were constructed by selecting the combination of values for three variables: the motion sequence, kinematics type, and variant. Figure 4.2 represents how each of these variables alters the shape of the corpus velocity profile. The motion sequence determines whether accelerations should be immediately followed by a pause (i.e. a short constant velocity phase), and whether the robot should perform a hesitation (i.e. slow down and accelerate again rather than maintaining a constant velocity). The kinematics type controls the acceleration value (i.e. the slope of the velocity profile) and the maximum velocity. The variant controls the smoothness and stepping of the velocity profile.

In Figure 4.3, we give an example of how a corpus velocity profile can be represented as a sequence $U = \{u_0, u_1 ... u_{N-1}\}$ of N motion phases, where $u_k = [a_k, t_k]$. A motion phase u_k consists of the slope of the velocity profile (acceleration) a_k, and a duration t_k over which the acceleration is applied. In conjunction with an initial position x_0 along the robot's forward axis, and initial linear velocity v_0, these values define the robot's trajectory in

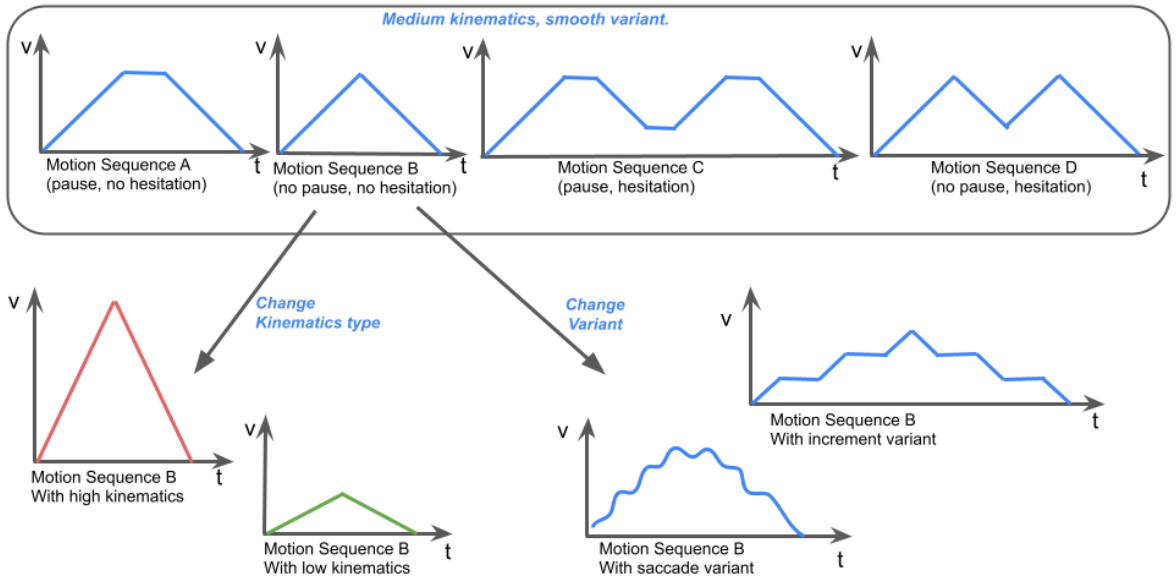


Figure 4.2: Illustration of the construction of the velocity profiles by combining the motion corpus variables. Top: all motion sequences represented with medium kinematics and smooth variant. Bottom: profiles resulting from applying different kinematics or variants to motion sequence B. In total, $4 * 3 * 3 = 36$ profiles can be obtained by combining the 4 motion sequences with 3 kinematics and 3 variants.

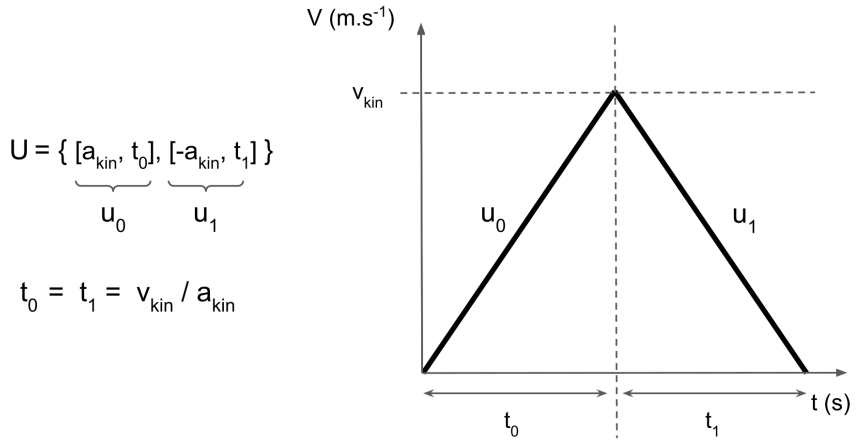


Figure 4.3: Representation of a corpus velocity profile using motion sequence B (no pauses, no hesitations) and the smooth variant as a sequence U of $N = 2$ motion phases u_0 and u_1. Values of v_{kin} and a_{kin} depend on the selected kinematics type (medium, low, high), and dictate the slope and maximum of the velocity profile.

space and time, and are related through the forward kinematics equation 4.1. This equation is simplified with respect to the full differential drive forward kinematics, since we do not control the angular velocity of the robot.

$$\begin{aligned} x_{k+1} &= x_k + v_k t_k + \frac{1}{2} a_k t_k^2 \\ v_{k+1} &= v_k + a_k t_k \end{aligned} \tag{4.1}$$

When recording the motion corpus, we simply select a given combination of motion variables (kinematics type, motion sequence, variant), defining the sequence of motion phases U. This sequence is used as input to a basic control loop which loops over each motion phase $u_k \in U$. Given a motion phase u_k, the controller sends an acceleration command equal to a_k to the low-level motor controller. After a time t_k has elapsed, the controller moves on to the next motion phase, sending the acceleration command a_{k+1}, until all motion phases of the sequence have been completed.

Since each combination of corpus variables uniquely defines the velocity profile, the distance travelled by the robot for a given type of movement prosody is fixed. For example, the profile using motion sequence B in Figure 4.3, using medium kinematics ($a_{kin} = a_{medium} = 0.35m.s^{-2}$, $v_{kin} = v_{medium} = 0.5m.s^{-1}$), results in $t_0 = t_1 = \frac{v_{kin}}{a_{kin}} = 1.428s$, meaning the profile makes the robot cover a distance equal to $a_{kin} * t_0^2 = 0.714m$. In order to change the distance travelled, we need to introduce some degrees of freedom back into the velocity profile, either through the acceleration values, motion phase duration, or adding more motion phases such as a constant velocity phase, and select the optimal values that minimize distance to the goal. This amounts to a trajectory optimization problem, and we now present a generic formulation using our notations for the velocity profiles.

4.4.2 Basic trajectory optimization formulation

Our goal is to build an algorithm that allows us to control the robot's translation acceleration such that it can move to a goal located in front of the robot at some arbitrary distance in a straight line, or in other words, only performing pure translation motions. We choose to focus only on the translation degree of freedom for simplicity, and also because our study of the robot's movement prosody was only performed on straight-line motions. Nevertheless, the principles we develop in this chapter could be transferred to the control of the robot's rotational acceleration once we develop our understanding of role of rotations or path shape in shaping people's perceptions of the robot.

Moving a robot towards a goal point while accounting for the robot's mechanical actuation limits can be cast as a discrete-time constrained minimization problem, where we optimize the sequence of control inputs $U = \{u_0, u_1 ... u_{N-1}\}$ such that the robot minimizes its distance to a goal position x_g. A control input $u_k = [a_k, t_k]$ corresponds to a motion phase, parameterized by a constant acceleration a_k and a duration t_k over which the acceleration is applied. The durations t_k take discrete values, $t_k = n * dt$, $n \in \mathbb{N}$, where dt is a constant determining the shortest possible control input duration. The state $\mathbf{x_k} = [x_k, v_k]$ of the robot is comprised of the robot's position along the x axis, and its linear velocity v. The control u_k affects the state $\mathbf{x_k}$ as described in the kinematics equation 4.1.

If t_k is unbounded, then the space of possible trajectories is infinite. A common approach is perform the trajectory optimization over a finite time T_h (Tedrake, 2023), where T_h is chosen to be long enough to enable the trajectory plan to cover the entire motion from the robot's initial position to the goal[1]. In our problem formulation the duration of a trajectory plan is determined by the sum of the control input durations, so to enforce a finite time horizon we introduce a constraint $\sum_{k=0}^{N-1} t_k = T_h$.

In this first formalization, we assume that the environment is static, and that there are no obstacles between the robot and the goal. The only constraints which act on the system are the physical limits of the motors, leading to upper and lower bounds on the accelerations and an upper bound on the velocity. We consider only positive translation velocities since our robot is not equipped with any rear-facing sensors, thus imposing a lower bound on the velocity. The resulting problem formulation is given in Equation 4.2.

$$\begin{aligned} \min_{u_0 \dots u_{N-1}} \quad & \sum_{k=0}^{N-1} ||x_g||^2 \\ \text{subject to:} \quad & \forall k \in \{0, 1 \dots N-1\}, 0 \leq v_k \leq v_{max} \\ & \forall k \in \{0, 1 \dots N-1\}, -a_{max} \leq a_k \leq a_{max} \\ & \sum_{k=0}^{N-1} t_k = T_h \end{aligned} \tag{4.2}$$

Solving this optimization problem would produce triangular or trapezoidal velocity pro-

[1]This arbitrary limit on the duration of the trajectory plan will be revisited more accurately when we convert the problem to a receding horizon control approach in section 4.6

files depending on the distance to be travelled. In our motion corpus, velocity profiles that use the saccade and increment variants, or hesitation and pause motion sequences are not purely trapezoidal or triangular and cannot be generated using this approach since they do not represent the optimal trajectory, e.g. hesitations introduce a deceleration in the middle of the motion, which increases the time taken to arrive at the goal position. The acceleration and maximal velocity values may also be different when compared to those associated with the three kinematics types.

In order to shape the trajectories produced by the optimization, we propose to modify the constraints such that they restrict the set of valid control sequences based on the values of the motion corpus variables. For example, the most straightforward corpus variable to account for would be the acceleration values enforced by the kinematics type, by restricting the acceleration a_k to take values from the finite set $\{-a_{kin}, 0, a_{kin}\}$.

The issue with simply applying new constraints until the optimization solutions exactly replicate the corpus profiles is that any given *combination* of the motion corpus values is only represented by one profile, meaning only one distance can be travelled with a given prosody parameterization. If we directly use only the exact velocity profiles from our corpus, the robot's initial distance to the goal would have to perfectly match the distance associated with a given corpus profile. Instead of using the corpus motion profiles as exact references, we require a means to relax some of the dimensions of the velocity profiles such that they can be adapted to different distances, which we discuss in the next section.

4.5 Extending the motion corpus profiles to arbitrary distances for offline trajectory optimization

In this section, we first discuss our proposal for adapting the original motion corpus profiles to longer or shorter distances while preserving their distinct characteristics. Then, we derive constraints which can be incorporated into a trajectory optimization scheme in order to constrain the solution space to motions which match a given set of motion corpus parameters. Lastly, we present an algorithm to plan variable distance, prosody compliant trajectories in an offline fashion, followed by open-loop control to execute the planned trajectory.

4.5.1 Adding flexibility to corpus profiles

In our corpus, a given combination of motion parameters generated a unique velocity profile resulting from the parameters' control over acceleration, peak velocity, and timing. Executing a given velocity profile results in the robot performing a unique trajectory in space and time, with a given length. In order to build a general navigation algorithm, we require a formulation of motion where the distance travelled is a free variable (altering the distance should not alter the impression generated by the robot). In other words, we want to transform the corpus profiles corresponding to a combination of parameters into a *class* of profiles which maintains as many characteristics of the original profiles as possible. While

adding more flexibility to the corpus velocity profiles, we must carefully consider whether it increases the chances for overlap and confusion between different parameter combinations. We keep the piecewise linear curve representation of the corpus profiles, given that using other functions might lead to different impressions. With these limits, changing the distance travelled by following a given velocity profile can be achieved by altering variables of the profile, each of which is already involved in the definition of the corpus profiles:

1. acceleration and maximum velocity (kinematics type);
2. successions of accelerations and decelerations (motion sequence and variant);
3. length of maximum velocity phase (motion sequence).

Ideally, another perception experiment would be performed with distance as one of the variables in order to study which of these modifications best preserves the mapping between motions and impressions. Conducting such an experiment with our methodology would require extending the corpus design, filming new videos, and running more online and in-person studies. Instead, in the following paragraphs we consider the impact of each of the modifications listed above, subsequently selecting the one that best preserves the original characteristics of the velocity profiles, and maximizes the distinctness of profiles using different parameter combinations.

Acceleration and maximum velocity

The acceleration value used for the slope of the velocity profile and the maximum velocity of the profile could each be changed to lengthen or shorten the distance travelled. However, both of these parameters are already constrained to precise values in the corpus by their role in defining the kinematics type parameter. In addition, altering the acceleration could lead to different kinematics types sharing identical motions given a distance to travel. This is problematic since the kinematics type was found to have an impact on all of our ten perceptual scales. Altering the maximal velocity without changing the acceleration value would preserve the distinctness of motions, but the maximal velocity is bounded by the robot's motors, so further modifications should be made to enable longer motions.

Alternating acceleration and deceleration

Another solution to alter distance traveled would be to perform successions of accelerations and decelerations. This would preserve the kinematics type and allow a given profile to be lengthened, but it doesn't provide a way to reduce profile length, meaning the robot would be incapable of performing short motions. Furthermore, the motion sequence and variant parameters of our corpus already define successions of accelerations and decelerations with motion sequences C and D, as well as the saccade and increment variant. Using this degree of freedom to alter the distance could lead to confusions between these corpus variables, which were also found to impact all of the perceptual scales.

Length of maximum velocity phase

The last solution is to modify how long the maximum velocity is maintained. This parameter is partially controlled in our original corpus by the motion sequence. The difference between motion sequences A, C and B, D is the introduction of pauses between acceleration and deceleration phases for sequences A and C, modeled as short ($300ms$) constant velocity phases. Extending the maximum velocity phase to increase distance would mostly preserve the distinctness of each parameter combination, except when the distance to be traveled requires a maximum velocity phase with a length similar to that of our pauses. Few differences were observed between the impressions generated by motion sequences with or without pauses, and the similarity incurred by changing the maximum velocity phase length only occurs for a relatively small subset of situations. Lengthening the maximum velocity phase does not allow shorter motions, hence we combine this with the lowering of the maximum velocity described previously. The transformations applied to alter the distance travelled when using motion sequences without hesitations is represented in Figure 4.4. For profiles using hesitations, there are several ways to add constant velocity phases to extend the motion, which we discuss in the next paragraph.

Adapting motion sequences

For some of the corpus profiles, there could be several ways to introduce constant velocity phases in order to lengthen a profile. Profiles using hesitation motion sequences (sequences C and D) make the robot slow down and accelerate back up to maximum velocity in the middle of the profile. One option to extend the profile would be repeating the slowing and accelerating motion in quick succession, however this would resemble profiles using the saccade variant. Instead, we chose to perform one slow-down and acceleration as soon as the maximal velocity is reached, followed by maintaining the maximal velocity. The slowing down pattern is repeated after a given time interval t_h, with more time elapsed between two hesitations than the length of the hesitation pattern.

Motion sequences A and C introduce the notion of pauses. We define pausing as always ensuring there is a constant velocity phase of length greater or equal to the pause length t_{pause} between an acceleration and deceleration phase. This means that shortening a pause profile conserves the same $300ms$ constant velocity phase, while reducing the value of the maximum velocity. Lengthening a pause profile is achieved by extending the constant velocity phase length. An issue arises when applying this procedure to profiles using both pauses and hesitations, since there are several constant velocity phases that could each be extended. We propose to extend only the last constant velocity phase, ensuring that the hesitation pattern occurs just after reaching the maximal velocity, and that its shape is preserved.

Adapting variants

Allowing the maximum velocity to be lower also raises the question of adapting the increment variant. A straightforward option would be to linearly scale the profile according

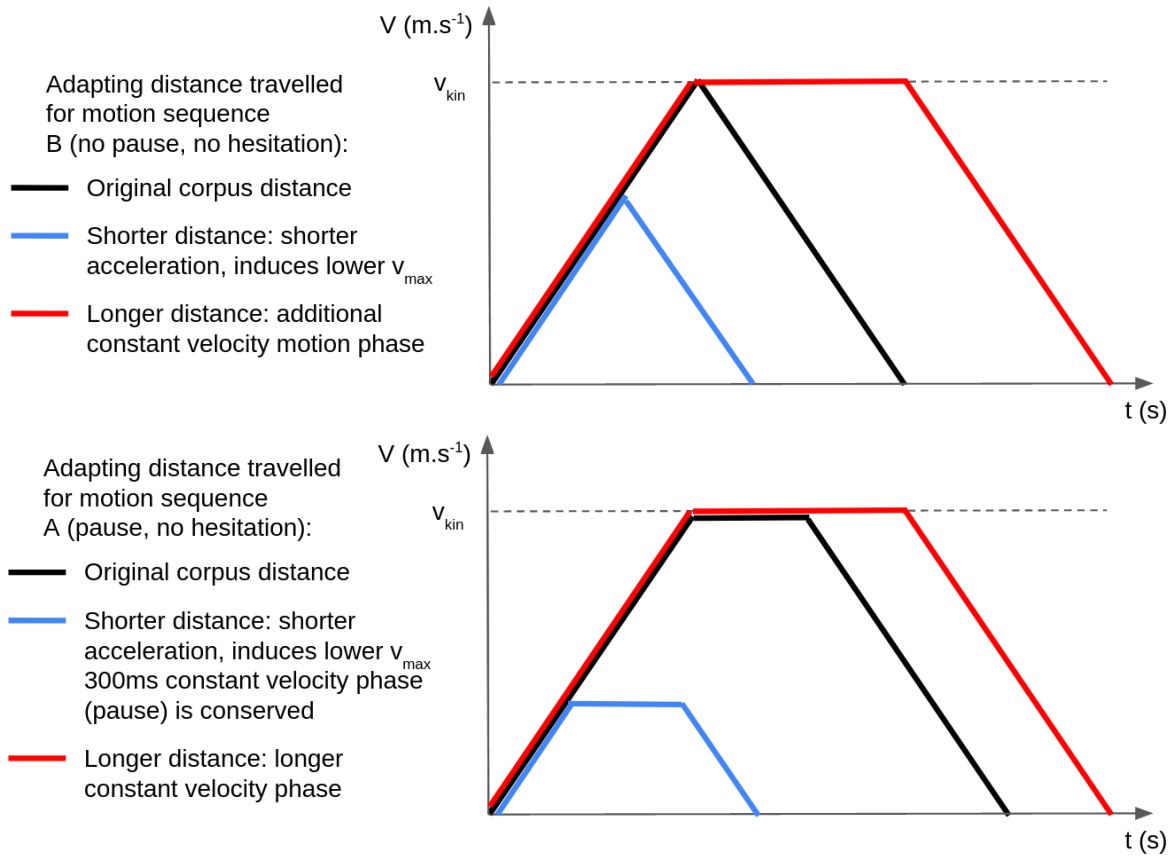


Figure 4.4: Illustration of the transformation of a corpus velocity profile to travel shorter or longer distances. Top: transformation for profiles without pauses or hesitations (sequence B). Bottom: transformation for profiles with pauses, and without hesitations (sequence A). When shortening profiles with pauses, we must conserve the short constant velocity phase representing the pause, even when performing shorter motions.

to the phase duration, altering the pause lengths and each acceleration length. We prefer instead to preserve the durations of the interleaved acceleration and pause phases, given they were chosen in order to be perceptible by humans. Also, linear scaling would have made a short incremental acceleration phase quite similar to a saccade acceleration, only with short interleaved accelerations and constant velocity phases rather than accelerations and decelerations. In the motion corpus profiles, the increment variant introduced two constant velocity phases into each acceleration and deceleration phase, separating them into three *increments*. All velocity profiles reached their maximum velocity v_{kin} given by the kinematics type, hence the increment length was always the same. In order to adapt the increments to variable distances we propose to simply use the same velocity profile as the corpus and allow the robot to interrupt the acceleration at any point in the profile. In other words, the constant velocity phases are inserted whenever the robot's velocity reaches one of the pre-determined increment velocities: $v_{inc\_low} = 1/3v_{kin}$ or $v_{inc\_high} = 2/3v_{kin}$. This simple approach may introduce a short stuttering motion when planning motions with given lengths. Alternatives which could be explored in further studies include altering the number of increments or the increment pause length based on the distance to be travelled, to remove such edge cases that introduce stutters. There remains a trade-off for accelerations which are too short to enable the insertion of an increment, resulting in an identical velocity profile to a smooth variant acceleration.

4.5.2 Variable distance prosody constraint formalization

In the previous subsection, we discussed the various ways in which the original corpus velocity profiles could be altered in order to handle arbitrary distances. We concluded that in order to perform shorter motions the profile's maximum velocity would be lowered, and for longer motions we introduce a variable length constant velocity phase at the maximum velocity. The resulting trajectory space gives enough flexibility to perform straight-line point-to-point motion. In this subsection, we formalize constraints which model each of the motion corpus prosody parameters, such that a trajectory which motion phases satisfy the constraint is representative of the corresponding motion corpus parameter value.

Integration of motion sequences

The corpus defined six motion sequences denoted A through F. We do not consider sequences E and F in our trajectory generation since they are truncated versions of sequence A (E and F velocity profiles depicted the robot accelerating and maintaining maximum velocity, or maintaining maximum velocity, then decelerating). The four remaining sequences represent the possible combinations of two concepts: pauses and hesitations. Pauses are used in sequences A and C, and hesitations are used in sequences C and D.

Trajectories using **pause motion sequences** (i.e. sequences A or C) require that an acceleration or deceleration phase $a_{k-1} \neq 0$ should be followed by a constant velocity phase $a_k = 0$ with a duration t_k greater or equal to the pause length $t_{pause} = 300ms$. This constraint is expressed in Equation 4.3, in such a way that it describes what should not occur: if the previous phase is not a constant velocity phase, and the current phase is not

the same acceleration as the previous, and the current phase is not a constant velocity phase as long or longer than a pause, then this trajectory does not satisfy the pause constraint.

$$\begin{aligned} PauseConstraint(U) \leftrightarrow & \forall k \in [1, N-1], \\ & \neg\big(a_{k-1} \neq 0 \wedge a_k \neq a_{k-1} \wedge \neg(a_k = 0 \wedge t_k \geq t_{pause})\big) \end{aligned} \tag{4.3}$$

Trajectories using **hesitation motion sequences** (i.e. sequences C or D) incorporate a deceleration from the current velocity down to some lower velocity, followed by the opposite acceleration, both with duration t_h. This hesitation should occur once immediately after the end of an acceleration phase, and then at regular time intervals $t_{h\_interval}$ along the trajectory. Therefore, the application of motion sequences is achieved by defining constraints on the possible ordering of motion phases within a trajectory. First, we impose that an acceleration phase should be followed by a hesitation deceleration phase. Since the second part of a hesitation is itself an acceleration phase, this constraint would make the robot perform successive decelerations and accelerations indefinitely. We need to distinguish normal acceleration phases from those that constitute a hesitation phase. In order to achieve this, we introduce an additional variable into the robot's state indicating the type of the motion phase: $type_k \in \{normal, hesitation, pause, increment\}$. By checking the type of the phase, we can enforce that only a normal acceleration which is not part of a hesitation sequence should be followed by a hesitation deceleration. Second, we impose that a hesitation deceleration should be inserted once the required time since the previous hesitation sequence has elapsed. In order to keep track of the time since the last hesitation, we introduce another state variable t_{since_hesit}, which is set to zero whenever a hesitation is performed, and incremented as time passes. These two situations are those in which a hesitation deceleration should be performed, and are formalized in Equation 4.4. A hesitation acceleration phase should be planned immediately after a hesitation deceleration phase, as expressed in Equation 4.5. Both of these constraints combined allow us to enforce the hesitation motion sequence (Equation 4.6).

$$\begin{aligned} HesitationDeceleration(U) \leftrightarrow & \forall k \in [1, N-1], \\ & (a_{k-1} = a_{kin} \vee t_{since\_hesit} \geq t_{h\_interval}) \\ & \wedge \neg(a_k = -a_{kin} \wedge t_k = t_h) \end{aligned} \tag{4.4}$$

$$\begin{aligned} HesitationAcceleration(U) \leftrightarrow & \forall k \in [1, N-1], \\ & (a_{k-1} = -a_{kin} \wedge type_{k-1} = hesitation) \\ & \wedge \neg(a_k = a_{kin} \wedge t_k = t_h) \end{aligned} \tag{4.5}$$

$$\begin{aligned} HesitationConstraint(U) \leftrightarrow & \neg HesitationDeceleration(U) \\ & \wedge \neg HesitationAcceleration(U) \end{aligned} \tag{4.6}$$

The previous constraint formulation is valid only when we are not using the pause motion sequence in combination with hesitations. When using the pause and hesitation sequences together the result should be that after an acceleration, first a pause phase is

performed, and then the hesitation deceleration. In order to address the issue, the hesitation constraint takes on different formulations depending on whether or not the pause constraint is active. If pauses and hesitations are active simultaneously, the pause constraints impose a pause phase after any acceleration or deceleration. Hence, we modify the constraints, requiring the hesitation deceleration to take place only after pause phases which follow a normal acceleration (Equation 4.7), and hesitation acceleration to take place only after pause phases which follow a hesitation deceleration (Equation 4.8).

$$\begin{aligned} &PauseHesitationDeceleration(U) \leftrightarrow \forall k \in [1, N-1]\,, \\ &((a_{k-2} = a_{kin} \wedge type_{k-2} \neq hesitation \wedge type_{k-1} = pause) \\ &\vee t_{since\_hesit} \geq t_{h\_interval}) \\ &\wedge \neg(a_k = -a_{kin} \wedge t_k = t_h) \end{aligned} \tag{4.7}$$

$$\begin{aligned} &PauseHesitationAcceleration(U) \leftrightarrow \forall k \in [1, N-1]\,, \\ &((a_{k-2} = -a_{kin} \wedge type_{k-2} = hesitation \wedge type_{k-1} = pause) \\ &\wedge \neg(a_k = a_{kin} \wedge t_k = t_h) \end{aligned} \tag{4.8}$$

Integration of variants

A trajectory using the **smooth variant** should result in acceleration and deceleration phases longer than a given minimal duration t_{smooth}, such that the trajectory does not resemble the saccade variant. We simply implement a lower bound constraint on the length of motion phases $t_{smooth} = 300ms$ (Equation 4.9). By applying this definition of the smooth variant, we are also limiting the robot's ability to perform short motions which would require an acceleration and deceleration with shorter phase lengths. If instead we decide that such short motions should be considered valid smooth motions, the constraints could be modified to allow short two-phase trajectories if they start and end at zero velocity.

$$SmoothConstraint(U) \leftrightarrow \forall k \in [0, N-1]\,, t_k >= t_{smooth} \tag{4.9}$$

The **increment variant** requires acceleration phases to be split into increments, such that the robot performs a constant velocity phase of duration $t_{pause} = 300ms$ when reaching certain velocities which are multiples of $v_{increment} = \frac{1}{3} stoppingTime(v_{kin}, a_{kin})$ (see Figure 4.7). The first part of the constraint (Equation 4.10) enforces that all acceleration or deceleration phases should end at one of the increment velocities. The second part of the constraint enforces that all acceleration and deceleration phases must be followed either by a pause phase, or by their opposite phase (Equation 4.11), i.e. an acceleration or deceleration phase cannot be extended, since it would violate the first constraint. The increment constraint is expressed by combining these two conditions in equation 4.12.

$$\begin{aligned} ValidVelocity(U) \leftrightarrow &\forall k \in [0, N-1]\,, \\ &v_k = i * v_{increment}, i \in \mathbb{N} \end{aligned} \tag{4.10}$$

$$\begin{aligned} BreakAccelerationPhase(U) \leftrightarrow & \forall k \in [1, N-1], \\ & a_{k-1} \neq 0 \\ & \wedge ((a_k = 0 \wedge t_k = t_{pause}) \vee a_k = -a_{k-1}) \end{aligned} \tag{4.11}$$

$$\begin{aligned} IncrementConstraint(U) \leftrightarrow & ValidVelocity(U) \\ & \wedge BreakAccelerationPhase(U) \end{aligned} \tag{4.12}$$

The **saccade variant** differs from the other prosody variables, since we do not formalize it as a constraint in the optimization problem, but rather as a post-processing step. In order to generate a velocity profile resembling the saccade variant, we can simply add oscillations generated by a triangular wave function around the velocity profile obtained by planning under the smooth variant constraint. In our motion corpus, the oscillation of the velocity signal has a high frequency and low amplitude, since the aim of this variant is to reproduce stuttering or shaking. Although it would be possible to formulate these oscillations as constraints on the motion phases we chose not to, since it would require a large number of motion phases leading to a large computation time. It would also tie the saccade frequency to the planner update rate and time discretization dt. Instead, we make the assumption that adding oscillations with small amplitudes and high frequencies will not affect the validity of the planned trajectory. In order to plan saccade variant trajectories, we plan under the smooth variant constraint. As the acceleration command computed by our algorithm is sent to the low-level motor controller, a time-varying offset given by a triangular wave is added. We use a period $\pi = 0.02s$, and an amplitude dependant on the kinematics type: $A\_low = 0.02m.s^{-2}$, $A\_medium = 0.05m.s^{-2}$, $A\_high = 0.07m.s^{-2}$. The period and amplitudes were empirically tuned so that they provide visually similar saccades as those implemented in the motion corpus videos, despite the differences in control update frequency[2].

Integration of kinematics types

The kinematics type specifies an acceleration value, or in other words, the slope of the velocity profile in acceleration and deceleration phases. When a kinematics type is specified, the robot must accelerate using that specific value. One could think of this as a constraint on the space of control inputs u_k of the robot, or a constraint on the set of admissible trajectories. We constrain the values of the accelerations such that they are either zero (for constant velocity phases), or equal to the acceleration specified by the kinematics type. The value of a_{kin} is determined by the kinematics type (high, medium, or low).

$$\begin{aligned} KinematicsAcceleration(U) \leftrightarrow & \forall k \in [0, N-1], \\ & a_k \in \{-a_{kin}, 0, a_{kin}\} \end{aligned} \tag{4.13}$$

[2]Saccades for the corpus video profiles were generated at a finer resolution due to the higher update rate of the hard-coded profile controller which ran at $20Hz$. Our planner operates at $10Hz$, hence the triangle wave period of $0.02s$.

In addition to an acceleration value, the kinematics type also specifies a maximum velocity that the robot should not exceed. This is simply expressed with an inequality constraint $v_k \leq v_{kin}$. The kinematics type also captures the amount of energy used for a motion, hence the velocity should approach v_{kin} when possible. For example, accelerating to $v_k < v_{kin}$, performing a constant velocity phase, and decelerating should not occur. The same distance could be covered by a longer acceleration and deceleration, increasing the velocity at which the constant velocity phase is executed (or removing the need for it entirely, depending on the distance to be travelled). The most obvious constraint to apply is that constant velocity phases should only be planned at the maximum velocity (Equation 4.14). This is sufficient if the pause and increment constraints are not active. In order to allow prosody styles that specify both a kinematics type *and* use pauses or increments, the kinematics constraint should allow constant velocity phases at velocities lower than the desired v_{kin} as long as the constant velocity phase is simply a short pause or increment with $t_k \leq t_{pause}$ (Equation 4.15). The velocity and acceleration constraints are combined to form the overall kinematics constraint expressed in Equation 4.14.

$$\begin{aligned} KinematicsVelocity(U) \leftrightarrow & \forall k \in [0, N-1] \\ & 0 \leq v_k \leq v_{kin} \\ & \wedge \neg(a_k = 0 \wedge v_k \notin \{v_{kin}, 0\}) \end{aligned} \tag{4.14}$$

$$\begin{aligned} KinematicsVelocityPauses(U) \leftrightarrow & \forall k \in [0, N-1] \\ & 0 \leq v_k \leq v_{kin} \\ & \neg(a_k = 0 \wedge v_k \notin \{v_{kin}, 0\} \wedge t_k > t_{pause}) \end{aligned} \tag{4.15}$$

$$\begin{aligned} KinematicsConstraint(U) \leftrightarrow & KinematicsAcceleration(U) \\ & \wedge KinematicsVelocity(U) \end{aligned} \tag{4.16}$$

4.5.3 Offline trajectory planning with open-loop control

Problem formulation for variable distance prosody-compliant optimization

The new problem formulation given in Equation 4.17 retains the same control variables and cost function as the previous formulation. In this formulation, we have replaced the constraints which only limited the robot's acceleration and velocity according to the mechanical limits of the motors with the prosody constraints formulated previously. The set of constraints derived in the previous section is denoted as $PC_{offline}$, summarized in Table 4.1. Each constraint enforces trajectory properties which are specific to a given corpus variable value. In order for the trajectory planning to produce plans that reflect the desired movement prosody, we must select a subset of the constraints from $PC_{offline}$ which corresponds to our chosen movement prosody. For example, in order to plan trajectories according to the corpus variable values of *pause motion sequence*, *high kinematics* and *smooth variant*, we define the subset $PC_{active} = \{Pause, Kinematics, Smooth\}$, and set $a_{kin} = a_{high}$ and $v_{kin} = v_{high}$ to specify which kinematics type should be applied.

Table 4.1: Constraints forming the set $PC_{offline}$ used for offline planning.

Constraint	**Equation**
Pause	4.3
Hesitation	4.6
Smooth	4.9
Increment	4.12
Kinematics	4.16

$$\min_{u_0 \dots u_{N-1}} \quad \sum_{k=0}^{N-1} ||x_g||^2$$
$$\text{subject to:} \begin{cases} PC_{active} \subset PC_{offline}\,, \\ \sum_{k=0}^{N-1} t_k = T_h\,. \end{cases} \tag{4.17}$$

Trajectory planning

In order to solve the optimization problem, given the set of control inputs is finite (a_k and t_k are both discrete bounded variables) a simple approach would be to generate all possible trajectories similarly to the classical Dynamic Window Approach (DWA) (Fox et al., 1997). This approach is only feasible with small search spaces, the size of which depends on the time discretization dt chosen for t_k. Given several of our prosody constraints impose bounds on the phase durations, dt should be chosen accordingly. We must be able to plan a pause phase with duration $t_k = n{*}dt = t_{pause} = 300ms$, so dt should be smaller or equal to t_{pause}, and be a divisor of t_{pause}. The choice of dt will also impact the flexibility of the trajectories generated by the planner, hence its ability to accurately reach a target position in space, or a target velocity. For instance, using $dt = 300ms$ would mean that when the robot is moving at $v = 1.0m.s^{-1}$, increasing the length of a constant velocity phase by one dt extends the distance travelled by $dist = 0.3m$. A shorter dt would increase the search space for the length of motion phases, hence we choose $dt = 100ms$ as a compromise between these aspects.

Even with a reasonable choice of dt, the search space is still quite large. To reduce it, we can exploit the restrictive nature of our prosody constraints to discard candidate trajectories as soon as they violate one of the prosody constraints. We approach the problem as building a tree of possible trajectories starting from the robot's current state, iteratively adding phases in a depth-first fashion. A node corresponds to a state $\mathbf{x_k}$, an edge corresponds to a motion phase u_k, and a path of depth N corresponds to a trajectory. The root node corresponds to the robot's initial state. The set of possible control inputs for the kth phase is given as $u_k \in \mathcal{A} \times \mathcal{T}$, where $\mathcal{A} = \{a_{kin}, 0, -a_{kin}\}$ is the set of acceleration values determined by the kinematics type, and $\mathcal{T} = \{dt, 2dt, ...t_{max}\}$ is the set of possible phase durations. The maximum phase duration t_{max} is computed by subtracting the durations of previous phases and the minimum duration of the following phases from the planning

Algorithm 1: Prosody-aware trajectory planning

Input: x_g, goal point. $\mathbf{x_{init}}$, initial state. $PC_{active} \subset PC_{offline}$, set of active prosody constraints. $\mathcal{A}$ set of phase accelerations. $\mathcal{T}$ set of phase durations. N, number of motion phases.

Output: $U^* = \{u_0, u_1 \ldots, u_{N-1}\}$, phases of the optimal trajectory.

Notations:

$\mathbf{x_k} = [x_k, v_k]$, kth robot state.

$u_k = [a_k, t_k]$, kth motion phase.

T trajectory tree.

Algorithm:

$T \leftarrow CreateTree(\mathbf{x_{init}})$
$\mathbf{x_k} \leftarrow \mathbf{x_{init}}$
while $\neg TraversalFinished(T)$ **do**
  if $T.Depth(\mathbf{x_k}) < N$ **then**
    for $u_k \in \mathcal{A} \times \mathcal{T}$ **do**
      $\mathbf{x_{k+1}} \leftarrow ForwardSimulation(\mathbf{x_k}, u_k)$ (Eq. 4.1)
      $is\_valid \leftarrow CheckConstraints(PC_{offline}, u_{k-1}, u_k, \mathbf{x_k}, \mathbf{x_{k+1}})$
      if is_valid **then**
        $T.AddChild(\mathbf{x_k}, u_k, \mathbf{x_{k+1}})$
  $\mathbf{x_k}, u_{k-1} \leftarrow T.DepthFirstNextNode$
$U^* \leftarrow EvaluateTrajectories(x_g, T)$
$return \quad U^*$

horizon duration T_h (Equation 4.18).

$$t_{max} = T_h - \sum_{i=0}^{k} t_i - (N-k)dt \tag{4.18}$$

Pseudo-code for our algorithm is given in Algorithm 1. In order to expand the tree we select a control $u_k \in \mathcal{A} \times \mathcal{T}$ (line 5), and compute the state $\mathbf{x_{k+1}}$ that would result from executing u_k ($ForwardSimulation$ function, line 6). We then verify whether this extension of the trajectory satisfies the constraints using the $CheckConstraints$ function (line 7). This function evaluates each constraint in problem 4.17, returning a boolean value indicating whether the edge corresponding to control u_k is valid. If adding the edge to the tree causes the corresponding trajectory to violate any of the constraints, the edge is discarded. If the edge complies with the constraints, we add the node corresponding to the state $\mathbf{x_{k+1}}$ to the tree (lines 8-9). This process is repeated for all controls u_k, after which we select the next node from which to expand the tree in a depth-first fashion (line 10).

The tree expansion stops once all branches have been terminated, either due to violating a constraint, or due to reaching the maximum depth N. The result is a tree where each leaf node represents the last state of a fully prosody-compliant trajectory. Finding the optimal trajectory among these amounts to searching the tree for the lowest cost root-to-leaf path. This is performed by the function $EvaluateTrajectories$ which evaluates every trajectory according to the cost function from problem 4.17, and returns the one with minimum cost. In this section, we deal with the case of static environments without obstacles, so this trajectory planning process is only performed once. The optimal sequence of control inputs U^* is then used as input to the open-loop control algorithm described in the next paragraph, which executes the control inputs with appropriate timing.

Open-loop control

Algorithm 2 describes the overall process to execute a prosody compliant trajectory in an open loop fashion. It uses the planning algorithm 1 as a subroutine. The input to the control algorithm is the goal position x_g given in the robot's local coordinate frame, as well as a selection of prosody constraints PC_{active}. We plan the trajectory using algorithm 1 to solve the optimization problem given in Equation 4.17, obtaining the optimal trajectory U^*. We then simply iterate over the controls $\{u_0, u_1 ... u_{N-1}\}$, sending the corresponding acceleration command a_t to the motors, and waiting for the duration t_t of the motion phase to elapse before sending the next command. In this approach, planning only occurs once, and is separated from the control. No feedback about the environment is used once the robot begins executing the planned trajectory.

This problem formulation has allowed us to introduce our notations and general process, showing how constraints can be formulated to structure the trajectory planning so that only trajectories that exhibit the desired movement prosody properties can be executed. We limited ourselves to static, known environments, where there are no obstacles between the robot and the goal position. We also assumed an ideal system with no inac-

Algorithm 2: Open-loop control

Input: x_g, goal point.
$PC_{offline}$ set of prosody constraints.
Output: a_t, acceleration command sent to the motors.
Notations:
$\mathbf{x_0} = [x, v]$, initial state of the robot.
$u_t = [a_t, t_t]$, motion phase executed at time t.
$U^* = \{u_0, u_1 \ldots, u_{N-1}\}$, sequence of motion phases describing the trajectory to be executed.
Algorithm:

```
U* ← PlanTrajectory(x_0, x_g, PC_offline) (alg. 1)
for k ∈ [0, N − 1] do
    a_t, t_t ← ExtractControl(U*, k)
    SendMotorCommand(a_t)
    DelayUntil(t + t_t)
```

curacies in sensing or motion control. These are far from the real deployment situations, hence the approach should be further extended to handle more complex situations. We are interested in deploying the robot in human populated, dynamic environments where the future state of the environment and other agents is either unknown or difficult to predict long-term. In such environments, pre-computing a trajectory and executing it in an open loop fashion could fail. Likewise, the goal itself may not be a static point in the environment, but a moving person, leading to similar issues. In the following section, we discuss how we extend the optimization problem formulation, constraints, and algorithm in order to handle dynamic environments.

4.6 Receding horizon control for dynamic environments

Mobile robots destined to operate around humans are bound to encounter dynamic environments where the future motion of other agents will only be partially known through predictions with varying degrees of accuracy. In addition, no matter how accurate the forward model of the robot's kinematics or dynamics is, there can always be some mismatch with the true motion of the robot due to factors such as wheel slip or hardware failures. For these reasons, a common approach is to re-plan trajectories frequently to adapt the robot's motion to changes in the environment and to mitigate inaccuracies in the robot's model. We follow this approach, thus aiming to design an online local planning module that generates trajectories at a fast update rate over a short time horizon. First, we describe how adding a re-planning mechanism can allow us to perform prosody-compliant motions in dynamic environments in some cases, as well as its shortcomings due to the inadequate prosody constraint formulation. Second, we propose extensions to the previous prosody constraints in order to adapt them to a re-planning approach. Thirdly, we update the problem formulation and algorithm to incorporate re-planning and the new constraints.

4.6.1 Impact of re-planning on prosody compliance

The previous problem formulation assumed that the environment was perfectly known, allowing a trajectory to be planned once, and executed "blindly", in an open loop fashion. In reality, mobile robots will have incomplete knowledge of their environment, in part due to the finite range of their sensors, and limited prediction accuracy for other agents in the environment. Re-planning the trajectory frequently can allow the trajectory to be updated based on new sensor data, new predictions, and the true state of the robot. An illustrative example is depicted in Figure 4.5, where a sudden motion of a person requires re-planning in order to avoid colliding while maintaining consistent movement prosody. In order to be useful, re-planning should be performed at a high frequency, which often prohibits planning an entire trajectory from the robot's current position to the goal. Instead, a common approach is to plan a trajectory that minimizes the cost function over a finite time horizon T. This kind of approach is employed in some classical local planning algorithms such as Dynamic Window Avoidance (DWA) (Fox et al., 1997) which plans a trajectory consisting of only a single control input over a short horizon at each planning cycle. A DWA approach would not be suited in our case, since our prosody constraints are defined over several control inputs, requiring trajectories to be planned over several control inputs. DWA can be seen as a special case of receding horizon control approaches, which typically optimize over horizons consisting of multiple control inputs. Instead of selecting a single control input, we select a sequence of actions covering the whole horizon. This is a more accurate representation of the flexibility of the robot's motion which alleviates the issues with DWA, at the cost of a higher computational complexity. The first control action of the trajectory is then executed on the robot over the period of the planning cycle update, after which a new trajectory is re-planned in a receding horizon fashion. The result is that when a previously valid trajectory plan becomes invalidated due to a change in the environment, the frequent re-planning allows us to find a new valid trajectory that also satisfies the prosody constraints.

In the next paragraph, we discuss the choice of the horizon length T. This choice is especially important in our algorithm since our prosody constraints often require certain fixed-duration motion phases to be applied (such as when using pause or hesitation constraints), thus requiring the horizon to be long enough to accommodate them.

Safety, terminal constraint, and planning horizon

In order to check if a trajectory collides with an obstacle or person, we assume we have access to the distance to the closest obstacle along the robot's x axis, denoted d_{obs}. Using this information, we formulate a simple collision constraint (Equation 4.19) by checking whether the closest obstacle is at least $d_{critical} = 0.05m$ further away than the robot's future planned positions x_k along the x axis of motion.

$$\begin{aligned} CollisionCritical(U) \leftrightarrow & \forall k \in [0, N-1]\,, \\ & d_{obs} - x_k > d_{critical} \end{aligned} \tag{4.19}$$

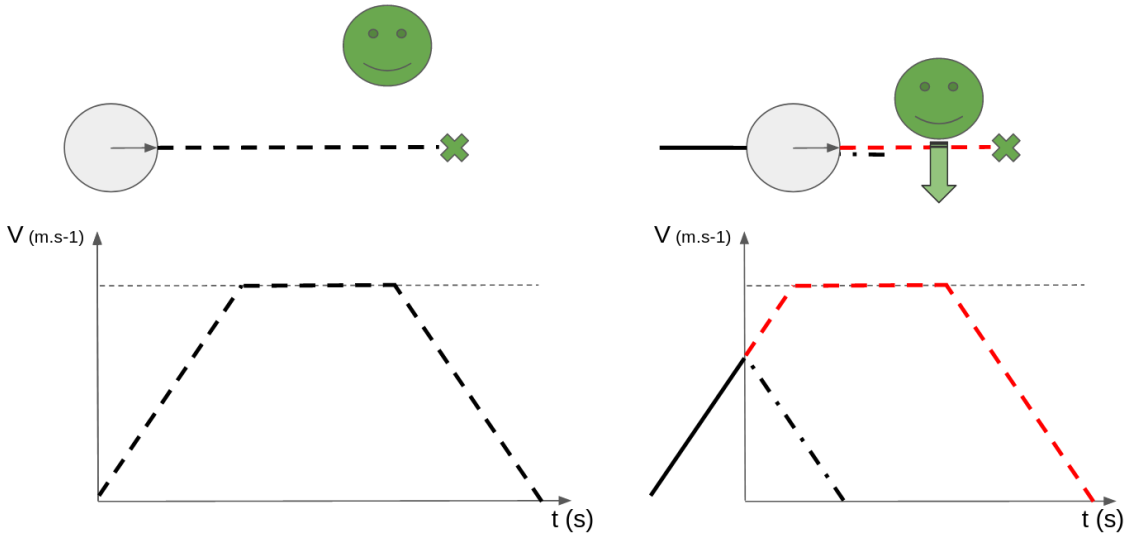


Figure 4.5: Illustration of a situation requiring re-planning. Left: initial plan to move the robot (grey) towards the goal (x). Right: a person (green) suddenly decides to move, crossing the robot's path. Without any safety mechanism there would be a collision. With an emergency stop we would no longer comply with the kinematics prosody constraint. Instead, we can re-plan a new trajectory which is shorter, hence avoiding collision and satisfying the kinematics prosody. Solid line: past motion. Dashed lines: valid plans (black), invalid plan (red).

This constraint by itself cannot guarantee that collisions will be prevented, even in static environments, since it's effectiveness depends on the time horizon T_h over which the trajectory is planned, and the robot's stopping time. For example, a trajectory consisting of maintaining a constant velocity might be collision-free but very close to an obstacle at one timestep, and then become invalid at then next timestep as the robot gets closer to the obstacle. If the T_h is not larger than the robot's stopping time, the robot cannot decelerate in time to avoid a collision. Therefore, when using receding horizon control approaches it is typical to determine the minimum horizon length in accordance with the robot's stopping time under the maximum deceleration of which the hardware is capable. Some approaches combine this with a terminal constraint (Mayne et al., 2000) which can be used to force the robot to be at rest at the end of the planning horizon, thus providing passive safety (Zheng, 2022). We adopt a terminal constraint that enforces that the robot's velocity at the last state of a planned trajectory should be zero (Equation 4.20).

$$TerminalConstraint(U) \leftrightarrow v_{N-1} = 0 \tag{4.20}$$

Although using a terminal constraint improves safety, it may modify trajectory plans in other ways, depending on T_h. Since the terminal constraint requires the robot to be stopped at the end of the trajectory which has a finite duration, it limits the maximum velocity at which the robot can travel based on the robot's stopping time. Many of the prosody constraints extend the robot's stopping time by altering the maximal deceleration (kinematics constraint), adding additional motion phases (hesitation, pause and increment constraints).

Therefore, if T_h is not changed to account for the additional stopping distance introduced by our prosody constraints, then the robot would not be able to satisfy the kinematics constraints requiring the robot to plan trajectories that reach the specified velocity v_{kin}. Our aim is for the robot to be able to comply with the prosody constraints as much as possible, hence we determine the minimum horizon length according to the stopping time subject to the prosody constraints. Given that we always plan under the prosody constraints, the effect of the terminal constraint becomes that it ensures the robot can stop by the end of the horizon, *while* satisfying the prosody constraints.

In addition to the horizon length T_h we must consider the parameter N, which is the number of motion phases contained in the horizon. For example, with $N = 2$ it is impossible to plan a trajectory using the pause constraint, since this requires three motion phases (acceleration, constant velocity, deceleration) to form the trapezoidal profile. Longer horizon times allow the planner to handle more complex navigation tasks, however the accuracy of the prediction of future motion of the robot and dynamic elements of the environment as well as the perception range for obstacles provide upper bounds on the horizon time, and the computational complexity also grows with the number of phases in the horizon. We choose to use the minimum horizon length and number of motion phases that allows prosody compliant motions to be planned. For example, when planning motions with the smooth or saccade variants, no pauses and no hesitations, a horizon consisting of two motion phases, and a length corresponding to the time required for an acceleration to maximal velocity followed by a deceleration to a stop.

Inadequacy of static prosody constraints for re-planning

Incorporating re-planning into our existing problem formulation can help to handle some situations in dynamic environments. For example, consider the robot has started executing a motion using the high kinematics and pause constraints, in order to move towards a person standing still. The robot has already performed an acceleration to the maximal velocity, and its current plan is to perform a long constant velocity phase, followed by a deceleration to stop at the goal near the person $U_t = \{[0m.s^{-2}, 600ms], [-0.5m.s^{-2}, 1500ms]\}$. If the person decides to start moving closer to the robot at $t + 1$, the plan computed at the next cycle can account for the shorter distance to be travelled by reducing the duration of the constant velocity phase, while still complying with both the kinematics and pause constraints. However if the goal distance gets close enough, the planner may plan to immediately decelerate. When considered in isolation, there is nothing wrong with such a plan: according to the pause constraints, we should not **plan** an acceleration followed by a deceleration. However in this instance, the plan does not even contain an acceleration, so it is valid. In this situation, if we want the robot's motion execution to comply with the prosody, we would like to express that a deceleration phase should not be planned because the **previously executed** phase was an acceleration. Similarly, if the robot had already performed a small portion of a pause phase then simply checking the previous phase's acceleration value would not be enough, we would also need to check if the pause phase had been maintained for long enough. In more general terms, the planned trajectory should not only be internally consistent with the prosody constraints, but also be consistent with respect to the past motion of the robot.

4.6.2 Prosody constraint adaptation to re-planning

In this section, we present the new constraint formulations for the motion prosody parameters that are affected by the re-planning mechanism. Constraints which are not mentioned in this section maintain the same formulation as previously.

Many of the constraints require a way to ensure a motion phase is executed for some minimum duration, such as the pause constraint imposing constant velocity phases to have a duration of $300ms$. When re-planning, there will be moments in the robot's motion when it has performed only part of the desired motion phase, requiring the first phase of the plan to complete the remaining duration of that phase. For example, if the robot has only been performing a constant velocity phase for 200ms, then the first motion phase of the trajectory is constrained to be an acceleration phase of at least 100ms. We introduce a new variable which keeps track of the cumulative duration of a motion phase denoted by t_cumul_k, defined in Equation 4.21.

$$t\_cumul_k = \sum_{i=k}^{j} t_i, \quad s.t. \quad \forall i \in [j,k]\,, a_i = a_k \tag{4.21}$$

In the following sections, we show how the motion history captured by t_cumul is used in order to formulate more complex prosody constraints that enable the robot to maintain consistency with respect to the past motion when re-planning.

Motion sequences

The first change to the pause constraint is that it also applies to the first phase of the trajectory, checking if the previously executed phase a_{prev} was an acceleration or deceleration (Equation 4.22). The second change is the addition of a constraint to make sure that if a pause started, the next phase should complete it (Equation 4.23). The updated pause constraint for re-planning is given in Equation 4.24.

$$\begin{aligned} PauseAfterAccel(U) \leftrightarrow & \forall k \in [0, N-1]\,, \\ & \neg\big(a_{k-1} \neq 0 \wedge a_k \neq a_{k-1} \wedge \neg(a_k = 0 \wedge t_k \geq t_{pause})\big) \end{aligned} \tag{4.22}$$

$$\begin{aligned} UnfinishedPause(U) \leftrightarrow & \forall k \in [0, N-1]\,, \\ & (a_{k-1} = 0, \wedge t_{k-1} < t_{pause}) \\ & \wedge \neg(a_k = 0 \wedge t\_cumul_k = t_{pause}) \end{aligned} \tag{4.23}$$

$$\begin{aligned} PauseConstraint(U) \leftrightarrow & PauseAfterAccel(U) \\ & \wedge \neg UnfinishedPause(U) \end{aligned} \tag{4.24}$$

Similarly to other constraints, the previous hesitation constraints should now also be applied to the first phase. We also need to modify the constraints in several ways: a hes-

itation acceleration should only be planned once the hesitation deceleration has finished entirely; and instead of requiring a non-hesitation acceleration to be followed by a hesitation deceleration, we simply require that it should *not* be followed by a constant velocity phase or a non-hesitation deceleration. This has the effect of allowing a previously executed acceleration to be continued. We also add new constraints so that partially executed hesitation phases are completed by the next motion phase, similarly to the unfinished pauses in prior constraints (Equation 4.27 and Equation 4.28).

$$\begin{aligned} &HesitationDeceleration(U) \leftrightarrow \forall k \in [0, N-1]\,, \\ &(a_{k-1} = a_{kin} \vee t_{since\_hesit} \geq t_{h\_interval}) \\ &\wedge \big((a_k = -a_{kin} \wedge t_k \neq t_h) \vee a_k = 0\big) \end{aligned} \tag{4.25}$$

$$\begin{aligned} &HesitationAcceleration(U) \leftrightarrow \forall k \in [0, N-1]\,, \\ &(a_{k-1} = -a_{kin} \wedge type_{k-1} = hesitation \wedge t\_cumul_{k-1} = t_h) \\ &\wedge \neg(a_k = a_{kin} \wedge t_k = t_h) \end{aligned} \tag{4.26}$$

$$\begin{aligned} &HesitationUnfinishedDecel(U) \leftrightarrow \forall k \in [0, N-1]\,, \\ &(a_{k-1} = -a_{kin} \wedge type_{k-1} = hesitation \wedge t\_cumul_{k-1} < t_h) \\ &\wedge \neg(a_k = -a_{kin} \wedge t\_cumul_k = t_h) \end{aligned} \tag{4.27}$$

$$\begin{aligned} &HesitationUnfinishedAccel(U) \leftrightarrow \forall k \in [0, N-1]\,, \\ &(a_{k-1} = a_{kin} \wedge type_{k-1} = hesitation \wedge t\_cumul_{k-1} < t_h) \\ &\wedge \neg(a_k = a_{kin} \wedge t\_cumul_k = t_h) \end{aligned} \tag{4.28}$$

$$\begin{aligned} HesitationConstraint(U) \leftrightarrow &\neg HesitationDeceleration(U) \\ &\wedge \neg HesitationAcceleration(U) \\ &\wedge \neg HesitationUnfinishedDecel(U) \\ &\wedge \neg HesitationUnfinishedAccel(U) \end{aligned} \tag{4.29}$$

Variants

If we only consider the immediate state of the robot at the time of re-planning, the motion phases which are part of the plan will be consistent with each other, and comply with the smooth variant, but this might not be the case when we consider the consistency with respect to the previously executed motion (an illustrative example is provided in Figure 4.6). We re-formulate the smooth constraint to use the cumulative time, such that if the robot has started an acceleration phase, it should continue it. The revised constraint is given in Equation 4.30.

$$Smooth(U) \leftrightarrow \forall k \in [0, N-1]\,, t\_cumul_k \geq t_{smooth} \tag{4.30}$$

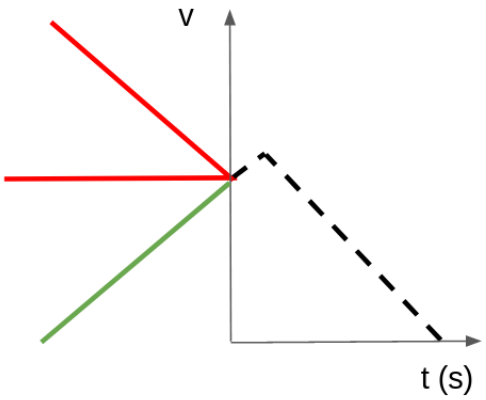


Figure 4.6: Illustration of the dependence of the planned trajectory on past motion. Dashed lines: planned trajectory. Green line: past motion for which the plan is consistent with the smooth variant. Red: past motions for which the plan violates the smooth variant by introducing a small stutter.

The increment constraint requires insertion of constant velocity phases similar to the pause constraint, as seen in Figure 4.7, so we add the same constraint to ensure that a partially executed pause is finished by the next motion phase (Equation 4.23. This constraint is added to the two previous increment constraints derived in the previous section (Equation 4.32). The constraint enforcing that accelerations should be followed by pauses remains similar, except now it should be applied over a wider range, including the check between the previously executed acceleration a_{prev} (i.e. when $k = -1$), and the first acceleration of the plan a_0 (Equation 4.31).

$$\begin{aligned} BreakAccelerationPhase(U) \leftrightarrow & \forall k \in [0, N-1], \\ & a_{k-1} \neq 0 \\ & \wedge ((a_k = 0 \wedge t_k = t_{pause}) \vee a_k = -a_{k-1}) \end{aligned} \tag{4.31}$$

$$\begin{aligned} IncrementConstraint(U) \leftrightarrow & ValidVelocity(U) \\ & \wedge BreakAccelerationPhase(U) \\ & \wedge \neg UnfinishedPause(U) \end{aligned} \tag{4.32}$$

4.6.3 Algorithm for prosody compliant receding horizon control in dynamic environments

In this section, we show how the re-planning is incorporated into our algorithm. Changes are made to the planner update loop, by re-planning the trajectory at each update cycle instead of only once. The planner operates at a fixed update frequency with a period of $dt = 100ms$. The trajectory optimization formulation given in Equation 4.33 is similar

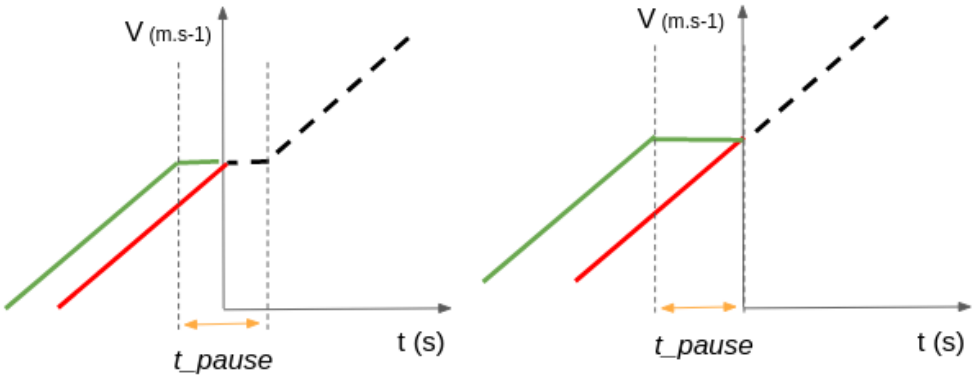


Figure 4.7: Illustration of the validity of motion phase sequences with respect to different past motions, using the increment variant. Dashed lines: planned trajectory. Green line: past motion for which the plan is consistent with the increment variant. Red: past motions for which the plan violates the increment variant.

Table 4.2: Constraints forming the set PC_{replan} used for re-planning in receding horizon control.

Constraint	**Equation**
Pause	4.24
Hesitation	4.29
Smooth	4.30
Increment	4.32
Kinematics	4.16

to the prior formulation, with the key difference being that it uses the new prosody constraints presented in section 4.6.2. The active prosody constraint set PC_{active} is a subset of PC_{replan} defined in Table 4.2, which includes the updated prosody constraints enabling consistent movement prosody to be maintained while re-planning. Additionally, the collision constraint is included to ensure the trajectories do not cause collisions. Lastly, the duration of the trajectory is constrained to a shorter time horizon T_h, chosen such that it is longer than the robot's stopping distance subject to the prosody constraints, to ensure the ability of the planner to find safe prosody-compliant trajectories.

$$\min_{u_0 \dots u_{N-1}} \quad \sum_{k=0}^{N-1} ||x_g||^2$$
$$\text{subject to:} \begin{cases} PC_{active} \subset PC_{replan}\,, \\ \sum_{k=0}^{N-1} t_k = T_h\,, \\ CollisionCritical \quad (4.19)\,, \\ TerminalConstraint \quad (4.20)\,. \end{cases} \tag{4.33}$$

Receding horizon control loop

Our online local planner using a receding horizon control approach is presented in Algorithm 3, which replaces the open-loop trajectory execution presented in Algorithm 2. This version implements closed-loop control by using information from perception modules to re-plan the trajectory at every control cycle (i.e. with a period $dt = 100ms$) (line 7). This algorithm uses two forms of environment feedback, the first being the distance to the closest obstacle along the robot's x axis denoted d_{obs}, and the second being the position of the goal x_g, which may be based on the position of a person given by a person tracking module, or a point in the environment given by a global path planner (line 3). The collision constraint (Equation 4.19) ensures that for all motion phases in the trajectory U, the distance between the robot and obstacle is never less than $d_{critical}$, which we set to $0.05m$. Updating the goal position x_g continuously enables the robot to adapt to mobile goals (e.g. approaching a moving person), or to compensate for inaccuracies in the robot's motion. As part of the perception update, we also update the variables which are used to check specific prosody constraints. The variable t_{since_hesit} is updated according to whether the phase executed at time t was a hesitation, and t_{cumul} is incremented if the phases at times t and $t - dt$ are of the same type (line 3).

We introduce the possibility of updating the set PC_{active} of prosody constraints currently being applied, via a basic user interface (lines 5-6). We only update this set when the robot is not moving, since instantaneously altering the constraints while in motion can make the problem infeasible. This issue of switching between prosody styles is discussed further in section 4.8.2. Given the state of the robot and the environment at time t, we solve the optimization problem given in 4.33. The first difference to the previous problem formulation is that the prosody constraints are those that account for the dynamic re-planning, derived in section 4.6.2, and summarized in Table 4.2. These constraints require access to the motion phase executed during the previous timestep, denoted u_{t-dt}. The second difference is the addition of the collision constraint. These changes only affect the inputs to the trajectory planning step, so we can keep using Algorithm 1, passing it the new constraints and the appropriate variables required to evaluate them.

Once the optimal trajectory is found, we send an acceleration command a_t corresponding to the acceleration a_0 of the first motion phase u_0. u_0 is stored as u_t in order to be passed as input to the optimization problem as u_{t-dt} at the next planning cycle[3] so that the prosody constraints requiring access to the past robot motion can be evaluated.

[3]Note: passing the optimal trajectory from a prior timestep may resemble a typical warm-start procedure, however we remind the reader that this is not how we use this information. Instead, the prior motion phase is used explicitly in our prosody constraints to ensure temporal consistency of the prosody style.

Algorithm 3: Prosody-aware receding horizon control

Input: d_{obs}, distance to closest obstacle. x_g, goal position.
$PC_{active} \subset PC_{replan}$ set of active prosody constraints.
Output: a_t, acceleration command sent to the motors.
Notations:
$\mathbf{x_t} = [x, v]$, state of the robot at the start of the current cycle.
$u_t = [a_t, t_t]$, motion phase executed after the trajectory optimization has been performed.
$U = \{u_0, u_1 \ldots, u_{N-1}\}$, set of motion phases describing the trajectory.
Algorithm:

$u_t, u_{t-dt}, \mathbf{x_t}, \mathbf{x_{t-dt}} \leftarrow InitializeToZero()$
while *True* **do**
  $d_{obs,t}, x_{g,t}, \mathbf{x}_t \leftarrow UpdatePerception()$
  $temp\_PC \leftarrow ReceiveProsodyConstraints()$
  if $Stopped(\mathbf{x_t})$ **then**
    $PC_{replan} \leftarrow temp\_PC$
  $U^*, X^* \leftarrow PlanTrajectory(\mathbf{x_t}, d_{obs,t}, x_{g,t}, u_{t-dt}, PC_{active})$ (alg. 1)
  $a_t, t_t \leftarrow ExtractControl(U^*, 0)$
  $SendMotorCommand(a_t)$
  $u_{t-dt} \leftarrow u_t$
  $DelayUntil(t + dt)$

4.7 Conserving partial prosody compliance when encountering an infeasible optimization problem

In the previous section, we discussed how adopting a receding horizon control approach allowed the planner to handle certain instances of trajectories become invalid due to changes in the environment by solving an updated optimization problem accounting for the new environment state, and finding a new prosody-compliant trajectory. In a similar fashion, re-planning can mitigate inaccuracies in our kinematics model and environment model caused by unmodeled delays in perception, planning, and motor control. Nevertheless, there may be situations where the environment's configuration, or the rate of change is such that there are no valid prosody-compliant trajectories, in which case re-planning would fail. These problems are not specific to our planner, but they are more likely to occur given the heavily restricted space of prosody-compliant trajectories. In this section we discuss extensions to our prosody-compliant local planner which aim to address these issues by explicitly accounting for perception, computing, and actuation delays in the system, and by building a hierarchy of constraints to allow the planner to plan trajectories which partially satisfy the prosody constraints when it is impossible to fully satisfy them. We first provide examples of the problematic situations, and explain the conceptual solutions. Then, we show how the solutions are integrated into our planning algorithm.

4.7.1 Mitigation of infeasible optimization through the use of a constraint hierarchy

Enabling partial prosody compliance to handle unpredictable environments

In some situations, simply re-planning frequently is not enough to avoid problems. For instance, Figure 4.8 illustrates an example of a situation where the robot using the hesitant constraint has performed an acceleration phase, and started preforming the hesitation deceleration. The previous plan consisted of completing the hesitation by finishing the deceleration, accelerating back to the maximal velocity, and finally decelerating to a stop at some desired distance from a person. As the robot approaches, the person may decide to start moving in some arbitrary direction which could not have been predicted, or a previously occluded person may become visible by the robot's onboard sensors, suddenly rendering the previous plan infeasible due to a collision risk. If the previous trajectory plan was already the shortest possible trajectory given the prosody constraints, then the algorithm proposed in the previous section would fail to produce a valid plan. We could implement some fallback solutions such as performing a kind of emergency braking by disregarding all prosody constraints and stopping as fast as the motors allow. This may shorten the trajectory enough to avoid collision, but it would also significantly alter the robot's movement prosody. Depending on which prosody constraints are being applied, the difference in length between a full prosody compliant trajectory and the emergency braking trajectory may be large (potentially several meters when using hesitation or increment constraints). More interestingly, there can be large differences even between different

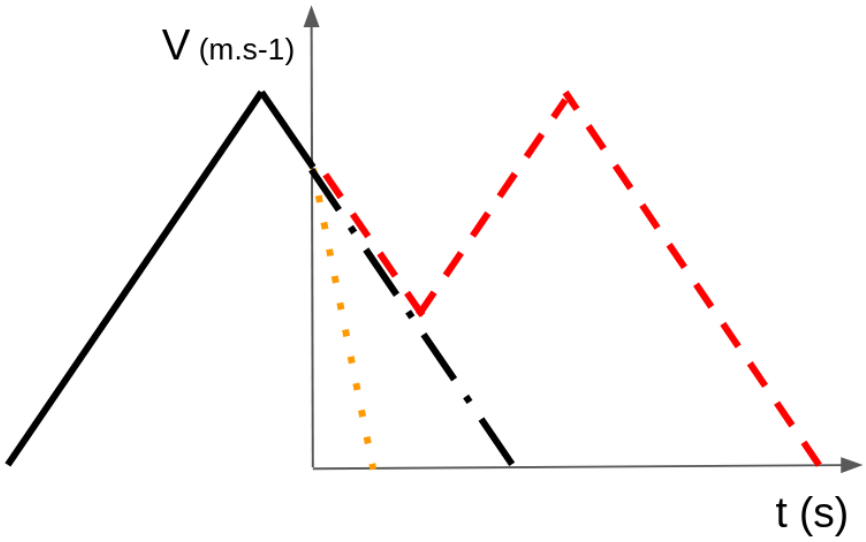


Figure 4.8: Trajectory re-planning alternatives after the original hesitation trajectory plan (red, dashed lines) becomes invalid due to an obstacle. Solid line: motion executed prior to the current planning cycle. Orange dashed line: emergency braking; black dashed line: deceleration satisfying the kinematics constraint, but not the hesitation constraint.

prosody constraints, for example trajectories that only comply with a kinematics constraint can be much shorter than those complying with kinematics *and* increment, pause, or hesitation constraints. Therefore, we would like to have the ability to plan a trajectory which sacrifices compliance with *some* of the prosody constraints in order to avoid collisions, but maintains compliance with a subset of the prosody constraints to limit the change in movement prosody. An example of such an intermediate solution is depicted by the black dashed lines in Figure 4.8.

When dealing with conflicting tasks or objectives, a common approach in robotics is to define a form of priority or ordering so that less important objectives are only optimized as long as they do not interfere with more important objectives (Siciliano & Slotine, 1991). We take inspiration from this idea and propose to order our prosody constraints according to $L-1$ levels in a hierarchy according to their priority. When searching for the optimal trajectory, the planner should always guarantee that the constraints at level l are satisfied before attempting to satisfy constraints at level $l+1$, meaning constraints at level l have higher priority than those at level $l+1$. Using our previous example, avoiding collision and complying with the kinematics type have the highest priority, so they are assigned to level 1. Complying with the hesitation constraint has lower priority, and is therefore assigned to level 2. The priority level of a *trajectory* is the level of the lowest priority constraint it satisfies, while also satisfying all higher level constraints i.e. $P_{traj} = l$ means that it satisfies all constraints in levels $[1, ...l]$. The planner can then use this hierarchy by first searching for trajectories with priority level $l = L-1$. If and only if no such trajectory exists, the planner can remove the constraints at level l, thus searching for trajectories with priority level $l-1$, i.e. that satisfy fewer levels of the constraint hierarchy. In simple navigation situations, the trajectory optimization is identical to the previous approach where all prosody constraints should always be satisfied. However in complex situations where the full set

Table 4.3: Constraint hierarchy

Priority level	Constraints
1 (highest)	CollisionCritical, TerminalConstraint, KinematicsType
2	Smooth, Increment, Hesitation, Pause
3 (lowest)	CollisionIdeal

of constraints makes the problem infeasible, this approach allows the planner to search for trajectories which satisfy at least some of the constraints, rather than immediately resorting to an emergency stop which violates most if not all of the prosody constraints. The constraint hierarchy is shown in Table 4.3.

With our approach, we cannot express the idea of planning trajectories that minimize the divergence from a constraint that cannot be satisfied. For example, if no trajectory is able to satisfy the kinematics velocity constraint imposing $a_k = 0 \Rightarrow v_k = v_{kin}$, we may wish to instead execute the trajectory with the closest velocity. With our basic formulation, if the velocity constraint cannot be satisfied then trajectories may be planned without any regard to their maximum velocity. A possible solution would be to instead apply lexicographic optimization techniques, where we formulate costs that are sequentially optimized according to a similar hierarchical structure, as was used for bipedal robot locomotion in (Ciocca, 2020) in order to handle the conflicting safety costs of avoiding collisions with people and maintaining balance.

This approach may also be useful if we determine that certain prosody parameters have more importance than others. By separating the prosody constraints into different priority levels, one could more smoothly transition from the full prosody trajectory planning to pure collision avoidance. Whether or not partial application of our prosody constraints is relevant with respect to how the robot is perceived by people requires further study.

Collision margin tolerance to mitigate sensor and actuation latency

Even in situations where the environment is static, our prior algorithm may fail to execute prosody compliant motions due to inaccuracies in the robot's actuation model and latency induced by computation time leading to states where the optimization with a strict collision constraint becomes infeasible. In the previous algorithm, the trajectory optimization is performed by assuming all computations happen instantaneously, and that the optimal control will be applied immediately. These delays can introduce errors into the system if they are left unaccounted for, and this issue is somewhat exacerbated by our use of a heavily constrained trajectory space, and necessity of fine control over the robot's motion. Such small inconsistencies may not be accounted for in many social navigation works, however we take inspiration from the autonomous wheelchair navigation presented in (Park, 2016) where they found that compensating for such timing issues was important to generate their desired quality of motion. Modeling the latency of the lower level motor control and physical motor response is non-trivial, and there will always be some level of uncertainty we cannot account for, such as in the robot's state estimation and sensor data.

If there remains even a small amount of error in the projection of the trajectory length, we may encounter situations similar to that expressed in the previous section where a previously valid trajectory is later found to cross just under the minimum collision distance threshold, leaving the planner with no valid trajectories. One solution would be to introduce a notion of collision tolerance margin, whereby the planner aims to stay some distance away from obstacles so that executing a trajectory with a slightly inaccurate length would not result in a collision. We can utilise the constraint hierarchy to achieve such an effect by formulating two collision constraints. The first imposes a distance $d_{critical} = 0.05m$ and has high priority. The second constraint imposes a larger distance $d_{ideal} = 0.2m$, and has a lower priority than the prosody constraints. The effect is that the planner will first consider trajectories that stay further than d_{ideal} from obstacles and comply with the prosody constraints. If the planner is unable to find a trajectory with $d_{obs} > d_{ideal}$, then it is able to plan prosody-compliant trajectories where $d_{critical} \leq d_{obs} \leq d_{ideal}$. The difference between the distance thresholds for these constraints represents an estimate of the uncertainty on the trajectory distance. The ideal obstacle distance constraint is given in Equation (Equation 4.34).

$$\begin{aligned} CollisionIdeal(U) \leftrightarrow & \forall k \in [0, N-1], \\ & d_{obs} - x_k > d_{ideal} \end{aligned} \tag{4.34}$$

4.7.2 Final algorithm for prosody compliant local navigation in dynamic environments

In the following sections, we show how the constraint hierarchy and latency compensation mechanisms are integrated into our receding horizon local planner. We first describe how the trajectory optimization is modified to include the constraint hierarchy. Then, we describe how the planning update cycle is modified to include the sensor and computation delay compensation.

Trajectory planning with a constraint hierarchy

The final version of our trajectory planning algorithm (algorithm 4) incorporates the constraint hierarchy mechanism. In the previous planning algorithm (alg. 1), a motion phase edge u_k was only added to the trajectory tree if it satisfied *all* constraints. When using a constraint hierarchy, we do not initially know which level of constraints will be satisfiable. On line 9, we check the new motion phase edge u_k using the $CheckConstraints$ function, which instead of a boolean value now returns P_{phase}, indicating the priority level satisfied by the motion phase. With our three-level constraint hierarchy (Table 4.3), $P_{phase} = 3$ indicates all levels were satisfied, whereas $P_{phase} = 1$ indicates that only the highest priority level was satisfied. Using this information, we determine the priority level of the trajectory denoted P_{curr}, which corresponds to the lowest common priority level satisfied by all its motion phases up to the current node (line 10).

As we perform the search, we will gradually discover trajectories that satisfy different

levels of the constraint hierarchy, so we keep track of the lowest priority level P_{best} satisfied by a complete trajectory since the start of the planning process (updated in lines 13-14). If we have already found a trajectory satisfying the constraints up to and including level l, then we can stop adding edges to trajectories which satisfy fewer levels of the hierarchy (lines $11 - 12$). P_{best} is updated when a trajectory has been completely planned, i.e. if the edge u_k was the Nth motion phase (lines 13-14). Once the tree of feasible trajectories has been built, the $EvaluateTrajectories$ function (line 16) only considers trajectories that have the same priority level as the best trajectory, ensuring that compliance with movement prosody is maintained even if there exist trajectories which would have had lower cost but that violate the prosody constraints.

The set of prosody constraints from which we select PC_{active} is unchanged with respect to the previous section (i.e. they are still those in the set PC_{replan} shown in Table 4.2). However, we introduce the additional collision constraint (*CollisionIdeal*, Equation 4.34), which assists in compensating for small model inaccuracies.

Receding horizon control with latency compensation

Algorithm 5 details the full planning cycle in our receding-horizon controller. Modifications have been made to the planning cycle in order to more accurately plan trajectories by explicitly taking into account sensor and perception latency and computation time delays. In order to maintain a constant control update frequency, the acceleration command computed at a given planning cycle is stored and applied only at the start of the next planning cycle (line 4). This is explicitly accounted for by computing estimates of the robot and obstacle states at the start of the next cycle, which are used as the initial states for the optimization problem (lines 5-7). The robot state at $t + dt$ is estimated given its current state x_t and the acceleration it will apply during the current planning cycle a_t. Obstacle positions are also projected towards their estimated positions at time $t + dt$, purely based on the robot's ego-motion, since in this work we do not use any prediction of dynamic obstacle motion.

Given the projected states of the robot and the environment at time $t + dt$, as well as the information regarding the motion phase to be executed between time t and $t + dt$, we solve the optimization problem 4.33, subject to the prosody constraints specified in PC_{active} (line 11). We detail this optimization in the following subsection. The acceleration command of the first control of the solution is stored in order to be executed at time $t + dt$, and passed as input to the optimization problem at the next planning cycle.

Algorithm 4: Trajectory planning with prosody constraint hierarchy

Input: d_{obs}, distance to the closest obstacle along the robot's x axis. x_g, goal point. u_{k-1}, previous motion phase. x_0, initial state. H, constraint hierarchy (Tab. 4.3). $PC_{active} \subset PC_{replan}$, set of prosody constraints.

Output: $U^* = \{u_0, u_1 \ldots, u_{N-1}\}$, motion phases of the optimal trajectory.

Notations:

$\mathbf{x_k} = [x_k, v_k, t_{since\_hesit}, t_{cumul}]$, kth robot state.

$u_k = [a_k, t_k]$, kth motion phase.

d_{obs}, distance to closest obstacle.

$P_{best}, P_{curr}, P_{phase}$, priority levels in the constraint hierarchy H.

$\mathcal{A} = \{a_{kin}, 0, -a_{kin}\}$, set of acceleration values.

$\mathcal{T} = \{dt, 2dt, ...t_{max}\}$, set of phase durations (Eq. 4.18).

T trajectory tree.

Algorithm:

$T \leftarrow CreateTree(\mathbf{x_0})$
$\mathbf{x_k} \leftarrow \mathbf{x_0}$
$P_{curr} \leftarrow max(H)$
$P_{best} \leftarrow min(H)$
while $\neg TraversalFinished(T)$ **do**
  if $T.Depth(\mathbf{x_k}) < N$ **then**
    for $u_k \in \mathcal{A} \times \mathcal{T}$ **do**
      $\mathbf{x_{k+1}} \leftarrow ForwardSimulation(\mathbf{x_k}, u_k)$ (Eq. 4.1)
      $P_{phase} \leftarrow CheckConstraints(PC_{active}, d_{obs}, u_{k-1}, u_k, \mathbf{x_k}, \mathbf{x_{k+1}})$
      $P_{curr} \leftarrow min(P_{curr}, P_{phase})$
      if $P_{curr} \geq P_{best}$ **then**
        $T.AddChild(\mathbf{x_k}, u_k, \mathbf{x_{k+1}}, P_{curr})$
      if $LastPhase(u_k)$ **then**
        $P_{best} \leftarrow max(P_{curr}, P_{best})$
  $\mathbf{x_k}, u_{k-1}, P_{curr} \leftarrow T.DepthFirstNextNode(\mathbf{x_k})$
$U^* \leftarrow EvaluateTrajectories(x_g, T, P_{best})$
$return \quad U^*$

Algorithm 5: Prosody-aware receding horizon control with latency compensation

Input: $d_{obs,t}$, $x_{g,t}$, distance to closest obstacle and goal position, at time t.
$PC_{active} \subset PC_{replan}$ set of prosody constraints.
Output: a_t, acceleration command sent to the motors.
Notations:
$\mathbf{x_t} = [x_t, v_t]$, state of the robot at the start of the current cycle.
$u_t = [a_t, t_t]$, motion phase computed during the previous cycle, executed from the start of the current cycle.
$U^* = \{u_0, u_1 \ldots, u_{N-1}\}$, motion phases of the optimal trajectory.
Algorithm:

$u_t, u_{t-dt}, v_t \leftarrow InitializeToZero()$
$\mathbf{x_t} \leftarrow \mathbf{x_0}$
while *True* **do**
  $SendMotorCommand(a_t)$
  $d_{obs,t-delay}, x_{g,t-delay}, \mathbf{x}_{t-delay}, delay \leftarrow UpdatePerception()$
  $d_{obs,t}, x_{g,t} \leftarrow CompensateDelay(d_{obs,t-delay}, x_{g,t-delay}, a_{t-dt}, \mathbf{x_{t-dt}}, delay)$
  $d_{obs,t+dt}, x_{g,t+dt}, \mathbf{x_{t+dt}} \leftarrow ProjectForward(d_{obs,t}, x_{g,t}, a_t, \mathbf{x_t}, dt)$
  $temp\_PC \leftarrow ReceiveProsodyConstraints()$
  if $v_{t+dt} = 0$ **then**
    $PC_{replan} \leftarrow temp\_PC$
  $U^* \leftarrow PlanTrajectoryHierarch(\mathbf{x_{t+dt}}, d_{obs,t+dt}, x_{g,t+dt}, u_t, PC_{active})$ (alg. 4)
  $u_{t-dt} \leftarrow u_t$
  $u_t \leftarrow ExtractControl(U^*)$
  $DelayUntil(t + dt)$

4.8 Discussion

4.8.1 Tradeoff between prosody and flexibility

We made a methodological choice to build our robot's motion design from the ground up, starting from a heavily constrained trajectory space which only utilises a very small subset of the capabilities of the robot's hardware and flexibility of motion in space. We note that this is an active choice made to ensure that we have control over each of the motion characteristics studied in our corpus, and not just a result of a computational or algorithmic limitation. The result is that using this trajectory space to generate motions will limit what the robot is able to accomplish in terms of navigation, however we have some knowledge of how these restricted motions impact people's perceptions of the robot.

The degree of the flexibility tradeoff is tied to our current model of human perception of the robot. It seems likely that the relationship between the physical motion characteristics and human perception are more complex than what we have modeled, and especially may be dynamic with respect to other factors. As a simple example, the kinematics type dictates that the robot should move at a given maximum velocity, which we found to be related to the aggressive and gentle qualifiers. In an open environment this might make sense, but if the task of the robot is to follow a person moving at some speed which is slower then the one set by the kinematics type, our current trajectory space would force the robot to either continue moving past the person at its kinematics velocity, or successively brake and accelerate in a periodic fashion, since we do not allow a constant velocity different to the kinematics velocity. Intuitively, one would think that the robot should adapt its speed to the person's in some way, in which case one of the motion characteristics giving some control over the aggressive or gentle perception is lost. Perhaps the distance between the person and the robot, or some other variable replaces the use of the velocity in its expression of this dimension. Further studies along these lines might allow us to formulate our trajectory space differently, thus restoring some flexibility to the navigation without losing the control over the impression the robot's motion generates.

4.8.2 Switching between prosody styles

In the previous sections we presented constraints that, when combined, can alter the trajectory space explored when solving the control problem in order to produce motion corresponding to a given prosody style derived from our motion corpus. For example, if we decide that the robot should be perceived as gentle, our perception experiment results indicate that we should use the low-energy kinematics type, which is modelled by the constraints described above. Our goal in future work is for the desired impression generated by the robot to be decided and changed over time according to the robot's interaction with the person. In its current formulation, our approach assumes the prosody constraints are decided once, and fixed with respect to time, which does not allow modelling something like changing from aggressive to gentle prosody. Immediately changing the constraints from one timestep to the next may result in discontinuities in the velocity profile, or even in an

infeasible problem. For example, if the robot was applying the high kinematics constraints at time t and we change the constraints to the low kinematics at time $t+1$, the problem will have no prosody-compliant solutions if the robot's velocity was greater than the maximum imposed velocity of the low kinematics type.

In our constraints' current form, it is possible to change them when the robot is at rest without rendering the problem infeasible, but requiring the robot to stop to alter its movement prosody may be too restrictive if the interaction dynamics require fast and/or continuous adjustment of the prosody. The mechanisms of *how* the robot's prosody should be altered over time remain unclear, but in any case the fact that they *will* change should be accounted for in the formulation of the control problem. Using the same example as above, we could make the bounds on the robot's acceleration and velocities change for each timestep such that the planner can plan a deceleration so that it's velocity at the next timestep is lower: $v_{kin}^{i+1} = v_{kin}^{i} - a_{kin}^{i} * dt$, where $a_{kin}^{i} = f(i)$ is adjusted in some way when transitioning between two kinematics types (linear interpolation, for example). Another layer of complexity is added when considering that even such a gradual adjustment of one constraint may still lead to other prosody constraints being violated, such as the pause constraint. If the robot was in the middle of executing a pause phase when we initiate the change to a lower kinematics type, should it finish the pause phase before transitioning? Should it interrupt the pause phase, which would amount to violating the pause constraint? Problems will also arise if the robot is in the middle of a deceleration phase since transitioning to a kinematics type with lower deceleration would extend the stopping distance, and may render a prosody-compliant trajectory infeasible. Questions such as how to perform a prosody transition, when it is possible to do so, and how to integrate it into the forward prediction horizon should be explored in future research.

4.8.3 Constraint hierarchy

In this work, we use a simple mechanism for ordering constraints where we first attempt to plan trajectories that comply with all of the prosody constraints. If none are found, the space of possible trajectories is immediately opened up to include all collision-free trajectories. This means that even a slight violation of one constraint may drastically change the robot's motion. Using the existing approach we may separate different prosody constraints such as kinematics and variant constraints into different priority levels, hence allow a more gradual change, but the problem still remains that there is no mechanism to push the trajectories back to a space where it is once again possible to comply with the prosody constraints. Part of the issue here is the modelling of prosody, which in its current state does not provide information on whether the relationship between the variables of the prosody constraints and perception of people is linear, or has some harsh discontinuities which may mean that simply attempting to minimize constraint violations may or may not make sense.

4.8.4 Trajectory optimization method

For complex optimization problems with a large number of variables, many types of optimization algorithms and solvers can be applied. In many cases the solvers are designed to exploit certain types of problem structure arising from the forms of the cost function, system dynamics and constraints (such as linearity or convexity). Gradient-based methods also require the cost function to be differentiable, and may require the transformation of constraints into penalty functions integrated into the cost. Adapting our problem formulation to use such solvers requires care due to the somewhat unusual combination of constraint types. The kinematics constraint requires the acceleration variable to take its values in a finite set, rather than being a continuous variable as is typically the case in most robotics problems. The hesitation constraints are time-dependant, requiring a different state space formulation. Many of our constraints depend on past control inputs. We also make use of a constraint hierarchy, which is not available in all solvers. In addition, the problem formulation for the constraints and the control parameterization only reflects our current understanding of movement prosody, which will evolve as further experiments are performed. Lastly, in this work we use a basic navigation task which is easily modeled by a cost function with a standard quadratic form, but the cost function and constraints for general social navigation are likely to be much more complex, integrating many aspects of social navigation explored in other works such as time to collision(Khambhaita & Alami, 2020) or legibility of the trajectory(Dragan et al., 2013). In order to use our prosody constraints and control parameterization in more complex tasks, it may be interesting to investigate the use of sampling-based optimization methods which typically require fewer assumptions about the problem structure, such as STOMP (Kalakrishnan et al., 2011), or the more recent MPPI (Williams et al., 2016). For these reasons, we leave the adaptation of our problem formulation to typical optimization framework requirements as future work.

4.9 Conclusion

In this chapter we presented an algorithm which integrates planning and control together in a receding horizon fashion, which enables the robot to perform simple navigation tasks, while providing precise control over *how* the task is performed. We first propose a method to transform the fixed-length trajectories studied in our perception experiments into variable distance representations which allow for planning useful motions. We then present the navigation problem formalization as a constrained minimization problem, driving the robot towards a goal position subject to constraints that ensure that the robot's space of valid trajectories is defined in accordance with the trajectories explored in the perception experiments. The first problem formulation considers static environments where the trajectory can simply be planned once, and then executed blindly without feedback or re-planning. The constraints required to ensure the trajectory's compliance with the desired movement prosody in such situations are relatively simple.

Subsequently, we discuss how the constraints must be re-designed in order to address the more complex problem formulation considering dynamic environments which are not fully observable or predictable. The key idea is that the constraints must be extended to

take into account the robot's past motion history in addition to its current state in order to ensure consistency of the executed motion over time. Lastly, due to the highly restrictive constraints there may be situations where they cannot all be satisfied simultaneously, which could lead to a planning failure and subsequent motion which completely breaks the desired movement prosody. We provide examples of such situations, and propose a constraint hierarchy approach to enable the algorithm to gradually disable prosody constraints until it finds a valid plan. This enables the robot's motion to at least partially comply with the desired movement prosody, even in complex situations.

In the following chapter, we discuss the implementation of the algorithm on our mobile robot, and perform an experimental validation of the algorithm's ability to plan and execute prosody-compliant motions.

Chapter 5

Algorithm implementation and validation

5.1 Introduction

In the previous chapter we presented a local planning and control algorithm which was designed to provide explicit control over the motion variables which were found to significantly impact humans' social perception of the mobile robot. This chapter is composed of two parts. Firstly, we discuss the implementation of the algorithm on a real mobile robot, using the ROS middleware. We detail aspects of the implementation which are critical to ensuring our integrated planning and control approach can successfully plan and execute prosody-compliant motions. Secondly, we present our experimental validation of the algorithm, by demonstrating that the robot is able to reproduce the desired movement prosody based on the selection of prosody constraints which are used. We study situations in static and dynamic environments that test each of the mechanisms introduced in chapter 4.

5.2 Implementation

In the previous chapter, we described an algorithm to generate trajectories which comply with various movement prosody parameters online in a receding horizon fashion. In this section, we discuss key aspects of the practical implementation of our algorithm which are essential in order to achieve an accurate reproduction of the desired motion prosody characteristics. In some cases, the difference between two sets of prosody parameters is very subtle, thus requiring fine-grained control over the exact values and timing of the motor commands in order to ensure consistent expression of a given prosody. We first discuss the ROS (Robot Operating System) framework, and its impact on implementation choices. Then, the remaining sub-sections discuss subtle implementation details related to timing of control commands, and delays induced by sensing and computation. These details are necessary due to our strict requirements for the accuracy of the motion generation.

5.2.1 ROS Architecture

ROS (Quigley et al., 2009) is a widely-used middleware that provides a framework, tools, and packages for robotics software. Computation is structured in terms of nodes (processes) which communicate via topics by message-passing. Typically, a node will implement a specific subset of the robot's software stack. An overview of our architecture is given in Figure 5.1. The main node is the prosody-based receding horizon control node, which implements the receding horizon control approach presented in section 4.7. The node implements algorithm 2 which re-plans trajectories dynamically using algorithm 1.

The control node requires a goal position to drive towards which is provided by another node. This may be a simple command sent via a user interface to manually drive the robot to a given position, or a command sent from a high-level planner. In our diagram, we illustrate the case where the goal is given by a perception node which implements person detection and tracking, enabling the robot to drive towards a person. The person-tracking node is an existing node, which we give an overview of in section 5.2.2. The receding horizon control

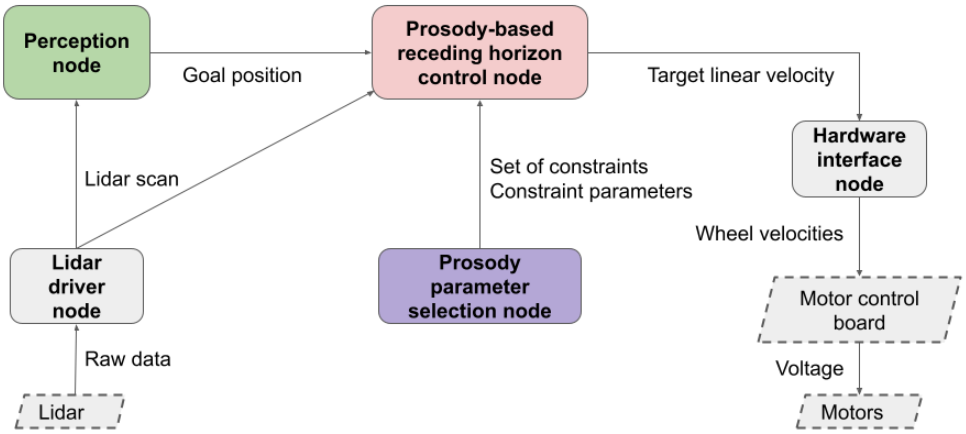


Figure 5.1: High-level architecture of our system. ROS nodes are represented with rounded boxes, hardware devices are represented with dashed boxes.

node also requires the lidar scan in order to perform the collision checking for each of the generated trajectories, which we detail in section 5.2.3.

The control node also receives the set of prosody constraints which should be applied to the trajectory generation, as well as the constraint parameters such as $a_{kin}, v_{kin}, t_{pause}$. In our current implementation, these are updated through a user interface built using the ROS *dynamic reconfigure* functionality, allowing us to save and load various configurations of the prosody constraints corresponding to the attitudes and intentions perceived by people. By separating the user interface node from the control node, we facilitate the implementation of future work where the set of active prosody constraints and their parameters will be modified online based on a computational model of the person-robot relation. The algorithm presented in the previous chapter outputs accelerations, however the low-level motor control board on the RobAIR robot only accepts velocity commands. Based on the current linear velocity, the control node computes the velocity command that matches the requested acceleration, and outputs the target linear velocity that the robot should reach at the end of the next planning cycle. Extending our control algorithm to plan angular accelerations would result in the control node also outputting an angular velocity, which the hardware interface node transforms into left and right wheel velocities via the differential drive kinematics model. Finally, these wheel velocity commands are used as input to a PID velocity regulator which controls the voltage sent to each wheel.

5.2.2 Perception module: person detection and tracking

We use an existing perception module for person detection and tracking, developed in our lab. The module takes the two lidar scans and odometry values as input, and outputs the position of the tracked person. The bottom lidar is used to detect legs, and the top lidar

is used to detect chests. The module requires the robot to be stationary for the detection phase. The main steps of the detection algorithm are to perform clustering of the points of the laser scan, followed by discarding clusters which are mostly static by comparing consecutive scans. Then, we find clusters in the bottom laser scan that have a similar size to legs, and clusters in the top laser scan with similar size to chests. Constraints on the correspondence between leg positions and chest positions allow us to determine whether there is a person. Once the person is detected, the robot may move, and tracking begins. Chest and leg candidates are generated as in the detector, and the role of the tracker is to associate one of the candidates to the person it is tracking. The tracker can operate even if one of the lasers cannot associate legs or a chest to the tracked person.

5.2.3 Prosody-based receding horizon control details

Horizon length

We chose the horizon length to be equal to the stopping time of the robot when using the high kinematics type, plus one dt, and use two motion phases. This ensures that we are able to plan a deceleration from the maximum velocity, thus satisfying the terminal constraint that the robot should be stopped at the end of each trajectory. The additional dt allows the first phase of the trajectory to take any value. This is the minimum necessary horizon length and number of motion phases in order to be able to safely plan trajectories up to the maximal velocity of the robot. This is sufficient for certain prosody constraint sets, such as smooth variant, without pauses, hesitations or increments. Adding any prosody constraints that force trajectories to be longer requires a longer horizon and more motion phases, which exponentially increases the possible combinations of motion phases, and therefore computation time. Instead of allowing the planner to use more motion phases for all trajectories, we instead alter the horizon length and number of motion phases on a per-trajectory basis. For example, when using the pause constraint, each acceleration is required to be followed by a constant velocity phase of at least $300ms$. If the previous motion phase in the trajectory currently being generated was an acceleration, we insert such a phase at that point in the trajectory, and assign a longer horizon and extra motion phase to the corresponding node in the trajectory tree. The effect is that the planner will be able to plan a trajectory consisting of an acceleration, pause, and deceleration, allowing the robot to plan a prosody-compliant motion, while still generating shorter trajectories with only two motion phases, in case the prosody-compliant trajectory is infeasible. In other words, we add flexibility to the trajectories which comply with prosody constraints, and plan simpler, shorter trajectories which are collision-free as fallbacks.

Trajectory distance resolution

Due to discretization of the phase lengths and the restriction to a single deceleration value, the robot can only plan trajectories that stop at discrete sets of positions in the environment. The resolution of the grid formed by possible stopping locations is related to the maximum velocity reached by the robot during its motion. For example, when moving at $0.8m.s^{-1}$,

the robot can plan to decelerate immediately, or plan a short constant velocity phase with a duration of one dt, extending its motion by $8cm$. Whichever one of these is closer to the goal position will be selected for execution. During its deceleration, the additional distance travelled by performing a single dt constant velocity phase decreases. It is possible that at some point, interrupting the continuous deceleration with a small constant velocity phase would result in the robot stopping closer to the desired goal location. Depending on the prosody constraints, such a trajectory may be discarded. However there remains the issue that once the robot stops, none of the prosody constraints will prevent it from planning a very short additional motion, since we want the robot to be able to perform small changes in position. In order to avoid such artifacts, we force the robot to keep decelerating, only allowing it to plan other motions if the goal position has changed, or the obstacle which was blocking it has moved. This ensures that a continuous deceleration will not be interrupted simply due to the time discretization, and the robot will not plan an additional short motion after having stopped. In our implementation, we use a constant threshold of $0.35m$ for the amount of displacement of the goal or obstacle required to allow the forced deceleration constraint to be lifted.

Collision checking

We implement a simple approach to check whether a given robot position is in collision by approximating the robot's footprint as a circle of radius $r_{robot} = 0.35m$. The lidar scan consists of 724 points spread over its $240°$ field of view which we downsample to 362 points by keeping the nearest point of each pair of points in the original scan. We exhaustively check the distance d_{hit} between each scan point and the robot's position. This process is repeated for each node in the trajectory, after transforming the scan points into the coordinate frame of the robot at its projected position. With this implementation and given the large number of nodes generated by our planning approach, collision checking represents roughly 70% of the total computation time. This could be vastly improved by representing the lidar scan as a bounding volume hierarchy, resulting in logarithmic time complexity (in the number of lidar points) rather than linear. One may also use the ROS costmap package which builds an occupancy grid representation of the space around the robot, however in our brief tests we were unable to configure it to simultaneously provide a resolution of less than $5cm$, over the full $5m$ range of our lidar, while meeting our performance requirement of running at $10hz$.

A trajectory is considered to be in collision if there exists a node at which at least one scan point has $d_{hit} < r_{robot} + d_{margin}$. d_{margin} is a constant safety margin which we set to $0.1m$. This condition implements the higher priority collision constraint in our constraint hierarchy. The lower priority collision constraint uses a more conservative condition $d_{hit} < r_{robot} + d_{margin} + d_{tolerance}$, where $d_{tolerance} = 0.1m$. This value was selected by accounting for the lidar sensor's measurement uncertainty, as well as the error in the forward prediction model. Once the tolerance is calibrated appropriately[1], it ensures that the execution of prosody-compliant trajectories is possible in static environments.

[1]Note that the collision tolerance may interfere with the ability to reach a goal point if it is located near an obstacle. Hence, one must also consider the trajectory resolution when designing the collision tolerance.

Latency compensation

We now discuss the impact of our node update frequencies and computation delays on the accuracy of motion planning. Our ROS nodes all run at a fixed frequency of $10hz$, which is the same as the acquisition frequency of the lidar sensor. The only exception is the low-level hardware interface node running on an Arduino Mega, running at $40hz$. The node update cycles are not synchronized, so in the worst case our control node may receive a lidar scan which was sent almost $100ms$ ago by the lidar driver. Additional delays are introduced further upstream by the lidar driver computation, serial communication with the lidar and most importantly, the actual scanning time of the lidar, which is also $100ms$. All things considered, in the worst case the control node could be planning based on data which is up to $200ms$ old which, even with our robot's relatively slow maximum velocity of $0.8m.s^{-1}$ equates to an offset of $0.16m$. In addition to the perception delay, the time taken to determine the optimal control action, and transmit the command through the hardware interface and serial communication to the motors also introduce an offset. Since we send the velocity command at the start of the next planning cycle, the control node introduces a constant delay of $100ms$, and we estimate the remaining delay through the low level controller to be roughly $50ms$.

In a typical mobile robotics application, these delays would simply lead to some inaccuracies in the robot's positioning and ability to stop at an exact distance from obstacles. Since typical control approaches allow the full use of the robot's motor capabilities, the planner could compensate for the overshoot by decelerating more. Once the planner reaches the maximum deceleration, the robot will simply keep decelerating with the same consistent deceleration profile. As long as one chooses the minimum imposed collision tolerance to be higher than the distance travelled at maximum velocity over the latency duration, the robot will not collide with anything. In our case, the planner will attempt to plan using the full prosody constraints, but it cannot perform slight adjustments to the deceleration value to compensate for unaccounted delays. Instead, the planner can either search for a trajectory that violates the prosody constraints or, if that fails, it will perform an emergency stop applying the maximal robot deceleration, completely violating the prosody style. Thus, for our application it is important to minimize the impact of these delays.

Firstly, we handle the delays related to the lidar data. The lidar device provides a timestamp $t_{acquisition}$ giving the time of the acquisition of the first hit of the scan. When receiving the scan in the control node, we compute the age of the lidar data dt_{age} based on the difference between the timestamp and current time. The lidar points are then transformed by integrating the robot's velocity over the last dt_{age} seconds. Secondly, we handle the delays related to the control computation time. Here, the lidar points are transformed based on the velocity which will be applied from the present time to the start of the next planning cycle.

In this section, we presented the ROS implementation of our algortihm, as well as its integration with the robot hardware and perception modules. We detailed aspects of the implementation which are important in order to maintain accurate control over the robot's motion so that it produces the expected motion characteristics. In the next section, we demonstrate that our algorithm and implementation on the real mobile robot successfully

produces motion which conserves the distinctness of the different movement prosody types defined by the motion corpus variables from chapter 3.

5.3 Implementation validation

In this section, we demonstrate the ability of our integrated planning and control algorithm to produce trajectories which accurately reproduce the different types of movement prosody defined by the combination of corpus variables. We first demonstrate the performance in open spaces, thereby testing the definition of the prosody constraints. We then demonstrate the performance in situations where there are obstacles, testing the replanning, constraint hierarchy, and latency compensation mechanisms.

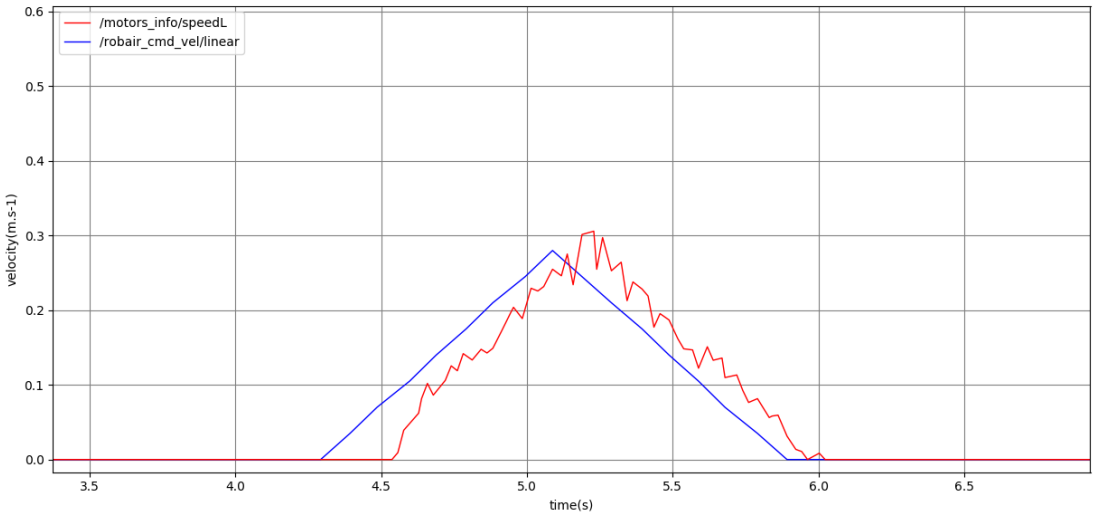
5.3.1 Generation of prosody-compliant motions in open space

In this section, we demonstrate our planner's ability to plan simple motions towards an unobstructed goal position, while complying with the prosody constraints. Plots of the velocity commands from our planner show that they are stable and consistent with the desired prosody. We also plot the raw encoder-based velocity estimation, showing that the commanded velocities are indeed achievable by our robot platform, thanks to our planner and prosody constraints taking the robot's mechanical limits into account. Unless stated otherwise, the prosody used in these examples are the medium kinematics, smooth variant, no pauses, no hesitations.

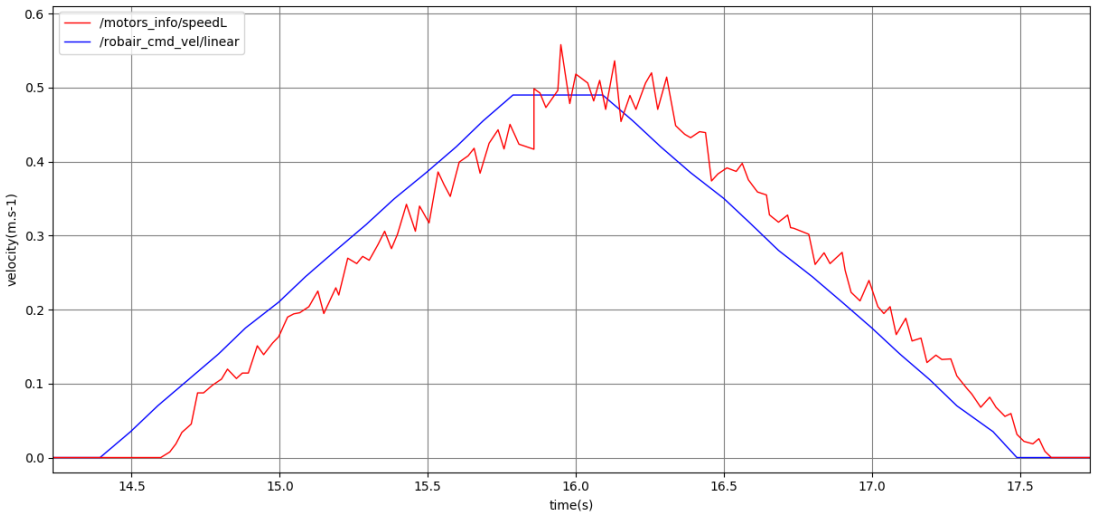
Kinematics

The three kinematics types (low, medium, high) require different accelerations, and different maximal velocities. We show examples of motions produced by running our planner with each of the kinematics types. Figure 5.2 shows two motions planned with the medium kinematics type over different distances. A short motion is shown in 5.2a, where the distance to be travelled is short enough that the robot does not reach the medium kinematics limit of $0.49m.s^{-1}$, instead decelerating when reaching just $0.37m.s^{-1}$. In contrast, Figure 5.2b shows a longer motion, leaving the robot enough time to reach the limit of $0.49m.s^{-1}$, at which point the kinematics velocity constraint renders further acceleration invalid. Decelerating immediately would make the robot undershoot, and so the trajectory evaluation leads the planner to select a trajectory which maintains the maximal velocity for a short period of time before decelerating. In both cases, the slope of the commanded velocity profile is constant, and corresponds to the acceleration of $0.35m.s^{-2}$. In the figures, we also show the raw estimate of velocity based on the integration of the motor's encoder readings over time, in order to demonstrate how the physical robot platform responds to the velocity commands. Overall, the measured velocity matches the commanded velocity, although the unfiltered estimates are quite noisy despite the real motion being smooth. We chose to show unfiltered sensor readings since they better demonstrate the motor's fast response time when the commanded velocity changes between different slopes, such as the transition

from stopped to accelerating, or from accelerating to decelerating. The time offset between the commanded and estimated velocities is the sum of the time for a command to be sent to the hardware interface, for the motor controller to achieve the requested velocity, and for the estimation to be computed and sent back to the main computer running our planner.



(a) Maximum velocity not attained during a short motion.



(b) Velocity limited to the maximum medium kinematics velocity over a long motion.

Figure 5.2: Plot representing the full point-to-point motion to a goal point, using medium kinematics. Past command velocities issued at $10hz$ (blue) and unfiltered encoder-based odometry estimated at $40hz$ (red). Top: short motion. Bottom: long motion.

Figure 5.3 shows a short motion with the low kinematics. The goal point is close enough that the robot only accelerates to $0.20m.s^{-1}$, slightly below the low kinematics maximum of $0.24m.s^{-1}$. The slope of the commanded velocity profile corresponds to the low kinematics acceleration of $0.2m.s^{-2}$ as expected, and the estimated velocity also follows the commanded velocity closely. The resulting motion can clearly be distinguished from the medium kinematics shown previously.

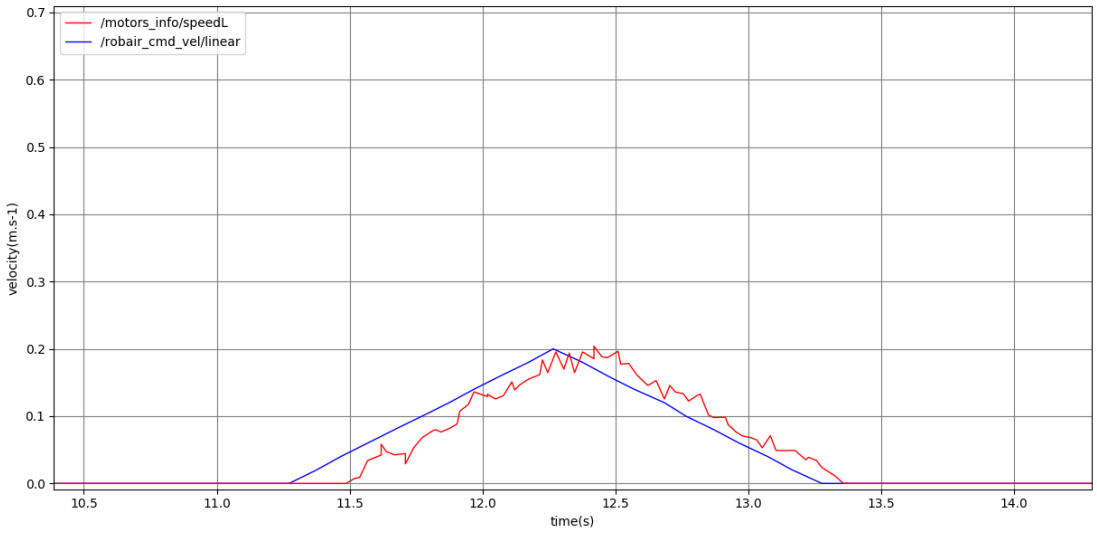


Figure 5.3: Plot representing the full point-to-point motion to a goal point, using low kinematics. Past command velocities issued at $10hz$ (blue) and unfiltered encoder-based odometry estimated at $40hz$ (red).

Figure 5.4 shows a short motion with high kinematics. Again, the goal point is close enough such that the robot does not need to accelerate to the maximum high kinematics velocity of $0.72m.s^{-1}$. The robot accelerates to $0.65m.s^{-1}$, with an acceleration of $0.5m.s^{-2}$, clearly distinguishing the motion from the low and medium kinematics.

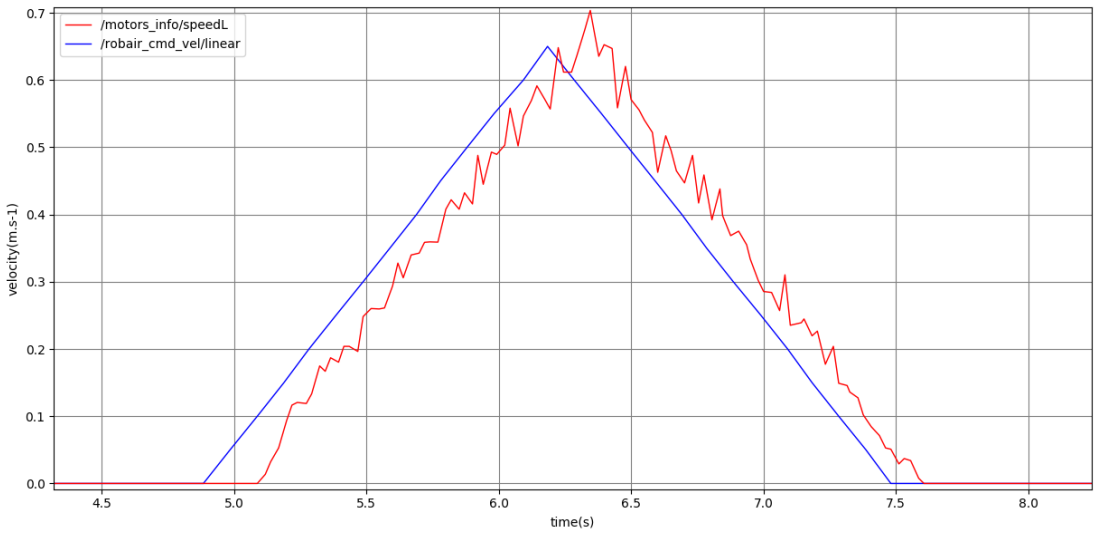


Figure 5.4: Plot representing the full point-to-point motion to a goal point, using high kinematics. Past command velocities issued at $10hz$ (blue) and unfiltered encoder-based odometry estimated at $40hz$ (red).

Pause constraint

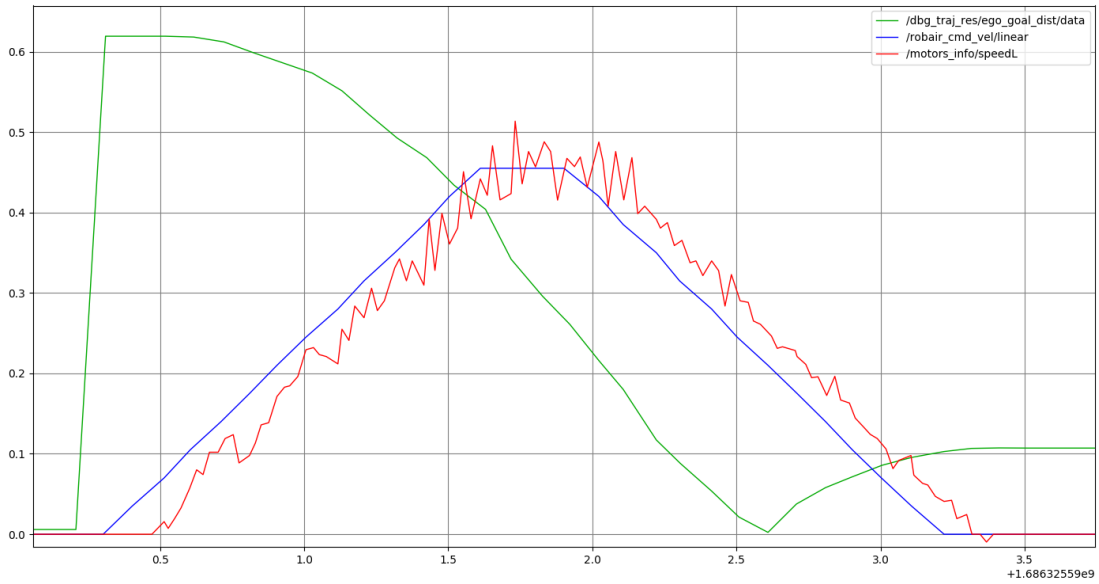
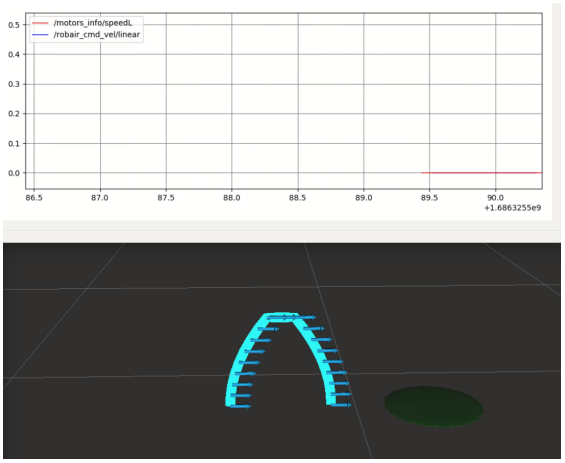


Figure 5.5: Plot representing the full point-to-point motion to a goal point. Past command velocities issued at $10hz$ (blue) and unfiltered encoder-based odometry estimated at $40hz$ (red), both given in $m.s^{-1}$. Distance to the goal, estimated at $10hz$, in m (green). All quantities are plotted w.r.t. time (s).

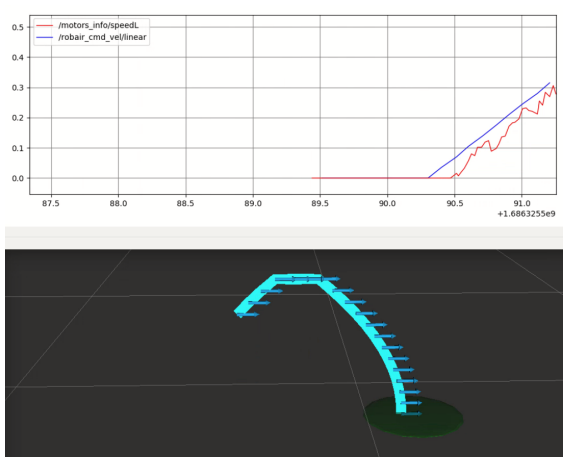
Figure 5.5 shows the plot of the robot's velocity, and distance to the goal during a point-to-point motion to a goal placed at $62cm$ from the robot, without obstacles. The active prosody constraints are the medium kinematics type, smooth variant, and pauses. The plans generated by the controller result in a velocity profile that conforms to the prosody constraints, a linear acceleration and deceleration phase, separated by a pause phase of $300ms$, and drives the robot towards the goal point. The controller induces a slight overshoot of the goal position, passing the goal at $t = 2.6s$ and stopping at $11cm$ past the goal, despite the fact that there exists a prosody-compliant trajectory accelerating to a slightly lower velocity that would have arrived closer to the goal. Our focus in this work is not on achieving extremely precise positioning, however we briefly discuss the causes for the overshoot and possible solutions. Firstly, our simple cost function formulation treats all nodes of the trajectory equally, meaning that a trajectory that maintains higher velocity before reaching the goal and overshoots slightly will in fact result in a lower overall cost than a trajectory that maintains a lower velocity before the goal and does not overshoot. The introduction of a terminal cost, adding a high cost to the deviation of the last position of the trajectory from the goal position could compensate for this issue. Secondly, our latency and delay compensation does not perfectly model all of the lower-level latency for communication with the motors, which also introduces some inaccuracy. Lastly, the time-discretization of the motion phases implies that for certain goal positions there will always be some unavoidable under or overshoot.

In Figure 5.6, we show how the trajectory is planned over time for a pause sequence[2]. The top part of the figures are graphs of the robot's past linear velocity plotted with respect to *time*. The lower parts of the figures are visualisations of the robot's future linear velocity according to its current plan (light blue), plotted with respect to the robot's *position* rather than time. Therefore, the horizontal position of the leftmost blue arrow corresponds to the robot's current position, and the vertical positions of arrows indicate the robot's planned velocity along the trajectory (higher being faster). In Figure 5.6a, the robot is stationary and plans a prosody-compliant trajectory which utilises the whole planning horizon in order to move the robot closer to the goal position to the right of the figure. The plan includes a constant velocity phase between the acceleration and deceleration phase, due to the application of the pause constraint. This first plan has an acceleration phase with a duration of only $8dt$, which does not reach maximum velocity due to the horizon time being limited to $16dt$. In Figure 5.6b, the robot has already accelerated to $0.3m.s^{-1}$, and is thus able to plan to continue accelerating to a higher velocity than its prior plan. The plan is also able to reach a position slightly before the goal position. In Figure 5.6c, the robot has changed the first motion phase from an acceleration to a constant velocity phase in order to perform the pause required by the prosody constraints. This plan reaches a position almost exactly at the goal position. In Figure 5.6d, the robot has performed part of the pause phase, and selects a trajectory that finishes the pause phase by planning the first phase as a constant velocity over $1dt$, which results in a slight overshoot of the goal position. This overshoot is due to some remaining mismatch in the forward prediction and latency compensation, and the real robot system response. We note that immediately decelerating would have led to the robot arriving closer to the goal, (hence, such a trajectory has a lower cost), however this trajectory was not selected by the planner, since it would violate the pause constraint. Figures 5.6e and 5.6f show the deceleration and final position of the robot, respectively. The robot stops within the $20cm$ radius defined around the goal position, which corresponds to the trajectory resolution.

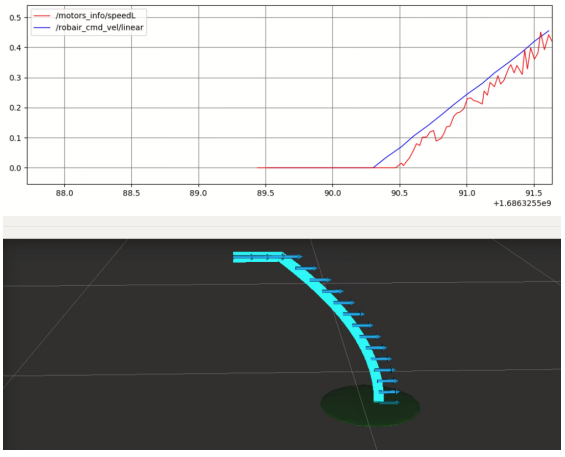
[2]Video for Figure 5.6 (slower than realtime for clarity): https://cloud.univ-grenoble-alpes.fr/s/f5G8kQR4rMx6MWi



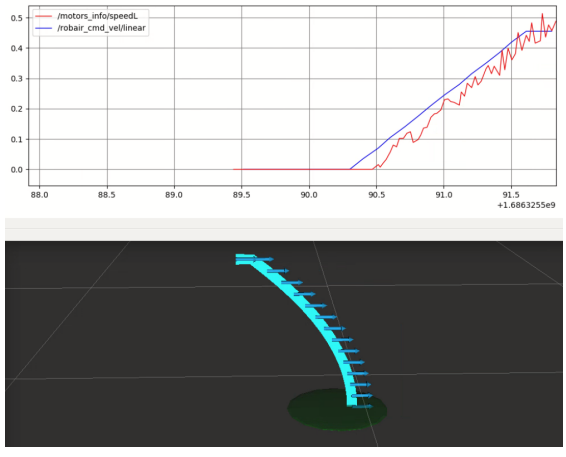
(a) Stationary robot.



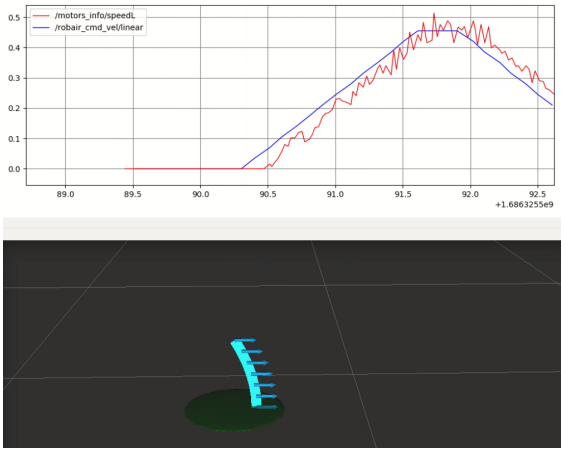
(b) Mid-acceleration plan.



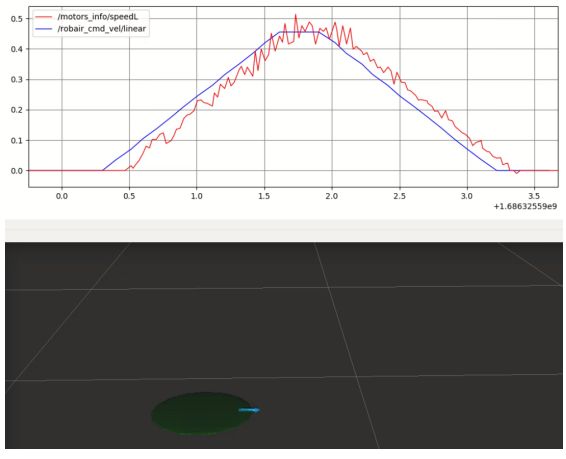
(c) Switching to pause phase.



(d) Finishing pause phase.



(e) Mid-deceleration plan.



(f) Stopped close to goal.

Figure 5.6: Top: past command velocities issued at $10hz$ (blue) and unfiltered encoder-based odometry estimated at $40hz$ (red)in $m.s^{-1}$, plotted w.r.t. time (s). Bottom: visualisation of the planned velocity w.r.t. position, arrows indicate discretization of the motion phases into time intervals of length $dt = 100ms$. The robot goal position is at the center of the green disk, with radius $20cm$. One grid square represents one meter.

Increment and saccade variants

In this subsection, we demonstrate motions planned under the increment or saccade variant constraints. Figure 5.7 shows a long increment motion, allowing the robot to reach the maximum velocity for the medium kinematics type. Figure 5.8, shows a short saccade motion, without pauses.

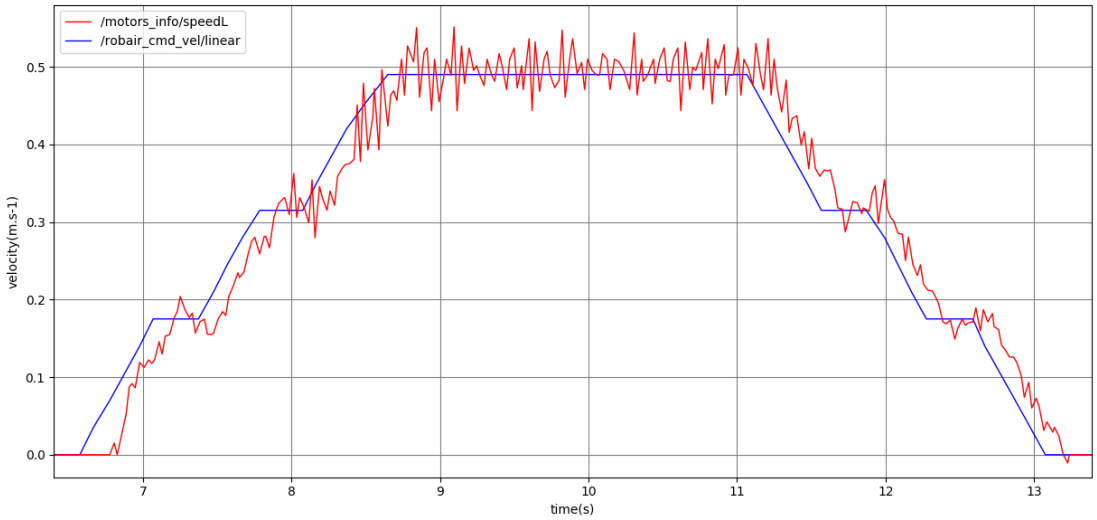


Figure 5.7: Point-to-point motion using the increment variant and medium kinematics. Command velocities issued at $10hz$ (blue) and unfiltered encoder-based odometry estimated at $40hz$ (red).

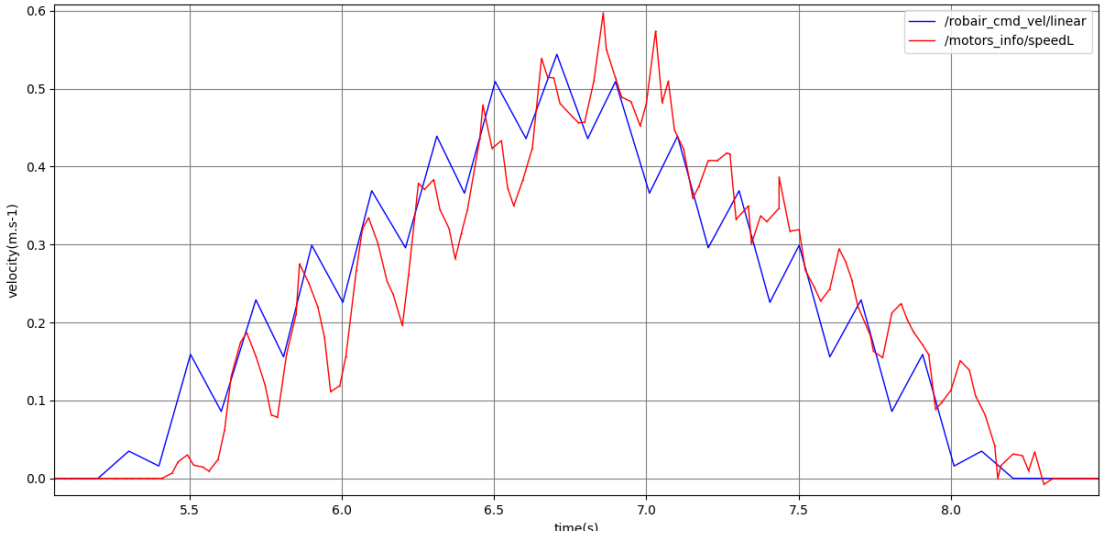


Figure 5.8: Point-to-point motion using the saccade variant and medium kinematics. Command velocities issued at $10hz$ (blue) and unfiltered encoder-based odometry estimated at $40hz$ (red).

5.3.2 Generation of prosody-compliant motions with static and dynamic obstacles

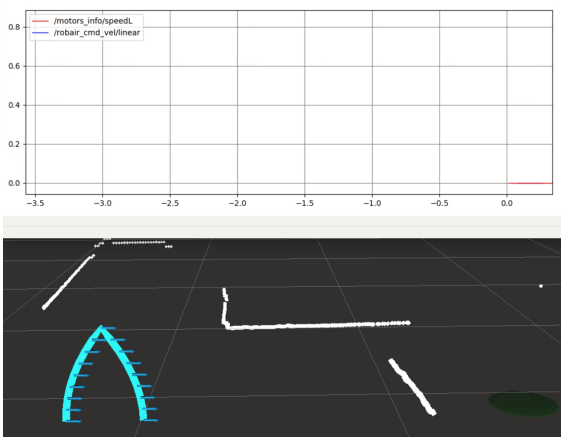
In this section, we demonstrate the impact of the constraint hierarchy and latency compensation on our planner. We first show how both mechanisms allow us to plan consistent prosody-compliant trajectories despite sensor noise and latency. Second, we show how the constraint hierarchy enables safe re-planning to avoid collisions, while maintaining part of the movement prosody.

Reducing and compensating model mismatch

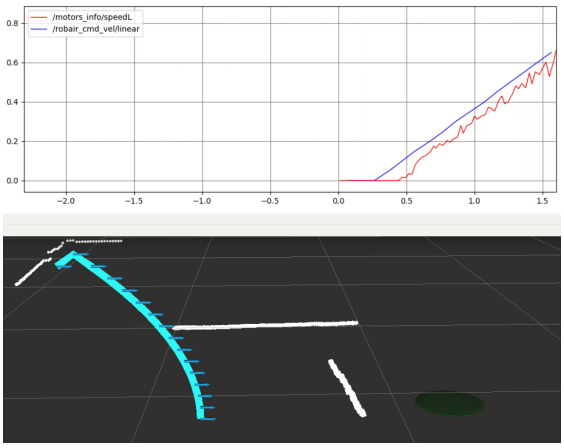
Without applying latency compensation for perception and computation time, the mismatch between the forward planning and real system can render previous plans infeasible. Due to the strict prosody constraints, this often results in the planner resorting to using an emergency braking which completely disregards all prosody constraints. We show such an example in Figure 5.9, where the initial plan for a smooth deceleration phase with medium kinematics is interrupted[3]. Since the planner cannot find any other collision-free trajectories, a deceleration which violates the kinematics constraint is planned. After reducing its velocity, the planner is once again able to find a prosody compliant trajectory, and plans an additional short acceleration and deceleration motion. The resulting motion of the robot is significantly different from the desired single smooth deceleration phase which was initially planned. When applying the latency compensation as well as the two-level collision constraint hierarchy, the robot is able to plan **and execute** a prosody-compliant motion[4], as illustrated in Figure 5.10. The latency compensation allows us to reduce the model mismatch between the trajectory planning and real execution, and the collision constraint hierarchy compensates for the remaining mismatch.

[3]Video for Figure 5.9 (slower than realtime for clarity): https://cloud.univ-grenoble-alpes.fr/s/swNs5HyW7fmGFZH

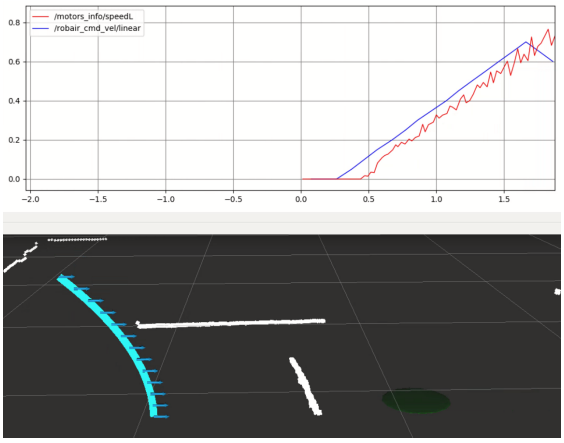
[4]Video for Figure 5.10 (slower than realtime for clarity): https://cloud.univ-grenoble-alpes.fr/s/b2iAA2mGibXbENd



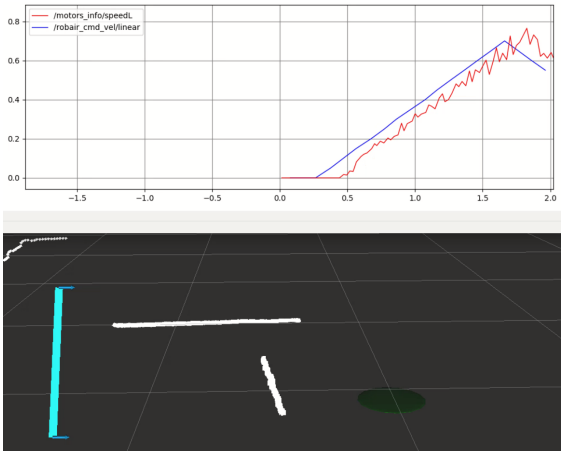
(a) Stationary robot, initial plan.



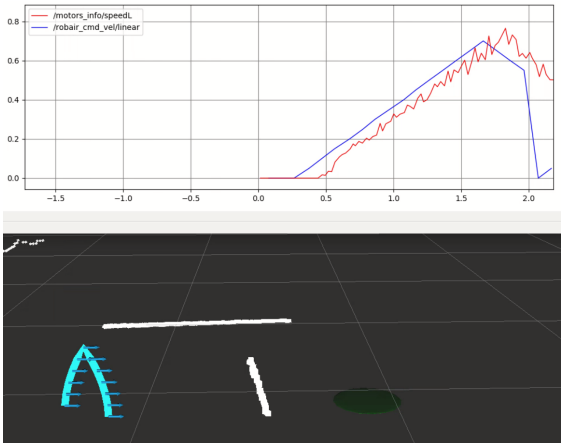
(b) Ending acceleration phase.



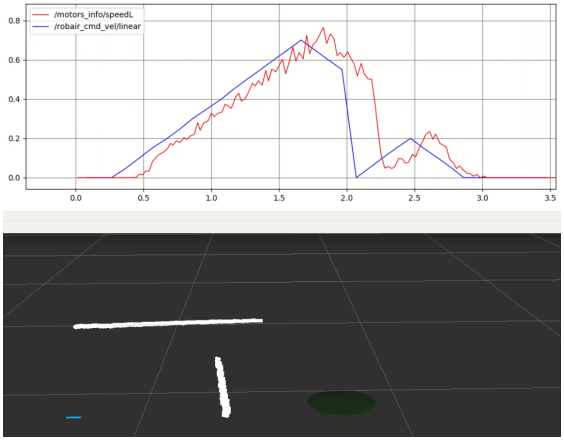
(c) Collision-free deceleration planned.



(d) Prior plan invalidated.



(e) An additional motion is planned.



(f) Motion violates the prosody.

Figure 5.9: Failure to execute prosody-compliant motion in the presence of a static obstacle, without the latency compensation and two-level collision constraint hierarchy. Top: past command velocities (blue) and odometry (red) in $m.s^{-1}$, plotted w.r.t. time (s). Bottom: visualisation of the planned velocity w.r.t. position. The robot goal position is at the center of the green disk. White points represent objects in the environment detected by the 2D lidar on the robot's base.

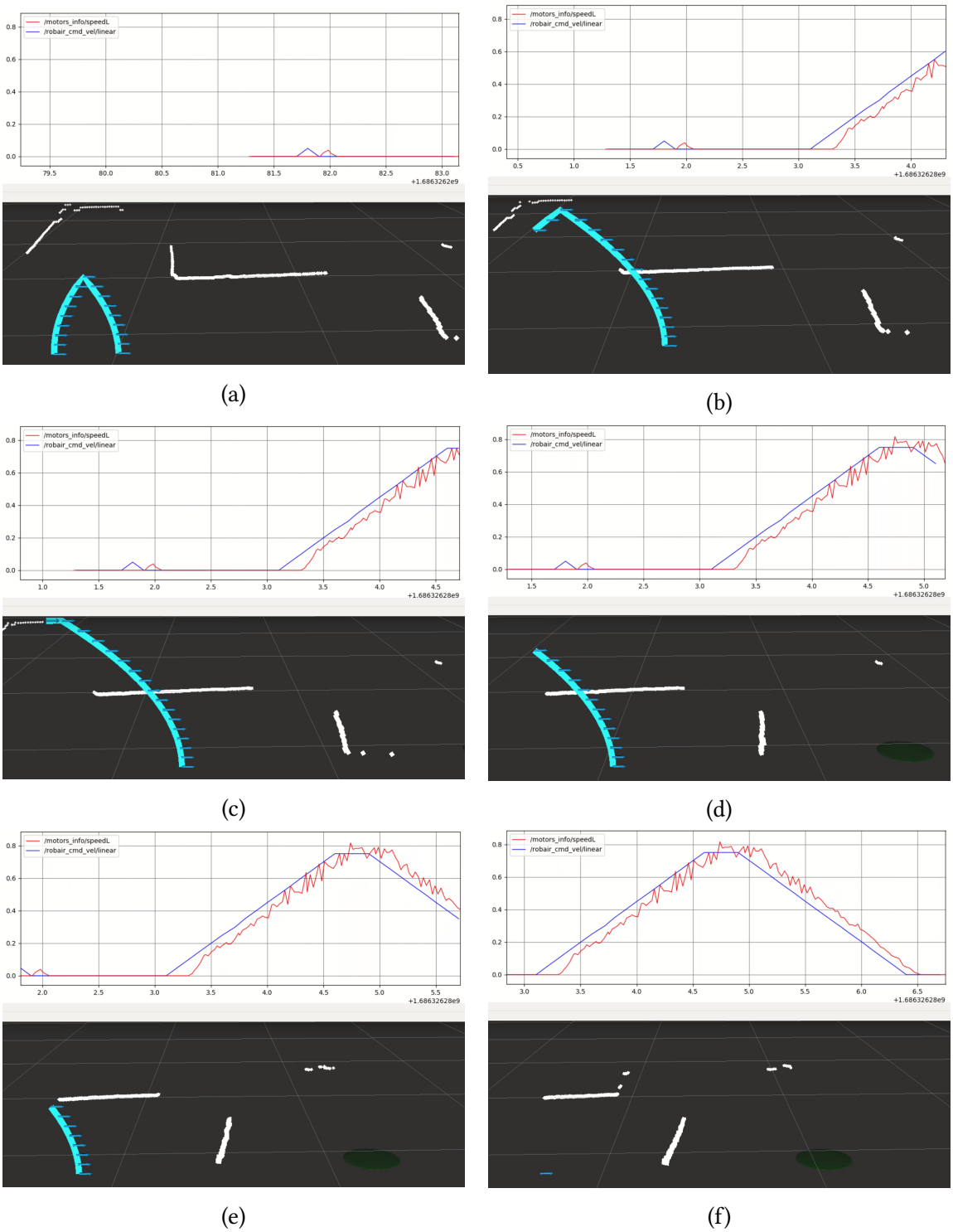


Figure 5.10: Successful planning of prosody-compliant motion in the presence of a static obstacle, using the latency compensation and two-level collision constraint hierarchy. Top: past command velocities issued at $10hz$ (blue) and odometry estimated at $40hz$ (red) in $m.s^{-1}$, plotted w.r.t. time (s). Bottom: visualisation of the planned velocity w.r.t. position. The robot goal position is at the center of the green disk, with radius $20cm$.

Collision avoidance and prosody tradeoff

In situations where people or other dynamic agents in the environment have unpredictable behaviour, we must consider the tradeoff between respecting all prosody constraints, or avoiding collisions. Note that we are still discussing only linear motions, hence *avoidance* here entails adapting the length of the trajectory, not rotating away. In our current constraint hierarchy formulation, we attempt to maintain all prosody constraints, but allow the planner to violate some of them in order to find collision-free trajectories. We use the hesitant prosody constraint as an example, since executing a hesitation significantly extends the length of a motion when compared to an immediate deceleration phase which only satisfies the kinematics constraint. Figure 5.11 shows a typical execution of a non-obstructed point-to-point motion using the medium kinematics and hesitant prosody, with a short hesitation duration of $6dt$ ($600ms$).

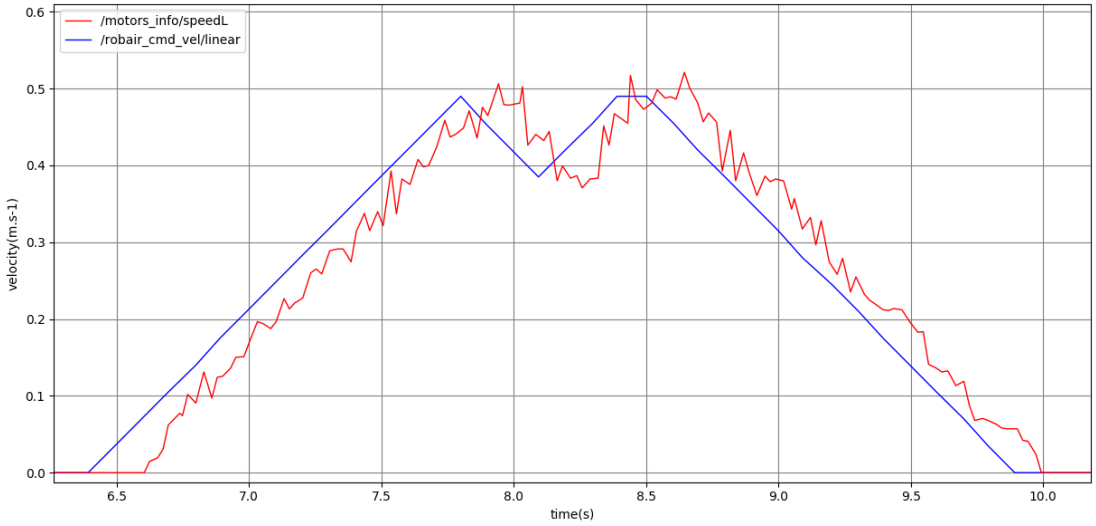
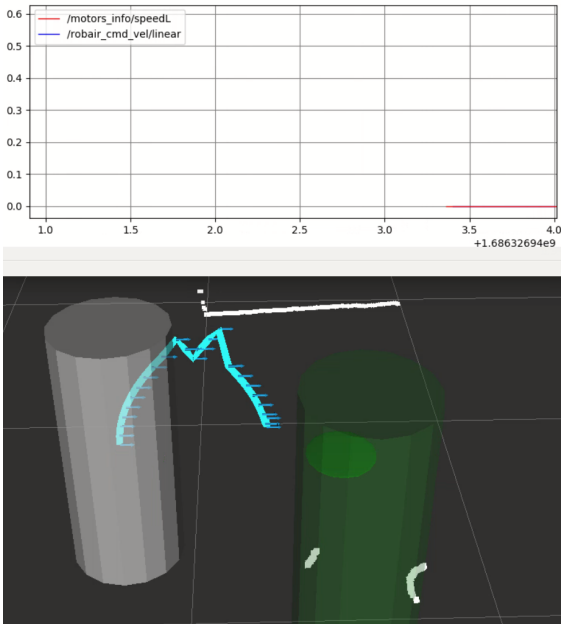


Figure 5.11: Point-to-point motion using hesitations and medium kinematics. Command velocities issued at $10hz$ (blue) and unfiltered encoder-based odometry estimated at $40hz$ (red).

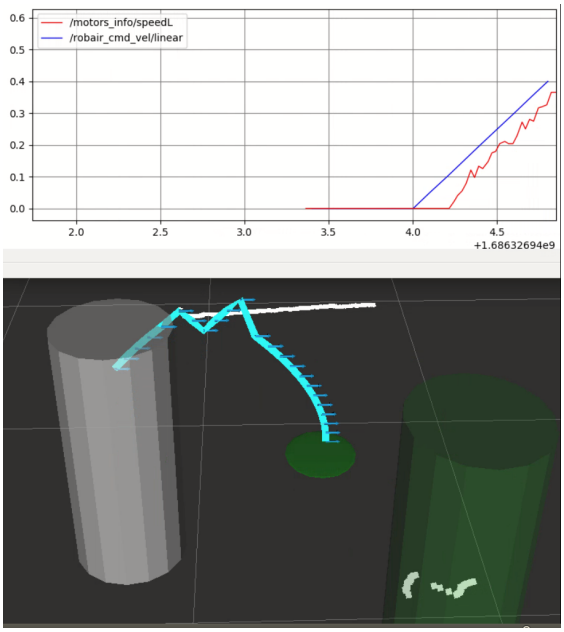
In Figure 5.12 we demonstrate the planner's response to a person suddenly stepping in front of the robot, just as it begins its hesitation[5]. In order to fully comply with the prosody constraints, the planner should plan to finish the hesitation deceleration phase, accelerate back to the peak velocity, and only then decelerate. Such a trajectory would result in a collision with the person, hence there are no collision-free prosody-compliant trajectories. Due to the constraint hierarchy approach, the planner checks trajectories that violate the hesitation constraint, but satisfy the collision avoidance. The planner determines that an immediate deceleration would avoid collision with the person, and selects that trajectory for execution (Figure 5.12c). The robot is able to stop while complying with the high kinematics constraint, partially complying with its set of prosody constraints. If we had not used a

[5]Videos for Figure 5.12:
(slower than realtime for clarity): https://cloud.univ-grenoble-alpes.fr/s/F86zTYwojJtmfDd
(realtime) https://cloud.univ-grenoble-alpes.fr/s/ECkGkGCnPDqRfSm

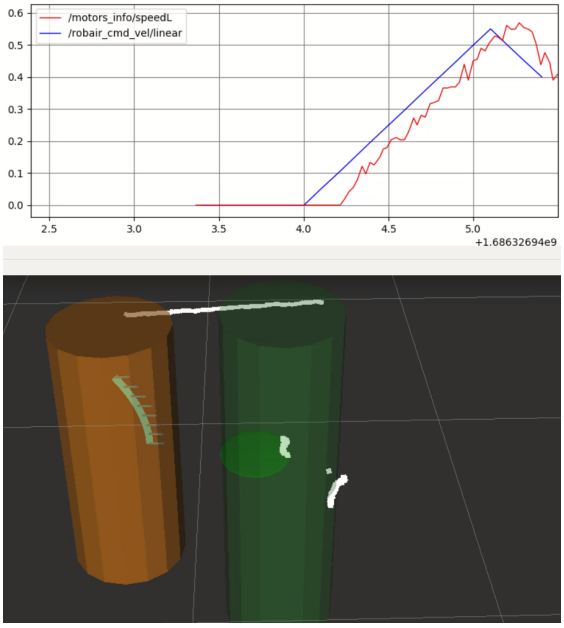
constraint hierarchy, and only considered trajectories which always satisfy all constraints, the planner would have resorted to performing an emergency braking with much harsher deceleration, violating both the hesitation constraint **and** the kinematics constraint.



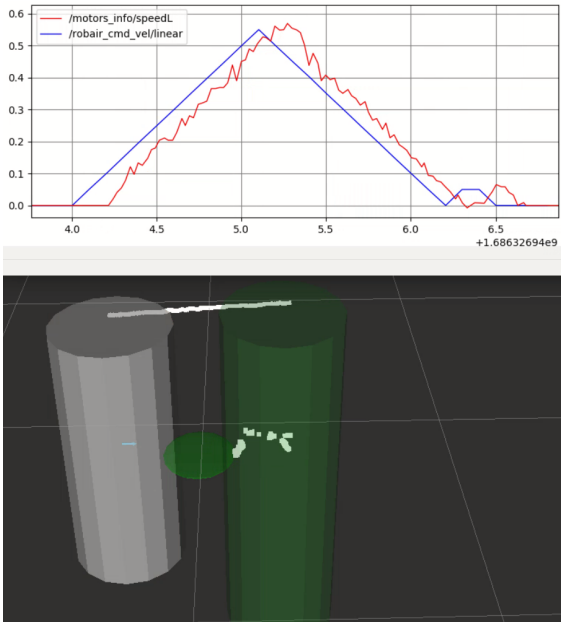
(a) Initial plan includes hesitation.



(b) Horizon reaches far enough to reach the goal, with a hesitation.



(c) Person steps in front at the moment when the robot should accelerate again. Instead, we plan a deceleration to avoid collision, violating the hesitation constraint.



(d) The robot stops without colliding, and while satisfying the kinematics constraint.

Figure 5.12: Interrupting a hesitation in order to avoid collision. Top: past command velocities issued at $10hz$ (blue) and odometry estimated at $40hz$ (red) in $m.s^{-1}$, plotted w.r.t. time (s). Bottom: visualisation of the planned velocity w.r.t. position. Robot (gray or orange cylinder), person (green cylinder). Orange cylinder indicates the robot is executing a motion that does not comply with all prosody constraints. The robot goal position is at the center of the green disk.

5.4 Conclusion

In this chapter, we first provided details of the implementation of our prosody-based receding horizon control algorithm using ROS, including precise details which are necessary given the very strict definition of the desired robot motion.

We then showed the ability of our proposed algorithm to both plan and execute straight-line motions towards goals while strictly adhering to the selected prosody constraints. We showed that the velocity commands reproduce the desired prosody, and also that the robot platform's low-level motor controller is able to reproduce the commanded velocities with good accuracy and fast response time. We demonstrated the necessity of accounting for computation and communication latency in the planner, combined with the hierarchical constraint approach, showing that the desired prosody cannot be consistently achieved without these mechanisms. We also demonstrated the utility of the constraint hierarchy for maintaining partial compliance with prosody constraints when full compliance becomes infeasible due to unpredicted dynamic obstacle motion in the environment.

Overall, we conclude that our implemented algorithm is capable of generating appropriate velocity commands in real-time, and that our robot platform is able to accurately track the requested velocity commands such that the resulting motion of the robot corresponds to the chosen movement prosody, in accordance with our experiments in chapter 3.

In the following chapter, we present a HRI study for which we develop a full software architecture with perception and high-level decision nodes interacting with our proposed control node described and evaluated in this chapter. The robot is deployed with our algorithm in a real interaction setting, in order to study the impact of the various combinations of movement prosody constraints on people's perception of the robot and their reaction to its motion.

Chapter 6

Integration of our algorithm into a full architecture to study its impact in an *in the wild* HRI experiment

6.1 Introduction

In this chapter we present a human-robot interaction study where the robot performs a real task in an uncontrolled environment, with naive participants. The robot is given the role of a flyer distributor, approaching people in the entrance hall of a building on a university campus. The robot performs the approach motion using two distinct prosody styles derived from the results of the online and lab-condition experiments described in chapter 2, which are the *confident* and *hesitant* styles. An initial version of the experiment was conducted by teleoperating the robot, and the second version made use of our proposed prosody-aware local planner described in chapter 3.

Several research questions are explored through this experiment. First of all, we aim to determine whether or not participants perceive the two prosody styles in a similar manner to the online and lab experiments where the robot was not performing an actual task, and had no role. Second, we aim to study how participants react to the different movement prosody styles both in terms of their physical behaviour, and their self-reported impressions of the robot. Lastly, we aim to study whether the movement prosody impacts participants' ratings of the robot's performance.

We first detail the motivations for this experiment, and review similar related studies in human-robot interaction. Second, we present the experiment design, protocol, and measures. We also give an overview of the implemented software architecture that integrates our algorithm designed in chapter 4, alongside our perception, localization, and high-level decision modules. Thirdly we present the results and perform a statistical analysis based on the post-interaction semi-structured interviews, as well as video, audio, and lidar data from the interactions themselves. Lastly, we discuss the findings of this experiment while considering their relation to our prior experiments in chapter 3.

6.1.1 Motivation

The online perception experiments were designed such that the participants are distant observers, bystanders, without any engagement. For the in-person perception experiment there was a physical engagement due to the physical embodiment, and given the robot's motion is directed towards the person (potential for collision). However, there was no reason for the robot to move towards the person, or for the person to pay attention to the robot other than their participation in this artificial, lab-style experiment.

The general goal of this new experiment was to study how people perceived and reacted to differences in the robot's movement prosody in an unplanned interaction with a credible task involving the person. The first goal was to determine whether our mapping between physical motion dimensions (velocity, acceleration, smoothness, saccades, pauses) and human perception of attitudes and socio-affects would still hold in an ecological interaction setting where the robot has a true role and task to perform. This would allow us to confirm or adapt our model of movement prosody. The second goal was to explore whether the movement prosody employed by the robot could impact participant's perception of the robot's performance in the task. Lastly, we wanted to explore whether the robot's move-

ment prosody could impact people's own physical behaviour.

6.1.2 Real-world HRI experiments

Studies evaluating the impact of mobile robot motion parameters are often conducted in lab experiments as in (Lohse et al., 2013; Saerbeck & Bartneck, 2010) and in our own lab study. In some cases, participants are asked to watch the robot perform an actual task (Schulz et al., 2020), before answering questionnaires about their perceptions of emotion, affect, or preferences regarding the motion. Instead, we aimed to perform an *in the wild* study where the robot is simply deployed in a populated environment without having recruited participants beforehand. This typically can allow for higher ecological and external validity than lab studies (Hoffman & Zhao, 2020), since participant behaviour can be influenced by the different setting. Our choice of measures is also different, since our studies presented in chapter 2 showed us that directly asking people to attribute various adjectives to the robot was a difficult task for many participants. Instead, we proceed by designing a believable pretext task for the robot to perform (in this case, approaching people to hand out flyers, see section 6.2.1), and initiating a semi-structured interview which gradually takes the discussion from general impressions of the robot, to performance on the pretext task, and finally to revealing our true focus on the robot's movements.

Prior works on the specific HRI task of approaching a person have extensively studied the distance at which a person should be approached (Joosse et al., 2020), as well as different velocities and paths (Brandl et al., 2016; Mizumaru et al., 2019). These works typically are concerned with determining what the ideal approach strategy should be, whereas we are simply using this task as a pretext, and as such we are only interested in whether distinct movement prosody parameters from our corpus will influence people's impressions and reactions to the robot. This experiment focuses on testing the results of our initial experiments for the *confident - hesitant* perceptual scale, which was found to be one of the strongest results, and involved all of the motion corpus variables. The *hesitant* impression was associated with saccadic velocity profiles and periodic decelerations, which have not yet been studied in the context of mobile robot approach. Studies on the concept of hesitation motions are rare, and are their goal is to design a gesture that can be used to demonstrate hesitation at a specific moment, rather than a set of motion features that make the robot's motions overall appear hesitant. For example, in (Reinhardt et al., 2021), a "back-off" motion is designed in order for a mobile robot to show the intent of yielding at an intersection, and in (Moon et al., 2013), a hesitation motion is designed for a robot arm to negotiate reaching for the same object as a human.

6.2 Experiment design

Firstly, we detail how we determined which prosody styles would be tested, and subsequently, which kind of task would be appropriate. Secondly, we present the experimental procedure, and a high-level description of the technical implementation. Thirdly, we describe the behavioural and subjective measures used in the experiment.

6.2.1 Choice of movement prosody dimension and robot task

Selection of contrasting movement prosody styles

Our goal was to select movement prosody styles which were found to have the most impact on participants' perceptions of the robot, and then design an experiment around the selected styles, rather than the opposite. We started by looking at the online and in-person experiment results, focusing on the adjective scales where we were able to find strong oppositions. This analysis used the response distributions based on the first 42 participants of the forced choice online experiment. In this instance, we define a strong opposition as the presence of physical variables that received more than 70% of participant responses on either side of the scales. This was the case for the scale opposing confident and hesitant / doubtful, where the in-person experiment showed that 70% of the high kinematics stimuli and 83% of the smooth stimuli were rated as confident, as opposed to 74% of the saccade stimuli and 61% of the low kinematics stimuli being rated as hesitant. Similarly, in the online binary response results, 65% of the smooth stimuli and 70% of long, uninterrupted motions (E and F combined) were rated as confident, whereas 75% of saccade stimuli and 67% of interrupted motions (C and D combined) were rated as hesitant.

The confident/hesitant scale was the one that presented the strongest oppositions, and those oppositions were mostly on the motion variables, which are the main focus of our study. Other scales presented some strong oppositions such as sturdy/frail, where saccades were strongly associated with frail, but the only corpus variable strongly associated with strong were the eyes, which we did not want to focus on.

The movement parameters used to illustrate the confident and hesitant styles were derived based on the results as described above. The confident style used the smooth variant, and long uninterrupted motions without hesitations (resembling A/B/E/F motion sequences) since these were strong results as mentioned previously. The hesitant style inversely used the saccade variant, and interrupted motions where the robot periodically slowed down, performing "hesitations" (resembling C/D motion sequences). Regarding the remaining motion parameters, we chose those that aligned best with the desired style. The confident style used the high kinematics whereas the hesitant style used low kinematics. For the appearance parameters, their impact on the confident/hesitant scale was relatively low, so they were identical between both styles in order to focus on motion parameters. The eyes were set to the round shape, the head was straight (always aligned with the robot's forward axis), and the base was stable [1]. The robot was equipped with laser-cut flyer holders, as can be seen on Figure 6.1.

[1]The head was securely attached, however the base balancing wheels had to be slightly loosened to aid navigation on the sometimes irregular surfaces in the entrance hall. The amount of body sway this induced was relatively small, hence closer to the "stable base" parameter than the "unstable base".

Figure 6.1: RobAIR robot equipped with the flyer holders, and tablet screen UI (translation: "*I am RobAIR, I hand out flyers. Help yourself!*").

Selection of robot task and role according to movement prosody styles

The pretext task (or, scenario) must meet several requirements. First of all, it must enable the robot to perform navigation in human environments, more specifically, navigation that involves the human in some way (following, guiding, approaching), not simply navigating through a crowd. Second, the task should be believable to avoid participants noticing they are part of an experiment, and it must not explicitly attract participants' attention to the manner in which the mobile robot navigates, to avoid them consciously analysing the navigation. Thirdly, the task should show some potential for a participant's rating of the robot's performance to be influenced by the perception of the attitudes or styles chosen in the previous step: Confident/Hesitant. Lastly, the task should fit within the feasible limits determined by the robot, the experiment location, and its ability to rapidly engage participants without them being recruited beforehand.

For these reasons, we selected the task of flyer distribution. We define this task in the following manner: the robot is positioned in the entrance hall of a building where it waits until it detects a person, approaches them using one of the two movement prosody styles, offers a flyer, and asks for a rating before moving back to its starting position. We detail the exact sequence of steps in the following sections. By its nature, the task requires the robot to approach a person. However it only approaches the person as a means to achieve its apparent primary goal of handing out a flyer. Furthermore, the entrance hall already had information boards and static flyer displays, making the presence of a flyer distribution robot quite credible. It also seems plausible that a person perceived as hesitant when distributing flyers might be rated as less effective than someone perceived as confident. Handing out a flyer is a relatively short interaction and can be initiated by the robot, which should enable a sufficient participation rate. It is also a task that is achievable both by teleoperating the robot for initial prototyping, and by implementing a relatively simple autonomous algorithm to improve the repeatability of the robot motion.

6.2.2 Approach motion

The core of the experiment is to perform a straight-line approach towards a person while employing a given movement prosody. This part of the experiment design is identical in the teleoperated and autonomous versions. It is also similar in principle to our lab-condition perception experiment. The velocity profiles used to achieve the confident and hesitant movement prosody styles followed the constraints as defined in our motion corpus. The difference with the lab perception experiment is that the distance to be traveled by the robot could be different for each encounter with a person, and the person may move while the robot is approaching it. This implies that the scripted, fixed length motions employed in the lab experiment cannot be re-used in this real-world experiment. We also require a method for correcting the robot's direction of travel in case the person moves out of alignment with the robot's forward direction. In the initial version of this experiment the robot's motion was controlled via assisted teleoperation, i.e. an experimenter teleoperated the robot with gamepad joysticks, and an algorithm filtered the joystick inputs to enforce only a subset of the movement prosody constraints. In the second version, the entire process was fully

autonomous, using our algorithm presented in the previous chapter to control the linear velocity, combined with a simple proportional controller for the rotation.

The distance from the person at which the robot stops has been studied in other HRI experiments, and may influence participants. In both versions of this experiment, we use the same distance as our prior lab experiment, setting the target final person-robot distance to be 50cm as measured between the widest part of the robot's "body", and the closest point on the person's legs, resulting in 80cm between the geometric centers of the person and robot, placing it just at the edge of arm's reach. This distance was enforced either manually by the teleoperator's timing of the deceleration, or automatically by altering the target goal point for our prosody-aware local planner based on the position of the person reported by the existing lidar-based person detection module.

6.2.3 Flyer handout task

We now describe the remaining design aspects of the robot's overall task "behaviour" in order to actually play the role of a flyer distributor, such as how the robot detects a person, how it orients towards them, and how it returns back to its initial position. These aspects are not variables we are directly interested in studying, however they are part of the participant's experience of the interaction with the robot, and should thus be systematically controlled when possible.

The robot is initially static. Once a person is detected, the robot performs a rotation to align its forward axis with the person's position. Once the rotation is finished, the robot initiates the approach motion, using a velocity profile using the motion parameter values which correspond to the confident or hesitant movement prosody styles. The robot stops at a distance of 80cm (center to center distance), and remains static near the person while they take a flyer. Once the interaction is finished, the robot rotates and subsequently moves back towards its initial position, still using the chosen movement prosody. While approaching the person, the robot corrects its heading in order to maintain the robot oriented towards the person. At each of these steps, the robot also uses voice and sounds as well as changing text displays on its tablet screen in order to engage the participants and explain the robot's task. The main difference between the teleoperated and autonomous conditions is how proactive the robot is when initiating the approach. Small differences were also present in the exact sequencing of the sounds and tablet displays changes. In the following parts, we motivate and describe the use of sound and display, and we detail the exact handout procedure for both versions of the experiment.

Voice and tablet display

During an initial pilot experiment, the robot did not use any sounds, voice, or display. The only indication of the robot's task were the flyers positioned on the robot. We found that few people noticed the robot's presence, and many simply kept walking when the robot was approaching them. When asked about why they didn't stop, or whether they understood what the robot was doing, some mentioned that they assumed some kind of

software testing was in progress, and did not want to disturb it. In order to clarify the robot's role, we designed several text and image displays to be shown on the robot's tablet. In order to better captivate people's attention, we also added sounds that would be played at various key stages of the interaction. We decided to use sounds from prior research on vocal prosody in HRI, more specifically the sounds used by the Emox robot (Sasa & Aubergé, 2017). These sounds were based on a highly modified human voice, with vocal prosody elements which are known to promote closeness and to encourage the establishment of socio-affective "glu" (Auberge, 2019). The sounds were single words, with the exception of one sound which was a non-lexical sound: "*Bonjour*" (*hello*, played at the start of an interaction, when a person was detected), "*hum-hum*" (bivocalic vocalisation made with closed mouth, played once the robot had arrived next to the participant in order to prompt them to provide a rating), "*Au revoir*" (*goodbye*, played at the end of an interaction, as the robot rotated away from the participant).

Teleoperated version

Detection of the person was performed using the microphones to hear people entering the hall, allowing the teleoperator to orient the robot towards their general direction of arrival, typically before they could even see the robot. Once the person became visible, the robot was oriented directly towards them. As soon as the robot's orientation was correct, the teleoperator started the linear approach motion. Shortly after starting the approach (0.5 to 1 second), the teleoperator activated the "hello" sound. If the person ignored or walked past the robot, the robot kept moving in the person's direction for an additional 5 seconds, before giving up and moving back to the its initial position. If the robot was able to reach the person, the tablet display was changed to invite people to take a flyer. Once the person took a flyer, the "hum hum" sound was played, and the display prompted the person to provide a rating by showing a number with their hands. Once the teleoperator saw the rating, the "goodbye" sound was played, and the tablet display changed accordingly, as the teleoperator rotated the robot to face its initial position. Then, the robot was driven back towards the initial position, still using the same movement prosody. Given the method for initiating the approach, the robot was often moving towards people who were still in motion, and moving towards the robot's general direction, often aiming to reach one of the exits located behind the robot. An example of an interaction with a participant with the hesitant movement prosody can be seen in the video in the supplementary material [2].

[2]Example interaction from teleoperated experiment: https://cloud.univ-grenoble-alpes.fr/s/HPD2a4wEpiAJwmA

(a) Person detected, robot says hello.

(b) Person stopped, robot rotates.

(c) Robot approaches.

(d) Person takes flyer.

(e) Person rates robot.

(f) Robot returns to base.

Figure 6.2: Still captures from some of the cameras for participant ap9, autonomous experiment, confident movement prosody.

Autonomous version

Detection of the person was performed based on the robot's lidar sensors, which have a maximum range of 5 meters. As soon as a person was detected, the "hello" sound was played. If the person continued walking past the robot, and went outside the sensor range,

the robot would do nothing. If the person stopped for more than 1 second, the robot oriented itself towards the person. Once it had oriented, if the person again did not move for one second, then the robot initiated the straight line approach motion described previously, displaying the message prompting people to take a flyer on the tablet. The robot kept moving towards the person until it reached a fixed distance from its initial position (5 meters). If this distance was reached, the robot gave up and returned to its base. Upon reaching the person, the robot allows some time for a participant to take a flyer (we used a fixed time of 2.5 seconds), after which it would play the "hum hum" sound, and display the prompt for the participant to rate the robot, and explaining that they could move away to finish the interaction. No automated detection of the rating was performed, instead we relied on people stepping away from the robot, at which point the robot played the "goodbye" sound and rotated in order to face the initial position, based on its estimation of its position relative to the initial base position. Upon reaching the base, the robot rotated so as to face the middle of the hall. Given the method for initiating the approach, the robot was often moving towards people who were static, and much closer than in the teleoperated experiment due to the short range of the lidar. Still images from an interaction using the confident movement prosody can be seen in Figure 6.2, and the video is also made available as supplementary material[3].

6.2.4 Software architecture

In this section, we present an overview of the software architecture that was implemented in order to generate the required robot behaviour during the HRI experiments. The ROS node structure is shown in Figure 6.3. The prosody-based receding horizon control node corresponds to the algorithm we proposed, implemented and validated in chapter 4, and chapter 5. The person detection and tracking module implements the method described in chapter 5. The decision node uses a finite state machine to implement the sequence of actions described in section 6.2.3, deciding which goal position should be sent to our control node (person or base), as well as when to send them (timing for initiation of approach, detecting when the person steps away from the robot to end the interaction). The localization node should provide the decision node with an estimation of the robot's position in the environment so that it can return to its base, and not go too far when following a person (e.g. avoid students having the robot follow them outside). We discuss this node in more detail, since the real-world experiment setting creates a non-trivial case of the localization problem.

Localization node

Many experiments using mobile robots in HRI require a way of localizing the robot in an indoor space. When such experiments are conducted in lab conditions, a common solution is to use external sensors installed around the lab environment such as OptiTrack[4] camera

[3]Example interaction from autonomous experiment: https://cloud.univ-grenoble-alpes.fr/s/GdnDKQbKD9GEgnG

[4]https://optitrack.com/

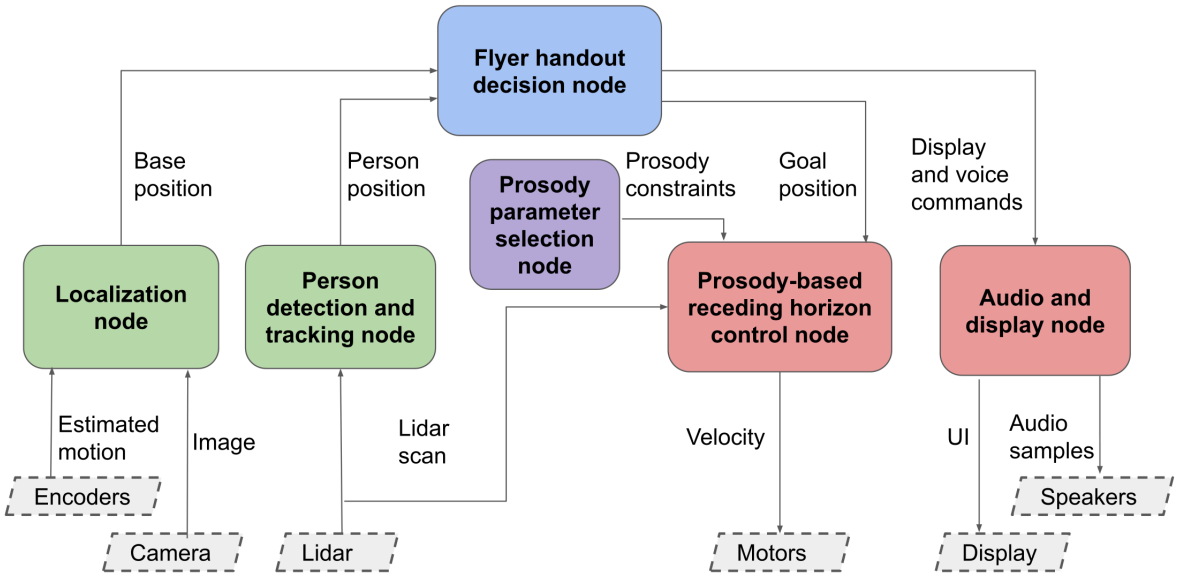


Figure 6.3: Our full software architecture for the autonomous version of the flyer experiment. Gray: Sensors and actuators. Green: Perception nodes. Blue: decision node implementing a state machine to iterate through the flyer handout process. Red: action/control nodes. Purple: user interface for the experimenter to select which movement prosody should be exhibited by our receding horizon controller presented in chapter 4.

systems or lidar sensors. When deploying the robot in ecological experimental conditions in the wild, heavy modification of the environment is often not possible, instead requiring the robot to rely on onboard sensors and computation. The most common off-the-shelf approach consists of using lidar sensors to build a map, and subsequently match the lidar scans to the map to estimate the robot's position (Dellaert et al., 1999). These approaches can fail in large environments when using short range lidars as found on most mobile robots used in HRI[5]. Additionally, when people are present in the environment they will obstruct features that are used for localization. Another type of solution consists of installing a camera pointing upwards so that the robot can use patterns and structures on the ceiling to localize itself. These approaches mostly rely on manually defined features (Jeong & Lee, 2005).

We propose an approach combining an upwards-facing camera on the robot with convolutional neural networks (CNNs) (Krizhevsky et al., 2017) to perform regression of the robot's pose based on a single image of the ceiling. We refer the reader to our paper (Scales, Rimel, et al., 2021) for details of the approach, since localization is not the main research question of this thesis. We use inception modules proposed in (Szegedy et al., 2014) to build two CNN regression architectures estimating position and orientation separately, which we found to perform better than a combined approach. We trained the model on a dataset captured in mostly empty environments by using the ROS implementation of lidar-based adaptive monte carlo localization (AMCL) as the ground truth, under varying lighting con-

[5]By short range we mean in the order of $5m$, as opposed to lidars used in autonomous driving applications with ranges in the order of $100m$, higher point density, and much higher cost.

Table 6.1: Position and orientation error for our CNN-based localization

Scenario	Average and 95% confidence interval	Standard deviation	Maximum
Without people	0.21±0.01m 0.10±0.01rad	0.14m 0.11rad	0.87m 1.18rad
With people	1.57±0.13m 0.26±0.02rad	1.57m 0.24rad	7.13m 2.20rad

ditions in order to make the model more robust to environment changes.

The results of our method when deployed in the experiment environment are given in Table 6.1. Without people, our method performed well with an average error of $0.21m$ and $0.1rad$, considering the entrance hall environment is $82m^2$. Most notably, the method maintained good performance in the middle of the room where the lidar based method often fails. Although our approach was able to handle the presence of people in the environment to some extent, it was not reliable enough when many people were present, as was often the case. When analysing high-error cases, we found that the network seemed to learn to rely on a small number of specific environment features despite other useful features being visible. Training the networks on datasets including people seems like a promising future work, however the construction of such a dataset would require another method for determining the ground-truth, for example by temporarily modifying the environment by installing external sensors.

This work was aimed at localising the robot at any point in the environment in a single-shot fashion, which is a harder and more general problem than our specific experimental setup requires. For the purposes of this experiment, we adopted a simpler method by combining odometry measurements from the robot's motor encoders with visual markers placed at the robot's base position. Integrating the odometry measurements over time enables the robot to have an estimation of its position relative to its base as it approaches the person. The odometry-based estimation is accurate enough for the robot to rotate back towards the direction of the base, enabling the aruco marker to be detected using an existing library (Garrido-Jurado et al., 2014).

6.2.5 Environment, participants and data collection

Experiment location

The experiments took place in a large entrance hall in the faculty of mathematics and computer science at the Université Grenoble Alpes, France (see Figure 6.4 and Figure 6.5. The location was selected based on several criteria. First we considered physical criteria such as the size of the hall and its open and clutter-free design enabling the robot to navigate without obstructions, and to perform long enough motions. Second, the hall is relatively busy, with students and university workers crossing through it at break and lunch time. Almost all participants in the autonomous version of the experiment were students of computer science and applied mathematics. In the teleoperated version, there were also teachers and

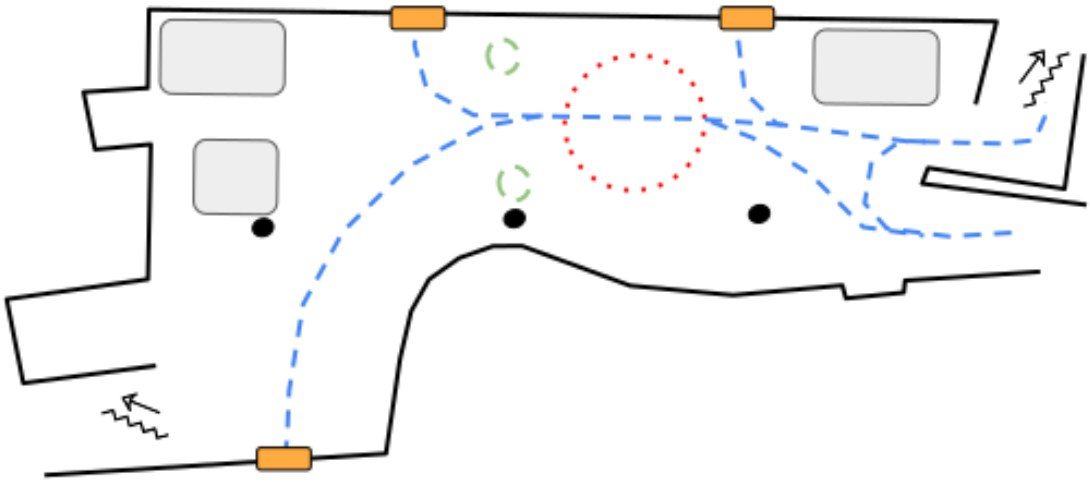


Figure 6.4: Top-down diagram of the entrance hall. Orange: building entrances. Blue: typical paths between entrances and stairs or corridor. Green dashed circles: robot initial location in autonomous experiment. Red dotted circle: robot initial location in teleoperated experiment. Grey: seating areas.

university workers due to the timing of the experiment [6].

Sensor data acquisition

During the robot's interaction with the participant, we aimed to record the participant's behaviour in order to study their spatial behaviour with respect to the robot, as well as capturing their overall behaviour such as facial expressions, gestures and speech which may also provide useful insights into their reaction to the robot and the various movement prosody styles. In order to achieve this, we used our synchronized sensor network using ROS and chrony to ensure data is synchronized across multiple computers (see Appendix B). Three tablet computers were placed around the entrance hall, each connected to a 2D lidar sensor to capture the person's position, and using their integrated webcams to record video from several angles. The robot's onboard sensors were also used: top and bottom lidar sensors to estimate the robot's position with respect to the person, a wide-angle fisheye camera to capture the participant's whole body, an intel realsense RGBD (color + depth) camera oriented to capture the upper body gestures and facial expressions, and high-quality stereo microphones to record participant's comments during the interaction. We also recorded data from the robot's motor sensors to accurately estimate its velocity, as well as other internal data related to the algorithm's execution such as the velocity commands given by the teleoperator or by our prosody-aware algorithm, and timing of sounds and tablet display changes.

[6] We did not ask this information from participants, we report this characterization of participants based on the interviewer's interactions with them, and estimated age.

Figure 6.5: Photo of the entrance hall. Orange: building entrances. Blue: typical paths between entrances and stairs or corridor.

Subjective measures: rating and interview

One of the experiment's aims was to determine whether the different movement prosody styles impacted participant's perception of the robot's performance in its apparent task, in our case, flyer distribution. In order to fit with our flyer distribution pretext, we decided to use a very quick and simple approach by asking participants to provide a rating between 1 and 5, 3 being neutral. This rating was requested by the robot itself once it had approached the participant since we assumed that participants may be reluctant to provide a bad rating if the interviewer asked them. Once the robot had approached the participant, it displayed a screen on the tablet prompting the person to provide the rating by using their hand to form the number.

In order to study the participant's impressions of the robot, we designed a semi-structured interview to be conducted after the interaction. The format of the interview had to be carefully considered, since the object of our study is a relatively small variation in the robot's movement. In our first online perception experiment, we had already seen that participants struggled when directly asked to associate the robot's motion to a pre-determined set of adjective scales. This called for a less structured interview style, however we still aimed to coax participants into talking about their impressions of the robot's motion, hence the semi-structured format. Semi structured interviews provide a way to explore a defined list of topics, while leaving enough freedom and open questions to understand participants' responses, and potentially discover new concepts or aspects which were not initially thought to be relevant. We conducted the interviews following a mix of open-ended questions, prompts to get more details or encourage the participant to talk more, and closed questions, as discussed in (Leech, 2002).

At the start of the interview, we took an indirect approach to gathering participants' perception of the robot by asking general questions that participants would find relatively easy to respond to, generally encouraging them to speak freely about the robot. We asked them what they thought of the robot, and whether they found it distributed flyers well, asking them to explain the rating they gave the robot. During this first phase, we wanted the participants to remain convinced that the sole goal of the robot and the interview was to test and evaluate the flyer distribution, allowing us to determine whether participants would spontaneously mention the robot's movement without being explicitly asked about it. After this initial phase, the interviewer revealed that this was in fact a larger study on how the manner in which the robot moves may influence the person's perception of it. At this point, the interviewer once again asked for the participants' impressions given this new information. Again, if the participants struggled to talk about the robot, the interviewer asked questions related to the robot, but avoided directly mentioning the movement prosody parameters (speed, acceleration, saccades). Instead, the interviewer asked questions about the robot's appearance, or how the participant felt when the robot was approaching them. Once the participant no longer had anything to add, the interviewer asked a more precise question, asking how they would describe the robot in terms of a personality. This question would enable us to study which kinds of words participants used when they were not provided with a pre-existing set of adjectives. Lastly, the interviewer asked participants if they would rather say that the robot seemed confident, or hesitant, and to explain their choice. This was the last question of the interview, after which the interviewer explained which data had been recorded during the experiment, and asked for oral consent to keep and analyse the data. Interviews in the teleoperated experiment were conducted by Raphaël Luffroy, while interviews in the autonomous experiment were conducted by the author of this thesis.

Participants

We consider as participants the people who performed an entire interaction with the robot as well as the post-interaction interview, and consented for their data to be used in the context of this study. People who ignored the robot entirely, or interactions where the robot did not move (due to the person approaching the robot before it could move) were not considered. Several people were not alone, walking with friends or colleagues. In the teleoperated experiment we interviewed only the member of the group towards which the robot was moving. In the autonomous experiment we interviewed the entire group, considering a person as a participant as long as they provided answers and gave their impressions during the interview. Therefore, a single interview could involve between 1 and 3 participants. We report the number of interviews and number of participants in Table 6.2. The split between confident and hesitant movement prosody settings for the robot was balanced in the teleoperated experiment, and close to being balanced (19 and 16 respectively) for the autonomous experiment, as reported in Table 6.3.

Table 6.2: Number of interviews and participants in the flyer distribution experiment

	Nb interviews	Nb participants
Teleoperated	20	20
Autonomous	21	35
Total	41	55

Table 6.3: Number of participants per movement prosody condition

	Confident	Hesitant
Teleoperated	10	10
Autonomous	19	16
Total	29	26

6.3 Results and analysis

In this section, we first characterize the robot's motions in both versions of the experiment, checking whether they were consistent with the prosody and similar for each participant. Then, in the following subsections, we present the results and statistical analysis regarding each of our research questions surrounding the perception of the robot's movement prosody, rating of the robot performance, and participant spatial behaviour.

6.3.1 Analysis of robot motions

In this subsection, we discuss the robot's actual performance of the motions, and verify that the motions complied with the desired movement prosody. We also report the initial and final distances between the robot and participants during the approach phase.

Compliance of motion with prosody

In most cases, the robot produced motion which was consistent with the desired movement prosody. There were however some instances in the teleoperated version of the experiment where issues arose with the robot's motion. Firstly, the teleoperator often initiated the deceleration too late when the robot approached a person moving towards the robot, not providing enough compensation for the person's motion. The result is that for six participants the robot initially decelerated with the correct deceleration value in accordance with the kinematics setting, but then performed a sudden stop triggered by the assisted teleoperation algorithm to avoid colliding with the person. Secondly, an issue in the ROS implementation caused the configuration of the saccade amplitude to be noticeably lower

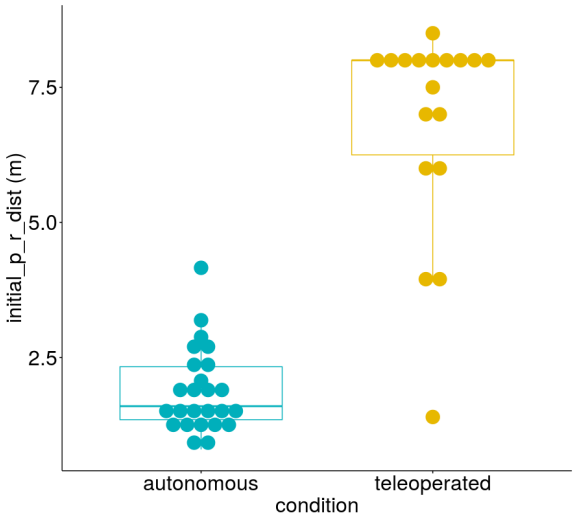


Figure 6.6: Initial person-robot distance comparison between the teleoperated and autonomous experiments. Autonomous approach motions are initiated at a closer distance due to the 5 meter range of the sensor used for person detection.

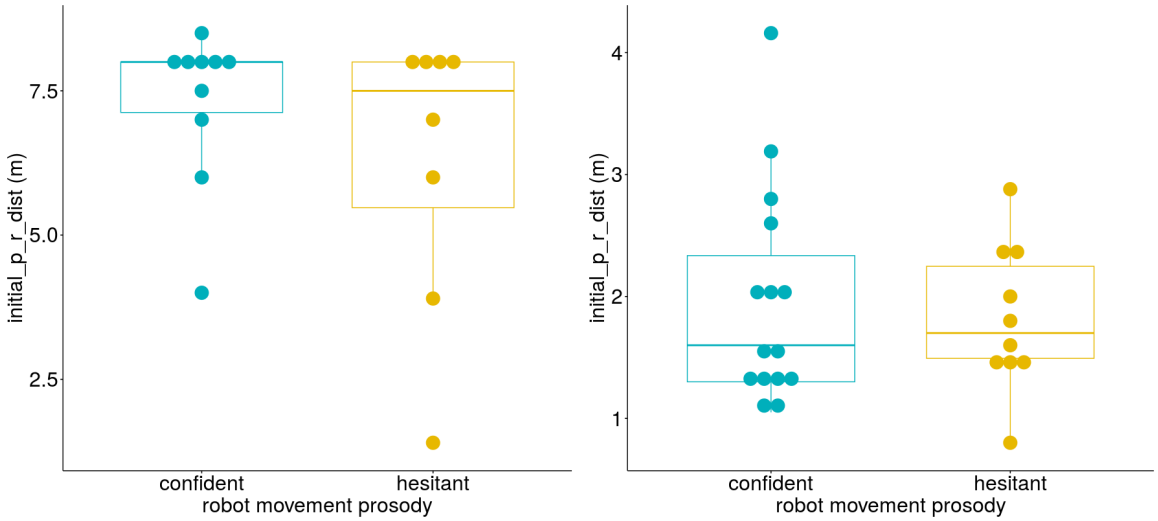
than it should have been for 5 out of the 10 participants in the hesitant condition [7]. During the autonomous experiment, no such issues arose, and the robot's motion corresponded to the desired movement prosody.

Approach distances

In the teleoperated condition, the robot often started approaching participants which were further away than in the autonomous condition, due to the difference in person detection method and approach initiation criteria described previously. We measured the initial person-robot distance at which the robot began its linear approach motion in order to quantify the difference between experiment versions, as well as check whether the distances were similar across the movement prosody styles. As can be seen in Figure 6.6, the initial distance to the person when starting the approach was longer in the teleoperated experiment (M=6.85, SD=1.93) than in the autonomous experiment (M=1.90, SD=0.772). A Welch two sample t-test confirmed that this difference is statistically significant ($p = 5.729\text{e}{-10}$).

Given the differences between the teleoperated and autonomous experiments, we consider them separately when observing the distribution of initial distances across the confident and hesitant movement prosody styles, shown in Figure 6.7. As expected, the initial distances are similar across movement prosody settings for both experiments in terms of their mean values and overall shape of the distribution. The differences in means were found to be not significant by Welch two sample t-tests (teleoperated: $p = 0.3183$, autonomous: $p = 0.6492$).

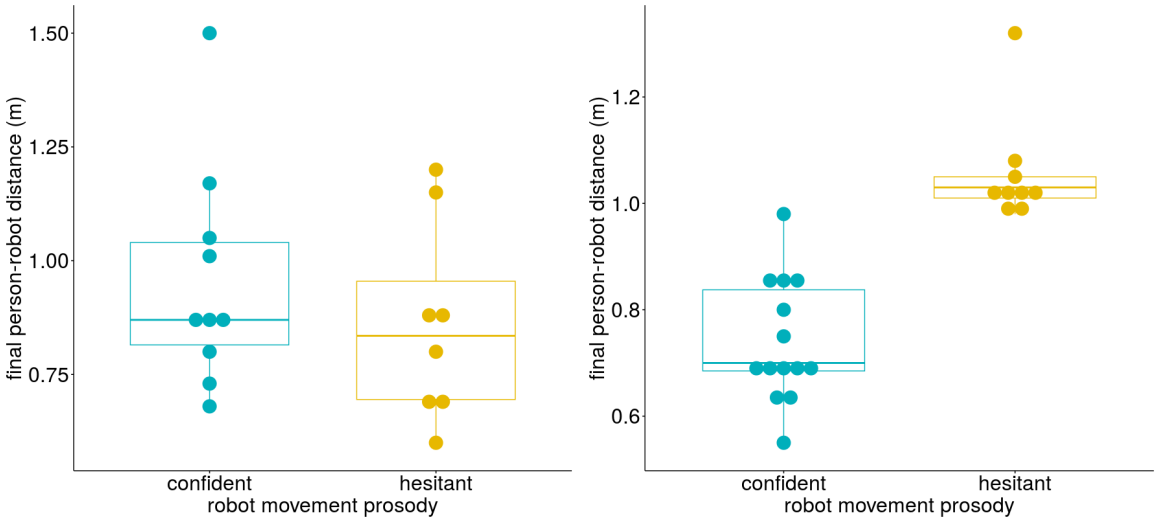
[7] This issue seems to be related to the ROS dynamic reconfigure functionality which we used to define the robot's movement prosody and its behaviour when nodes are restarted. A full restart of ROS is required to recover the correct configuration.



(a) Teleoperated (b) Autonomous

Figure 6.7: Initial person-robot distance comparison between the teleoperated and autonomous experiments, for each robot movement prosody.

The final person-robot distance at which the robot stopped next to the person was also measured in order to establish whether we were able to control this variable appropriately, since it may impact participants' perceptions. We recall that the desired final distance was set to $0.8m$. The distributions of final distances for both experiments, across movement prosody styles is presented in Figure 6.8. In the teleoperated experiment, the distributions are quite similar, showing no statistically significant difference in the mean (Welch two sample t-test $p = 0.7973$). The final distance means are slightly higher than expected which may be due to the teleoperator being cautious, preferring to undershoot rather than overshoot (confident: M=0.954, SD=0.242; hesitant M=0.861, SD=0.217). For both prosody styles, there is a high spread of values due to the teleoperator struggling to consistently time the deceleration appropriately, especially since participant were moving in most cases. In the autonomous experiment, the approaches performed with the confident movement prosody stop $0.31m$ closer to the participant on average than the hesitant movement prosody (confident: M=0.743, SD=0.114; hesitant: M=1.06, SD=0.102). This difference is statistically significant by a Welch two sample t-test ($p = 8.325\text{e}{-07}$). The distributions also differ in terms of their shape and spread, with hesitant distances being tightly grouped around the mean, whereas confident distances are more scattered and follow a bi-modal distribution. After investigation, we found that an error in the configuration file defining the hesitant movement prosody settings for our algorithm caused the target position to be set $0.3m$ further from the participant, which could explain the shift in the mean distance. The higher dispersal in the confident setting is likely to be the result of our algorithm's discretized space of control actions and frequency of re-planning, causing it to be less accurate as velocity increases.



(a) Teleoperated

(b) Autonomous

Figure 6.8: Final person-robot distance comparison between the teleoperated and autonomous experiments, for each robot movement prosody.

6.3.2 Participant perception of movement prosody

In this subsection we discuss the results relative to participant's responses when asked whether the robot seemed confident or hesitant. Our hypothesis was that if the prosody style parameters are independent from the task, role and situatedness of the interaction, we should see participants perceiving them in a similar way to the online experiment. Figure 6.9 shows the confusion matrix based on $n = 45$ participants who answered the question. For the 10 remaining participants, either the question was not asked ($n = 6$), or participants had no idea which one to pick ($n = 2$), or they refused to use either word to describe the robot ($n = 2$). The number of hesitant and confident interactions is not balanced in this sample, hence if the participants' perceptions had perfectly matched the robot's movement prosody, we would expect to see overall percentages of 40% and 60% in the diagonal elements, and zeros in the off-diagonal elements. Instead, we see 8.9% and 57.8% on the diagonal, and imbalanced percentages in the off-diagonal elements, indicating that participant perception did not match the movement prosody. More specifically, observing the column percentages we see that 96.3% of the confident movement prosody interactions were perceived as confident (corresponding to only 1 out of 27 participants perceiving the confident prosody as hesitant). On the other hand, only 22.2% of the hesitant movement prosody interactions were perceived as hesitant (corresponding to 14 out of 18 participants perceiving the hesitant prosody as confident). A McNemar test (with correction) indicates that the difference in off-diagonal elements is statistically significant ($p = 0.001946$).

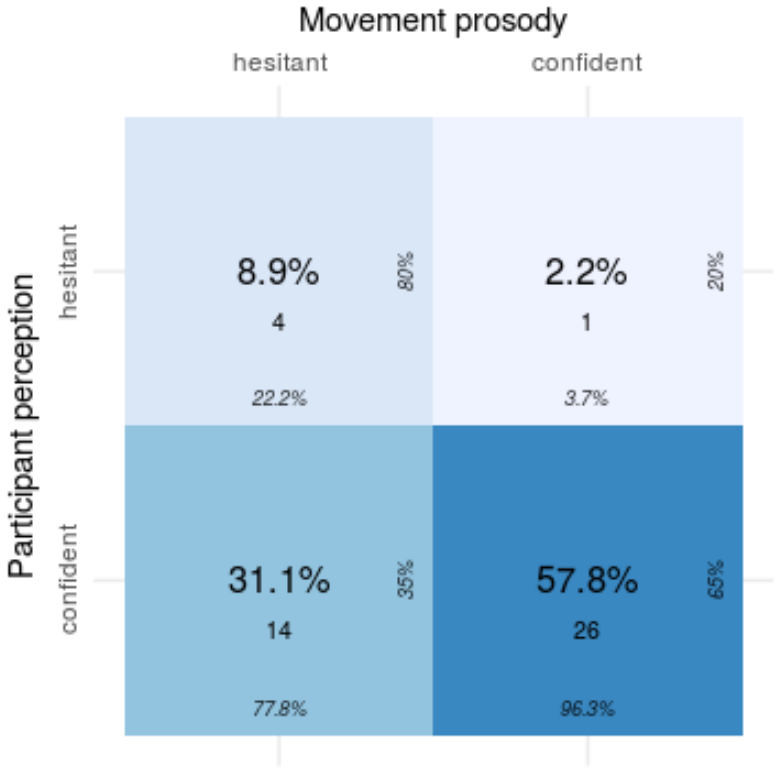


Figure 6.9: Confusion matrix of participant perception of the two movement prosody style (responses from teleoperated and autonomous experiments combined, n=45). Middle of the cells: overall percentages and counts. Bottom: column percentage. Right: row percentage.

From these results, it seems that participants' choice between calling the robot confident or hesitant were mostly influenced by factors outside of those differentiating the two movement prosody styles (kinematics type, variant, and motion sequence). Since almost all participants perceived the robot as confident rather than hesitant, we might turn to the aspects of the robot's behaviour which were common across the movement prosody condition for a potential explanation. Participants were also asked to explain their choice, hence their explanations may indicate which aspects could have influenced them. When considering the explanations of the 4 participants who perceived the hesitant prosody as hesitant, they seemed to have mixed perceptions. We show the transcripts of their explanations below, where ap6 and ap7 were part of the same group.

> Participant ap7: Hesitant in its movement, but in- it knows where it is going, yeah confident, even if it stumbles on the way there.
> *Original, French: Hésitant dans le deplacement, mais dans- il sait où il va, ouais confiant meme si il y va en trébuchant..*

> Participant ap6: Hesitant in its movement, it does some jerky motions like 'tac tac tac', but that might be because of the floor tiles. I can see what the others mean [referencing ap7 saying it seems confident in its decision to approach],

> but for me the movement gives off a frail impression.
> *Original, French: Hésitant par rapport à son déplacement, ça fait des à-coups, "tac tac tac", mais c'est peut-être a cause du carreleage. Je vois ce que les autres veulent dire (confiant dans sa décision) mais pour moi le mouvement fait fragile.*

> Participant ap4: I'd say it's in the middle, there's some moments of uncertainty. [...] maybe it depends on how quickly I move, sometimes it follows me but there's a little moment of hesitation. But otherwise, rather confident at first glance.
> *Original, French: Je dirai entre les deux, y'a des petits moments de flou. [...] Peut-être par rapport à ma vitesse de déplacement, parfois, il me suit mais y'a un petit moment d'hesitation. Mais sinon plus confiant à première vue.*

> Participant tp3: It seemed like it hesitated for a second or two, and then it targeted me. *Original, French: J'avais l'impression qu'il a hésité pendant une seconde ou deux, puis après il m'a ciblé.*

In all four cases, participants associated the hesitant impression to specific aspects related to the physical motion quality itself. However, in addition to their justification for choosing hesitant, participants also explained why they may also find some aspects confident, such as ap6 and 7 who agreed that the robot "knew where it was going", referring more to a form of attention, which could also be reflected by participant tp3 saying the robot "targeted" them. Many participants who perceived the robot as confident referred to similar aspects involving how the robot "decided" to move towards them, the "direct" or "straight" path it took, and the fact that the robot did not switch between approaching different people. This occurred both for participants who saw the confident and hesitant movement prosody. These comments all refer to aspects of the robot's motion which were not involved in our online or lab-based perception studies, and they are tied to the robot's actual task of detecting and approaching a person. Many participants mentioned such aspects. Several participants mentioned speed and absence of stutters, hence referring to part of our movement prosody dimensions. Only few participants mentioned the robot's general appearance or the voice, or did not give much explanation at all.

These insights into participants' explanations imply that for the task of approaching a person and the role of distributing flyers, these aspects of the robot's behaviour may have been the primary influence on participants' choice between the confident and hesitant labels, with the movement prosody only playing a secondary role. In the more general sense, this implies that even within the realm of the robot's spatial and temporal behaviour, or in other words, its movement, the mapping between physical parameters and the impressions the robot generates can differ based on the robot's task and role.

6.3.3 Participant rating of robot performance

After having approached the person, the robot displayed a screen prompting them to evaluate the robot on a scale from 1 to 5. Our goal was to explore whether the participants'

Table 6.4: Ratings according to movement prosody

Rating	Confident	Hesitant	Confident %	Hesitant %
3	1	0	3.44	0
4	8	4	27.58	16.66
5	20	20	68.96	83.33

Table 6.5: Ratings according to participant perception

Rating	Confident	Hesitant	Confident %	Hesitant %
3	1	0	2.63	0
4	8	0	21.05	0
5	29	5	76.31	100

ratings would depend on the robot's movement prosody. The resulting rating distribution over both the teleoperated and autonomous experiment data is shown in Table 6.4. Two participants did not rate the robot. Of the remaining 53, the vast majority (40/53) gave the highest rating of 5, with only 12 participants choosing 4, and one participant choosing 3. The confident movement prosody led to worse ratings (M=4.665) than the hesitant prosody (M=4.833) on average (9/29 non-maximal ratings for confident, and 4/24 for hesitant). Despite these differences, no significant effect was found by an exact Wilcoxon-Mann-Whitney test (p=0.2992).

Given the differences observed between the robot's movement prosody and participants' perception in the previous subsection, we also present the rating distribution across participant perception in Table 6.5. Although the number of participants who perceived the robot as hesitant was low (only 5 out of 45), all of them gave the maximal rating (M=5.00), which might be expected given the overall tendency for participants to give the maximal rating. The distribution of ratings for participants who perceived the robot as confident is still heavily biased towards the maximal rating (M=4.737). Again, the difference in rating distribution across participant perception was not found to be statistically significant (p=0.3923).

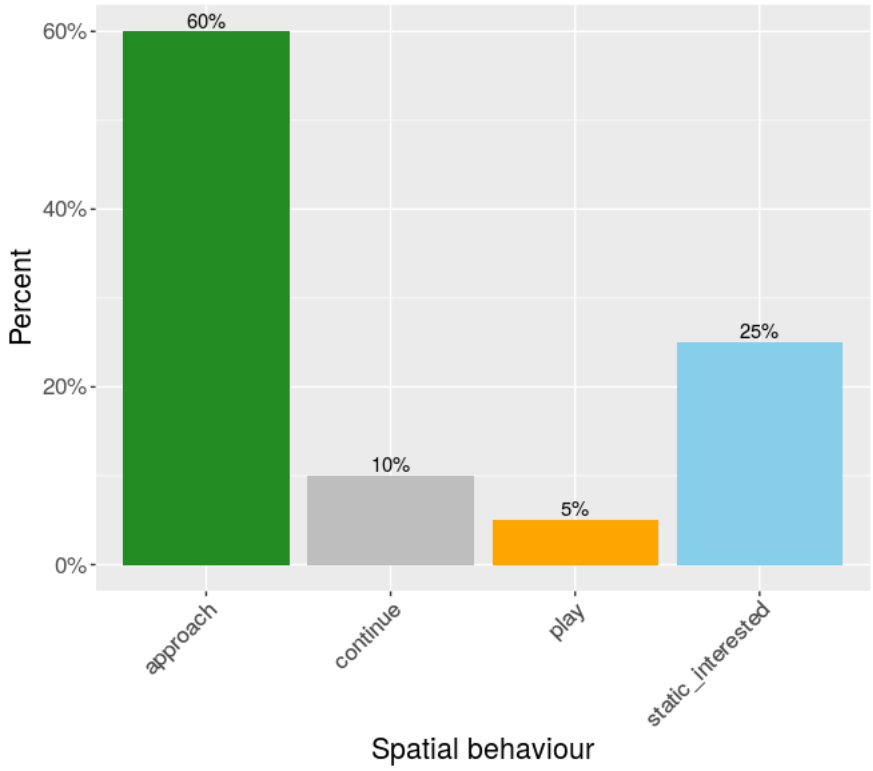
When reviewing the explanations given by the 13 participants who gave a rating of 3 or 4, no overall trends could be found. Three participants disliked the robot's voice, calling it "strange" or "scary", or stating it "bothered [them]". Three participants gave no explanation, and the remaining 7 participants cited various reasons such as not finding the robot useful, finding the interaction too limited, or practical issues related to reading the tablet screen. Two participants cited aspects that could be related to the robot's motion, stating that it "got too close", or "handed out flyers as aggressively as someone in the street".

6.3.4 Participant spatial behaviour

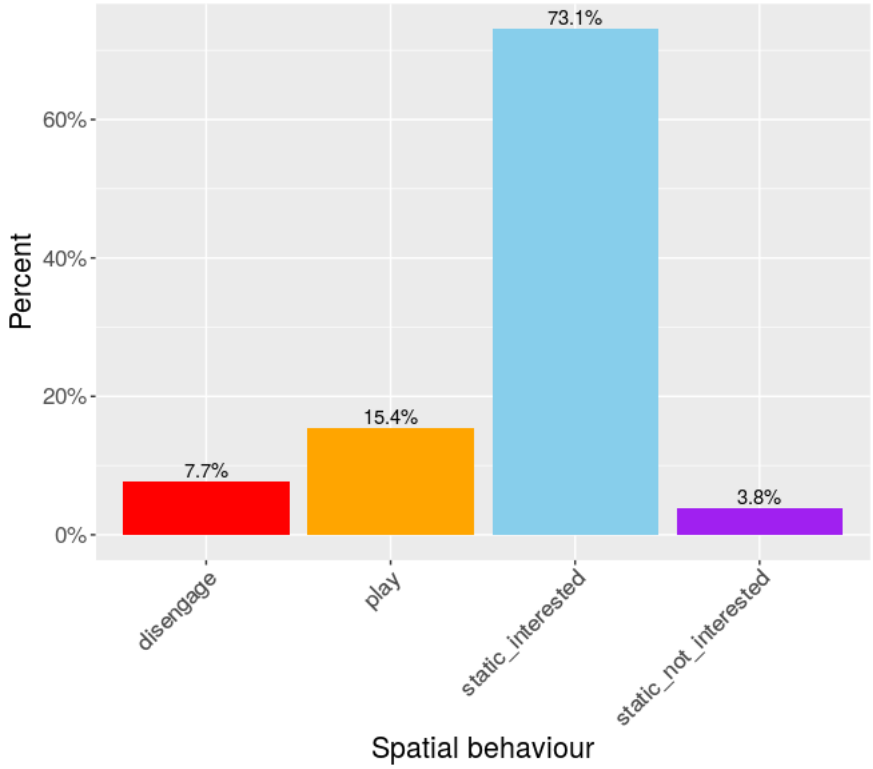
Participants displayed various spatial behaviours when the robot started approaching them. We reviewed the video recordings of the experiments, and formed categories that encompassed all of the participants reactions. These categories describe the participants' behaviour from the moment the robot began its linear approach motion. We only annotated spatial behaviours for the participants towards which the robot performed its approach motion (n=44). If a participant was standing to the side, several meters away from the person the robot was approaching, we did not annotate their spatial behaviour.

- **Approach**: the participant moves closer to the robot;
- **Static_interested**: the participant remains static, does not move, and is looking at the robot;
- **Static_not_interested**: the participant remains static, does not move, and is not looking at the robot;
- **Disengage**: the participant moves away from the robot, avoiding it;
- **Continue**: the participant does not pay attention to the robot, continues on their original path;
- **Play**: the participant "plays" with the robot, typically walking backwards, looking at the robot following them.

We present the distribution of spatial behaviours in each experiment in Figure 6.10. In the teleoperated experiment, *approach* was the most represented (60%) category, followed by *static_interested* (25%), *continue* (10%), and *play* (5%). In contrast, in the autonomous experiment *static_interested* was the most represented (73%), followed by *play* (15%), *disengage* (8%), and *static_not_interested* (4%). In the teleoperated experiment, the robot initiated its approach motion while participants were still walking, whereas in the autonomous experiment the robot waited for participants to stop before approaching. Furthermore, the robot's initial positioning meant that it was often between the participants and their goal (typically an exit of the building), meaning that they would often already be walking towards the robot's general direction as it began its approach. Considering this, a person moving towards the robot (*approach* behaviour) in the teleoperated experiment often corresponded to them continuing their current trajectory, as opposed to stopping when the robot started approaching them (*static_interested* behaviour). Likewise, a person remaining static in the autonomous experiment also corresponded to maintaining their current state, as opposed to *disengage* or *play* where they started moving once again as a response to the robot's approach. In the following paragraphs, we study the teleoperated and autonomous data separately.



(a) Teleoperated



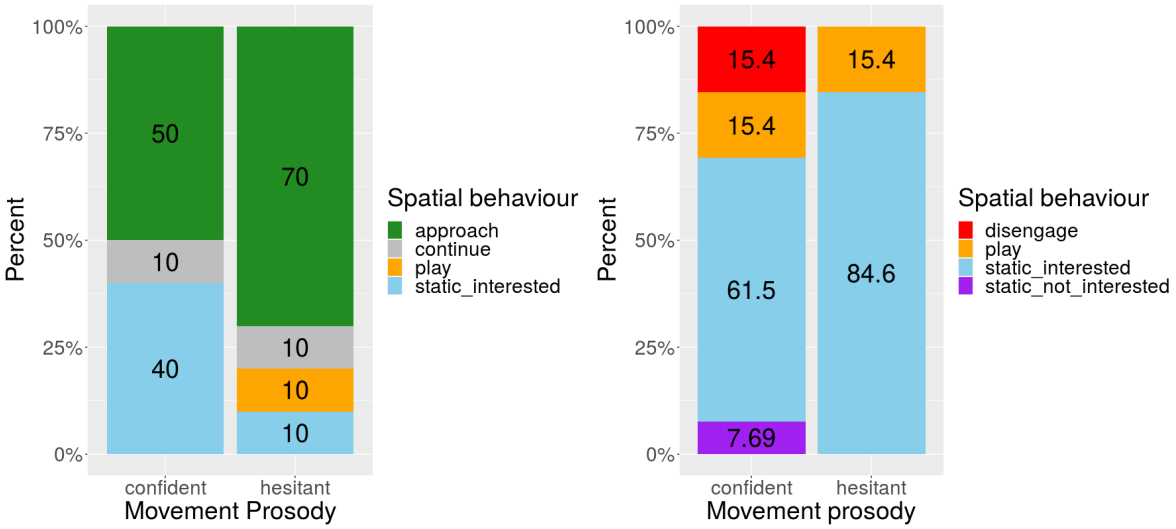
(b) Autonomous

Figure 6.10: Participant spatial behaviour in the teleoperated and autonomous experiments.

Effect of movement prosody on spatial behavior

In Figure 6.11, we show the participants' spatial behaviour according to the robot movement prosody. In the teleoperated experiment, participants' spatial behaviour showed some differences between the two movement prosody styles. For the hesitant prosody, the *approach* behaviour represented 70% of participants (7/10), followed by *continue, play*, and *static_interested* all at 10% (1/10). For the confident prosody, fewer participants approached the robot (50%, 5/10), and more participants remained static (40%, 4/10). This difference results in participants having more of a tendency to stay further away from the robot with the confident prosody. In the autonomous experiment, the spatial behaviour distribution is quite similar across the two movement prosody styles, in both cases the most common behaviour being *static_interested*. For the hesitant prosody, *static_interested* accounts for 85% of the participants (11/13), and *play* accounts for the remaining 15% (2/13). For the confident prosody, *static_interested* only accounts for 61% (8/13), followed by *play* and *disengage* both at 15% (2/13), and lastly *static_not_interested* at 7% (1/13). While the difference is less marked than in the teleoperated condition, the general trend of participants to take actions that increase the distance, or reduce the rate of approach with respect to the confident prosody robot is seen again.

In order to assess the statistical significance of these differences, we merged several behaviours together and ran Fisher tests. For the teleoperated experiment data, the difference in the proportion of *approach* behaviours was not found to be significant (p=0.6499). For the autonomous experiment data, the difference in the proportion of *static_interested* behaviours was also not found to be significant (p=0.2087).



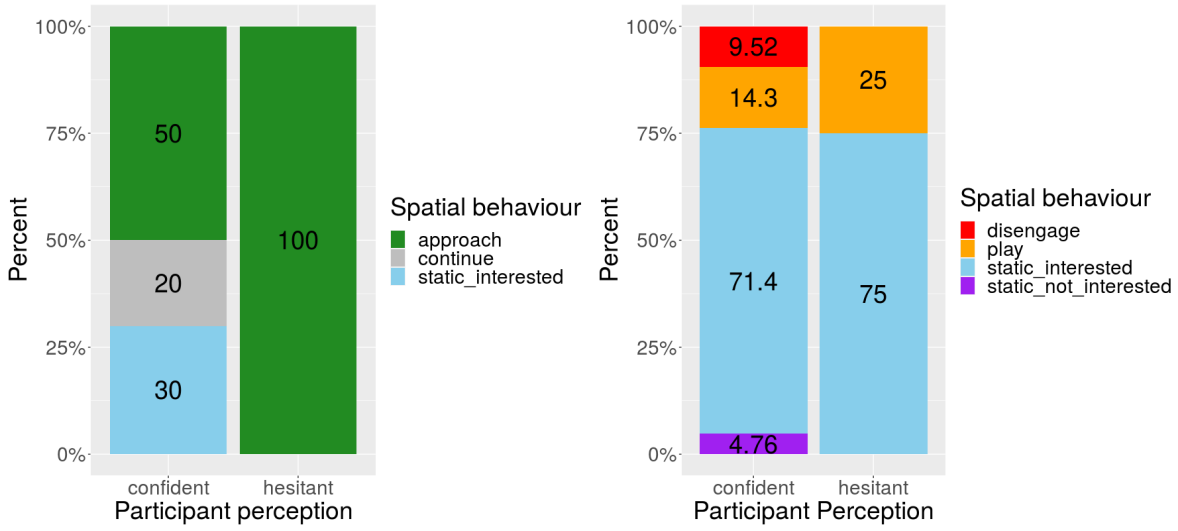
(a) Teleoperated

(b) Autonomous

Figure 6.11: Participant spatial behaviour in the teleoperated and autonomous experiments, according to the robot movement prosody.

Effect of participant perception on spatial behaviour

Since participants would not necessarily perceive the movement prosody as intended, we also explored the relation between participant perception of the robot and their spatial behaviour, shown in Figure 6.12. As discussed in the previous section, few participants perceived the robot as hesitant, hence one must bear in mind that the distributions are based on heavily imbalanced data. Nevertheless we report the results for the sake of completeness, and as inspiration for further study. In the teleoperated experiment, only one participant perceived the robot as hesitant, and they approached the robot. As expected, for those who perceived it as confident, the distribution of spatial behaviours matches the overall distribution of spatial behaviours in the teleoperated experiment. In the autonomous experiment, four participants perceived the robot as hesitant, three of which displayed the *static_interested* behaviour, and one of which played with the robot. Two out of the 21 participants who perceived the robot as confident displayed the *disengage* behaviour, which is in agreement with the distribution of *disengage* behaviours according to the robot's movement prosody. In both cases these differences are subtle, and any test of statistical significance will have limited power compared to balanced groups.



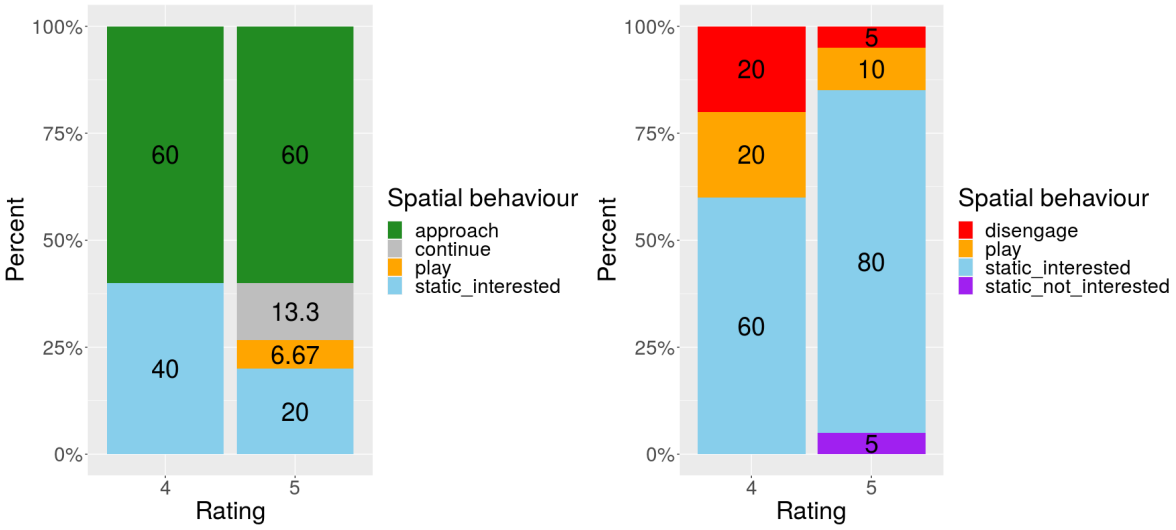
(a) Teleoperated

(b) Autonomous

Figure 6.12: Participant spatial behaviour in the teleoperated and autonomous experiments, according to their perception of the robot.

Relationship between rating and spatial behaviour

Lastly, we wanted to explore whether there was any relationship between the rating and spatial behaviour. The distributions are reported in Figure 6.13. In the teleoperated experiment, we observe that participants giving the lower rating tended to stay static more often (40%, 2/5) than those giving a higher rating (20%, 3/15). In the autonomous experiment, participants giving the lower rating tended to disengage more often (20%, 1/5) than those giving a higher rating (5%, 1/20).



(a) Teleoperated

(b) Autonomous

Figure 6.13: Participant spatial behaviour in the teleoperated and autonomous experiments, according to their perception of the robot.

Participant avoidance motion

In some cases, participants displayed a *backstep* behaviour, stepping back quickly when the robot approached. All instances of the backstep behaviour were in the autonomous experiment. Table 6.6 shows the occurrences of *backstep* according to the movement prosody, showing that most instances occurred with the confident prosody. An exact Fisher test reports a p-value of 0.3569, indicating that this difference is not significant. When observing the video recordings of participants who performed a backstep, all were already close to the robot (less than 1.5m). We plot the distribution of approach initiation distances according to whether participants performed a backstep in Figure 6.14. The mean initial distance for backstep participants is $1.2m$, whereas for non-backstep participants the mean is $4.4m$ (significant, $p = 5.183\text{e}{-8}$, Welch two sample t-test).

Table 6.6: Backstep according to movement prosody across both experiments

Backstep	Confident	Hesitant
yes	4	1
no	22	21

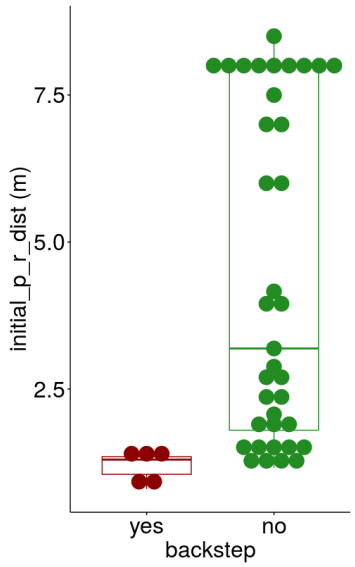


Figure 6.14: Initial person-robot distance in the teleoperated and autonomous experiments, according to participant backstep behaviour.

These results suggest a relationship between the initial distance and participants taking a backstep action as the robot begins its approach. Although non significant, it seems plausible that the confident prosody settings may also have a role to play given the higher acceleration and velocity, which may surprise participants. This behaviour might also be related to participants' expectations about whether or not the robot would move, since not all participants who were close by performed a backstep.

6.4 Conclusion

The first goal of this experiment was to explore whether participants recognised the robot's hesitant or confident movement prosody in a similar fashion to our prior online and lab-based experiments. The two movement prosody styles did not result in the same participant attribution of confident or hesitant adjectives as in our prior experiments, with most participants choosing the confident adjective, even when they had interacted with the robot using the hesitant movement prosody style, whereas the confident movement prosody was almost never described as hesitant (this difference was found to be statistically significant). Participant explanations of their choice revealed a trend to attribute the confident adjective based on the robot's "direct" approach strategy, fast reaction, and "decision" to go towards a specific participant, which suggests that these aspects are relevant dimensions in shaping how people perceive the robot in its role of a flyer distributor. If the dimensions involved in movement prosody depend on the robot's role and task during the interaction, this implies that being able to compare between different HRI studies will be essential in order to properly understand movement prosody, in order to determine which aspects are invariant across tasks, roles, and presumably also across robot appearance.

The second goal of this experiment was to explore whether the movement prosody

would impact participants ratings of the robot's performance. Most participants gave the highest grade to the robot, leading to difficulties in determining any impact of other factors on the participants' rating of the robot. This problem could be related to the "wow effect" that many HRI studies are confronted with when participants have never seen the robot before. While our data does show some slight differences in participant ratings, the lack of contrast (low number of non-maximal grades) does not allow us to make any strong conclusions about relationships between rating and movement prosody or participant perception. It may be interesting to prolong the study in order to assess whether ratings of performance change over time as people become more used to the robot's presence. Such studies could help to explore the slight trends observed in this experiment such as the confident movement prosody receiving worse ratings.

The final goal of this experiment was to explore whether participants' movement and spatial behaviour around the robot was influenced by the movement prosody. Participants showed differences in their reactions when comparing the teleoperated and autonomous experiment, although they are likely caused by the approach initiation method used for each experiment. Within each experiment, we found that participants showed more of a tendency to maintain more distance with the robot when confronted with the confident movement prosody, either by stopping (if they were already walking, as was typical in the teleoperated experiment), or by moving away from the robot (if they were already standing still, as was typical in the autonomous experiment). These differences were not found to be statistically significant, and would require further investigation.

Chapter 7

Conclusion

Mobile robots are being deployed to perform tasks requiring them to navigate alongside humans, whether in public spaces, on factory floors, or in care homes. Many algorithms have been proposed to plan the robot's motion so that it is considered more acceptable by humans, but the definition of what exactly constitutes acceptable social navigation is still an active topic of research. Acceptability is usually defined in terms of human perception of comfort, naturalness, legibility, and perceived safety. Often, robots are perceived as social agents rather than objects, and are attributed intentions and attitudes. We hypothesized that subtle variations in how a navigation task is performed might alter people's social perception of a mobile robot, leading them to interpret the robot's actions as conveying different intentions, attitudes, or physical characteristics, which may in turn impact acceptability. Hence, this thesis addressed the question: "How can we design a social navigation algorithm that enables the robot's motion to be adapted based on its impact on human social perception of the robot?"

Our first contribution, described in chapter 3, consisted in building a model linking our robot's motion and appearance variables to human social and physical perception of the robot. We constructed a corpus of robot motions representing different types of movement prosody by analogy with vocal prosody, and used it as the stimuli for two online and one in-person perception experiments. The experiment results determined which motion variables were significantly associated with each perceptual scale, guiding our navigation algorithm design. Our second contribution, described in chapter 4, consisted in designing a local planning algorithm capable of controlling our mobile robot's linear velocity profile such that the motion accurately and consistently matched a given movement prosody defined through the motion variables studied in chapter 3. We demonstrated how our particular trajectory features require careful consideration when designing the constraints that shape the space of prosody-compliant trajectories, based on the previously executed motions. We implemented our algorithm on a real robot and demonstrated its ability to plan trajectories with consistent movement prosody in chapter 5. Our third contribution, described in chapter 6, consisted in designing and running a HRI experiment in ecological conditions. We developed a full software architecture integrating our algorithm with other modules for perception and decision, and performed a thorough statistical analysis of the experimental

data. The experiment allows us to study whether the model of human perception derived in chapter 3 remained valid outside of lab conditions, as well as to study what effects two different robot movement prosody styles may have on participant behaviour. In the following sections, we detail each contribution.

Table 7.1: Perceptual scales

Adjective 1	**Adjective 2**
Aggressive	Gentle
Authoritative	Polite
Seems Confident	Doubtful, Hesitant
Inspires confidence	Doesn't inspire confidence
Nice	Disagreeable
Sturdy	Frail
Strong	Weak
Smooth	Abrupt
Rigid	Supple
Tender	Insensitive

7.1 Answers to our research questions

7.1.1 Which features of navigation motion contribute to human's social perception of the mobile robot?

We measured human perceptions using an impressionistic paradigm, asking participants to rate robot motions along ten perceptual scales, opposing adjectives which had been used by participants in a prior human-robot interaction study (Sasa, 2018) to describe a robot (shown for reference in Figure 7.1).

Based on the multi-modal and holistic nature of human robot interaction, we selected motion and appearance variables which spanned the degrees of freedom available on our mobile robot and proposed between two and six possible values for each variable. Three variables directly specified the robot's linear velocity profile:

- **kinematics type:** defining the acceleration, as well as maximum velocity;
- **motion sequence:** related to sequences of accelerations and decelerations;
- **variant:** related to the local shape of the velocity profile.

Four more variables alter the robot's appearance :

- **eye shape:** related to gaze and anthropomorphism;
- **head orientation:** related to gaze and attention;
- **base:** related to the stability of the mobile robot on its base;
- **motor sound:** an artificial manipulation performed for the online perception experiments (noisy or silent robot).

These corpus variables were systematically combined using different values of each variable, resulting in a corpus of 450 videos of robot motion being recorded. Through three perception experiments, we determined that each of these variables was significantly associated with variations in participant perception of the robot along different subsets of our perceptual scales. This suggests that each of these variables may be useful to explore in future social navigation works. Likewise, all of our perceptual scales were found to be significantly influenced by at least one of the corpus variables, suggesting that all of these scales can be useful in understanding human perceptions of a mobile robot.

This work was published as a long journal article in *Interaction Studies* (Scales et al., 2023). Our video corpus of robot motions is made available for further research [1].

7.1.2 How can we design an algorithm to accurately generate socio-affective trajectories in dynamic uncertain environments?

Our review of works in Social Navigation and functional expressive motion generation indicated that optimal control formulations were common in both fields due to the flexibility afforded by the specification of the desired motion through the cost function and constraints. However, none of the works aim to model socio-affects for mobile robot trajectories in dynamic uncertain environments. Moreover, the specific trajectory features we derived from our perception experiments had not been explored previously. In chapter 4, we presented how the variables from our motion corpus are incorporated into a receding horizon local planner as constraints, thereby re-shaping the space of possible trajectories to span only the space of those conveying a user-specified socio-affect. We consider how typical issues encountered in social navigation may impact the planner's ability to maintain the desired socio-affect accurately. The prosody constraints are re-formulated to explicitly account for the necessity of re-planning to handle dynamic environments. We also make use of a constraint hierarchy in order to maintain at least partial prosody compliance when there are no solutions that comply with all prosody constraints. We demonstrated our planner's ability to plan linear trajectories that accurately reproduced the distinguishing features of the different movement prosody styles in static and dynamic environments.

This algorithm significantly extends the algorithm developed during the author's master's thesis which was published at the *ICRA 2020* conference (Scales et al., 2020).

[1] *BotEmoMove* video corpus available at: https://osf.io/5csrg/

7.1.3 What is the impact on people of integrating our algorithm into a fully autonomous software architecture to perform an in-the-wild study?

We designed an ecological experiment in order to test the effect of two opposed movement prosody styles: confident (fast, high acceleration, smooth motion) and hesitant (slow, low acceleration, saccadic motion, sporadic decelerations). Our robot was deployed in the role of a flyer distributor robot in a university building entrance hall, where it detected people, rotated towards them, and approached them using our algorithm parameterized to use the confident or hesitant movement prosody. We developed a full architecture to enable the robot to autonomously perform the experiment. The statistical analysis of the experiment suggests that our model of movement prosody developed through the online and lab-based studies may be missing some dimensions, since participants systematically replied they perceived the robot as confident, even when seeing the hesitant movement prosody. Through the analysis of their explanations given during the semi-structured interviews, many of them mentioned the fact that the robot immediately came straight towards them, or did not hesitate between them and other people in the space. This suggests that the way in which the robot orients itself, and possibly the timing of that action may also be involved in movement prosody, impacting participants' perception. We also aimed to explore whether the perceived socio-affect could be related to participants' rating of the performance of the robot, however the overwhelming majority of participants gave the maximal rating, regardless of the movement prosody. In terms of participant spatial behaviour, there seemed to be a tendency for participants to distance themselves from the confident robot more. This effect was not statistically significant, and would be interesting to explore in further studies.

A version of our localization algorithm included in our HRI software architecture was published as a conference paper at *VISAPP 2021* (Scales, Rimel, et al., 2021).

7.2 Future works

7.2.1 Refining the definition of movement prosody

In chapter 3, we made an initial proposition for a set of motion variables that define movement prosody. The model of movement prosody's ties to social perception of the robot was constructed based on perception experiments where the robot only performed straight line motions. Subsequently, it remains unclear how the path followed by the robot as well as its use of the rotation degree of freedom would impact people's perceptions. Technically, the motion variables studied in our work can simply be applied to the rotational velocity and accelerations in a similar fashion to their linear counterparts. However, further experimentation would be required to determine whether the mapping from the motion variables to social perception would be the same. More generally, there are likely many other trajectory characteristics that could be explored beyond the restriction to piecewise linear velocity profiles.

7.2.2 Studying dynamics of movement prosody

Our model of movement prosody makes the assumption that a given motion will always have the same social interpretation, regardless of the interaction context or the respective roles of the person and robot. Prior works focusing on vocal prosody (Sasa, 2018) showed that the features or primitives which were used to express a given prosody could change over the course of an interaction. In our ecological experiment presented in chapter 6, the interaction with the robot is very short, which likely doesn't leave enough time to observe such dynamics, even if they do occur. Conducting longer studies using our navigation algorithm would be a valuable way to explore if a similar gradual alteration of movement prosody could be observed.

7.2.3 Augmenting our prosody-aware algorithm with other social navigation principles

In this thesis, we focus exclusively on the aspects of motion which may generate different social perceptions of the robot. The goal of the movement prosody constraints applied to our local planner is not to replace existing social navigation principles such as legibility, perceived safety, or personal space, but rather to complement them. Therefore, it would be interesting to combine such aspects into a common framework.

7.2.4 Absolute or relative perception of movement prosody?

In our perception experiments, participants were exposed to 45 different combinations of motion variable settings each. Although the experiment design required them to rate each video individually, it may be the case that they compare the different motions and consider them in relative terms. Beyond the relative consideration for the same robot, another aspect to explore would be whether the values of the motion variables such as the velocity and acceleration are affected by the robot's size and shape, and more generally, to what extent our model of movement prosody generalizes to other robots.

7.2.5 Towards an algorithm for real-time adaptation of movement prosody

The work presented in this thesis can be viewed as a stepping stone towards a robot that can dynamically adjust its movement prosody based on the person's own prosody and attitude. A prerequisite to such co-adaptation would be to develop a real-time perception algorithm capable of classifying a person's movement prosody. It may be interesting to test whether using the same model of movement prosody as the one we constructed for our robot would be coherent with human judgements of another person's attitudes and intentions. Given the subtle nature of vocal prosody, as well as certain motion variables used in our model of movement prosody, the algorithm should be able to detect subtle motion

cues. Extracting detailed information about the person's whole body motion is becoming increasingly reliable using sensors such as RGBD cameras (Ferrini & Lemaignan, 2022) like the one equipped on our robot. Works in the field of social signal processing are dedicated to extracting subtle social signals from sensor data (Vinciarelli et al., 2009), and it may be possible to use similar methods to those used for affect recognition such as (Filntisis et al., 2019).

References

Argyle, M., & Dean, J. (1965). Eye-Contact, Distance and Affiliation. *Sociometry*, *28*(3), 289. https://doi.org/10.2307/2786027

Auberge, V. (2019). Socio-affective glue robots and elderly isolated people. *Int Conf Emotional Attachment to Machines, EMTECH*, 25–26.

Augustine, A. C., Ryusuke, M., Liu, C., Ishi, C. T., & Ishiguro, H. (2020). Generation and evaluation of audio-visual anger emotional expression for android robot. *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 96–98. https://doi.org/10.1145/3371382.3378282

Baayen, R. H. (2008). *Analyzing linguistic data: A practical introduction to statistics using r*. Cambridge University Press. https://doi.org/10.1017/CBO9780511801686

Barchard, K. A., Lapping-Carr, L., Westfall, R. S., Fink-Armold, A., Banisetty, S. B., & Feil-Seifer, D. (2020). Measuring the perceived social intelligence of robots. *ACM Transactions on Human-Robot Interaction*, *9*(4). https://doi.org/10.1145/3415139

Bartneck, C., Kulić, D., Croft, E., & Zoghbi, S. (2009). Measurement Instruments for the Anthropomorphism, Animacy, Likeability, Perceived Intelligence, and Perceived Safety of Robots. *Int J Soc Robot*, *1*, 71–81. https://doi.org/10.1007/s12369-008-0001-3

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Beck, A., Stevens, B., Bard, K. A., & Cañamero, L. (2012). Emotional body language displayed by artificial agents. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *2*(1), 1–29. https://doi.org/10.1145/2133366.2133368

Belhassein, K., Buisan, G., Clodic, A., Alami, R., Belhassein, K., Buisan, G., Clodic, A., Alami, R., & Kathleen, B. (2019). Towards methodological principles for user studies in Human-Robot Interaction To cite this version : HAL Id : hal-02282600 Towards methodological principles for user studies in Human-Robot Interaction.

Bellman, R. (1957). *Dynamic Programming*. Dover Publications.

Brandl, C., Mertens, A., & Schlick, C. M. (2016). Human-Robot Interaction in Assisted Personal Services: Factors Influencing Distances That Humans Will Accept between Themselves and an Approaching Service Robot. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *26*(6), 713–727. https://doi.org/10.1002/hfm.20675

Breazeal, C., Kidd, C., Thomaz, A., Hoffman, G., & Berlin, M. (2005). Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 708–713. https://doi.org/10.1109/IROS.2005.1545011

Butler, J. T., & Agah, A. (2001). Psychological effects of behavior patterns of a mobile personal robot. *Autonomous Robots*, *10*(2), 185–202. https://doi.org/10.1023/A:1008986004181

Campbell, N. (2004). Perception of affect in speech - towards an automatic processing of paralinguistic information in spoken conversation. *Interspeech*. https://api.semanticscholar.org/CorpusID:32349328

Campbell, N., & Mokhtari, P. (2003). Voice quality: The 4th prosodic dimension. *Proc. 15th Int. Congr. Phonetic Sciences*, pp. 2417–2420.

Carpenter, J. (2013). The Quiet Professional: An investigation of U.S. military Explosive Ordnance Disposal personnel interactions with everyday field robots. *Doctoral dissertation University of Washington*.

Carpinella, C. M., Wyman, A. B., Perez, M. A., & Stroessner, S. J. (2017). The Robotic Social Attributes Scale (RoSAS): Development and Validation. *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*, 254–262.

Carton, D., Olszowy, W., Wollherr, D., & Buss, M. (2017). Socio-Contextual Constraints for Human Approach with a Mobile Robot. *International Journal of Social Robotics*, *9*(2), 309–327. https://doi.org/10.1007/s12369-016-0394-3

Chan, L., Zhang, B. J., & Fitter, N. T. (2021). Designing and validating expressive cozmo behaviors for accurately conveying emotions. *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 1037–1044. https://doi.org/10.1109/RO-MAN50785.2021.9515425

Chen, Y. F., Everett, M., Liu, M., & How, J. P. (2017). Socially aware motion planning with deep reinforcement learning. *IEEE International Conference on Intelligent Robots and Systems*, *2017-Septe*, 1343–1350. https://doi.org/10.1109/IROS.2017.8202312

Ciocca, M. (2020). *Balance Preservation and Collision Mitigation for Biped Robots* (Theses 2020GRALM042). Université Grenoble Alpes [2020-....] https://theses.hal.science/tel-03065088

Coşar, S., Fernandez-Carmona, M., Agrigoroaie, R., Pages, J., Ferland, F., Zhao, F., Yue, S., Bellotto, N., & Tapus, A. (2020). ENRICHME: Perception and Interaction of an Assistive Robot for the Elderly at Home. *International Journal of Social Robotics*, *12*(3), 779–805. https://doi.org/10.1007/S12369-019-00614-Y/FIGURES/30

Dautenhahn, K., Nehaniv, C. L., Walters, M. L., Robins, B., Kose-Bagci, H., Mirza, N. A., & Blow, M. (2009). KASPAR - a minimally expressive humanoid robot for human-robot interaction research. *Applied Bionics and Biomechanics*, *6*(3-4), 369–397. https://doi.org/10.1080/11762320903123567

Dellaert, F., Fox, D., Burgard, W., & Thrun, S. (1999). Monte carlo localization for mobile robots. *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, *2*, 1322–1328 vol.2. https://doi.org/10.1109/ROBOT.1999.772544

Di Cesare, G., De Stefani, E., Gentilucci, M., & De Marco, D. (2017). Vitality Forms Expressed by Others Modulate Our Own Motor Response: A Kinematic Study. *Frontiers in Human Neuroscience*, *11*, 565. https://doi.org/10.3389/fnhum.2017.00565

Dragan, A. D., Lee, K. C., & Srinivasa, S. S. (2013). Legibility and predictability of robot motion. *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 301–308. https://doi.org/10.1109/HRI.2013.6483603

Drumm, P. (2012). Köhler, W. In R. W. Rieber (Ed.), *Encyclopedia of the history of psychological theories* (pp. 610–612). Springer US. https://doi.org/10.1007/978-1-4419-0463-8_153

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/https://doi.org/10.1016/j.patrec.2005.10.010

Ferrini, L., & Lemaignan, S. (2022). Kinematically-consistent real-time 3d human body estimation for physical and social hri. *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 765–767. https://doi.org/10.1109/HRI53351.2022.9889358

Filntisis, P. P., Efthymiou, N., Koutras, P., Potamianos, G., & Maragos, P. (2019). Fusing body posture with facial expressions for joint recognition of affect in child–robot interaction. *IEEE Robotics and Automation Letters*, *4*(4), 4011–4018. https://doi.org/10.1109/LRA.2019.2930434

Fischer, K., Jensen, L. C., Suvei, S. D., & Bodenhagen, L. (2016). Between legibility and contact: The role of gaze in robot approach. *25th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2016*, 646–651. https://doi.org/10.1109/ROMAN.2016.7745186

Forer, S., Banisetty, S. B., Yliniemi, L., Nicolescu, M., & Feil-Seifer, D. (2018). Socially-aware navigation using non-linear multi-objective optimization. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–9. https://doi.org/10.1109/IROS.2018.8593825

Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, *4*(1), 23–33. https://doi.org/10.1109/100.580977

Francis, A., Pérez-D'Arpino, C., Li, C., Xia, F., Alahi, A., Alami, R., Bera, A., Biswas, A., Biswas, J., Chandra, R., Chiang, H.-T. L., Everett, M., Ha, S., Hart, J. W., How, J. P., Karnan, H., Lee, T.-W. E., Manso, L. J., Mirksy, R., … Apple. (2023). Principles and Guidelines for Evaluating Social Robot Navigation Algorithms. *ArXiv*, *abs/2306.1*. https://api.semanticscholar.org/CorpusID:259287246

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., & Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, *47*(6), 2280–2292. https://doi.org/https://doi.org/10.1016/j.patcog.2014.01.005

Gil, Ó., Garrell, A., & Sanfeliu, A. (2021). Social robot navigation tasks: Combining machine learning techniques and social force model. *Sensors*, *21*(21). https://doi.org/10.3390/s21217087

Girard-Rivier, M., Magnani, R., Aubergé, V., Sasa, Y., Tsvetanova, L., Aman, F., & Bayol, C. (2016). Ecological Gestures for HRI: the GEE Corpus. https://aclanthology.org/L16-1235

Gobl, C., & Ní Chasaide, A. (2003). The role of voice quality in communicating emotion, mood and attitude. *Speech Communication*, *40*(1-2), 189–212. https://doi.org/10.1016/S0167-6393(02)00082-1

Guillaume, L., Aubergé, V., Magnani, R., Aman, F., Cottier, C., Sasa, Y., Wolf, C., Nebout, F., Neverova, N., Bonnefond, N., Nègre, A., Tsvetanova, L., & Girard-Rivier, M. (2015). Hri in an ecological dynamic experiment: The gee corpus based approach for the emox robot. *2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 1–6. https://doi.org/10.1109/ARSO.2015.7428207

Hagane, S., & Venture, G. (2022). Robotic Manipulator's Expressive Movements Control Using Kinematic Redundancy. *Machines*, *10*(12), 1118. https://doi.org/10.3390/machines10121118

Hall, E. T., Birdwhistell, R. L., Bock, B., Bohannan, P., Diebold, A. R., Durbin, M., Edmonson, M. S., Fischer, J. L., Hymes, D., Kimball, S. T., Barre, W. L., Frank Lynch, S. J., McClellan, J. E., Marshall, D. S., Milner, G. B., Sarles, H. B., Trager, G. L., & Vayda, A. P. (1968). Proxemics [and comments and replies]. *Current Anthropology*, *9*(2/3), 83–108. http://www.jstor.org/stable/2740724

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, *4*(2), 100–107. https://doi.org/10.1109/TSSC.1968.300136

Hebesberger, D., Koertner, T., Gisinger, C., & Pripfl, J. (2017). A long-term autonomous robot at a care hospital: A mixed methods study on social acceptance and experiences of staff and older adults. *International Journal of Social Robotics*, *9*. https://doi.org/10.1007/s12369-016-0391-6

Henderson, M., & Ngo, T. D. (2021). Rrt-smp: Socially-encoded motion primitives for sampling-based path planning. *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 330–336. https://doi.org/10.1109/RO-MAN50785.2021.9515460

Hoffman, G., & Zhao, X. (2020). A primer for conducting experiments in human–robot interaction. *J. Hum.-Robot Interact.*, *10*. https://doi.org/10.1145/3412374

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, *6*(2), 65–70. Retrieved October 10, 2023, from http://www.jstor.org/stable/4615733

Honig, S., & Oron-Gilad, T. (2020). Comparing laboratory user studies and video-enhancedweb surveys for eliciting user gestures in human-robot interactions. *ACM/IEEE International Conference on Human-Robot Interaction*, 248–250. https://doi.org/10.1145/3371382.3378325

Honour, A., Banisetty, S. B., & Feil-Seifer, D. (2021). Perceived Social Intelligence as Evaluation of Socially Navigation. *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 519–523. https://doi.org/10.1145/3434074.3447226

Irfan, B., Kennedy, J., Lemaignan, S., Papadopoulos, F., Senft, E., & Belpaeme, T. (2018). Social Psychology and Human-Robot Interaction: An Uneasy Marriage. *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction - HRI '18*, 13–20. https://doi.org/10.1145/3173386.3173389

Jeong, W., & Lee, K. M. (2005). Cv-slam: A new ceiling vision-based slam technique. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3195–3200. https://doi.org/10.1109/IROS.2005.1545443

Joosse, M., Lohse, M., van Berkel, N., Sardar, A., & Evers, V. (2020). Making appearances: How robots should approach people. *ACM Transactions on Human-Robot Interaction*, *10*. https://doi.org/10.1145/3385121

Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., & Schaal, S. (2011). STOMP: Stochastic trajectory optimization for motion planning. *Proceedings - IEEE International Conference on Robotics and Automation*, 4569–4574. https://doi.org/10.1109/ICRA.2011.5980280

Kamezaki, M., Kobayashi, A., Yokoyama, Y., Yanagawa, H., Shrestha, M., & Sugano, S. (2019). A Preliminary Study of Interactive Navigation Framework with Situation-Adaptive Multimodal Inducement: Pass-By Scenario. *International Journal of Social Robotics*. https://doi.org/10.1007/s12369-019-00574-3

Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, *30*(7), 846–894. https://doi.org/10.1177/0278364911406761

Karunarathne, D., Morales, Y., Nomura, T., Kanda, T., & Ishiguro, H. (2019). Will Older Adults Accept a Humanoid Robot as a Walking Partner? *International Journal of Social Robotics*, *11*(2), 343–358. https://doi.org/10.1007/s12369-018-0503-6

Kavraki, L., Svestka, P., Latombe, J.-C., & Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, *12*(4), 566–580. https://doi.org/10.1109/70.508439

Khambhaita, H., & Alami, R. (2020). Viewing Robot Navigation in Human Environment as a Cooperative Activity. Springer, Cham. https://doi.org/10.1007/978-3-030-28619-4_25

Kirk, D. B. (1973). On the numerical approximation of the bivariate normal (tetrachoric) correlation coefficient. *Psychometrika*, *38*(2), 259–268. https://doi.org/10.1007/BF02291118

Kitagawa, R., Liu, Y., & Kanda, T. (2021). Human-inspired motion planning for omni-directional social robots. *2021 16th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 34–42.

Kivrak, H., Uluer, P., Kose, H., Gumuslu, E., Erol Barkana, D., Cakmak, F., & Yavuz, S. (2020). Physiological Data-Based Evaluation of a Social Robot Navigation System. *29th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2020*, 994–999. https://doi.org/10.1109/RO-MAN47096.2020.9223539

Knight, H. (2016). Expressive Motion for Low Degree-of-Freedom Robots. https://doi.org/10.1184/R1/6716615.v1

Knight, H., & Simmons, R. (2014). Expressive motion with x, y and theta: Laban Effort Features for mobile robots. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, *2014-Octob*(October), 267–273. https://doi.org/10.1109/ROMAN.2014.6926264

Knight, H., Thielstrom, R., & Simmons, R. (2016). Expressive path shape (Swagger): Simple features that illustrate a robot's attitude toward its goal in real time. *IEEE International Conference on Intelligent Robots and Systems*, *2016-Novem*, 1475–1482. https://doi.org/10.1109/IROS.2016.7759240

Kollmitz, M., Hsiao, K., Gaa, J., & Burgard, W. (2015). Time dependent planning on a layered social cost map for human-aware robot navigation. *2015 European Conference on Mobile Robots, ECMR 2015 - Proceedings*. https://doi.org/10.1109/ECMR.2015.7324184

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Kruse, T., Basili, P., Glasauer, S., & Kirsch, A. (2012). Legible robot navigation in the proximity of moving humans. *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts, ARSO*, 83–88. https://doi.org/10.1109/ARSO.2012.6213404

Kruse, T., Pandey, A. K., Alami, R., & Kirsch, A. (2013). Human-Aware Robot Navigation: A Survey. *Robotics and Autonomous Systems*, *61*(12), pp.1726–1743. https://hal.archives-ouvertes.fr/hal-01684295

Lastrico, L., Duarte, N. F., Carfí, A., Rea, F., Mastrogiovanni, F., Sciutti, A., & Santos-Victor, J. (2022). If You Are Careful, So Am I! How Robot Communicative Motions Can Influence Human Approach in a Joint Task. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13817 LNAI*, 267–279. https://doi.org/10.1007/978-3-031-24667-8_24/COVER

Leech, B. L. (2002). Asking questions: Techniques for semistructured interviews. *PS: Political Science & Politics*, *35*(4), 665–668. https://doi.org/10.1017/S1049096502001129

Lenth, R. V. (2023). *Emmeans: Estimated marginal means, aka least-squares means*. https://CRAN.R-project.org/package=emmeans

Li, J. (2015). The benefit of being physically present: A survey of experimental works comparing copresent robots, telepresent robots and virtual agents. *International Journal of Human-Computer Studies*, *77*, 23–37. https://doi.org/https://doi.org/10.1016/j.ijhcs.2015.01.001

Lohse, M., Berkel, N. V., Dijk, E. M. V., Joosse, M. P., Karreman, D. E., & Evers, V. (2013). The influence of approach speed and functional noise on users' perception of a robot. *IEEE International Conference on Intelligent Robots and Systems*, 1670–1675. https://doi.org/10.1109/IROS.2013.6696573

Lu, Y. (2015). *Etude contrastive de la prosodie audio-visuelle des affects sociaux en chinois mandarin vs.français : vers une application pour l'apprentissage de la langue étrangère ou seconde* (Theses 2015GREAL001). Université Grenoble Alpes. https://theses.hal.science/tel-01227267

Luber, M., Spinello, L., Silva, J., & Arras, K. O. (2012). Socially-aware robot navigation: A learning approach. *IEEE International Conference on Intelligent Robots and Systems*, 902–907. https://doi.org/10.1109/IROS.2012.6385716

Luke, S. G. (2017). Evaluating significance in linear mixed-effects models in r. *Behavior Research Methods*, *49*, 1494–1502.

Magnani, R., Aubergé, V., Bayol, C., & Sasa, Y. (2017). Bases of Empathic Animism Illusion: audio-visual perception of an object devoted to becoming perceived as a subject for HRI. *Proc. of the 1st International Workshop on Vocal Interactivity in-and-between Humans, Animals and Robots – VIHAR 2017*.

Matsumoto, M. (2021). Fragile Robot: The Fragility of Robots Induces User Attachment to Robots. *International Journal of Mechanical Engineering and Robotics Research*, *10*(10), 536–541. https://doi.org/10.18178/ijmerr.10.10.536-541

Mavrogiannis, C., Baldini, F., Wang, A., Zhao, D., Trautman, P., Steinfeld, A., & Oh, J. (2023). Core challenges of social robot navigation: A survey. *J. Hum.-Robot Interact.*, *12*(3). https://doi.org/10.1145/3583741

Mavrogiannis, C., Hutchinson, A. M., MacDonald, J., Alves-Oliveira, P., & Knepper, R. A. (2019). Effects of Distinct Robot Navigation Strategies on Human Behavior in a Crowded Environment. *ACM/IEEE International Conference on Human-Robot Interaction*, *2019-March*(March), 421–430. https://doi.org/10.1109/HRI.2019.8673115

Mayne, D., Rawlings, J., Rao, C., & Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*(6), 789–814. https://doi.org/https://doi.org/10.1016/S0005-1098(99)00214-9

McGinn, C., & Torre, I. (2019). Can you tell the robot by the voice? an exploratory study on the role of voice in the perception of robots. *Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction*, 211–221.

McGurk, H., & MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, *264*(5588), 746–748. https://doi.org/10.1038/264746a0

Menne, I. M., & Schwab, F. (2018). Faces of Emotion: Investigating Emotional Facial Expressions Towards a Robot. *International Journal of Social Robotics*, *10*(2), 199–209. https://doi.org/10.1007/s12369-017-0447-2

Mizoguchi, H., Sato, T., Takagi, K., Nakao, M., & Hatamura, Y. (1997). Realization of expressive mobile robot. *Proceedings - IEEE International Conference on Robotics and Automation*, *1*, 581–586. https://doi.org/10.1109/ROBOT.1997.620099

Mizumaru, K., Satake, S., Kanda, T., & Ono, T. (2019). Stop doing it! approaching strategy for a robot to admonish pedestrians. *ACM/IEEE International Conference on Human-Robot Interaction*, *2019-March*, 449–457. https://doi.org/10.1109/HRI.2019.8673017

Moon, A., Hashmi, M., Loos, H. F. M. V. D., Croft, E. A., & Billard, A. (2021). Design of hesitation gestures for nonverbal human-robot negotiation of conflicts. *J. Hum.-Robot Interact.*, *10*(3). https://doi.org/10.1145/3418302

Moon, A., Parker, C. A. C., Croft, E. A., & Van der Loos, H. F. M. (2013). Design and Impact of Hesitation Gestures during Human-Robot Resource Conflicts. *Journal of Human-Robot Interaction*, *2*(3). https://doi.org/10.5898/jhri.2.3.moon

Morales Saiki, L. Y., Satake, S., Huq, R., Glass, D., Kanda, T., & Hagita, N. (2012). How do people walk side-by-side?: Using a computational model of human behavior for a social robot. *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI '12*, (May 2016), 301. https://doi.org/10.1145/2157689.2157799

Mumm, J., & Mutlu, B. (2011). Human-robot proxemics: Physical and psychological distancing in human-robot interaction. *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 331–338. https://doi.org/10.1145/1957656.1957786

Munzer, T., Mollard, Y., & Lopes, M. (2017). Impact of robot initiative on human-robot collaboration. *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 217–218. https://doi.org/10.1145/3029798.3038373

Mutlu, B., & Forlizzi, J. (2008). Robots in organizations. *Proceedings of the 3rd international conference on Human robot interaction - HRI '08*, (May 2014), 287. https://doi.org/10.1145/1349822.1349860

Naik, L., Palinko, O., Bodenhagen, L., & Krüger, N. (2021). Multi-modal proactive approaching of humans for human-robot cooperative tasks. *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 323–329.

Nomura, T., Suzuki, T., Kanda, T., & Kato, K. (2006). Measurement of negative attitudes toward robots. *Interaction Studies*, *7*(3), 437–454. https://doi.org/https://doi.org/10.1075/is.7.3.14nom

Noordzij, M. L., Schmettow, M., & Lorijn, M. R. (2014). Is an accelerating robot perceived as energetic or as gaining in speed? *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, 258–259. https://doi.org/10.1145/2559636.2559793

Okal, B., & Arras, K. O. (2016). Learning socially normative robot navigation behaviors with Bayesian inverse reinforcement learning. *Proceedings - IEEE International Conference on Robotics and Automation*, *2016-June*, 2889–2895. https://doi.org/10.1109/ICRA.2016.7487452

Pacchierotti, E., Christensen, H. I., & Jensfelt, P. (2005). Human-robot embodied interaction in hallway settings: A pilot user study. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, *2005*(September), 164–171. https://doi.org/10.1109/ROMAN.2005.1513774

Park, J. J. (2016). Graceful Navigation for Mobile Robots in Dynamic and Uncertain Environments by. *Disertation*.

Paulin, R., Fraichard, T., & Reignier, P. (2019). Using Human Attention to Address Human-Robot Motion. *IEEE Robotics and Automation Letters*, *4*(2), 2038–2045. https://doi.org/10.1109/LRA.2019.2899429

Peine, A., Marshall, B. L., Martin, W., Neven, L., Ertner, M., & Lassen, A. J. (2021). Chapter 3 Fragile robots and coincidental innovation. https://doi.org/10.4324/9780429278266-3

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., & Ng, A. (2009). Ros: An open-source robot operating system. *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*.

Ramirez, O. A., Khambhaita, H., Chatila, R., Chetouani, M., & Alami, R. (2016). Robots learning how and where to approach people. *25th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2016*, *1*, 347–353. https://doi.org/10.1109/ROMAN.2016.7745154

Reeves, B., & Nass, C. I. (1996). *The media equation: How people treat computers, television, and new media like real people and places*. Cambridge University Press.

Reinhardt, J., Prasch, L., & Bengler, K. (2021). Back-off. *ACM Transactions on Human-Robot Interaction*, *10*(3), 1–25. https://doi.org/10.1145/3418303

Repiso, E., Ferrer, G., & Sanfeliu, A. (2017). On-line adaptive side-by-side human robot companion in dynamic urban environments. *IEEE International Conference on Intelligent Robots and Systems*, *2017-Septe*, 872–877. https://doi.org/10.1109/IROS.2017.8202248

Rios-Martinez, J., Spalanzani, A., & Laugier, C. (2015). From Proxemics Theory to Socially-Aware Navigation: A Survey. *International Journal of Social Robotics*, *7*(2), 137–153. https://doi.org/10.1007/s12369-014-0251-1

Rios-Martinez, J., Renzaglia, A., Spalanzani, A., Martinelli, A., & Laugier, C. (2012). Navigating between people: A stochastic optimization approach. *Proceedings - IEEE International Conference on Robotics and Automation*, (July 2015), 2880–2885. https://doi.org/10.1109/ICRA.2012.6224934

Robair mobile robot, designed and built by fabmastic, grenoble. (2021).

Robinson, F. A., Velonaki, M., & Bown, O. (2021). Smooth operator: Tuning robot perception through artificial movement sound. *ACM/IEEE International Conference on Human-Robot Interaction*, 53–62. https://doi.org/10.1145/3434073.3444658

Rosenthal-von der Pütten, A. M., Schulte, F. P., Eimler, S. C., Sobieraj, S., Hoffmann, L., Maderwald, S., Brand, M., & Krämer, N. C. (2014). Investigations on empathy towards humans and robots using fMRI. *Computers in Human Behavior*, *33*, 201–212. https://doi.org/10.1016/j.chb.2014.01.004

Rosmann, C., Hoffmann, F., & Bertram, T. (2015). Timed-Elastic-Bands for time-optimal point-to-point nonlinear model predictive control. *2015 European Control Conference, ECC 2015*, 3352–3357. https://doi.org/10.1109/ECC.2015.7331052

Rösmann, C., Hoffmann, F., & Bertram, T. (2015). Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control. *2015 European Control Conference (ECC)*, 3352–3357. https://doi.org/10.1109/ECC.2015.7331052

S. R. Searle, F. M. S., & Milliken, G. A. (1980). Population marginal means in the linear model: An alternative to least squares means. *The American Statistician*, *34*(4), 216–221. https://doi.org/10.1080/00031305.1980.10483031

Sabelli, A., & Kanda, T. (2016). Robovie as a mascot: A qualitative study for long-term presence of robots in a shopping mall. *International Journal of Social Robotics*, *8*. https://doi.org/10.1007/s12369-015-0332-9

Saerbeck, M., & Bartneck, C. (2010). Perception of affect elicited by robot motion, 53–60. https://doi.org/10.1109/hri.2010.5453269

Saldien, J., Vanderborght, B., Goris, K., Van Damme, M., & Lefeber, D. (2014). A Motion System for Social and Animated Robots. *International Journal of Advanced Robotic Systems*, *11*(5), 72. https://doi.org/10.5772/58402

Sasa, Y. (2018). *Intelligence Socio-Affective pour un Robot : primitives langagières pour une interaction évolutive d'un robot de l'habitat intelligent* (Theses 2018GREAM041). Université Grenoble Alpes. https://theses.hal.science/tel-01945238

Sasa, Y., & Aubergé, V. (2016). Perceived isolation and elderly boundaries in eee (emoz elderly expressions) corpus: Appeal to communication dynamics with a socio-affectively gluing robot in a smart home. *Gerontechnology*, *15*.

Sasa, Y., & Aubergé, V. (2017). SASI: perspectives for a socio-affectively intelligent HRI dialog system. *1st Workshop on "Behavior, Emotion and Representation: Building Blocks of Interaction"*. https://hal.inria.fr/hal-01615470

Savery, R., Rose, R., & Weinberg, G. (2019). Establishing human-robot trust through music-driven robotic emotion prosody and gesture. *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 1–7. https://doi.org/10.1109/RO-MAN46459.2019.8956386

Savery, R., Zahray, L., & Weinberg, G. (2021). Emotional musical prosody for the enhancement of trust: Audio design for robotic arm communication. *Paladyn, Journal of Behavioral Robotics*, *12*(1), 454–467. https://doi.org/doi:10.1515/pjbr-2021-0033

Scales, P., Aubergé, V., & Aycard, O. (2021). From vocal prosody to movement prosody, from HRI to understanding humans. *VIHAR Vocal Interactivity in-and-between Humans, Animals and Robots*. https://hal.science/hal-03504825

Scales, P., Aubergé, V., & Aycard, O. (2022). Socio-expressive robot navigation: How motion profiles can convey frailty and confidence. *2022 10th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, 1–8. https://doi.org/10.1109/ACIIW57231.2022.10086013

Scales, P., Aubergé, V., & Aycard, O. (2023). From vocal prosody to movement prosody, from hri to understanding humans. *Interaction Studies*, *24*(1), 131–168. https://doi.org/https://doi.org/10.1075/is.22010.sca

Scales, P., Aycard, O., & Aubergé, V. (2020). Studying navigation as a form of interaction: A design approach for social robot navigation methods. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 6965–6972. https://doi.org/10.1109/ICRA40945.2020.9197037

Scales, P., Rimel, M., & Aycard, O. (2021). Visual-based Global Localization from Ceiling Images using Convolutional Neural Networks. *16th International Conference on Computer Vision Theory and Applications*, 927–934. https://doi.org/10.5220/0010248409270934

Scandolo, L., & Fraichard, T. (2011). An anthropomorphic navigation scheme for dynamic scenarios. *2011 IEEE International Conference on Robotics and Automation*, 809–814. https://doi.org/10.1109/ICRA.2011.5979772

Scheflen, A. E., & Scheflen, A. (1972). *Body language and the social order: Communication as behavioral control*. Prentice-Hall.

Schröder, M. (2003). Experimental study of affect bursts. *Speech Communication*, *40*(1-2), 99–116. https://doi.org/10.1016/S0167-6393(02)00078-X

Schulman, J., Ho, J., Lee, A. X., Awwal, I., Bradlow, H., & Abbeel, P. (2013). Finding locally optimal, collision-free trajectories with sequential convex optimization. *Robotics: Science and Systems IX*. https://api.semanticscholar.org/CorpusID:2393365

Schulz, T., Holthaus, P., Amirabdollahian, F., Koay, K. L., Torresen, J., & Herstad, J. (2020). Differences of Human Perceptions of a Robot Moving using Linear or Slow in, Slow

out Velocity Profiles When Performing a Cleaning Task. *2019 28th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2019*. https://doi.org/10.1109/RO-MAN46459.2019.8956355

Sciutti, A., Ansuini, C., Becchio, C., & Sandini, G. (2015). Investigating the ability to read others' intentions using humanoid robots. *Frontiers in psychology*, *6*, 1362. https://doi.org/10.3389/fpsyg.2015.01362

Seibt, J., Vestergaard, C., & Damholdt, M. F. (2021). The complexity of human social interactions calls for mixed methods in hri: Comment on "a primer for conducting experiments in human-robot interaction," by g. hoffman and x. zhao. *J. Hum.-Robot Interact.*, *10*(1). https://doi.org/10.1145/3439715

Sekino, H., Kasano, E., Hsieh, W.-F., Sato-Shimokawara, E., & Yamaguchi, T. (2020). Robot behavior design expressing confidence/unconfidence based on human behavior analysis. *2020 17th International Conference on Ubiquitous Robots (UR)*, 278–283. https://doi.org/10.1109/UR49135.2020.9144862

Sharpe, D. (2015). Your chi-square test is statistically significant: Now what? *Practical Assessment, Research and Evaluation*, *20*, 1–10.

Shiomi, M., Shatani, K., Minato, T., & Ishfguro, H. (2018). Does a Robot's Subtle Pause in Reaction Time to People's Touch Contribute to Positive Influences? *RO-MAN 2018 - 27th IEEE International Symposium on Robot and Human Interactive Communication*, 364–369. https://doi.org/10.1109/ROMAN.2018.8525849

Shiomi, M., Zanlungo, F., Hayashi, K., & Kanda, T. (2014). Towards a Socially Acceptable Collision Avoidance for a Mobile Robot Navigating Among Pedestrians Using a Pedestrian Model. *International Journal of Social Robotics*, *6*(3), 443–455. https://doi.org/10.1007/s12369-014-0238-y

Shochi, T. (2008). *Prosodie des affects socioculturels en japonais, et anglais: à la recherche des vrais et faux-amis pour le parcours de l'apprenant* (Theses). Université Stendhal - Grenoble III. https://theses.hal.science/tel-00366612

Siciliano, B., & Slotine, J.-J. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, 1211–1216 vol.2. https://doi.org/10.1109/ICAR.1991.240390

Siciliano, B., & Khatib, O. (Eds.). (2016). *Springer handbook of robotics*. Springer. https://doi.org/10.1007/978-3-319-32552-1

Sisbot, E. A., Marin-Urias, K. F., Alami, R., & Siméon, T. (2007). A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, *23*(5), 874–883. https://doi.org/10.1109/TRO.2007.904911

Song, S., & Yamada, S. (2018). Designing expressive lights and in-situ motions for robots to express emotions. *HAI 2018 - Proceedings of the 6th International Conference on Human-Agent Interaction*, 222–228. https://doi.org/10.1145/3284432.3284458

Sorrentino, A., Khalid, O., Coviello, L., Cavallo, F., & Fiorini, L. (2021). Modeling human-like robot personalities as a key to foster socially aware navigation. *2021 30th IEEE Inter-*

national Conference on Robot and Human Interactive Communication, RO-MAN 2021, 95–101. https://doi.org/10.1109/RO-MAN50785.2021.9515556

Sripathy, A., Bobu, A., Li, Z., Sreenath, K., Brown, D. S., & Dragan, A. D. (2022). Teaching Robots to Span the Space of Functional Expressive Motion. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 13406–13413. https://doi.org/10.1109/IROS47612.2022.9981964

Stern, D. N. (2010). *Forms of Vitality: Exploring Dynamic Experience in Psychology, the Arts, Psychotherapy, and Development*. Oxford University Press. https://doi.org/10.1093/med:psych/9780199586066.001.0001

Suguitan, M., Gomez, R., & Hoffman, G. (2020). Moveae: Modifying affective robot movements using classifying variational autoencoders. *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 481–489.

Sviestins, E., Kanda, T., Hagita, N., Ishiguro, H., & Mitsunaga, N. (2007). Speed adaptation for a robot walking with a human. (May 2014), 349. https://doi.org/10.1145/1228716.1228763

Szafir, D., Mutlu, B., & Fong, T. (2014). Communication of intent in assistive free flyers. *ACM/IEEE International Conference on Human-Robot Interaction*, 358–365. https://doi.org/10.1145/2559636.2559672

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going deeper with convolutions.

Takenaka, H. (2005). Loss experience and rebirth of elderly people. Seitosha Publishing.

Tanaka, K. (1997). Geratology isagoge. Nihon hyoron sha.

Tedrake, R. (2023). *Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation*. https://underactuated.csail.mit.edu

Tennent, H., Moore, D., Jung, M., & Ju, W. (2017). Good vibrations: How consequential sounds affect perception of robotic arms. *RO-MAN 2017 - 26th IEEE International Symposium on Robot and Human Interactive Communication*, *2017-January*, 928–935. https://doi.org/10.1109/ROMAN.2017.8172414

Thrun, S. (1998). Finding landmarks for mobile robot navigation. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, *2*, 958–963 vol.2. https://doi.org/10.1109/ROBOT.1998.677210

Torre, I., Linard, A., Steen, A., Tumová, J., & Leite, I. (2021). Should robots chicken? How anthropomorphism and perceived autonomy influence trajectories in a game-theoretic problem. *ACM/IEEE International Conference on Human-Robot Interaction*, 370–379. https://doi.org/10.1145/3434073.3444687

Trautman, P., & Krause, A. (2010). Unfreezing the robot: Navigation in dense, interacting crowds. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 797–803. https://doi.org/10.1109/IROS.2010.5654369

Tsvetanova, L., Aubergé, V., & Sasa, Y. (2017). Multimodal breathiness in interaction : From breathy voice quality to global breathy "body behavior quality". *Proc. of the 1st International Workshop on Vocal Interactivity in-and-between Humans, Animals and Robots – VIHAR 2017*.

Van Otterdijk, M. T., Neggers, M. M., Torresen, J., & Barakova, E. I. (2021). Preferences of Seniors for Robots Delivering a Message with Congruent Approaching Behavior. *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts, ARSO*, *2021-July*, 65–71. https://doi.org/10.1109/ARSO51874.2021.9542833

Vannucci, F., Di Cesare, G., Rea, F., Sandini, G., & Sciutti, A. (2019). A Robot with Style: Can Robotic Attitudes Influence Human Actions? *IEEE-RAS International Conference on Humanoid Robots*, *2018-Novem*, 952–957. https://doi.org/10.1109/HUMANOIDS.2018.8625004

Vaufreydaz, D., Johal, W., & Combe, C. (2016). Starting engagement detection towards a companion robot using multimodal features. *Robotics and Autonomous Systems*, *75*, 4–16. https://doi.org/10.1016/j.robot.2015.01.004

Vega, A., Manso, L. J., Macharet, D. G., Bustos, P., & Núñez, P. (2019). Socially aware robot navigation system in human-populated and interactive environments based on an adaptive spatial density function and space affordances. *Pattern Recognition Letters*, *118*, 72–84. https://doi.org/10.1016/j.patrec.2018.07.015

Venture, G., & Kulić, D. (2019). Robot Expressive Motions. *ACM Transactions on Human-Robot Interaction*, *8*(4), 1–17. https://doi.org/10.1145/3344286

Vinciarelli, A., Pantic, M., & Bourlard, H. (2009). Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, *27*(12), 1743–1759. https://doi.org/https://doi.org/10.1016/j.imavis.2008.11.007

von Laban, R., & Ullmann, L. (1975). *Modern educational dance*. Macdonald; Evans. https://books.google.fr/books?id=zY-1AAAAIAAJ

Walker, N., Mavrogiannis, C., Srinivasa, S., & Cakmak, M. (2022). Influencing Behavioral Attributions to Robot Motion During Task Execution. In A. Faust, D. Hsu, & G. Neumann (Eds.), *Proceedings of the 5th conference on robot learning* (pp. 169–179). PMLR. https://proceedings.mlr.press/v164/walker22a.html

Watanabe, K., Greenberg, Y., & Sagisaka, Y. (2014). Sentiment analysis of color attributes derived from vowel sound impression for multimodal expression. *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, 1–5. https://doi.org/10.1109/APSIPA.2014.7041586

White, M. J. (1975). Interpersonal Distance as Affected by Room Size, Status, and Sex. *The Journal of Social Psychology*, *95*(2), 241–249. https://doi.org/10.1080/00224545.1975.9918710

Williams, G., Drews, P., Goldfain, B., Rehg, J. M., & Theodorou, E. A. (2016). Aggressive driving with model predictive path integral control. *Proceedings - IEEE International Conference on Robotics and Automation*, *2016-June*, 1433–1440. https://doi.org/10.1109/ICRA.2016.7487277

Winter, B. (2019). *Statistics for linguists: An introduction using r (1st ed.)* Routledge. https://doi.org/https://doi.org/10.4324/9781315165547

Xu, Y., Chakhachiro, T., Kathuria, T., & Ghaffari, M. (2023). Solo t-dirl: Socially-aware dynamic local planner based on trajectory-ranked deep inverse reinforcement learning.

2023 IEEE International Conference on Robotics and Automation (ICRA), 12045–12051. https://doi.org/10.1109/ICRA48891.2023.10160536

Yoshioka, G., Sakamoto, T., & Takeuchi, Y. (2015). Inferring affective states from observation of a robot's simple movements. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, *2015-Novem*, 185–190. https://doi.org/10.1109/ROMAN.2015.7333582

Zecca, M., Endo, N., Momoki, S., Itoh, K., & Takanishi, A. (2008). Design of the humanoid robot KOBIAN - preliminary analysis of facial and whole body emotion expression capabilities-. *2008 8th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2008*, 487–492. https://doi.org/10.1109/ICHR.2008.4755969

Zheng, P. (2022). *Towards safe robot arm motion close to humans* (Theses 2022GRALM042). Université Grenoble Alpes [2020-....] https://theses.hal.science/tel-04067986

Zhou, A., & Dragan, A. D. (2018). Cost Functions for Robot Motion Style. *IEEE International Conference on Intelligent Robots and Systems*, 3632–3639. https://doi.org/10.1109/IROS.2018.8594433

Appendix A

Robot Motion Corpus Acquisition Details

A.1 Video Corpus Creation

A.1.1 Camera Equipment

The following equipment was used to film the final corpus videos:

- Camera body: Canon EOS 5D IV ;
- Lens : Canon EF 16-35mm f/4 L IS ;
- a Tripod to mount the camera.

A.1.2 Camera Settings

Below, we present the settings we decided to use to capture the video corpus. The footage was recorded in 4K (3840 x 2160 pixels), which required the use of cropping (1.64x). Due to this, the 16-35mm Lens acts as a 26-57mm lens.

- Video recording format: 4K 25.00Fps MJPG MOV;
- Exposure: Manual;
- White balance: Automatic;
- Aperture: f/4;
- Shutter speed: 1/60;

- ISO speed: between 400 and 800, depending on the lighting (time of day, weather). This was adjusted such that the final video result when viewed on a screen would have similar colors and visual aspect to the rest of the video corpus.

A.1.3 Video Post-processing

375 videos were filmed in total. The first pre-processing step was to split the videos corresponding to motion sequence E into two parts, essentially two motion sequences (we used the ffmpeg library for this). After the split, we refer to the acceleration to constant speed motion sequence as sequence E, and the constant speed to deceleration as sequence F. This brings the total number of videos to 450. All of these videos have audio tracks, and we do not include the muted copies in the corpus since for experimental purposes, the speakers or video player can be muted.

The original files were very large due to the 4K resolution and high-quality MJPEG format. Video playback also proved to require a moderately powerful processor. Our goal was to re-use the videos in an online experiment where participants would be using their own computers to view the videos, hence the size had to be reduced and the videos had to be re-encoded to a lightweight format, all while maintaining as much quality as possible. We consulted the videographer who recommended to re-encode the videos using the H.264 codec (we used the Handbrake software to do this). This reduced the size of the corpus from around 220GB to 33GB. If further optimization was required, we could downscale the videos to a resolution of 1920x1080 (also using Handbrake), given that most people likely do not use displays with a higher resolution.

When recording the videos, the photographer was operating the camera, and an experimenter was in charge of positioning the robot, selecting the parameters (eye shape, variant), selecting and launching the velocity profile execution, and manually controlling the head rotation if needed. The experimenter and photographer agreed on when to start recording, and the experimenter waited three or four seconds before starting the robot movement. The experimenter said "stop" once the robot had stopped moving for three or four seconds. In order to remove the sound of the experimenter saying stop, and to obtain a waiting time of two seconds before and after the motion, we cut on or two seconds off the beginning and the end of each video in a semi-automated fashion (a script using ffmpeg was applied with different parameters depending on the batch of videos being processed).

A.2 Discussion

The camera settings and equipment were the result of some compromises. Initially, a 24-70mm lens with 1/125 shutter speed (f/2.8 L) was selected. Due to the optical cropping factor and the wide field of view to be covered (the longest robot movement was 6 meters long), we had to place the camera quite near the robot. This forced us to use the 16-35mm lens with a 1/60 shutter speed(effectively losing 1 EV), despite the fact that it would have allowed us to work with better light. The 1/125 shutter speed could have reduced the motion blur when using the saccade motion variant, although it may have made the room's neon

lighting oscillations visible.

Despite the varying natural light, we decided not to use the automatic ISO in order to make sure we always had an adequate amount of under-exposure. Slight under-exposure was necessary in order to make the RobAIR's LED eyes clearly visible.

In hindsight, it may have proved beneficial to film the corpus in a lower resolution for several reasons:

- 4K files were very large, which meant regularly transferring files off the camera;
- with our camera, filming at a full HD (1920x1080) resolution would have allowed us to avoid using the crop factor, and to use the 24-70mm lens. This would also have allowed to film at 50 or 100 fps in case we wanted to use slow-motion.

Alternatively, with a Canon EOS R6 or R5 instead of our 5D IV, we could have used a very small crop factor (1.07x) even while filming 4K. The problem of large filesize would still have to be considered.

One aspect to be cautious of when filming such a corpus is to make sure not to lose any data. We suffered from some problems while using a card reader connected to a PC, which corrupted the card. Fortunately, we were able to recover the data by using the protocol defined by the camera's manufacturer, consisting in directly plugging the camera into the pc via a cable, and using their software. This method avoids file corruption and guarantees a proper file transfer.

Appendix B

Tool: Synchronized sensor network

B.1 Motivation

In order to capture the human-robot interaction we required access to sensors on the robot platform as well as external computers placed around the environment to capture several different angles with cameras and lidars. Given the often subtle nature of social signals, synchronisation of the sensor data from these different sources was necessary, and had to be accurate, ideally to less than $100ms$.

B.2 Implementation

Within the ROS ecosystem, the immediate solution that comes to mind is to simply run ROS drivers on each machine to capture the sensor information, and record all the data on a single PC using the rosbag tool. There are two issues with this approach: data will be timestamped according to when the recording PC received the message, and centralising the data recording on a single machine when using several image streams can lead to very high network usage. If the network is saturated, this will impact the arrival time of new data to the recording machine, therefore breaking the synchronisation.

To avoid overloading the network, we record data on each machine individually. Since the data would be recorded with timestamps corresponding to each machine's local clock which may drift out of synchronisation, we ensure accurate clock synchronization by using chrony[1]. The bagfiles containing sensor data from each machine are later merged into a single file, with accurate synchronization of robot sensors, external sensors, and internal logs about the navigation algorithm. The sensor data includes high quality audio from professional grade microphones on the robot, odometry information, RGB streams, depth image stream, UI and sound output requests, and lidar data.

[1] https://chrony-project.org/

Appendix C

Impact of covid-19 pandemic on the thesis

In this appendix, we discuss the impact of the covid-19 pandemic on our research methodology, and present the two in-person HRI experiments that had to be abandoned in favour of the online experiments presented in this thesis.

C.1 Impact on methodology

The first goal of our work is to determine which aspects of the robot's motion impact people's social perceptions of the robot. There is a bootstrap problem since without having an adjustable navigation algorithm, we cannot perform experiments with autonomous robots, and without performing experiments it seems hard to determine which motion characteristics one should explore and implement into an algorithm. Initially, we planned to approach the issue by performing real in-person experiments involving wizard of oz protocols, by teleoperating the robot. The idea would have been to then analyse the teleoperator and the interacting human's motions in relation to their behaviour and interviews, to determine which motion variables seemed important. Additionally, rather than imposing a set of adjectives or questionnaire scales for the participants to describe the robot, we would have asked open-ended questions similar to the flyer distribution experiment presented in the final chapter of this thesis, and analysed the interviews to find recurring patterns in how participants described the robot. Unfortunately, in-person experiments were functionally impossible for several months, and after investing months of design and preparation time into two in-person experiments, we opted to switch to an online-based study instead. This altered the methodology. Assuming the effect of subtle motion variations would be even more difficult for participants to perceive in an online study, we decided to ask participants to respond to a fixed set of perceptual scales rather than let the perceptions emerge from free-form responses. Similarly, instead of the motion characteristics emerging from human teleoperation data, we designed a motion corpus a priori. The online methodology also poses problems with respect to how well the results will generalize to real interactions,

despite our attempts to minimize the amount of perceptual artifacts induced by the online format.

Originally, an online experiment was planned as part of the thesis, but only at the end once relevant motion variables and perceptions had been identified in real experiments. The online study would have served as a way to test the generalization across a broader range of participants than is available when performing real experiments. The study would have been less cumbersome, with fewer parameters, and based on motions which were already found to be relevant in real situations.

C.2 In-person experiments aborted due to covid-19

C.2.1 First experiment: impact of interaction modalities on the human-human or human-robot relation

The goal of the first cancelled experiment was to create a corpus of high-precision multimodal data traces acquired during Human-Robot Interactions involving navigation and spatial aspects to solve tasks. The aim was to study how the human-human and human-robot relationship is affected when two people are subjected to different interaction conditions involving another human, a teleoperated robot, or an autonomous robot. The experimental protocol was long and elaborate, requiring participants to stay for one hour or more in the DOMUS living lab smart apartment. In order to capture accurate, complete and multimodal data for the corpus, we implemented a multi-computer synchronised sensor network. The first lockdown (March to July 2020) began during the technical preparations for the experiment, once the design was already well underway, stopping us from accessing the living lab, the robot, and the sensor network. The sensor network implementation was re-used in our final experiment presented in this thesis.

C.2.2 Second experiment: influence of robot motion energy level on humans in a joint navigation task

The aim of our second cancelled experiment was to establish which kinematic and motion parameters impact HRI with a mobile robot. We designed an assisted teleoperation tool to be able to drive the robot manually while still applying constraints on the style of the robot's motion in terms of accelerations, velocities, proximity to people. This would be used to approach people in a university campus building and engage them in a task that required co-navigation with the robot. This experiment was designed in such a way that it would rely on spontaneous participation, rather than requiring participants to book and plan in advance like the first experiment, given the constant uncertainties around a potential second lockdown. Unfortunately, before the experiment design was finalized, the second lockdown occurred (November 2020), prompting our switch to online experiments. Part of the teleoperation software initially designed for this experiment was used as a base for the assisted teleoperation version of the flyer experiment presented in this thesis.

Appendix D

GLMM models

This appendix contains the generalized mixed effects linear regression models resulting from the second online perception experiment presented in chapter 3.

Table D.1: GLMM logistic regression coefficients for the authoritative-polite scale.

Variable	**log(OR)**[1]	**95% CI**[1]	**p-value**
Intercept	0.08	-0.26, 0.41	>0.9
kinematics			
high	—	—	
low	2.1	1.9, 2.3	<0.001
medium	1.1	0.93, 1.3	<0.001
sequence			
A	—	—	
B	0.13	-0.12, 0.37	>0.9
C	-0.11	-0.35, 0.14	>0.9
D	-0.09	-0.33, 0.16	>0.9
E	-0.62	-0.86, -0.38	<0.001
F	-0.38	-0.62, -0.14	0.021
variant			
increment	—	—	
saccade	-0.60	-0.82, -0.38	<0.001
smooth	-0.13	-0.36, 0.09	>0.9
eyes			
none	—	—	
round	0.01	-0.21, 0.23	>0.9
squint	-0.62	-0.84, -0.40	<0.001
base			
stable	—	—	
unstable	0.20	-0.02, 0.42	0.7
head			
straight	—	—	
side	0.03	-0.19, 0.25	>0.9
turn_side	-0.05	-0.27, 0.17	>0.9
turn_straight	-0.05	-0.27, 0.17	>0.9
id.sd__(Intercept)	0.95		

[1]OR = Odds Ratio, CI = Confidence Interval

Table D.2: GLMM logistic regression coefficients for the confident-hesitant scale.

Variable	**log(OR)**[1]	**95% CI**[1]	**p-value**
intercept	-1.1	-1.4, -0.75	<0.001
kinematics			
high	—	—	
low	1.3	1.2, 1.5	<0.001
medium	0.79	0.62, 0.96	<0.001
sequence			
A	—	—	
B	0.32	0.09, 0.55	0.042
C	0.92	0.68, 1.2	<0.001
D	1.3	1.1, 1.6	<0.001
E	-1.2	-1.4, -0.94	<0.001
F	-0.77	-1.0, -0.53	<0.001
variant			
increment	—	—	
saccade	0.99	0.77, 1.2	<0.001
smooth	-0.86	-1.1, -0.64	<0.001
eyes			
none	—	—	
round	-0.23	-0.44, -0.01	0.12
squint	-0.27	-0.49, -0.06	0.055
base			
stable	—	—	
unstable	0.93	0.71, 1.2	<0.001
head			
straight	—	—	
side	-0.12	-0.34, 0.10	0.3
turn_side	-0.18	-0.40, 0.04	0.2
turn_straight	-0.30	-0.52, -0.08	0.042
id.sd__(Intercept)	0.79		

[1]OR = Odds Ratio, CI = Confidence Interval

Table D.3: GLMM logistic regression coefficients for the inspires-doesn't inspire confidence scale.

Variable	**log(OR)**[1]	**95% CI**[1]	**p-value**
Intercept	0.31	0.00, 0.63	0.3
kinematics			
high	—	—	
low	-0.59	-0.75, -0.43	<0.001
medium	-0.22	-0.39, -0.06	0.048
sequence			
A	—	—	
B	0.01	-0.21, 0.24	>0.9
C	0.42	0.19, 0.65	0.004
D	0.61	0.38, 0.84	<0.001
E	-0.34	-0.56, -0.11	0.029
F	-0.39	-0.62, -0.17	0.006
variant			
increment	—	—	
saccade	1.1	0.86, 1.3	<0.001
smooth	-0.46	-0.66, -0.26	<0.001
eyes			
none	—	—	
round	-0.39	-0.59, -0.19	0.002
squint	0.03	-0.17, 0.23	>0.9
base			
stable	—	—	
unstable	0.53	0.32, 0.73	<0.001
head			
straight	—	—	
side	-0.15	-0.36, 0.06	0.8
turn_side	-0.14	-0.35, 0.07	0.8
turn_straight	-0.10	-0.31, 0.11	>0.9
id.sd__(Intercept)	0.91		

[1]OR = Odds Ratio, CI = Confidence Interval

Table D.4: GLMM logistic regression coefficients for the nice-disagreeable scale.

Variable	**log(OR)**[1]	**95% CI**[1]	**p-value**
Intercept	0.50	0.11, 0.90	0.2
kinematics			
high	—	—	
low	-1.2	-1.4, -1.1	<0.001
medium	-0.56	-0.72, -0.39	<0.001
sequence			
A	—	—	
B	-0.24	-0.48, 0.00	0.6
C	-0.04	-0.27, 0.20	>0.9
D	-0.01	-0.25, 0.23	>0.9
E	0.07	-0.17, 0.31	>0.9
F	-0.21	-0.45, 0.03	>0.9
variant			
increment	—	—	
saccade	0.51	0.30, 0.73	<0.001
smooth	-0.13	-0.35, 0.08	>0.9
eyes			
none	—	—	
round	-0.62	-0.84, -0.40	<0.001
squint	0.18	-0.03, 0.40	>0.9
base			
stable	—	—	
unstable	0.14	-0.08, 0.36	>0.9
head			
straight	—	—	
side	-0.08	-0.30, 0.14	>0.9
turn_side	-0.11	-0.33, 0.11	>0.9
turn_straight	-0.14	-0.35, 0.08	>0.9
id.sd__(Intercept)	1.5		

[1]OR = Odds Ratio, CI = Confidence Interval

Table D.5: GLMM logistic regression coefficients for the sturdy-frail scale.

Variable	**log(OR)**[1]	**95% CI**[1]	**p-value**
Intercept	-1.3	-1.6, -0.90	<0.001
kinematics			
high	—	—	
low	1.1	0.97, 1.3	<0.001
medium	0.77	0.59, 0.94	<0.001
sequence			
A	—	—	
B	0.46	0.22, 0.71	0.001
C	0.51	0.26, 0.75	<0.001
D	0.97	0.72, 1.2	<0.001
E	-0.82	-1.1, -0.56	<0.001
F	-0.46	-0.71, -0.21	0.002
variant			
increment	—	—	
saccade	1.3	1.1, 1.6	<0.001
smooth	-0.90	-1.1, -0.68	<0.001
eyes			
none	—	—	
round	-0.08	-0.31, 0.14	0.5
squint	-0.25	-0.47, -0.02	0.10
base			
stable	—	—	
unstable	1.3	1.1, 1.5	<0.001
head			
straight	—	—	
side	-0.26	-0.48, -0.03	0.10
turn_side	-0.14	-0.36, 0.09	0.4
turn_straight	-0.41	-0.63, -0.18	0.002
id.sd__(Intercept)	1.2		

[1]OR = Odds Ratio, CI = Confidence Interval

Table D.6: GLMM logistic regression coefficients for the strong-weak scale.

Variable	**log(OR)**[1]	**95% CI**[1]	**p-value**
Intercept	-1.0	-1.4, -0.65	<0.001
kinematics			
high	—	—	
low	1.6	1.4, 1.8	<0.001
medium	0.94	0.76, 1.1	<0.001
sequence			
A	—	—	
B	0.41	0.17, 0.65	0.006
C	0.46	0.22, 0.71	0.001
D	0.91	0.66, 1.2	<0.001
E	-0.66	-0.91, -0.41	<0.001
F	-0.38	-0.63, -0.14	0.010
variant			
increment	—	—	
saccade	0.98	0.75, 1.2	<0.001
smooth	-0.67	-0.89, -0.45	<0.001
eyes			
none	—	—	
round	-0.11	-0.33, 0.10	0.6
squint	-0.41	-0.63, -0.18	0.002
base			
stable	—	—	
unstable	0.88	0.66, 1.1	<0.001
head			
straight	—	—	
side	-0.23	-0.46, -0.01	0.12
turn_side	-0.07	-0.29, 0.15	0.6
turn_straight	-0.28	-0.50, -0.06	0.057
id.sd__(Intercept)	1.3		

[1]OR = Odds Ratio, CI = Confidence Interval

Table D.7: GLMM logistic regression coefficients for the smooth-abrupt scale.

Variable	**log(OR)**[1]	**95% CI**[1]	**p-value**
Intercept	-0.71	-1.1, -0.29	0.010
kinematics			
high	—	—	
low	-1.3	-1.5, -1.1	<0.001
medium	-0.48	-0.65, -0.31	<0.001
sequence			
A	—	—	
B	0.21	-0.04, 0.45	0.6
C	0.39	0.14, 0.63	0.017
D	0.53	0.29, 0.78	<0.001
E	0.13	-0.11, 0.38	>0.9
F	-0.04	-0.29, 0.21	>0.9
variant			
increment	—	—	
saccade	0.84	0.62, 1.1	<0.001
smooth	-0.32	-0.54, -0.10	0.037
eyes			
none	—	—	
round	-0.37	-0.60, -0.14	0.015
squint	0.21	-0.01, 0.44	0.4
base			
stable	—	—	
unstable	0.40	0.17, 0.62	0.007
head			
straight	—	—	
side	-0.03	-0.25, 0.19	>0.9
turn_side	-0.05	-0.28, 0.17	>0.9
turn_straight	-0.08	-0.30, 0.14	>0.9
id.sd__(Intercept)	1.6		

[1]OR = Odds Ratio, CI = Confidence Interval

Table D.8: GLMM logistic regression coefficients for the rigid-supple scale.

Variable	**log(OR)**[1]	**95% CI**[1]	**p-value**
Intercept	-1.3	-1.7, -0.95	<0.001
kinematics			
high	—	—	
low	1.1	0.91, 1.3	<0.001
medium	0.44	0.26, 0.61	<0.001
sequence			
A	—	—	
B	0.04	-0.20, 0.28	>0.9
C	-0.04	-0.28, 0.20	>0.9
D	-0.38	-0.62, -0.13	0.030
E	-0.22	-0.46, 0.02	0.7
F	-0.05	-0.29, 0.19	>0.9
variant			
increment	—	—	
saccade	-0.71	-0.94, -0.48	<0.001
smooth	0.15	-0.07, 0.37	>0.9
eyes			
none	—	—	
round	0.27	0.06, 0.49	0.13
squint	-0.07	-0.29, 0.14	>0.9
base			
stable	—	—	
unstable	0.00	-0.22, 0.22	>0.9
head			
straight	—	—	
side	0.03	-0.18, 0.25	>0.9
turn_side	-0.02	-0.24, 0.20	>0.9
turn_straight	0.06	-0.16, 0.28	>0.9
id.sd__(Intercept)	1.3		

[1]OR = Odds Ratio, CI = Confidence Interval

Table D.9: GLMM logistic regression coefficients for the tender-insensitive scale.

Variable	**log(OR)**[1]	**95% CI**[1]	**p-value**
Intercept	1.5	1.1, 2.0	<0.001
kinematics			
high	—	—	
low	-1.2	-1.4, -1.0	<0.001
medium	-0.66	-0.84, -0.47	<0.001
sequence			
A	—	—	
B	-0.22	-0.47, 0.03	0.8
C	0.12	-0.13, 0.38	>0.9
D	0.13	-0.13, 0.38	>0.9
E	0.33	0.08, 0.58	0.12
F	-0.03	-0.28, 0.22	>0.9
variant			
increment	—	—	
saccade	0.47	0.24, 0.71	0.001
smooth	0.04	-0.19, 0.27	>0.9
eyes			
none	—	—	
round	-0.91	-1.1, -0.68	<0.001
squint	0.07	-0.16, 0.30	>0.9
base			
stable	—	—	
unstable	-0.01	-0.24, 0.23	>0.9
head			
straight	—	—	
side	-0.17	-0.40, 0.07	>0.9
turn_side	-0.15	-0.38, 0.08	>0.9
turn_straight	-0.04	-0.27, 0.20	>0.9
id.sd__(Intercept)	1.8		

[1]OR = Odds Ratio, CI = Confidence Interval