POLITECNICO MILANO 1863

School of Industrial and Information Engineering

Master of Science in Automation and Control Engineering

Improving the quality of human-robot collaboration by exploiting hands-head tracking

Supervisor: Prof. Paolo Rocco
Co-supervisors: Prof. Andrea Maria Zanchettin
Dr. Andrea Casalino

Author:
Costanza Messeri, student ID 841341

Academic Year 2017/2018

C'è uno spettacolo più grande del mare: il cielo,
c'è uno spettacolo più grande del cielo: l'interno dell'anima.

A tutte le persone che hanno sempre creduto in me e hanno saputo donarmi,
per mezzo del loro affetto, la forza necessaria per affrontare questo percorso.

Abstract

In a human-robot collaborative framework, endowing the robot with the capability of recognizing in advance the operator's intention proved to be beneficial for enhancing the quality of the cooperation. In this Thesis the problem of intention inference is addressed by using a new recursive Bayesian classifier which is capable of inferring simultaneously the most likely right hand and left hand reaching targets, relying on a set of measured skeletal positions. These measurements, retrieved by means of an RGB-D camera, include both wrist positions and an estimate of the operator's gaze. Indeed, this latter measure constitutes a powerful means of nonverbal communication exploited by humans to make their inner intents more explicit. In a collaborative framework, its use appears to be crucial for further improving the inference process, since its observation could provide an additional indication about the target the operator's hand will be intended to reach. The likelihood function which jointly considers the contribution of this wide set of observations is modelled through a Gaussian Mixture Model (GMM), learnt from data in a supervised manner. Since the exact goal positions are uncertain, each target location is described as a random variable having a certain probability distribution. The performance achieved through the new inference algorithm highlighted that the gaze measure was fundamental for guaranteeing higher robustness to the inference process. Moreover, the benefits of establishing a human-robot bidirectional information exchange during the collaborative process are investigated. Indeed, while the inference process is ongoing, the human co-worker could be kept informed about the prediction performed by the robot, for instance, by receiving an haptic feedback in the crucial phases of the collaboration. Considering all these aspects together can definitely improve the effectiveness of the collaboration.

Sommario

Nell'ambito della collaborazione uomo-robot, dotare il robot della capacità di riconoscere in anticipo l'intenzione dell'operatore si è rivelato vantaggioso al fine di migliorare la qualità della collaborazione. In questa Tesi il problema di dedurre la suddetta intenzione è affrontato ricorrendo a un nuovo classificatore ricorsivo Bayesiano capace di stimare simultaneamente quale sia il più probabile goal che verrà raggiunto da ciascuna mano, basandosi su un insieme di posizioni scheletali di interesse misurate. Tali misure, ottenute tramite una telecamera RGB-D, includono la posizione di entrambi i polsi e una stima dello sguardo dell'operatore. Quest'ultima misura, infatti, costituisce un potente mezzo di comunicazione non verbale impiegato dall'uomo per rendere più esplicite le proprie intenzioni. In un contesto collaborativo il suo impiego appare fondamentale per migliorare il processo di inferenza, dal momento che questa osservazione può fornire un'indicazione aggiuntiva dell'obiettivo che la mano dell'operatore è intenzionata a raggiungere. La funzione di verosimiglianza che considera congiuntamente il contributo di questo ampio insieme di osservazioni è modellizzata mediante un Gaussian Mixture Model (GMM) appreso dai dati in modo supervisionato. Poiché le posizioni obiettivo potrebbero essere non note con esattezza, ognuna di esse è espressa come variabile aleatoria caratterizzata da una certa distribuzione di probabilità. Le prestazioni raggiunte mediante il nuovo algoritmo di inferenza evidenziano che la misura dello sguardo è fondamentale per garantire una maggiore robustezza al processo di inferenza. Sono stati inoltre analizzati i vantaggi di stabilire uno scambio di informazioni bidirezionale tra uomo e robot durante il processo di inferenza. Infatti, mentre quest'ultimo è in corso, l'uomo potrebbe esser tenuto informato sulla predizione effettuata dal robot, ad esempio, ricevendo un feedback tattile durante le fasi cruciali della collaborazione. Considerare congiuntamente i suddetti aspetti migliora significativamente l'efficacia della collaborazione.

Ringraziamenti

Durante la stesura di questa Tesi e, in particolar modo, mentre rivolgo i miei pensieri alle persone che hanno reso possibile questa straordinaria esperienza e per le quali nutro una profonda riconoscenza, la mia mente non può fare a meno di soffermarsi sui momenti più significativi e incisivi del mio percorso di studio al Politecnico e di tutti gli eventi ad esso connessi. Ricordo distintamente il momento in cui tutto è cominciato e in cui, fortemente emozionata, stavo per a varcare la soglia del maestoso ingresso di Piazza Leonardo. Fin dal primo istante devo riconoscere che ciò mi ha guidata durante tutto questo percorso è stata una profonda passione per la conoscenza, per l'apprendimento, una decisa volontà di acquisire nozioni nuove e senza dubbio importanti, fondamentali per la crescita culturale personale e per essere impiegate a vantaggio del meraviglioso ed eterogeneo mondo in cui viviamo.
Infatti ho sempre ritenuto Ingegneria la facoltà capace di fornire gli strumenti, le conoscenze, le metodologie applicative che avrebbero permesso ad ogni singolo studente di provare a dar vita in modo concreto alle proprie specifiche abilità e a generare il proprio valore aggiunto.
Lo svolgimento di questa Tesi mi ha permesso di ampliare ulteriormente le conoscenze acquisite durante gli studi e di approfondire svariati ambiti e metodologie molto interessanti. Inoltre essa mi ha dato la possibilità di avvicinarmi ad un ambito fortemente innovativo e attuale, quale la robotica collaborativa.
A questo proposito un ringraziamento particolare è rivolto al Professor Rocco per i consigli che mi ha fornito durante questo percorso e per la sua disponibilità. Lo ringrazio inoltre per avermi dato la possibilità di dedicarmi all'argomento che ho svolto nella Tesi e di avvicinarmi quindi al mondo della human-robot interaction.
Un ulteriore ringraziamento è rivolto al Professor Zanchettin per la sua disponibilità, i suoi illuminanti consigli e lungimiranti discussioni che mi hanno permesso di svolgere al meglio delle mie capacità questo lavoro di Tesi.
Ringrazio particolarmente il Dott. Casalino per la sua profonda dedizione, pazienza, i preziosi consigli, fondamentali per lo svolgimento di questo lavoro e per il tempo dedicato a seguire ogni sviluppo di questa Tesi.
Un ringraziamento molto speciale è dedicato alla mia famiglia, che mi ha dato la possibilità di seguire il corso di studi e l'università che desideravo, che mi è

stata sempre accanto sostenendomi in modo impareggiabile e trasmettendomi un profondo senso di fiducia specialmente nei momenti più difficoltosi. A questo proposito ringrazio di cuore la mia carissima sorella che sa far brillare di luce ogni circostanza grazie al suo acume, alla sua intelligenza e al suo carattere spontaneo e solare.
Ringrazio di cuore Danilo per l'inestimabile affetto, sostegno e fiducia che mi ha donato in ogni momento standomi costantemente vicino.
In ultimo vorrei rivolgere un particolare ringraziamento a tutti i cari amici e le carissime amiche che durante questo percorso mi sono stati accanto: nel corso della mia vita e nel periodo di studio trascorso a Milano ho avuto la fortuna di conoscere persone meravigliose che terrò strette nel mio cuore.

Contents

List of Figures

1

Introduction

1.1 Collaborative robotics

In recent years a revolutionary change in the paradigm of industrial robots occurred. Indeed, the definition of the industrial manipulator traditionally makes reference to a programmable, sensorized mechanical system mainly devoted to handling massive loads in manifacturing operations, designed to work without the supervision of a human operator and rigidly segregated in specific environments.

Far from this framework, while the so-called 'Industry 4.0' is redefining the trend for smart factories and pushing on the concept of interoperability, the term 'cobot' –listed as a new word by Wall Street Journal on Jan. $1, 2000$– is now spreading, becoming part of the current robotic dictionary and paving the way for a new concept of robot [35].

The term 'cobot' is therefore used to identify a collaborative robot: a smaller and lighter manipulator that does not need to be surrounded by any protection fences, that shares its workspace with the human operator, specifically designed to assist him and work in close proximity and cooperatively with him. In fact, the collaborative robot is intended to be a co-worker that helps human in carrying out a sequence of operations. In the light of the above, a collaboration between human and robot makes it possible to combine the human capability of performing very complex tasks and the cobot ability to execute repetitive and high-precision operations. Taking all these factors into consideration, this can lead to an increase in the overall production and improving the quality of the final products, while reducing the cycle time and variable costs.

Indeed it has been recently found that the cooperation between humans and robots has the effect of enhancing the productivity of about 85% with respect to the corresponding work done by only humans or robots.

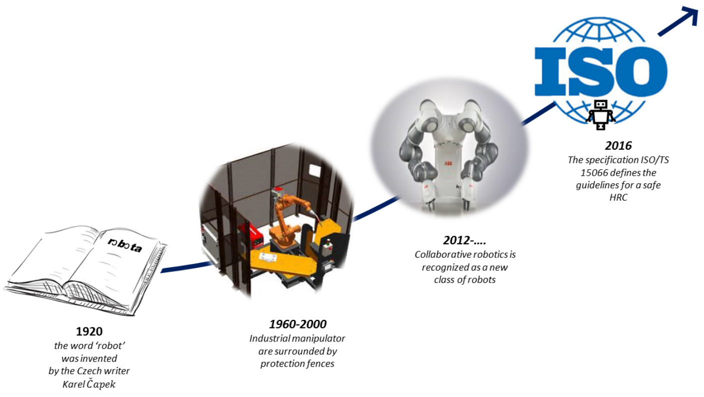


Figure 1.1: The evolution of the robot's history

This scenario provides the foundation for the definition of two important concepts: on the one hand the 'collaborative workspace' –namely, the region where human and robot are supposed to jointy work–, on the other hand the 'collaborative operation' –for instance, the task or the sequence of tasks that they are willing to perform together, taking advantage of their own skills and capabilities.

In this scenario it is essential to guarantee that the overall collaboration is carried out safely, to avoide damages to the human operator. Hence the robot is tasked with performing those operations which are highly stressful, tiring and repetitive for most individuals.

This latter aspect is practically solved in industrial context by creating cobots that exploit their characteristic kinematic redundancy to move as close as possible the way the human arm moves, so as to reduce the physical stress of the operator and ensure him a more ergonomic work.

For what concerns the field of safe human-robot interaction, the international boards define the standards and guidelines, together with protective measures and information for the use of these industrial robots. Moreover the technical specification ISO/TS 15066, a recent update and concretization of ISO 10218, assesses that the contact between robot and human operator is possible, provided that the values of maximum force and maximum energy that can be applied on a human without any harm are not overcome.

Hence, the collaborative robots are lighter than the traditional ones and without sharp edges; furthermore they are generally topped with filling that can soften the strenght of the impact with human. In addition these cobots are equipped with a variety of sensors which allow them to be intrinsically safe and precise: they can promptly recognize the presence of a potential obstacle and stop immediately or reduce the speed to avoid collisions.

It is quite evident that the use of a cobot can provide significant advantages: fast set-up and programming, operational flexibility that allows to employ them in a variety of applications, a reduction of the time-to-market, an easy re-programming; finally, the capability of working safely and efficiently. Due to their simple and non-invasive layout, they generally take up less space than the traditional robots and require a very simple equipment and work environment, resulting in reduced establishment costs.

Ultimately, the collaboration between robot and human operator guarantees that the task is constantly supervised, thus improving the quality of the final products and, in view of the shared workspace, a reduced occupancy of the working area.

The collaborative robotics is a growing sector which represents an intermediate segment between completely automated assembly plants, characterized by large and fixed production batches, and those completely manual, with small and variable production batches.
Due to their advantages, cobots are supposed to spread everywhere, not solely in the industrial context; however the traditional robots will remain a valuable alternative in those industrial contexts characterized by high productivity and a fixed production.

The statistics reported by the International Federation of Robotics (IFR) confirm that the overall interest in collaborative robotics is actually growing and it is expected to increase further during next years: the guideline of this revolution is the transformation of the robot in a cyber-physical system that can store data, communicate via cloud, that is equipped with sophisticated artificial intelligent software, a system that is easily adaptable and capable of cooperating efficiently with the human.

1.2 Human intention inference: state of the art

In a collaborative scenario, endowing the cobot with the capability of predicting what the human operator is going to do and how he will perform it can guarantee numerous advantages as allowing for planning ahead for reactive responses to human activities and improving the quality of the overall collaboration [17], ensuring a fluent meshing of operator and manipulator activities. In fact, it can be easily noticed that, even in the framework of human-human cooperation in day-to-day activities, the effectiveness of the collaboration is significantly determined by their level of coordination-synchronization and by the quality of their interaction. In other words, the success of the cooperation depends on the capability of anticipating the partner's next activity, i.e. the ability of understanding implicit commands and not directly observable desires. This aspect can be equivalently valid in a context where human an robot cooperate, as long as the manipulator behaves in a similar manner to its human counterpart and cooperating with it becomes as intuitive as working with other human operators. Hence, it is apparent that the cooperation between humans and robots could become more appealing if the robot is able to make anticipative decisions that matches the operator's ones, as long as the human-robot joint activity is well synchronized.

The cognitive mechanism and, more specifically, the problem of human intention inference has become relevant during last years. Indeed, this issue has been widely addressed in the literature: in [23] a method based on Anticipatory Temporal Conditional Random Field (ACTRF) was used to predict the human's next action. Each observed human activity was first modeled using a spatio-temporal graph denoted as conditional random field (CRF); then, in order to anticipate the future human action, a temporal segment was added to the CRF so as to create the ATCRF model, where each ATCRF represents a kind of evolution of the current CRF. By considering each possible ATCRF as a particle and exploiting particle filtering algorithm, the probability distribution over the possible ACTRF was computed and the most likely future action was estimated. In [19], based on a RGB-D video of a human performing some operations in a certain environment, the authors were able to predict his future activity and his future pose by using Gaussian Process Latent Conditional Random Field (GP-LCRF): a structure which allowed them to jointly model the human motion dynamics and poses. Then, they applied Gibbs sampling for evaluating the possible future human configurations and to anticipate the next human action. In [42] a Bayesian Network called Sequential Interval Network (SIN) was used to graphically model the human intention evolution. In fact, a SIN is basically a graphical model where the variable nodes represent the start and end times of component actions: hidden nodes are action timings, observed nodes are the output of primitive action detectors and edges describes the relations between action and action or action and detection. Once the conditional probability table describing the dependencies of the network had been learnt, the message-passing algorithm was used to perform exact inference.

[21] used a trained HMM to estimate the most likely human future activity from the perspective of a mobile robot.

[32] proposed a method that allowed real-time online prediction of the goal

which the human hand is willing to achieve by representing the human motion realized at each time step as a multivariate gaussian distribution over the degrees of freedom of the human arm. The prediction was then obtained by performing a Bayesian classification using the initial portion of the trajectory executed by the human arm. Clearly, this mechanism required an offline training phase prior to proceeding with online data processing. This approach overcomed the performances of [27] attaining a faster prediction. In fact, also in [27] a library of human motion trajectories was codified offline using a GMM and the swept volumes were computed. Then, during the online phase, the GMM was used to compute the robot activity that minimizes the interference with the human, after the prediction of the workspace ocuppancy had been carried out.

In [24] the operator's hand position was identified through the use of a 3D occupancy grid: this position constituted the input for a trained HMM which was used to estimate the action that the human will perform.

[33] modeled intentions as the target locations of reaching motions in the 3D space, where the motion was represented as a nonlinear function whose parameters were human intentions. These parameters were learned using a neural network and intentions were inferred by applying an approximate version of the customary expectation-maximization algorithm. Moreover an adaptive identifier was used to take into account the possible variations of human motions due to different initial conditions or different human subjects performing the experiment.

Finally, [2] used a probabilstic state machine to recognize human intentions. They considered two possible situations: explicit intention, namely, when the human was truly collaborating with the robot, and implicit intention, i.e when the human was not actively cooperating and the robot decided to initiate a proactive collaboration on the basis of the observed scenario and the human actions. The set of objects, admissible human actions, and possible human intentions were defined prior to running the intention recognition algorithm.

It is clear that the majority of the previous approaches required an offline training phase and a subsequent a priori classification of set of objects, human actions and, sometimes human intentions. The works which will be explained afterwards are quite similar to the method that will be presented in this Thesis.

[31] proposed a method that acts by continuously tracking the human arm position and, given a predefined set of goals, infers their probability distribution on the basis of the movement of human arm. For what concerns the cobot action given the human one, they used a module called POMPD (Partially Observable Markov Decision Process) which idenitified the best robot behaviour and, assuming that human and robot cannot achieve simultaneously the same

goal position, generated commands for the robot arm which reaches for the closest free target. In order to update the probability distribution of the goals, the Bayes rule was applied and the likelihood function was evaluated using a model where the probability decreases exponentially with the cost, that is, the Euclidean distance. This technique adapts to the HRC the one developed in [12] for teleoperation to human-robot shared workspace collaboration. In [37] the problem of recognizing the most likely human reaching target was addressed from a different perspective. In fact, the recognition problem was solved by using a probabilistic model that inferred the human intentions from the observed actions. The use of this probabilistic model was motivated by the fact that these observations, together with the correlation between observed actions and corresponding intentions, were subject to uncertainty. More specifically, they used a Hybrid Dynamic Bayesian Network (HDBN), where 'hybrid' referred to the fact that both continuous and discrete-value states were used to model the relations between sensors measurements, intentions and actions. The advantage of this model is that it allows to consider the causal dependency of the user actions from its unknown intentions, that is why is denoted also as 'forward model'.

Lastly, in [34] the intention inference issue was addressed in a different manner. One of the benefits of this approach was that it did not require a preliminarly training phase. Hence, this method was extremely advantageous, since it could be applied in all sort of contexts, eliminating the time usually required during the training set-up.

With this approach the human hand reaching path was modelled with a minimum curvature trajectory and the probability of reaching each target was computed by exploiting the recursive Bayes' rule. The measurement monitored at each time step was a single position: the location of the operator's wrist. At each iteration, the tangent vector associated with the measured wrist position was compared with the corresponding tangent vector associated with the ideal reaching path that leads from the measured wrist position to each goal center of mass. The angle betwen the abovementioned tangent vectors was evaluated according to a Gaussian distribution; hence, the corresponding conditional probability of observing this angle under the assumption of reaching a certain goal position was used to update the probability associated to that target. However, each target was represented by a single point coordinate, without taking into account the uncertainty associated with the knowledge of the exact target position. Moreover the intended goal was inferred relying on the single wrist position, as explained previously, without taking into consideration the information related to other skeletal measurements.

1.3 Thesis purpose

The aim of this Thesis is to propose a novel intention inference algorithm which, based on the application of a recursive Bayesian classifier, is capable of inferring simultaneously the most likely human reaching goals of both hands by exploiting a large number of observations. Indeed, this work is inspired by a previous inference algorithm ([34]) where only the single wrist position was used to estimate the target the measured wrist was heading to.
In this Thesis, the set of observations which will be used to perform inference on both hands reaching targets will include the positions of both right and left wrists along with the orientation of the operator's head which is considered to be an estimate of his/her gaze. These quantities will be retrieved by means of a Microsoft Kinect camera. The proposed inference algorithm, in fact, will recursively estimate the human intention by characterizing it (from a probabilistic point of view) as the goal position that, when the operator's hands are moving, iteratively acquires a higher value of probability with respect to the others which are part of the finite set of possible target locations. Hence, these goals corresponds to the positions of the objects which could be intended by the human operator during the collaboration with the robot.
The purpose of the intention inference is to ensure a fluent meshing between the robot and the human counterpart. In fact, if the robot is endowed with the capability of recognizing in a robust way the task of its human partner, it can start to complement appropriately the human activity, thus guaranteing a reactive behaviour towards human intentions as well as an overall proactive collaboration. Even in the case of human-human collaboration, the gaze is proved to beneficial since it consitutes a means of non verbal communication through which individuals can make more explicit their latent intentions, [39], [43].
Motivated by these facts, this Thesis will investigate how the gaze estimate, when jointly exploited with the other available observations contributes to improving the performance of the overall inference process.
At a first stage this work will be focused on investigating whether a suitable model of the human hand's reaching path can be determined such that it could result to be particularly effective for further improving the inference process.
Moreover, once the likelihood function has been properly modelled, the proposed algorithm must be able to work online.

In order to implement this algorithm, the following elements are needed:

- some prior knowledge about the workspace: in particular, the number and location of the targets which the human operator could intend to reach must be defined before starting the inference process;

- a possible model describing the actual reaching path followed by the operator's hand when reaching a goal;
- a model for the likelihood function which includes the joint contribution of all the available observations.

1.4 Achievements

The results achieved in this Thesis through the application of the novel intention inference algorithm are here summarized:

- the intended goal is correctly recognized at approximately half of the hand reaching path, thus obtaining a slight improvement with respect to the corresponding result described in [34].
- including the information related to the distance between each wrist position and the goal center of mass within the features vector used to evaluate the likelihood allows to obtain a significant reduction of the percentage of 'true negatives', i.e., those cases where an intended goal is not recognized by the inference algorithm, compared to the approach described in [34].
- introducing also the estimate of the operator's gaze in the features vector helps to further increase the robustness of the inference process, since it leads to a reduction of the percentage of the 'false positives', i.e, the cases where a high value of probability is assigned to non intended goal positions, with respect to the method used in [34].
- equipping the operator with a wearable device that sends him a vibrotactile feedback during the crucial phases of the cooperation with a collaborative robot has the advantageous effect of creating a bidirectional communication channel between human and robot. This means that, while the robot estimates the operator's intended goal exploiting the inference algorithm, its human partner is kept informed on the current state of the inference by means of the haptic feedback. This additional contribution enhance the effectiveness of the human-robot cooperation. Moreover, for non-skilled subjects a reduction of the average time required for executing the overall collaborative task has been observed.

1.5 Thesis structure

The rest of this Thesis is structured as follows. In Chapter 2 an overview of the previous approach which exploited one single observation to update the intentions' estimate will be presented together with the probabilistic framework that lays on the basis of the early prediction of humans intentions, namely, the Bayesian approach. This approach was extended in this Thesis. In Chapter 3 the problem of finding a suitable model for descibing the actual path followed by the human hand during its reaching motions will be addressed. Different types of paths will be presented and their performance will be analyzed and compared with the corresponding state of the art in order to draw conclusions about their contribution to enhancing the process of intention inference. In Chapter 4, the possibility of including a larger set of observations such as the estimated gaze, head position, hands' distance from the target and both hand positions to improve the inference process will be discussed. Thus, the new intention inference algorithm which estimates the operator's intention based on the quantities discussed so far will be described. In Chapter 5 the uncertainty associated with the exact position of each goal will receive a proper probabilistic characterization such that each goal will be represented as a random variable having a certain probability distribution. Chapter 6 will describe the results achieved through the application of the new inference algorithm and its performance in terms of average distance at which the goal is correctly recognized; furthermore, some statistics about the robusteness properties will be shown. These results will be compared to the performance achieved by applying the algorithm described in Chapter 2. In Chapter 7 a realistic collaborative experiment between a human operator and a dual-arm collaborative robot will be described. The aim will be not only to further highlight the benefits gained with a correct estimation process, but also to investigate how beneficial could be sending an haptic feedback to the operator as soon as the robot, based on the new algorithm, understood his/her intention. In Chapter 8 some conclusions about this work will be drawn and future developments will be discussed.

2

Background on Bayesian framework and inference algorithm

2.1 Introduction

In this chapter the approach described in [34] will be resumed. That method made it possible to infer the human hand reaching target based on one single observation, the wrist position, exploiting a model-based trajectory to represent the human-hand reaching path. Moreover, this procedure lays the foundations for the new inference algorithm which will be described afterwards.
Furthermore, the very first part of this chapter will address the probabilistic theory and, more specifically, the principles of Bayesian inference underpinning the abovementioned algorithm.

2.2 The concept of human intentions

Before adressing the issue of human intention estimation it is worth defining properly the meaning assigned to the concept of human intention along with the reason for applying a probabilistic approach. From an abstract point of view, the expression 'human intention' obviously refers to a latent human will, belonging to a variety of possible willingnesses, which progressively becomes more and more evident due to the observation of actions, movements and gestures the human does. As a matter of fact, these latter information, when considered together, help in clarifying the underlying intention. More specifically, in this Thesis, the variety of possible human intentions will be confined to the framework of human robot collaboration and will be used to refer to the set of human hand reaching targets. As a consequence, it is then apparent that, from a practical perspective, the set of operator intentions will be limited to a number of physical goals which the human hand could be directed to. Hence the willingness of reaching one of the targets can be interpreted as hidden state [37] that, living in the operator's mind, cannot be directly observed. However, this state can be progressively estimated through the set of gestures the worker performs and through the analysis of the environment in which he is operating. In fact, since human actions generate observable events, they can be easily recognized.

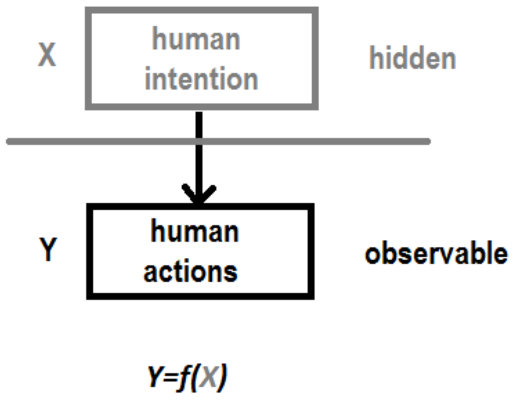


Figure 2.1: Schematic illustration of unobservable human intention (X) and observable human actions (Y) through which the intention can be deduced

Therefore, if we denote the underlying intention with the variable X and the set of observable human actions with the variable Y, these latter will be somehow related to the unknow intention through a certain function f. Since this function is not deterministically given, and, as observed in [37], the relationship between intentions and actions suffers from uncertainty, it seems reasonable to model it from a probabilistic perspective; namely, in a way that make it possible to progressively infer the intention, by exploiting the likelihoood of the observations of the human actions, the working environment, and the knowledge of the finite set of possible goals.

The tool that allows us to address this problem is the Bayesian analysis.

2.3 Bayesian framework

The Bayesian analysis is a probabilistic tool that constitutes the very heart of statistical inference. Bayesian intention inference was already addressed in literature when dealing with estimation matters: in [34] the recursive Bayesian method, applied to the observations related to the predictive hand reaching path, was used to make inference about the operator's intended target. In addition, in [5] this procedure was used to predict the intended goal destination and the agent's future trajectory which was considered conditional on the intention of the agent and known to the agent itself only.
In fact the recursive Bayesian method makes it possible to evaluate the probability of a certain event by iteratively updating its estimate while including at each iteration the information related to the new incoming observation along with the a priori probability of that event. This probability is usually denoted as 'prior'.
Hence the Bayes' rule allows us to compute the so-called 'posterior' probability of a certain event, starting from the knowledge of its prior and the so-called 'likelihood' or 'evidence' of the data, a quantity that represents how likely is a certain observation under a prescribed assumption.

Due to the possibility of applying this method in a recursive way, the Bayesian Inference provides a way to be reactive to the changing system conditions. Therefore, it clearly becomes a powerful tool that is particularly advantageous in dynamic circumstances, when the probability of the event at a certain time instant strongly depends on the new information acquired by the measurement system, other than the probability at the previous time instant.

As previously explained, the recursive Bayesian approach operates by applying the following formula, denoted as 'Bayes' rule' [15], in a recursive way:

Proposition (Bayes' rule) *Let (Ω,$\mathscr{F}$,P) be a probability space*[1] *and* F_1,F_2,...,F_n

[1] A *probability space*, [15], is the triple (Ω,$\mathscr{F}$,P) where Ω is the *sample space* that contains all possible outcomes, $\mathscr{F}$ is the *event space* and P is the *probability function* $P\colon \mathscr{F} \to \mathbb{R}$ for which the following axioms must be fulfilled:

- $P(E) \geq 0 \; \forall \; E \in \mathscr{F}$;
- $P(\Omega)$=1;
- if E_1,E_2,...$\in$ $\mathscr{F}$ are disjoint events, for instance, $E_h \cap E_h = \emptyset$ if h $\neq$ k, then $P(\bigcup_{k=1}^{\inf} E_k) = \sum_{k=1}^{\inf} P(E_k)$.

.

$\in \mathscr{F}$ *a finite partition* Ω *such that* $P(F_k) > 0$ *for k=1,..n. If* $E \in \mathscr{F}$ *is such that* $P(E) > 0$*, then*

$$P(F_h \mid E) = \frac{P(E \mid F_h)P(F_h)}{\sum_{k=1}^{n} P(E \mid F_k)P(F_k)} \quad h = 1, ..., n \tag{2.1}$$

In view of this reasoning, it is possible to find a way to solve even the problem of estimating the human intention, by exploiting the Bayes method. In fact, given the location of a number of objectives that the operator might reach, the intended target location can be interpreted, from a probabilistic point of view, as that goal position that is acquiring a higher probability with respect to the others, given at each iteration the observation related to a certain set of human movements. Hence, the recursive application of the Bayes formula allows us to infer at each iteration how the probability is distributed over the number of the goals.

In order to better specify, let us assume that we want to address the problem of computing the probability which at the generic k_{th} step the human hand reaches a certain goal position p_{G_i} that belongs to a set of viable goal positions $\mathscr{G}$.

Let us also suppose that the set of goal positions is a priori known and consitutes a sort of common knowledge shared by both the human and the robot. Moreover, let each goal position p_{G_i} be described, for the time being, as the three-dimensional vector that expresses the coordinates of the center of mass of each target (as done in [34]). In Chapter 5 a broader description of the target positions will be provided and the possibility of representing them in a more specific way will be discussed.

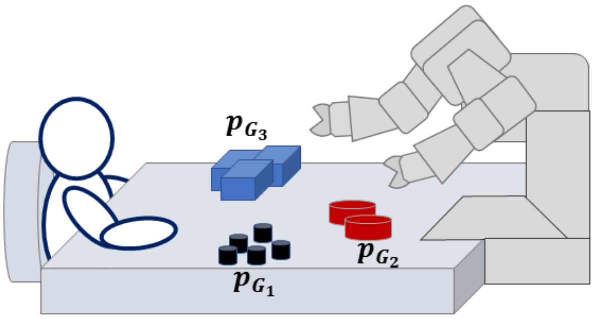


Figure 2.2: The goal position $p_{G_i} \in \mathscr{G}$ that the human hand is intended to reach must be inferred

Thus, the i_{th} goal position is, for the time being, defined as:

$$p_{G_i} = \begin{bmatrix} x_{G_i} & y_{G_i} & z_{G_i} \end{bmatrix}^T$$

Before proceeding with the mathematical formulation, it is possible to preliminarly define the meaning assigned to some quantities.
Let us denote with:

- $P^{(0)}(p_{G_i})$ the a priori probability distribution associated to the i_{th} goal position $p_{G_i} \in \mathscr{G}$ (at iteration zero);
- $x^{(0:k-1)}$ the set of contextual observed measurements related to the movement of the human operator that have been collected from time step zero to time step $k-1$;
- $f(p_{G_i} \mid x^{(0:k-1)})$ the conditional probability density function (pdf) of the goal given the set of observations $x^{(0:k-1)}$;
- $f(x^{(0:k-1)} \mid p_{G_i})$ the conditional probability density function (pdf) of observing the measures $x^{(0:k-1)}$ given the willingness of reaching goal position p_{G_i}, also called 'likelihood of the observation'.

Then, the a posteriori probability of the goal position p_{G_i} at the k_{th} time step can be written as:

$$P^{(k)}(p_{G_i} \in \mathscr{G}) = f(p_{G_i} \mid x^{(0:k)}) \tag{2.2}$$

hence, at each k_{th} time step, the probability of reaching the goal position p_{G_i} can be updated by iterating Bayes' rule:

$$P^{(k)}(p_{G_i} \in \mathscr{G}) = f(p_{G_i} \mid x^{(0:k)}) = \frac{P^{(0)}(p_{G_i}) f(x^{(0:k)} \mid p_{G_i})}{\sum_{p_{G_i} \in \mathscr{G}} P^{(k)}(p_{G_i})} \tag{2.3}$$

that for the first time step clearly reduces to,

$$P^{(1)}(p_{G_i}) = \frac{P^{(0)}(p_{G_i}) f(x^{(0:1)} \mid p_{G_i})}{\sum_{p_{G_i} \in \mathscr{G}} P^{(k)}(p_{G_i})} \tag{2.4}$$

Thus, it can be easily noticed that at the subsequent time step $k+1$, the quantity $P^{(k)}(p_{G_i})$ that at time step k represented the posterior probability of p_{G_i}, acquires a new meaning, becoming now the prior probability associated to p_{G_i}.

This means that, essentially, apart from the normalization term of the denominator, the following relation holds:

$$P^{(k+1)}(p_{G_i}) \propto P^{(k)}(p_{G_i}) f(x^{(k+1)} \mid p_{G_i}, x^{(0:k)}) \tag{2.5}$$

where $f(x^{(k+1)} \mid p_{G_i}, x_{(0:k)})$ represents the conditional density of observing the measure coming at $k+1$ given the intention of reaching p_{G_i} and all the previous history of observations from the initial time step to the k_{th} step.

It is now evident that the process of considering at each time step the contribution given by the whole sequence of observations up to the current time instant has two disadvantages, as observed in [34]: firstly, the application of this procedure would require an amount of memory that increases at each iteration; secondly, the next expected observation would be considered strongly dependent on the trend obtained according to the previous history of observations.

As suggested in [34], if the sequence of incoming of observations $x^{(0:k)}$ is interpreted as a Markov' chain such that the Markov assumption [2] holds, the previous formula can be easily reduced to

$$P^{(k+1)}(p_G) \propto P^{(k)}(p_G) f(x^{(k+1)} \mid p_G, x^{(k)}) \tag{2.6}$$

thus obtaining the advantage of decreasing the computational complexity and of softening the dependance of the expected observation from the course of all the preceeding ones.

A proper model of the prior probability and of the likelihood will be discussed in section 2.4.

So far $x^{(k)}$ was used to denote the set of observations that can be used at time step k to compute the likelihood, hence, the measurements that can provide significant cues for estimating the agent's unknown intention. It is important to underline that the number and the type of measurements that are used within the expression of the likelihood is completely arbitrary and depends on how the intention recognition mechanism is organized. For the time being, the state variables $x^{(k)}$ are confined to:

[2] The Markov property [14] states that the conditional probability $x^{(k+1)} = j$, $x^{(k)} = i$, $x^{(k-1)} = i^{(k-1)}$, ..., $x^{(1)} = i^{(1)}$, $x^{(0)} = i^{(0)}$ is the same as the conditional probability $x_{k+1} = j$ given only the previous state $x^{(k)} = i$. In other words, the conditional probability of the future states depends only upon the current state, not on the whole sequence of preceeding states so that any other information about the past is irrelevant for predicting $x^{(k+1)}$.

- the position of the centre of the right wrist, denoted as p_{RW};
- the position of the centre of the left wrist, denoted as p_{LW};

Here, the wrist position is supposed to be a good estimate of the operator's hand position. So, from now on, the operator's hand will be equivalently referred as operator's wrist.

In view of the previous considerations, the state vector whose evolution can be considered is:

$$x^{(k)} = \begin{bmatrix} p_{RW}^{(k)} \\ t_{RW}^{(k)} \end{bmatrix}$$

or

$$x^{(k)} = \begin{bmatrix} p_{LW}^{(k)} \\ t_{LW}^{(k)} \end{bmatrix}$$

where t_{RW} and t_{LW} represent the derivative of the right and left wrist positions, respectively, with respect to the natural coordinate s that describes the ideal path followed by the human operator.

In Chapter 4 an extention of this state vector will be discussed and it will be explained that it is possible to include the observations related to the head position and an estimate of the human gaze to improve the inference process.

2.4 Intention inference algorithm: background

Once the main components that are needed to perform Bayesian inference have been explained, it is possible to show in detail how all this information can be combined to estimate the human hand reaching target. Hence, in this section an overview of the procedure used in [34] will be proposed.

This approach, in fact, made it possible to infer the intended goal based on the measurement of the single wrist position.

Indeed, the structure of the inference algorithm becomes more and more complex as the number of observations that are simultaneously exploited to make inference increases. Therefore, it seemed reasonable to provide a first description of the basic architecture used when dealing with a single observation so as to understand the general mechanism of functioning of an intention inference algorithm. The latter, in fact, lays the foundations for the new algorithm proposed in this Thesis, that updates the estimate of each goal by exploiting a larger set of observations (refer to Chapter 4).

Therefore, let us focus on the analysis of the generic structure of the inference algorithm that relies on one single observation, the wrist position.
As previously said, in order to infer the operator intended goal, the intention estimation algorithm basically receives as input the three-dimensional coordinates of the target positions. Clearly, these goal positions represent the objectives of the collaboration between the human operator and the cobot: hence, it is implicit that the layout of the human-robot common workspace must be uniquely determined before starting the estimation. This issue will be widely addressed in Chapter 5. Moreover this algorithm receives as input the position of the right wrist or of the left wrist, according to the hand that the operator is moving to reach the intended object. In fact, in this chapter it is assumed that the operator executes a sequence of tasks by using a single hand for all the duration of the task: be it the left one or the right one.

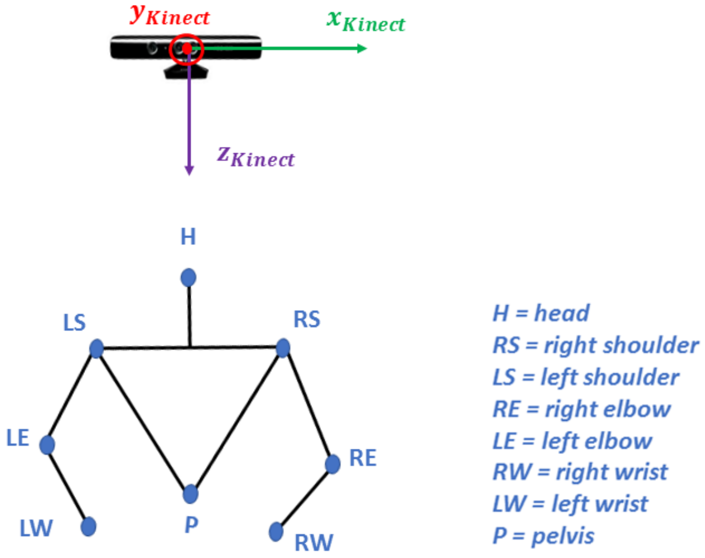


Figure 2.3: The Kinect camera detects the skeletal points of the human operator

The right hand or left hand positions can be easily retrieved by means of a sensor device. In fact, a Microsoft Kinect depth camera is used to return an estimate of the set of interesting points of the operator's body. These positions, usually denoted as 'skeletal points', are schematically represented in Figure 2.3. As previously exposed, the relevant variables (see Chapter 4) that will be exploited in this Thesis will be the ones illustrated in Figure 2.3.

- position of the centre of the right wrist, denoted as p_{RW};
- position of the centre of the left wrist, denoted as p_{LW};
- position of the centre of the head, denoted as p_H;
- vector that provides an estimate of the head's orientation, denoted as $z_{HeadVector}$.

The algorithm returns, for each set of input positions received, the most likely human reaching target.

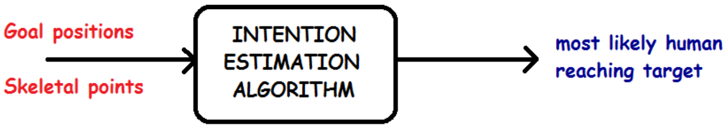

Goal positions
Skeletal points
INTENTION ESTIMATION ALGORITHM
most likely human reaching target

Figure 2.4: Schematic illustration of the input and output of the intention estimation algorithm

Let us come back to consider the case where only p_{RW} or p_{LW} are available. The algorithm works as follows.
Once the goal positions have been defined and their number is known, it is fundamental to assign to each one of them a certain value of probability, that will represent the prior probability of each goal at the initial time step. Since at the beginning of the collaboration the operator has not yet reached any target and there is no evidence that one is more likely than the others, it seems reasonable to consider the probability to be uniformly distributed over the number of the goals; namely, each target is assigned a value of probability that is equal to the ratio between 1 and the total number of targets.

Once the algorithm receives as inputs goal positions, skeletal measures, initial probabilities associated with the goal location, it collects all this information in a structure that from now on will be referred to as 'Inference Engine'. This structure is in charge of processing the new incoming information and updating the probabilities.

Going into the implementative details, the Inference Engine is endowed with a buffer which has the task of storing the wrist position returned by the Kinect at each iteration. Each buffer is a 3xDb matrix whose rows represents respectively the x, y, z coordinates associated with the wrist position observed at a certain k iteration, $p_W^{(k)}$. The dimension of the buffer, Db, is an arbitrary odd number which must be selected so as to allow the collection of a sufficient number of observations.

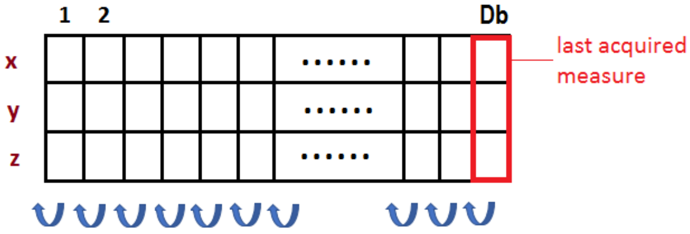


Figure 2.5: A schematic picture of the buffer filled according to a FIFO logic

The buffer is filled according to a First-In-First-Out (FIFO) logic.
To be more precise, when a new admissible measure $p_W^{(k)}$ arrives, firstly, all the measurements already present in the buffer are moved back one place; secondly, the new acquired one is positioned in the last position of the buffer (see the column marked in red in Figure 2.5). Clearly this logic is iterated for each new relevant measure retrieved. This way the measurements stored in the buffer are always ordered from the oldest one to the most recent one. To be inserted in the buffer the new measure must overcome a specified 'spatial' threshold, which is an arbitrary parameter of the algorithm. Thus, this specifies the minimum distance that two subsequent detected skeletal positions must have to become part of the buffer. More specifically, the criterion for deciding whether or not to introduce in the buffer the skeletal position $p_W^{(k)}$ measured at time step k is the following:

$$\begin{cases} \text{put } p_W^{(k)} \text{ in the buffer} & \text{if } ||p_W^{(k)} - p_W^{(k-1)}|| > \text{threshold} \\ \text{do not put } p_W^{(k)} \text{ in the buffer} & \text{otherwise} \end{cases} \tag{2.7}$$

where $p_W^{(k-1)}$ is here used to denote the last measure previously introduced at the end of the buffer.

Thus, the role of the buffer is to store some acquired samples in order to estimate the direction of the wrist position; namely, the buffer allows the computation of these two quantities:

- the one that from now on will be referred to as 'previous tangent vector', $\hat{t}_{prev}$

- the one that from now on will be referred to as ‘future tangent vector’, $\hat{t}_{fut}$.

In order to clarify the mechanism for computing them, let us consider the situation depicted in Figure 2.6 that shows the sequence of wrist positions measured by the Kinect when the operator moves from goal position 1 to goal position 4:

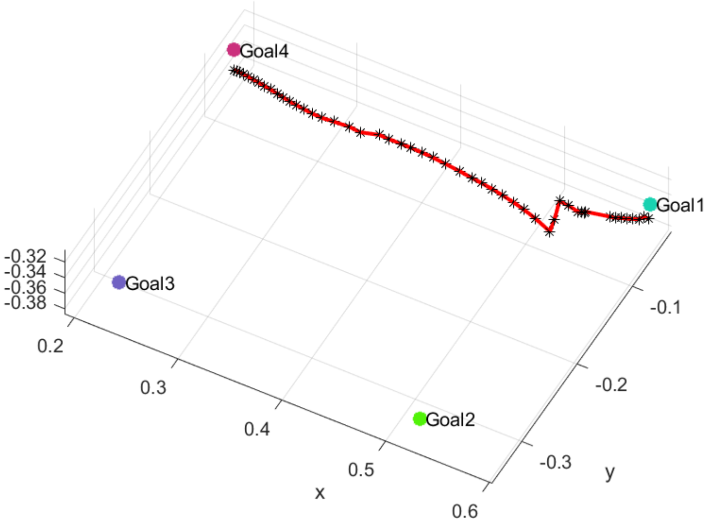


Figure 2.6: An example of the set of the wrist poitions (black stars on the path) returned by the kinect camera when the human hand moves from goal position 1 to goal position 4

Let us assume that the measurements satisfying the spatial threshold have been introduced in the buffer and that it became full. Since these measurements belong to the real curve followed by the human operator when moving from a generic starting position to a prescribed goal position, it is possible to compute the tangent vector to the curve as the partial derivative of a vector position with respect to the spatial coordinate s that parametrizes the curve:

$$\mathbf{t} = \frac{\partial p}{\partial s} \tag{2.8}$$

Then the associated unit tangent vector can be computed as

$$\hat{\mathbf{t}} = \frac{\mathbf{t}}{|\mathbf{t}|} \tag{2.9}$$

The motivation behind this computation is that the tangent vector intrinsically expresses a direction associated with a position. Thus, the tangent vector associated with the set of measured wrist positions can provide a clue where the hand of the operator is directed to and can help inferring the intended goal. In fact, in [34] the path followed by the operator's hand when moving from a certain measured position to a goal positon was modelled by using third-order polynomial.
Then, for each i_{th} target, the value of the angle θ_{W_i} between the unit tangent vector predicted on the ideal path that leads to the center of mass of that goal and the unit measured future tangent vector can be used to determine the probability that the human hand is intended to reach the considered target. Moreover, in that case the likelihood of observing θ_{W_i} under the assumption of reaching the i_{th} target was modelled as a univariate Gaussian distribution:

$$f(\theta_{W_i}|p_{G_i}) \sim \mathscr{N}(\mu, \sigma^2) \tag{2.10}$$

Thus, in order to allow the computation of the so-called 'future tangent vector' it is necessary to split the buffer in two parts. In fact, since it stores the measurements that are ordered according to the FIFO logic previously dismissed, at each iteration, position $\frac{Db}{2}$ can be considered the last position acquired in the past, so it will be denoted from now on as 'previous position', $\mathbf{p_{prev}}$. Then, if the last position of the buffer (Db) is regarded from the perspective of position $\frac{Db}{2}$ it will represent, for each iteration, the future location of the operator's hand after a certain number of samples. Hence it will be denoted as 'future position', $\mathbf{p_{fut}}$ as shown in Figure 2.7.

Hence, at each k_{th} iteration of the algorithm, the 'previous tangent vector' $\hat{\mathbf{t}}^{(k)}_{prev}$ is computed as:

$$\hat{\mathbf{t}}^{(k)}_{prev} = \frac{(p_W^{(\frac{Db}{2})} - p_W^{(1)})}{||p_W^{(\frac{Db}{2})} - p_W^{(1)}||} \tag{2.11}$$

where $p_W^{(1)}$ is the wrist position stored in the first column of the buffer.
And the 'future tangent vector' $\hat{\mathbf{t}}^{(k)}_{fut}$ is computed as:

$$\hat{\mathbf{t}}^{(k)}_{fut} = \frac{(p_W^{(Db)} - p_W^{(\frac{Db}{2})})}{||p_W^{(Db)} - p_W^{(\frac{Db}{2})}||} \tag{2.12}$$

where:

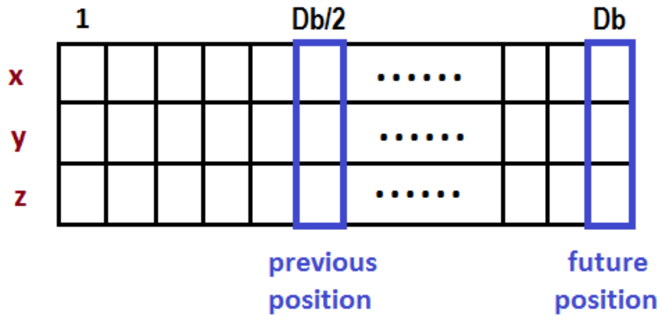


Figure 2.7: column $\frac{Db}{2}$ stores the three-dimensional coordinates of $\mathbf{p_{prev}}$, the point from which the human hand path is supposed to start the reaching motion at each iteration; columns Db contains the three-dimensional coordinates of $\mathbf{p_{fut}}$, the location reached by the human wrist starting from position stored in the $\frac{Db}{2}$-th column of the buffer

- $\hat{\mathbf{t}}^{(k)}_{prev}$ is used as initial condition for computing the cubic polynomial ideal path.
- $\hat{\mathbf{t}}^{(k)}_{fut}$ is exploited for the computation of the angle $\theta^{(k)}_{W_i}$, hence for evaluating the likelihood (refer to equation 2.10) and make inference on the intended goal.

Once the prior probabilities of each goal have been initialized as previuosly mentioned, for each new set of skeletal positions retrieved by the Kinect, the intention inference algorithm takes the following steps:

For each k_{th} sample acquired and for each i_{th} target position:

1. once the buffer is full, the previous position are moved back one step and the newly acquired measure is introduced in the last position as previously mentioned;
2. $\hat{\mathbf{t}}^{(k)}_{prev}$ and $\hat{\mathbf{t}}^{(k)}_{fut}$ are computed according to the mechanism discussed so far;
3. the minimum curvature path (cubic curve) representing the ideal reaching path is evaluated, imposing the following boundary conditions:

$$p(0) = \mathbf{p^{(k)}_{prev}} \qquad p'(0) = \hat{\mathbf{t}}^{(k)}_{prev} \qquad p(1) = p_{G_i}; \tag{2.13}$$

4. the predicted tangent vector $\hat{\mathbf{t}}^{(k)}_{pred_i}$ is evaluated as the spatial derivative of the forecasted position on the minimum curvature path that leads to the i_{th} goal position with respect to the spatial coordinate s.

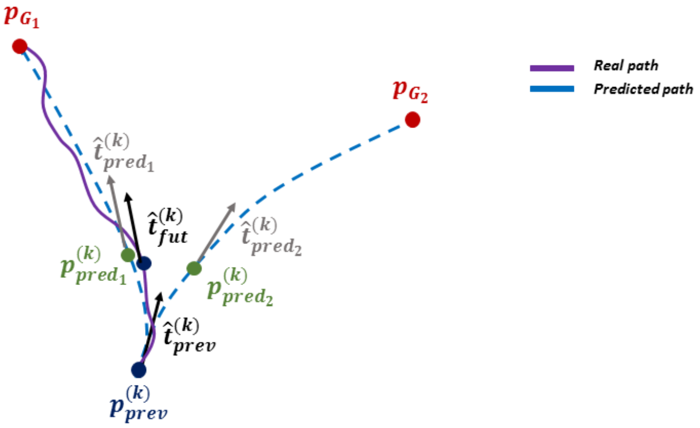


Figure 2.8: Schematic illustration of the meaning of $\hat{\mathbf{t}}^{(k)}_{prev}$, $\hat{\mathbf{t}}^{(k)}_{fut}$, $\hat{\mathbf{t}}^{(k)}_{pred_i}$: the ideal path departing from the sixth position stored in the buffer, $\mathbf{p}^{(\mathbf{k})}_{\mathbf{prev}}$ with initial tangent equal to $\hat{\mathbf{t}}^{(k)}_{prev}$ leading to each target position is computed. Then the predicted tangent vector $\hat{\mathbf{t}}^{(k)}_{pred_i}$ associated with the path leading to the i_{th} target is compared with the measured tangent vector, $\hat{\mathbf{t}}_{fut^{(k)}}$

5. the angle $\theta^{(k)}_{W_i}$ between the predicted unit tangent vector and the measured unit future tangent vectors is computed as:

$$\theta^{(k)}_{W_i} = \arccos((\hat{\mathbf{t}}^{(k)}_{pred_i})^T(\hat{\mathbf{t}}^{(k)}_{fut})) \tag{2.14}$$

as shown in Figure 2.9:

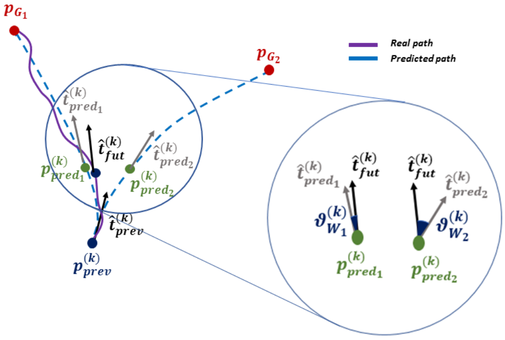


Figure 2.9: For each target, the angle $\theta^{(k)}_{W}$ between the measured future tangent and the predicted one is computed

6. the likelihood is evaluated according to (2.10) and the posterior probability of each goal is updated following the Bayes' rule.

$$f(p_{G_i}|\theta^{(k)}_{W_i}) = P^{(k-1)}(p_{G_i})f(\theta^{(k)}_{W_i}|p_{G_i}) \tag{2.15}$$

hence, the recursive updating rule is:

$$P^{(k)}(p_{G_i}) = P^{(k-1)}(p_{G_i})f(\theta_{W_i}^{(k)}|p_{G_i}) \tag{2.16}$$

7. the value of the posterior probability is normalized only if the product $P^{(k-1)}(p_{G_i})f(\theta_{W_i}|p_{G_i}^{(k)})$ overcomes 1 so that if the operator's hand is directed to another location different from the prescribed target positions, this event can be recognized from a probabilistic point of view.

3

Modelling the ideal human hand reaching path

3.1 Context

The overall problem of human intention inference can be regarded as the interrelation of two subproblems: the first one, as previously explained, is the probabilistic characterization of the human intention; the second one is the search for a model which reproduces quite accurately the actual human hand reaching paths.
The idea behind this issue, in fact, is that if it is possible to know or model appropriately the human hands motion, it could be also easy to define, for each detected hand position, the ideal path connecting a certain measured position to the center of each possible goal. Indeed, in this case, it would be also possible to foresee on all these paths the future hand position and to evaluate, based on the angle between the predicted tangent and the observed one, the most likely reaching target, as shown in Figure 3.1.

Obviously, this aspect could be beneficial not only from the intention's recognition perspective but also from the point of view of efficient work coordination, since it would allow to plan ahead the robot trajectory that synchronizes with the human counterpart's motion.

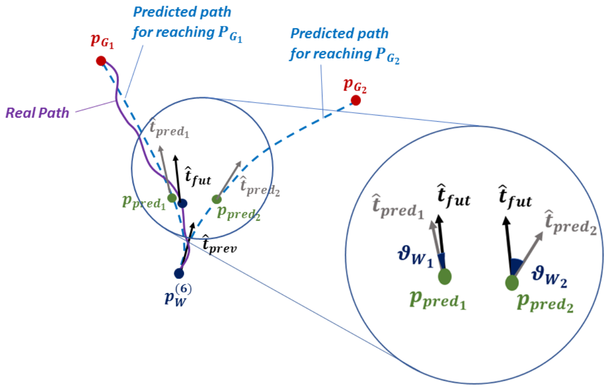


Figure 3.1: For each goal position the predictive hand reaching path starting in correspandance of the measured wrist position and leading to the center of each target can be determined

3.2 Human hand path: state of the art

In the literature there are many discussions that address the problem of determining the ideal human hand reaching motion. However, this issue can be considered mainly solved according to two different approaches:

1. by exploiting a model-based trajectory;
2. by applying a data-driven approach.

Let us focus on the first one.
The idea behind this approach (refer to [16], [29], [40]) is that, when a hand receives the input to move towards a certain target, the problem of trajectory planning and control is solved by the Central Nervous System (CNS) at a higher level, namely, in the task-oriented coordinates. To be more accurate, it is believed that when the human CNS commands the hand to move from its actual position to the desired one, it does so according to a certain criterion that makes it possible to eventually select one specific trajectory among infinite possible choices which could connect the actual position to the intended one. In other words, the CNS chooses the trajectory that optimizes a certain objective function. By the way, since this criterion is unknown, the open question is what this cost function is and how it can be determined.
In literature there are numerous theories related to the definition of the abovementioned cost function, however the features that have been jointly observed are the following: the unconstrained motions between two pairs of targets are approximately straight and characterized by a bell-shaped tangential velocity

profile. Moreover, it has been noticed [16] that this kind of behaviour is maintained, regardless of the workspace size. In this thesis, only the point-to-point motions will be considered of particular interest, since the operator, when cooperating with the robot, repeates a sequence of motions that start from an arbitrary position of the workspace to a target one.

In [16], it is proposed the so-called minimum jerk model. In fact, it has been observed that human movements are commonly smooth and graceful. This aspect seems to indicates that the CNS could be intended to select the smoothest movement possible when moving the hand from a certain equilibrium position to another one. Thus, since the jerk is obtained by deriving the acceleration and intrinsically describes the rate of change associated with acceleration, it seems reasonable to select the trajectory that, minimizing the jerk, maximizes the smoothness.

Consequently, the proposed cost function tends to minimize the time integral of the square of the magnitude of the jerk C_{jerk}, as expressed in equation 3.1:

$$C_{jerk} = \frac{1}{2}\int_0^{t_f}((\frac{d^3x}{dt^3})^2 + (\frac{d^3y}{dt^3})^2)dt \tag{3.1}$$

where t_f is the movement duration and x(t), y(t) are the coordinates of the hand position.

The advantage of this method is that the minimum jerk model is independent from the neuromuscular dynamics. It is only necessary that the movement remains within the capabilities of the neuromuscular system. Hence, some constraints on achievable movements are required.

The intuition about the existence of a unifying rule governing all human hand motions has been disputed, based on the observation that two identical movements are never performed twice. However, the supporters of the minimum jerk model explained that their approach does not force all movements to be exactly the same. Indeed, they motivated the variability in hand's motions because of a slightly change in the perceived location of the positions the hand is heading to. This model also confirms the idea related to an hierarchical organization of motor command: hence, the fact that trajectories are firstly planned at a higher level where the mechanical nature of human actuators is neglected and then, at a lower level, they are converted in terms of torques and forces which are needed to generate that motion.

In [40] an objective function which was related to the physical variables concerning the arm's dynamics was proposed. The resulting criterion function was the one that minimizes C_{torque}:

$$C_{torque} = \frac{1}{2}\int_0^{t_f} (\sum_{i=1}^{n} \frac{dz_i}{dt})^2 dt \tag{3.2}$$

where z_i is the torque generated by the i_{th} out of the n actuators. In fact, according to this model, it is considered that the human hand trajectory is planned and controlled in a way that ensures the minimum variation of the motor torque. Hence, even though those that used this model agreed on the characteristics of the point-to-point motion previously discussed, they disagreed on the fact that hand motions are independent from the dynamics of the musculoskeletal system and the region of the workspace where the movement is performed. The disadvantage of this model was that it is very difficult to deal with since it would ideally require to specify the dynamic equations of the musculoskeletal system. To overcome this difficulty, in [40] the dynamic equations of a two-joint manipulator were exploited and the nonlinear optimization problem was solved by using an iterative learning scheme; while in [41] a neural network model was exploited. However the use of the dynamic equations of a SCARA manipulator allows to model only planar motions.
In [7] it is argued that the trajectory planning is performed at joint level by means of a path planning mechanism. More precisely, under this theory, the arm's movements are generated by the CNS according to the ratio of the tensions between antagonist and agonist muscles. Hence the transition from an equilibrium state to another depends on the muscular contraction time and on the mechanical arm's properties. In [34] the use of a cost function which aimed at minimizing the overall curvature of the reaching path was suggested .

For what concerns the second approach, applied in [32], [25], typically multiple demonstrations of human hand reaching motions were tracked for a certain amount of time so as to collect a sufficiently large training datatset. Then, a library of possible motion trajectories was offline learnt, for instance, using a generic unsupervised classification algorithm, commonly, a gaussian mixture model. Hence, given an observed portion of trajectory, it was possible to foresee its remaining part by exploiting a gaussian mixture regression. In [3] this approach was applied to learn and then predict the trajectories of walking people. In [26] the Inverse Optimal Control was used to learn the cost function that best explains the human motion starting from a number of example trajectories.

Other approaches, as in [24], used a 3D occupancy grid to retrieve the operator's hand position. The approach followed by [10] interpreted the time-varying configuration achieved during reaching movements as the resultant of the composite of attractive and repulsive potential forces acting on the hand. In [20] a

4 degrees of freedom (redundant) arm model capable of generating trajectories in the 3D space was created and the chosen criterion function was the minimum angular jerk. A proper time adjustment was also included for both shoulder and elbow motion so as to reproduce in a correct manner the characteristics of the actual human arm trajectories.

In this Thesis a model-based approach has been preferred due to the possibility of avoiding the bundersome offline training phase and considering the significant advantage that a model-based trajectory could produce: namely, the chance of applying it to any reaching motion performed by the operator, despite the specific collaborative task.
Hence, three different model-based paths will be discussed in order to find the best one capable of reproducing the actual path followed by the human operator, thus enhancing the performance of the intention inference process. It should be recalled that the term 'path' denotes the sequence of positions followed by the human hand in the space. Hence, since in this Thesis the problem of finding a model for the hand reaching path will be addressed, it means that here there is no interest in analyzing the time sequence of the velocities during the movements along the path.

3.3 Minimum curvature path

A first attempt of reproducing the ideal human hand reaching path was the selection of a minimum curvature path, following the approach used by [34]. This choice, in fact, appears to be consistent with the observed human paths that, as previously said, are approximately straight.
Considering that the resulting path is obtained by minimizing the integral of a certain function over a certain temporal or spacial horizon, it is apparent that if a time-based method was used, this would force us to impose a limit on the duration of each reaching motion (see the parameter t_f in the cost function of 3.1 and 3.2). This aspect would not be consistent with the purpose of a collaborative operation. Indeed, we are interested in imposing that during the collaborative task the predictive path lead the human operator to a prescribed position of the space (the target one), no matter the amount of time required to reach the goal. In this way the ideal path would also be consistent with the purpose of a generic operator's reaching motion, which, for sure, at any point,

will be headed to one of the goals.
In this Thesis, the path p will be parametrized with respect to the natural coordinate s where $s \in (0, 1)$, hence:

$$p = p(s) \tag{3.3}$$

As a consequence, the minimum curvature path can be found by minimizing the following objective function $J_{minCurv}$:

$$J_{minCurv} = \int_0^1 p''(s)^T p''(s) ds \tag{3.4}$$

where $p''(s)$ denotes $\frac{\partial^2 p}{\partial s^2}$ the second derivative of the wrist position with respect to coordinate s.

In order to reproduce the constraints of the actual human motion from a generic initial point to the i_{th} target, the following boundary conditions can be introduced:

$$p(0) = p_{prev}^{(k)} \quad p'(0) = \hat{t}_{prev}^{(k)} \quad p(1) = p_{G_i} \tag{3.5}$$

Their meaning is the following:

- at iteration k the path must start in the previously observed wrist position $p_{prev}^{(k)}$. In Chapter 4 it will be explained that at each iteration a buffer will be in charge of storing the left wrist and right wirst position detected at some past instants. $p_{prev}^{(k)}$ will refer to the position that at iteration k was contained in a certain location of buffer which corresponds to a prescribed previous time instant.
- the initial direction of the path will correspond to the previously observed unit tangent vector $\hat{t}_{prev}^{(k)}$. In Chapter 4 the procedure for computing this, relying on the buffer, will be explained.
- the path must end in the considered (i_{th}) target position.

In order to solve the optimization problem, the desired path can be chosen as a $3rd$ order polynomial satisfying the constraints and minimizing $J_{minCurv}(C^{\star})$.

$$p(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3 \tag{3.6}$$

where a_0, a_1, a_2, a_3 are three-dimensional vectors.

Since we are interested in solving a constrained optimization problem: namely, in finding the minimum of a certain function $J_{minCurv}(C^\star)$ subject to equality constraints $g(C^\star) = 0$, a commonly used solution is the application of the method of Lagrange multipliers. This procedure allows to solve the optimization problem by minimizing the so-called Lagrangian expression:

$$J^\star_{minCurv}(C^\star, \lambda) = J_{minCurv}(C^\star) - \lambda g(C^\star) \tag{3.7}$$

where $C^\star$ is the matrix of coefficients that have to be found:

$$C^\star = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{3.8}$$

Therefore, the steps performed for solving the minimization problem are:

1. compute the partial derivative of $p(s)$ with respect to s.

$$p'(s) = a_1 s + 2a_2 s + 3a_3 s^2 \tag{3.9}$$

and the corresponding second derivative:

$$p''(s) = 2a_2 + 6a_3 s \tag{3.10}$$

2. substituting the initial conditions on $p(0)$ and $p'(0)$, retrieve the trivial expressions of the coefficients a_0 and a_1:

$$a_0 = p^{(k)}_{prev} \qquad a_1 = \hat{t}^{(k)}_{prev} \tag{3.11}$$

3. re-write $J_{minCurv}$ substituting in $p''(s)$ the expression of the coefficients of 3.11:

$$J_{minCurv} = \int_0^1 \left(\begin{bmatrix} 2a_2 & 6a_3 \end{bmatrix} \begin{bmatrix} 1 \\ s \end{bmatrix} \begin{bmatrix} 1 & s \end{bmatrix} \begin{bmatrix} 2a_2 \\ 6a_3 \end{bmatrix} \right) ds \tag{3.12}$$

Thus, denoting with:

$$C = \begin{bmatrix} 2a_2 \\ 6a_3 \end{bmatrix} \tag{3.13}$$

and:

$$\beta = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} \end{bmatrix} \tag{3.14}$$

equation 3.12 can be re-written as:

$$J_{minCurv} = C^T \beta C \tag{3.15}$$

4. substituting both the initial the final condition $p(1) = p_{G_i}$ and the expression of a_0 and a_1 in the expression of $p(s)$, obtain the so-called 'homogeneous constraint equation':

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \end{bmatrix} - (p_{G_i} - p^{(k)}_{prev} - \hat{t}^{(k)}_{prev}) = 0 \tag{3.16}$$

   let us name E the row vector $[1 \quad 1]$ and α_2 the expression $p_{G_i} - p^{(k)}_{prev} - \hat{t}^{(k)}_{prev}$.

5. write the expression of $J^{\star}_{minCurv}(C^{\star}, \lambda)$:

$$J^{\star}_{minCurv}(C^{\star}, \lambda) = C^T \beta C + \lambda^T (EC^{\star} - \alpha_2) \tag{3.17}$$

   where:

$$C = \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix} C^{\star} \tag{3.18}$$

6. subistituting the expressions of C, β, E and $J^{\star}_{minCurv}(C^{\star}, \lambda)$ in 3.17, obtain the final equation:

$$J^{\star}_{minCurv}(C^{\star}, \lambda) = 4a_2^2 + 12a_2a_3 + 12a_3^2 + \lambda^T (a_2 + a_3 - p_{G_i} + p^{(k)}_{prev} + \hat{t}^{(k)}_{prev}) \tag{3.19}$$

7. in order to find the stationary points of this function and retrieve the expression of coefficients a_2 and a_3, solve the following system of equations:

$$\begin{cases} \frac{\partial(J^{\star}(a_2, a_3, \lambda))}{\partial a_2} = 0 \\ \frac{\partial(J^{\star}(a_2, a_3, \lambda))}{\partial a_3} = 0 \\ (\frac{\partial(J^{\star}(a_2, a_3, \lambda))}{\partial \lambda})^T = 0 \end{cases} \tag{3.20}$$

   In conclusion, the coefficients of the cubic minimum curvature path are:

$$a_0 = p^{(k)}_{prev} \tag{3.21}$$

$$a_1 = \hat{t}^{(k)}_{prev} \tag{3.22}$$

$$a_2 = \frac{3}{2} p_{G_i} - \frac{3}{2} \hat{t}^{(k)}_{prev} - \frac{3}{2} p^{(k)}_{prev} \tag{3.23}$$

$$a_3 = -\frac{1}{2} p_{G_i} + \frac{1}{2} \hat{t}^{(k)}_{prev} + \frac{1}{2} p^{(k)}_{prev} \tag{3.24}$$

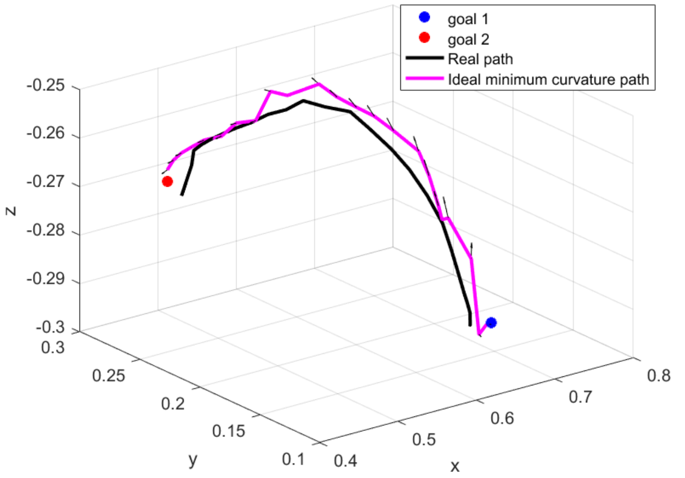


Figure 3.2: The minimum curvature path (magenta) that connects goal 1 to goal 2 is shown together with the set of predicted tangent vectors (black arrows on the magenta path) obtained at each iteration. The overall path represented in magenta is composed by the sequence of predicted segments obtained at each k_{th} iteration based on the measured $p_{prev}^{(k)}$ and $\hat{t}_{prev}^{(k)}$

In Figure 3.2 the obtained minimum curvature path is compared with corresponding one retrieved by means of the Kinect camera, for an observed reaching movement.

As can be easily noticed by observing the figure, the predictive path reproduces quite well the measured one. However, a number of discontinuities are visible on the predictive path.

This aspect can be explained as follows: let us assume that at generic iteration k the algorithm that is in charge of computing the predictive path, receives the initial measured position $p_{prev}^{(k)}$, the i_{th} target position p_{G_i}, and the spatial displacement at which it has to evaluate the next position along the minimun jerk path. Hence, at iteration k, the algorithm will compute the minimum jerk path $p(s)$ that starts in $p_{prev}^{(k)}$ and ends in p_{G_i}, by imposing that $p(0) = p_{prev}^{(k)}$, $p'(0) = \hat{t}_{prev}^{(k)}$ and $p(1) = p_{G_i}$.

Notice that the final condition imposed at each k_{th} iteration is always $p(1) = p_{G_i}$ and not $p(1 - \Delta s^{(k \to k+1)}) = p_{G_i}$, where $\Delta s^{(k \to k+1)}$ represents the spatial displacement between $p_{prev}^{(k)}$ and $p_{prev}^{(k+1)}$.

At the new iteration $k+1$ the minimum jerk path must be computed again according to the same criterion as before: however this time it is imposed to find the path $p(s)$ that starts in a forward position of the space, $p_{prev}^{(k+1)}$, and

ends in the same target position as before ,p_{G_i}, by imposing that $p(0) = p_{prev}^{(k+1)}$, $p'(0) = \hat{t}_{prev}^{(k+1)}$ and $p(1) = p_{G_i}$, as previously explained.
Consequently, when computing the path at time instants $k+1$ it is not taken into account the previous history of path.
This aspect is consistent with the Markovian hypotheis expressed in section 2.3, namely, the fact that the new observed state dependes only on the previously observed one, but has the disadvantage of producing a sequence of path segments that are discontinuous.

In addition, it could be argued that the choice of a path that attempts at minimizing the overal curvature af a reaching motion could result, to some extent, in contrast with the observation that a slight curvature on human reaching path should be always accounted for, due to the presence of the shoulder that inherently provides a sort of pivot for arm's motions [38].
This is the reason why in this Thesis an attempt of creating a path having certain characteristics has been made. In fact the aim was to create a path that on the one hand was capable of satisfying the observed human path features and on the other hand took into consideration the impact of the observed shoulder position in determining the shape of the actual human path. That aspect will be addressed in the following section.

3.4 Taking into account the position of the center of the shoulder

The aim of this section is to present a new path formulation which takes into account the shoulder position. The purpose is to investigate whether and how it is possible to include this information in order to better reproduce the human behaviour in terms of reaching motions. Let us consider that the following observations are available:

- initial position $p_{prev}^{(k)}$:

$$p_{prev^{(k)}} = \begin{bmatrix} p_{prev_x}^{(k)} \\ p_{prev_y}^{(k)} \\ p_{prev_z}^{(k)} \end{bmatrix} \tag{3.25}$$

- initial tangent $\hat{t}_{prev}^{(k)}$:

$$\hat{t}_{prev}^{(k)} = \begin{bmatrix} \hat{t}_{prev_x}^{(k)} \\ \hat{t}_{prev_y}^{(k)} \\ \hat{t}_{prev_z}^{(k)} \end{bmatrix} \tag{3.26}$$

- final position which corresponds to the i_{th} target position p_{G_i}:

$$p_{G_i} = \begin{bmatrix} p_{G_{i_x}} \\ p_{G_{i_y}} \\ p_{G_{i_z}} \end{bmatrix} \tag{3.27}$$

- position of the center of the shoulder $p_s^{(k)}$:

$$p_s^{(k)} = \begin{bmatrix} p_{s_x}^{(k)} \\ p_{s_y}^{(k)} \\ p_{s_z}^{(k)} \end{bmatrix} \tag{3.28}$$

Therefore, in order to obtain a slightly curved path that reproduces as much as possible the features of the observed ones (refer to [38]), it can be initially considered the ideal circumference which is centred in shoulder position, has radius equal to the Euclidean distance between $p_s^{(k)}$ and $p_{prev}^{(k)}$ and lays on the plane which contains $p_{prev}^{(k)}$, $p_s^{(k)}$ and p_{G_i}, as shown in Figure 3.3. In this way the shoulder can be interpreted as a sort of pivot for the reaching motion.

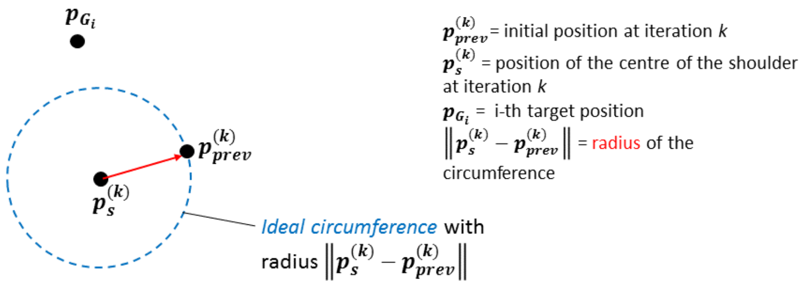

$\boldsymbol{p}_{G_i}$
$\boldsymbol{p}_{prev}^{(k)}$ = initial position at iteration k
$\boldsymbol{p}_s^{(k)}$ = position of the centre of the shoulder at iteration k
$\boldsymbol{p}_{G_i}$ = i-th target position
$\left\|\boldsymbol{p}_s^{(k)} - \boldsymbol{p}_{prev}^{(k)}\right\|$ = radius of the circumference
$\boldsymbol{p}_{prev}^{(k)}$
$\boldsymbol{p}_s^{(k)}$
Ideal circumference with radius $\left\|\boldsymbol{p}_s^{(k)} - \boldsymbol{p}_{prev}^{(k)}\right\|$

Figure 3.3: Ideal circumference centred in the shoulder position and having radius equal to the distance between $p_s^{(k)}$ and $p_{prev}^{(k)}$

Hovewer it must be also ensured that the path ends in p_{G_i} since it is the objective of the operator reaching motion. In order to take simultaneously into considerations all these requirements, it is necessary to find a path that on the one hand keeps as much as possible close to the curve describing the ideal circumference and on the other hand fulfills the constraints related to the initial position, initial tangent and final position. To this end, let us first consider a generic 4-th order polynomial:

$$p(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3 + a_4 s^4 \tag{3.29}$$

In order to obtain the desired path, this polynomial must comply with the following initial conditions:

$$p(0) = p_{prev}^{(k)} \quad p'(0) = \hat{t}_{prev}^{(k)} \tag{3.30}$$

and final conditions:

$$p(1) = p_{G_i} \quad p'(1) = 0 \tag{3.31}$$

Thus, substituting conditions 3.30 in 3.29 one obtains the trivial expressions of a_0 and a_1:

$$a_0 = p_{prev}^{(k)} \qquad a_1 = \hat{t}_{prev}^{(k)} \tag{3.32}$$

The schematic illustration of the abovementioned path and circumference is shown in Figure 3.4:

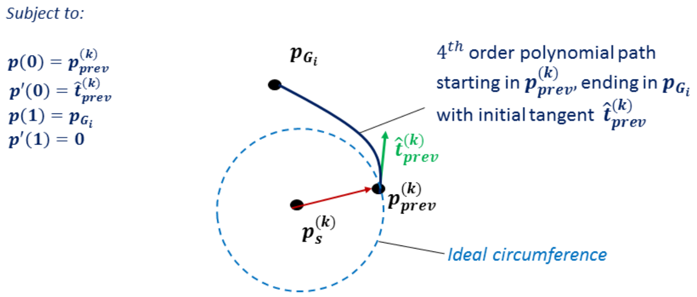

Subject to:
$p(0) = p_{prev}^{(k)}$
$p'(0) = \hat{t}_{prev}^{(k)}$
$p(1) = p_{G_i}$
$p'(1) = 0$
p_{G_i}
4^{th} order polynomial path starting in $p_{prev}^{(k)}$, ending in p_{G_i} with initial tangent $\hat{t}_{prev}^{(k)}$
$\hat{t}_{prev}^{(k)}$
$p_{prev}^{(k)}$
$p_s^{(k)}$
Ideal circumference

Figure 3.4: The forth order polynomial path that fulfills the constraints on initial and final position can be determined

Assuming that the expression of the circumference centred in the shoulder position and having radius equal to $||p_s^{(k)} - p_{prev}^{(k)}||$ has been determined, let us consider N_c intermediate points laying on this curve, as shown in Figure 3.5 where σ_j be the j_{th} intermediate point.

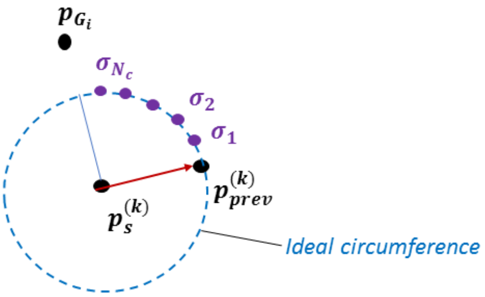


Figure 3.5: Equally spaced intermediate points laying on the ideal circumference centred in the shoulder position

Let us also consider the corresponding equally spaced intermediate points laying on the polynomial path, as illustrated in Figure 3.6:
where $p(\sigma_j)$ represents the j_{th} intermediate point laying on this path.

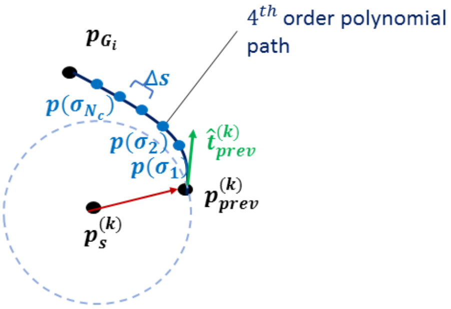


Figure 3.6: Equally spaced intermediate points laying on the 4-th order polynomial path

Hence, the objective function that is capable of generating the desired path could be the one that minimizes the sum of the distances between each j_{th} intermediate point laying on the circumference and the corresponding intermediate point laying on the 4-th order polynomial path.
The minimization problem can be expressed as minimizing J_{distCS}:

$$J_{distCS} = \sum_{j=1}^{N_c} ((p(\sigma_j) - \sigma_j)^T (p(\sigma_j) - \sigma_j)) \tag{3.33}$$

subject to:

$$p(1) = p_{G_i} \qquad p'(1) = 0 \tag{3.34}$$

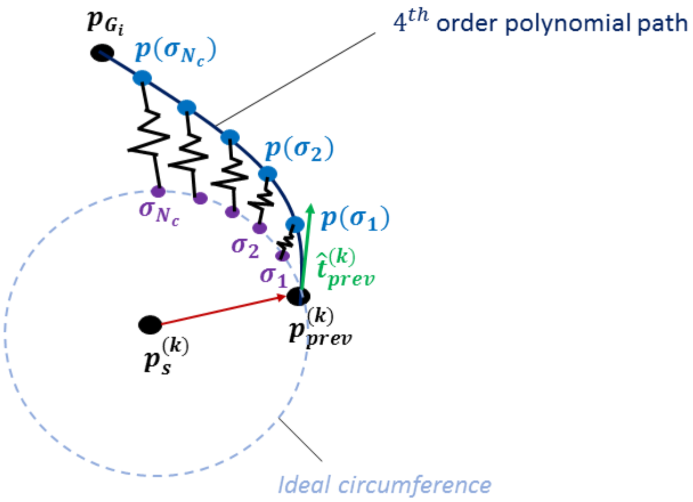


Figure 3.7: The minimization problem can be equivalently reformulated from a mechanical point of view as the minimization of the sum of the displacements of Nc springs which link each σ_j to each $p(\sigma_j)$

In order to solve the minimization problem, the procedure is the same applied in section 3.3 for finding the minimum curvature path. Thus:

1. substitute in equation 3.29 the coefficients of equation 3.30; hence $p(s)$ can be re-written as:

$$p(s) = a_0 + t^{(k)}_{prev} s + w(s)C^\star \tag{3.35}$$

   where

$$w(s) = \begin{bmatrix} s^2 & s^3 & s^4 \end{bmatrix} \qquad C^\star = \begin{bmatrix} a_2 \\ a_3 \\ a_4 \end{bmatrix} \tag{3.36}$$

2. rewrite expression 3.33 substituting to the abovementioned quantities. After some computation the minimization problem can be expressed as:

$$J_{distCS} = {C^\star}^T \beta C^\star + {C^\star}^T \alpha \tag{3.37}$$

   where:

   - $\beta = \sum_{j=1}^{N_c} w^T(\sigma_j) w(\sigma_j)$.

- $\alpha = 2\sum_{j=1}^{N_c} w^T(\sigma_j)(p_{prev}^{(k)} + \hat{t}_{prev}^{(k)}\sigma_j - \sigma_j)$; These quantities can be evaluated once the intermediate points have been found out.

3. write the Lagrangian expression:

$$J^\star(C^\star, \lambda) = {C^\star}^T \beta C^\star + {C^\star}^T \alpha + \lambda^T(EC^\star - \alpha_2) \tag{3.38}$$

where:

$$E = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix} \qquad \alpha_2 = \begin{bmatrix} p_{G_i} - p_{prev}^{(k)} - \hat{t}_{prev}^{(k)} \\ -\hat{t}_{prev}^{(k)} \end{bmatrix} \tag{3.39}$$

4. in order to find the stationary points of $J^\star(C^\star, \lambda)$ and retrieve the expression of coefficients a_2 and a_3, solve the following system of equations:

$$\begin{cases} (\frac{\partial(J^\star(C^\star,\lambda))}{\partial C^\star})^T = 0 \\ (\frac{\partial(J^\star(C^\star,\lambda))}{\partial \lambda})^T = 0 \end{cases} \tag{3.40}$$

5. therefore, in order to obtain the coefficient of matrix $C^\star$, the following system must be solved:

$$\begin{bmatrix} 2\beta & E^T \\ E & 0 \end{bmatrix} \begin{bmatrix} C^\star \\ \lambda \end{bmatrix} = \begin{bmatrix} -\alpha \\ \alpha_2 \end{bmatrix} \tag{3.41}$$

6. Let us name A and γ the following matrices:

$$A = \begin{bmatrix} 2\beta & E^T \\ E & 0 \end{bmatrix} \qquad \gamma = \begin{bmatrix} -\alpha \\ \alpha_2 \end{bmatrix} \tag{3.42}$$

Consequently, the coefficient matrix $C*$ can be obtained by first computing matrix M:

$$M = (A)^{-1}\gamma \tag{3.43}$$

where M is a 5x5 matrix. Therefore, the matrix of coefficient $C^\star$ can be easily obtained as the 3x5 submatrix of M. It should be noticed that γ (see equation 3.43) only depends on the values of $p_{prev}^{(k)}$, $\hat{t}_{prev}^{(k)}$ and p_{G_i}, while A is affected by the number and the values of the intermediate points.

Given the above, the problem becomes now how to determine them. More precisely, it must be defined the expression of the ideal circumference these points belongs to.

Let us firstly set a reference sistem, whose unitary axes are defined as:

$$\mathbf{x} = \frac{(p_{prev}^{(k)} - p_s^{(k)})}{||p_{prev}^{(k)} - p_s^{(k)}||}; \tag{3.44}$$

$$\mathbf{z} = \frac{(p_{prev}^{(k)} - p_s^{(k)}) \wedge (p_{G_i} - p_s^{(k)})}{||(p_{prev}^{(k)} - p_s^{(k)}) \wedge (p_{G_i} - p_s^{(k)})||}; \tag{3.45}$$

where vector $\mathbf{z}$ is orthogonal to the plane that contains $p_{prev}^{(k)}$, $p_s^{(k)}$ and p_{G_i} and points outward. And:

$$\mathbf{y} = \frac{\mathbf{z} \wedge \mathbf{x}}{||(\mathbf{z} \wedge \mathbf{x})||}; \tag{3.46}$$

so as to obtain a reference system consistent with the right-hand rule. Let us also define the rotation matrix R_1^0 that expresses the rotation of frame 1 with respect to frame 0 and the vector $p_s^{(k)}$ that expresses the shoulder position with respect to frame 0:

$$R_1^0 = \left[\mathbf{x}|\mathbf{y}|\mathbf{z}\right] \tag{3.47}$$

as shown in Figure 3.8:

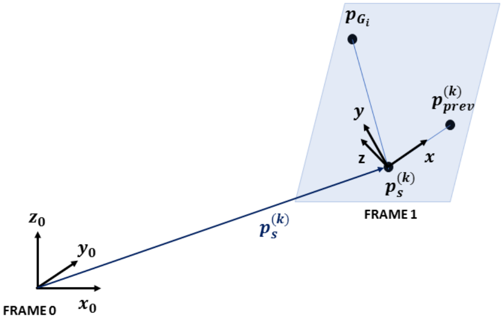


Figure 3.8: The reference system associated with the plane that contains $p_s^{(k)}$, $p_{prev}^{(k)}$ and $p_{G_i}^{(k)}$ is determined in order to find the ideal circumference laying on that plane

With respect to frame 1 the circumference having radius r equal to $||p_{prev}^{(k)} - p_s^{(k)}||$ can be depicted as illustrated in Figure 3.9:

Thus, with respect to frame 1 the angle θ_0 that corresponds to the arc of circumference which starts in $p_{prev}^{(k)}$ and ends in the intersection point between the whole circle and the line $p_{G_i} - p_s^{(k)}$ can be computed as follows:

$$\theta_0 = \text{atan2}(p_{G_{iy}}, p_{G_{ix}}) \tag{3.48}$$

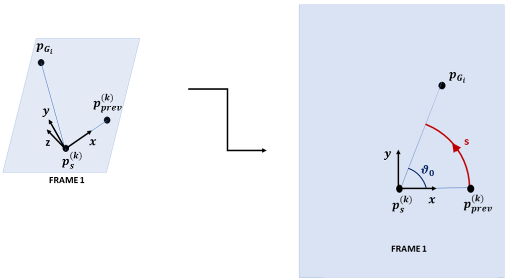


Figure 3.9: The expression of circumference that is parametrized with respect to the natural coordinate s and having radius r can be determined with respect to frame 1.

while, with respect to frame 0 expression 3.48 becomes:

$$\theta_0 = \text{atan2}(\mathbf{y}^T(p_{G_{iy}} - p_s^{(k)}), \mathbf{x}^T(p_{G_{ix}} - p_s^{(k)})) \tag{3.49}$$

Considering the meaning of the natural coordinate s that describes the length of the considered arc of circumference, the circumference Cfr that lays on the plane associated with frame 1 can be expressed with respect to frame 0 and as a function of $s \in (0,1)$ according to the following way:

$$Cfr(s) = p_s^{(k)} + R_1^0(\begin{bmatrix} r\cos(s\theta_0) & r\cos(s\theta_0) & 0 \end{bmatrix}) \tag{3.50}$$

Hence, once selected the number of intermediate points, the desired path is completely determined. In Figure is represented the predictive path obtained by minimizing J_{distCS}. This path, represented in magenta is compared with the corresponding real path retrieved by using the Kinect camera.

Even in this circumstance, the same considerations made for the previous predictive path holds; namely, the path shows some discontinuities because of the same motivations expressed before.
The advantage of this formulation is that, since it does not attempt at minimizing the curvature, it returns a slightly curved path even in the case where the initial position and final position are aligned. It should be recalled that, obviously, the initial curvature of the path is also affected by the direction of the initial tangent vector that is one of the boundary condition imposed before solving the minimization problem.

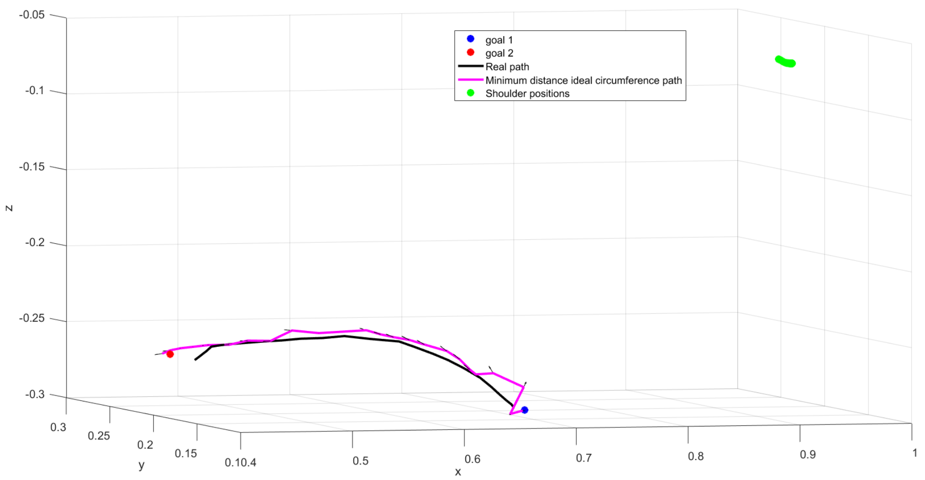


Figure 3.10: The path obtained by minimizing J_{distCS} is shown in magenta together with the predicted tangent vectors (black arrows laying on the magenta curve). The corresponding actual path followed by the human operator is the black curve.

3.5 Parabola

In order to solve the problem of the aforementioned discontinuities while preserving the curvature of the path, a further attempt was made by using a second order degree polynomial, a parabolic path, exploiting the possibility of keeping fixed its axis of symmetry, as it will be explained afterwards.
The parabola belongs to the family of conic sections. As a matter of fact, it is obtained by intersecting the surface of a cone with a plane that is parallel to one and only one generating line of the cone.
Being a planar and symmetric curve, another crucial element of a parabola is its axis of symmetry.
Let us consider the traditional equation of the parabola whose axis of symmetry is parallel to the Cartesian y-axis. This is defined by the following second order equation:

$$y = ax^2 + bx + c \tag{3.51}$$

It may seem that the parabola expressed in equation 3.51 is completely determined, for instance, by imposing the passage through two points and the tangent vector associated with one of the two. This is not sufficient: in fact, in order to guarantee that only one parabola satisfying the previous conditions

exists, also the reference system, with respect to which the axis of symmetry is parallel must be preliminarly chosen. Here, indeed, we assume that the parabolic path that we want to determine always has its axis of symmetry parallel to the y-axis of the selected reference frame.
In fact, if the reference system was not a priori determined before imposing the abovementioned conditions, it would be always possible to find an infinite number of parabolas that fulfill these conditions, as shown in Figure 3.11:

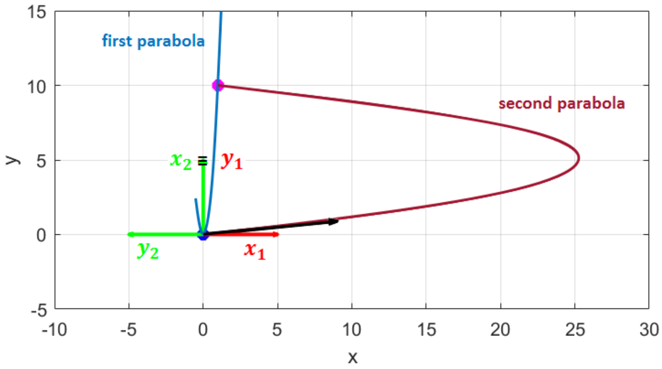


Figure 3.11: In this picture it is illustated the parabolas (blue and purple) obtained by imposing the passage through the same two points (blu and magenta) and the same tangent (black arrow), however the reference systems with respect to which they are computed are different, respectively x1-y1 and x2-y2. Hence two different curves are obtained

Let us now consider the parabolic curve γ_1 defined by the second order polynomial expressed as a function of the usual natural coordinate s, so $\gamma_1 = \gamma_1(s)$. Hence:

$$\gamma_1(s) = \begin{bmatrix} a_{0_x} + a_{1_x}s + a_{2_x}s^2 \\ a_{0_y} + a_{1_y}s + a_{2_y}s^2 \\ a_{0_z} + a_{1_z}s + a_{2_z}s^2 \end{bmatrix} = \begin{bmatrix} a_{0_x} \\ a_{0_y} \\ a_{0_z} \end{bmatrix} + \begin{bmatrix} a_{1_x} \\ a_{1_y} \\ a_{1_z} \end{bmatrix} s + \begin{bmatrix} a_{2_x} \\ a_{2_y} \\ a_{2_z} \end{bmatrix} s^2 \tag{3.52}$$

imposing the following boundary conditions:

$$\gamma_1(0) = p_{prev}^{(k)} \tag{3.53}$$

$$\gamma_1'(0) = \hat{t}_{prev}^{(k)} \tag{3.54}$$

$$\gamma_1(1) = p_{G_i} \tag{3.55}$$

the expressions of the coefficients can be easily obtained as:

$$a_0 = p_{prev}^{(k)} \tag{3.56}$$

$$a_1 = \hat{t}_{prev}^{(k)} \tag{3.57}$$

$$a_2 = p_{G_i} - p_{prev}^{(k)} - \hat{t}_{prev}^{(k)} \tag{3.58}$$

This curve can be considered expressed with respect to a certain reference system which will be named 'frame 1'.
As can be noticed by observing the expression of the parametric curve γ_1, since the matrix of coefficients a_1, a_2 and a_3 is full with respect to the x, y, z directions, it is not trivial to determine the orientation of the symmetry axis, as it was done in equation 3.51.
The reason why we are interested in the direction of the axis of symmetry is that we want to guarantee that all the segments of parabolic paths which are iteratively returned by the intention inference algorithm belong exactly to the same parabolic path that was computed at the first iteration (based on $p_{prev}^{(1)}$, $\hat{t}_{prev}^{(1)}$ and p_{G_i}), avoiding discontinuity problems.
The method which makes it possible to achieve this result is to keep fixed the direction of the axis of symmetry obtained for the parabolic path computed at the first iteration (iteration 1).
Hence, only at iteration 1 the axis of symmetry will be computed; then, for all the successive iterations $(2 \geq k)$ the previous axis of symmetry will be iteratively projected in the plane which contains the newly acquired initial point, initial tangent and the same target position $p_{prev}^{(k)}$, $\hat{t}_{prev}^{(k)}$ and p_{G_i}, as it will be explained afterwards.

More details about the computation of the symmetry axis will be here provided. It can be observed that it is always possible to find the reference system whose y-axis is parallel to the unknown axis of symmetry associated with a parabola γ_1. The problem is how to determine the aforementioned frame. Let us write the expression of the parabolic curve assuming that its axis of symmetry is parallel to the y-axis of a certain reference system which from now on will be denoted as 'frame 0'. With respect to frame 0, the parabolic curve, named γ_0 can be written as:

$$\gamma_0 = \begin{bmatrix} x \\ ax^2 + bx + c \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ c \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ b \\ 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ a \\ 0 \end{bmatrix} x^2 \tag{3.59}$$

Thus, it is possible to write the transformation which links γ_1 and γ_0 as follows:

$$\gamma_0 = T_1^0 + R_1^0\gamma_1 \tag{3.60}$$

or, more compactly,

$$\begin{bmatrix}\gamma_0\\1\end{bmatrix} = \begin{bmatrix}R_1^0 & T_1^0\\0 & 1\end{bmatrix}\begin{bmatrix}\gamma_1\\1\end{bmatrix} \tag{3.61}$$

as shown in Figure 3.12:

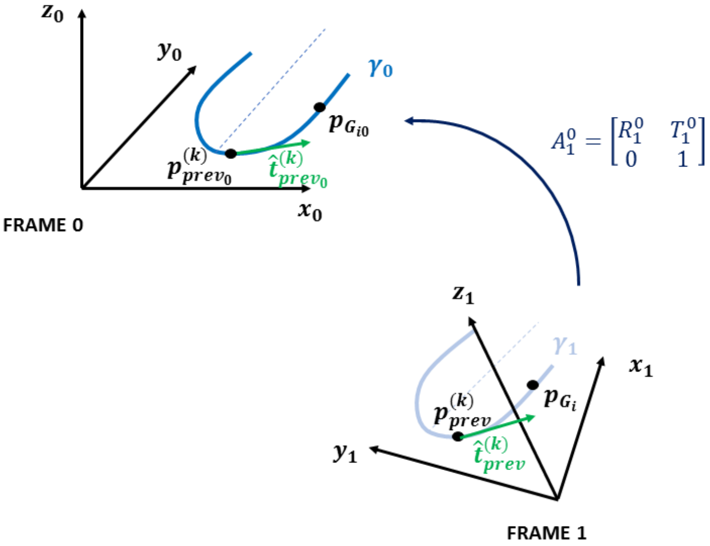


Figure 3.12: The homogeneous transformation matrix allows to link the expressions of γ_0 and γ_1

Let us write the matrix R_1^0 as:

$$R_1^0 = \left[\mathbf{x_0}|\mathbf{y_0}|\mathbf{z_0}\right] \tag{3.62}$$

hence, substituting in 3.60 equation 3.62 and 3.59 it results:

$$\gamma_0 = \left[\mathbf{x_0}|\mathbf{y_0}|\mathbf{z_0}\right]\left(\begin{bmatrix}0\\c\\0\end{bmatrix} + \begin{bmatrix}1\\b\\0\end{bmatrix}x + \begin{bmatrix}0\\a\\0\end{bmatrix}x^2\right) + T_1^0 \tag{3.63}$$

therefore, group the term according to the power of x:

$$\gamma_0 = (c\mathbf{y_0} + T_1^0) + (\mathbf{x_0} + b\mathbf{y_0})x + a\mathbf{y_0}x^2 \tag{3.64}$$

In view of the considerations expressed at the beginning of this section, a parabola defined by an equation of the type 3.51 has, by definition, an axis

of symmetry parallel to the y-axis of the reference frame with respect to which it is defined. Consequently, observing the expression of γ_0 in 3.64, it clear that the axis of symmetry of γ_0 is parallel to $\mathbf{y_0}$ which, in fact, is the y-axis of frame 0 (see Figure 3.12).
Since γ_0 can be also expressed as in 3.52 the following equivalence holds:

$$a\mathbf{y_0} = \begin{bmatrix} a_{2x} \\ a_{2y} \\ a_{2z} \end{bmatrix} \tag{3.65}$$

where the value of coefficient a_2 was already determined by equation 3.58 by imposing the boundary conditions.
In conclusion it has been demonstrated that the direction of the axis of symmetry of the desired parabola corresponds to the one of the three-dimensional coefficient a_2.

In order to evaluate at each iteration k the parabola that fulfills the boundary conditions previously explained, it is firstly defined the reference system associated with the plane containing $p^{(k)}_{prev}$, $\hat{t}^{(k)}_{prev}$ and p_{G_i}, thus $\mathbf{x_0}$, $\mathbf{y_0}$, $\mathbf{z_0}$.

- $\mathbf{y_0}$ correspond to the normalized vector obtained in equation 3.65.
- $\mathbf{z_0}$ is obtained by normalizing the cross product $\hat{t}^{(k)}_{prev} \wedge (p_{G_i} - p^{(k)}_{prev})$;
- $\mathbf{x_0}$ is obtained by normalizing $\mathbf{y_0} \wedge \mathbf{z_0}$, so as to comply with the right-hand rule.

Then, in order to find the coefficient of the parabola expressed with respect to frame 0, the coordinates of $p^{(k)}_{prev}$, $\hat{t}^{(k)}_{prev}$ and p_{G_i} that belong to frame 1 are expressed with respect to frame 0 according to the following homogeneous transformation matrix:

$$M^0_1 = \begin{bmatrix} \mathbf{x_0} & \mathbf{y_0} & \mathbf{z_0} & \mathbf{p^{(k)}_{prev}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.66}$$

As a consequence:

$$p^{(k)}_{prev_0} = (M^0_1)^{-1} p^{(k)}_{prev} \tag{3.67}$$

where $p^{(k)}_{prev_0}$ defined the coordinate of point $p^{(k)}_{prev}$ with respect to frame 0. The same holds to obtain $p_{G_{i0}}$ and $\hat{t}^{(k)}_{prev_0}$. Thus, imposing the passage through $p^{(k)}_{prev_0}$ and $p_{G_{i0}}$ and that the parabola is tangent to $\hat{t}^{(k)}_{prev_0}$ (this is eqivalent to imposing that the discriminant is equal to zero), the following coefficients are obtained:

$$\begin{cases} a = \left(p_{Gy_{i0}} - \frac{\hat{t}^{(k)}_{yprev_0}}{\hat{t}^{(k)}_{xprev_0}} p_{Gx_{i0}} \right) \frac{1}{p^2_{Gx_{i0}}} \\ b = 0 \\ c = 0 \end{cases} \tag{3.68}$$

Let us now assume that at time step k $\mathbf{y_0^{(k)}}$ has been determined . This is the y-axis of the frame 0 which is parallel to the axis of simmetry of the parabola laying on the plane which contains $p^{(k)}_{prev}$, p_{G_i} and $\hat{t}^{(k)}_{prev}$.
Let us now assume that at iteration $k+1$ the a new initial point and intial tangent arrives. Let us denote them with $p^{(k+1)}_{prev}$ and $\hat{t}^{(k+1)}_{prev}$. Since the direction of the y-axis of frame 0, hence, the direction of axis of symmetry, is equal to coefficient a_2 which (refers to equation 3.58) depends on the value of $p^{(k+1)}_{prev}$, $\hat{t}^{(k+1)}_{prev}$ and p_{G_i}, it could happen that the new axis $\mathbf{y_0^{(k+1)}}$ does not lay anymore on the same plane as before. As a consequence, in order to try to avoid that the segment of parabolic curve computed at time step $k+1$ belongs to a completely different parabola with respect to the one obtained at iteration k and, at the same time, complying with the boundary conditions, $\mathbf{y_0^{(k)}}$ is projected on the new plane identified by $p^{(k+1)}_{prev}$, $\hat{t}^{(k+1)}_{prev}$ and p_{G_i}.

Thus, the new $\mathbf{z_0^{(k+1)}}$ is computed as before normalizing the cross product $\hat{t}^{(k+1)}_{prev} \wedge (p_{G_i} - p^{(k+1)}_{prev})$.
Finally, the projected $\mathbf{y_0}$ at iteration $k+1$, $\mathbf{y^{(k+1)}_{0_{proj}}}$ is computed as:

$$\mathbf{y^{(k+1)}_{0_{proj}}} = \mathbf{y_0^{(k+1)}} - (\mathbf{y_0^{(k+1)}} \mathbf{z_0^{(k+1)}}) \mathbf{z_0^{(k+1)}} \tag{3.69}$$

$\mathbf{x_0^{(k+1)}}$ is computed, as before, so as to comply with the right-hand rule.
As it is quickly noticeable by observing Figure 3.13, the predictive path is pretty similar to the measured one and the number of discontinuities that characterize this parabolic path is reduced, thanks to the adopted method which consists in recursively projecting the axis of symmetry of the parabola in order to keep fixed its direction as much as possible. It can also be noticed that, since the parabolic segments returned at each iteration belong to approximately the parabolic path computed at the initial time step, it reproduces quite well the characteristics of the actual path followed by the human operator.

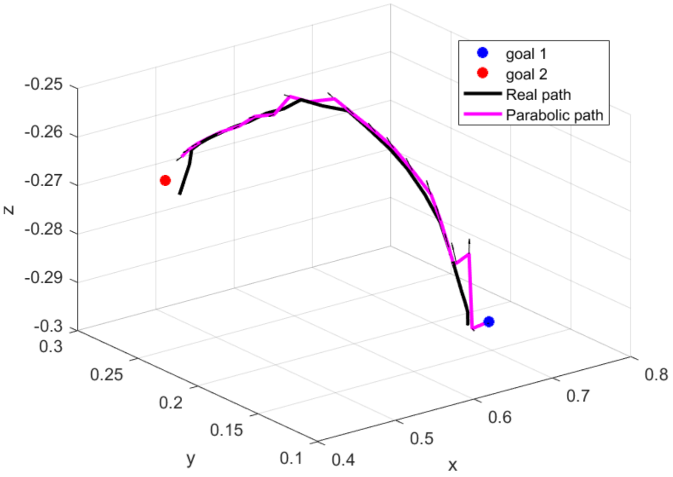


Figure 3.13: The parabolic path (magenta) that connects goal 1 to goal 2 is represented in this figure along with the predicted tangent vectors (black arrows on the magenta path)

3.6 Validation of the best path: the Fréchet Distance criterion

An overview of all the types of predictive paths is illustrated in Figure 3.14 along with a measured path:

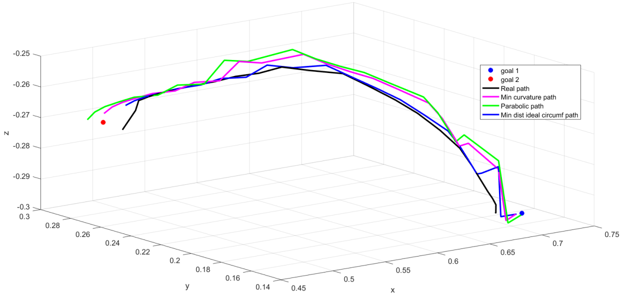


Figure 3.14: Representation of the different predictive paths obtained for the measured path (black curve) followed by the human operator

In order to compare the different approaches proposed for the computation of a nominal path, [30] suggested the use of the Fréchet distance criterion, which is

particularly exploited in the field of computational geometry as similarity metric between a couple of curves.
The meaning of Fréchet distance will be briefly exposed. Let us suppose that we are interested in evaluating how much two curves, namely π_1 and π_2, are similar. Let us also suppose that there exist two points: P_1 laying on π_1 and P_2 laying on π_2 which are traveling forward along the curve they belong to. Let us also consider that the rate of speed for either points may not necessarily be uniform. Hence, the Fréchet distance is defined as the minimum cord length that suffices to join P_1 and P_2.

Let us number the curves as:

- 1, the measured path;
- 2, the minimum curvature path;
- 3, the path obtained by minimizing the distance from the ideal cicumference centred in the shoulder position;
- 4, the parabolic path.

Therefore, we are interesting on analyzing the predictive curve that resulted most similar to curve 1, in order to understand if the use of one predictive path could provide a greater advantage with respect to the others.
Analyzing the results achieved in 5 different experiments, the trend respresented in Figure 3.15 is obtained:

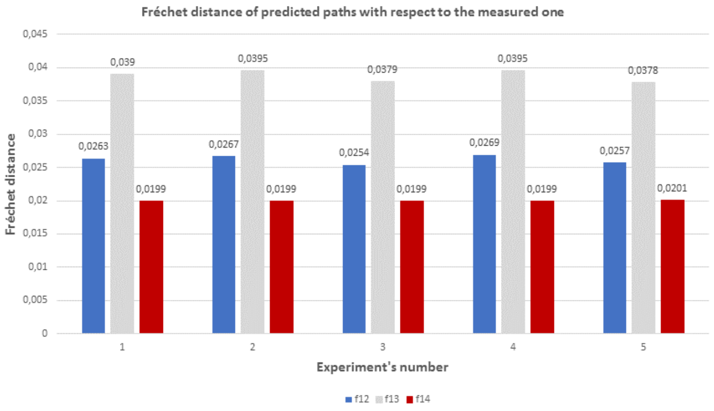
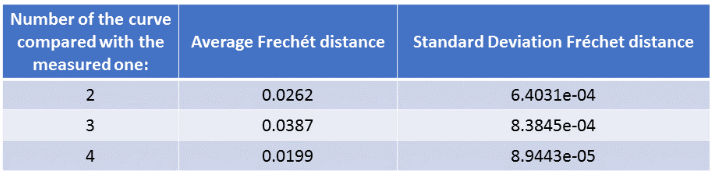


Figure 3.15: The histogram of the Fréchet distance obtained in each experiment by analyzing the similarity between curve 1 (the measured path) and each predictive path (curve 2, curve 3, curve 4) shows that the parabola is the most similar to the measured path. However all the predictive paths seems to be quite close to the real one since all the Fréchet distances are small values

where f_{1j} indicates the Fréchet distance beteween the j_{th} curve $(j = 2, 3, 4)$ and the measure curve (curve 1). The obtained statistics are expressed in table 3.16:

Obviously, based on the definition of Fréchet distance, the most similar path corresponds to the one having the lowest Fréchet distance. By observing the trend represented in Figure 3.15, it can be concluded that, since the parabola is the curve which always presents the smallest Fréchet distance, it may result the best method for describing the real path followed by the human operator. However, it can be further noticed that the method used to compute this kind of parabola is not straightforward, (see section 3.5) and it requires more computational memory than the minimum curvature path, which was used in the original formulation ([34]). Hence, it seems reasonable to conclude that the benefits of the application of these predictive paths are almost the same. As a consequence, since we are interested in analyzing the possibility of enhancing the performance of the intention inference algorithm described in [34] where a minimum curvature path was already used, the application of a parabolic path instead of the previous one seems not to be the most advantageous way to improve these performance.



Number of the curve compared with the measured one:	Average Frechét distance	Standard Deviation Fréchet distance
2	0.0262	6.4031e-04
3	0.0387	8.3845e-04
4	0.0199	8.9443e-05

Figure 3.16: Results of the Frechet distances of the predictive path with respect to the real one

Therefore, from now on, the search will be focused on investigating the advantages of including a larger number of observations, instead of a single one, in order to improve the overall inference process. This aspect will be addressed in Chapter 4.

4

Bayesian inference with multiple observations

4.1 Introduction

In Chapter 2 the Bayesian framework has been analyzed, as well as the architecture of the simplest intention recognition algorithm. That algorithm, in fact, estimated the human reaching target on the basis of one single observation during time: the wrist position.
Besides, the results achieved in Chapter 3, showed that the choice of one predictive path over another seemed not to be the most advantageous way to improve the inference process: all the predictive paths, in fact, returned pretty similar results, almost equal to the ones described in the original formulation (see [34]).
The aim of this chapter is to discuss the possibility of enhancing the performance of that algorithm by exploiting a wider set of observations when inferring the intended goal position. Then, the structure of a newer intention recognition algorithm based on this larger set of observed variables will be presented in detail.

If one observes the natural behaviour of human individuals in their daily lives, it is evident that humans exploit, to a large extent, the chance of using simultaneously their hands to perform even different operations at the same time. For instance, during activities as opening a water bottle or performing a small assembly task or writing something on a paper, humans generally use their principal hand to carry out the part of their work that requires a higher level of dexterity and control, while the other hand has generally the auxiliary task of helping finalise the overall operation. This is clearly done in accordance with the type of activity performed.
Another important observation related to the human natural behaviour is the

following: humans have a tendency to face the objects they are interested in and to keep visually monitored the task they are doing for all the duration of its realization, [43]. In this regard, the human gaze can be considered a relevant link to the human latent intentions and a powerful means of nonverbal communication even in human-human cooperation, [22]. As a consequence, if one takes inspiration from these kind of behaviours, it is apparent that the observation of the joint dual-arm movements and gaze direction can provide explicits clues for the purpose of recognizing the human's underlying intention. In the literature the concept of gaze monitoring was already addressed even in the field of human-robot interaction. However, some of the previous works used the gaze estimation for reproducing human-like behaviours, as in [43].
Despite these contributions, the purpose of this Thesis is to exploit the gaze estimate as an additional measure that, along with other available observations such as hands positions and their actual distance from goal locations, can result useful for inferring the intended goal.
To a similar extent, in [4] the gaze estimate was used to avoid the so-called 'Mida's touch problem' which takes place when every goal which at a certain time instant is in the operator's field of view is considered intended. Moreover, [8] proposed the use of a dual-feature HMM model that is capable of recognizing the intended goal on the basis of the joint gaze-hands observation. In view of these considerations, the importance of the observation related to the gaze during the process of intention inference appears to be confirmed.The question is now how to acquire and manage this measure.
In this work an estimate of the operator's gaze is obtained by exploiting the potential of a Microsoft Kinect camera. This depth camera, in fact, is capable of detecting some facial points, computing the plane that best interpolates them and retrieving a vector which is orthogonal to that plane, as schematically shown in Figure 4.1.

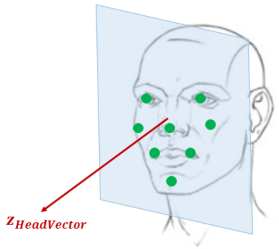


Figure 4.1: The kinect camera is capable of detecting some facial points, finding the plane that interpolates them and returing the vector orthogonal to that plane. This vector is denoted as $z_{HeadVector}$

This vector, denoted as $z_{HeadVector}$, can be considered representative of the orientation of the user's head while he is facing a certain target, as illustrated in Figure 4.2.

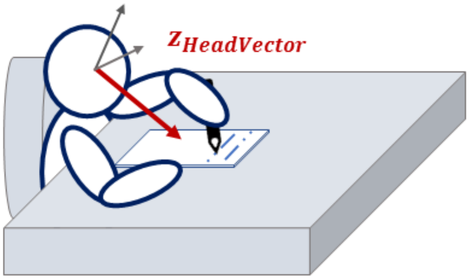


Figure 4.2: The outcoming vector belonging to the reference system associated with the head is representative of head direction when the human is facing a certain target

It is important to underline that the depth camera is also capable of providing a boolean value, $v_{HeadVector}$, that expresses a measure of the gaze validity. In fact, when the operator head is oriented in a very different direction with respect to the z-axis of the Kinect reference frame, the retrieved $z_{HeadVector}$ has no meaning, hence this boolean value becomes equal to 0. It should be recalled that the Kinect camera can retrieve, as previously expressed, also the estimated head position p_H it can be considered associated with.

We conclude this discussion by pointing out that, since all these measurements provide, also singularly, an indication of the intended goal location, it can be expected that their joint observations will be crucial for improving the overall inference process.

Hence, at each iteration k, the complete set of measured variables, which will be contained in vector $\Theta^{(k)}$, can be expressed as:

$$\Theta^{(k)} = \begin{bmatrix} p_{RW}^{(k)} & p_{LW}^{(k)} & p_{H}^{(k)} & z_{HeadVector}^{(k)} & v_{HeadVector}^{(k)} \end{bmatrix} \tag{4.1}$$

As a consequence, it will be necessary to find a method to model the likelihood function associated with these wide set of observations.

4.2 Reformulation of the system as a multi-observations Bayesian framework

In order to visualize in a better way the circumstance where multiple observations are available and can be exploited to compute their likelihood, one can graphically depict this framework as a multi-nodal Bayesian Network. A generic Bayesian Network, [28], in fact, consists in a graphic model which allows to represent the probabilistic relationships between continuous or discrete random variables: in fact, as the name itself says, the links between random variables are represented by means of a directed graph, hence, a network.
The reason for considering a Bayesian Network with multiple observations is that it is supposed to be an effective tool to represent joint probabilities and causal dependencies [37]. Indeed, in a Bayesian network each random variable can be graphically represented by a node, while the direct dependencies between two quantities can be expressed through a directed edge. Obviously the tip of the arrow points towards the variable that is causally dependent upon the variable stored in the node where the arc comes from. The most simple situation is depicted in Figure 4.3:

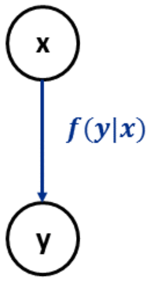


Figure 4.3: The most simple Bayesian network with a single observation node: the conditional probability of y given the x can be graphically depicted by a directed edge

It is so evident that even the problem of estimating the intended goal based on the observation of the wrist position can receive a proper graphic interpretation by means of this Bayes Network. In fact, in that case, x stood for a generic goal position $p_{G_i} \in \mathscr{G}$ which the algorithm had to recognize, while y represented the measured wrist position p_W. Clearly, the arc that connects x to y depicted the conditional density function of observing the wrist measurement under the assumption that the human was intended to reach p_{G_i}. Therefore, it expressed the causal dependence of the observed action upon the intention. The node from which the arc departs is usually denoted as 'parent node'; the one where the edge is directed to is named 'child node'.

Coming back to the analysis of the context where multiple observations are available, the situation can be represented, from a graphical point of view, as illustrated in Figure 4.4:

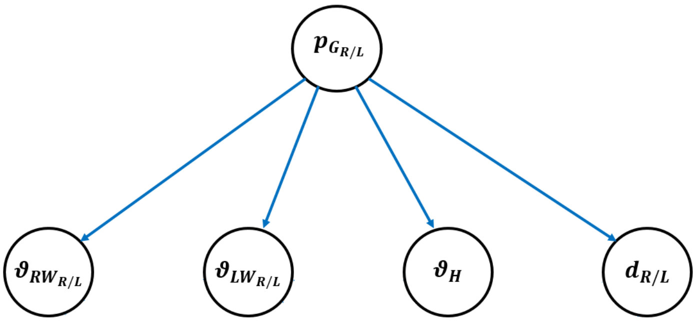


Figure 4.4: The Bayesian Network with multiple observation nodes which represent the whole set of available measurements. In order to simplify the graphical representation, the links among the children nodes are not reported in this figure

Since, in general, a task involves the simultaneous motion of both operator's hands, it could be possible to compute two distinct probability distributions: the probability that the i_{th} goal position is reached by using the right hand and the corresponding quantity for what concerns the left hand. This is the reason for using the subscripts R/L, where obviously R refers to the right wrist and L to the left wrist. As in the previous case, the underlying intention of reaching a certain goal position by using the right or the left hand, is graphically represented through the node $p_{G_{R/L}}$, while the children nodes represent the whole set of observations available to update the estimate. Let us focus on the meaning of each lower node.

For what concerns $\theta_{RW_{R/L}}$ and $\theta_{LW_{R/L}}$, their meaning will be clarified in section 4.5.

Moreover, in order to exploit the coordinated head-hands motions, under the aforementioned observations, it is possible to consider the significant contribution given by the outcoming vector associated with the head position $z_{HeadVector}$. In [43] it was observed that when humans are loooking at a generic target, if one defines a virtual plane laying on the operator's face defined by the neck and by the eye-eye line, the line that connects the middle of the eyebrows to the target

tends to be perpendicular to that plane. Hence, according to these considerations, the value of θ_H, the angle between $z_{HeadVector}$ and the unit vector that connects the estimated head position p_H to that target, can provide significant clues about the human state-of-mind, hence, about his intended goal.
Furthermore, it is believed, [39], that when a person is intentioned to grasp a certain object, prior to proceeding at reaching it by hands, it formulates a sort of 'mental map' of the location of the object. It is also reported, [39], that the ocular movements tend to occur about 100 ms before the start of an actual hand reaching movement; namely, a little anticipatory offset can be noticed between the gaze orientation and the corresponding hands reaching motions. Hence, the importance of the anlysis of $z_{HeadVector}$ in the process of recognizing the intended goal is further confirmed.
Lastly, another relevant information could be $d_{R/L}$, the distance of the considered wrist position from the center of mass of the target. The importance of including the distance within Θ is obvious: in fact it is clear that if a hand is getting away from a certain target, it is likely that the user is not interested in reaching it by using that hand. This distance is computed as the L2-norm of the center of each wrist from the center of mass of the considered goal.

4.3 The model of the likelihood function: a Gaussian Mixture Model

In the previous section a graphical model has been described: a sort of multi-observations Bayesian Network, capable of representing the contribution of a large set of observations in the process of inferring the probability of the goal. The problem is now how to manage and treat the dependencies between the variables, hence, how to model the conditional density of observing these set of measurements under the hypothesis of reaching the i_{th} goal's location, p_{G_i}. The fact that these observations are somehow interdependent is confirmed by the following cosiderations: since any hand reaching motion is firstly programmed at a higher level in our mind, the actual position of the wrist will be somehow related to the direction that the head assumed and that generally points towards the object the subject is willing to reach.
Furthermore, if one considers the framework of the human-robot collaboration it is quite unusual that the operator reaches simultaneously a certain object with both his/her hands. On the contrary, as previously said, it is reasonable to

assume that the operator could simultaneously use each hand to reach different goals. Consequently, in this framework, if there will be a strong evidence that one hand is moving towards the same i_{th} goal, this observation will reduce the probability that the other hand moves towards the i_{th} target.
Hence, in view of these considerations, it is also important to find a way to model the conditional density functions such as to take into account the mutual interrelation of the head-hands coordinated motion. In literature, [13], it is reported that a quite common practice consists in approximating the conditional density functions by using a mixture of Gaussian, hence exploiting the so-called 'Gaussian Mixture Models'. Hence, the situation could be graphically re-formulated as in Figure 4.5:

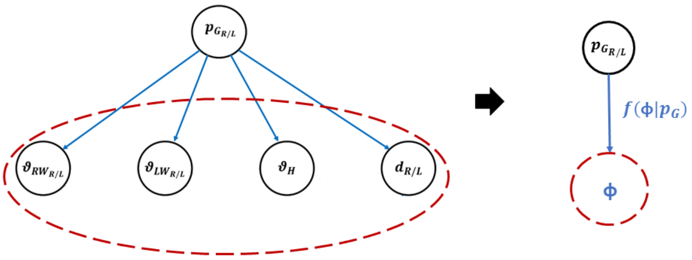


Figure 4.5: Multi-observation bayesian framework: all observations are jointly considered. In order to simplify the graphical representation. (The temporal propagation of the network is here not reported)

Thus, let us denote with Φ_i the feature vector, the vector that contains the following quantities related to the i_{th} goal position:

$$\Phi_i = \begin{bmatrix}\theta_{RW_{R/L_i}} & \theta_{LW_{R/L_i}} & \theta_{H_i} & d_{R/L_i}\end{bmatrix}^T \tag{4.2}$$

then the conditional probability of observing Φ_i under the assumption that the intended goal is p_{G_i} can be written as:

$$f(\Phi_i|p_{G_i}) = f(\theta_{RW_{R/L_i}}, \theta_{LW_{R/L_i}}, \theta_{H_i}, d_{R/L_i}|p_{G_i}) = \sum_{c=1}^{Nc} w_c f(\Phi|\mu_c, \Sigma_c) \tag{4.3}$$

where:

- Nc is the number of Gaussian components;
- $f(\Phi|\mu_c, \Sigma_c)$ is the probability density function associated with the c_{th} Gaussian component having mean vector equal to μ_c and covariance matrix equal to Σ_c;

- w_c is the weight associated with the c_{th} Gaussian component.

As can be easily seen in formula 4.3, a Gaussian Mixture Model is a parametric probability density function that can be represented by means of a weighted sum of Gaussian component densities, also denoted as 'clusters' such that the weight associated with the Gaussian components must satisfy the following relation:

$$\sum_{c=1}^{Nc} w_c = 1 \tag{4.4}$$

The probability density function associated with each component of the GMM (refer to equation 4.3) is represented by a D-variate Gaussian (according to the number of variables whose distribution we want to learn), namely:

$$f(\Phi|\mu_c, \Sigma_c) = \frac{1}{(2\pi)^{D/2}(det(\Sigma_c))^{1/2}} exp\{-\frac{1}{2}(\Phi - \mu_c)^T(\Sigma_c)^{-1}(\Phi - \mu_c)\} \tag{4.5}$$

where D is the number of variables or features contained in vector Φ whose distribution we want to model through a GMM.
A Gaussian Mixture Model, usually abbreviated to the acronym 'GMM', is largely used in recognition applications, since it has the capability of describing a wide class of sample distributions. In fact one of the most important aspects that lays on the basis of the GMMs is that they are capable to smoothly approximate arbitrarily shaped densities.
Thus, since we are interested in modeling the conditional density of the set of observations given a certain target by using a GMM, the first step is the collection of a very large dataset from which it is possible to apply the GMM fitting. To do that, some subjects were asked to execute a certain number of operations which required them to perform a large number of reaching motions towards three possible goal locations. The layout of this experimental campaign is depicted in Figure 4.6:

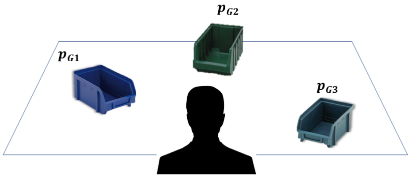


Figure 4.6: Layout for the experimental campaign for the acquisition of the training dataset from which the GMM can be extracted

These subjects were monitored by the Kinect camera during the execution of each reaching motion. Then, the corresponding $p_{RW}^{(k)}$, $p_{LW}^{(k)}$, $z_{HeadVector}^{(k)}$, $v_{HeadVector}^{(k)}$ retrieved at each k iteration were recorded and stored. Consequently, given the knowledge of the intended goal position for each reaching motion, it was possible to extract from the collected data a population of samples S:

$$S = \begin{bmatrix} \Phi_1 & \dots & \Phi_m & \dots & \Phi_M \end{bmatrix} \tag{4.6}$$

which could be used to learn the underlying GMM distribution. M is the number of total available observations. About 1650 observations were recorded.

In our case, two different Gaussian Mixture models have been derived by applying the EM algorithm. TIn fact, we are interested in obtaining the models for the following probability density functions:

1. $f(\theta_{RW/LW_i}^{(k)}, d_{RW/LW_i}^{(k)} | p_{G_i}^{(k)})$, which will be used in the case $v_{HeadVector}^{(k)}$ is false.
   Hence, here D is equal to 2 and each Φ_m is the following reduced 2x1 feature vector:

   $$\Phi_m = \begin{bmatrix} \theta_{RW/LW_i} & d_{RW/LW_i} \end{bmatrix}^T \tag{4.7}$$

   The GMM model associated with these set of observations will be referred from now on as 2D-GMM.

2. $f(\theta_{RW_{R/L_i}}^{(k)}, \theta_{LW_{R/L_i}}^{(k)}, \theta_{H_i}^{(k)}, d_{RW/LW_i}^{(k)} | p_{G_i}^{(k)})$, that will be exploited in the case $v_{HeadVector}^{(k)}$ is true.
   Hence, in this case D is equal to 4 and each Φ_m was the complete 4x1 vector expressed in equation 4.2.
   The GMM model associated with these set of observations will be referred from now on as 4D-GMM.

Thus, we are now interested in estimating the parameters of the components of the GMM that best fit the two groups of collected data (those for the 2D-GMM and those for the 4D-GMM). A common practice for obtaining the parameters of the Gaussian Mixture density components is the application of the Expectation-Maximization (EM) algorithm, which will be briefly reviewed in the following section.

4.4 The Expectation Maximization algorithm

The Expectation-Maximizaton algorithm, [6], [11], is a well known technique which is usually exploited to solve the problem of finding the Maximum likelihood extimator for the parameters belonging to a certain probability distribution, for instance, a mixture of gaussian density functions. Let us recall the definition of a maximum-likeliood estimator.
Let us consider that we have extracted M samples which are indipendent and identically distributed:

$$S = \{\Phi_1, \Phi_2, \dots, \Phi_m, \dots, \Phi_M\} \tag{4.8}$$

from an unkown distribution f which can be Gaussian or not. This distribution associated to the sampled data will be clearly characterized by a set of parameters Θ_{Par}.

$$\Theta_{Par} = \{w_1, \Sigma_1, \mu_1, \dots, w_c, \Sigma_c, \mu_c, \dots, w_{Nc}, \Sigma_{Nc}, \mu_{Nc}\} \tag{4.9}$$

Then, since the samples are considered independent, the density function could be expressed with the following product:

$$f(S|\Theta_{Par}) = \prod_{m=1}^{M} f(\Phi_m|\Theta_{Par}) = \mathscr{L}(\Theta_{Par}|S) \tag{4.10}$$

where $\mathscr{L}(\Theta_{Par}|S)$ represents the likelihood of the parameters, given the data. Therefore, the maximum-likelihood estimator is that set of parameters $\Theta^{\star}_{Par}$ that maximize the likelihood function, namely:

$$\Theta^{\star}_{Par} = \arg\max_{\Theta_{Par}} \mathscr{L}(\Theta_{Par}|S) \tag{4.11}$$

A common practice is to maximize the log-likelihood function that makes the problem's solution easier from a computational point of view.
In practice, the EM algorithm results to be helpful in situations where we have a large set of observed samples that we guess being generated from Nc different Gaussians components but we have no a piori knowledge of which datum comes from which Gaussian. Indeed, if the parameters of these component densities were known, one could easily compute the probability that each point belongs

to each cluster. However, since they are unknown, a plausible solution could be to iteratively compute the probability that each datum belongs to each cluster.

Focusing on the problem of estimating the parameters of a Gaussian Mixture Distribution with Nc components, let us consider again the M extracted samples of S (see 4.6).
Then, the parameters set can be described as:

$$\Theta_{Par} = \{w_c, \mu_c, \Sigma_c\} \quad \text{for} \quad c = 1, \dots, Nc \tag{4.12}$$

and the D-variate density expressed as in 4.5. Since EM is an iterative algorithm, let us also define the probability that at the t_{th} iteration the m_{th} sample belongs to the c_{th} Gaussian component with $\gamma_{mc}^{(t)}$. Hence,

$$\gamma_{mc}^{(t)} \triangleq P(Z_m = c | Y_m = \Phi_m, \theta_{Par^{(t)}}) = \frac{w_c^{(t)} f(\Phi_m | \mu_c^{(t)}, \Sigma_c^{(t)})}{\sum_{l=1}^{Nc} w_l^{(t)} f(\Phi_m | \mu_l^{(t)}, \Sigma_l^{(t)})} \tag{4.13}$$

where $Z_m \sim \mathscr{N}(0,1)$ and $\gamma_{mc}^{(t)}$ is such that $\sum_{c=1}^{Nc} \gamma_{mc}^{(t)} = 1$.

Essentially, the EM algorithm, given a random inizialization of Nc Gaussian distribution components, performs a soft-clustering of all the available data. In fact, for each sample y_m it computes the probability that the m_{th} sample belongs to the c_{th} Gaussian and it exploits this result to extimate again the means and covariances to fit better the available points. At each iteration the EM algorithm not only returns the estimated mean vectors, covariance matrices and weight vector associated to each cluster, but also a value that represents how good is the estimate (log-likelihood). The procedure is iterated until the difference between two consecutive values of the log-likelihood function is smaller that an arbitrarily tiny threshold, then the algorithm ends.
The expression 'soft clustering', which is opposed to 'hard clustering', refers to the fact that each sample is not deterministically assigned to a single cluster, but each sample Φ_m is associated with a value ($\in [0,1]$) that represents the probability that Φ_m belongs to that Gaussian component.
A simple example of how the EM algorithm works is shown in Figure 4.7:

Basically, after receiving the set of data points shown in Figure 4.7 on the left-hand side, having a certain unknown distribution and coming from Nc unknown sources, it iteratively estimates the mean vector, the covariance matrices and the weight vector associated to each Gaussian component. As it is visible in Figure 4.7, the estimated means, which are represented by the coloured rings, are

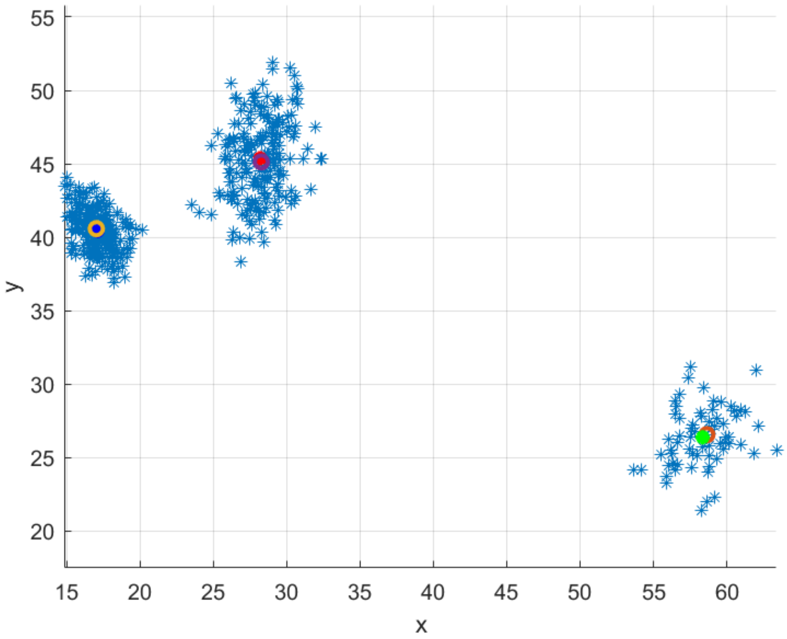


Figure 4.7: The EM algorithm receives a set of data points (blue) having a certain distribution and coming from Nc unknown sources. Then it iteratively estimates the mean, the covariance, the weights associated to each Gaussian component. The estimated means (coloured rings) are basically equal to the real ones (plain circles)

.

basically equal to the real ones which are represented by the full circles pictured on that figure. For what concerns the convergence properties, it is reported that EM suffers from sub-optimality problems, namely it is not guaranteed than the global optimum of the log-likelihood function is found.

In order to overcome this difficulty the solution that is commonly applied is to randomly initialize the algorithm many times and choose as final guess for the parameters vector the inizialization that produced the highest likelihood. Generally, k-means algorithm, [18], (where k is part of the algorithm's name, even if it should represent the number of components, hence, Nc in our case) is used to obtain a good initialization, namely the mean, covariance matrix and weigth of each Gaussian component.

The difference between k-means and EM is that the first one performs a hard clustering of data, hence assigns each sample to a cluster or another.

Hence the k-means algorithm takes as input the whole set of available samples and computes the mean vectors and covariance matrices of the Nc gaussians

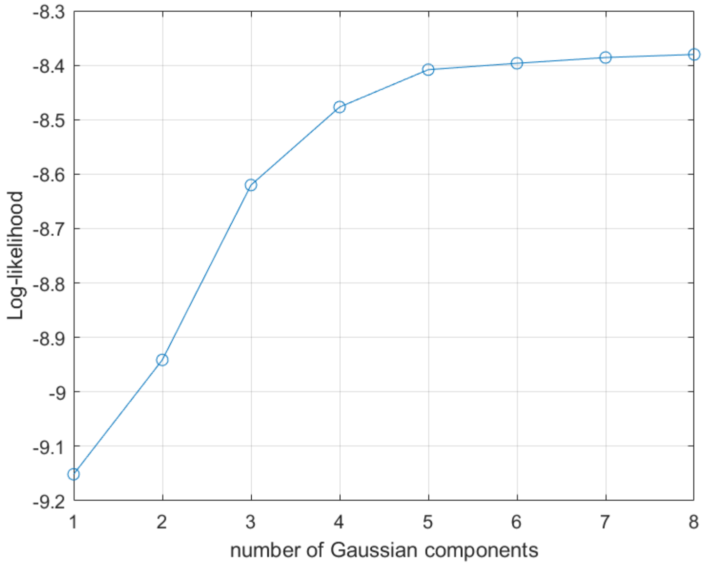


Figure 4.8: In order to select the proper number of Gaussian components (Nc) defining the GMM distribution, EM is run multiple times for increasing values of Nc. The number of components is definitively selected as the value for which the convergence of the corresponding Log-Likelihood occurred. For instance, in this figure Nc corresponds to 5

that are used as means and covariance matrices of the c_{th} gaussian components at the initial iteration (iteration zero) of Expectation-Maximization. The number Nc, that represents the cardinality of EM, must be a priori selected. This selection is generally made as explained in Figure 4.8.
Basically, the k-means algorithm works as follows:

1. Initialization: given the number of clusters, Nc, k-means starts randomly generating Nc centroids (one for each cluster) which represent the initial candidate mean vectors;

2. Compute distances from centroids: since k-means is a distance-based clustering algorithm, it then computes the distance of each data Φ_m from each centroid.

3. Compute the minimum distance and assign a label to each datum: hence it assigns to each data a label that represent the cluster whose centroid resulted closer to it;

4. Data clustering: the samples that have the same label are grouped together;

5. Compute the mean vectors: the mean vector of each cluster is computed as the sampled mean of the data belonging to that cluster;

6. Compute the covariance matrices: the covariance matrix is obtained as the covariance of the data that have been grouped together and become the components of a cluster;

7. Compute the weight vector: the initial weights are evaluated by dividing the number of data belonging to each cluster by the total number of data;

8. Convergence check: if the distance between the just found mean vectors and the initial mean vector randomly computed is greater than an arbitarily small threshold, the centroids become equal to the ones just obtained and the procedure is iterated again from step 2. Otherwise the algorithm ends and returns the obtained mean vectors, covariance matrices and weight vector.

The steps considered by EM are, [11] :

1. Initialization: The initial estimates of the weights $w_c^{(0)}$, mean vectors $\mu_c^{(0)}$, and covariance $\Sigma_c^{(0)}$, $c = 1, \ldots, Nc$ are those returned by the k-means algorithm. Hence, it is computed the corresponding log-likelihood of the M data points, under the assumption that each sample is extracted by the gaussian mixture characterized by the obtained parameters:

$$L^{(0)} = \frac{1}{M}\sum_{m=1}^{M} log(\sum_{c=1}^{Nc} w_c^{(0)} f(\Phi_m|\mu_c^{(0)}, \Sigma_c^{(0)})) \tag{4.14}$$

   So, $L^{(0)}$ represents the initial log-likelihood.

2. E-step: For each m_{th} sample it is evaluated the probability that it belongs to each c_{th} Gaussian component

$$\gamma_{mc}^{(t)} = \frac{w_c^{(t)} f(\Phi_m|\mu_c^{(t)}, \Sigma_c^{(t)})}{\sum_{l=1}^{Nc} w_l^{(t)} f(\Phi_m|\mu_l^{(t)}, \Sigma_l^{(t)})}, m = 1, \ldots, M, c = 1, \ldots, Nc \tag{4.15}$$

   and, for each Gaussian component, it is evaluated the sum of the probabilities that each m_{th} sample belongs to the c_{th} Gaussian component:

$$n_c^{(t)} = \sum_{m=1}^{M} \gamma_{mc}^{(t)}, c = 1, \ldots, Nc. \tag{4.16}$$

3. M-step: The new weigths, mean vectors and covariance matrices are estimated, based on the pobabilities $\gamma_{mc}^{(t)}$ computed at the previous time step,

as follows:

$$w_c^{(t+1)} = \frac{n_c^{(t)}}{M}, c = 1, \dots, Nc. \tag{4.17}$$

$$\mu_c^{(t+1)} = \frac{1}{n_c^{(t)}} \sum_{m=1}^{M} \gamma_{mc}^{(t)} \Phi_m, c = 1, \dots, Nc. \tag{4.18}$$

$$\Sigma_c^{(t+1)} = \frac{1}{n_c^{(t)}} \sum_{m=1}^{M} \gamma_{mc}^{(t)} (\Phi_m - \mu_c^{(t+1)})(\Phi_m - \mu_c^{(t+1)})^T, c = 1, \dots, Nc. \tag{4.19}$$

4. Convergence check: The log-likelihood is evaluated again under the assumptions that the data sources are now the gaussian components characterized by the parameters obtained at time step $t+1$:

$$L^{(t+1)} = \frac{1}{M} \sum_{m=1}^{M} log(\sum_{c=1}^{Nc} w_c^{(t+1)} f(\Phi_m | \mu_c^{(t+1)}, \Sigma_c^{(t+1)})) \tag{4.20}$$

Return to step 2 if $|L^{(t+1)} - L^{(t)}| > \delta$, otherwise end the algorithm. Therefore, if the difference between the previous log-likelihood and the actual one is greater than a predefined threshold δ, the algorithm is iterated again starting from step 2; otherwise the algorithm is terminated.

An example of how the Log-likelihood evolves at the various iterations of EM before it terminates is illustrated in Figure 4.9.

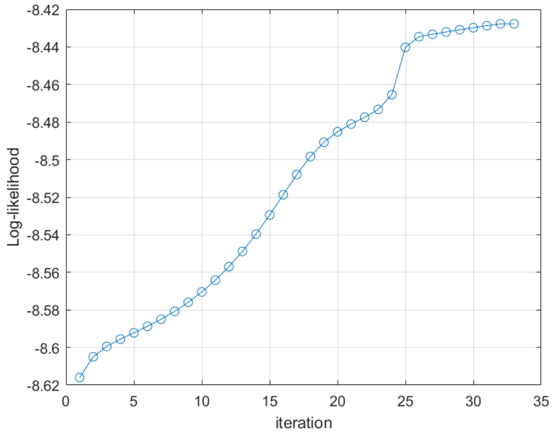


Figure 4.9: An example of the evolution of the Log-likelihood during the application of EM for a threshold δ equal to 10^{-3}. In this case the convergence is reached at the 33-th iteration

Finally, it is worth underlying that the model learnt by exploiting EM is general, in the sense that can be applied to any typology of robotic cell (assuming that the layout of the workspace, namely, the goal locations, are known). Then it is also obviuos that a specific GMM learned for a certain robotic cell could achieve higher performances, when considering the inference problem, than a generic one.

4.5 Making inference about the simultaneous hands' motions

The purpose of this section is to illustate the procedure according to which the probabilities are assigned to the goals and how they are updated. Even for the case of multiple observations are available, the intention estimation algorithm recursively updates a discrete probability distribution over the set of possible goals that can be reached by the operator's hands. Hovewer, the structure of the intention recognition algorithm and the updating mechanism will be organized in a different way with respect to the one described in Chapter 2.
First of all, in view of the considerations expressed in the previous sections, the intention recognition algorithm must be endowed with the capability of potentially recognizing at the same time each hand reaching target.
Hence, based on what has been observed at iteration k, the intention recognition algorithms will update simultaneously the probability that at iteration $k+1$ the generic i_{th} target is intended by the left or by the right hand on the basis of the whole set of retrieved measurements, that is common to both the updating algorithms. Therefore, let us recall the recursive Bayes formula that allowed to update the probability of reaching the i_{th} goal position when only the wrist position p_W (left or right) was observed (see Chapter 2 for details).

$$P^{(k+1)}(p_{G_i}) \propto P^{(k)}(p_{G_i}) f(\theta_W^{(k+1)} \mid p_{G_i}, \theta_W^{(k)}) \tag{4.21}$$

where

$$P^{(k+1)}(p_{G_i}) = f(p_{G_i} \mid \theta_W^{(k+1)}) \tag{4.22}$$

As shown in expression the probability of reaching a goal was based, in addition to its prior, only on the observation of the angle (see Chapter 2) associated with the hand that was moving: be it the right one or the left one. As a consequence the previous intention recognition algorithm was not capable of taking into account the mutual interrelation of the hands when only one of them was moving.

Now, as explained previously, the probability of reaching the goal with right wirst or with the left one are simultaneosuly updated as if they were distinct quantities, even if their 'mutual contribution' in increasing or reducing the probability of each goal, is always taken into consideration, as will be explained afterwards. Thus, let us rewrite the posterior probabilities of reaching the i_{th} goal position, as:

$$f_{LW}(p_{G_i} \mid \theta_{LW_L}^{(k+1)}, \theta_{RW_L}^{(k+1)}, \theta_{H}^{(k+1)}, d_{LW}^{(k+1)}) = P_{LW}^{(k+1)}(p_{G_i}) \tag{4.23}$$

when inference is made for the left wrist
and

$$f_{RW}(p_{G_i} \mid \theta_{RW_R}^{(k+1)}, \theta_{LW_R}^{(k+1)}, \theta_{H}^{(k+1)}, d_{LW}^{(k+1)} = P_{RW}^{(k+1)}(p_{G_i}) \tag{4.24}$$

when inference is made for the right one.

Hence the Bayes formulae expressed in equations 4.25 and 4.26 can be recursively applied for what concerns the left hand and for the right hand respectively:

$$P_{LW}^{(k+1)}(p_{G_i}) \propto P_{LW}^{(k)}(p_{G_i}) f_{LW}(\theta_{LW_L}^{(k+1)}, \theta_{RW_L}^{(k+1)}, \theta_{H}^{(k+1)}, d_{LW}^{(k+1)} \mid p_{G_i}, \theta_{LW_L}^{(k)}, \theta_{RW_L}^{(k)}, \theta_{H}^{(k)}, d_{LW}^{(k)}) \tag{4.25}$$

$$P_{RW}^{(k+1)}(p_{G_i}) \propto P_{RW}^{(k)}(p_{G_i}) f_{RW}(\theta_{RW_R}^{(k+1)}, \theta_{LW_R}^{(k+1)}, \theta_{H}^{(k+1)}, d_{RW}^{(k+1)} \mid p_{G_i}, \theta_{RW_R}^{(k)}, \theta_{LW_R}^{(k)}, \theta_{H}^{(k)}, d_{RW}^{(k)}) \tag{4.26}$$

It should be underlined the fact that, regarding at equation 4.25, the probability that the i_{th} goal is reached by the left hand depends not only on the observation related to the left wrist position but also on the observation that concerns the right wrist position (refer to 4.26).
An equivalent consideration holds when evaluating the probability of reaching the goal using the right hand. The reason for including the measure associated with the right wrist position when $P_{LW}^{(k+1)}(p_G)$ must be computed lies in the fact that, as previously explained, we assume that the operator reaches a certain goal location with one hand. As a consequence, if at time step k there is a high evidence that the left hand is directed towards the i_{th} target, this will decrease the probability that at the same time step the i_{th} goal is also intended

by the right hand. If this aspect was not taken into account, the probability distribution associated with each hand would evolve in a completely independent manner, creating an irrealistic context.

Let us describe the new structure of the intention recognition algorithm.
It is worth clarifying that, even if we consider two separate algorithms, one making inference for the right wrist and the other making inference for the left wrist, these are processed simultaneously, having a simmetric structure.
Let us denote the structures that allows to make inference about the left and the right hand with the names 'Inference Engine Right' and 'Inference Engine Left', respectively.
The generic structure is represented in Figure 4.10:

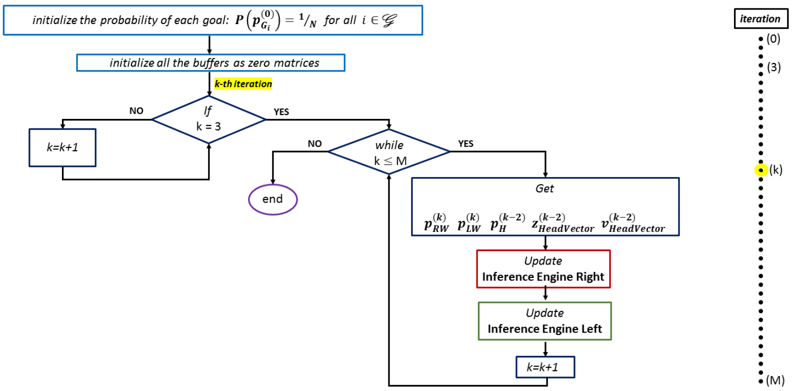

initialize the probability of each goal: $P\left(p_{G_i}^{(0)}\right) = {}^{1}/_{N}$ for all $i \in \mathscr{G}$
initialize all the buffers as zero matrices
k-th iteration
NO
If
k = 3
YES
k=k+1
NO
while
k ≤ M
YES
end
Get
$p_{RW}^{(k)}$ $p_{LW}^{(k)}$ $p_{H}^{(k-2)}$ $z_{HeadVector}^{(k-2)}$ $v_{HeadVector}^{(k-2)}$
Update
Inference Engine Right
Update
Inference Engine Left
k=k+1
iteration
(0)
(3)
(k)
(M)

Figure 4.10: Flow diagram representing the general functioning mechanism of the multi-observations intention inference algorithm. Each iteration correspond to a new set of measures retrieved by the Kinect. This sensor device samples the skeletal points according to a space-based criterion.

In view of these considerations, only the algorithm which recursively computes the probability of each goal from the righ-hand perspective will be analized afterwards. The corresponding procedure associated with the left wrist will be simultaneously described by putting in round brackets the terms that are used by the complementary left-hand algorithm.

As explained in Chapter 2, we assume that the location of all the possible targets is a finite set and a priori known. Hence, the very first step is the initialization of the probability associated with each goal.
Even in this case, since at the beginning of the collaboration the operator has not already reached any target and there is no evidence that one is more likely with

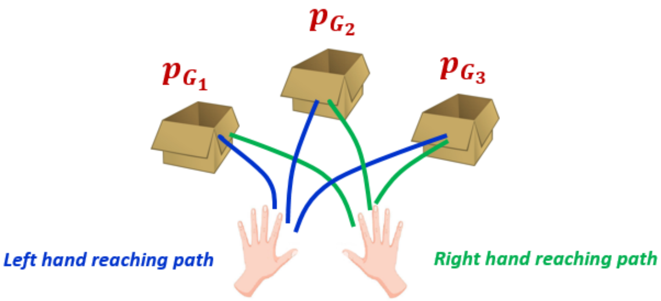


Figure 4.11: Two distinct probability distributions are simultaneously computed: the probability that the i_{th} goal is intended to be reached by the left hand and the probability that he i_{th} goal is intended to be reached by the right hand

respect to the others, each target is supposed to have the same probability of being reached. Hence, the probability distribution at iteration zero is considered uniformly distributed over the number of the goals.
Moreover, as previously said, two distinct Inference Engines (see Chapter 2) are present:

1. Inference Engine Right, in charge of collecting the whole set of available information (goal positions, skeletal measures, probability associated with the goals) and making inference about the righ hand;
2. Inference Engine Left, which operates in a equivalent manner and makes infernce about the left hand.

Clearly they have a mirror functioning.

Focusing on the implementative details, each Inference Engine is endowed with four buffers:

- Buffer$_{RW}$ that is in charge of collecting the measurements of the right wrist provided;
- Buffer$_{LW}$ that is in charge of storing the measurements of the left wrist;
- Buffer$_{Head}$ that is in charge of collecting the measurements of the operator's head position;
- Buffer$_{ZHV}$ that is in charge of storing the measurements related to $z_{HeadVector}$;

It should be recalled that all these measurements are obtained by means of a Microsoft Kinect camera.

In a similar way as set out in Chapter 2, each buffer is filled and updated according to a FIFO logic. The updating rule of the buffers must be managed so as to ensure that each new group of measurements introduced in the different buffers are always syncronized.
In fact for what concerns Inference Engine Right (Inference Engine Left), if the distance between the new acquired measures of right wrist (left wrist) and the last one introduced in the Buffer$_{RW}$ (Buffer$_{LW}$) overcomes a spatial threshold δ_{Buffer} , the positions contained in Buffer$_{RW}$, Buffer$_{LW}$, Buffer$_{Head}$ and Buffer$_{ZHV}$ are all moved backwards one place such as to insert the new measure in the last position of the buffer, according to the same criterion expressed in 2.7.
This procedure ensures that all the measures remain always synchronized. However, in order to manage the observed anticipatory effect of the gaze of about 100 ms ([39]) with respect to the corresponding hand reaching motions, at each k_{th} iteration of the algorithm, $p_{RW}^{(k)}$ and $p_{LW}^{(k)}$ are introduced in Buffer$_{RW}$ and Buffer$_{LW}$ respectively, while $z_{HeadVector}^{(k-2)}$ and $p_{H}^{(k)}$ are introduced in Buffer$_{ZHV}$ and Buffer$_{Head}$. In fact it has been observed that the temporal distance between two consecutive samples corresponded to approximately 100 ms.
Hence the anticipative effect of the gaze direction with respect to the corresponding hand reaching motion is taken into account.

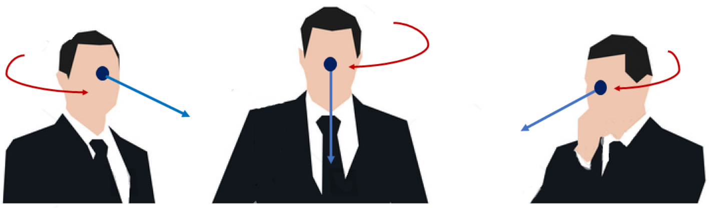


Figure 4.12: When a human decides to shift its focus from one objective to another one, the $z_{HeadVector}$ changes direction and helps claryfing the intended goal. This change in head orientation occurs slightly before the corresponding hand reaching motion

Each buffer mantains the same function described in Chapter 2: namely, storing a certain number of sufficiently distant measurements and using the ones related to the wrist positions to compute the so-called ‘previous unit tangent vector’ and the so-called ‘future unit tangent vector’ (see Chapter 2).
However, the quantities ‘previous unit tangent vector’ and ‘future unit tangent vector’ are, this time, computed in a slightly different manner than the one illustrated in Chapter 2.

In fact, in order to take into account the evolution of the actual path followed by the operator’s hand when moving from a certain position to a target one,

it seemed reasonable to compute $\mathbf{t}_{prev}$ and $\mathbf{t}_{fut}$ in a way that not all positions used for computing them are equally weighted, but such that their weight, hence, their contribution in computing the tangent vector, varies according to the acquisition time: namely, the weight should be higher for the most recent ones and should decrease for the eldest ones. This can be obtained by exploiting the Exponentially Weighted Moving Average ($EWMA$) technique, [36].
Therefore, for each new measure introduced in the buffer, $\hat{t}_{prev}$ and $\hat{t}_{fut}$ are computed applying the $EWMA$ to the preceeding unit tangent vectors. This is equivalent to the application of a low-pass filter on these vectors.
Let us assume that the dimension of the Buffers is an odd number, Db. The buffers Buffer$_{RW}$ and Buffer$_{LW}$ are split in two parts. The first half is used for computing $\hat{t}_{prev}$, while the second half is used for computing $\hat{t}_{fut}$. For the first half of each buffer the following set of vectors are evaluated:

$$\hat{t}^{(\frac{Db+1}{2}-i)} = \frac{p_W^{(\frac{Db+1}{2})} - p_W^{(i)}}{||p_W^{(\frac{Db+1}{2})} - p_W^{(i)}||} \qquad \forall i = 1, 2, \ldots, \frac{Db-1}{2} \tag{4.27}$$

where $p_W^{(pb)}$ is the pb-th wrist position (be it left or right) stored in the considered buffer. While for the second half of the buffers Buffer$_{RW}$ and Buffer$_{LW}$ the following set of vectors are computed:

$$\hat{t}^{(Db-i)} = \frac{p_W^{Db} - p_W^{(i)}}{||p_W^{(Db)} - p_W^{(i)}||} \qquad \forall i = \frac{Db+1}{2}, \frac{Db+3}{2}, \ldots, Db-1 \tag{4.28}$$

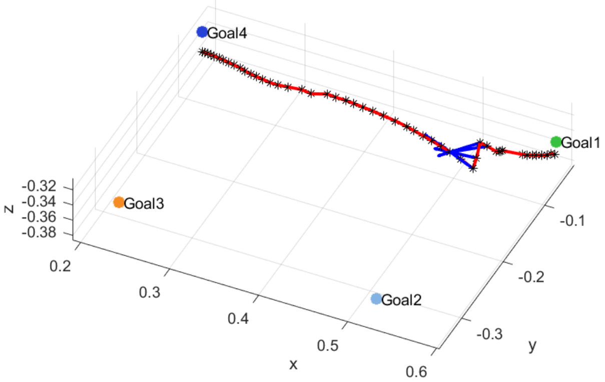


Figure 4.13: Representation of the vectors (blue arrows) which the EWMA technique will be applied on

These vectors corresponds to the blue arrows illustrated in Figure 4.13.

Hence, in order to compute $\hat{t}_{prev}$, these vectors are arranged in a matrix A_{prev} whose i_{th} column is equal to $\hat{t}^{(\frac{Db+1}{2}-i)}$.

While, when computing $\hat{t}_{fut}$, these vectors are arranged in a matrix A_{fut} whose i_{th} column is equal to $\hat{t}^{(Db-i)}$.

Then, the first step of the application of the $EWMA$ technique to compute is to initialize z_{old} as the oldest tangent:

$$z_{old} = A_{prev/fut}(:, 1) \tag{4.29}$$

where $A_{prev/fut}$ indicates that obviously select A_{prev} must be used for computing $\hat{t}_{prev}$, while A_{fut} for $\hat{t}_{fut}$.

secondly, for the i_{th} column of $A_{prev/fut}$:

$$z = (1 - \lambda)z_{old} + \lambda A_{prev/fut}(:, i) \tag{4.30}$$

where λ is a tunable parameter such that $0 \leq \lambda \leq 1$. Clearly λ small entails a slow system, while λ big implies a dead-beat system. Here λ is set equal to 0.7. Eventually,

$$z_{old} = z \tag{4.31}$$

From a practical point of view, the $EWMA$ technique, applying on these tangents a weight that decreases exponentially from the most recent one to the oldest one, allows to detect in a fast way whether or not a change in the direction of the tangent occurred and, eventually, take it into consideration. Thus, as soon as a new position is introduced in the buffer, the $EWMA$ tangents are recomputed obtaining for each iteration the vectors depicted in Figure 4.14.

As a result, each Inference Engine contains, at each k_{th} iteration, the following unit tangent vectors:

- $\hat{t}^{(k)}_{LW_{prev}}$ and $\hat{t}^{(k)}_{LW_{fut}}$;
- $\hat{t}^{(k)}_{RW_{prev}}$ and $\hat{t}^{(k)}_{RW_{fut}}$;

having the same meaning explained in chapter 2.

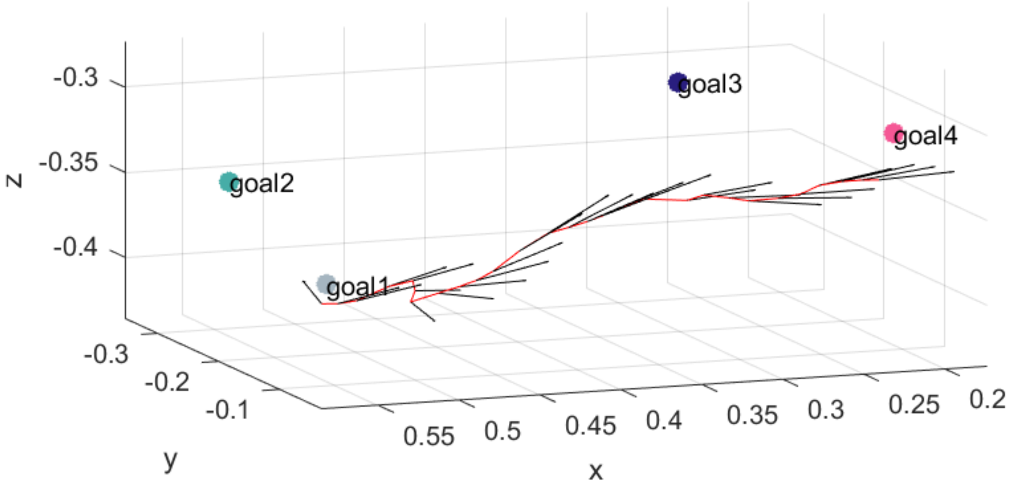


Figure 4.14: The figure displays the resulting EWMA unit tangent vectors computed for all the samples that has been iteratively introduced in the buffer

Let us now describe the steps performed by the new proposed Intention Inference Algorithm.

1. assuming that the layout of the robotic cell is known, hence, the location of the target positions is established, the probability distribution is initially uniformly split among the number of the goals (N), as previously mentioned. It should be pointed out that the algorithm can deal with a goal description in terms of single point or in terms of confidence ellipsoid (as it will be described in Chapter 5). Hence at iteration zero the probability associated with the i_{th} goal is equal to:

$$P^{(0)}(p_{G_i}) = \frac{1}{N} \qquad i = 1, \ldots, N \tag{4.32}$$

2. the buffers of the Right Inference Engine and those of the Left Inference Engine are all initialized as zero matrices.
3. when a prescribed number of sampled positions have been retrieved by the sensing device, the buffers start to be filled with the measurements according to the logic previously described.
4. when the buffers are full, the updating mechanism of both Inference Engine Right and Inference Engine Left starts.
5. hence for what concerns Inference Engine Right (Inference Engine Left) it is evaluated whether the last measure p_{RW} (p_{LW}) stored in the buffer is

inside one of the goal. Indeed, if the i_{th} goal is described in terms of single point, it is evaluated whether p_{RW} (p_{LW}) is inside a sphere centered in p_{G_i} having radius equal to 2 cm; otherwise, it is evaluated whether p_{RW} (p_{LW}) lays inside one of the confidence ellipsoids (see Chapter 5) associated with each goal.
If p_{RW} (p_{LW}) is inside one goal, the probabilities of all the goals are kept equal to those of the previuos time steps:

$$P^{(k+1)}(p_{G_i}) = P^{(k)}(p_{G_i}) \qquad \forall p_{G_i} \in \mathscr{G} \tag{4.33}$$

This procedure ensures that for all the iterations where the human hand results to be located quite close to the center of mass of the target or, actually, inside a confidence ellipsoid, the probability associated with that keeps constant and equal to a high value.

6. otherwise, in both Inference Engine Right and Inference Engine Left, $\hat{t}_{LW_{prev}}$, $\hat{t}_{LW_{fut}}$, $\hat{t}_{RW_{prev}}$ and $\hat{t}_{RW_{fut}}$ are computed through the application of the $EWMA$ technique, as previously discussed.

7. then, the minimum distance $minDist$ between the considered actual hand position p_{RW} (p_{LW}) and the center of mass of each target is evaluated, as shown in equations 4.34 and 4.35.

$$MinDist = \min_{p_{G_i} \in \mathscr{G}} ||p_{RW} - p_{G_i}|| \tag{4.34}$$

and, equivalently, for what concerns Inference Engine Left

$$(MinDist = \min_{p_{G_i} \in \mathscr{G}} ||p_{LW} - p_{G_i}||) \tag{4.35}$$

Moreover each target is associated with a flag, namely a boolean value, that aims at distinguishing the potential intended goals form the not intended ones. The criterion is the one expressed in the following equations:

$$flag_i = \begin{cases} 1 & \text{if } (\hat{t}_{RW_{fut}})^T (p_{G_i} - p_{LW}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

and, equivalently, for what concerns Inference Engine Left:

$$flag_i = \begin{cases} 1 & \text{if } (\hat{t}_{LW_{fut}})^T (p_{G_i} - p_{LW}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

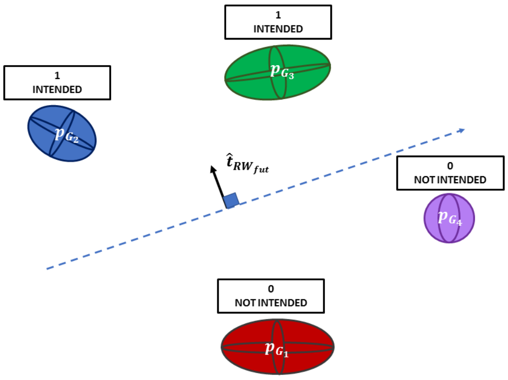


Figure 4.15: According to the previously expressed criterion and observing the direction of the actual tangent $\hat{t}_{RW_{fut}}$ the goals that lay below the dashed arrow cannot be considered intended

Hence, if a goal is labelled with value 1 it means that it belongs to the set of intended ones, otherwise it can be considered not intended. The situation is schematically depicted in Figure 4.15.

8. thus, if no goal is intended (all the goals are labelled with 0) and the value of MinDist is greater than a specified value δ_{DistNI} that represents a reasonable distance for the layout under analysis (in this Thesis this distance is set to 15 cm), it means that the considered operator's hand is too far from all the prescribed targets. Consequently, it can be assumed that the operator is not intended to reach none of the goals and all the probabilities of the whole goal set are decreased according to a penalization coefficient PC which is a tunable parameter between 0 and 1.

$$P^{(k+1)}(p_{G_i}) = PC \;\; P^{(k)}(p_{G_i}) \qquad\qquad \forall p_{G_i} \in \mathscr{G} \tag{4.36}$$

9. otherwise, if the operator's hand is not located inside any goal and the previous condition is not fulfilled, the probability of all the intended targets must be updated, evaluating their likelihood. Instead, the probability related to non intended goal locations is set equal to a small value.

It should be recalled that the vector of measurements $\Theta^{(k)}$ which contains the measured variables according to which the likelihood has to be evaluated is:

$$\Theta^{(k)} = \begin{bmatrix} p_{RW} & p_{LW} & p_H & z_{HeadVector} & v_{HeadVector} \end{bmatrix}$$

Hence, if at the k_{th} iteration $v_{HeadVector}$ is true, the likelihood associated with the i_{th} goal position, $L_{p_{G_i}}$, is evaluated according to the learnt 4d-GMM (see section 4.3) that for what concerns the right hand corresponds to the expression 4.37:

$$f(\theta^{(k)}_{RW_{Ri}}, \theta^{(k)}_{LW_{Ri}}, \theta^{(k)}_{H_i}, d^{(k)}_{RW_i}|p_{G_i}) = \sum_{c=1}^{Nc} w_c f(\Phi^{(k)}_i|\mu_c, \Sigma_c) \tag{4.37}$$

where in this case:

$$\Phi^{(k)}_i = \begin{bmatrix}\theta^{(k)}_{RW_{Ri}} & \theta^{(k)}_{LW_{Ri}} & \theta^{(k)}_{H_i} & d^{(k)}_{RW_i}\end{bmatrix}^T \tag{4.38}$$

Equivalently, when making inference about the left hand, the corresponding expression will be:

$$f(\theta^{(k)}_{LW_{Li}}, \theta^{(k)}_{RW_{Li}}, \theta^{(k)}_{H_i}, d^{(k)}_{LW_i}|p_{G_i}) = \sum_{c=1}^{Nc} w_c f(\Phi^{(k)}_i|\mu_c, \Sigma_c) \tag{4.39}$$

where in this case:

$$\Phi^{(k)}_i = \begin{bmatrix}\theta^{(k)}_{LW_{Li}} & \theta^{(k)}_{RW_{Li}} & \theta^{(k)}_{H_i} & d^{(k)}_{LW_i}|p_{G_i}\end{bmatrix}^T \tag{4.40}$$

Otherwise, if at that iteration $v_{HeadVector}$ results to be false, the algorithm evaluates the likelihood according to the learnt 2d-GMM (see section 4.3). In the 2d case, the expression of the likelihood associated with the i_{th} goal is:

$$f(\theta^{(k)}_{RW_{Ri}}, d^{(k)}_{RW_i}|p_{G_i}) = \sum_{c=1}^{Nc} w_c f(\Phi^{(k)}_i|\mu_c, \Sigma_c) \tag{4.41}$$

where:

$$\Phi^{(k)}_i = \begin{bmatrix}\theta^{(k)}_{RW_{Ri}} & d^{(k)}_{RW_i}\end{bmatrix}^T \tag{4.42}$$

and

$$f(\theta^{(k)}_{LW_{Li}}, d^{(k)}_{LW_i}|p_{G_i}) = \sum_{c=1}^{Nc} w_c f(\Phi^{(k)}_i|\mu_c, \Sigma_c) \tag{4.43}$$

where:

$$\Phi^{(k)}_i = \begin{bmatrix}\theta^{(k)}_{LW_{Li}} & d^{(k)}_{LW_i}|\end{bmatrix}^T \tag{4.44}$$

Hence for computing the likelihood, the Intention Inference Algorithm at each iteration can switch between the two learnt GMMs according to the value of $v^{(k)}_{HeadVector}$.

Let us focus on the problem of determining the likelihood:

(a) The quantities $\theta^{(k)}_{LW_L}$, $\theta^{(k)}_{RW_L}$, $\theta^{(k)}_{RW_R}$, $\theta^{(k)}_{LW_R}$ can be easily computed. Let us explain their precise meaning of these angles and the motivation for considering the subscripts $_L$ or $_R$.

(b) When inference is made for what concerns the left hand, we evaluate:

- θ_{LW_L} represents the angle between the measured $\hat{t}_{LW_{fut}}$ and the normalized vector that connects the position of the left hand to the center of mass of the considered target.
- θ_{RW_L} represents the angle between the measured $\hat{t}_{LW_{fut}}$ and the normalized vector that connects the position of the right hand to the center of mass of the considered target.

as shown in Figure 4.16:

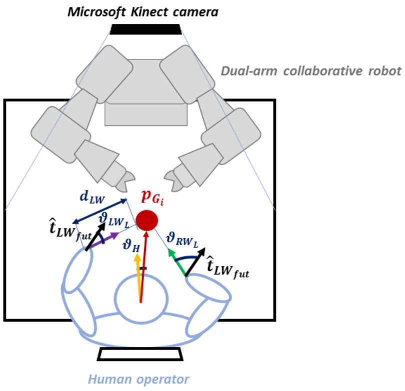


Figure 4.16: Measures retrieved by means of the Kinect Camera and used for making inference about the left hand

On the contrary, when inference is made on the right hand, we compute:

- θ_{RW_R} represents the angle between the measured $\hat{t}_{RW_{fut}}$ and the normalized vector which connects the position of the right hand to the center of the considered target.
- θ_{LW_R} represents the angle between the measured $\hat{t}_{RW_{fut}}$ and the normalized which connects the position of the left hand to the center of the considered target.

as depicted in Figure 4.17:

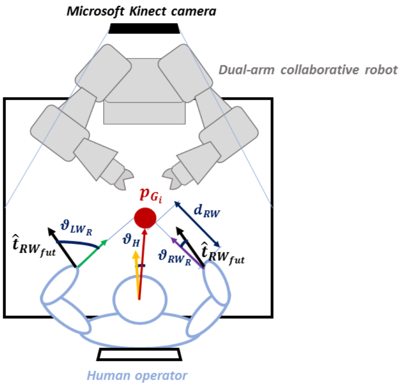


Figure 4.17: Measures retrieved by means of the Kinect Camera and used for making inference about the right hand

After these computations, the last quantities that must be evaluated are, the distance (in the case we are exploiting the 2d-GMM) and even θ_H (if $v^{(k)}_{HeadVector}$ is valid and we are using the 4d-GMM). As mentioned in section 4.2, θ_H is the angle between $z^{(k)}_{HeadVector}$ and the vector connecting the estimated head position p_H to the center of the i_{th} target. In fact it is implicit that the value of these angles must be evaluated for each target. Utimately, for what concerns the inference process related to the left hand, the distance is computed as the L2-norm of the actual left wrist position and the center of the i_{th} target:

$$d_{LW_i} = ||p_{G_i} - p_{LW}|| \tag{4.45}$$

while, referring to the right hand, the corresponding distance is expressed as:

$$d_{RW_i} = ||p_{G_i} - p_{RW}|| \tag{4.46}$$

Then, when all these variables have been properly computed for each target, they are evaluated according to the appropriate learnt GMM that returns the likelihood of the available group of observations Φ_i under the assumption that the intended goal is the i_{th} one. Hence, if the considered target position is described as a single point, the process of computing the likelihood ends, otherwise, if that goal is described in terms of confidence ellipoids (refer to 5.3), a number H of 'goal samples' that are sampled from the distribution associated with the i_{th} goal (gs_{hi}) are generated.

Hence, each one of these 'goal sample' is considered as an hypothesis on where the true center of mass of goal i, p_{G_i}, is located and the same computations from step (a) are replicated for each h_{th} 'goal sample' by substituting the coordinates of gs_{hi} to the ones of p_{G_i}. Therefore if the i_{th} target is described in terms of confidence ellipsoid that contains a number H of 'goal samples', there will be exactly H likelihoods associated with each i_{th} goal.

(c) consequently, the likelihood associated with each goal p_{G_i} is computed as the average likelihood of all the 'goal samples' belonging to it, as expressed in equation:

$$L_{p_{G_i}} = \frac{\sum_{h=1}^{H} L_{h_{p_{G_i}}}}{H} \tag{4.47}$$

where $L_{h_{p_{G_i}}}$ is the likelihood associated with the h_{th} 'goal sample'.

10. finally, for each goal the posterior probabilities are computed according to the recursive Bayes' rule. For the i_{th} target it corresponds to:

$$P^{(k)}(p_{G_i}) = \frac{P^{(k-1)}(p_{G_i}) f(\Phi_i^{(k)} | p_{G_i}^{(k)})}{\sum_{l=1}^{N} P^{(k-1)}(p_{G_l}) f(\Phi_i^{(k)} | p_{G_l}^{(k)})} \tag{4.48}$$

Then the Intention Inference Algorithm ends and for each new incoming set of available observations retrieved by the Microsoft Kinect camera, it starts again from step 1.

The flow diagram of the algorithm is illustrated in Figure 4.18:

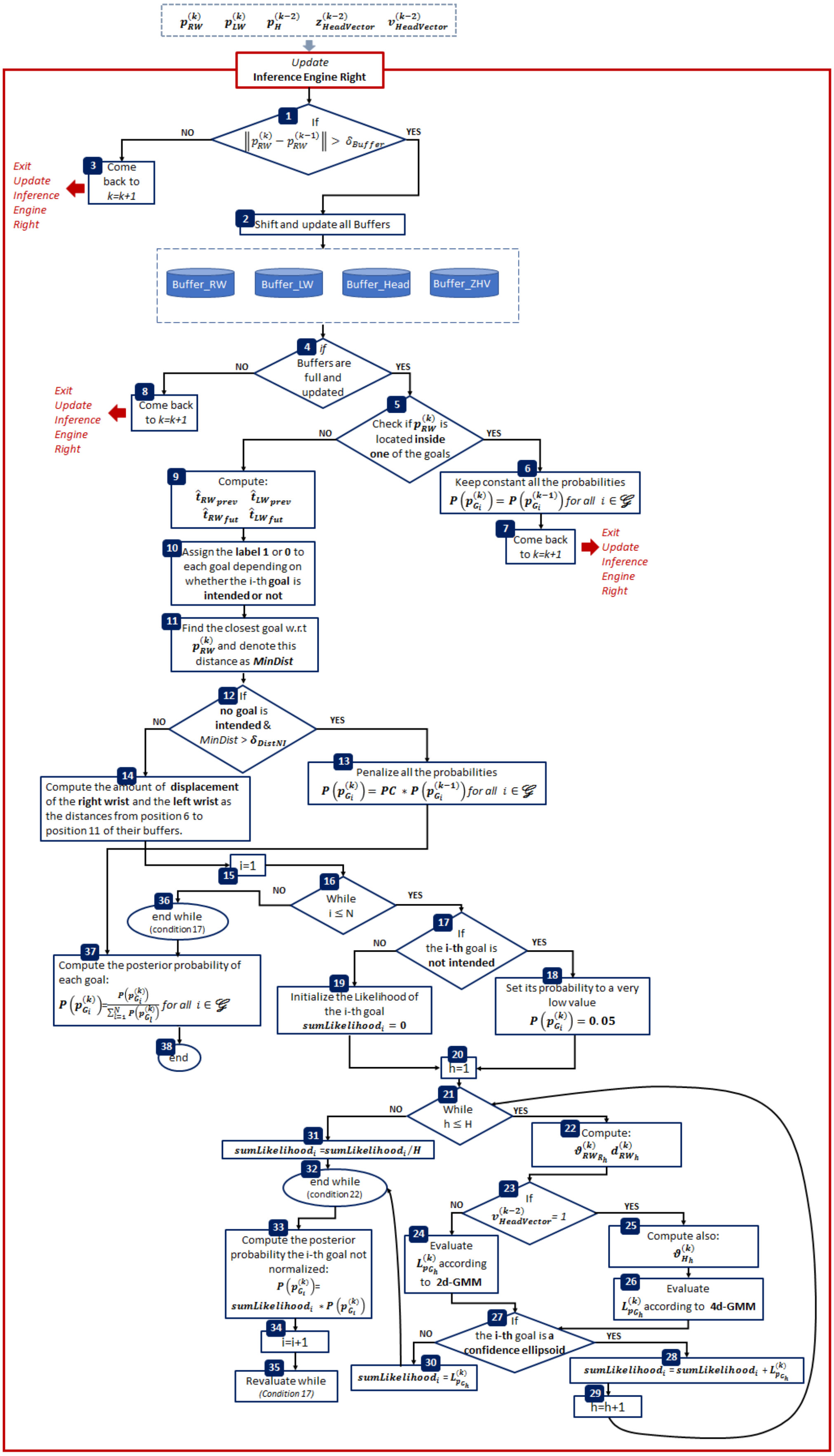

$p_{RW}^{(k)}$ $p_{LW}^{(k)}$ $p_{H}^{(k-2)}$ $z_{HeadVector}^{(k-2)}$ $v_{HeadVector}^{(k-2)}$
Update
Inference Engine Right
1 If $\left\| p_{RW}^{(k)} - p_{RW}^{(k-1)} \right\| > \delta_{Buffer}$
NO
YES
3 Come back to k=k+1
Exit Update Inference Engine Right
2 Shift and update all Buffers
Buffer_RW
Buffer_LW
Buffer_Head
Buffer_ZHV
4 If Buffers are full and updated
NO
YES
8 Come back to k=k+1
Exit Update Inference Engine Right
5 Check if $p_{RW}^{(k)}$ is located inside one of the goals
NO
YES
6 Keep constant all the probabilities $P\left(p_{G_i}^{(k)}\right) = P\left(p_{G_i}^{(k-1)}\right)$ for all $i \in \mathscr{G}$
7 Come back to k=k+1
Exit Update Inference Engine Right
9 Compute: $\hat{t}_{RW\,prev}$ $\hat{t}_{LW\,prev}$ $\hat{t}_{RW\,fut}$ $\hat{t}_{LW\,fut}$
10 Assign the label 1 or 0 to each goal depending on whether the i-th goal is intended or not
11 Find the closest goal w.r.t $p_{RW}^{(k)}$ and denote this distance as MinDist
12 If no goal is intended & MinDist > δ_{DistNI}
NO
YES
13 Penalize all the probabilities $P\left(p_{G_i}^{(k)}\right) = PC * P\left(p_{G_i}^{(k-1)}\right)$ for all $i \in \mathscr{G}$
14 Compute the amount of displacement of the right wrist and the left wrist as the distances from position 6 to position 11 of their buffers.
i=1
15
16 While i ≤ N
NO
YES
36 end while (condition 17)
17 If the i-th goal is not intended
NO
YES
37 Compute the posterior probability of each goal: $P\left(p_{G_i}^{(k)}\right) = \frac{P\left(p_{G_i}^{(k)}\right)}{\sum_{i=1}^{N} P\left(p_{G_i}^{(k)}\right)}$ for all $i \in \mathscr{G}$
19 Initialize the Likelihood of the i-th goal $sumLikelihood_i = 0$
18 Set its probability to a very low value $P\left(p_{G_i}^{(k)}\right) = 0.05$
38 end
20 h=1
21 While h ≤ H
NO
YES
31 $sumLikelihood_i = sumLikelihood_i / H$
22 Compute: $\vartheta_{RW_{R_h}}^{(k)}$ $d_{RW_h}^{(k)}$
32 end while (condition 22)
23 If $v_{HeadVector}^{(k-2)} = 1$
NO
YES
33 Compute the posterior probability the i-th goal not normalized: $P\left(p_{G_i}^{(k)}\right) = sumLikelihood_i * P\left(p_{G_i}^{(k)}\right)$
24 Evaluate $L_{p_{G_h}}^{(k)}$ according to 2d-GMM
25 Compute also: $\vartheta_{H_h}^{(k)}$
26 Evaluate $L_{p_{G_h}}^{(k)}$ according to 4d-GMM
27 If the i-th goal is a confidence ellipsoid
NO
YES
34 i=i+1
35 Revaluate while (Condition 17)
30 $sumLikelihood_i = L_{p_{G_h}}^{(k)}$
28 $sumLikelihood_i = sumLikelihood_i + L_{p_{G_h}}^{(k)}$
29 h=h+1

Figure 4.18: Flow diagram of Update Inference Engine Right. M is the number of sample observed, N is the number of targets and H is the number of 'goal samples'

5

Identification and modelling of the target positions

5.1 Introduction

In the previous chapters the location of the goals has been always considered given and described by means of a three-dimensional vector that represented the spatial coordinates associated with the center of mass of each target.

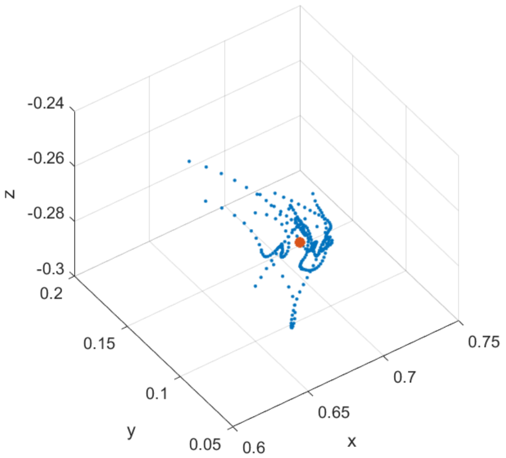


Figure 5.1: In the previous discussions each goal position was represented as the c.om (red point) of a set of positions (blue points cloud)

However, it could be possible that the goal positions within the collaborative workspace are not precisely a priori known and must be identified. Moreover, in

a real context each goal is represented by a physical object that obviously has its own shape and dimension. Therefore, identifying a goal with just its center of mass can sometimes result limiting.
Given the above, it may be interesting to find a way to describe and represent the target not only through the object's center of mass but also depending on how its dispersion is distributed in the space.
To do that, firstly a density-based clustering algorithm has been applied to all the set of wrist positions retrieved by the Kinect, then, for the sake of simplicity each goal position is assumed to be distributed as a multivariate Gaussian whose mean vector and covariance matrix are the ones associated with each group of clustered data. This will be addressed in the following section.

5.2 Application of OPTICS

As set out above, the fist step for identifying the target positions requires an offline phase where the operator which is intended to cooperate with the robot performs a sequence of motions so as to reach with his left or right wrist different points of each physical object and cover the majority of the object's surface. After this phase, the Kinect camera returns a point cloud that represents the sequence of positions assumed by the operator's wrist during his reaching motions towards the goals. The wrist positions tracked by the Kinect camera are shown in Figure 5.2.

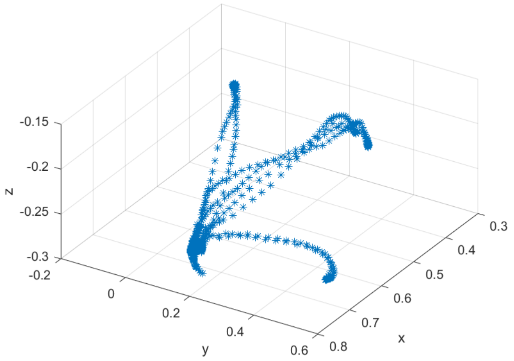


Figure 5.2: The Kinect camera retrieves the set of the operator's wrist positions tracked during his reaching motions: the retrieved right wrist position are reported in this figure

Looking at Figure 5.2 the set of points representing the target positions can be recognized by visual inspection as the most dense areas of the point cloud, so they can be easily distinguished from the set of position describing the executed trajectories.
In view of these considerations it seems reasonable to use a density-based clustering technique to identify the goals. In fact, the application of a clustering approach is strongly suggested in those circumstances where we are intended to derive the natural grouping or structure of data.

Given these assumptions, the use of OPTICS, [1], seems to be adequate. In fact it is a density-based clustering algorithm that, given a point cloud characterized by varying density regions, allows the identification of meaningful clusters, in the sense that is capable of recognizing and group together the data that are concentrated in some areas of the space. Indeed, the key idea of a clustering algorithm is to evaluate the density of the points located in the neighborhood of a fixed radius ϵ that is determined by the algorithm on the basis of the value of parameters received as inputs.
The term 'OPTICS' is an acronym for Ordering Points To Identify The Clustering Algorithm and referes to the procedure followed by this algorithm to perform the clustering.
In fact, OPTICS receives as inputs:

- the mxn matrix of the data, where m is the number of objects and n is the number of variables;
- k which is a parameter that represents the number of objects in a neighborhood of the selected object, namely, it defines the minimal number of objects that can be considered part of cluster.

Once received these inputs, the algorithm starts ordering the initial dataset so that the points which are located close to each other in the space become also close in the ordering. This means that the new ordering represents the density-based clustering structure of the input data. Moreover each point is labelled with a number that represents the cluster it belongs to. If the point under analysis does not belong to any cluster, it is considered as noise and it is labelled with zero. Therefore, the points that have the same label are grouped together and create a cluster.

The result of the application of the clustering is shown in Figure 5.3:

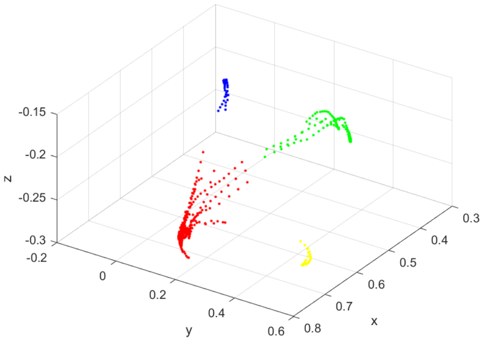


Figure 5.3: The clustering algorithm groups together a certain of positions according to a density-based criterion

However, it can happen that the clustering algorithm considers as being part of the cluster some positions that, by visual inspection, results to be residual noise that could be further removed, as shown in Figure 5.4 .

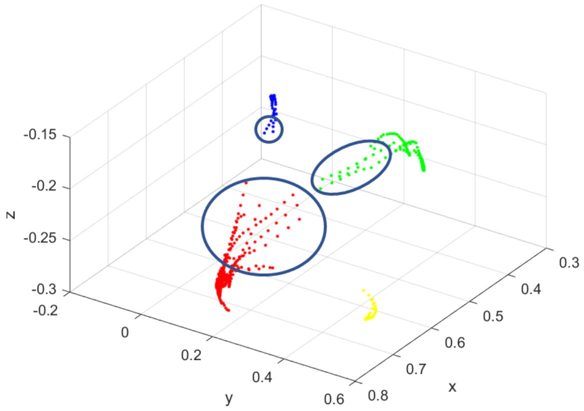


Figure 5.4: Samples that need to be a priori removed before applying OPTICS

In order to solve the problem the adopted technique is to make a prefiltering of the data so as to obtain a reduction of the initial dataset and, hopefully, the removal of the most significant component of the noise. This prefiltering of the initial samples can be performed according to the following consideration: since the goal positions represents the arrival points of each reaching motion, so, the set of points where the human wrist stopped, it is possible to make an a priori selection of the candidate goal positions by evaluating the distance

between each two consecutive acquired samples.
Indeed, once a reasonable threshold δ_{stPos} has been selected, it can be evaluated whether the distance between the k_{th} and the $k+1_{th}$ measure overcomes the threshold. In this case the k_{th} point is rejected, otherwise it is considered a candidate goal position.

$$if \begin{cases} ||p_W^{(k+1)} - p_W^{(k)}|| \geq \delta_{stPos} \implies \text{reject } p_W^{(k)} \\ ||p_W^{(k+1)} - p_W^{(k)}|| < \delta_{stPos} \implies \text{accept } p_W^{(k)} \end{cases} \tag{5.1}$$

The application in cascade of the pre-selection and the clustering algorithm leads to the result shown in Figure 5.5:

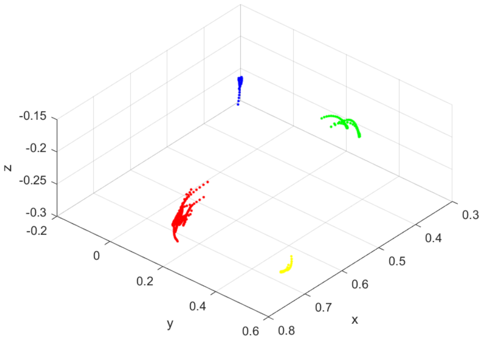


Figure 5.5: The application of the prefiltering and of OPTICS in cascade makes it possible to identify in a efficient way the group of points representing the goal positions

5.3 The confidence ellipsoids

Once the sets of points representing the goal positions have been identified by the clustering algorithm, it is useful to find a method that allows us to describe the region of the space (volume) which represents each goal position.
In this way it will be possible to evaluate (refer to the initial step of the algorithm described in section 4.5) whether or not each position retrieved by means of the Kinect camera is inside one of the goal volumes.

A viable solution is the creation of the so-called 'confidence ellipsoid'. Assuming that the each group of data returned by the clustering algorithm is normally distributed in the 3D space, it is possible to derive for each obtained cluster the volume that contains an arbitrary percentage δ_{Perc} of the considered group of samples. The δ_{Perc} - confidence ellipsoid is the name of the entity which defines that volume.
In fact, the probability density function of a multivariate normal distribution is characterized by surfaces of equal density, which are represented by ellipsoids.

Hence, for each i_{th} group of clustered data, in order to compute the δ_{Perc}-confidence ellipsoid associated with the i_{th} cluster dataset, the following steps can be performed:

- Compute the mean vector μ_i and the covariance matrix Σ_i associated with the i_{th} cluster dataset:

$$\begin{bmatrix} \mathbf{x_i} \\ \mathbf{y_i} \\ \mathbf{z_i} \end{bmatrix} \sim N(\begin{bmatrix} \mu_{x_i} \\ \mu_{y_i} \\ \mu_{z_i} \end{bmatrix}, \begin{bmatrix} var(x)_i & cov(x,y)_i & cov(x,z)_i \\ cov(y,x)_i & var(y)_i & cov(y,z)_i \\ cov(z,x)_i & cov(z,y)_i & var(z)_i \end{bmatrix}) \tag{5.2}$$

$$\mu_i = \begin{bmatrix} \mu_{x_i} \\ \mu_{y_i} \\ \mu_{z_i} \end{bmatrix} \qquad \Sigma_i = \begin{bmatrix} var(x)_i & cov(x,y)_i & cov(x,z)_i \\ cov(y,x)_i & var(y)_i & cov(y,z)_i \\ cov(z,x)_i & cov(z,y)_i & var(z)_i \end{bmatrix} \tag{5.3}$$

- compute the eigenvalues and eigenvector associated with each i_{th} group of data.
  Where the matrix of eigenvalues is:

$$\Lambda_i = \begin{bmatrix} \lambda_{1_i} & 0 & 0 \\ 0 & \lambda_{2_i} & 0 \\ 0 & 0 & \lambda_{3_i} \end{bmatrix} \tag{5.4}$$

  where λ_{1_i}, λ_{2_i}, λ_{3_i} represent the eigenvalues and describe the magnitude of data spread along the Cartesian x-axis, y-axis and z-axis respectively. Therefore they represent the variance of the data along those directions.

  And the matrix of eigenvectors is:

$$U_i = \left[u_{1_i} | u_{2_i} | u_{3_i} \right] \tag{5.5}$$

where u_{1_i} is the eigenvector associated with the first eigenvalue, u_{2_i} the eigenvector associated with the second eigenvalue and u_{3_i} the one that is related to the third eigenvalue.

- define a temporary reference system centered in zero whose axes coincides with the canonical x,y,z-axes. Let us denote this frame with number 0.
- compute the ellipsoid that is centred in zero, whose semi-axes coincide with the directions of frame zero and semiaxes' magnitude correspond to the retrieved eigenvalues:

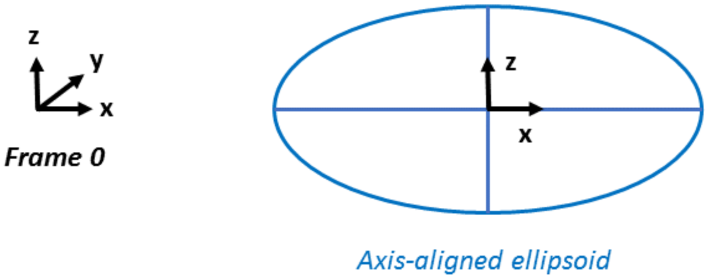


Figure 5.6: Schematic picture of the zero mean axis-aligned ellipsoid

In order to compute this ellipsoid, let us recall the equation of a zero-mean ellipsoid whose semi-axes are aligned with the directions of the usual Cartesian reference system.

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 = 1 \tag{5.6}$$

where a, b and c are the lengths of the ellipsoid semi-axes along the Cartesian x,y,z axes. Since, according to the previous considerations, the magnitude of the semiaxes is the variance of the data that, in this case coincides with the eigenvalues of the covariance matrix, equation 5.6 can be re-written as:

$$\left(\frac{x}{\lambda_1}\right)^2 + \left(\frac{y}{\lambda_2}\right)^2 + \left(\frac{z}{\lambda_3}\right)^2 = 1 \tag{5.7}$$

Since each data point is considered to be extracted by a trivariate Gaussian distribution characterized, for the time being, by a diagonal covariance

matrix and zero mean, the random variables x, y and z can be considered normally distributed too.
Hence, in view of these considerations, the terms $\frac{x}{\lambda_1}$, $\frac{y}{\lambda_2}$, $\frac{z}{\lambda_3}$ will correspond each to a standard normal:

$$\frac{x}{\lambda_1} \sim \mathscr{N}(0,1) \tag{5.8}$$

$$\frac{y}{\lambda_2} \sim \mathscr{N}(0,1) \tag{5.9}$$

$$\frac{z}{\lambda_3} \sim \mathscr{N}(0,1) \tag{5.10}$$

Therefore, recalling that the square of standard Normal random variable is also known as a 1-degree of freedom Chi-square distribution:

$$\left(\frac{x}{\lambda_1}\right)^2 \sim (\mathscr{N}(0,1))^2 \sim \chi_1^2 \tag{5.11}$$

$$\left(\frac{y}{\lambda_2}\right)^2 \sim (\mathscr{N}(0,1))^2 \sim \chi_1^2 \tag{5.12}$$

$$\left(\frac{z}{\lambda_3}\right)^2 \sim (\mathscr{N}(0,1))^2 \sim \chi_1^2 \tag{5.13}$$

where expression χ_1^2 represents the 1-degree of freedom Chi-square distribution. Therefore, equation 5.7 defines the sum of three 1-degree of freedom Chi-square distributions. This sum S can be also expressed as 3-degrees of freedom Chi-Square Distribution.

Since we are interested in determining that ellipsoid whose dimension is defined and scaled according to a prescribed confidence level δ_{Perc}, equation 5.7 should be modified accordingly.

$$\left(\frac{x}{\lambda_1}\right)^2 + \left(\frac{y}{\lambda_2}\right)^2 + \left(\frac{z}{\lambda_3}\right)^2 = S \tag{5.14}$$

Let us set δ_{Perc} to be equal to 95%.
Hence, since we are interested in the 95% confidence interval, the value

of that sum S can be determined by evaluating that S such that the probability that S is less then or equal to a specific value is equal to δ_{Perc}, as follows:

$$P(S \leq \chi^2_{3,\delta_{Perc}}) = \delta_{Perc} \tag{5.15}$$

Hence S can be easily obtained by looking at the Chi-square cumulative probability table.
The cumulative probability of a $\chi^2_{3,0.95}$ corresponds to 7.815. Hence the equation of the confidence ellipsoid with respect to frame 0 can be rewritten as:

$$\left(\frac{x}{\lambda_1}\right)^2 + \left(\frac{y}{\lambda_2}\right)^2 + \left(\frac{z}{\lambda_3}\right)^2 = 7.815 \tag{5.16}$$

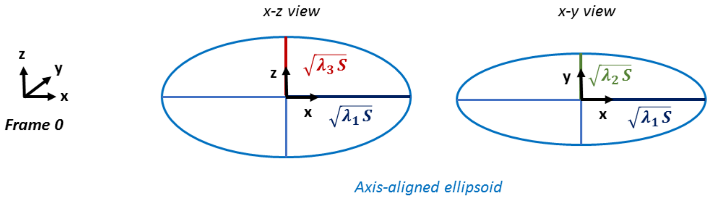


Figure 5.7: Zero mean confidence ellipsoid scaled according to the desired confidence level and expressed with respect to frame 0

Thus, for an axis-aligned 95% confidence ellipsoid the length of the i_{th} semi-axis is equal to $\sqrt{\lambda_i S}$ as shown in Figure 5.7.

- in order to find the 95% confidence ellipsoid that represents the original cluster dataset, it is possible to perform a roto-translation, as will be described afterwards.

Let us define a new reference system, denoted as frame 1, that is centered in μ_i (refers to equation 5.3) and oriented according to the directions of the eigenvectors matrix (refer to equation 5.5).
Let us also define the rotation matrix R^0_1 that describes the rotation of frame 1 with respect to frame 0. This rotation matrix coincides with the eigenvectors matrix.
Hence, defining the homogeneous transformation matrix A^0_1 as follows:

$$A_1^0 = \begin{bmatrix} u_{1_i} & u_{2_i} & u_{3_i} & \mu_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.17}$$

it holds that:

$$\begin{bmatrix} x^{(0)} \\ y^{(0)} \\ z^{(0)} \\ 1 \end{bmatrix} = A_1^0 \begin{bmatrix} x^{(1)} \\ y^{(1)} \\ z^{(1)} \\ 1 \end{bmatrix} \tag{5.18}$$

then from equation 5.18 it is possible to find $x^{(1)}$, $y^{(1)}$ and $z^{(1)}$ which are the three-dimensional coordinates of a generic point belonging to the confidence ellipsoid expressed with respect to frame 1.

Hence the ellipsoid shown in Figure 5.8 is obtained.

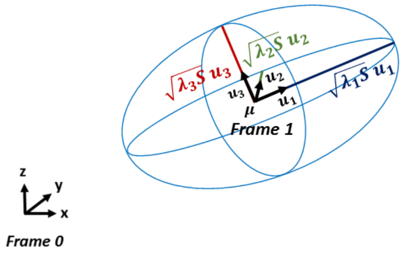

$\sqrt{\lambda_3 S}\, u_3$
$\sqrt{\lambda_2 S}\, u_2$
$\sqrt{\lambda_1 S}\, u_1$
u_2
u_3
u_1
μ
Frame 1
z
y
x
Frame 0

Figure 5.8: Schematic illustration of the ellipsoid obtained by applying a roto-translation of the initial zero mean axis-aligned ellipsoid

An example of the 95% confidence ellipsoid obtained for one of clusters described in section 5.2 is shown in Figure 5.9.

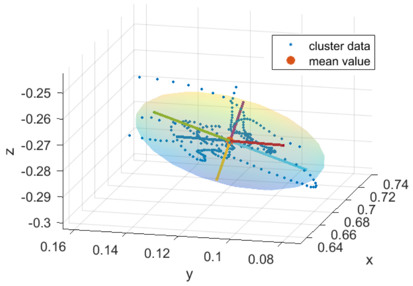


Figure 5.9: The 95% confidence ellipsoid stores the 95% of each cluster's data

Hence, the confidence ellipsoids that represent the volumes associated with each initial clustered positions (see Figure 5.5) are shown in Figure 5.10.

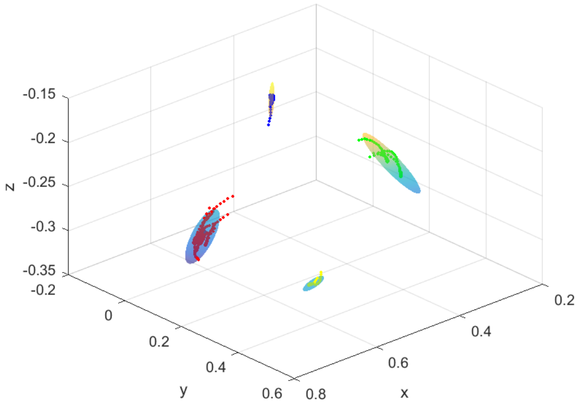


Figure 5.10: Confidence ellipoids associated with each group fo data returned by the clustering algorithm

In order to evaluate whether or not each new retrived wrist position lays inside one of the ellipsoids (refer to section 4.5), it is firstly performed a rototranslation that allows to express the new acquired skeletal position with respect to the reference frame associated with the considered ellipsoid; then, evaluating whether the position is inside it or not is straightforward.

5.4 Definition of the ‘goal samples’

As expressed in section 4.3 the computation of the feature vector Φ_i (refer to 4.2 and 4.3) and the evaluation of the likelihood function requires the knowledge of the exact position of each i_{th} goal.
Since the exact goal positions are not precisely known, each target is actually described as a random variable having a certain probability distribution, as already explained in this chapter. Then, in order to take into consideration the uncertainty affecting each target's center of mass, the inference algorithm considers a set of possible locations drawn from the stochastic distribution describing each goal position. Hence, each point belonging to the i_{th} set can be used to represent an hypothesis on the exact location of the i_{th} target, as will be explained afterwards.

To characterize this uncertainty when evaluating the likelihood of the observation under the hypothesis that the intended target is the i_{th}, it is then possible to define a number of points that are randomly sampled from the multivariate Gaussian Distribution (described by μ_i and Σ_i) associated with the i_{th} target. These points are referred as 'goal samples'. Let us assume that a total number of H 'goal samples' are obtained, as shown in Figure 5.11.

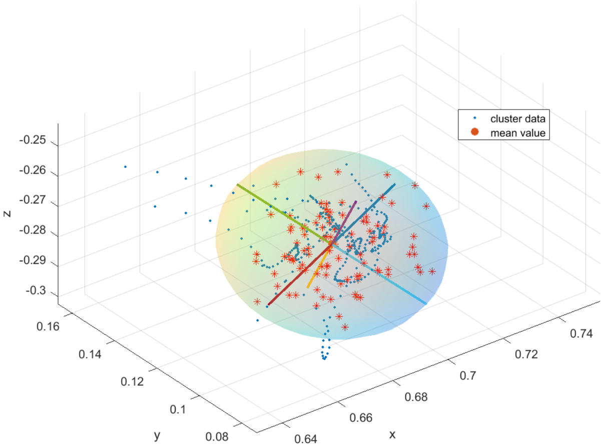


Figure 5.11: Illustration of the goal samples randomly generated according to the multivariate Gaussian parameters that describes the cluster's data

The uncertainty on the exact goals locations is evident by observing that, even if the coordinates of the position of the intended goal were precisely known, the human hand, when heading to that target, would not always reach the same exact position during its reaching motions. In this regard, the set of 'goal samples' associated with goal i represents the spread of the possible positions of arrival of the human hand within the volume of the i_{th} target. Let us denote the h_{th} goal sample, out of the H associated with the i_{th} goal as gs_{hi}. As a consequence, each gs_{hi} is considered as a plausible hypothesis about the true location of the i_{th} target. Hence it is possible to compute for each h_{th} sample belonging to the i_{th} goal, its likelihood function $L_{h_{pG_i}}$ and then evaluate the likelihood associated with the i_{th} goal as indicated in equation 4.47.

6

Comparison of the implemented inference strategies

6.1 Introduction

In this chapter an overview of the results achieved through the application of the intention inference algorithm will be presented. The metrics that will be used to evaluate the performance described so far will be the distance at which the considered goal is correctly recognized and the percentages of false positives and true negatives. Ultimately, the advantages and disadvantages of the various implementations of the algorithm will be discussed and compared with the performance attained in [34].

6.2 Analysis of the different formulations of the algorithm

In the previous chapters the methodology through which the algorithm is capable of performing the inference process has been analyzed in detail. Indeed, it has been explained that the algorithm returns, for each iteration and based on the observed features and the prior probabilities of each goal, how the probability distribution is split among the goals.
As a consequence, at each k_{th} iteration, the estimated intended goal will correspond to the one that has acquired the highest value of probability.

In order to offer the reader a deeper insight on how the algorithm works and on the output produced, Figure 6.1 reports the evolution of probabilities during some reaching movements.

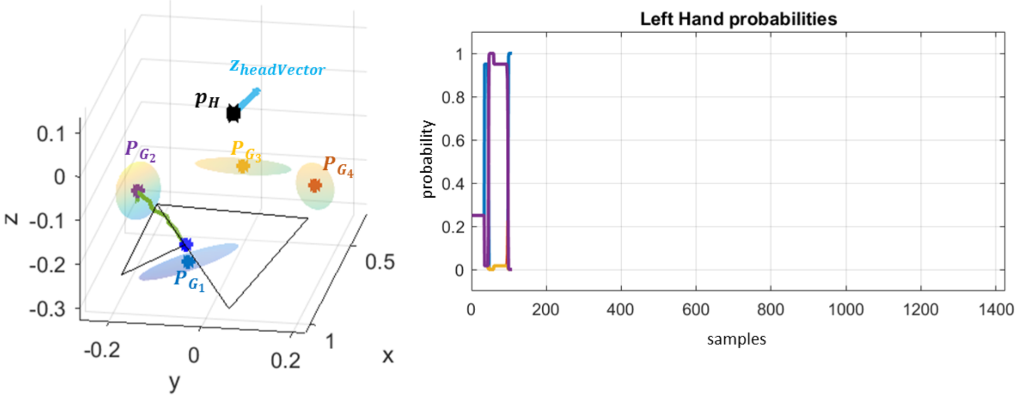


Figure 6.1: Schematic illustration of the evolution of probabilities returned by the inference algorithm at each iteration during some reaching movements. On the left hand side the upper body of the human operator which is performing a certain task can be visualized, while on the right hand side the corresponding estimated goal while the operator is moving is illustrated. The colors shown in the right hand plot correspond to the ones of the goal centers of mass illustrated on the left hand side. Here the initial phase of the inference algorithm is shown

Figure 6.1 represents the results obtained when inference is made on the left wrist by analyzing the first 100 detected samples. By looking at the plot on the right, it is clearly visible what has been explained in Chapter 4: initially the probability is uniformly distributed over the number of the goals. This aspect is evident by looking at the plot on the right hand side where a segment can be seen in correspondence of 0.25 for about the first 50 samples. As the buffer becomes full, the algorithm starts making inference and recognizes that the operator's hand is standing initially still in home position (blu goal). As, the operator starts moving from goal 1 to goal 2 the probability associated with target 2, the purple one, starts growing. Finally, as it is visible looking at the green curve that describes the movement performed by the left wrist of the operator (in the left figure), the subject comes back to goal 1 (home position). It should be noticed that, as depicted in Figure 6.1, the fact that the operator's wrist is intended to reach goal 1 has been already recognized (notice the high value associated with blue probability on the right hand plot) before his left hand actually reached the considered goal, as shown in detail in Figure 6.2.

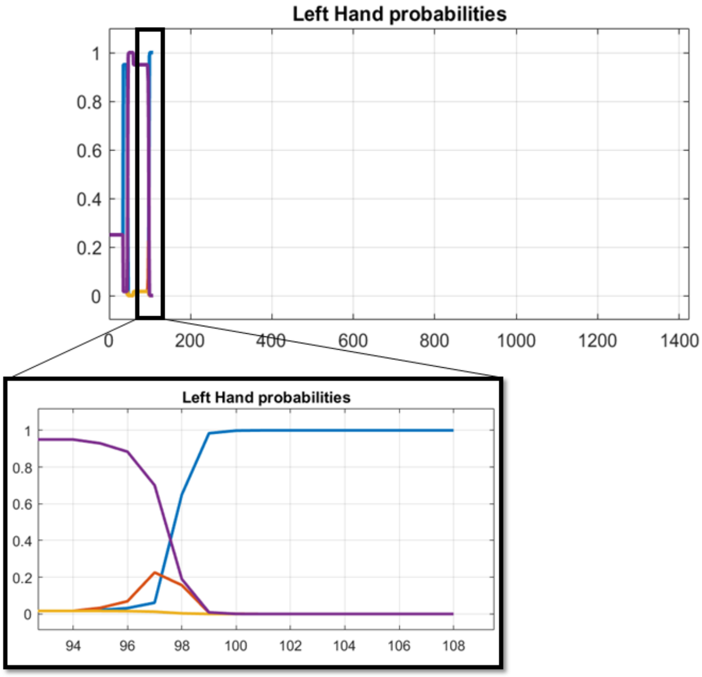


Figure 6.2: Detail of the probabilities shown in Figure 6.1

Obviuosly an equivalent result is returned, at the k_{th} iteration, by the intention inference algorithm that makes inference about the right hand.

Now we are interested in evaluating the contribution given by the application of each single type of methods described above, in order to highlight the benefits that their use could provide to enhance the inference process.
The basis for comparison is the approach proposed in [34] and described in Chapter 2.

6.2.1 Comparison of the different implemented predictive paths

The first proposed comparison focuses on the very first aspect analyzed in this Thesis: namely, we want to understand whether the use of a certain predictive path or another can actually help to improve the overall performance of the intention inference algorihm. Since, in this case, we are interested in analyzing only the contribution of the predictive path, without taking into consideration other potential avdvantageous features, the comparison will be structured as follows.

- all the goals will be considered equal to their centers of mass, as in [34];
- the vector of measurements is composed by a single observation: the wrist position p_W;
- the likelihood of observing p_W under the hypothesis of reaching the i_{th} target is evaluated according to a $\mathscr{N}(\mu, \sigma^2)$, following the idea of [34].

On the basis of those general aspects, the following method for computing the predicted path are compared:

1. minimum curvature path, hence the method proposed in [34], denoted as 'curve 1';
2. path resulting by minimizing the distance from the circumference centred in the shoulder position, denoted as 'curve 2';
3. parabolic path with recursive projection of the axis of symmetry, denoted as 'curve 3'.

Some reaching movements were recorded and only the first method was applied online, while the other two were simulated off-line by exploiting the same measurements retrieved during time. Two goals were kept monitored for this comparison, and the corresponding reaching motions performed by three different subjects were analyzed to extract meaningful informations.
The total number of reaching motion was 22.

The results are evaluated in terms of distance of recognition before reaching the desired goal. Here, the sample at which the i_{th} goal was considered recognized was the one for which the corresponding probability has overcome the threshold of 0.8, for each reaching motion.

The distribution of the distance at which the considered goal was correctly recognized are illustrated in Figure 6.3 for each method previously described.

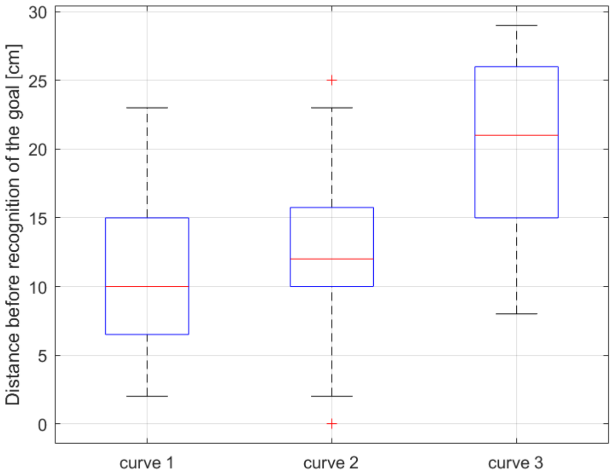
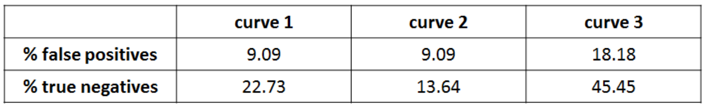


Figure 6.3: For each considered method (1,2,3) the boxplot shows the median (red line), the minimum value (black segment on the bottom) and maximum value (black segment on the top) obtained

Moreover, we define as an additional evaluation criterion: the number of false positives and of true negatives. The expression ‘false positive’ is used to describe the case where the probability of the i_{th} goal has risen beyond the 0.8 threshold, but the operator was not directed towards that target; while that of ‘true negative’ refers to the case where the operator was actually going to the i_{th} target but the corresponding probability did not reach the threshold.



	curve 1	**curve 2**	**curve 3**
% false positives	9.09	9.09	18.18
% true negatives	22.73	13.64	45.45

Figure 6.4: Percentage of false positives and true negatives with respect to the total number the operator went to the monitored goals, 22

These results confirm, to a certain extent, what the Fréchet distance analysis had already underlined: the performance obtained by using these predictive paths are quite similar.
In fact, even if the parabola seems to recognize earlier the correct goal with respect to method 1, it has to be considered that, by looking at 6.4 in about 45% of the overall reaching motions the goal was not recognized, hence its performance are worse with respect to both 1 and 2.
Equivalently, the performance of method 2 can be considered to be comparable to those of 1. Notice that 2 samples are beyond the whiskers in the boxplot

related to 2.
To further understand how similar are the performance of the three methods, it is possible to look at Figure 6.5 which shows that only few samples occur between the time when the 0.8 threshold is overcome by the first curve and when the same happens for the other curves.

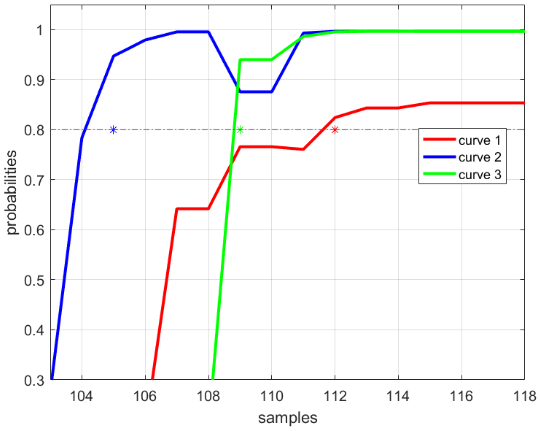


Figure 6.5: Probability plot that shows how fast the three methods raise beyond the recognition threshold (0.8). The difference also in term of number of samples is pretty negligible

By looking at these results it seems reasonable to conclude that the use of a predictive path, though well-structured and quite similar to the measured one, does not result to be so beneficial for what concerns the inference process. In other words, the use of a quite complex predictive path seems not be the key for obtaining a better intention estimate.

It should be recalled that up to now the predictive path was used to compute the predicted tangent $\hat{\mathbf{t}}_{W_{pred_i}}$ involved in the computation of the angle θ_{W_i} required to evaluate the likelihood:

$$\theta_{W_i} = \arccos((\hat{\mathbf{t}}_{W_{pred_i}})^T(\hat{\mathbf{t}}_{W_{fut}})) \tag{6.1}$$

The above results, by the way, seem to underline that it could be convenient to tackle the problem of computing θ_{W_i} in a different, more simple, way. Thus, we are interested in computing this angle by applying a method that does not require the use of a predictive path.

Hence, we evaluated the possibility of computing the generic angle θ_{W_i} in the most simple way, as follows:

$$\theta_{W_i} = \arccos((\hat{\mathbf{t}}_{G_i-W})^T(\hat{\mathbf{t}}_{W_{fut}})) \tag{6.2}$$

where

$$\hat{\mathbf{t}}_{G_i-W} = \frac{p_{G_i} - p_{W_{fut}}}{||p_{G_i} - p_{W_{fut}}||} \tag{6.3}$$

and the subscript W denotes the wrist position (left or right according to the hand we are making inference about).

In other words, the angle between the normalized vector connecting the last measured wrist position $p_{W_{fut}}$ to the center of mass of the i_{th} goal and last measured unit tangent vector $\hat{\mathbf{t}}_{W_{fut}}$ is used as a criterion for evaluating the likelihood, as shown in Figure 6.6:

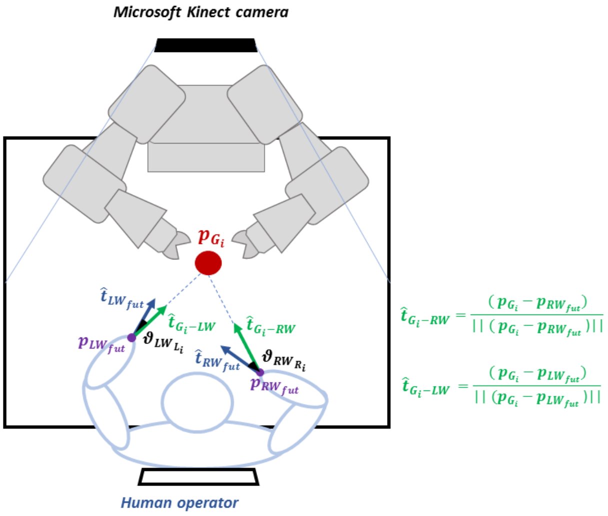


Figure 6.6: The angle between the measured unit tangent vector $\hat{\mathbf{t}}_{W_{fut}}$ and the normalized vector connecting the wrist position to the center of mass of the i_{th} target is now computed.

hence the computation of angles θ_{RW_R}, θ_{LW_R}, θ_{LW_L} and θ_{RW_L}, whose meaning was explained in chapter 4, is now changed accordingly.

The use of a very simple method for computing the angle, in fact, offers a two-fold benefit: on the one hand the computational complexity of the algorithm reduces, due to fact that the difficult computation of the predictive path is no more required; on the other hand, if this method is compared with the previous ones for the same data, the percentage of true negatives reduces a lot (it even goes to zero in this specific case), while the other statistics remain quite similar.

This aspect seems to offer a further motivation for using this method from now on.

6.2.2 Comparison of the use of single gaussian, 2D-GMM and 4D-GMM

As explained in Chapter 4 a further attempt to improve the performance of the basic intention inference process was the introduction of a larger number of observations to evaluate the likelihood and compute the posterior probability of each goal.
The new observations that had been included in the measurements' vector were:

- d_W the distance of the wirst from the target's center of mass;
- the joint observations of the wrist angles (for instance, θ_{RW_R} and θ_{LW_R} for the right wrist and the corresponfing one when making inference on the left wrist) as well as an estimate of the head orientation $z_{HeadVector}$ associated with the head position;
- the boolean value, $v_{HeadVector}$ always returned together with $z_{HeadVector}$, that expresses a measure of the gaze validity.

Thus, the second proposed comparison aims at highliting the contribution that these two groups of obervations can provide in addition to the wrist measurement. As already done in Chapter 4 we will denote as:

- 2D-GMM the likelihood function that jointly evaluates the measurements of wrist position p_W and d_W ;
- 4D-GMM the likelihood function that jointly evaluates all the available measurements. For instance, for what concerns the right hand it exploits: θ_{RW_R}, θ_{LW_R}, $z_{HeadVector}$, d_{RW}. Equivalently it is done for what concerns the left hand.

Therefore, still assuming that each goal under analysis coincides with its center of mass and computing the angles according to the method previously explained (refer to equation 6.2), the following formulations will be compared:

- method proposed in [34] (which considers only the wrist position as available measure), denoted as '1G';

- approach that updates the probabilities and computes the likelihood always using the 2D-GMM (simulating that the measure $z_{HeadVector}$ is unavailable for all the duration of the simulation). This approach is denoted as '2D-GMM';
- approach that updates the likelihood according to the 4D-GMM when $v_{HeadVector}$ is true and switches to the corresponding 2D-GMM when the gaze measure results to be not valid. (Obviously, in order to ensure that contribution of the measure $z_{HeadVector}$ was not negligible and could be correctly evaluated, it was checked that $v_{HeadVector}$ was true for at least 90% of the overall collected samples). This approach is denoted as '4D-GMM'.

The results are illustrated in Figure 6.7:

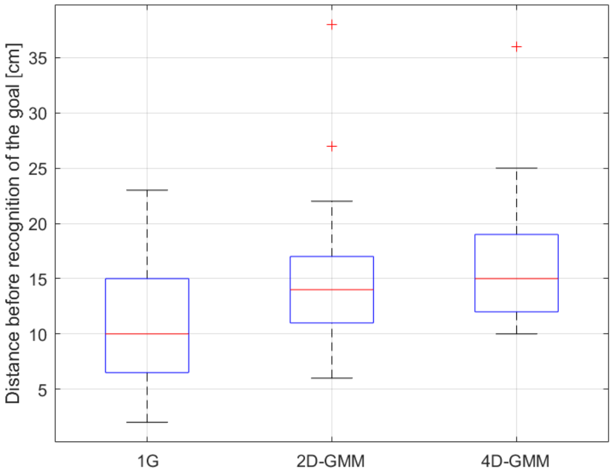
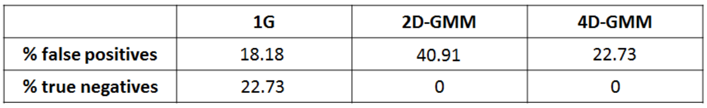


Figure 6.7: The boxplot shows that the performances in terms of distance of recognition improves as the number of observations increases

Moreover, also the performance in terms of false positives and true negatives must be evaluated. Their percentage, which has been extracted with respect to the total number of times (22) the operator went to the considered goal, are represented in Figure 6.8 :



	1G	2D-GMM	4D-GMM
% false positives	18.18	40.91	22.73
% true negatives	22.73	0	0

Figure 6.8: Percentage of false positives and true negatives with respect to the total number the operator went to the monitored goals, 22

By jointly looking at the results obtained in terms of distance of recogniton and robustness, the following conclusion can be drawn:

- including the distance into the set of measurements available for the computation of the likelihood has the advantage of largely reducing the number of true negatives that goes to zero. In other words, the introduction of the observation related to the distance helps to avoid the possibility that there exists a goal which is not recognized and improves the distance before recognition of the goal of about 40% with respect to the approach used in [34]. This aspects are obtained at the cost of increasing the reactivity of the algorithm that is highlighted by the growth of the number of false positives with respect to the base case, '1G'. Therefore, adding only the distance (beyond the wrist measure) seems to provide only partial advantages.

- if the information about the gaze is further introduced, the performance in terms of distance before correct recognition remains more or less the same (with respect to the case 2D-GMM). However, as indicated in Figure 6.8, it contributes in reducing of about 18% the number of false positives, while keeping fixed to zero the percentage of goals not recognized.

In conclusion it can be pointed out that adding d_W as a further available observation helps recognizing more quickly the intended goal and contributes in reducing largely the possiblity that an intended goal is not recognized, while the role of the gaze estimate is to guarantee a higher robustness during the inference process.

6.2.3 Managing the uncertainty for the goal locations

Until now the effectiveness of the so-called 4D-GMM method has been discussed but still tested on the case where the considered goals were represented by means of their centers of mass.
Now we propose a last comparison between the performance obtained using [34] and the ones that could be achieved by applying complete approach proposed in this Thesis. The expression ‘complete approach’ refers to the following aspects:

- the goals representation in terms of the set of ‘goal samples’;
- use of the likelihood function denoted as 4D-GMM (taking care to switch to the corresponding 2D-GMM when the measure of the gaze is not valid)
- computation of the angles required for evaluating the likelihood function according to the last explained method.

Therefore, let us indicate the approach used in as ‘Initial method’, and the one just described as ‘4D-GMM+GS’ where the acronym ‘GS’ refers to the goal description in terms of set of ‘goal samples’.

In this circumstance, in order to allow to understand the effectiveness of the use of the population of goal samples instead of a point-shaped goal, another dataset was required where the subject were asked to perform some reaching movements that did not lead necessarily to the center of mass of the goals. The results, that included 22 reaching motions, are reported in Figure 6.9:

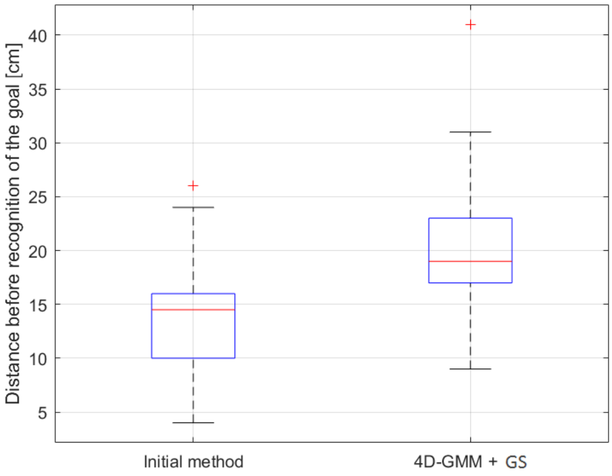
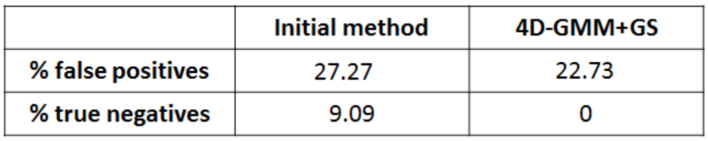


Figure 6.9: The two boxplots allows us to evaluate the percentage of overall improvement obtained through the application of approach ‘4D-GMM+GS’ with respect to the method described in [34], denoted as ‘initial method’



	Initial method	4D-GMM+GS
% false positives	27.27	22.73
% true negatives	9.09	0

Figure 6.10: Percentage of false positives and true negatives for the case ‘Initial method’, the one described in [34], and of the complete approach, ‘4D-GMM+GS’. The percentages are computed with respect to 22 reaching motions under analysis.

These results confirm what already observed in section 6.2.2: evaluating the distance avoid that a goal is not recognized but, conversely, increases a little the percentage of the so-called false positives obtained in the case 2D-GMM, introducing also the gaze estimate decreases that number again. To this extent, the information of the gaze helps increasing the robustness of the overall inference process, thus reducing the number of false positive and keeping to zero the numbers of true negatives. In terms of early recognition of the goal, it is clear that the performances achieved by the method described in [34] could be already considered quite satisfactory (the intended goal is recognized at almost one half of the entire path); in particular, considering that the goal is inferred by using an inference algorithm that works online without resorting to an offline trained model which is only applied online at a later stage.

However, in this circumstance, namely, when the uncertainty on the exact goal location is taken into account and a 4D-GMM is used, a small improvement in terms of distance of recognition can be noticed, without decreasing the robustness performance. This aspect seems to indicate that representing a goal in terms of confidence ellipsoid can help enhancing a little the overall performances.

In conclusion it should be recalled that the performance of the inference process, being based on the detected measurements, also depends on the quality of the measures retrieved by the sensing device. This aspect could partially explain the variability in terms of results and performance observed when the same method is applied to a different dataset.

7

Human-robot interaction: experimental test

7.1 Introduction

This chapter describes a collaborative task where the intention inference algorithm discussed in Chapter 4 is used to infer, at each iteration, the most likely human reaching target.

The purpose of this experimental section is to demonstrate how the improvements achieved by the new inference algorithm (described in Chapter 4) proved to be beneficial in a true collaborative scenario which involved the presence of the robot and where a vibrotactile feedback was sent to the operator (as soon as his intended goal was recognized) to inform him that the robot understood his intention. Indeed the importance of recognizing in adavanced the human's next action will be further highlighted. In fact, it will be shown that, particularly in a true collaborative task, the earlier the human intention is correctly recognized, the earlier the feedback can be sent and the better the overall cooperative process turned out to be.

Moreover even the advantages of the use of this wearable interface, capable of informing the operator about prediction's reliability, will be discussed and tested.

In fact, it is believed that, if the human operator can receive an acknowledgment that makes him aware of the current prediction process, confirming him that the prediction returned by the algorithm is correct too, a double advantage is obtained. On the one hand the operator can avoid keeping monitored the robot's complementary operation and can starts performing the next one, as soon as he receives the haptic feedback; on the other hand a reduction of the waiting time can be obtained.

7.2 Description of the collaborative experiment

During this collaborative task the human and the robot are required to cooperate in order to assemble a small box that is meant to host a USB pen drive, as shown in Figure 7.1:

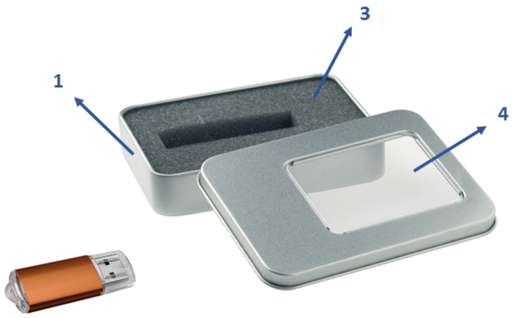


Figure 7.1: Illustration of the small box that has to be assembled during the collaborative operation and the pen drive which have to put in there

As represented in Figure 7.1, the box is composed of four different parts:

- the metallic base, denoted as '1' in the figure;
- a first thin layer of foam, not visibile in the figure;
- a second moulded and thick layer of foam, denoted as '3' in the figure;
- a metallic cover, denoted as '4' in the figure;

Since the metallic box is equipped with soft components (the two foam layers), that are difficult to be managed by a robot, the latter will be required to deal with a more resistent metallic component (the cover). In fact, the manipulator employed in the experiment is an ABB dual-arm robot YuMi equipped with a suction tool that allows it to easily grasp the complete assembled box.
The experimental set-up also involves the use of a Microsoft Kinect depth camera to track the operator's motions by detecting his skeletal points together with his head orientation $z_{HeadVector}$.
The operator's left hand can be further equipped with a vibrotactile ring which is intended to send him acknowledgments during the crucial phases of the collaboration, as it will be explained afterwards.
This vibrotactile ring, realized by the University of Siena, [9], and shown in Figure 7.2, is a wearable device which is also equipped with a small controller box

which contains a 4 mm vibration motor and is controlled through an Arduino Pro Mini5.

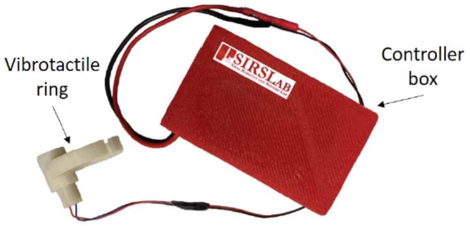


Figure 7.2: Illustration of the vibrotactile ring realized by the University of Siena. This ring can during the experiment is worn on the operator's left hand and the controller box is attached to the Velcro bracelet worn on the operator's left forearm. The communication with the ring is wireless

Both the sensing device, the vibrotactile ring and the dual-arm robot are connected to a CPU which implements the inference algorithm. Hence, at each iteration, the data retrieved by the Kinect camera are read by the CPU that, according to the logic of the inference algorithm described in Chapter 4, provides the estimated probability distribution over the goals and send commands to both the ring and the robot accordingly.

The layout of the collaborative task, represented in Figure 7.3, is composed by the following 5 operator's target positions:

1. home position, goal 1;
2. feeder of the moulded thick layer foam, goal 2;
3. feeder of the thin layer foam, goal 3;
4. collaborative station, goal 4;
5. feeder of the empty metallic boxes and pen drives goal 4.

The human-robot collaborative task is organized according to the following sequence of operations:

1. the operator takes the empty box from goal 5 and takes it back to the home position; then it fills it with the two foam layers and the USB stick;
2. the operator brings the filled box from the home position 1 towards the collaborative station, goal 4;

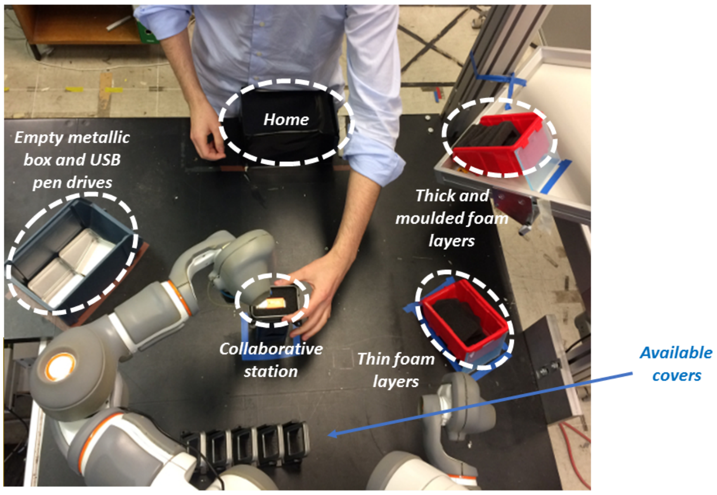

Home
Empty metallic box and USB pen drives
Thick and moulded foam layers
Collaborative station
Thin foam layers
Available covers

Figure 7.3: Layout of the experimental colaborative framework: the goal positions are marked with dotted ellipses

3. the robot puts the cover over the box;
4. the operator fixes the cover to the box with some tape and can do a new cycle starting from 1;
5. the robot takes away the finished box and puts it in a storage box.

The vibrotactile ring is exploited twice during the two critical phases of the collaborations: the first time to convey the information to the operator that, thanks to the inference algorithm, the robot understood his intention of bringing the filled box to the collaborative station (goal 4). This corresponds to the beginning of the operation 2. The second time the haptic device is used when the operator has finished covering the box with the tape (end of operation 4) to inform him that it understood that the operator has completed step 4.
The logic according to which a vibration is sent through the ring and the robot is allowed to start complementing the human operation, is obviously based on the probability distribution returned at each iteration by the inference algorithm and according to the evolution of the state machine visible is Figure 7.4.

Initially the state machine persists in state 0, meaning that no human has been detected on the scene by the Kinect camera. As soon as at least one human is tracked, the state machine moves to state 1.
The machine persists in this state until the probability associated with goal 4 (the collaborative station) raises beyond a predefined threshold, which is completely arbitrary and corresponds, in the current case, to 0.8. When this threshold is overcome, in fact, it means that the operator has completed the

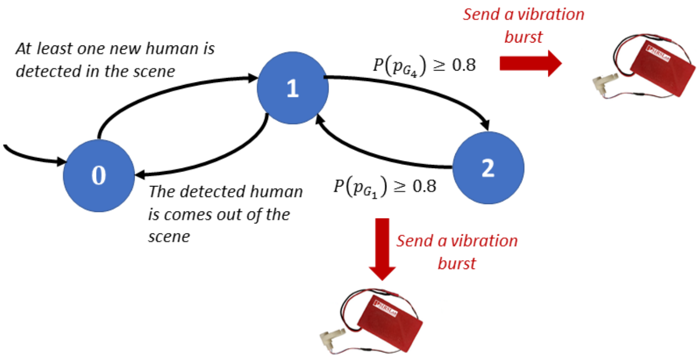

At least one new human is detected in the scene
1
$P(p_{G_4}) \geq 0.8$
Send a vibration burst
0
The detected human is comes out of the scene
$P(p_{G_1}) \geq 0.8$
2
Send a vibration burst

Figure 7.4: A finite state machine describes is used to describe the logic according to which a vibration burst is send to the human operator

set of operations that involved the box filling and he is intended to deliver the filled box towards the collaborative station to let the robot putting the cover. So, as soon as $P(p_{G_4}) > 0.8$ the machine goes to state 2 and send to the ring worn by the operator a vibration burst lasting 120 ms.
The machine remains in state 2 until the probability of goal 1 overcomes the threshold, meaning that the operator has finished fixing the cover to the box with the tape (operation 4 is terminated) and he is coming back to the home position (goal 1) to start a new cycle. The same type of vibration as before is sent to the operator.
Eventually, when the operator has completed all the cycles, he exits the scene and the state machine comes back to its initial state. The ring represents, in this way a reactive means of explicit communication between the robot and the human: in fact, through this the robot infroms the operator that it has understood his/her intention.

7.3 Results of the collaborative experiment

16 subjects were asked to take part to the experiments which consisted in performing (5 times for each subject) the cooperative task previously described. Half of the partecipants carried out the experiment by wearing the haptic device on their left hand and were instructed about the meaning of the vibration's burst; half performed the same experiment without it. In both groups only 5 out of 8 could be considered 'not skilled' since they declared they had never cooperated with a robot before and they were not familiar with the use of robots in general. The subject's left hand was tracked by the Kinect and the execution time of each trial was recorded.

In [34] it was already demonstrated, that the possibility of performing a early recognition of the human intention could improve the quality of the overall collaboration. However in [34] the operator was never informed about the correct estimate of its intention through a feedback of any type. Hence, the human was not explicitly aware whether the robot had estimated correctly his intention or not, unless he waited and see the next robot move. In fact, in that case, the robot was instructed to proceed by choosing the complementary human's assembly action to complete the overall task.

In view of the improvements (in terms of robustness and recognition distance) obtained by reformulating the overall inference algorithm as illustrated in the previous chapters, it is now possible to evaluate the further advantages that can be obtained by equipping the operator with a feedback that makes him aware about his/her estimated intention.

Also this time we want to compare the performance obtained by applying the following methods:

- method proposed in [34];
- method that computes the likelihood always according to the 2D-GMM;
- method that computes the likelihood according to the 4D-GMM when the gaze measure is valid and exploits the corresponding 2D-GMM otherwise;

The results are shown in Figure 7.5:

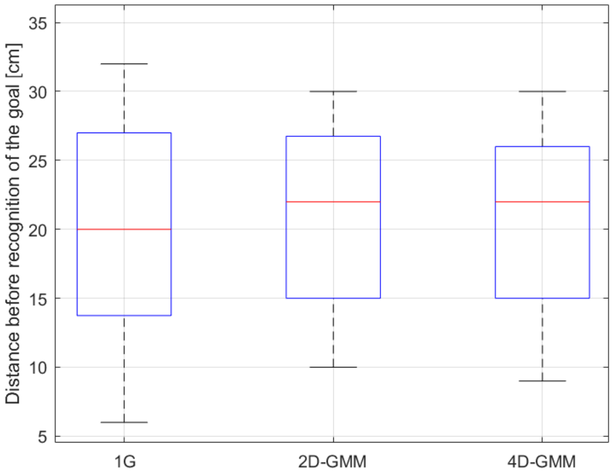
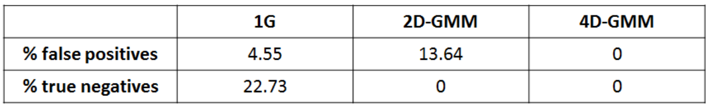


Figure 7.5: Comparison of the implemented inference methods for the case during the collaborative experiments. Distributions of the distance at which goal 4 was correctly recognized

for what concerns the robustness performance, one can in this case refer to table 7.6:



	1G	2D-GMM	4D-GMM
% false positives	4.55	13.64	0
% true negatives	22.73	0	0

Figure 7.6: Percentages of false positives and true negatives during the collaborative experiments. These results refers to a total number of 22 reaching motions towards goal 4.

Also in the case that involved the collaboration between the human operator and the robot the obtained results confirm what had already been observed even in absence of the robot (see section 6.2.2): hence the performances in terms of distance before recognition are quite similar for all the methods (even if a slight improvement can be observed in the case 4D-GMM with respect to the case 1G), however the introduction of the distance as a further observation contributes in a significant reduction of the possibility that an intended goal is not recognized (zero for the considered experiments), while the recognition distance does not decreases. This is obtained at the cost of decreasing of the robustness of the algorithm, thus increasing the number of false positive. The role of the additional gaze estimate is to improve the robustness of the algorithm

without worsening the abovementioned performance.
For what concerns the benefit of using the haptic feedback the cycle time can be analysed:

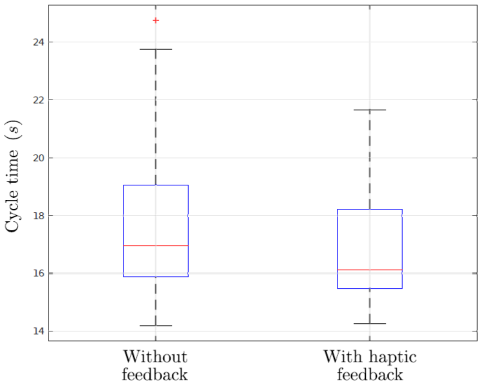


Figure 7.7: Cycle time with and without feedback for the all the partecipants

The performance achieved by the introduction of the vibrotactile ring when the overall population of skilled and non skilled partecipants is observed was a reduction of the variability cycle time. By looking at Figure 7.7 is also visible a small decrease of the average cycle time. Hovewer the reduction of the average cycle time in this case was smaller with respect to what is shown in Figure 7.8 that refers to non-skilled people only. The reason which could motivate this fact is that the assembly task previously presented is quite simple, as a consequence people with previous experience in robotics do not seem to take advantage of the haptic feedback.

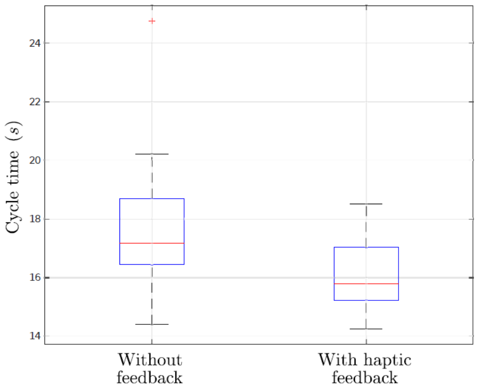


Figure 7.8: Cycle time with and without feedback for non-skilled partecipants

As already said, by observing the results shown in Figure 7.8 that considers only the group of non skilled partecipants, the reduction of the average cycle time when they received the feedback is clearly visible, while, this time, an equivalent decrease in terms of variability of the cycle time cannot be observed. By the way, the results obtained for non-skilled people are, to some extent, significant, since these perfomance could probably be representative of what could happen in a real industrial framework where the production lines change freequently. In that case, in fact, the operators usually have to learn the new sequence of operations very often without having a preliminarly experience in the new sequence of operation they will have to do; hence, to some extent, they could be considered as they were always non-skilled people.

8

Conclusions

8.1 Conclusions and future developments

In this Thesis a novel inference algorithm capable of estimating the operator's intention within a collaborative scenario was presented. The inference process is accomplished by exploiting a recursive Bayesian classifier which, at each iteration, provides the estimate of the most likely human hand reaching target among a prescribed and finite set of possible goal positions. In order to manage the uncertainty on the exact target's location, the proposed algorithm considers a set of possible positions drawn from a stochastic distribution which describes the goal location. This way the unpredictability on the actual goal position can assume a probabilistic meaning. Besides, the variability characterizing the possible positions of arrival of the operator's hand reaching motions can be taken properly into consideration.
The algorithm presented in this Thesis infers the most likely reaching target relying on the skeletal points retrieved by means of a Microsoft Kinect camera. This approach had already been adopted in a previous work described in [34], where the user's intention was, however, estimated based on the observed wrist measure only. The novel algorithm is structured in a way that allows to handle the situation where the operator simultaneously exploits both his hands to execute a certain task. This is practically done by updating, at each iteration, two different probability distributions: namely, it is made inference on the most likely reaching target from the perspective of the right hand and from the perspctive of the left hand. Furthermore, the novel inference algorithm extends the approach of [34], by taking into consideration a larger set of measurements that comprehends the joint observation of both hands' positions (independently on the one we are making inference on), the distance of the hand from the goal center of mass and an estimate of the gaze direction.

In fact it is claimed that this particular set of observations, when jointly exploited, can improve from different perspectives the overall performance of the inference algorithm. For instance, considering where both hands are located or are about to move, can enhance the evidence of where each one of them is directed to. Moreover, taking into account the wrist distance from the target center of mass can help discriminating the intended goal when more than one is almost equally likely; lastly, the head orientation can provide an additional clue on what the intended reaching target is and increasing the robustness of the overall inference process.

These expectations are confirmed: by using the novel approach, the so-called 4D-GMM and taking into account the uncertainty related to each goal position, as described so far, the distance before goal's recognition slightly improves of about 40% with respect to the corresponding result mentioned in [34]. This means that the goal is correctly recognized at approximately half of the hand reaching path. Moreover, the use of the observed distance between the wrist and the goal center of mass allows us to obtain a significant reduction of the percentage of true negatives. For instance, during the previously mentioned experimental test, for identical boundary conditions (same robotic cell, same goal locations), the percentage of true negatives resulted to be equal to zero with respect to the corresponding case where the algorithm of [34] was applied. In addition, the presence of the gaze estimate allows us to achieve greater robustness performance, decreasing significantly the percentage of eventual false positives.

Through an experimental campaign where 16 volunteers partecipated, the benefits of the improvements obtained after the actual reformulation of the inference algorithm were further highlighted and the importance of recognizing in advance where the human hand is heading to was further pointed out by the presence of the haptic feedback. Indeed, the proposed algorithm was applied to a true collaborative assembly task which included the use of a vibrotactile ring to send a feedback to the human operator that worn it.

Even in this case, similar results to those just mentioned were achieved. During the experimental test, half of the partecipants were equipped with a wearable vibroctile ring which sent the operator the haptic feedback to make him aware of when the robot infers correctly his intention. The contribution provided by the addition of the vibrotactile feedback was more evident in the case of non-skilled subjects, where the average execution time significantly decreased with respect to the case where the haptic device was not exploited.

The performance obtained by combining the presented inference algorithm with the wearable vibrotactile ring are quite satisfactory and seem to pave the way for considering their applicability on a variety of human-robot collaborative scenarios.

In fact, as highlighted by the experimental results, the better the inference process, the earlier the prediction of the operator's intention, the greater the time available to send a feeeback to the human and improve the quality of the collaboration. It should be recalled that an additional beneficial aspect of this inference algorithm is that it could be applied to whatever robotic cell, without requiring an additional a priori training phase.

Future works could address the possibility of using a single device, worn, for instance, on the operator's arm and capable of sending feedback to the human worker as well as helping in the process of retrieving its skeletal points by making use of some sensor fusion techniques to consider also the measurements coming from the Kinect camera.
Moreover, some pattern recognition techniques could be exploited to understand the usual sequence of goals reached by the operator when performing a certain task. This information could be then effectively used to improve the robustness of the overall inference process. In fact, if a high evidence about a certain sequence of goals was derived after the application of these techniques, at each iteration the intention inference algorithm could penalize the goal or the set of goals which, given the expected sequence, resulted to be less likely.

Bibliography

[1] M. Ankerst, M. M. Breunig, H. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. In ACM, editor, *Proceedings of the ACM SIGMOD '99 International Conference on Management of Data, Philadelphia*, 1999.

[2] M. Awais and D. Henrich. Human-robot collaboration by intention recognition using probabilistic state machine. In IEEE, editor, *Robotics in Alpe-Adria-Danube Region (RAAD), 2010 IEEE 19th International Workshop*, 2010.

[3] L. Bascetta, G. Ferretti, P. Rocco, H. Ardö, H. Bruyninckx, E. Demeester, and E. Di Lello. Towards safe human-robot interaction in robotic cells: an approach based on visual tracking and intention estimation. In IEEE, editor, *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[4] R. Bednarik, H. Vrzakova, and M. Hradis. What do you want to do next: a novel approach for intent prediction in gaze-based interaction. In ACM, editor, *Proceedings of the symposium on eye tracking research and applications*, 2012.

[5] G. Best and R. Fitch. Bayesian intention inference for trajectory prediction with an unknown goal destination. In IEEE, editor, *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference*, 2015.

[6] J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. April 1998.

[7] E. Bizzi, W. Accornero, W. Chapple, and N. Hogan. Posture control and trajectory formation during arm movement. *The Journal of Neuroscience*, 1984.

[8] M. Carrasco and X. Clady. Prediction of user's grasping intentions based on eye-hand coordination. In IEEE, editor, *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010.

[9] R. Casalino, C. Messeri, M. Pozzi, A. M. Zanchettin, P. Rocco, and D. Prattichizzo. Operator awareness in human-robot collaboration through wearable vibrotactile feedback. submitted for RA-L and IROS to the IEEE Robotics and Automation Letters (RA-L) on February 24, 2018.

[10] W. Chen, C. Xiong, and S. Yue. On configuration trajectory formation in spatiotemporal profile for reproducing human hand reaching movement. *IEEE Transactions on Cybernetics*, 2016.

[11] Y. Chen and M. R. Gupta. EM demystified: an expectation-maximization tutorial. February 2010.

[12] A. D. Dragan and S. S. Srinivasa. A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 2013.

[13] E. Driver and D. Morrell. Implementation of continuous bayesian networks using sum of weighted gaussians. In Besnard and Hanks, editors, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 1995.

[14] R. Durrett. Essential of stohastic processes. August 21, 2010.

[15] I. Epifani, L. Ladelli, and G. Posta. Appunti per il corso di calcolo delle probabilità. 2005.

[16] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience*, 1985.

[17] G. Hoffman and C. Breazeal. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In ACM, editor, *Proceedings of the ACM/IEEE international conference on Human robot interaction*, 2007.

[18] G James, D Witten, T Hastie, and R Tibshirani. *An introduction to statistical learning*. Springer, 2013.

[19] Y. Jiang and A. Saxena. Modeling High-Dimensional Humans for Activity Anticipation using Gaussian Process Latent crfs. *IEEE*, 2014.

[20] T. Kashima, K. Yanagihara, and I. Masao. Trajectory formation based on a human arm model with redundancy. In IEEE, editor, *2012 IEEE International Conference on Systems, Man, and Cybernetics*, 2012.

[21] R. Kelley, A. Tavakkoli, C. King, M. Nicolescu, and G. Bebis. Understanding Human Intentions via Hidden Markov Models in Autonomous Mobile

Robots. In ACM, editor, *Proceedings of the 3rd ACM/IEEE International Conference on Human robot interaction*, 2008.

[22] C. L. Kleinke. Gaze and eye contact: a research review. 1986.

[23] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[24] C. Lenz, A. Sotzek, T. Röder, H. Radrich, A. Knoll, M. Huber, and S. Glasauer. Human workflow analysis using 3d occupancy grid hand tracking in a human-robot collaboration scenario. In IEEE, editor, *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[25] R. Luo and D. Berenson. A framework for unsupervised online human reaching motion recognition and early prediciton. In IEEE, editor, *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[26] G. Mainprice, R. Hayne, and D. Berenson. Predicting human motion in collaborative tasks using inverse optimal control and iterative re-planning. In IEEE, editor, *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[27] J. Mainprice and D. Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In IEEE, editor, *2015 International Conference on Robotics and Automation (ICRA)*, 2015.

[28] K.P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, Fall 2002.

[29] R. Osu. Coordinate for trajectory formation of human multi-joint arm movement. In IEEE, editor, *International Workshop on robot and human communication*, 1993.

[30] A. V. Papadopoulos, L. Bascetta, and G. Ferretti. Generation of human walking paths. *Autonomous Robots*, 2016.

[31] S. Pellegrinelli, H. Admoni, S. Javdani, and S. Srinivasa. Human robot shared workspace collaboration via hindsight optimization. In IEEE, editor, *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference*, 2016.

[32] C. Pérez-D'Arpino and J. A. Shah. Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time

series classification. In IEEE, editor, *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[33] H. C. Ravichandar and A. P. Dani. Human intention inference using expectation-maximization algorithm with online model learning. *IEEE Transactions on Automation Science and Engineering*, 2016.

[34] P. Rocco and A. M. Zanchettin. Probabilistic inference of human arm reaching target for effective human-robot collaboration. In IEEE, editor, *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference*, 2017.

[35] P. Rocco and A. M. Zanchettin. Robotica: la collaborazione uomo-macchina per creare nuovo valore. *SISTEMI&IMPRESA*, gennaio–febbraio 2016.

[36] R. Scattolini. Process monitoring. 2016.

[37] Oliver C. Schrempf and Uwe D. Hanebeck. A generic model for estimating user intentions in human-robot cooperation. In ICINCO, editor, *ICINCO*, 2005.

[38] M. Suzuki, Y. Yamazaki, N. Mizuno, and K. Matsunami. Trajectory formation of the center-of-mass of the arm during reaching movements. *Neuroscience*, 1996.

[39] Y. Tamura, M. Sugi, J. Ota, and T. Arai. Estimation of User's Intention Inherent in the Movements of Hand and Eyes for the Deskwork Support System. In IEEE, editor, *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[40] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. *Biological Cybernetics*, 1989.

[41] Y. Uno, M. Kawato, Maeda Y., and Suzuki R. Repetitively structured cascade neural network model which gnerates an optimal arm trajectory. In IEEE, editor, *Proceeedings of the 28th Conference on Decision and Control*, 1989.

[42] N. N. Vo and A. F. Bobick. From Stochastic Grammar to Bayes Network: Probabilistic Parsing of Complex Activity. *IEEE*, 2014.

[43] Z. Zhang, A. Beck, and N. Magnenat-Thalmann. Human-like behavior generation based on head-arms model for robot tracking external targets

and body parts. In IEEE, editor, *IEEE Transactions onCybernetics*, 8 August 2015.