



HAL
open science

Planning Socially Expressive Mobile Robot Trajectories

Philip Scales, Olivier Aycard, Véronique Aubergé

► **To cite this version:**

Philip Scales, Olivier Aycard, Véronique Aubergé. Planning Socially Expressive Mobile Robot Trajectories. *Sensors*, 2024, Special Issue Applications of Intelligent Robots: Sensing, Interaction, Navigation and Control Systems), 24 (11), pp.3533. 10.3390/s24113533 . hal-04616294

HAL Id: hal-04616294

<https://hal.science/hal-04616294>

Submitted on 18 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Article

Planning Socially Expressive Mobile Robot Trajectories

Philip Scales ^{1,2,*} , Olivier Aycard ¹ and Véronique Aubergé ²¹ GIPSA-Laboratory, University Grenoble Alpes, CNRS, Grenoble INP, 38000 Grenoble, France² LIG-Laboratory, University Grenoble Alpes, CNRS, Grenoble INP, 38000 Grenoble, France

* Correspondence: philip.scales@univ-grenoble-alpes.fr

Abstract: Many mobile robotics applications require robots to navigate around humans who may interpret the robot's motion in terms of social attitudes and intentions. It is essential to understand which aspects of the robot's motion are related to such perceptions so that we may design appropriate navigation algorithms. Current works in social navigation tend to strive towards a single ideal style of motion defined with respect to concepts such as comfort, naturalness, or legibility. These algorithms cannot be configured to alter trajectory features to control the social interpretations made by humans. In this work, we firstly present logistic regression models based on perception experiments linking human perceptions to a corpus of linear velocity profiles, establishing that various trajectory features impact human social perception of the robot. Secondly, we formulate a trajectory planning problem in the form of a constrained optimization, using novel constraints that can be selectively applied to shape the trajectory such that it generates the desired social perception. We demonstrate the ability of the proposed algorithm to accurately change each of the features of the generated trajectories based on the selected constraints, enabling subtle variations in the robot's motion to be consistently applied. By controlling the trajectories to induce different social perceptions, we provide a tool to better tailor the robot's actions to its role and deployment context to enhance acceptability.

Keywords: trajectory planning; human–robot interaction; perception experiment; logistic regression



Citation: Scales, P.; Aycard, O.; Aubergé, V. Planning Socially Expressive Mobile Robot Trajectories. *Sensors* **2024**, *24*, 3533. <https://doi.org/10.3390/s24113533>

Academic Editors: Wataru Sato, Daniel Galan, Ramon A. Suarez Fernandez and Francisco Javier Badesa

Received: 26 March 2024

Revised: 23 May 2024

Accepted: 27 May 2024

Published: 30 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

There is an increasing number of application domains for mobile robots operating in environments alongside humans, both in public spaces (e.g., train stations and shops) as well as spaces such as hospitals [1], care-homes, or private homes [2]. In the early days of human–robot interaction research, it was quickly established that using traditional navigation algorithms that only consider humans as obstacles results in unacceptable robot behaviour. This led to the emergence of the field of Social Navigation (SN) [3], which aims to design human-aware navigation algorithms to improve acceptability. The deployment of mobile robots in human environments requires overcoming both technical challenges as well as human factors and social perception challenges. An important example of such perception challenges is the Uncanny Valley phenomenon [4], whereby the appearance of a robot greatly influences its likability and human affinity with it, which will inevitably impact robot adoption and integration into society. Another major dimension of the Uncanny Valley is the role of movement dynamics in changing human perception of the robot [5] as well as the role of the robot's attitude [6], although these dimensions have not received as much attention in studies. Therefore, it is necessary to determine whether a robot's movements may impact human social perception of the robot's attitudes in order to improve acceptability and interaction quality.

Recent works in robotics have been carried out on the topic of functional expressive motion generation [7], whereby subtle aspects of a robot's movement can be modified to express the robot's intentions or emotions while simultaneously executing a task. In [8], it was shown that a humanoid robot arm handing over an object with a rude or gentle attitude could influence the human interacting with the robot. While algorithms have

been proposed to generate legged robot movements expressing emotions such as happy or sad [9], navigation algorithms for mobile robots focus on other dimensions such as naturalness and comfort [10–13]. These dimensions are different from social attitudes such as aggressiveness, hesitancy, or politeness, whose impact on human interactions has been studied, particularly in the field of vocal prosody [14,15]. It is unknown which movement features may lead to different perceptions of social attitudes in mobile robots, and existing navigation algorithms are not able to adjust the robot's motion so that it generates different perceptions of social attitudes. This lack of understanding of the social perception of robot motion, and the inability to adjust navigation features, may be partly responsible for the low acceptance that can be observed when mobile robots are deployed in real environments [16,17].

In this paper, we explore two questions in order to improve acceptability and integration of robots in human environments. Firstly, which features of a mobile robot's motion may lead a person to interpret its motion in terms of social attitudes such as aggressive, gentle, authoritative, or polite? Secondly, how may we formalize these features and incorporate them into a novel navigation algorithm capable of altering the navigation style to adjust how the robot is perceived by humans? In order to answer these questions, we make the following contributions:

- We build a statistical model of human perception of different combinations of trajectory features capturing a robot's movement dynamics, bringing new knowledge on human social perception of robot motion.
- We formalize these trajectory features found to cause different social perceptions and design a novel optimization-based trajectory planning algorithm that can accurately reproduce the social motion features while performing a navigation task.

Our statistical analysis demonstrates that subtle motion features can strongly shape perception of the attitudes and physical attributes of mobile robots. Our algorithm enables control over the robot's expression of social attitudes through its motion, which was not possible using existing algorithms focused on comfort and naturalness or on the expression of emotions.

The structure of the paper is as follows: Section 2 reviews existing approaches for evaluating and designing social navigation algorithms. Section 3 summarizes the perception experiment from our prior work and presents our approach to model the relationship between the robot's subtle motion features and the participant's perception of social attitudes and physical characteristics. Section 4 describes our trajectory planning algorithm formulated as a constrained optimization problem using novel constraints and control input formulations, which ensure that the robot's motion always contains the features relevant for altering human perception of the robot. Section 5 presents the implementation and validation of our algorithm on a real mobile robot, demonstrating that the plans can be accurately reproduced by the robot in a way that maintains the key movement features. Finally, we draw conclusions and present ideas for future work in Section 6.

2. State of the Art

2.1. Design and Evaluation of Social Navigation Algorithms

Works in the field of Social Navigation [18] typically concern themselves with enabling mobile robots to navigate in complex environments [19], around many (potentially dynamic) pedestrians [20], and modeling uncertainty of surrounding pedestrian motion [21]. Social navigation approaches tend to focus on ensuring the robot plans safe, comfortable, and natural motion, following the definitions in [3]. Questionnaires such as the Godspeed Questionnaire Series [22], Negative Attitudes towards Robots Scale (NARS) [23], Perceived Social Intelligence scale (PSI) [24], or Robot Social Attributes Scale (RoSAS) [25] are often used to assess naturalness, comfort, and likability, aiming to maximize all of them with a single navigation style. Although these metrics are useful, there are still issues with the acceptability of mobile robots, particularly when humans attribute social intentions or attitudes to a mobile robot's navigation. Vocal prosody (pitch, rhythm, and tone) has

been widely studied as a means to convey social attitudes and relation information in interactions between both humans and robots [26,27]. In addition to studying vocal prosody, ref. [28] observed that the changes in a participant's vocal prosody over the course of an interaction with a robot were aligned with changes in their spatial behavior, gaze, and voice quality [29]. We hypothesize that a robot's navigation movement prosody (i.e., subtle differences in its style of navigation [30]) may play a role in how people perceive different social attitudes. This is different to existing works that study naturalness and comfort [10] or the expression of emotions or intent through motion [7].

A typical approach to design a social navigation algorithm is to first develop the algorithm, then subsequently evaluate the impact of the motions it generates. When designing the algorithm, the social aspects can either be implicitly incorporated through machine-learning methods that aim to replicate average human navigation [31,32] or explicitly by manually observing and modeling human behavior [33]. Another approach is to implement existing models of human behavior such as the Social Force Model [34]. Spatial and proximity factors are the most commonly addressed in earlier works [35], often being derived from the concept of proxemics [36]. The algorithm is then used to control a real or simulated robot in order to conduct experiments to evaluate the algorithm. Participants can be asked to compare different navigation algorithms [37,38] or the same algorithm with different parameters such as avoidance distance or speed [39]. Although these approaches enable researchers to evaluate the whole algorithm in terms of its efficiency, comfort, or naturalness, they do not allow a precise understanding of exactly which features of the generated motions are responsible for each aspect of the evaluation.

We propose to first assess which motion variables are important by using hand-crafted motions built through a systematic combination of different motion features across several motion variables. 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 [15,28,29,40]. 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 these will be used to guide the design of our social navigation algorithm.

2.2. Algorithmic Approaches for Social Navigation

The goal of social navigation and expressive navigation works is to generate motion that accomplishes a physical task while taking into account a variety of metrics that 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 that best match the social or expressive features. These objectives are often in conflict with each other. Formulating trajectory generation as an optimization problem is a common approach both within social robotics [41,42] and in other areas such as crowd behavior generation [43]. A typical solution when faced with conflicting objectives is to adopt a scalarization approach to treat the multi-objective problem as a single-objective problem, usually by computing a weighted average of cost terms [18]. This requires tuning the weights to obtain the desired behavior, balancing the different social and expressive objectives as well as the task objectives such as making progress towards a navigation goal.

In [44], the authors propose two costs, modeling 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, ref. [41] present an approach that jointly plans cooperative trajectories for a

single human and the robot, accounting for metrics such as the expected time to collide 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 [45], maintaining a desired position and velocity while accompanying a person [46], and avoiding intrusion into group formations and the information processing space in front of people [47]. Similarly, for expressive motion, ref. [48] 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, with the most common being personal space around people derived from proxemics [35]; 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 in [49], where the expressive features for a robot arm can only be expressed through degrees of freedom that have absolutely no effect on the practical task. On the contrary, in [50], 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. Machine-learning techniques are also popular in social navigation [31,32,51]; however, they require many demonstrations or large annotated datasets, which would have to be annotated with the corresponding social perceptions. Currently, such datasets with annotations of human social perception of the mobile robot do not exist and would be complex and time-consuming to create. Furthermore, learned navigation models lack the explainability and controllability given by optimization-based approaches. Hence, we adopt a trajectory optimization problem formulation. It is crucial that our algorithm generates motions that are very accurately matched to those we 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 [50], 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 as well as its ability to plan a future trajectory with appropriate movement prosody.

3. Model of Human Social Perception of Mobile Robot Motion

In this section, we discuss our method for modeling how different variations in a mobile robot's motion and appearance influence people's social and physical perception of the robot. In our prior work [30], we designed a robot motion corpus that consists of motion and appearance variables, which are combined to define many different styles of motion. The corpus was used to conduct perception experiments by asking participants to rate their perception of a mobile robot by viewing videos of it performing motions from the corpus. In the prior work, early experimental results using simple chi-square tests suggested that the variables included in our corpus had significant impacts on people's perception of the robot. In this paper, we present experimental results from a larger participant pool as well as the results of mixed effects logistic regression modeling to determine precisely how each corpus variable impacts the probability of different social perceptions of the robot. In the following subsections, we first give the definitions of each of the corpus variables, followed

by the ten perceptual scales we used to evaluate participant perceptions; lastly, we present the results of the regression analyses.

3.1. Robot Motion Corpus Background

In this subsection, we briefly present our motion corpus and describe which features of the robot are manipulated. Firstly, we present the motion variables that impact the robot's motion by defining its velocity, acceleration, and style of motion. Secondly, we present the appearance variables that also contribute to changing a person's perceptual experience when interacting with a mobile robot. The corpus is available online at the following address: <https://osf.io/5csrg/> (accessed on 25 March 2024), and an example video is available at: <https://youtu.be/EiH8o1PjIOW> (accessed on 25 March 2024).

3.1.1. Motion Variables

The goal of the motion corpus variables is to uniquely define a velocity profile for the robot's linear (forward translation) velocity. Each variable affects different features of the profile determining the robot's velocity over time in order to perform a point-to-point straight line motion. The most basic way to achieve such a motion would be to perform acceleration using the maximal motor acceleration followed by a similar deceleration, forming a triangular velocity profile. The first variable is the kinematics type, which defines both the maximum velocity as well as the slope of the profile, i.e., the acceleration value. This variable has three values (low, medium, and high), enabling comparisons between the different values listed in Table 1.

Table 1. Kinematics type parameters.

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

The next motion variable is the motion sequence. This variable affects the overall shape of the velocity profile by defining the ordering and succession of acceleration and deceleration phases, shown in Figure 1. The previous example of a single acceleration followed by a deceleration corresponds to motion sequence B. We introduce two features that impose variations of this sequence: pauses and hesitations. A pause motion sequence is defined as always introducing a short, constant velocity phase between an acceleration and deceleration phase, transforming the profile from a triangular shape to a trapezoidal shape, as in sequence A. A hesitation motion sequence is defined as introducing a "V" shape into the profile after an acceleration phase, meaning that the robot decelerates partially before accelerating back up to its previous velocity peak, as in sequence D. Pauses and hesitations are combined in sequence C. Sequences E and F are simply extensions of profile B, representing only the start or end of a robot's motion as it arrives or leaves.

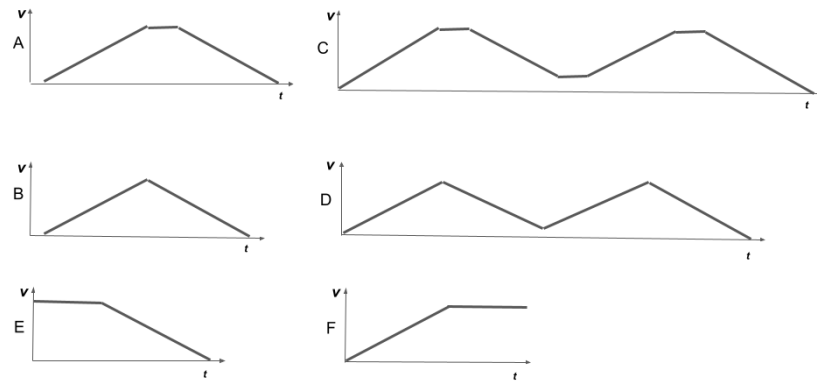


Figure 1. Illustrations of the effect of the six motion sequence values on the velocity profiles, shown in the six subfigures (A–F). The slope and maximum values of the profiles are determined by the kinematics variable. These profiles use the smooth variant.

The last motion variable is the variant. The previous illustrations represent profiles using the smooth variant, where each acceleration and deceleration is a linear segment. In contrast, we define two other variants that alter the overall style of the robot’s motion, shown in Figure 2. The saccade variant is defined as the velocity profile oscillating periodically to induce stuttering and shaking into the robot’s motion. The increment variant is defined as dividing the acceleration and deceleration phases into three separate phases interleaved by short constant velocity phases, creating a stepping or incremental motion.

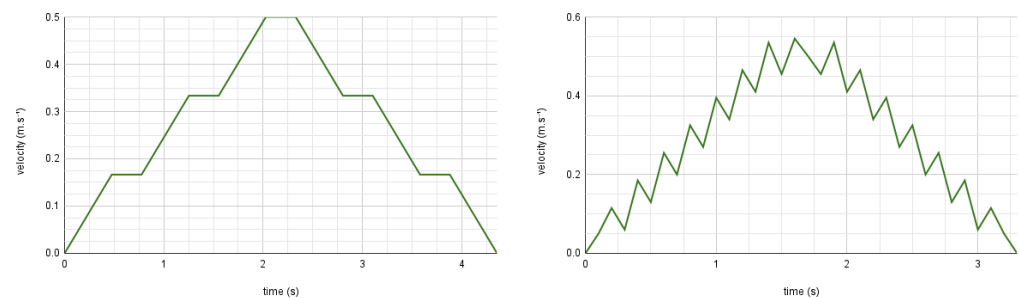


Figure 2. Velocity profiles resulting from combining the increment (left) or saccade (right) variants with motion sequence A and medium kinematics.

3.1.2. Appearance Variables

In addition to altering the robot’s velocity profile, we use several variables to alter the visual appearance of the robot. The first variable is the robot base type, which is either stable or unstable. When the base is unstable, the robot’s front and back balancing wheels are loosened making the robot tilt backwards or forwards when accelerating or decelerating. The robot’s head is also loosened, making it shake and move when using the shaking motion induced by the saccade variant.

The second appearance variable is the head motion. The robot’s head can be fixed in different orientations: either straight ahead or 90° to the side. Two other settings involve the head rotating from the straight to the side orientation or from the side to the straight orientation while the robot performs its motion. In the videos presented to the participants, the side orientation directs the robot’s gaze towards the viewer, whereas the straight orientation directs the gaze towards the direction of travel of the robot—to the right-hand side of the video frame.

The third appearance variable is the eye shape. The LED eye displays on the robot’s head are set to three different display settings. The first setting corresponds to white round eyes, which represent a neutral appearance. The second setting corresponds to green squinting eyes that represent a colder, unsettling appearance. In general, a robot’s

appearance has been found to impact interaction in prior studies [52,53]; hence, we include these variables to be able to distinguish the effect of the appearance from the effect of the motion variables.

3.2. Perceptual Scales

In Table 2, we present the ten semantic differential scales used 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, and 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, and 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 [26,27]. For more details about the choice of the adjectives in our scales, we refer the reader to [30].

Table 2. 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

3.3. Logistic Regression Modeling

3.3.1. Method

During the experiment, each participant rated 45 different videos each showing a unique combination of the corpus variable values along the 10 binary perceptual scales. A total of $n = 100$ participants of various ages ($M = 33.61$, $SD = 14.76$) and genders (56 female, 36 male, 6 other, and 2 without response) were recruited through social media, local university and lab experiment mailing lists, as well as fliers handed out in public spaces. There was no selection criteria other than fluently speaking the language used to express the adjectives (in this study, we used French). Participants were instructed to perform the experiment on a device with a large screen and smooth video playback to ensure good perception of the robot motions. Each of the 450 videos was rated 10 times; hence, each participant performs ten binary classifications for each video, where the input is the set of values for each of the corpus motion variables and the output is a binary choice between the two adjectives of each scale. We chose to fit a mixed effects logistic regression model for each scale [54], allowing us to account for dependencies within the data since participants each provide 45 responses. Each of the corpus variables is treated as a categorical fixed effect, and the participant *id* is used as a random effect. The models were implemented in R using the *lme4* package [55]. The logistic models give the probability of the participant choosing the second (rightmost) adjectives of the scales in Table 2. In R syntax, the model structure can be given as follows:

$$scale \sim kinematics + sequence + variant + eyes + base + head + (1|id) \quad (1)$$

We did not include interaction terms in the final model, since pairwise interactions resulted in a worse model fit on the test data, as measured by the Area Under Curve (AUC) of the Receiver Operator Characteristic (ROC) curve [56,57] (average 0.798 with interaction terms, 0.805 without interactions).

The log odds scale used for the logistic regression coefficients is practical for fitting the models and computing predictions; however, it is not a very intuitive scale to interpret the results. An alternative way to understand the results of mixed effect logistic regression is to compute the estimated marginal means (EMM) based on the model predictions [58]. 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. 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 [59].

3.3.2. Results

Table 3 presents the marginal effects of each value of each corpus variable on the perceptual scales. The 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 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 p.p. 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 [60] was applied to the p -values to adjust for multiple comparisons. The head rotation variable is not represented in the table, given that the only significant effects were on the sturdy–frail scale, with contrasts 5 p.p. and –5 p.p. for straight and turn straight values (both $p < 0.05$), respectively.

The contrast values range from –28 p.p. (high kinematics effect on aggressive–gentle) to 27 p.p. (sequence D effect on confident–hesitant), with all values in between, including some null contrasts (sequence C effect on aggressive–gentle). All of the motion corpus variables have statistically significant effects on how participants perceived the robot. Many of the corpus variable values alter the average probability of differing perceptions by 10, 20, or even almost 30 percentage points. In addition to gaining an understanding of the direction and magnitude of the influence of the corpus variables on human perception, the mixed effect logistic regression models can be used to perform predictions of combinations of the corpus variable values that may be perceived.

The kinematics and variant variables both have consistently large effects on every perceptual scale, mostly greater than 10 p.p., and greater than 20 p.p. for two to three scales. Kinematics mostly affects the aggressive–gentle and authoritative–polite scales, while variants mostly affect the confident–hesitant and sturdy–frail scales. These are followed by the motion sequence variable with effects greater than 10 p.p. on the confident–hesitant, inspires–doesn't inspire confidence, and sturdy–frail scales. The base stability and eyes have effects of more than 10 p.p. on two scales. The head variable has little to no effect on any of the scales. These perception experiment results show that all of the corpus variables related to the robot's linear velocity profiles have strong effects on how the robot is perceived while navigating. These results can be used to derive how to combine the robot motion features in order to generate a desired impression. An example is shown in Figure 3, where the hesitant motion is obtained by combining low kinematics with the saccade variant and motion sequence D; whereas confident motion is obtained with high kinematics, smooth variant, and motion sequence B.

Table 3. Marginal effects of the corpus variables on the perceptual scales in percentage points (p.p.).
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

	– +	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

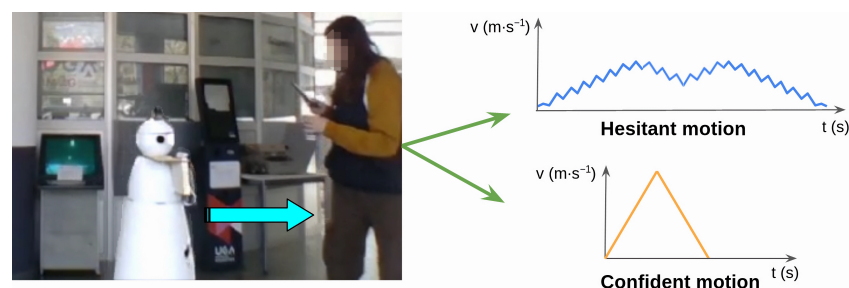


Figure 3. Examples of linear trajectories to approach a person using different velocity, acceleration, and timing features resulting in confident or hesitant perception.

4. Algorithm for Trajectory Planning with Configurable Movement Styles

In the previous section, we used logistic regression models to determine how different features of the linear velocity of a mobile robot can impact people's perception of the robot. The results of the statistical analysis demonstrated which motion features were important, namely, the accelerations, velocities, and inclusion of hesitations and pauses in the move-

ment. In this section, we present our approach to design a trajectory planning algorithm that can be configured in order to enforce these motion features in the generated trajectories. We propose to derive constraints that enable control over the features of the robot's linear velocity profile to match our model of movement prosody. In this paper, we assume that the environment is static and that there are no obstacles between the robot and the goal in order to focus on accurate reproduction of the movement characteristics from our perception experiment. Firstly, we formalize the velocity profile representation. Secondly, we generalize the profiles to enable trajectories to span different distances. Thirdly, we discuss how a typical trajectory optimization that only constrains motion based on the motor limits cannot allow us to configure the robot's motion according to our desired movement prosody. Fourthly, we derive a novel set of constraints to enable the trajectory generation to be configured based on the desired prosody. Lastly, we present the algorithm to perform the offline trajectory planning and open loop control to realize the planned motion.

4.1. Velocity Profile Representation

In this section, we first introduce the notation that will be used to describe the velocity profiles and then explain how the robot's motion is controlled when executing the fixed distance velocity profiles from the corpus.

The corpus profiles were constructed by selecting the combination of values for three variables: the motion sequence, kinematics type, and variant. Figure 4 represents how each of these variables alters the shape of the corpus velocity profile.

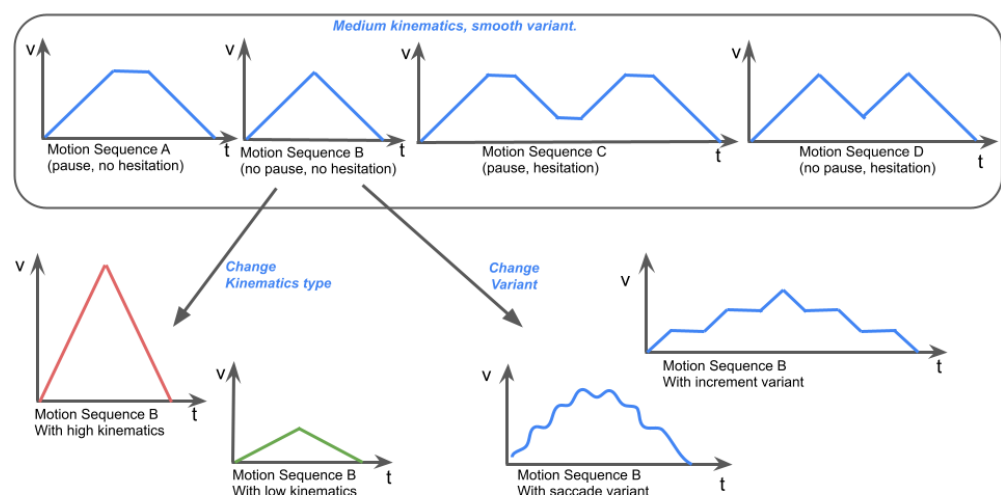


Figure 4. Illustration of the construction of the velocity profiles by combining the motion corpus variables. **Top:** all motion sequences represented with medium kinematics and smooth variant. **Bottom:** profiles resulting from applying different kinematics or variants to motion sequence B. In total,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

In Figure 5, we give an example of how a corpus velocity profile can be represented as a sequence $U = \{u_0, u_1 \dots u_{N-1}\}$ 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 space and time, and are related through the forward kinematics Equation (2). This equation is

simplified with respect to the full differential drive forward kinematics, since we do not control the angular velocity of the robot.

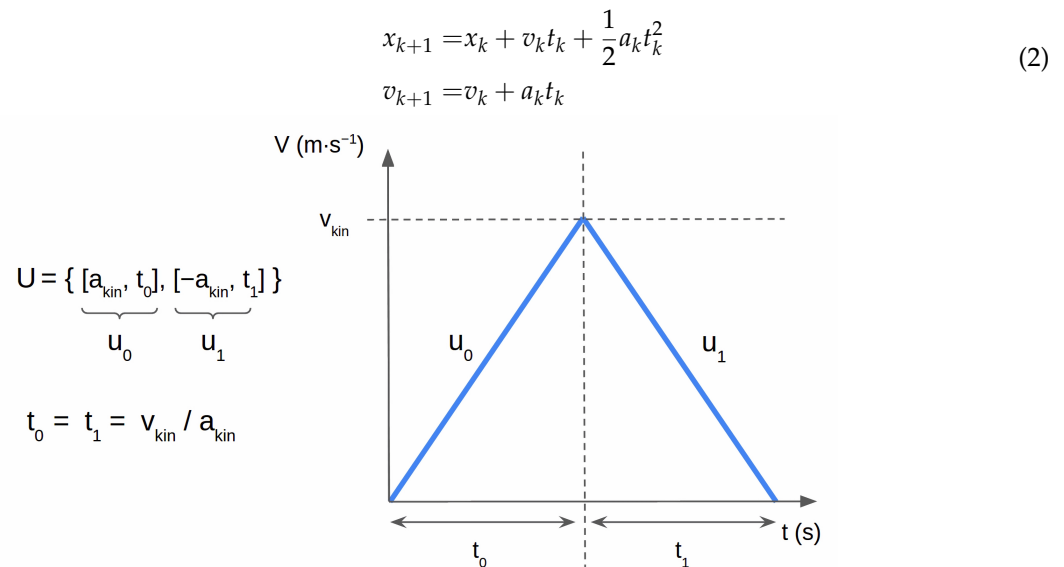


Figure 5. Representation of a corpus velocity profile using motion sequence B (no pauses, no hesitations) and the smooth variant as a sequence U of $N = 2$ motion phases u_0 and u_1 . Values of v_{kin} and a_{kin} depend on the selected kinematics type (medium, low, and high), and dictate the slope and maximum of the velocity profile.

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 5, using medium kinematics ($a_{kin} = a_{medium} = 0.35 \text{ m}\cdot\text{s}^{-2}$, $v_{kin} = v_{medium} = 0.5 \text{ m}\cdot\text{s}^{-1}$), results in $t_0 = t_1 = \frac{v_{kin}}{a_{kin}} = 1.428 \text{ s}$, meaning the profile makes the robot cover a distance equal to $a_{kin} \times t_0^2 = 0.714 \text{ m}$. In order to change the distance traveled, we need to introduce some degrees of freedom back into the velocity profile. In the following section, we discuss how we add flexibility while maintaining the distinct characteristics of each corpus variable as much as possible.

4.2. Generalization of Corpus Profiles to Variable Distances

Executing a given velocity profile results in the robot performing a unique trajectory in space and time with a given length. To design a planning algorithm, we require a formulation where the distance is a free variable. In other words, we want to transform the corpus profiles corresponding to a combination of parameters into a class of profiles that maintains as many characteristics of the original profiles as possible. 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 traveled 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. Duration of maximum velocity phase (motion sequence).

Since the kinematics type and variants had high impacts on people's perceptions of the robot, we choose the third solution of changing the duration of the maximum velocity to adapt the velocity profiles to variable distances. This variable is only 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 (300 ms), constant velocity phases. In order to

lengthen profiles, we introduce a constant velocity phase at the maximal velocity. To shorten the profiles, we reduce the maximum velocity while maintaining the profile shape (motion sequences) and the slopes (accelerations of the kinematics type). An example of the transformations applied to alter the distance travelled can be seen in Figure 6. These profiles are achieved by changing the motion phases $u_k = [a_k, t_k]$ by altering the values of the accelerations a_k as well as the durations t_k such that generating a trajectory to cover a given distance amounts to searching for their optimal values. In the following section, we formalize this trajectory generation process as a constrained optimization problem.

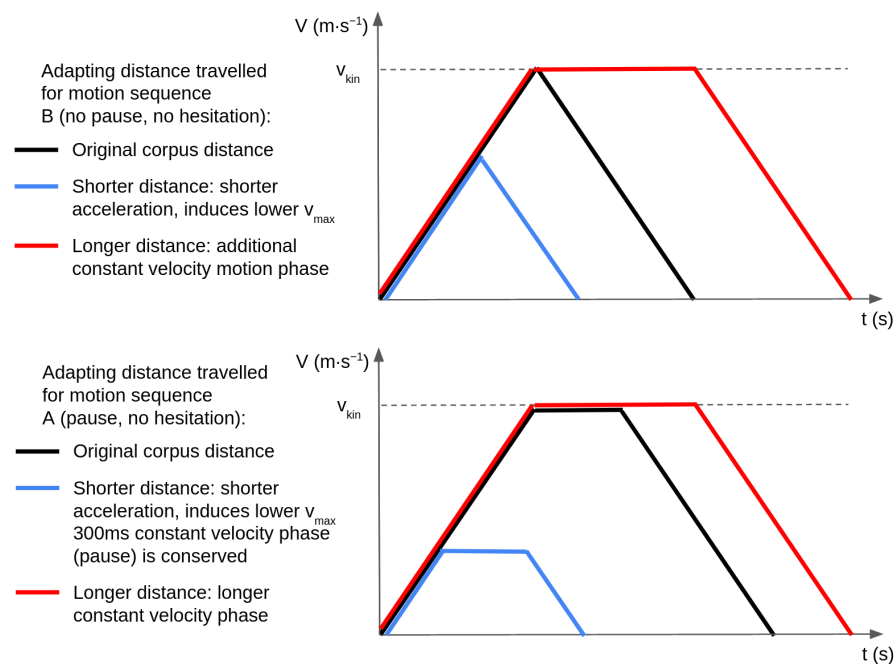


Figure 6. Illustration of the transformation of a corpus velocity profile to travel shorter or longer distances. **Top:** transformation for profiles without pauses or hesitations (sequence B). **Bottom:** transformation for profiles with pauses and without hesitations (sequence A).

4.3. Problem Formulation

We first formalize a typical optimization problem that does not take into account our corpus variables. 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 $U = \{u_0, u_1 \dots u_{N-1}\}$ 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 comprises the robot's position along the x axis and its linear velocity v , since we focus only on linear motion in this paper. The control u_k affects the state \mathbf{x}_k , as described in the kinematics Equation (2).

The trajectory optimization is performed over a finite time T_h [61], 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. 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 $\sum_{k=0}^{N-1} t_k = T_h$. The resulting classical trajectory planning problem formulation is given in Equation (3).

$$\begin{aligned}
& \min_{u_0 \dots u_{N-1}} \sum_{k=0}^{N-1} \|x_g\|^2 \\
& \text{subject to: } \forall k \in \{0, 1 \dots N-1\}, 0 \leq v_k \leq v_{max} \\
& \quad \forall k \in \{0, 1 \dots N-1\}, -a_{max} \leq a_k \leq a_{max} \\
& \quad \sum_{k=0}^{N-1} t_k = T_h
\end{aligned} \tag{3}$$

Solving this optimization problem would produce triangular or trapezoidal velocity profiles depending on the distance to be traveled. 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 designing novel constraints that extend this optimization problem to restrict the set of valid control sequences based on the values of our prosody motion corpus variables.

4.4. Prosody Constraint Formalization

In this section, we propose novel constraints that model each of the motion corpus prosody parameters such that a trajectory whose motion phases satisfy the constraint is representative of the corresponding motion corpus parameter value.

4.4.1. 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 simply truncated versions of sequence A. 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 $a_{k-1} \neq 0$ is followed by a constant velocity phase $a_k = 0$ with a duration t_k greater or equal to the pause length $t_{pause} = 300$ ms (see Figure 7). This constraint is expressed in Equation (4), in such a way that it describes what should not occur: if the previous phase is not a constant velocity phase, 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.

$$\begin{aligned}
\text{PauseConstraint}(U) \leftrightarrow \forall k \in [1, N-1], \\
\neg(a_{k-1} \neq 0 \wedge a_k \neq a_{k-1} \wedge \neg(a_k = 0 \wedge t_k \geq t_{pause}))
\end{aligned} \tag{4}$$

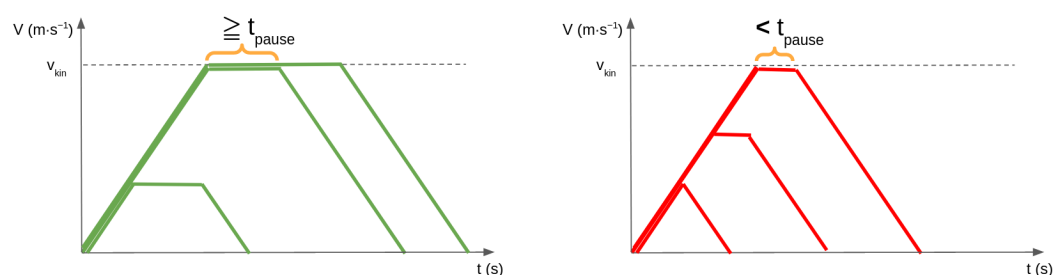


Figure 7. Illustration of the pause constraint. **Left:** valid trajectories. **Right:** invalid trajectories due to insufficient length of the constant velocity phase.

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 deceleration should occur immediately after the end of an acceleration phase and then at regular time intervals $t_{h_interval}$. Hesitation deceleration phases are enforced by the constraint in Equation (5). The variable t_{since_hesit} is introduced to ensure that a hesitation deceleration is added after $t_{h_interval}$. Hesitation accelerations are enforced by the constraint in Equation (6). The variable $type_k \in \{normal, hesitation\}$ is introduced so that a hesitation acceleration is not enforced after a normal deceleration. These constraints are combined to enforce the hesitation motion sequence (Equation (7)).

$$\begin{aligned} HesitationDeceleration(U) \leftrightarrow \forall k \in [1, N - 1], \\ (a_{k-1} = a_{kin} \vee t_{since_hesit} \geq t_{h_interval}) \\ \wedge \neg(a_k = -a_{kin} \wedge t_k = t_h) \end{aligned} \quad (5)$$

$$\begin{aligned} HesitationAcceleration(U) \leftrightarrow \forall k \in [1, N - 1], \\ (a_{k-1} = -a_{kin} \wedge type_{k-1} = hesitation) \\ \wedge \neg(a_k = a_{kin} \wedge t_k = t_h) \end{aligned} \quad (6)$$

$$\begin{aligned} HesitationConstraint(U) \leftrightarrow \neg HesitationDeceleration(U) \\ \wedge \neg HesitationAcceleration(U) \end{aligned} \quad (7)$$

4.4.2. 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 $t_{smooth} = 300$ ms (Equation (8)). 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.

$$SmoothConstraint(U) \leftrightarrow \forall k \in [0, N - 1], t_k \geq t_{smooth} \quad (8)$$

The increment variant requires acceleration phases to be split into increments such that the robot performs a constant velocity phase of duration $t_{pause} = 300$ ms when reaching certain velocities, which are multiples of $v_{increment} = \frac{1}{3} stoppingTime(v_{kin}, a_{kin})$. The first part of the constraint (Equation (9)) 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 (10)), 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 (11).

$$\begin{aligned} ValidVelocity(U) \leftrightarrow \forall k \in [0, N - 1], \\ v_k = i \times v_{increment}, i \in \mathbb{N} \end{aligned} \quad (9)$$

$$\begin{aligned} BreakAccelerationPhase(U) \leftrightarrow \forall k \in [1, N - 1], \\ a_{k-1} \neq 0 \wedge ((a_k = 0 \wedge t_k = t_{pause}) \vee a_k = -a_{k-1}) \end{aligned} \quad (10)$$

$$\begin{aligned} IncrementConstraint(U) \leftrightarrow ValidVelocity(U) \\ \wedge BreakAccelerationPhase(U) \end{aligned} \quad (11)$$

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 our motion corpus, saccades correspond to oscillations of the velocity over time, with a high frequency and low amplitude, since the aim of this variant is to reproduce stuttering or shaking. The saccade variant can, therefore, be accomplished by adding a time-varying offset given by a triangular wave to the profile given by planning under the smooth variant constraint, without rendering the trajectory invalid. We use a period $\pi = 0.02$ s and an amplitude dependent on the kinematics type: $A_{low} = 0.02 \text{ m}\cdot\text{s}^{-2}$, $A_{medium} = 0.05 \text{ m}\cdot\text{s}^{-2}$, and $A_{high} = 0.07 \text{ m}\cdot\text{s}^{-2}$.

4.4.3. 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, which we enforce with a constraint on the space of control inputs u_k of the robot. The acceleration values a_k are constrained to the finite set $\{-a_{kin}, 0, a_{kin}\}$. The value of a_{kin} is determined by the kinematics type (high, medium, or low).

$$\begin{aligned} \text{KinematicsAcceleration}(U) \leftrightarrow \forall k \in [0, N - 1], \\ a_k \in \{-a_{kin}, 0, a_{kin}\} \end{aligned} \quad (12)$$

The kinematics type specifies a maximum velocity that the robot should not exceed, which we enforce 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 constraint in Equation (13) forces constant velocity phases to only be planned at the maximum velocity. The velocity and acceleration constraints are combined to form the overall kinematics constraint expressed in Equation (13).

$$\begin{aligned} \text{KinematicsVelocity}(U) \leftrightarrow \forall k \in [0, N - 1] \\ 0 \leq v_k \leq v_{kin} \wedge \neg(a_k = 0 \wedge v_k \notin \{v_{kin}, 0\}) \end{aligned} \quad (13)$$

$$\begin{aligned} \text{KinematicsConstraint}(U) \leftrightarrow \text{KinematicsAcceleration}(U) \\ \wedge \text{KinematicsVelocity}(U) \end{aligned} \quad (14)$$

4.5. Trajectory Planning and Open-Loop Control

Our set of proposed prosody constraints $PC_{offline}$ is summarized in Table 4. These constraints are integrated into our new problem formulation given in Equation (15), which retains the same control variables and cost function as the previous formulation. Each constraint enforces trajectory properties that 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 PC_{active} of the constraints from $PC_{offline}$. 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.

$$\begin{aligned} \min_{u_0 \dots u_{N-1}} \quad & \sum_{k=0}^{N-1} \|x_g\|^2 \\ \text{subject to:} \quad & \left\{ \begin{array}{l} PC_{active} \subset PC_{offline}, \\ \sum_{k=0}^{N-1} t_k = T_h. \end{array} \right. \end{aligned} \quad (15)$$

Table 4. Constraints forming the set $PC_{offline}$ used for offline planning.

Constraint	Equation
Pause	(4)
Hesitation	(7)
Smooth	(8)
Increment	(11)
Kinematics	(14)

4.5.1. Trajectory Planning

In order to solve the optimization problem (15), given that both optimization variables a_k and t_k are discretized, we use a tree-based approach to search the space of trajectories with a fixed length of N motion phases. 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 k th 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 $\mathcal{T} = \{dt, 2dt, \dots, t_{max}\}$ 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 horizon duration T_h (Equation (16)). The discretization level for the accelerations is already enforced by the kinematics constraint. For the durations, we choose a discretization $dt = 100$ ms, which is short enough to enable all prosody constraints to be enforced accurately (such as the 300 ms pause constraint) and also long enough to maintain a low number of possible trajectories.

$$t_{max} = T_h - \sum_{i=0}^k t_i - (N - k)dt \quad (16)$$

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 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 (15), 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 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 result is a tree of depth N , where each leaf node represents the last state of a fully prosody-compliant trajectory. The *EvaluateTrajectories* function exhaustively evaluates the trajectories according to the cost function from problem (15). The minimum-cost sequence of control inputs U^* is then used as the input to the open-loop control algorithm described in the next paragraph, which executes the control inputs with appropriate timing.

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

Notations:
 $\mathbf{x}_k = [x_k, v_k]$, k th robot state. $u_k = [a_k, t_k]$, k th motion phase. T trajectory tree.

Algorithm:

```

1  $T \leftarrow CreateTree(\mathbf{x}_{init})$ 
2  $\mathbf{x}_k \leftarrow \mathbf{x}_{init}$ 
3 while  $\neg TraversalFinished(T)$  do
4   if  $T.Depth(\mathbf{x}_k) < N$  then
5     for  $u_k \in \mathcal{A} \times \mathcal{T}$  do
6        $\mathbf{x}_{k+1} \leftarrow ForwardSimulation(\mathbf{x}_k, u_k)$  (Equation (2))
7        $is\_valid \leftarrow CheckConstraints(PC_{offline}, u_{k-1}, u_k, \mathbf{x}_k, \mathbf{x}_{k+1})$ 
8       if  $is\_valid$  then
9          $T.AddChild(\mathbf{x}_k, u_k, \mathbf{x}_{k+1})$ 
10     $\mathbf{x}_k, u_{k-1} \leftarrow T.DepthFirstNextNode$ 
11  $U^* \leftarrow EvaluateTrajectories(x_g, T)$ 
12 return  $U^*$ 

```

4.5.2. Open-Loop Control

Algorithm 2 describes the overall process to execute a prosody compliant trajectory in an open loop fashion. It uses 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 (15), obtaining the optimal trajectory U^* . We then simply iterate over the controls $\{u_0, u_1 \dots u_{N-1}\}$, 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.

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 .
 $U^* = \{u_0, u_1 \dots, u_{N-1}\}$, sequence of motion phases describing the trajectory.

Algorithm:

```

1  $U^* \leftarrow PlanTrajectory(\mathbf{x}_0, x_g, PC_{offline})$  (Algorithm 1)
2 for  $k \in [0, N - 1]$  do
3    $a_t, t_t \leftarrow ExtractControl(U^*, k)$ 
4    $SendMotorCommand(a_t)$ 
5    $DelayUntil(t + t_t)$ 

```

5. Implementation and Validation

5.1. Implementation

Firstly, we present the RobAIR wheeled mobile robot platform shown in Figure 8. The RobAIR platform [62] 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. The robot is 1.20 m high, and has a diameter of 0.50 m at its widest point—at the base. The robot has a differential drive configuration, can reach a maximum

velocity of $0.8 \text{ m}\cdot\text{s}^{-1}$, and accelerates at $2.667 \text{ m}\cdot\text{s}^{-2}$. Two Hokuyo URG-04-LX-UG01 laser range-finders are mounted on the robot's head and at the base in order to detect obstacles and track people while navigating.



Figure 8. Left: RobAIR mobile robot. Right: RobAIR base.

Certain parameters of our algorithm must be selected based on the types of movement prosody the robot should produce. We employed $N = 10$ motion phases in order to be able to plan the most complex motions such as those using the increment variant. The time discretization was $dt = 100 \text{ ms}$ in order to maintain a fine temporal resolution so that constraints such as the 300 ms pause may be accurately enforced. The short time discretization also allows for more accurate position tracking. When deploying our algorithm on a given robot, the velocity and acceleration constraint values should be selected such that they are within the specifications of the robot motor hardware to ensure that the generated trajectories are feasible.

Algorithm 2 is implemented as an ROS node in C++. One planning cycle takes 50 ms on average and no more than 100 ms on a single core of a low-power tablet PC (Intel $i5 - 8365U$). Figure 9 shows the overall architecture. The goal point x_g to be reached is given by a LIDAR-based perception module, allowing the robot to be driven to a person detected by a multiple hypothesis tracker based on clustering of the laser data. The planning node then uses Algorithm 1 to plan the trajectory to x_g . The planned acceleration commands are converted to sequences of linear velocity commands, given that our motors do not allow acceleration-based control. The planner node sends these velocity commands at 10 Hz to the hardware interface node, ensuring accurate timing. The linear velocity commands are finally translated into wheel velocities and sent to the motors. The set of constraints PC_{active} used by the planner can be altered by the means of a prosody parameter selection node, which implements a simple ROS Dynamic Reconfigure interface to save and load parameter presets to represent the different movement prosody styles.

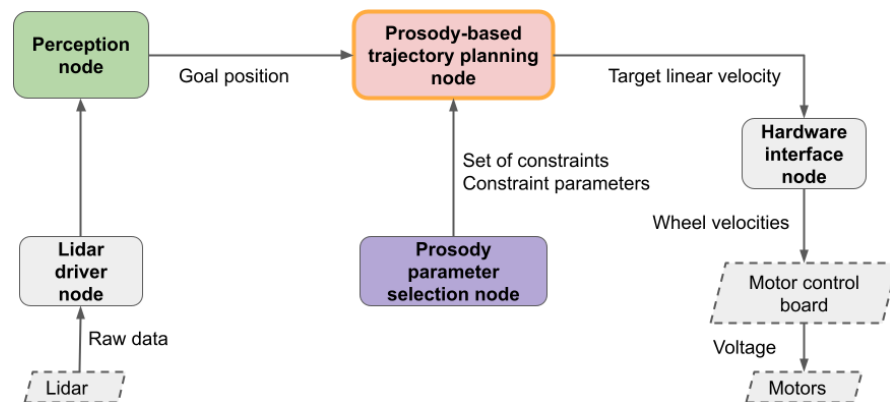


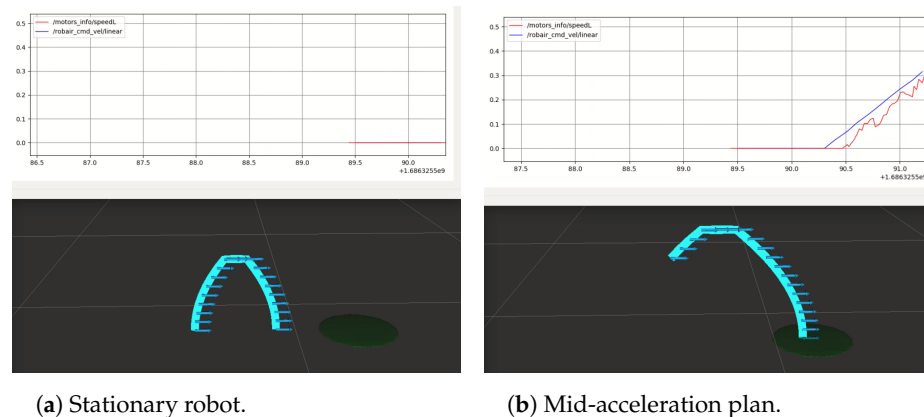
Figure 9. High-level architecture of our system. ROS nodes are represented with rounded boxes; hardware devices are represented with dashed boxes.

5.2. Validation

In this section, we demonstrate the ability of our planning algorithm to produce trajectories that accurately reproduce the different types of movement prosody defined by the combination of corpus variables while ensuring the robot reaches its goal. 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, and no hesitations.

5.2.1. Point-to-Point Trajectory Execution

Firstly, we demonstrate the ability of our proposed trajectory planning and control algorithms to successfully drive the robot towards a goal position. Figure 10 shows the execution of a plan consisting of an acceleration, constant velocity, and deceleration that have been optimized to reach the goal while satisfying all movement prosody constraints on the acceleration, velocity, and timing of the motion. In this case, the constraints involved are the acceleration, maximum velocity, as well as the pause constraint, requiring the robot to perform a 300 ms constant velocity phase before decelerating. In the remainder of this section, we focus on demonstrating the accurate reproduction of the movement prosody features in the planned velocity profiles.



(a) Stationary robot.

(b) Mid-acceleration plan.

Figure 10. Cont.

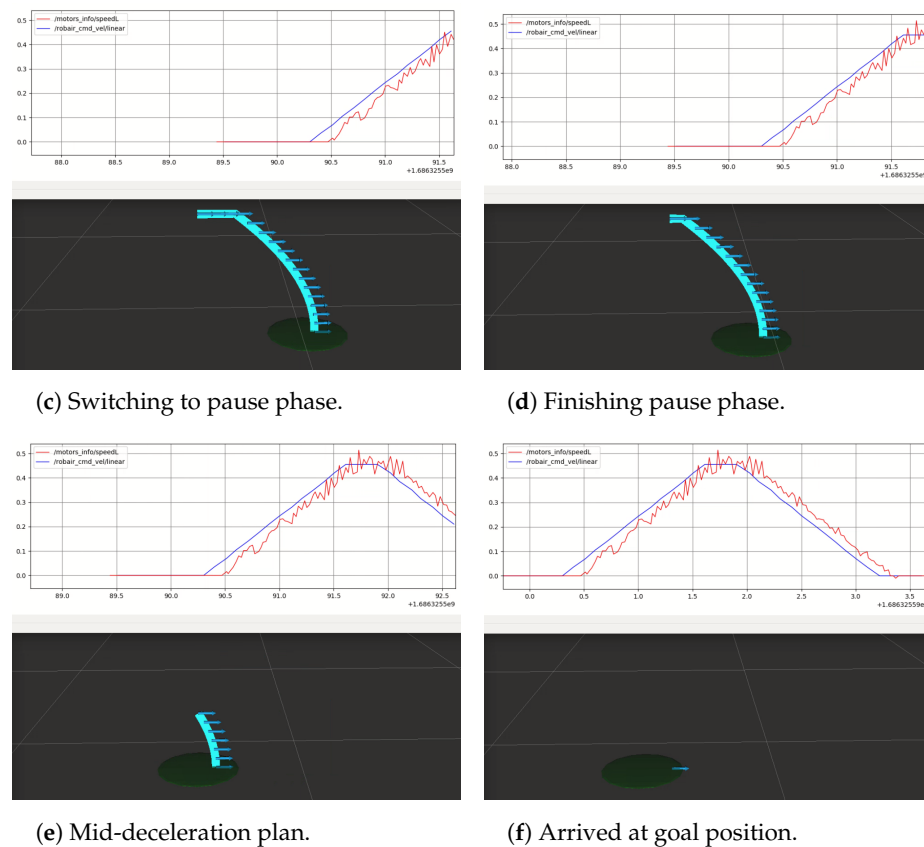


Figure 10. **Top:** past command velocities issued at 10 Hz (blue) and encoder-based odometry estimated at 40 Hz (red) in $\text{m}\cdot\text{s}^{-1}$, plotted with respect to time (s). **Bottom:** visualization of the planned trajectory's velocity, discretized into time intervals of length $dt = 100$ ms. The robot stops within 10 cm of its goal position (green).

5.2.2. Kinematics

The three kinematics types (low, medium, and high) require different accelerations, and different maximal velocities. We show examples of motions produced by running our planner with each of the kinematics types. We plot the raw odometry 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. The unfiltered odometry is noisy due to the cheap encoder sensors, whereas the true motion of the robot is smooth. The plots of the unfiltered commands allow us to see that the response time of the motors is very fast, allowing the robot to accurately track even the most subtle and fast changes in the velocity commands.

Figure 11 shows a short motion with the low kinematics. The goal point is close enough that the robot only accelerates to $0.20 \text{ m}\cdot\text{s}^{-1}$, slightly below the low kinematics maximum of $0.24 \text{ m}\cdot\text{s}^{-1}$. The slope of the commanded velocity profile corresponds to the low kinematics acceleration of $0.2 \text{ m}\cdot\text{s}^{-2}$ as expected, and the estimated velocity also follows the commanded velocity closely.

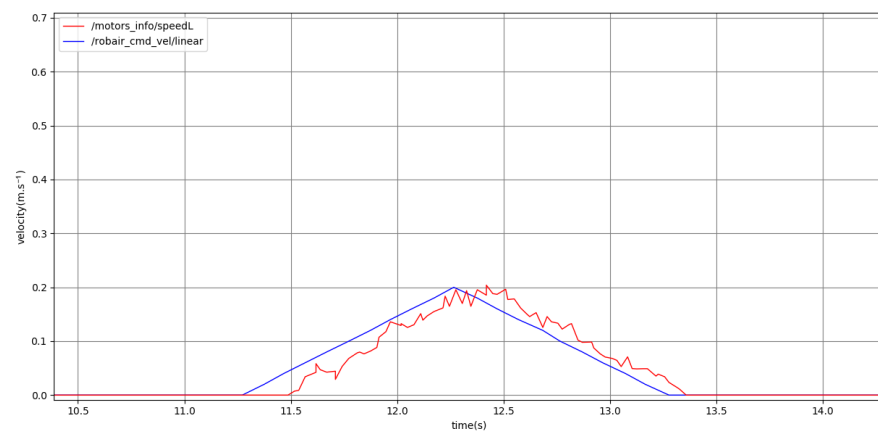


Figure 11. Plot representing the full point-to-point motion to a goal point, using low kinematics. Past command velocities shown in blue and unfiltered odometry shown in red, both given in $\text{m}\cdot\text{s}^{-1}$.

Figure 12 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 $0.72 \text{ m}\cdot\text{s}^{-1}$. The robot accelerates to $0.65 \text{ m}\cdot\text{s}^{-1}$, with an acceleration of $0.5 \text{ m}\cdot\text{s}^{-2}$, clearly distinguishing the motion from the low kinematics setting. The profiles shown in the following subsections all use the medium kinematics setting, which is also distinct from the low and high settings.

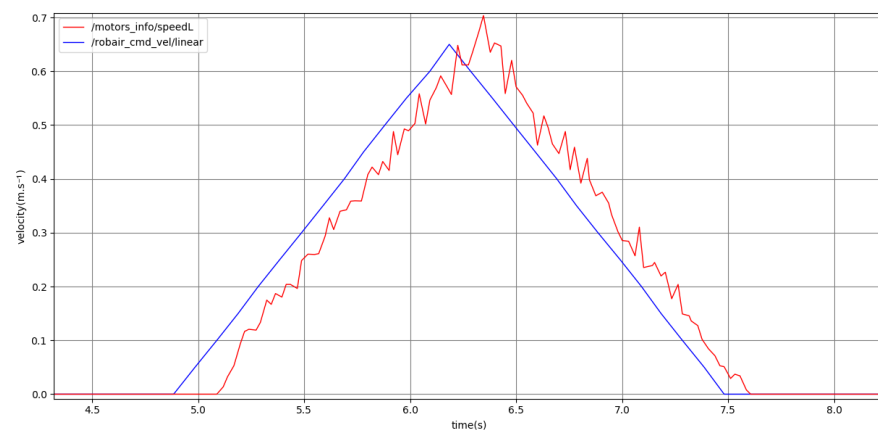


Figure 12. Plot representing the full point-to-point motion to a goal point using high kinematics. Past command velocities shown in blue and unfiltered odometry shown in red, both given in $\text{m}\cdot\text{s}^{-1}$.

5.2.3. Pause and Hesitation Sequences

Figure 13 shows the plot of the robot's velocity, and distance during a point-to-point motion to a goal placed 62 cm 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 300 ms—and drives the robot towards the goal point (video with visualization of the plan execution using Rviz (shown at $0.2\times$ speed for clarity): <https://cloud.univ-grenoble-alpes.fr/s/f5G8kQR4rMx6Mwi> (accessed on 25 March 2024)).

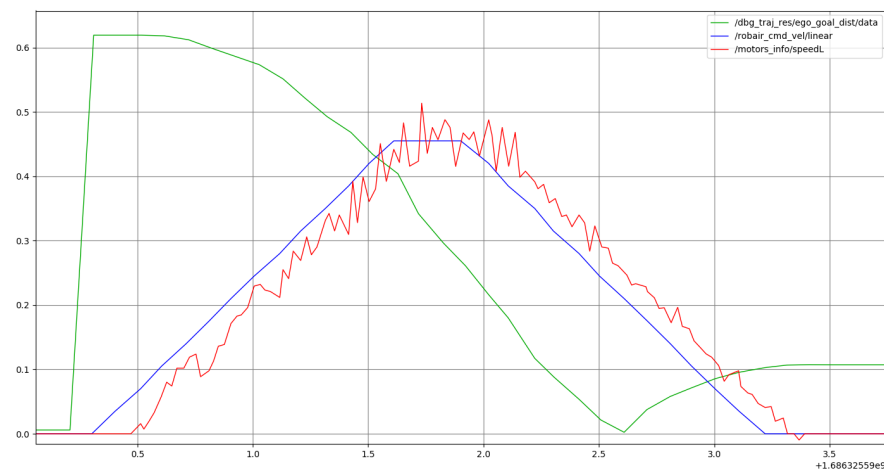


Figure 13. Plot representing the full point-to-point motion to a goal point. Past command velocities shown in blue and unfiltered odometry shown in red, both given in $\text{m}\cdot\text{s}^{-1}$. Distance to the goal in m shown in green.

Figure 14 shows the plot of the robot's velocity when using the hesitation constraint. Once again, the generated plan for the robot's velocity enforces the hesitation feature by including the succession of a deceleration and acceleration upon reaching the robot's maximum velocity. The robot is able to accurately reproduce motions including the hesitation feature.

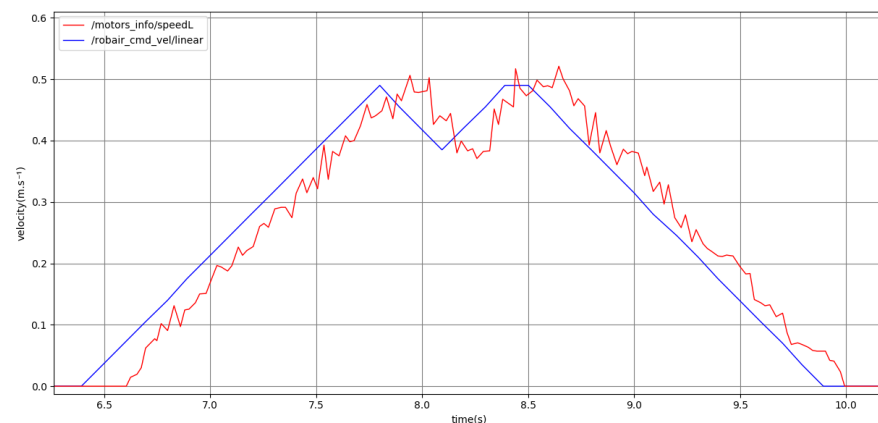


Figure 14. Point-to-point motion using the hesitation sequence (without pauses) and medium kinematics. Past command velocities shown in blue and unfiltered odometry shown in red, both given in $\text{m}\cdot\text{s}^{-1}$.

5.2.4. Increment and Saccade Variants

In this subsection, we demonstrate motions planned under the increment or saccade variant constraints. Figure 15 shows a long increment motion, allowing the robot to reach the maximum velocity for the medium kinematics type. The planner correctly inserts constant velocity phases at regular intervals, which are tracked by the robot motors, reproducing the stepped acceleration pattern.

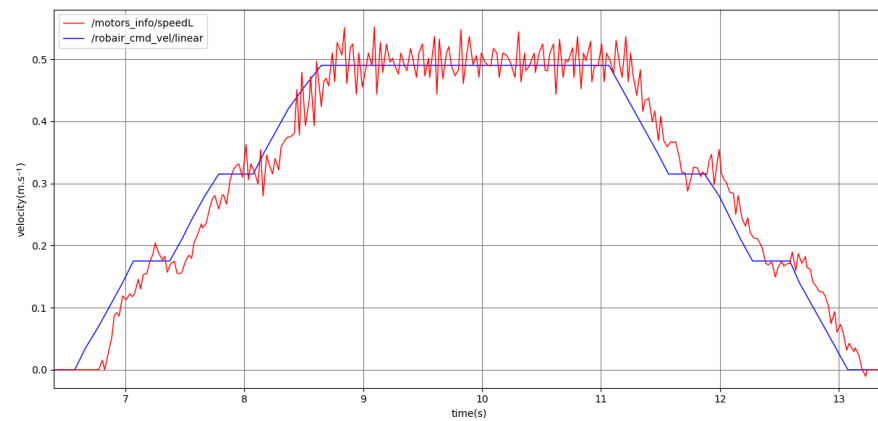


Figure 15. Point-to-point motion using the increment variant and medium kinematics. Past command velocities shown in blue and unfiltered odometry shown in red, both given in $\text{m}\cdot\text{s}^{-1}$.

Figure 16 shows a short saccade motion without pauses. The planner reproduces the expected oscillation in the velocity commands, and the robot is able to accurately track the rapid changes in the requested velocity, enabling the robot to perform the stuttering, saccadic movements as desired.

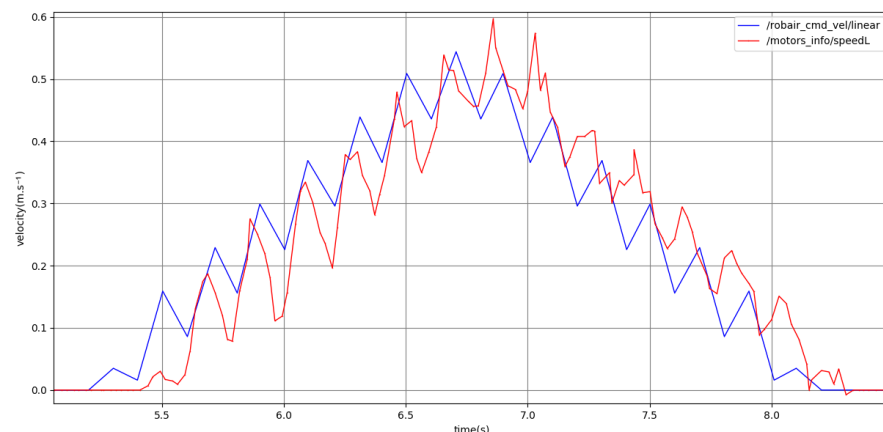


Figure 16. Point-to-point motion using the saccade variant and medium kinematics. Past command velocities shown in blue and unfiltered odometry shown in red, both given in $\text{m}\cdot\text{s}^{-1}$.

6. Discussion

6.1. Generalization of the Human Perception Model

Our model of human perception was derived from the analysis of experimental data from our online study and in-person studies presented in our prior work [30]. The results show that accelerations, velocities, and timing have significant impacts on the social perception of our mobile robot. However, prior studies have shown that the size of a robot [63], its shape and color [64], as well as its human-like or machine-like appearance [65] may also impact interaction. While our study did include three appearance variables (head orientation, eye shape, and base stability), further studies are necessary to explore the generalization of our model to different robot types.

Our experiments were designed in such a way that the robot was not shown in any specific scenario or social environment, since prior research has shown that a robot's behavior may be perceived differently and lead to different acceptance outcomes in different social settings, such as two different hospital services in [16]. Further work is also necessary to study how the robot's task, its social role, and its social environment may impact and alter human's social perceptions of the robot.

6.2. Limitations of the Trajectory Planning Algorithm

The main limitation of our algorithm is that it assumes static and known environments. In many real-world use cases, the environment will be dynamic and the robot's perception of pedestrians or other dynamic obstacles will be uncertain. One approach is to use our algorithm as a global planner and, subsequently, attempt to follow the global plan and avoid obstacles when necessary; however, such an approach would not guarantee that the robot's movement style is accurately maintained by the obstacle avoidance algorithm. Another approach to adapt trajectory planning algorithms to deal with changing and uncertain environments is to perform frequent re-planning [61]. However, given the subtle and time-dependent nature of the motion features we aim to reproduce, this would lead to inconsistencies in the robot's motion without careful consideration of the re-planning mechanism in the algorithm design, hence changing the human's social perception of the robot. Extending our algorithm to dynamic environments while maintaining accurate control over the robot's motion features is therefore non-trivial and requires further research.

6.3. Ethical Considerations

The statistical analysis of our perception experiment showed that the mobile robot's motion features could significantly impact a human's social perception of the robot. Subsequently, we proposed an approach to formalize the relevant motion features and integrate them into an optimization-based trajectory planner, taking a first step towards controlling the motion features responsible for altering social perceptions. On the one hand, these contributions may be used to analyze existing navigation algorithms to understand their impact on people and potentially avoid generating inappropriate social attitudes. On the other hand, altering the social perceptions of humans must be conducted with care and while considering the goal of such manipulations. For example, generating an impression of frailty can lead to a person being more engaged and active in an interaction, which may be useful in assistive or care use-cases; however, this may also induce attachment effects, which are not well understood [26,66]. Determining when and how to alter the generated social attitude in a given deployment scenario should be determined with the input of domain experts and end-users in addition to HRI researchers.

7. Conclusions and Future Work

In this paper, we studied how changes in a mobile robot's motion features alter human social perception of the robot, in order to better integrate robots into human environments. The statistical analysis of a perception experiment with $n = 100$ participants showed that motion features such as the robot's acceleration, velocity, and saccades have statistically significant impacts on human perception of social attitudes in mobile robots. Each of these features altered the probability of perceiving the robot as aggressive or gentle, authoritative or polite, or sturdy or frail by up to 30 percentage points. These results demonstrate that even subtle motion features have strong impacts on social perception, and therefore on the acceptance and integration of robots in human environments. Subsequently, we proposed a trajectory planning algorithm that can be configured to integrate these motion features into the trajectory while performing a point-to-point navigation task. We formulated the problem as a constrained optimization and derived a novel set of constraints to enforce the motion features that impact human social perception of the robot. The algorithm was implemented and validated on a real mobile robot, demonstrating that the trajectories produced by our planner accurately reproduce the features used in our perception experiment. Our algorithm enables a mobile robot's motion to be adjusted according to the desired social perception of the robot by humans, which was previously not possible using existing social navigation algorithms. Providing explicit control over how the robot is perceived ensures that the robot's actions are appropriate with respect to its role and the people it is interacting with.

In future work, we aim to extend our algorithm to handle dynamic uncertain environments by introducing temporal coherence constraints to enable accurate re-planning.

We also plan to deploy our algorithm in a realistic task to evaluate the impact of the different trajectory styles on humans when the interaction is situated in a social environment (preliminary video of a participant interacting with our robot using the proposed algorithm, configured to convey a confident attitude: <https://cloud.univ-grenoble-alpes.fr/s/GdnDKQbKd9GEgnG> (accessed on 25 March 2024)). Further experiments should also be conducted in different social environments and scenarios, as well as with different robot types, to determine the extent to which the model of human perception generalizes.

Author Contributions: Conceptualization, P.S., O.A. and V.A.; methodology, P.S., V.A. and O.A.; software, P.S.; validation, P.S., O.A. and V.A.; formal analysis, P.S., O.A. and V.A.; investigation, P.S., O.A. and V.A.; resources, O.A.; data curation, P.S.; writing—original draft preparation, P.S.; writing—review and editing, P.S., O.A. and V.A.; visualization, P.S.; supervision, O.A. and V.A.; project administration, O.A. and V.A.; funding acquisition, O.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kivrak, H.; Uluer, P.; Kose, H.; Gumuslu, E.; Erol Barkana, D.; Cakmak, F.; Yavuz, S. Physiological Data-Based Evaluation of a Social Robot Navigation System. In Proceedings of the 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), Naples, Italy, 31 August 2020–4 September 2020; pp. 994–999. [\[CrossRef\]](#)
2. Coşar, S.; Fernandez-Carmona, M.; Agrigoroaie, R.; Pages, J.; Ferland, F.; Zhao, F.; Yue, S.; Bellotto, N.; Tapus, A. ENRICHME: Perception and Interaction of an Assistive Robot for the Elderly at Home. *Int. J. Soc. Robot.* **2020**, *12*, 779–805. [\[CrossRef\]](#)
3. Kruse, T.; Pandey, A.K.; Alami, R.; Kirsch, A. Human-Aware Robot Navigation: A Survey. *Robot. Auton. Syst.* **2013**, *61*, 1726–1743. [\[CrossRef\]](#)
4. Mori, M. The uncanny valley. *Energy* **1970**, *7*, 33.
5. Destephe, M.; Brandao, M.; Kishi, T.; Zecca, M.; Hashimoto, K.; Takanishi, A. Walking in the uncanny valley: Importance of the attractiveness on the acceptance of a robot as a working partner. *Front. Psychol.* **2015**, *6*, 204. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Zlotowski, J.A.; Sumioka, H.; Nishio, S.; Glas, D.F.; Bartneck, C.; Ishiguro, H. Persistence of the uncanny valley: The influence of repeated interactions and a robot's attitude on its perception. *Front. Psychol.* **2015**, *6*, 883. [\[CrossRef\]](#)
7. Venture, G.; Kulić, D. Robot Expressive Motions. *ACM Trans. Hum.-Robot Interact.* **2019**, *8*, 1–17. [\[CrossRef\]](#)
8. Vannucci, F.; Di Cesare, G.; Rea, F.; Sandini, G.; Sciutti, A. A Robot with Style: Can Robotic Attitudes Influence Human Actions? In Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Beijing, China, 6–9 November 2018; pp. 952–957. [\[CrossRef\]](#)
9. Sripathy, A.; Bobu, A.; Li, Z.; Sreenath, K.; Brown, D.S.; Dragan, A.D. Teaching Robots to Span the Space of Functional Expressive Motion. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 13406–13413. [\[CrossRef\]](#)
10. Francis, A.; Pérez-D'Arpino, C.; Li, C.; Xia, F.; Alahi, A.; Alami, R.; Bera, A.; Biswas, A.; Biswas, J.; Chandra, R.; et al. Principles and Guidelines for Evaluating Social Robot Navigation Algorithms. *arXiv* **2023**, arXiv:2306.16740.
11. Gil, Ó.; Garrell, A.; Sanfeliu, A. Social robot navigation tasks: Combining machine learning techniques and social force model. *Sensors* **2021**, *21*, 7087. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Carton, D.; Olszowy, W.; Wollherr, D.; Buss, M. Socio-Contextual Constraints for Human Approach with a Mobile Robot. *Int. J. Soc. Robot.* **2017**, *9*, 309–327. [\[CrossRef\]](#)
13. Kamezaki, M.; Kobayashi, A.; Yokoyama, Y.; Yanagawa, H.; Shrestha, M.; Sugano, S. A Preliminary Study of Interactive Navigation Framework with Situation-Adaptive Multimodal Inducement: Pass-By Scenario. *Int. J. Soc. Robot.* **2019**, *12*, 567–588. [\[CrossRef\]](#)
14. Shochi, T. Prosodie des Affects Socioculturels en Japonais, et Anglais: À la Recherche des Vrais et Faux-Amis pour le Parcours de l'Apprenant. Ph.D. Thesis, Université Stendhal-Grenoble III, Grenoble, France, 2008.
15. Gobl, C.; Ní Chasaide, A. The role of voice quality in communicating emotion, mood and attitude. *Speech Commun.* **2003**, *40*, 189–212. [\[CrossRef\]](#)
16. Hebesberger, D.; Koertner, T.; Gisinger, C.; Pripfl, J. A Long-Term Autonomous Robot at a Care Hospital: A Mixed Methods Study on Social Acceptance and Experiences of Staff and Older Adults. *Int. J. Soc. Robot.* **2017**, *9*, 417–429. [\[CrossRef\]](#)

17. Mutlu, B.; Forlizzi, J. Robots in organizations. In Proceedings of the 3rd International Conference on Human Robot Interaction-HRI '08, Amsterdam, The Netherlands, 12–15 March 2008; p. 287. [\[CrossRef\]](#)
18. Mavrogiannis, C.; Baldini, F.; Wang, A.; Zhao, D.; Trautman, P.; Steinfeld, A.; Oh, J. Core Challenges of Social Robot Navigation: A Survey. *J. Hum.-Robot Interact.* **2023**, *12*, 1–39. [\[CrossRef\]](#)
19. Vega, A.; Manso, L.J.; Macharet, D.G.; Bustos, P.; Núñez, P. Socially aware robot navigation system in human-populated and interactive environments based on an adaptive spatial density function and space affordances. *Pattern Recognit. Lett.* **2019**, *118*, 72–84. [\[CrossRef\]](#)
20. Henderson, M.; Ngo, T.D. RRT-SMP: Socially-encoded Motion Primitives for Sampling-based Path Planning. In Proceedings of the 2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN), Vancouver, BC, Canada, 8–12 August 2021; pp. 330–336. [\[CrossRef\]](#)
21. Kollmitz, M.; Hsiao, K.; Gaa, J.; Burgard, W. Time dependent planning on a layered social cost map for human-aware robot navigation. In Proceedings of the 2015 European Conference on Mobile Robots, Lincoln, UK, 2–4 September 2015. [\[CrossRef\]](#)
22. Bartneck, C.; Kulić, D.; Croft, E.; Zoghbi, S. Measurement Instruments for the Anthropomorphism, Animacy, Likeability, Perceived Intelligence, and Perceived Safety of Robots. *Int. J. Soc. Robot.* **2009**, *1*, 71–81. [\[CrossRef\]](#)
23. Nomura, T.; Suzuki, T.; Kanda, T.; Kato, K. Measurement of negative attitudes toward robots. *Interact. Stud.* **2006**, *7*, 437–454. [\[CrossRef\]](#)
24. Barchard, K.A.; Lapping-Carr, L.; Westfall, R.S.; Fink-Armold, A.; Banisetty, S.B.; Feil-Seifer, D. Measuring the perceived social intelligence of robots. *ACM Trans. Hum.-Robot Interact.* **2020**, *9*. [\[CrossRef\]](#)
25. Carpinella, C.M.; Wyman, A.B.; Perez, M.A.; Stroessner, S.J. The Robotic Social Attributes Scale (RoSAS): Development and Validation. In Proceedings of the 2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Vienna, Austria, 6–9 March 2017; pp. 254–262.
26. Sasa, Y.; Aubergé, V. Perceived isolation and elderly boundaries in EEE (EmOz Elder-ly Expressions) corpus: Appeal to communication dynamics with a socio-affectively gluing robot in a smart home. *Gerontechnology* **2016**, *15*, 162.
27. Guillaume, L.; Aubergé, V.; Magnani, R.; Aman, F.; Cottier, C.; Sasa, Y.; Wolf, C.; Nebout, F.; Neverova, N.; Bonnefond, N.; et al. HRI in an ecological dynamic experiment: The GEE corpus based approach for the Emox robot. In Proceedings of the 2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO), Lyon, France, 30 June 2015–2 July 2015; pp. 1–6. [\[CrossRef\]](#)
28. Sasa, Y.; Aubergé, V. SASI: Perspectives for a socio-affectively intelligent HRI dialog system. In Proceedings of the 1st Workshop on “Behavior, Emotion and Representation: Building Blocks of Interaction”, Bielefeld, Germany, 17 October 2017.
29. Tsvetanova, L.; Aubergé, V.; Sasa, Y. Multimodal breathiness in interaction : From breathy voice quality to global breathy “body behavior quality”. In Proceedings of the Proc. of the 1st International Workshop on Vocal Interactivity in-and-between Humans, Animals and Robots—VIHAR 2017, Stockholm, Sweden, 25–26 August 2017.
30. Scales, P.; Aubergé, V.; Aycard, O. From vocal prosody to movement prosody, from HRI to understanding humans. *Interact. Stud.* **2023**, *24*, 131–168. [\[CrossRef\]](#)
31. Ramirez, O.A.; Khambhaita, H.; Chatila, R.; Chetouani, M.; Alami, R. Robots learning how and where to approach people. In Proceedings of the 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), New York, NY, USA, 26–31 August 2016; Volume 1, pp. 347–353. [\[CrossRef\]](#)
32. Chen, Y.F.; Everett, M.; Liu, M.; How, J.P. Socially aware motion planning with deep reinforcement learning. *IEEE Int. Conf. Intell. Robot. Syst.* **2017**, *2017*, 1343–1350. [\[CrossRef\]](#)
33. Kitagawa, R.; Liu, Y.; Kanda, T. Human-inspired Motion Planning for Omni-directional Social Robots. In Proceedings of the 2021 16th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Online, 9–11 March 2021; pp. 34–42.
34. Shiomi, M.; Zanlungo, F.; Hayashi, K.; Kanda, T. Towards a Socially Acceptable Collision Avoidance for a Mobile Robot Navigating Among Pedestrians Using a Pedestrian Model. *Int. J. Soc. Robot.* **2014**, *6*, 443–455. [\[CrossRef\]](#)
35. Rios-Martinez, J.; Spalanzani, A.; Laugier, C. From Proxemics Theory to Socially-Aware Navigation: A Survey. *Int. J. Soc. Robot.* **2015**, *7*, 137–153. [\[CrossRef\]](#)
36. Hall, E.T.; Birdwhistell, R.L.; Bock, B.; Bohannon, P.; Diebold, A.R.; Durbin, M.; Edmonson, M.S.; Fischer, J.L.; Hymes, D.; Kimball, S.T.; et al. Proxemics [and Comments and Replies]. *Curr. Anthropol.* **1968**, *9*, 83–108. [\[CrossRef\]](#)
37. Honour, A.; Banisetty, S.B.; Feil-Seifer, D. Perceived Social Intelligence as Evaluation of Socially Navigation. In Proceedings of the Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction, New York, NY, USA, 9–11 March 2021; pp. 519–523. [\[CrossRef\]](#)
38. Mavrogiannis, C.; Hutchinson, A.M.; MacDonald, J.; Alves-Oliveira, P.; Knepper, R.A. Effects of Distinct Robot Navigation Strategies on Human Behavior in a Crowded Environment. *ACM/IEEE Int. Conf.-Hum.-Robot. Interact.* **2019**, *2019*, 421–430. [\[CrossRef\]](#)
39. Sorrentino, A.; Khalid, O.; Coviello, L.; Cavallo, F.; Fiorini, L. Modeling human-like robot personalities as a key to foster socially aware navigation. In Proceedings of the 2021 30th IEEE International Conference on Robot and Human Interactive Communication, Vancouver, BC, Canada, 8–12 August 2021; pp. 95–101. [\[CrossRef\]](#)
40. Campbell, N.; Mokhtari, P. Voice quality: The 4th prosodic dimension. In Proceedings of the 15th International Congress of Phonetic Sciences, Barcelona, Spain, 3–9 August 2003; pp. 2417–2420.

41. Khambhaita, H.; Alami, R. *Viewing Robot Navigation in Human Environment as a Cooperative Activity*; Springer: Cham, Switzerland, 2020; pp. 285–300. [CrossRef]
42. Luo, L.; Guo, T.; Cui, K.; Zhang, Q. Trajectory Planning in Robot Joint Space Based on Improved Quantum Particle Swarm Optimization Algorithm. *Appl. Sci.* **2023**, *13*, 7031. [CrossRef]
43. Akopov, A.S.; Beklaryan, L.A.; Beklaryan, A.L. Cluster-Based Optimization of an Evacuation Process Using a Parallel Bi-Objective Real-Coded Genetic Algorithm. *Cybern. Inf. Technol.* **2020**, *20*, 45–63. [CrossRef]
44. Sisbot, E.A.; Marin-Urias, K.F.; Alami, R.; Siméon, T. A human aware mobile robot motion planner. *IEEE Trans. Robot.* **2007**, *23*, 874–883. [CrossRef]
45. Kruse, T.; Basili, P.; Glasauer, S.; Kirsch, A. Legible robot navigation in the proximity of moving humans. In Proceedings of IEEE Workshop on Advanced Robotics and Its Social Impacts, ARSO, Munich, Germany, 21–23 May 2012; pp. 83–88. [CrossRef]
46. Repiso, E.; Ferrer, G.; Sanfeliu, A. On-line adaptive side-by-side human robot companion in dynamic urban environments. *IEEE Int. Conf. Intell. Robot. Syst.* **2017**, *2017*, 872–877. [CrossRef]
47. Rios-Martinez, J.; Renzaglia, A.; Spalanzani, A.; Martinelli, A.; Laugier, C. Navigating between people: A stochastic optimization approach. In Proceedings of the IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 2880–2885. [CrossRef]
48. Zhou, A.; Dragan, A.D. Cost Functions for Robot Motion Style. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018; pp. 3632–3639. [CrossRef]
49. Hagane, S.; Venture, G. Robotic Manipulator’s Expressive Movements Control Using Kinematic Redundancy. *Machines* **2022**, *10*, 1118. [CrossRef]
50. Park, J.J. Graceful Navigation for Mobile Robots in Dynamic and Uncertain Environments by. Ph.D. Dissertation, The University of Michigan, Ann Arbor, MI, USA, 2016.
51. Tail, L.; Zhang, J.; Liu, M.; Burgard, W. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, QLD, Australia, 21–25 May 2018; pp. 1111–1117. [CrossRef]
52. Fischer, K.; Jensen, L.C.; Suvei, S.D.; Bodenhagen, L. Between legibility and contact: The role of gaze in robot approach. In Proceedings of the 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), New York, NY, USA, 26–31 August 2016; pp. 646–651. [CrossRef]
53. Mumm, J.; Mutlu, B. Human-robot proxemics: Physical and psychological distancing in human-robot interaction. In Proceedings of the 2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Lausanne, Switzerland, 6–9 March 2011; pp. 331–338. [CrossRef]
54. Winter, B. *Statistics for Linguists: An Introduction Using R*, 1st ed.; Routledge: London, UK, 2019. [CrossRef]
55. Bates, D.; Mächler, M.; Bolker, B.; Walker, S. Fitting Linear Mixed-Effects Models Using lme4. *J. Stat. Softw.* **2015**, *67*, 1–48. [CrossRef]
56. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]
57. Baayen, R.H. *Analyzing Linguistic Data: A Practical Introduction to Statistics Using R*; Cambridge University Press: Cambridge, UK, 2008. [CrossRef]
58. Searle, S.R.; Speed, F.M.; Milliken, G.A. Population Marginal Means in the Linear Model: An Alternative to Least Squares Means. *Am. Stat.* **1980**, *34*, 216–221. [CrossRef]
59. Lenth, R.V. Emmeans: Estimated Marginal Means, aka Least-Squares Means. R Package Version 1.8.8, 2023. Available online: <https://cran.r-project.org/web/packages/emmeans/index.html> (accessed 25 March 2024)
60. Holm, S. A Simple Sequentially Rejective Multiple Test Procedure. *Scand. J. Stat.* **1979**, *6*, 65–70.
61. Tedrake, R. *Underactuated Robotics*; Course Notes for MIT 6.832, 2023. Available online: <https://underactuated.csail.mit.edu> (accessed on 25 March 2024).
62. RobAIR Mobile Robot, Designed and Built by FabMASTIC, Grenoble. 2021. Available online: <https://air.imag.fr/index.php/RobAIR> (accessed on 19 July 2021).
63. Hiroi, Y.; Ito, A. Influence of the Size Factor of a Mobile Robot Moving Toward a Human on Subjective Acceptable Distance. *Mobile Robots—Current Trends. InTech* **2011**, *9*, 177–190. [CrossRef]
64. Magnani, R.; Aubergé, V.; Bayol, C.; Sasa, Y. Bases of Empathic Animism Illusion: Audio-visual perception of an object devoted to becoming perceived as a subject for HRI. In Proceedings of the 1st International Workshop on Vocal Interactivity in-and-between Humans, Animals and Robots—VIHAR 2017, Stockholm, Sweden, 25–26 August 2017.
65. Walters, M.; Koay, K.; Syrdal, D.; Dautenhahn, K.; Boekhorst, R. Preferences and Perceptions of Robot Appearance and Embodiment in Human-Robot Interaction Trials. In Proceedings of the New Frontiers in Human-Robot Interaction, Symposium at the AISB09 Convention, Edinburgh, Scotland, 8–9 April 2009.
66. Matsumoto, M. Fragile Robot: The Fragility of Robots Induces User Attachment to Robots. *Int. J. Mech. Eng. Robot. Res.* **2021**, *10*, 536–541. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.