LIÈGE université
Sciences Appliquées

UNIVERSITY OF LIÈGE
FACULTY OF APPLIED SCIENCES

DOCTORAL COLLEGE IN AEROSPACE AND MECHANICS

Multibody and Mechatronic Systems Laboratory

DOCTORAL THESIS

A human hand motion tracking method for robot programming by demonstration

Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD) in Engineering Sciences

by

Robin Pellois

Supervisor: Prof. Olivier Brüls

March 2023

LIÈGE université
Sciences Appliquées

UNIVERSITY OF LIÈGE
FACULTY OF APPLIED SCIENCES

DOCTORAL COLLEGE IN AEROSPACE AND MECHANICS

Multibody and Mechatronic Systems Laboratory

DOCTORAL THESIS

A human hand motion tracking method for robot programming by demonstration

Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD) in Engineering Sciences

by

Robin Pellois

Supervisor: Prof. Olivier Brüls

March 2023

Résumé

La programmation par démonstration est une approche intuitive permettant de transmettre une tâche à un robot à partir d'une ou plusieurs démonstrations faites par un enseignant humain. L'acquisition des démonstrations nécessite cependant la résolution d'un problème de correspondance quand les systèmes sensitifs et moteurs de l'enseignant et de l'apprenant diffèrent. De nombreux travaux utilisent des démonstrations faites par kinesthésie, i.e., l'enseignant manipule directement le robot pour lui faire faire la tâche. Ce dernier enregistre ses mouvements grâce à ses propres encodeurs. De cette façon, le problème de correspondance est trivial. Lors de telles démonstrations, la dextérité de l'enseignant peut être altérée et impacter tout le processus de programmation par démonstration. Les méthodes d'acquisition de démonstration moins invalidantes pour l'enseignant nécessitent souvent des procédures spécifiques pour résoudre le problème de correspondance. Ainsi l'acquisition des démonstrations se base sur un compromis entre complexité de ces procédures, le niveau de précision des éléments enregistrés et la commodité pour l'enseignant. Cette thèse propose ainsi une méthode de mesure du mouvement humain par capteurs inertiels pour la programmation par démonstration de tâches de "pick-and-place". Les capteurs inertiels sont en effet pratiques et faciles à utiliser, mais sont d'une précision limitée. Nous étudions leur potentiel pour la programmation par démonstration.

Pour estimer la trajectoire de la main de l'enseignant, des capteurs inertiels sont placés sur son bras, son avant-bras et sa main afin d'estimer leurs orientations. Une méthode est proposée afin de compenser partiellement la dérive de l'estimation de l'orientation des capteurs autour de la direction de la gravité. Cette méthode, appelée "heading reset", est basée sur l'hypothèse que le capteur passe plusieurs fois par son azimut initial avec des phases stationnaires lors d'une démonstration. Cette méthode est implémentée dans un algorithme d'intégration et d'observation de vecteur. Des expériences illustrent les avantages du "heading reset".

Cette thèse développe ensuite une méthode complète de mesure des mouvements de la main humaine par capteurs inertiels (IHMT). Elle comprend une première procédure d'initialisation pour estimer l'orientation des capteurs par rapport aux segments du bras humain ainsi que l'orientation initiale des capteurs par rapport au repère de référence de l'humain. Cette procédure, consistant en une rotation et une position statique du bras tendu, est robuste au positionnement des capteurs. Une seconde procédure est proposée pour estimer la position de l'humain par rapport au robot et pour calibrer les paramètres de la méthode. Finalement, l'erreur moyenne sur la trajectoire de la main humaine est mesurée expérimentalement entre 28.5 mm et 61.8 mm, ce qui n'est cependant pas suffisant pour la programmation par démonstration.

Afin d'atteindre le niveau de précision nécessaire, une nouvelle méthode est développée afin de corriger la trajectoire de la main par IHMT à partir de données issues d'un système de vision, complémentaire des capteurs inertiels. Pour maintenir une certaine simplicité et robustesse, le système de vision ne suit que les objets et pas l'enseignant. La méthode de correction, basée sur des "Positions Of Interest (POIs)", est constituée de 3 étapes: l'identification des POIs dans les données issues des capteurs inertiels et du système de vision, puis l'association de POIs liées à la main et de POIs liées aux objets correspondant à la même action, et enfin, la correction de la trajectoire de la main à partir des paires de POIs. Finalement, la méthode IHMT corrigée est expérimentalement évaluée dans un processus complet de programmation par démonstration. Cette expérience montre l'avantage de la méthode proposée sur la kinesthésie dans le contexte de ce travail.

Abstract

Programming by demonstration (PbD) is an intuitive approach to impart a task to a robot from one or several demonstrations by the human teacher. The acquisition of the demonstrations involves the solution of the correspondence problem when the teacher and the learner differ in sensing and actuation. Kinesthetic guidance is widely used to perform demonstrations. With such a method, the robot is manipulated by the teacher and the demonstrations are recorded by the robot's encoders. In this way, the correspondence problem is trivial but the teacher dexterity is afflicted which may impact the PbD process. Methods that are more practical for the teacher usually require the identification of some mappings to solve the correspondence problem. The demonstration acquisition method is based on a compromise between the difficulty of identifying these mappings, the level of accuracy of the recorded elements and the user-friendliness and convenience for the teacher. This thesis proposes an inertial human motion tracking method based on inertial measurement units (IMUs) for PbD for pick-and-place tasks. Compared to kinesthetic guidance, IMUs are convenient and easy to use but can present a limited accuracy. Their potential for PbD applications is investigated.

To estimate the trajectory of the teacher's hand, 3 IMUs are placed on her/his arm segments (arm, forearm and hand) to estimate their orientations. A specific method is proposed to partially compensate the well-known drift of the sensor orientation estimation around the gravity direction by exploiting the particular configuration of the demonstration. This method, called heading reset, is based on the assumption that the sensor passes through its original heading with stationary phases several times during the demonstration. The heading reset is implemented in an integration and vector observation algorithm. Several experiments illustrate the advantages of this heading reset.

A comprehensive inertial human hand motion tracking (IHMT) method for PbD is then developed. It includes an initialization procedure to estimate the orientation of each sensor with respect to the human arm segment and the initial orientation of the sensor with respect to the teacher attached frame. The procedure involves a rotation and a static position of the extended arm. The measurement system is thus robust with respect to the positioning of the sensors on the segments. A procedure for estimating the position of the human teacher relative to the robot and a calibration procedure for the parameters of the method are also proposed. At the end, the error of the human hand trajectory is measured experimentally and is found in an interval between 28.5 mm and 61.8 mm. The mappings to solve the correspondence problem are identified. Unfortunately, the observed level of accuracy of this IHMT method is not sufficient for a PbD process.

In order to reach the necessary level of accuracy, a method is proposed to correct the hand trajectory obtained by IHMT using vision data. A vision system presents a certain complementarity with inertial sensors. For the sake of simplicity and robustness, the vision system only tracks the objects but not the teacher. The correction is based on so-called Positions Of Interest (POIs) and involves 3 steps: the identification of the POIs in the inertial and vision data, the pairing of the hand POIs to objects POIs that correspond to the same action in the task, and finally, the correction of the hand trajectory based on the pairs of POIs. The complete method for demonstration acquisition is experimentally evaluated in a full PbD process. This experiment reveals the advantages of the proposed method over kinesthesy in the context of this work.

Author's contact details

Robin Pellois

Multibody and Mechatronic Systems Laboratory
Aerospace and Mechanical Engineering Department
Université de Liège (Liège, Belgium)

Allée de la Découverte 13A (B52/3)
B-4000 Liège
Belgium

email: robin.pellois@uliege.be
phone: +32 483 24 05 85

Examination comittee

Prof. Olivier Brüls - *Supervisor*
University of Liège (Liège, Belgium)

Prof. Cédric Schwartz - *President*
University of Liège (Liège, Belgium)

Prof. Pierre Duysinx
University of Liège (Liège, Belgium)

Prof. Pierre Sacré
University of Liège (Liège, Belgium)

Prof. Thibaut Raharijaona
University of Lorraine (Metz, France)

Prof. Renaud Ronsse
Catholic University of Louvain (Louvain-la-Neuve, Belgium)

Financial support

This work has been supported by the Robotix Academy project within the European INTERREG V A Greater Region program.

Acknowledgements

This Ph.D. thesis is the achievement of 6 years of work at the University of Liège within the laboratory of multibody and mechatronic systems.

First of all, I would like to thank warmly Prof. Olivier Brüls who was always available to give me valuable advice. Working under his supervision made this journey enjoyable. Of course, I would like to thank also all the colleagues, current and former, from the laboratory of multibody and mechatronic systems who were always good partners in serious as well as casual times. I would like to address a special thank you to Arthur with whom it was a pleasure to work.

I would also like to thank all the people I had the pleasure to meet and work with during the Robotix Academy events. It was a real pleasure to discuss and share ideas with such interesting people. These moments have energized and enriched this research journey.

A special thought goes to Léa who was always present to support me, and, even though barely understanding the technical aspects of this work, gave me invaluable advice all the way along my path.

Finally, to my parents, my sister, and all my family, to all my friends from Liège, from Normandy and elsewhere, thank you for your unconditional support. Thank you Geoffrey for your interest in my work. I would especially like to thank my father for his artistic contribution to this work.

Contents

List of acronymes and main math symbols

HRI	: **H**uman-**R**obot **I**interaction
PbD	: **P**rogramming **b**y **D**emonstration
IMU	: **I**nertial **M**easurement **U**nit
I2S	: **I**nertial **to** **S**egment
IHMT	: **I**nertial **H**uman **M**otion **T**racking
Z_t and A_t	: States and actions of a task when performed by the teacher
Z_l and A_l	: States and actions of a task when performed by the learner
d_s	: Recorded dataset
d_t	: Dataset encoding Z_t and A_t
d_l	: Dataset encoding Z_l and A_l
${}^0\hat{q}_n$	: Quaternion from frames n to 0
${}^n\boldsymbol{\omega}^{\text{gyro}}(t)$	: Gyroscope measurement with respect to frame n
${}^n\mathbf{a}(t)$	: Accelerometer measurement with respect to frame n
${}^n\boldsymbol{\omega}_n$	: Estimated sensor angular velocity with respect to frame n
${}^n\mathbf{g}$	: Gravity vector with respect to frame n
${}^n\boldsymbol{\omega}_n^{\text{mod}}$	: Estimated sensor angular velocity from gravity vector estimation and gyroscope measurement with respect to frame n
${}^n\boldsymbol{\omega}_n^{\text{reset}}$	: Heading reset correction to the angular velocity with respect to frame n
k_r	: Threshold for the application of the heading reset
k_a	: Threshold for the detection of stationary moments : criterion on the norm of the accelerometer data
k_d	: Threshold for the detection of stationary moments : criterion on the norm of the derivative of the accelerometer data
H	: Human reference frame
A	: Origin of H
$1n$, $2n$, $3n$	: Arm segment frames at time step n
$S1n$, $S2n$, $S3n$	: Sensor frames at time step n
${}^H\mathbf{AD}_n$	: Hand trajectory with respect to frame H
${}^H\hat{q}_{Si0}$ $(i=1,2,3)$	: Initial sensor orientation with respect to the human reference frame
${}^{Si0}\hat{q}_{Sin}$ $(i=1,2,3)$	: Sensor orientation with respect to the initial sensor orientation
${}^{Sin}\hat{q}_{in}$ $(i=1,2,3)$	: Arm segment orientation with respect to the sensor frame
R	: Robot reference frame
O	: Origin of R
${}^R\hat{q}_H$	: Orientation between the human frame and the robot frame
${}^R\mathbf{OP}_n$	: Gripper trajectory measured through robot encoders with respect to frame R
$\mathscr{P}$	: Position of Interest $\mathscr{P}=[s,\mathbf{P},e]$ with s and e the temporal boundaries and $\mathbf{P}$ the 3D position
$\mathfrak{P}$	: pair of POIs
$d(\mathscr{o},\mathscr{h})$	: distance between POIs
$d^t(\mathscr{o},\mathscr{h})$	: temporal distance between POIs

CHAPTER 1

INTRODUCTION

Contents

1.1 Context and objective

Robots are increasingly present in our world in many aspects. In industry, of course, the quantity of robots is increasing. At the hospital, robots help surgeons. On the road, cars are autonomous. In the farm, cows are milked by machines. At home, some machines clean the house or mow the lawn autonomously and children play with robotic toys. This growing presence of robots in our lives comes from the increasing skills of robots. The superiority of robots over human for specific tasks is not in dispute. Robots count faster, work harder and more accurately without getting tired. But one cannot ignore a complementarity between human and robots. For Kyrarini et al. [78] "some tasks that are difficult for a robot may be easy for a human (for example cognitive reasoning, recognition of objects newly appeared in the scene, and manipulation of very small objects). Similarly, some tasks that are difficult or tedious for a human are easy for a robot (for example, performing mathematical computations and performing precise repeated movements with a short cycle time)". In order to take advantage of this complementarity (but not limited to this aspect), research efforts about human-robot interactions (HRI) have been conducted.

Human-robot interactions are studied from different perspectives addressing different problems. Let us give three examples of possible interactions modes. Firstly, facial expressions (from both humans and robots) are studied to improve the communication between robots and humans [110]. Challenges appear when recognizing different human facial expressions or when automatically creating facial expressions on the robot. Secondly, some robots, designated as collaborative, are specially designed to share their workspace with humans. In such context, the problem of safe human-robot interactions appears [139]. Thirdly, human-robot interactions are also present during the programming of robot tasks. An important challenge for that purpose is to build intuitive interfaces between the human and the robot [127].

Among others, interactions between a human and a robot should be considered when a human needs to transfer skills or tasks to a robot. The development for the last decades of new robot programming tools shows the will to improve human-robot interactions at this level. Programming languages become more intuitive with, for instance, programming by blocks or the use of virtual environments. Teach-pendants and other joystick-based interfaces become less relevant for collaborative robots that can be directly manipulated by the human. On this topic, Tsarouchi et al. [123] state that "intuitive robot programming and the promising idea of its simplification is an important aspect".

This thesis is about programming by demonstration (PbD) methods also known as learning from demonstration (LfD) or imitation learning methods, which intend to offer more intuitive robot programming paradigms. The present project took place in the context of the Robotix Academy which addresses applications of robotics for industrial production purposes. Therefore, this thesis focuses on PbD methods for industrial applications. Application fields of PbD will be reviewed in the next section.

1.1.1 Programming by Demonstration

In the Encyclopaedia of Robotics [28], Calinon defined programming by demonstration as "the process used to transfer new skills to a machine by relying on demonstrations from a user".

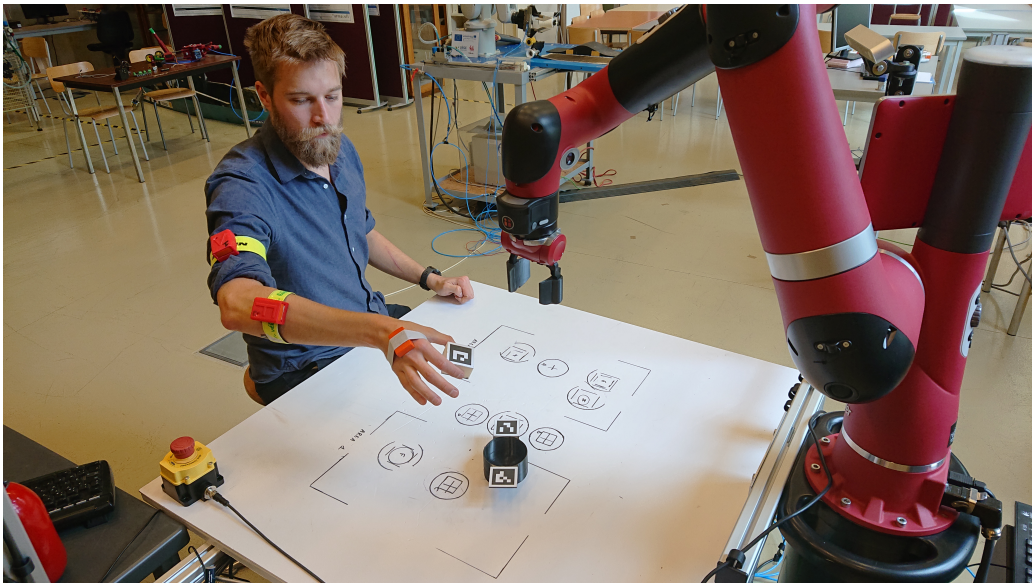

FIGURE 1.1: Illustration of robot programming by demonstration. The human teacher demonstrates a single-armed collaborative robot the task of picking up the cube and dropping it into the jar. The demonstration is captured by inertial sensors on the teacher's arm and a fixed camera tracking objects positions.

In this work, the user is called the teacher or the human and is also the agent who performs the demonstrations. The machine is called the learner, the robot, or the robotic system (i.e., a robot including external sensors). This agent learns and executes the taught task. A demonstration is defined as the teacher-initiated performance of the task – in a specific configuration – to be learned by the learner. The configuration of the task could be, for instance, a specific position of an object to be manipulated. PbD is directly inspired by the way human interact and makes robot programming easier and more accessible. This technology would enhance the application of robots in an industrial context as well as in daily life [41]. According to Vakanski et al. [125], "the intuitive programming style of PbD has the potential to facilitate robotic applications in both the manufacturing industry and the service sector (e.g., office or household environments)". Although PbD currently remains in the academic domain, the diversity of potential applications is quite large. In [43], a robot is taught how to clean different surfaces from demonstrations. In [73], a method to teach task requiring positions and forces is applied to an ironing and a door opening tasks. The task of rolling out a pizza dough is taught to a robotic system in [48]. In [1], helicopter aerobatics maneuvers are taught to a robotic system that only a few human experts are capable of performing. In [141], virtual reality is used to transmit to a robot a task consisting in the complex manipulation of rigid and soft bodies (like textiles).

Several reviews have been proposed in the literature to formalize this paradigm [8, 20, 34, 109]. From these reviews, a PbD process is decomposed into two main steps, as illustrated in Figure 1.2. The first step is to *gather the dataset* from the demonstrations. The second step consists in generalizing or "understanding" the task from the dataset in order to enable the proper execution of the task by the robot in an unknown configuration, for instance, with object positions never encountered in the demonstrations. This second step is called *learning the task* in this work.

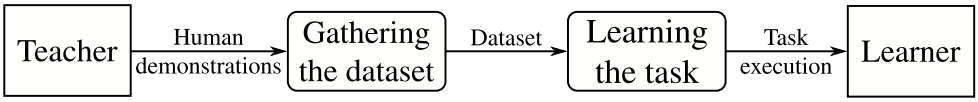

FIGURE 1.2: Programming by demonstration process

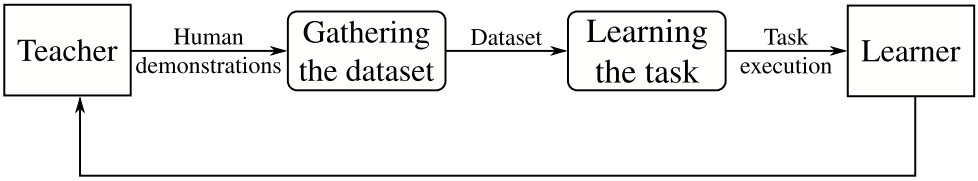

FIGURE 1.3: Programming by demonstration *refining a task* process

It is sometimes possible to repeat these two steps in an iterative manner as illustrated in Figure 1.3. These approaches are denoted as *refining a learned task* in [34] or as *active and interactive demonstrations* in [109]. Following this approach, the learner interacts with the teacher during the process of learning the task and can ask for more information or demonstrations. Chernova et al. [34] define it as "the learner, in turn, helps the instructor by making their learning process transparent through communicative acts, and by demonstrating their current knowledge and mastery of the task".

This work focus on the first step of a PbD process: *gathering the dataset*. A review of existing methods regarding this particular problem is given in the following section. However, the two steps influence each other and more information about the second part will be given in the last chapter 8 of this work in which a complete PbD process will be implemented.

1.1.2 Gathering the dataset

In order to investigate this step of a PbD process we propose, following Argal et al. [8], to model the task in terms of states Z and actions A. Let us illustrate these concepts for the task of picking up a pen from the table to place it into a jar. In this example, "picking up" or "place" are the actions and "pen on the table" or "position of the jar" are the states. In this way, a task is considered as one or more actions associated with a desired final state, for instance, the task can be reduced to "place the pen into the jar". The execution of the task is a succession of states and actions such that the actions modify the current state into a new state until the desired task state is reached. The task can be executed in multiple ways. One way is called a behaviour and noted π. For instance, "picking up the pen on the table with the hand and dropping it into the jar" is one behaviour, "grabbing the jar and slide the pen in it with the hand" is another one. Following this model, the *learning the task* part is to estimate, through demonstrations, a learner's behaviour π_l to execute the task. Such execution involves a succession of states and actions Z_l and A_l specific to the learner.

A demonstration is performed under the teacher initiative with a certain behaviour π_t involving the sets of states and actions A_t and Z_t. These states and actions Z_t and A_t are usually not accessible and only a partial representation of them is recorded in a dataset d_t. According to

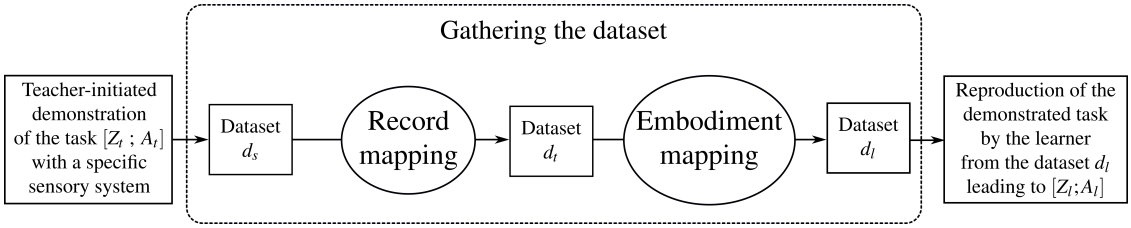

FIGURE 1.4: Illustration of the step of gathering the dataset.

Argal et al. [8], "for [PbD] to be successful, the states and actions in the learning dataset must be usable by the [learner]". In other words, the learner should be able to reproduce the demonstration performed by the teacher (in the same task configuration) thanks to this learning dataset. However, the teacher and the learner usually differ in sensing and actuation. In such cases, the sets of states and actions Z_t and A_t of the execution π_t are irrelevant for the learner. For instance, the behaviour "grabbing the jar and slide the pen in it with the hand" which requires both hands of the teacher cannot be achieved by a single robotic arm. In other words, the dataset d_t that encodes the states and actions Z_t and A_t is useless for the learner to reproduce the demonstrated task. This problem is referred to as the correspondence issue. To solve the correspondence issue, an embodiment mapping is necessary. The embodiment mapping transforms the dataset d_t into a new dataset d_l. This dataset d_l encodes the states and actions Z_l and A_l that correspond to the demonstation (π_t, Z_t and A_t) in a relevant way for the learner. For instance, if the human motion is described with joint angles and if the human and the robot have different kinematics, an embodiment mapping is necessary to adapt the recorded set of joint angles to the robot's kinematics.

Furthermore, the demonstration is recorded with a specific sensing system. From this sensing system, the recorded dataset d_s is created and does not necessarily correctly encode the states Z_t and actions A_t. In this case, from the recorded dataset d_s, the record mapping creates a new dataset d_t that better encodes the behaviour π_t and the states Z_t and actions A_t of the demonstration. For instance, while the teacher picks up the pen and drops it into the jar, he/she is filmed. The set of recorded images is d_s. It is used to extract the human motion as well as the object positions. The extraction method is the recorded mapping and the human motion and the object positions are the dataset d_t. The human motion described in d_t can be used to describe the actions A_t. For the same behaviour, the teacher's motion can be recorded through inertial sensors. This way, the dataset d_s directly represents the human motion and encodes the actions A_t. A record mapping may not be necessary but can still be useful to better encode states and actions, e.g. to obtain the hand trajectory instead of the sensor acceleration and angular velocity which can facilitate the identification of the embodiment mapping. Figure 1.4 illustrates the process of gathering the dataset and the two mappings.

The correspondence issue can be more or less difficult to solve depending on the demonstration acquisition method. As explained in [7] "correspondence issues are minimized when the learner records directly from its own sensors while under the control of the teacher". In other words, some methods lower the difficulty of solving the correspondence issue by making the learner perform the demonstration under the teacher's initiative. One largely used approach is kinesthesy. Acquiring a demonstration by kinesthesy consists for the teacher to directly manipulate the robot to make it perform the task. During the demonstration the robot motion is recorded from its own encoders. In this way, the recorded data directly encodes the relevant states Z_l and

actions A_l for the learner. In other words: $d_s = d_l$. On the other hand, some methods allow the teacher to perform the demonstration more freely with her/his own body while being "observed" by a sensory system. The correspondence issue needs to be solved ($d_t \neq d_l$) and the recorded dataset d_s may need to be adapted. Thus, some record and/or embodiment mappings, depending on the demonstration acquisition method, need to be identified.

The above described dichotomy is present in several taxonomies of the demonstration acquisition methods. Chernova et al. [34] classify the demonstration acquisition methods into *learning by doing* for those that tend to circumvent or limit the correspondence issue and *learning from observation* for those allowing the teacher to move freely. Similarly, Calinon [28] proposes the two following categories: *observational learning* and *kinesthetic teaching*. In a recent survey [109], three categories are proposed. The *kinesthetic teaching* category is still present as well as the *observational learning* under the label *passive observation*. The third category is *teleoperation* which gathers the demonstration acquisition methods using a remote controller of the robot. Teleoperation-based demonstrations can be applied to a wider range of applications and robotics systems (underwater robots, virtual reality ...) than *kinesthetic teaching*.

Ravichandar et al. [109] confirm that *kinesthetic teaching* – and *teleoperation* – circumvent the correspondence issue but they also present some limitations: "the quality of the demonstrations depends on the dexterity and smoothness of the human user, and even with experts, data obtained through this method often require smoothing or other post-processing techniques". Methods from *kinesthetic teaching* and *teleoperation* are easy to implement but prevent natural human motion which reduces the range of applications. For instance, a painting task [125], requiring dexterity, would be difficult to achieve in that way. On the other hand, demonstrations from "observation" may present difficulties in implementation and mappings identification. In [30], a human operator demonstrates basket referee gestures with his own body wearing inertial sensors to a humanoid robot. Due to correspondence issues, the demonstration motion cannot be used directly and is refined by kinesthesy. Demonstration acquisition methods usually require a compromise between the difficulty of identifying mappings and convenience for the teacher to perform the demonstrations. Demonstration acquisition methods should also meet the goal of PbD of facilitating robot programming and must be easy to implement. The development of an observational demonstration acquisition method is the objective of this thesis as will be detailed later. The proposed method should overcome the limitations of kinesthesy and should also limit the difficulty of identifying the required correspondence mappings.

It has been seen that the process of *gathering the dataset* starts with sensor measurements. The selected sensors influence the complete process since some mappings may or may not be necessary. Therefore, the technology to measure human motion has to be selected while taking into account the constraints of the PbD process.

1.1.3 Human motion tracking technology

It has been stated that the process of *gathering the dataset* should limit as little as possible human motion for the teacher convenience, for intuitive interactions and, for a larger application range. The technology used to measure the human motion should be compatible with these objectives. Field et al. [47] expose a survey of human motion capture methods in robotics and present 4 main categories.

- Cameras and optoelectronic systems have the advantage of giving the position in a well-known reference frame. Such devices usually present a high accuracy. But they face concealment problems due to the environment (light, dust, smokes) or to the movements of the operator. In [46], such a system is used to track a special luminous tool for an industrial painting application. In [63], the human forearm motion is tracked with fiducial markers and the concealment issue is managed using additional inertial sensors.

- Mechanical devices such as exoskeletons measure the relative motion of human joints directly which can be easily mapped to a human body model but do not directly give the absolute trajectory with respect to a fixed reference frame. Exoskeletons and other systems may also be heavy, intrusive and uncomfortable for the operator and could also limit the operator motions by the mechanical constraints of the structure. An example is given in [37].

- Devices based on the measurement of an artificially generated magnetic field are mentioned but are not appropriate in an industrial environment due to their high sensitivity to magnetic disturbances.

- Inertial Measurement Units (IMUs) are small, light and cheap (for the MEMs version) making them suitable for wearable devices with possible integration in clothes [133], and do not require the use of external apparatus such as cameras, which facilitates the implementation. However, IMU data require complex treatment and signal filtering for the trajectory reconstruction process and the accuracy is affected by drift phenomenon.

Each technology has its own advantages and drawbacks making it more or less suitable depending on the context of the application. Regarding the practicality of the system, optoelectronic and mechanical devices are more cumbersome than IMUs. IMUs can thus better meet the expected intuitiveness and easiness of use in PbD. Furthermore, the applicability of inertial sensors as human motion sensors has already been demonstrated for robotic applications. Filippeschi et al. [49] declared that inertial systems are “a candidate alternative to optical ones for online human motion capture”. Miller et al. [96] claim that “the functionality of [their inertial motion suit] was demonstrated through experiments in teleoperating the NASA Robonaut”. Recently, Wilson et al. [132] address the validation of IMU sensors for HRI in a context of robotic manipulator teleoperation. As inertial sensors present features that fulfil the requirement of a PbD process, this technology is chosen in the context of this thesis.

1.1.4 Objective of the thesis

In the context of industrial robotics, this work proposes an observational method for acquiring human demonstrations in a PbD process based on inertial sensors. The method should be easy to implement and to use for the teacher and should allow him/her to perform natural motions. More specifically, the study focuses on pick-and-place tasks, which are frequently encountered in industrial applications. In the demonstration of a pick-and-place task, the main variable of interest to be recorded is the human hand trajectory and state which should then be replicated by the end-effector of the robot. Object positions should also be recorded during the demonstrations.

1.2 State-of-the-art and contributions of the thesis

To meet the objective of this work, a method for tracking human motion based on inertial sensors is proposed. Several elements are involved in this method and this work proposes contributions at different levels. The following paragraphs present the state-of-the-art on the relevant elements and introduce the contributions of the thesis.

1.2.1 Inertial human motion tracking

Inertial human motion tracking (IHMT) has been investigated for a wide range of applications including robotics [49, 130]. In this work, we focus on the tracking of the human arm. Several problems which have been addressed by the robotics community are discussed here.

First, a kinematic model of the human arm is needed to reconstruct the motion. The human arm behaves as a complex kinematic chain. For instance, the shoulder center of rotation depends on the position of the arm [3]. Several human arm models have been proposed in the literature. For instance, Klopcar et al. [69] propose to model the shoulder as a complex joint composed of a universal and a spherical joint for the shoulder and to model the elbow as a revolute joint. In [82], the elbow and the wrist are modeled in a way to mimic the human skeleton with high fidelity. Such models can capture the complexity of the human arm but require detailed information such as inner shoulder length, which can be difficult to access. In a simpler way, the human arm can be modeled as a 7-degrees-of-freedom (DoF) system. In [64] and [106], the model is composed of a spherical joint for the shoulder and the wrist and a 1-DoF joint for the elbow. In [79], the elbow is modeled as a 2-DoF joint. Another model consists of 3 spherical joints for the shoulder, the elbow and the wrist [59, 140]. These papers show that such model can be useful even if a human arm is close to a 7-DoF kinematic chain. In our context, the simplicity requirement drives us to use this last model so that the complexity of identifying the record and embodiment mappings is also reduced. The model is explained in Chapter 4.

Secondly, a method is needed to evaluate the hand position from the segment attached sensors. As explained in [95], two approaches have been identified to estimate it: the kinematic chain and the free segment approach. The kinematic chain approach consists in estimating joint angles from measurements of sensors placed before and after the joint in the kinematic chain [33, 55]. The free segment approach consists in estimating the orientation of each segment with respect to an inertial reference frame using the sensor attached to the segment [42, 86, 137]. The kinematic chain technique is well suited if the joint angle is the variable of interest like in neuromuscular disorders assessment [87] or if the motion should be mapped to a human-like model [71, 96]. In this work, it is only necessary to evaluate the hand trajectory and both approach can be considered. The free segment approach has been chosen for that purpose.

Third, one challenge in IHMT is to identify the orientation of the sensors with respect to the human segments as illustrated in Figure 1.5. It is referred in literature as the inertial-to-segment (I2S) calibration problem. Some methods assume that the sensors are aligned with the segment; in this case, these parameters do not need to be evaluated [136, 143]. The implementation of such method only requires a precise alignment procedure of the sensor with respect to the segment frame. Other calibration methods rely on measurements during pre-defined static positions. The orientation of the human segments is assumed to be known in such positions. A

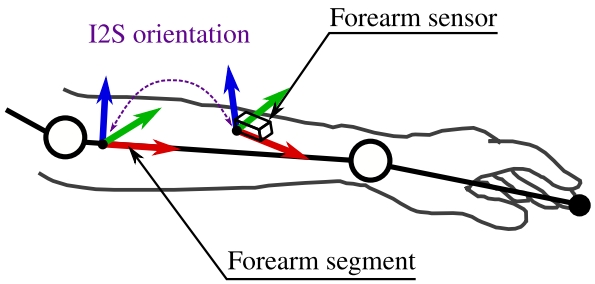

FIGURE 1.5: Illustration of the orientation between the inertial sensor and the human arm segment frame: the inertial-to-segment (I2S) orientation.

commonly used pose is the N-pose (neutral pose) as in [22, 106, 142] where the human stands up with his/her arms straight along his/her body. Another method relies on the T-pose where the operator stands with arms straight horizontal in the coronal plane [111]. Other methods rely on different static positions such as in [104, 120, 138]. Another calibration approach exploits measurements in complex functional motions. In [128], joint axes of rotation in the human arm are estimated thanks to 13 standardized motions in order to estimate the I2S orientation. This method seems rather complex for applications in a PbD process. A similar approach is proposed in [87] for ambulatory measurements. A hybrid solution with one functional motion and a static pose is proposed in this work to fulfil PbD requirements. In summary, various IHMT methods have been proposed in literature. A method which provides anatomically interpretable data in an ambulatory application will not be relevant in another context such as virtual characters animation. The choice of the IHMT method depends on the application, the data of interest and the constraints.

Contribution

In this work, an IHMT method is developed to fulfill the requirements of a PbD process. Within this method, a specific I2S calibration method is proposed, which is based on a static position and a functional motion. This procedure, which is explained in Chapter 4, is a part of the record mapping.

The estimation of the segment orientation is obtained from the estimation of the IMU orientation using the IMU signals. The next part presents the state-of-the-art regarding this problem.

1.2.2 Sensor orientation estimation

An IMU sensor is composed of a gyroscope measuring the angular velocity and an accelerometer measuring the acceleration of the sensor. Many methods have been proposed to estimate the orientation of a rigid body using inertial sensors. Different methods are compared in a context of pedestrian navigation in [94], some non-linear methods are reviewed in [91] and methods related to human motion tracking are exposed in [49, 72, 85, 130]. An important problem when working with IMU sensors is to deal with the drift. The drift is the continuously increasing error in the orientation due to the integration of the gyroscope data that measures the angular velocity but contains a bias and a noise. The gravity direction estimated from accelerometer data partially compensate the drift. The problem is to manage the remaining drift.

In order to manage the remaining drift, inertial sensors are often coupled with a magnetometer which measures the local magnetic field. Such coupled sensors can be called MIMU (or mIMU or IMMU) (Magnetic and Inertial Measurement Unit), MARG (Magnetic, Angular Rate, Gravity) or AHRS (Attitude and Heading Reference System). Magnetic sensors measure the Earth magnetic north and give a partial information on the orientation of the sensor. But the amplitude of the Earth magnetic field being low, its measurement is sensitive to disturbances. Managing these magnetic disturbances is a main challenge when working with such devices. According to Ligorio et al. [83], there are two approaches to deal with magnetic disturbances: the threshold-based approach and the model-based approach. De Vries et al. in [129] study the disturbances of the Earth magnetic field in a laboratory. The impact of the disturbances in the estimation of the orientation can be high. The authors recommend evaluating the magnetic distortion in the volume of the laboratory for a better estimation. This solution does not fulfill the easiness-of-use targeted in the present work. In a PbD context, the presence of the robot and its electrical actuators in the neighborhood of the IMUs represents an additional source of disturbances. It is expected that an orientation estimation partly based on the magnetometer data would not be reliable for that reason. At the beginning of this project, an experiment has been conducted to verify the effect of the magnetometer on the orientation estimation error in a robotics context (see Appendix B). Three algorithms which can be used with MARG or IMU devices have been implemented (see Appendix A). Sensors were mounted on a robot arm end-effector. The sensor orientation estimated from the different algorithms with and without using magnetometers was compared to the orientation of the end-effector measured by robot encoders. This experiment has shown that magnetometer data are not reliable in a robotic environment and can even deteriorate the estimation of the sensor orientation. Therefore, it has been decided not to use magnetometers in our context.

In order to manage the drift without any magnetometer, IMUs can be coupled with other sensors, for instance, with an ultra-wideband device [145]. However, adding extra sensors may decrease the easiness of implementation targeted in this work. Therefore, the challenge addressed in this work is to estimate the orientation of the sensor only from IMU data, i.e., from gyroscope and accelerometer measurement.

IMU orientation estimation algorithms have been proposed in literature and a number of review papers can be listed [49, 72, 85, 130]. Four main methods can be identified:

- Complementary filters [10, 44, 51, 89]. The complementary filter is based on 2 complementary transfer functions: one to filter the gyroscope data and the other to filter the accelerometer data.
- Kalman filters [35, 91]. The Kalman filter, the Extended Kalman Filter [67, 92] and the Unscented Kalman filter [75] can be used for data fusion of gyroscope and accelerometer data. The design of the Kalman filter is based on a model of the measurement process which accounts for the noise and the bias of the gyroscope and the accelerometer see, e.g., [137].
- Gradient descent filters [88]. This method is based on the optimization of the orientation estimation at each time step.
- Integration and vector observation [26, 60]. This approach consists in a strapdown integration of the gyroscope data. The orientation resulting from this process is corrected with

gravity estimation from accelerometer data. A key point in this method is to manage the disturbances of the linear acceleration on the gravity estimation.

- Hybrid methods can be mentioned such as in [86] where the integration and vector observation method is completed with an estimation of the gyroscope bias using a Kalman filter.

- Other approaches can be mentioned such as in [81] where the human motion is estimated using accelerometer data only. This method is based on gravity measurement and is applicable only when the sensor is not accelerated. To overcome this limitation, two sensors are used on each segment and the distance and orientation between the two sensors has to be precisely known. Similarly, two accelerometers at the opposite corners of a cube are used in [122]. These two sensors cannot be too close to each other. These approaches do not present a high precision and imply many constraints. They are not considered in this work.

According to the surveys [49, 72, 85, 130], none of the methods clearly outperforms the others. The integration and vector observation approach has been selected for its easiness of use and another approach, specific to our PbD problem, is proposed to overcome the drift issue.

Contribution

In the context of human motion tracking, we propose a novel approach to catch up the drift from gyroscope data with only data from an IMU sensor: the heading reset. The heading reset is applied on an IMU orientation estimation algorithm based on angular rate integration and gravity observation. The heading reset relies on the hypothesis that the teacher, wearing the sensors, is expected to do motions that bring the sensors in their original heading with stationary phases during the demonstration. At this moment, a reset of the heading of the sensor is applied. This method is explained in Chapter 2.

1.2.3 Trajectory Correction

In the context of PbD, the demonstration is based on human motion measurements which should be performed with a certain level of accuracy. The required accuracy depends on the tasks to be taught. Considering a pick-and-place task, the required level of accuracy can be related to the size of the manipulated object, the gripper and the objective of the task. It can be mentioned that in such a context, the same level of accuracy is not required during the whole demonstration. The picking and placing positions require a higher accuracy than the rest of the trajectory. On the other hand, actual industrial robots usually have accuracy about 0.1 mm. However, it does not seem possible to reach such level of accuracy with IHMT. In their survey, Filippeschi et al. [49] evaluate the accuracy of 5 different IHMT methods between 23.6 mm and 89.2 mm. These values seem insufficient for a PbD process. This point is further discussed in Chapter 5 based on experimental data. The problem addressed here is to reach a sufficient level of accuracy of the position of the human hand for PbD while keeping the easiness-of-use offered by inertial sensors. Some approaches are discussed in the following.

The elbow kinematic constraint is explored in some approaches. For instance, in [80], the elbow joint axis with respect to the forearm and arm sensors is estimated. During the trajectory, the sensors are considered motionless with respect to their respective segment frames. Therefore, the elbow joint axis with respect to the sensor frames is invariant. The elbow joint axis with

respect to the inertial reference frame can be estimated from both forearm and arm sensor orientation estimations. The two estimations of the elbow axis should be equal, so the difference is used to correct the two sensors orientation. This method can indeed improve the accuracy of the hand trajectory. However, the consequences for the implementation can be high, for instance by imposing a procedure to identify elbow axis with respect to the sensor frame. Furthermore, even if the accuracy is improved, there is no guarantee that the complete method reaches a sufficient level of accuracy.

Other methods exploit the double integration of the linear acceleration estimated from the accelerometer data to estimate the displacement of the sensor. With a sensor attached to the hand, the trajectory of the hand can be estimated. It seems, however, that this approach is only suitable for gait analysis [51, 58, 101]. Indeed, human gait naturally alternates short motions with motionless moments. Therefore, the acceleration is only integrated on a short period of time and motionless moments are exploited to reset errors from double integration. Such approaches do not seem suitable in an upper limb motion tracking context, and will thus not be considered in our context.

An important trend in the literature is to merge vision-based measurements with inertial-based measurements. This approach is motivated by the complementarity between these technologies. According to [47], vision-based systems have the accuracy required for human demonstration acquisition. Vision-based systems can give accurate and absolute position estimation but are light-sensitive and suffer from concealment. On the other hand, inertial sensors can transmit a continuous flow of data, they are light, cheap, wearable and easy-to-use, but suffer from inaccuracy. The complementarity of vision and inertial technologies has already been investigated for human motion measurement. In [107], an accurate system to track the tool trajectory for industrial painting is propounded. In a similar approach, the human forearm is robustly and accurately tracked in [5] using several cameras to track a set of markers on the human arm. Several cameras are needed to avoid occlusion problems. In [63], a single kinect camera is used to track human arm motions but a hand position error up to 10 cm occurs during occlusion phases. In [70], a similar method is used to teleoperate a robot. Vision and inertial systems are merged with ultra-wideband device for an accurate indoor positioning in [100]. The previously cited works reach a suitable level of accuracy for a PbD process, however, they require the use of fiducial markers and special cameras or extra sensors to overcome occlusion issues. Such setups are complex and do not fulfill the requirement of simplicity targeted in this work.

In the previous section 1.1.3, vision-based systems were not selected for human motion measurement due to their limitations. However, these systems may still be considered for environment observations. Measuring the environment does not imply the same constraint as measuring human motion. Furthermore, observing elements in the environment is needed during the demonstration as mentioned in the section 1.1.4. Merging vision-based and inertial-based measurement of different elements appears as an attractive strategy for the targeted application of this work. In [99, 114], the position of a mobile robot is estimated accurately using an embedded camera. In these works, the environment observations enable the robot to estimate its position which is then used to correct the inertial-based estimation of the robot position. In [97], inertial sensors set on the human arm measure a motion that is then used as command to teleoperate a robot. Vision-based tracking of objects and robot gripper is used in a feedback loop to better control the teleoperation of the robot. This method presents good features for our purpose but would suffer

from similar limitations as kinesthetic demonstrations exposed before, i.e., it may hinder natural human motion. The proposed solutions do not seem adapted to the objective of this work which motivates the development of a specific method to solve to this problem.

Contribution

A novel method is presented in this work to correct the IHMT hand trajectory using vision-based environment observations to acquire accurate human demonstrations in a PbD process. This approach is developed in the context of pick-and-place tasks. In such a context, the environment observations consist in tracking the manipulated objects. As shown in [2] and [57], numerous tools for object detection in robotics are available and do not require heavy equipment, which fulfils the simplicity of use and the robustness required by a PbD process. The correction method is presented in chapter 6. It is based on the identification and matching of position of interest (POIs).

1.3 Manuscript Structure

The objective of the thesis is to propose a demonstration acquisition method for programming by demonstration (PbD) for pick-and-place tasks. The method relies on the human hand trajectory estimation from inertial sensors. A method to limit the drift of inertial sensor orientation estimation in this context is proposed. In addition, an inertial human motion tracking (IHMT) method adapted to the context of PbD is developed. Finally, based on vision, a method to correct the IHMT hand trajectory is proposed to reach a sufficient level of accuracy.

This manuscript is organized as follows:

- Chapter 2 : The method used for IMU orientation estimation is presented as well as the heading reset method to manage the drift.
- Chapter 3 : The experimental evaluations of the heading reset method and the assessment of the complete IMU orientation estimation method are described.
- Chapter 4 : The complete IHMT method developed for this context is presented. The I2S method and other elements required to adapt the dataset to be usable by the robot are presented.
- Chapter 5 : Several experiments are conducted to validate the different elements of the IHMT method. The experiments enable to identify the different factors influencing the accuracy of the method.
- Chapter 6 : The vision-based method to correct the IHMT measurements is presented.
- Chapter 7 : The experimental validation of the corrected IHMT method described in Chapter 6 is presented.
- Chapter 8 : Finally, the complete demonstration method is tested in a real PbD process.
- Chapter 9 : This last chapter concludes this work.

1.4 Author publications

The contributions of the thesis have been published in two journal papers and two papers in conference proceedings. The first paper in the list was also presented orally at the International Conference on Automation, Robotics and Applications, in 2022.

1. Robin Pellois and Olivier Brüls, "A vision-based correction of inertial measurement of human motion for robot programming by demonstration", *International Journal of Mechanical Engineering and Robotics Research*, Vol. 11, No. 6, pp. 411-416, DOI: 10.18178/ijmer r.11.6.411-416, 2022

2. Robin Pellois and Olivier Brüls, "An inertial human upper limb motion tracking method for robot programming by demonstration", *Robotics and Autonomous Systems*, Vol. 156, 104201, DOI: 10.1016/j.robot.2022.104201, 2022

3. Robin Pellois and Olivier Brüls,"Human arm motion tracking using IMU measurements in a robotic environment", In: *Proceedings of the 21st International Symposium on Measurement and Control in Robotics (ISMCR 2018)*, Mons (Belgium), pp. 65-68, 2018

4. Robin Pellois, Laura Joris, Olivier Brüls,"Robot control based on human motion analysis with IMU measurements", In: *Proceedings of the Robotix Academy Conference for Industrial Robotics (RACIR)*, Luxembourg (Luxembourg), 2017

CHAPTER 2

IMU ORIENTATION ESTIMATION METHOD

Contents

In this chapter, the method to estimate the orientation of the IMU sensor is presented. The method is based on angular rate integration and gravity measurement. For this algorithm, identifying moments when the sensor is stationary is a key point for relevant gravity measurement. Our contribution on this point is presented. Furthermore, an original procedure called heading reset is presented. The heading reset is introduced to limit the drift around the gravity direction.

2.1 Formalism

2.1.1 Frames

A sensor frame is used to describe the sensor motion.

- Frame 0 denotes the sensor frame at time step 0. It defines the inertial reference frame for the orientation of the sensor. The sensor is considered motionless at this time step.
- Frames n and $n-1$ are the sensor frames at time step n and $n-1$ respectively.

It can be mentioned that the Earth frame defined by the gravity direction and the magnetic north is not considered here since no magnetometer is used.

2.1.2 Quaternion formulation

The unit quaternion ${}^0\hat{q}_n$ denotes the rotation from frame n to frame 0 and ${}^n\hat{q}_0 = ({}^0\hat{q}_n)^{-1}$. A three-dimensional vector $\mathbf{u}$ can be expressed as a quaternion $\hat{u}$ with no real part but only a vector part $\mathbf{u}$:

$$\hat{u} = \begin{bmatrix} 0 \\ \mathbf{u} \end{bmatrix} = \begin{pmatrix} 0 \\ u_x \\ u_y \\ u_z \end{pmatrix}. \tag{2.1}$$

2.1.3 Sensor data

An IMU sensor measures the angular velocity ${}^n\boldsymbol{\omega}^{\text{gyro}}(t)$ and the acceleration ${}^n\mathbf{a}(t)$ of the sensor in frame n. In this work, the model of the gyroscope data is

$${}^n\boldsymbol{\omega}^{\text{gyro}}(t) = {}^n\boldsymbol{\omega}^{\text{true}}(t) + {}^n\boldsymbol{\beta}(t) + {}^n\boldsymbol{\gamma}(t) \tag{2.2}$$

with ${}^n\boldsymbol{\omega}^{\text{true}}(t)$ the true angular velocity of the sensor, ${}^n\boldsymbol{\beta}(t)$ a bias and ${}^n\boldsymbol{\gamma}(t)$ a noise. The model of the accelerometer data consists of a gravity component ${}^n\mathbf{g}(t)$, a linear acceleration component ${}^n\mathbf{l}(t)$ and a noise ${}^n\mathbf{n}(t)$:

$${}^n\mathbf{a}(t) = {}^n\mathbf{l}(t) + {}^n\mathbf{g}(t) + {}^n\mathbf{n}(t). \tag{2.3}$$

IMU data needs to be filtered to manage noise and calibrated to determine bias, misalignment and gain/sensitivity [17, 121]. However, this question is not the purpose of this work and it is assumed that the sensor data have been pre-filtered and calibrated.

2.2 Angular velocity integration

Let us represent the quaternion product by $\otimes$. The kinematic equation of a rigid body describes the variation of its orientation as:

$$\dot{\hat{q}} = \frac{1}{2}\hat{q} \otimes {}^{n}\hat{\omega}(t) \tag{2.4}$$

with ${}^{n}\hat{\omega}(t)$ the quaternion representation of the angular velocity at time t in frame n. The above equation is solved using the explicit Euler method on the Lie group of unit quaternions [6]

$${}^{0}\hat{q}_n = {}^{0}\hat{q}_{n-1} \otimes \exp(\frac{1}{2}\Delta t \, {}^{n}\boldsymbol{\omega}_n) \tag{2.5}$$

with Δt the time step value and ${}^{n}\boldsymbol{\omega}_n$ the estimated angular velocity at time step n in the frame n and the exponential map

$$\exp(\mathbf{u}) = \begin{pmatrix} \cos(\|\mathbf{u}\|) \\ \sin(\|\mathbf{u}\|)\frac{u_x}{\|\mathbf{u}\|} \\ \sin(\|\mathbf{u}\|)\frac{u_y}{\|\mathbf{u}\|} \\ \sin(\|\mathbf{u}\|)\frac{u_z}{\|\mathbf{u}\|} \end{pmatrix}. \tag{2.6}$$

$\|.\|$ denotes the Euclidean norm. The inverse of the exponential map is the logarithm map $\log(\hat{q})$ which satisfies

$$\log(\hat{q}) = \frac{2\,\mathrm{acos}(\hat{q}_1)}{\sqrt{\hat{q}_2^2 + \hat{q}_3^2 + \hat{q}_4^2}} \begin{pmatrix} \hat{q}_2 \\ \hat{q}_3 \\ \hat{q}_4 \end{pmatrix}. \tag{2.7}$$

The integration of the gyroscope measurement ${}^{n}\boldsymbol{\omega}_n^{\mathrm{gyro}}$ using Eq. (2.5) introduces a drift in the orientation due to the gyroscope bias ${}^{n}\boldsymbol{\beta}$ and noise ${}^{n}\boldsymbol{\gamma}$. To limit this drift, a modified angular velocity ${}^{n}\boldsymbol{\omega}_n^{\mathrm{mod}}$ is introduced in the next section.

2.3 Vector observation in stationary moments

The gravity vector estimation from accelerometer measurement enables to partly compensate the drift in phases when the sensor is almost stationary. In such stationary phases, the accelerometer measurement ${}^{n}\mathbf{a}$ is an estimate of the orientation of ${}^{n}\mathbf{g}$, which can be exploited to compensate some drift components. Let us introduce the geometric vector $\mathbf{d}$, a unit vector which follows the motion of the sensor during the interval $[t_{n-1}, t_n]$ (see Figure 2.1). This vector is defined as aligned with the gravity vector at time step $n-1$, i.e., in the frame $n-1$, we have ${}^{n-1}\mathbf{d}_{n-1} = {}^{n-1}\mathbf{g}$. Also the same vector $\mathbf{d}$ at time step $n-1$ but projected in the frame n satisfies ${}^{n}\mathbf{d}_{n-1} = {}^{n}\mathbf{g}$. The vector $\mathbf{d}$ being constant with respect to the sensor frame, the projection of $\mathbf{d}$ at time step $n-1$ in the sensor frame $n-1$ is equal to the projection of $\mathbf{d}$ at time step n in sensor frame n,

$${}^{n}\mathbf{d}_n = {}^{n-1}\mathbf{d}_{n-1} = {}^{n-1}\mathbf{g}. \tag{2.8}$$

The rotation ${}^{n-1}\hat{q}_n$ can be decomposed as

$${}^{n-1}\hat{q}_n = {}^{n-1}\hat{q}_* \otimes {}^{*}\hat{q}_n. \tag{2.9}$$

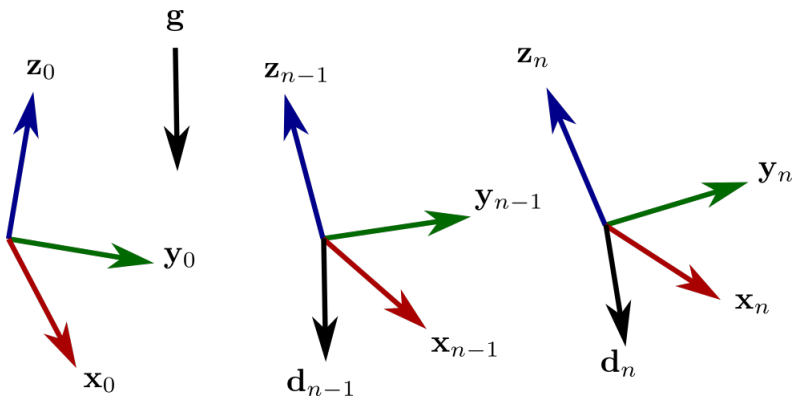

FIGURE 2.1: Definition of the vector $\mathbf{d}$ which follows the sensor motion during the interval $[t_{n-1}, t_n]$ and which satisfies ${}^{n-1}\mathbf{d}_{n-1} = {}^{n-1}\mathbf{g}$.

The rotation ${}^{*}\hat{q}_n$ is defined as the rotation around a horizontal axis perpendicular to the plane containing the vectors ${}^{n}\mathbf{d}_n$ and ${}^{n}\mathbf{d}_{n-1}$ which rotates the vector ${}^{n}\mathbf{d}_{n-1}$ to ${}^{n}\mathbf{d}_n$. The rotation angle of ${}^{*}\hat{q}_n$ being small when the sensor is almost stationary, its rotation vector is estimated as ${}^{n}\mathbf{d}_{n-1} \times {}^{n}\mathbf{d}_n$, which is equal to ${}^{n}\mathbf{g} \times {}^{n-1}\mathbf{g}$ with $\times$ the cross product. The rotation ${}^{n-1}\hat{q}_*$ is a heading rotation about an axis aligned with ${}^{n}\mathbf{d}_{n-1} = {}^{n}\mathbf{g}$ and it can be estimated by the integration of the projection of the gyroscope measurement on ${}^{n}\mathbf{g}$. Thereby, during stationary phases, a modified angular velocity ${}^{n}\boldsymbol{\omega}_n^{\text{mod}}$ is defined as

$$ {}^{n}\boldsymbol{\omega}_n^{\text{mod}} = \frac{1}{\Delta t}({}^{n}\mathbf{g} \times {}^{n-1}\mathbf{g}) + (I_3 + {}^{n}\tilde{\mathbf{g}}\,{}^{n}\tilde{\mathbf{g}})\,{}^{n}\boldsymbol{\omega}_n^{\text{gyro}} \tag{2.10} $$

with I_3 is the identity matrix of dimension 3 and $\tilde{\mathbf{u}}$ represents the skew-symmetric matrix defined from the vector $\mathbf{u}$ as

$$ \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix}. \tag{2.11} $$

Figure 2.2 illustrates the different components of ${}^{n}\boldsymbol{\omega}_n^{\text{mod}}$. The first term in Eq. (2.10) represents a rotation about a horizontal axis and the second term represents a heading rotation about the vertical axis.

A key point of the proposed algorithm is to evaluate ${}^{n-1}\mathbf{g}$ in Eq. (2.10) using the estimated rotation at time $n-1$ according to

$$ {}^{n-1}\hat{g} = {}^{n-1}\hat{q}_0 \otimes {}^{0}\hat{g} \otimes {}^{0}\hat{q}_{n-1}, \tag{2.12} $$

where ${}^{0}\mathbf{g}$ is the gravity measured in the initial configuration of the sensor. Indeed, the sensor being motionless at time step 0, the linear acceleration is negligible ${}^{0}\mathbf{l} \approx \mathbf{0}$ and ${}^{0}\mathbf{g} \approx \frac{{}^{0}\mathbf{a}_0}{\|{}^{0}\mathbf{a}_0\|}$. This evaluation of ${}^{n-1}\mathbf{g}$ is thus affected by the global drift of the estimated rotation. In contrast, ${}^{n}\mathbf{g}$ is evaluated by direct estimation from the accelerometer signal as ${}^{n}\mathbf{l} \approx \mathbf{0}$ and ${}^{n}\mathbf{g} \approx \frac{{}^{n}\mathbf{a}_n}{\|{}^{n}\mathbf{a}_n\|}$, which is valid when the sensor is stationary. The evaluation of ${}^{n}\mathbf{g}$ is thus free from any drift. In this way, the algorithm will compensate the accumulated rotation drift captured by the operation ${}^{n}\mathbf{g} \times {}^{n-1}\mathbf{g}$.

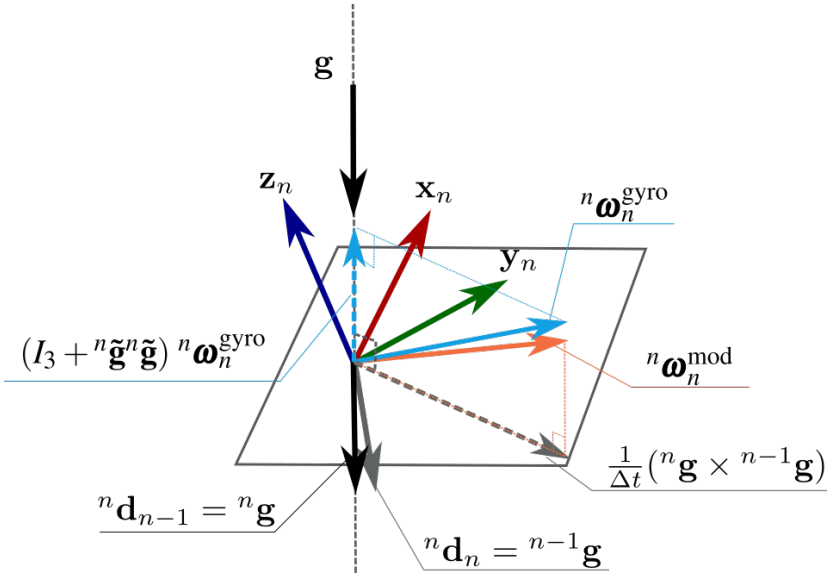

FIGURE 2.2: Illustration of the terms of the estimated angular velocity ${}^n\boldsymbol{\omega}_n^{\text{mod}}$ during stationary moments: the term $\frac{1}{\Delta t}({}^n\mathbf{g} \times {}^{n-1}\mathbf{g})$ is evaluated based on accelerometer measurement and the term $(I_3 + {}^n\tilde{\mathbf{g}}\,{}^n\tilde{\mathbf{g}})\,{}^n\boldsymbol{\omega}_n^{\text{gyro}}$ is evaluated based on gyroscope measurement.

The component of ${}^n\boldsymbol{\omega}_n^{\text{mod}}$ along the gravity direction $(I_3 + {}^n\tilde{\mathbf{g}}\,{}^n\tilde{\mathbf{g}})\,{}^n\boldsymbol{\omega}_n^{\text{gyro}}$ is still obtained from the gyroscope measurement and the drift along this direction is not compensated.

2.4 Stationary moment detection

We make a distinction between stationary moments and static moments. On the one hand, a static moment corresponds to inactivity [102]. Being able to identify static moments is useful for different cases. In pedestrian inertial navigation, static phases are observed and their detection allows one to apply a double integration of the linear acceleration only during motion phases [54, 74]. In motion recognition, the identification of static moments enables one to segment the flow of data [126]. For IMU orientation estimation, an estimation of the gyroscope bias can be done during the static phase to limit the drift. On the other hand, a stationary moment corresponds to a moment when the sensor is not undergoing any linear acceleration. This means that the sensor is considered stationary when it is completely static or under a non-accelerated motion. During such moments the accelerometer gives a reliable measurement of the gravity vector. In other words, in such moments, the linear acceleration is negligible, meaning ${}^{n}\mathbf{g} \approx \frac{{}^{n}\mathbf{a}_n}{\|{}^{n}\mathbf{a}_n\|}$. The proposed algorithm requires a criterion to detect stationary moments at which Eq. (2.10) is applicable.

2.4.1 Accelerometer norm criterion

A common method to detect stationary moments is to compare the norm of the accelerometer measurement to the value of the gravity acceleration g as $|\|{}^{n}\mathbf{a}_n\| - g| < k_a$ with the threshold k_a tuned to $k_a = 0.1g$, where g is the gravity acceleration, see, for instance, [51]. We propose to analyze the validity of this criterion experimentally. An IMU sensor is mounted on a robot end-effector (see Figure 2.3). The robot imposes a rotation to its end-effector while keeping the same end-effector position. A lever effect due to the length L, which is approximately 60 mm, introduces a linear acceleration. Figure 2.4 represents the norm of the accelerometer data and the error on the gravity vector. The reference gravity vector ${}^{EEn}\mathbf{g}_{robot}$ is computed from the robot end-effector orientation measured through its encoders. It is considered that the axis $\mathbf{z}$ of the robot base frame R is along the gravity direction so that ${}^{R}\mathbf{g}_{robot} = [0\ 0\ -g]^T$. So the reference gravity vector ${}^{EEn}\mathbf{g}_{robot}$ is computed as :

$$ {}^{EEn}\hat{g}_{robot} = {}^{R}\hat{q}_{EEn}^{-1} \otimes {}^{R}\hat{g}_{robot} \otimes {}^{R}\hat{q}_{EEn} \tag{2.13} $$

The sensor is mounted in a way that the frames EEn and n are aligned, so ${}^{EEn}\hat{g}_{robot} = {}^{n}\hat{g}_{robot}$. The estimated gravity vector ${}^{n}\mathbf{g}$ from sensor data is computed as ${}^{n}\mathbf{g} = \frac{{}^{n}\mathbf{a}_n}{\|{}^{n}\mathbf{a}_n\|}$. The error ε_n on the estimated gravity vector is computed as

$$ \varepsilon_n = \text{atan}(\|{}^{n}\mathbf{g}_{robot} \times {}^{n}\mathbf{g}\|\ ,\ {}^{n}\mathbf{g}_{robot} \cdot {}^{n}\mathbf{g}) \tag{2.14} $$

with the scalar product noted $\cdot$. As can be seen in Figure 2.4, the robot is in motion between 12.5 s and 13.1 s. At the beginning and at the end of the motion, the error on the gravity vector estimation is pretty high and the norm of the accelerometer data is out of the threshold k_a. During the motion, the norm of the accelerometer is within the threshold, however, the accelerometer measurement is affected by the linear acceleration ${}^{n}\mathbf{l}(t)$, leading to an important error on the gravity vector orientation. This means that the accelerometer norm criterion is not sufficiently sensitive to isolate stationary phases.

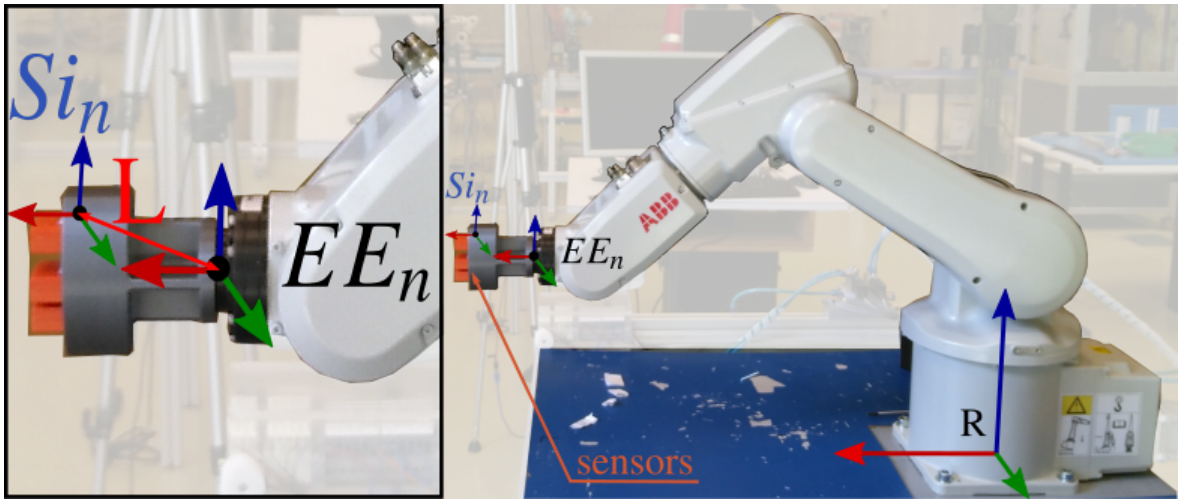

FIGURE 2.3: Experimental setup error measurement of sensor orientation estimation from Algorithm 1. Three IMUs are attached to the robot end-effector. L is the distance between the robot end-effector frame EEn and the sensor frame Sin. The robot motion is planned such that the origin of the frame EEn is fixed.

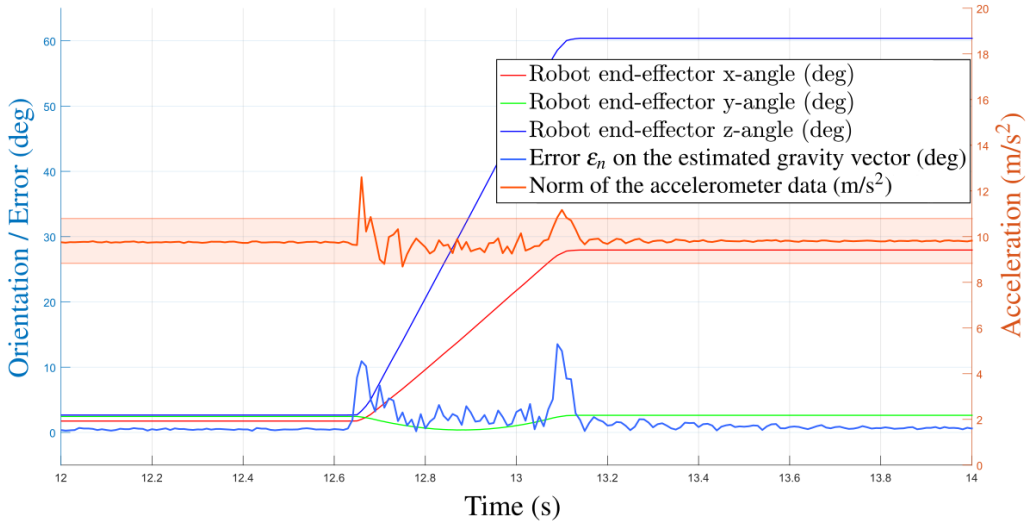

FIGURE 2.4: Experimental comparison between the norm of the accelerometer data and the error on the gravity vector during a motion.

2.4.2 Criterion on the norm of the derivative of the accelerometer

The previous experiment shows that, when the end-effector undergoes a rotational motion, the accelerometer signals do not provide a reliable estimation of the gravity vector even if the criterion $|\|{}^n\mathbf{a}\| - g| < k_a$ is satisfied. Such rotational motion implies a time evolution of ${}^n\mathbf{g}$, which itself induces a time evolution of ${}^n\mathbf{a}$, according to Eq. (2.3). Therefore, we propose an additional criterion on the time derivative $\frac{d^n\mathbf{a}}{dt}$ to refine the detection of stationary phases. Then, if $\|\frac{d^n\mathbf{a}}{dt}\|$ is higher than a certain threshold, it is considered that the motion is not stationary and that Eq. (2.10) cannot be used.

2.5 Heading reset

2.5.1 Analysis of the heading error

The measure of the gravity vector thanks to the accelerometer can be exploited to catch up the drift due to gyroscope data integration along horizontal axes. This can be easily seen in simulated data. Let us consider a trajectory of a body frame n consisting of 3 successive rotations around each axis of $\frac{\pi}{2}$ with respect to a reference inertial frame 0 as shown in Figure 2.5. Frame 0 has its **z**-axis aligned with the gravity direction and pointing upward. From this reference trajectory, the angular velocity and the gravity vector with respect to the body frame n can be computed. The body frame position does not change during the trajectory, the linear acceleration is then null all along the trajectory. The simulated accelerometer data is then equal to gravity vector. In order to simulate gyroscope data, an offset o of $o = [0.01\ 0.01\ 0.01]^T$ rad is added to the angular velocity. Simulated sensors data are shown in Figure 2.6. The orientation of the body frame is then computed from the simulated data first only with gyroscope data integration, then using in addition the correction from accelerometer data. The orientations are represented in Figure 2.7. The error is represented in Figure 2.8. In Figure 2.8 b), the components of the error along **x** and **y** axes is null when the accumulated error along this direction is still present in Figure 2.8 a). On the other side, the **z**-axis component of the error remains the same in Figures 2.8 a) and b) which reveals that the drift along this direction is not caught up. Therefore a solution to catch the remaining drift is desirable.

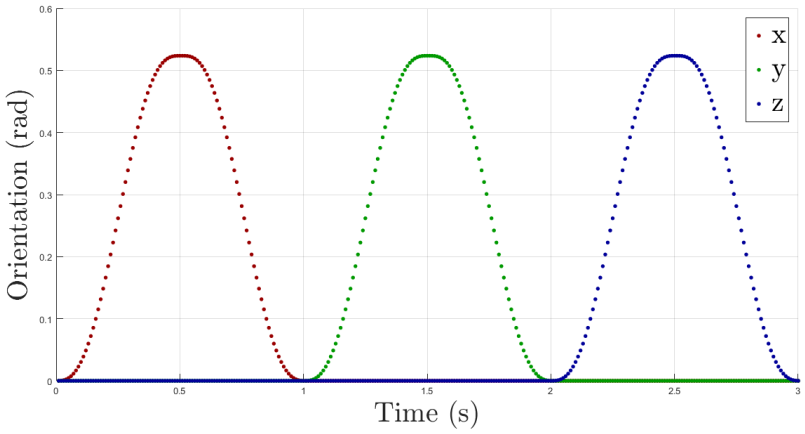

FIGURE 2.5: Simulated trajectory composed of 3 rotations.

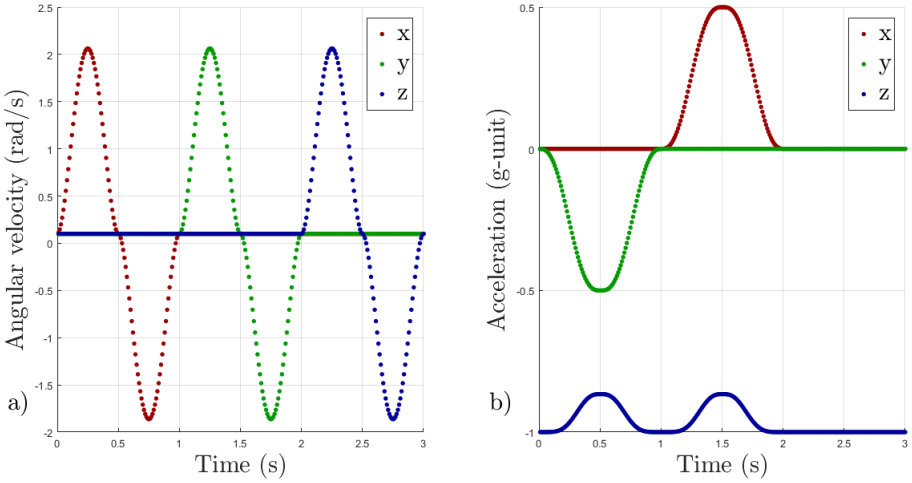

FIGURE 2.6: Simulated a) gyroscope and b) accelerometer data.

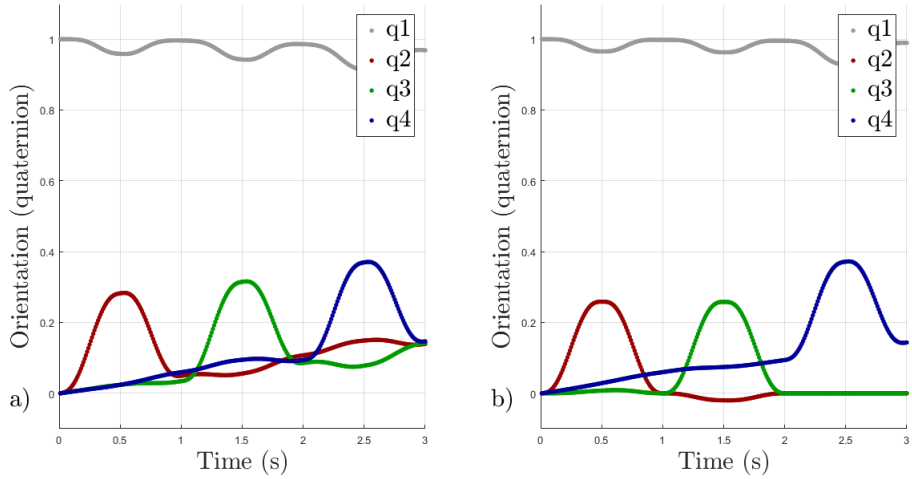

FIGURE 2.7: Orientation of the body frame with respect to the reference frame from a) gyroscope data integration and b) with correction from accelerometer data in addition.

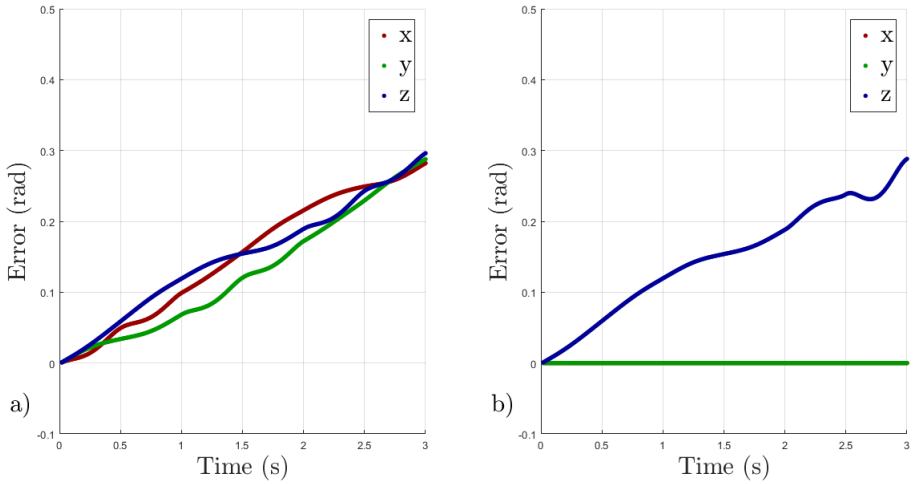

FIGURE 2.8: Body frame orientation error from a) gyroscope data integration and b) with correction from accelerometer data in addition.

2.5.2 Heading reset algorithm

Definition and hypothesis

The heading of a sensor is defined as the part of its orientation around the gravity direction. As seen in the previous section, the heading is still affected by the drift even after the compensation method based on the observation of the gravity vector.

The proposed approach to limit the drift around the gravity direction, called heading reset, relies on a hypothesis on the human behaviour in a context of teaching task. In this specific context, it is assumed that the initial heading orientation of the arm segments will be frequently visited with stationary phases during the task. Some elements in the literature tend to support this assumption. In a first-person vision study, a camera is set on the head of the human who has to perform a task in front of him. In [12], Bandini, et al. state that "hands and manipulated objects tend to appear at the center of the image". Furthermore, this position is also centrally located in the workspace of the human arm as discussed in [36, 113]. This indicates that the position of the human hand in front of the subject is a privileged position which is frequently visited during a manipulation task. In this work, we assume that a privileged heading orientation exists for each segment of the arm. It can be mentioned that such privileged heading may be encountered during the task independently of the hand position.

When the arm segment heading passes by its privileged initial heading, we propose to apply a reset of the heading of the sensor.

Formulation

The heading reset is applied through a correction ${}^{n}\boldsymbol{\omega}_{n}^{\text{reset}}$ at the angular velocity level. The total rotation ${}^{0}\hat{q}_{n-1}$ can be decomposed into an inclination ${}^{0}\hat{q}_{*}$ and a heading part ${}^{*}\hat{q}_{n-1}$ as

$$ {}^{0}\hat{q}_{n-1} = {}^{0}\hat{q}_{*} \otimes {}^{*}\hat{q}_{n-1}. \tag{2.15} $$

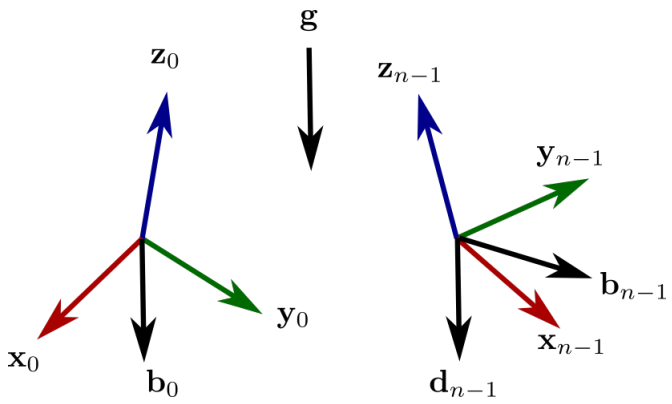

FIGURE 2.9: Definition of the vector **b** attached to the sensor frame and aligned with the gravity vector at the initial time step.

The rotation axis of the heading part ${}^*\hat{q}_{n-1}$ is the gravity vector, so ${}^{n-1}\mathbf{g} = {}^*\mathbf{g}$. Similarly to the vector **d** previously defined, a vector **b** is defined as a vector constant for an observer following the sensor frame and aligned with the gravity vector at time step 0: ${}^0\mathbf{b}_0 = {}^0\mathbf{g} = {}^{n-1}\mathbf{b}_{n-1}$ (see Figure 2.9). The rotation ${}^*\hat{q}_0$ is the inclination part and its rotation axis ${}^*\boldsymbol{\theta}_0$ is computed as

$$ {}^*\boldsymbol{\theta}_0 = \frac{{}^0\mathbf{g} \times {}^{n-1}\mathbf{g}}{\|{}^0\mathbf{g} \times {}^{n-1}\mathbf{g}\|} \text{asin}(\|{}^0\mathbf{g} \times {}^{n-1}\mathbf{g}\|). \tag{2.16} $$

Therefore ${}^*\hat{q}_0$ is computed as

$$ {}^*\hat{q}_0 = \exp({}^*\boldsymbol{\theta}_0). \tag{2.17} $$

Then ${}^n\boldsymbol{\omega}_n^{\text{reset}}$ is computed as

$$ {}^*\hat{q}_{n-1} = {}^*\hat{q}_0 \otimes {}^0\hat{q}_{n-1} \tag{2.18} $$

$$ {}^n\boldsymbol{\omega}_n^{\text{reset}} = \frac{1}{\Delta t} \log({}^*\hat{q}_{n-1}) \tag{2.19} $$

and is subtracted to the angular velocity.

The sensor is considered close to its initial heading when $\|\Delta t \, {}^n\boldsymbol{\omega}_n^{\text{reset}}\| < k_r$. A too low value of the tolerance k_r will not catch the drift and a too high value will annihilate any rotation around the gravity direction. The value of this tolerance is tuned at 0.1 rad meaning that the sensor is considered close to its original heading when the estimated heading is under 0.1 rad. In order to limit overly heading reset, the corrective term is applied only when the sensor is stationary. This extra condition to apply the reset is necessary since, by definition a sensor starts from its original heading. Without any restriction, the reset could be applied to each iteration annihilating any rotation around the gravity direction. Applying the reset only when the sensor is stationary enables to catch up the accumulated error from gyroscope data integration but not the heading component of the sensor orientation.

2.6 Summary

In this chapter, the complete algorithm used in this work for IMU orientation estimation is presented. This algorithm is based on the integration of the gyroscope data and the gravity estimation from accelerometer data. It is summarized in Algorithm 1.

The drift from the integration of gyroscope data can be partially caught up thanks to the gravity estimation in stationary moments (see Algorithm 1 line 5). The moments where the sensor is stationary, i.e., not undergoing any linear acceleration, are defined by a criterion involving both the norm of the accelerometer signal and the norm of its derivative (see Algorithm 1 line 1).

In order to limit the drift of the sensor orientation around the gravity direction, an original heading reset method is proposed. This method is based on the hypothesis that, during a demonstration, the teacher is expected to execute motion where sensors pass by their original heading several times with stationary phases. When the sensor is both stationary and close to its original heading, a reset is applied at the angular velocity level (see Algorithm 1 line 6-10).

The following chapter presents the experiments conducted to analyze the different elements of the method presented in this chapter: the extra criteria for stationary moment detection and the heading reset.

Algorithm 1 : Complete IMU orientation algorithm

1:	**if** $\lvert\lVert {}^{n}\mathbf{a}_n\rVert - g\rvert < k_a$ **and** $\lVert\frac{{}^{n}\mathbf{a}_n - {}^{n-1}\mathbf{a}_{n-1}}{\Delta t}\rVert < k_d$ **then**	(Stationary condition - Section 2.4)
2:	$\quad {}^{0}\mathbf{g} = \frac{{}^{0}\mathbf{a}_0}{\lVert {}^{0}\mathbf{a}_0\rVert}$	
3:	$\quad {}^{n}\mathbf{g} = \frac{{}^{n}\mathbf{a}_n}{\lVert {}^{n}\mathbf{a}_n\rVert}$	
4:	$\quad {}^{n-1}\hat{g} = {}^{n-1}\hat{q}_0 \otimes {}^{0}\hat{g} \otimes {}^{0}q_{n-1}$	
5:	$\quad {}^{n}\boldsymbol{\omega}_n^{\text{mod}} = \frac{1}{\Delta t}({}^{n}\mathbf{g} \times {}^{n-1}\mathbf{g}) + (I_3 + {}^{n}\tilde{\mathbf{g}}\,{}^{n}\tilde{\mathbf{g}})\,{}^{n}\boldsymbol{\omega}_n^{\text{gyro}}$	(Eq. (2.10))
6:	$\quad {}^{*}\boldsymbol{\theta}_0 = \frac{{}^{0}\mathbf{g} \times {}^{n-1}\mathbf{g}}{\lVert {}^{0}\mathbf{g} \times {}^{n-1}\mathbf{g}\rVert}\,\text{asin}(\lVert {}^{0}\mathbf{g} \times {}^{n-1}\mathbf{g}\rVert)$	(Eq. (2.16))
7:	$\quad {}^{*}\hat{q}_{n-1} = \exp({}^{*}\boldsymbol{\theta}_0) \otimes {}^{0}\hat{q}_{n-1}$	(Eq. (2.17) and Eq. (2.18))
8:	$\quad {}^{n}\boldsymbol{\omega}_n^{\text{reset}} = \frac{1}{\Delta t}\log({}^{*}\hat{q}_{n-1})$	(Eq. (2.19))
9:	$\quad$ **if** $\lVert \Delta t\,{}^{n}\boldsymbol{\omega}_n^{\text{reset}}\rVert < k_r$ **then**	(Heading reset condition - Section 2.5)
10:	$\quad\quad {}^{n}\boldsymbol{\omega}_n = {}^{n}\boldsymbol{\omega}_n^{\text{mod}} - {}^{n}\boldsymbol{\omega}_n^{\text{reset}}$	
11:	$\quad$ **else**	
12:	$\quad\quad {}^{n}\boldsymbol{\omega}_n = {}^{n}\boldsymbol{\omega}_n^{\text{mod}}$	
13:	$\quad$ **end if**	
14:	**else**	
15:	$\quad {}^{n}\boldsymbol{\omega}_n = {}^{n}\boldsymbol{\omega}_n^{\text{gyro}}$	
16:	**end if**	
17:	${}^{0}\hat{q}_n = {}^{0}\hat{q}_{n-1} \otimes \exp(\frac{1}{2}\Delta t\,{}^{n}\boldsymbol{\omega}_n)$	(Eq. (2.5))

CHAPTER 3

EXPERIMENTAL TESTS ON THE IMU ORIENTATION ESTIMATION METHOD

Contents

In this chapter, the method for IMU orientation estimation is implemented, tested and compared with other algorithms. The objective is to experimentally evaluate the method in the context of the acquisition of demonstrations for robot PbD. More precisely, the experimental study has two goals:

- evaluate the efficiency of the new elements in the IMU estimation algorithm: the stationary moment detection and the heading reset in a context of human motion tracking for PbD,
- evaluate the accuracy of the complete algorithm described in Algorithm 1.

3.1 IMU orientation estimation algorithms for comparison

Some comparisons of IMU orientation estimation algorithms can be found in the literature. Michel et al. compared seven MARG algorithms in [94] for pedestrian navigation with smartphones under different specific situations (texting, phoning, swinging and back pocket static). The mean error on orientation is found in an interval between 4.7° and 9.5° for a measurement that lasts 180 seconds. In [26], different fusion methods are evaluated in the context of a grasping task. The average orientation error is found in an interval between 5.9° and 20.4° for all the tested algorithms.

According to these studies each method has its own advantages and drawbacks and the optimal choice depends on the targeted application. This motivates the development of our own evaluation in a manner that reflects the condition of the targeted application. The targeted application being human motion tracking for robot programming by demonstration, the experiment should include the following constraints:

- the motion imposed to the sensor during the experiment should be humanly realistic,
- the experiment should be conducted in a robotic environment.

We propose to compare our method to the following algorithms:

- Xsens (XKF3-hm) algorithm [15] developed by the company Xsens, based on a Kalman filter and specially designed for human motion tracking.
- Madgwick's algorithm [88] based on a gradient descent.
- Mahony's algorithm [89] based on a complementary filter.
- Fourati's algorithm [52] based on an integration and vector observation approach.

The last three algorithms are known and a description is given in Appendix A. These algorithms have been chosen because they are based on the main approaches found in the literature and can work without magnetometers. The Xsens algorithm being embedded in a commercial system, the exact algorithm is unknown. However, according to [15], this algorithm uses magnetometer data to estimate the Earth magnetic north for an absolute orientation estimation but only when it is reliable. It can therefore be assumed that, in the environment of our robotics laboratory, this algorithm discards the magnetometer data most of the time.

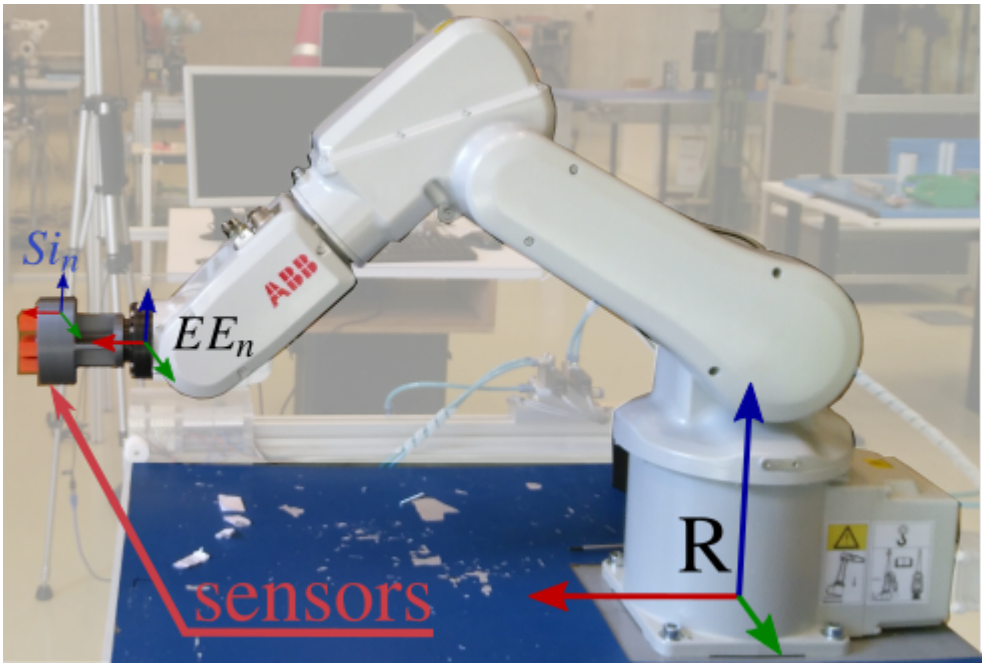

FIGURE 3.1: Experimental setup for IMU orientation estimation method assessment.

3.2 Experimental protocol

In this experiment, the robot's end-effector orientation is estimated using IMUs and the precise measurement of the end-effector orientation using robot's encoders is used as reference. The equipment involved in this experiment is described in the first subsection and the second subsection describes how the human realistic motion imposed to the robot is built.

3.2.1 Experimental setup

The proposed orientation estimation algorithm has been applied to the filtered accelerometer and gyroscope data from Xsens MTw Awinda sensors [15]. It is considered that the filtered data from commercial sensors present limited noise. The sensors communicate wirelessly by radio frequency at a sample rate of 100 Hz. Three sensors are mounted on the end-effector of an ABB IRB 120 industrial robotic arm (see Figure 3.1). The bracket that links the sensors to the robot is 3D printed and ensures that the sensors frames are aligned with the robot end-effector frame. The initial sensor frame $Si0$ is aligned with the robot base frame R as well as the body-attached frame Sin and the end-effector frame EEn. The robot end-effector orientation ${}^{R}\hat{q}_{EEn}$, evaluated from encoder measurements, is used as reference. The orientation error ε is computed as

$$ {}^{R}\boldsymbol{\theta} = \log({}^{Si0}\hat{q}_{Sin} \otimes {}^{R}\hat{q}_{EEn}^{-1}) \tag{3.1} $$

$$ \varepsilon = \|{}^{R}\boldsymbol{\theta}\| \tag{3.2} $$

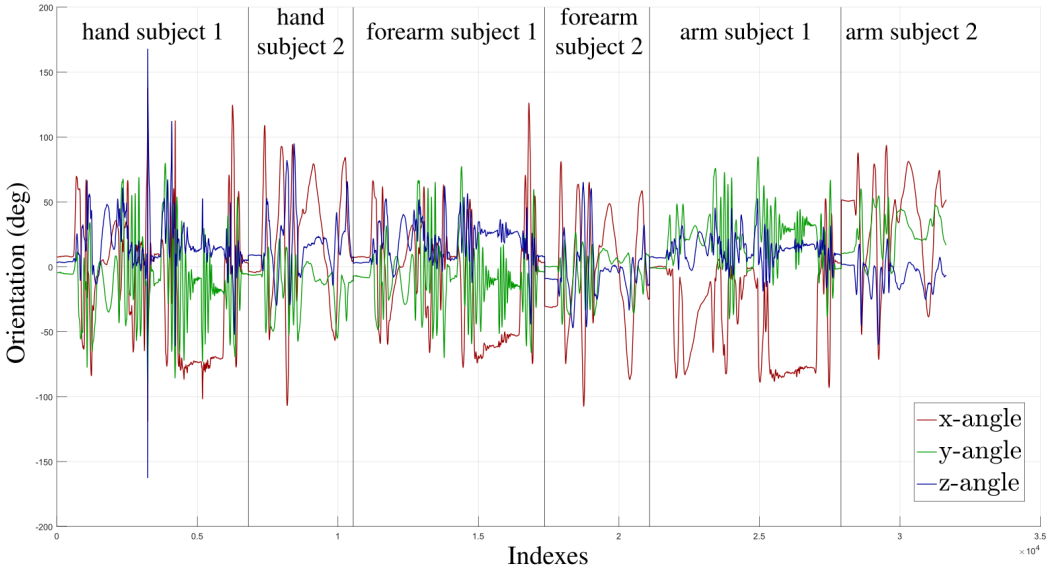

FIGURE 3.2: Human arm segment orientation acquisition as rotation vector.

3.2.2 Robot motion

In order to impose some orientations to the robot that reflect a realistic human arm motion, a set of orientation values is created from measurements of the motion of the arm of two subjects. The two subjects wear one Xsens MTw Awinda IMU sensor on each segment (hand, forearm and arm) and they are asked to move their arm freely during some time (68 seconds for subjects 1 and 37 seconds for subject 2) with the only instruction to cover as much as possible the range of human arm motion. During that time, the orientations of each sensor, estimated by the Xsens algorithm, are recorded at a sample rate of 100 Hz to constitute a set of human realistic orientations (see Figure 3.2). This recording has been conducted indoor but far from any obvious source of magnetic disturbances for consistent orientation estimation. During the recording, the gyroscope measured angular velocity takes its value between 0.9°/s and 505.8°/s. The measured motion is comparable to the values presented in [27] for tasks of everyday life and are thus humanly realistic in the targeted context of this work. This motion is played by the robot in two different ways: continuously, then with a 1 second break between each orientation. Furthermore, this motion is repeated 5 times at different maximum robot end-effector velocities: 200mm/s and 200°/s, 100mm/s and 100°/s, 50mm/s and 50°/s, 100mm/s and 100°/s and then 200mm/s and 200°/s. This sequence is illustrated in Figure 3.3. Some orientations from the initial dataset are discarded because the robot is not able to reach them. This set of repeated trajectories constitutes a motion that lasts 3707 seconds so that the gyroscope bias has a significant impact on the motion estimation. The described motion is executed 3 times by the robot and each time, 3 sensors measure the end-effector orientation.

No reference values regarding accelerations of the human arm segment have been found in the literature to verify the human realism of the robot trajectory. In the proposed trajectory, the TCP of the robot, corresponding to the center of the EE_n frame, is fixed during the complete measurement. The distance between the center of EE_n and the sensors is approximately 60 mm. This distance, combined with the robot angular velocity, induces a linear acceleration within a similar range than human motion observed during our acquisitions with the two subjects. The

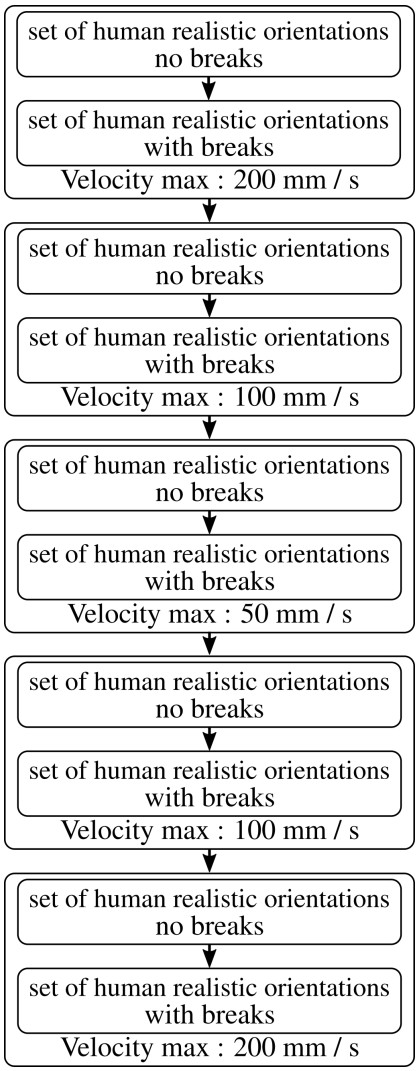

FIGURE 3.3: Sequence of imposed robot motion for the experimental evaluation of the sensor orientation method.

norm of the acceleration measured during our recordings of the human motion is between 0.5 m/s^2 and 41.4 m/s^2 with a mean value of 10.0 m/s^2. It encompasses the interval of the norm of the linear acceleration measured by the sensors during the robot experiment which is between 0.9 m/s^2 and 22.0 m/s^2 with a mean value of 9.8 m/s^2. The motion of the robot during the experiment can be considered humanly realistic regarding the acceleration.

3.3 Results: focus on the stationary moment detection

This section aims at studying the influence of the criterion based on the norm of the accelerometer signals and their derivative for stationary moment detection. For this purpose, two versions of the complete algorithm (see Algorithm 1) are implemented :

- a basic version: no heading reset is implemented and the stationary moment detection is only based on the norm of the accelerometer,
- an intermediate version: no heading reset is implemented and the stationary moment detection is based on the norm of the accelerometer signals and on the norm of their derivative.

Table 3.3 presents the error on the sensor orientation for the two versions of the proposed algorithm. The orientation error from the Xsens commercial algorithm is presented as indicative information. The color cases highlight the best (green) and the worst (red) error values between the basic and the intermediate version of Algorithm 1. The discrepancies of the results over the three measurements despite the same robot motion can be explained from the sensor noise that has a non-deterministic influence.

Measurement	Sensors	Xsens algorithm		basic version		intermediate version	
		ε (in°)	std (in°)	ε (in°)	std (in°)	ε (in°)	std (in°)
1	1	34.8	15.9	6.8	5.2	26.3	12.0
	2	25.4	13.9	16.3	12.4	14.4	12.5
	3	12.5	6.3	6.8	4.8	35.0	19.2
2	1	19.9	13.1	6.9	5.0	29.7	15.7
	2	10.2	5.2	8.6	5.6	5.1	2.4
	3	37.7	11.9	23.0	12.3	14.3	5.2
3	1	26.6	14.6	31.0	12.8	7.2	3.7
	2	16.4	11.7	53.4	17.5	6.5	3.1
	3	8.4	3.0	8.3	5.6	5.6	3.8

TABLE 3.1: Mean error of the sensor orientation estimated from the basic and the intermediate versions of Algorithm 1.

First, both the basic version and the intermediate version are better than the Xsens algorithm for 7 out of 9 measurements. The Xsens algorithm is designed for human motion tracking but not specially designed to work in the environment of a robotics laboratory. These results show that the proposed angular velocity and vector observation algorithm is appropriate for the targeted application.

Secondly, concerning the impact of the extra criterion on the derivative of the accelerometer signals for stationary moment detection, 6 cases on 9 present an improvement in the estimation of the orientation. In the other 3 cases, a deterioration can be observed. Even if the experimental results are not fully positives, they tend to confirm the positive impact of the extra criterion to detect stationary moments. In the following analysis, the intervals of stationary moments are compared for a better understanding of the differences between the basic and the intermediate versions of the algorithm.

A reference criterion for stationary moments detection based on the robot orientation is proposed for validation purpose. As the **z**-axis of the robot base frame is aligned with the gravity direction: ${}^{R}\mathbf{g} = [0\,0\,-1]$ g-unit. So, the gravity vector with respect to the end-effector frame can be reliably computed as

$$ {}^{EEn}\hat{g} = {}^{EEn}\hat{q}_R \otimes {}^{R}\hat{g} \otimes {}^{EEn}\hat{q}_R^{-1} \ . \tag{3.3} $$

Then, if the angle between ${}^{EEn}\mathbf{g}$ and $\mathbf{a}_n$ is under a certain threshold $k = 3^\circ$, it is considered that the moment is stationary. The percentages of stationary moments according to each criterion is estimated and presented in Table 3.2. According to the reference criterion, over all sensors and measurements, the percentage of stationary moments is around 90 %. The percentage of stationary moments detected with the criterion only based on the accelerometer norm is around 98 %. The comparison of the first and the last but one column shows that almost all the reference stationary moments are detected with this criterion. So, the criterion based on the accelerometer norm generates about 8 % of false positive in the detection of stationary moments. On the other side, only 66 % of stationary moments are detected with the complete criterion and almost all of these correspond to the reference stationary moments. Even if the optimal criterion is case-dependent (see Table 3.2) it seems safer to detect a lower number of stationary moments than to consider as stationary some moments that are not. Therefore, the criterion based on both $\|{}^{n}\mathbf{a}\|$ and $\|\frac{d^n\mathbf{a}}{dt}\|$ is kept in this work.

Measurement	Sensors	Reference criterion	$\|\|{}^{n}\mathbf{a}_n\| - g\| < k_a$	$\|\|{}^{n}\mathbf{a}_n\| - g\| < k_a$ **and** $\|\frac{{}^{n}\mathbf{a}_n - {}^{n-1}\mathbf{a}_{n-1}}{\Delta t}\| < k_d$	Reference criterion **and** $\|\|{}^{n}\mathbf{a}_n\| - g\| < k_a$	Reference criterion **and** $\|\|{}^{n}\mathbf{a}_n\| - g\| < k_a$ **and** $\|\frac{{}^{n}\mathbf{a}_n - {}^{n-1}\mathbf{a}_{n-1}}{\Delta t}\| < k_d$
1	1	90.9	98.8	66.3	90.8	66.2
	2	90.4	98.6	65.8	90.3	65.7
	3	90.8	98.7	66.0	90.7	65.9
1	1	92.9	98.8	65.9	92.8	65.9
	2	92.1	98.6	65.5	92.0	65.5
	3	92.8	98.7	65.9	92.7	65.9
3	1	92.5	98.7	66.3	92.4	66.3
	2	92.0	98.8	66.1	91.9	66.0
	3	92.1	98.6	65.8	92.0	65.8

TABLE 3.2: Percentages of stationary moments according to different criteria.

3.4 Results: focus on the heading reset

3.4.1 Results

Let us recall that the robot trajectory is constructed such that the end-effector orientations are humanly realistic. No specific instruction was given to the two subjects and no adaptation of the robot trajectory to ensure the heading reset would be applicable.

In order to study the impact of the heading reset, the complete algorithm (see Algorithm 1) is compared to the intermediate version without the heading reset as well as the 4 algorithms presented in Section 3.1. The results of the comparative study are presented in Table 3.3. The influence of the different elements of the algorithm is illustrated in Figure 3.4. This figure represents the orientation error of the sensor 3 during the measurement 3 for different versions of the complete algorithm. Figure 3.4 a is the orientation estimated only from gyroscope integration. The drift is highly visible on each component. Figure 3.4 b shows the error for the intermediate version. The drift is compensated on the 2 components along the **x** and **y**-axes which are perpendicular to the gravity direction. The last component, corresponding to the **z**-axis (aligned with the gravity direction), is still subject to the drift. Finally, Figure 3.4 c shows the impact of the heading reset. The drift on the 3rd component is caught up.

According to Table 3.3, the intermediate version presents quite good results compared to the 4 other algorithms (Xsens, Mahony's, Madgwick's and Fourati's). In 6 cases, it gives the best results over the 4 algorithms. Furthermore, the maximum error is 35.0 ° (similar to the Xsens algorithm) when the maximum error for the other algorithms are 118.4 °, 154.9 ° and 87.4 °. In this context, the intermediate version appears more robust than Mahony's, Madgwick's and Fourati's algorithms and can be considered as a reliable base for inertial human motion tracking in a context of PbD.

Regarding the complete algorithm, the results are even better since, it gives the best results in all cases. Furthermore, most of all the error values are below 5 ° when none of the other algorithms reach this level of accuracy. These encouraging results tend to validate the hypothesis made on human motion in the context of manipulation task. However, the behaviour of the heading reset during the trajectory depends on the value of k_r. For the results presented in this section, k_r is tuned to 0.1 rad. The next section studies the influence of this parameter on the orientation estimation error.

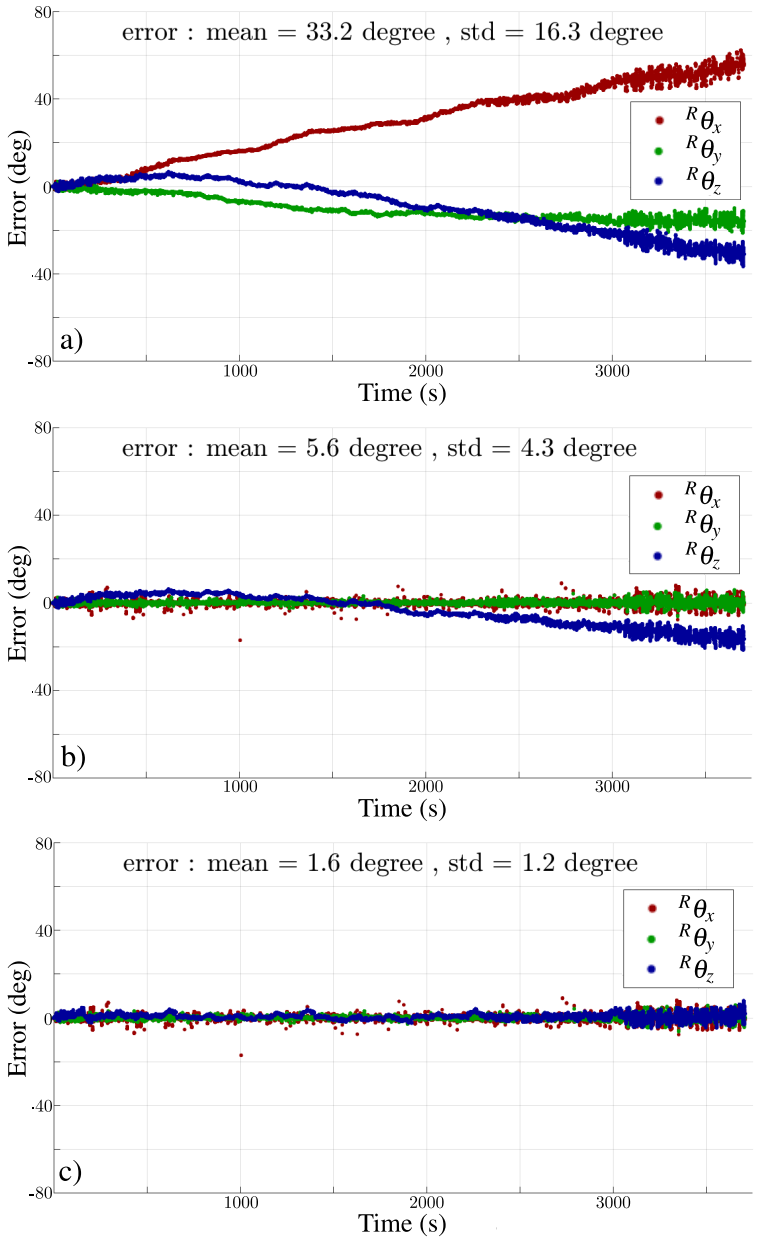

FIGURE 3.4: Mean error for the sensor 3 during measurement 3: a) gyroscope integration only, b) intermediate version, c) complete algorithm.

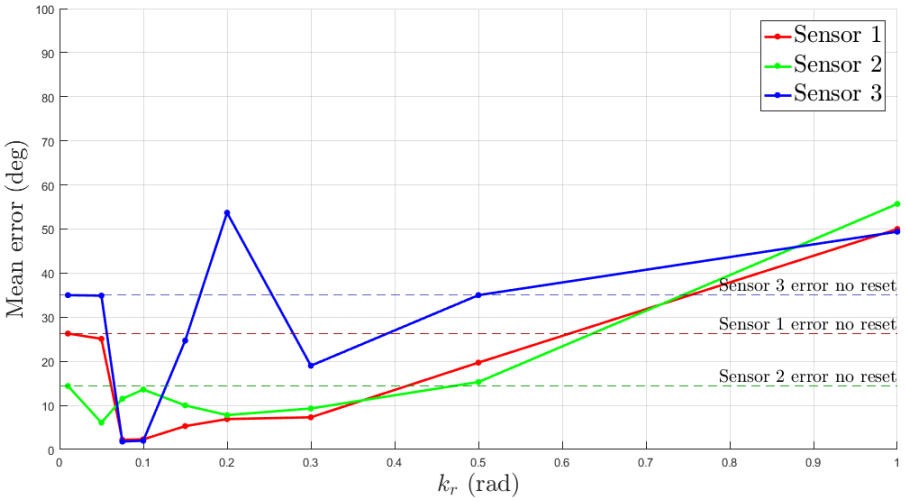

FIGURE 3.5: Mean error on orientation estimation for different values of k_r for measurement 1

Measurement	Sensors	Xsens algorithm		Mahony's algorithm		Madgwick's algorithm		Fourati's algorithm		Intermediate version (without $\boldsymbol{\omega}^{\text{reset}}$)		Complete algorithm (with $\boldsymbol{\omega}^{\text{reset}}$)	
		ε	std	ε	std	ε	std	ε	std	ε	std	ε	std
1	1	34.8	15.9	23.3	9.2	24.8	10.0	24.8	10.0	26.3	12.1	2.3	1.5
	2	25.4	13.9	118.4	45.7	139.1	42.1	65.3	36.8	14.4	12.5	13.6	12.7
	3	12.5	6.3	38.4	15.3	154.9	4.5	33.3	16.3	35.0	19.2	2.0	1.5
2	1	19.9	13.1	21.9	7.7	30.3	13.1	30.4	13.2	29.7	15.7	2.3	1.1
	2	10.2	5.2	74.2	29.2	5.5	2.8	5.5	2.8	5.1	2.4	2.6	1.1
	3	37.7	11.9	116.3	44.7	110.5	45.8	87.4	46.6	14.3	5.2	4.0	2.1
3	1	26.6	14.6	107.5	46.2	97.0	46.4	81.4	46.2	7.2	3.7	2.1	1.7
	2	16.4	11.7	27.0	10.0	7.8	2.9	7.9	2.9	6.5	3.0	1.8	1.2
	3	8.4	3.0	50.9	22.7	47.8	21.0	6.2	5.3	5.6	3.8	1.6	1.3

TABLE 3.3: Error on the orientation of the proposed algorithm with and without heading reset, the Xsens algorithm and three other IMU algorithms.

3.4.2 Influence of k_r

This section studies the influence of the threshold k_r. The orientation of the sensor is estimated from the same data as presented in the previous section and using the complete algorithm but with different values of k_r. The errors are presented in Figures 3.5, 3.6 and 3.7. The error values for the intermediate algorithm, i.e., without heading reset, are represented on these figures by doted lines.

In most cases, the error according to the value of k_r presents a minimum around $k_r = 0.1$ rad. As explained in the previous section 2.5.2, the presence of a minimum is expected for the following reason. When k_r is too low, the drift is not caught up but when k_r is too high, the

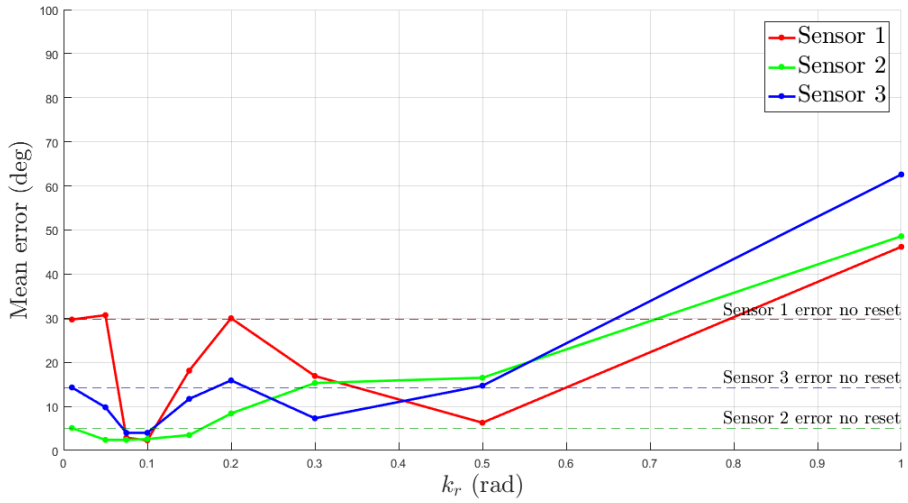

FIGURE 3.6: Mean error on orientation estimation for different values of k_r for measurement 2

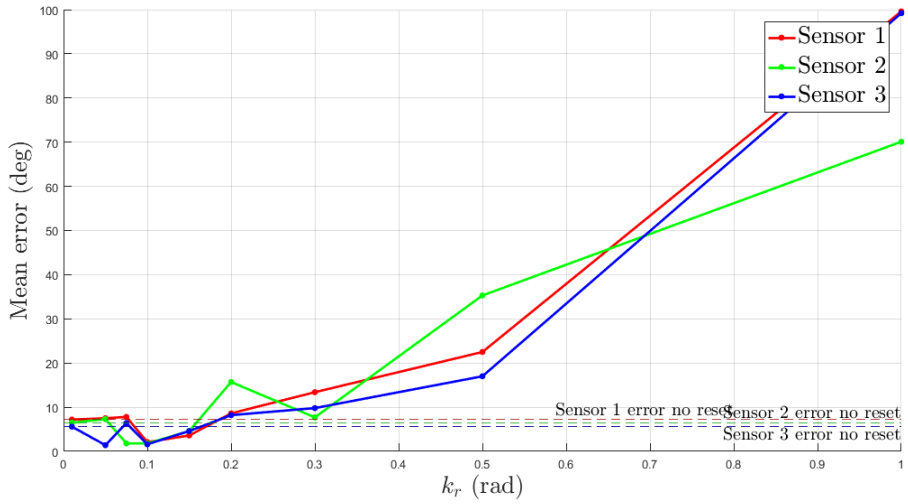

FIGURE 3.7: Mean error on orientation estimation for different values of k_r for measurement 3

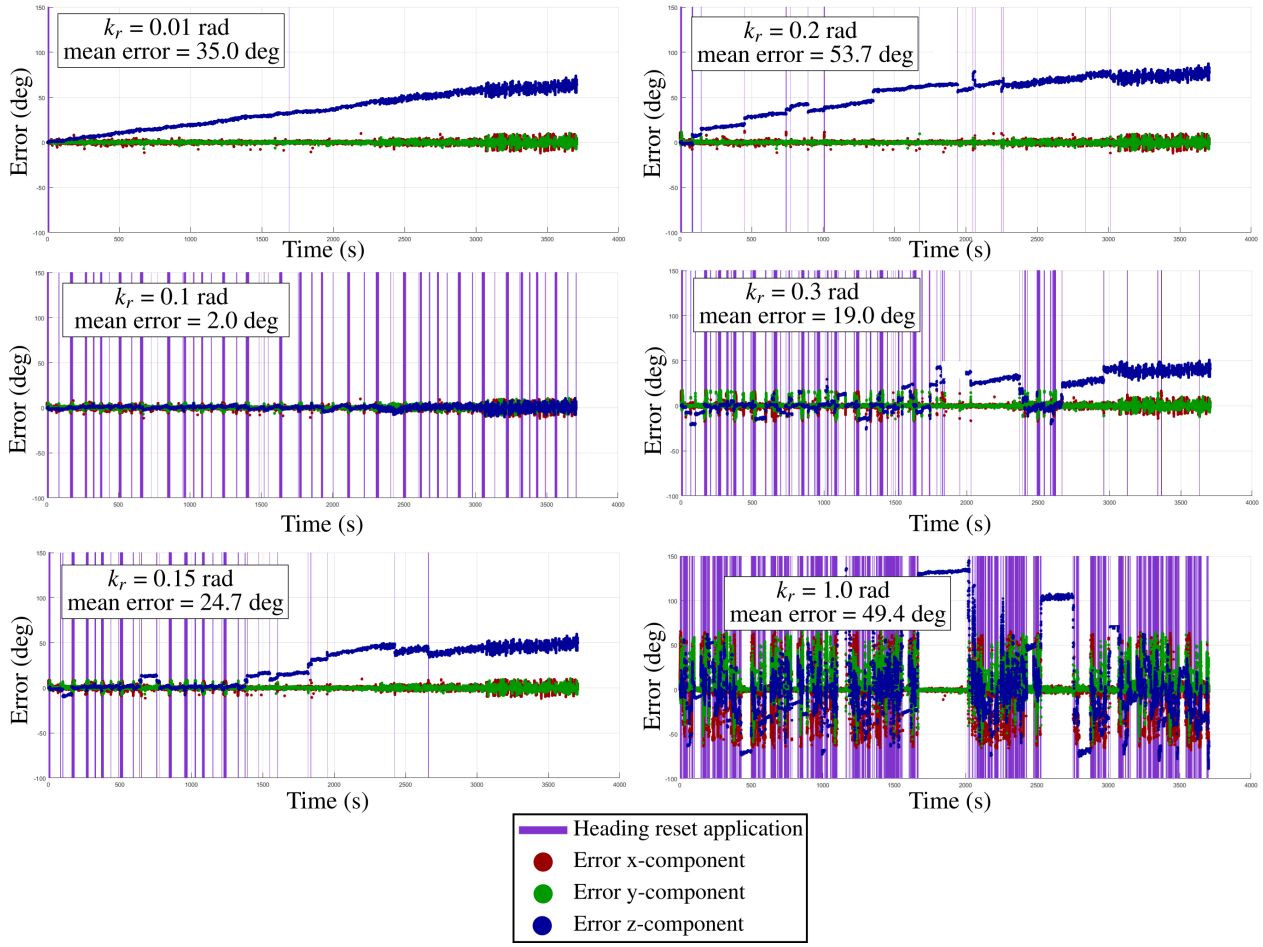

FIGURE 3.8: Error on the sensor orientation estimation and visualization of the heading reset applications for different values of k_r for sensor 3 at measurement 1.

rotation around the vertical axis can be annihilated. However, in some cases, an extra peak in the error can be observed:

- when $k_r = 0.2$ rad for sensor 3 in measurement 1, sensors 1 and 3 in measurement 2 and sensor 2 in measurement 3;
- when $k_r = 0.075$ rad for sensor 3 in measurement 3 ;
- when $k_r = 0.1$ rad for sensor 2 in measurement 1.

This peak can be higher than the error without the heading reset which can be seen as a lack of robustness. In the following, we give some explanation about this phenomenon based on the peak at $k_r = 0.2$ rad for sensor 3 at measurement 1 (Figure 3.8).

The heading reset can have a positive or a negative impact on the orientation error each time it is applied. If the difference between the estimated orientation without heading reset and the true orientation is higher than between the true orientation and 0, then the impact of the heading reset is positive, and the estimated orientation error decreases (see Figure 3.9). But, if the difference between the estimated orientation without heading reset and the true orientation is lower than the difference between the true orientation and 0, the heading reset has a negative impact

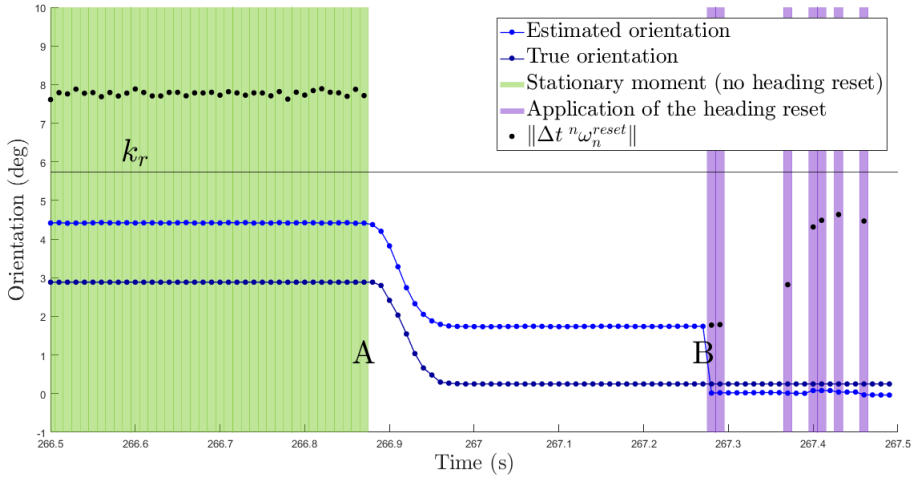

FIGURE 3.9: 3rd component of the orientation estimation for $k_r = 0.1$ rad to illustrate the positive impact of the heading reset. Before "A", the sensor is stationary, but its heading estimation is too high to apply the heading reset. At "B", the sensor is stationary again and close to its original heading, the heading reset is applied with a positive impact on the estimation of the orientation.

and increases the error (see Figure 3.10). In the case of $k_r = 0.2$ rad for sensor 3 at measurement 1, the heading reset presents too much negative impact at the beginning making it difficult to be caught up later. The error being too high, the estimated sensor orientation rarely comes back to the original heading, so that the heading reset cannot be applied properly. However, it is sometimes possible to catch up the drift after such situation. The evolution of the error for $k_r = 0.3$ rad around 2500 sec shows that the heading reset can catch up the drift even if the error is important (approximately 40 degrees).

The low level of errors obtained for $k_r = 0.1$ rad is encouraging, even though the above analysis shows that the results are sensitive to the value of k_r. The criterion to apply the heading reset remains a critical step of the procedure.

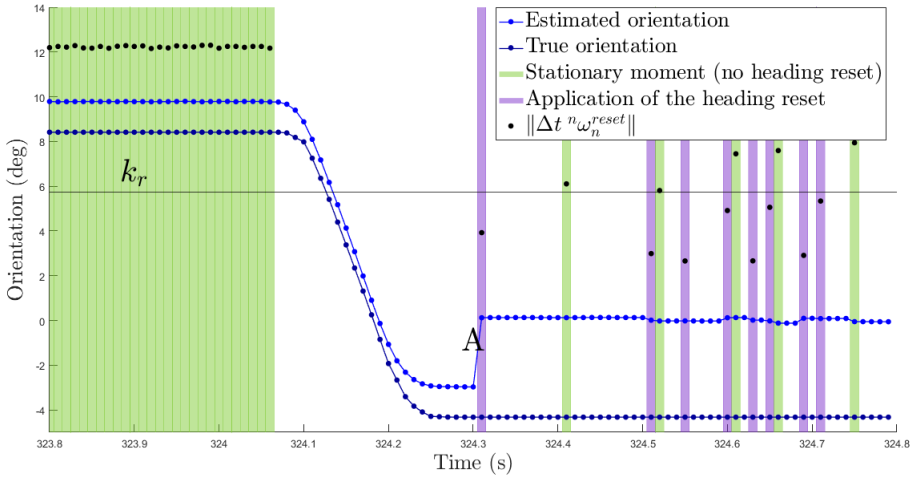

FIGURE 3.10: Error on the 3rd component of the orientation estimation for $k_r = 0.1$ rad which illustrates the negative impact of the heading reset in "A".

3.5 Summary

The method for IMU orientation estimation is implemented based on measurement signals of a commercial IMU (MTw Awinda XSens). The proposed method is compared to 4 other algorithms for robot motions which mimic human arm motions. The orientation measured through robot encoders is used as reference.

Firstly, experimental results show the added value of the criterion on the derivative of the accelerometer data to detect stationary moments. Within this criterion, the algorithm detects less stationary moments and is thus more conservative.

Secondly, the experimental results show that the heading reset can be used to catch up the drift on the gravity direction. The complete algorithm including the heading reset gives encouraging results in the context of human hand motion tracking for task demonstrations. The analysis shows that for the threshold $k_r = 0.1$ rad the heading reset improves the accuracy in the presented experiment.

It has been observed that the heading reset can sometimes have a negative influence for other values of k_r. This problem has been analyzed in detail. The criterion based on the threshold k_r might be reconsidered for a more robust application of the heading reset.

The complete IMU orientation estimation algorithm is experimentally evaluated and it will be used in the next chapters with $k_r = 0.1$ rad. The next chapter addresses the estimation of the human hand position from the sensors orientation.

CHAPTER 4

INERTIAL HUMAN HAND MOTION TRACKING METHOD

Contents

This chapter presents the complete IHMT method developed for PbD applications. The objective is to express the human hand trajectory with respect to the robot base frame. The method gathers the record and embodiment mappings required to build a dataset from demonstrations in a PbD process. The method presented in Chapter 2 (see Algorithm 1) is also part of the record mapping.

4.1 Human hand trajectory

As explained in the introduction, the estimation of the human hand trajectory relies on the free segment approach. The free segment approach requires to estimate the absolute orientation of each segment of the human arm: the arm, the forearm and the hand. To do so, one sensor is fastened on each arm segment with an arbitrary position and orientation. The hand is considered as a single rigid segment, i.e., no finger motion is measured. The torso, as well as the rest of the human body, is considered motionless. Being aware that the human arm presents complex joints, the kinematic model of the human arm used in this work is composed of 3 spherical joints for the shoulder, elbow and wrist as described in Figure 4.1. Each arm segment is labeled with the numbers 1, 2 and 3 respectively for the arm, forearm and hand.

The human reference frame noted H is centered on the shoulder joint, its **z**-axis is along the gravity and pointing upwards and its **x**-axis, parallel to the sagittal plane, is defined later. Frames $1n$, $2n$ and $3n$ respectively denote the frames attached to the segments 1, 2 and 3 at time step n and centered at the joint. Their **x**-axes are aligned with the direction of the segment and pointing towards the next joint center in the chain. L_1, L_2 and L_3 are the segment lengths. The values for these parameters are directly measured on the operator. L_3 is defined as the distance from the finger tip D to the middle point between the radial styloid and the ulnar styloid. L_2 is defined as the distance from the middle point between the radial styloid and the ulnar styloid to the middle point between the lateral and the medial epicondyle. L_1 is defined as the distance from the middle point between the lateral and the medial epicondyle to the tip of the acromion bone.

The vector ${}^{H}\mathbf{AD}_n$ represents the position of the hand with respect to the frame H at time step n and can be evaluated from the orientations ${}^{H}\hat{q}_{1n}$, ${}^{H}\hat{q}_{2n}$ and ${}^{H}\hat{q}_{3n}$ of the segments 1, 2 and 3 as

$$ {}^{H}\hat{AD}_n = {}^{H}\hat{q}_{1n} \otimes {}^{1n}\hat{AB}_n \otimes {}^{1n}\hat{q}_H + {}^{H}\hat{q}_{2n} \otimes {}^{2n}\hat{BC}_n \otimes {}^{2n}\hat{q}_H + {}^{H}\hat{q}_{3n} \otimes {}^{3n}\hat{CD}_n \otimes {}^{3n}\hat{q}_H \tag{4.1} $$

with ${}^{1n}\mathbf{AB}_n = [L_1\ 0\ 0]^T$, ${}^{2n}\mathbf{BC}_n = [L_2\ 0\ 0]^T$, ${}^{3n}\mathbf{CD}_n = [L_3\ 0\ 0]^T\ \ \forall n$.

The frames attached to the sensors, at time step n, are noted $S1n$, $S2n$ and $S3n$, respectively for the segments 1, 2 and 3 (see Figure 4.1). The initial sensor frames are therefore respectively noted $S10$, $S20$, $S30$.

The rotation ${}^{H}\hat{q}_{in}$ of the segment i ($i = 1,2,3$) can be decomposed as

$$ {}^{H}\hat{q}_{in} = {}^{H}\hat{q}_{Si0} \otimes {}^{Si0}\hat{q}_{Sin} \otimes {}^{Sin}\hat{q}_{in} \tag{4.2} $$

Algorithm 1 described in Chapter 3 gives an estimation of the sensor orientation ${}^{Si0}\hat{q}_{Sin}$ based on IMU measurements. The other two orientations ${}^{H}\hat{q}_{Si0}$ and ${}^{Sin}\hat{q}_{in}$ are estimated through the

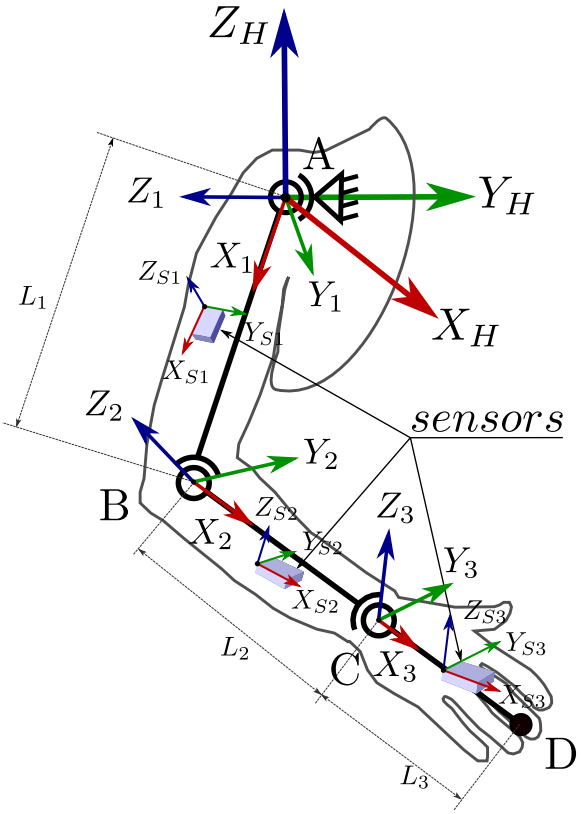

FIGURE 4.1: The human arm model and definition of the frames. H is the human attached frame centered on the shoulder. Frames 1, 2 and 3 are the human arm segments (arm, forearm, hand) attached frames. Frames $S1n$, $S2n$ et $S3n$ are the sensor frames.

initialization procedure described in the next section.

The human arm model relies on the assumptions listed below:

- the shoulder, elbow and wrist motions are pure rotation with respect to a fixed joint center,
- the sensors are rigidly attached to the segment, i.e., soft tissue displacements are neglected,
- the shoulder center remains in a fixed position during the trajectory (no torso displacement).
- the hand is considered as a rigid segment (no finger motion).

The model presents some deviations compared to the real human arm that could lead to disturbances on the hand trajectory estimation. Firstly, the human shoulder joint center depends on the arm position as explained, for instance, in [3]. The humerus rotates into the scapula when the arm is low, then the scapula itself observes a displacement for the higher position of the arm. The deviations are less important for the elbow and the wrist. Secondly, the sensors are not rigidly attached to the arm segments. The muscles and the skin as well as the clothes of the human can deform and induce a displacement of the sensor with respect to the arm segment. It might be possible to limit this effect by positioning the sensors on the arm where soft tissue displacements are reduced or to attach the sensors directly on the skin and not over the cloths. But the application of technical solutions limiting such soft tissue displacements has to be put in perspective with the targeted simplicity of PbD process. Thirdly, the teacher is asked to keep is torso motionless and his/her hand rigid. The non-respect of this instruction might introduce an error on the estimation of the hand position. These model deviations and parasitic motions are denoted *modelling errors* and contribute to the error on the hand trajectory.

4.2 Initialization procedure: the record mapping

The procedure described in this section estimates the two orientations ${}^{H}\hat{q}_{Si0}$ and ${}^{Sin}\hat{q}_{in}$. ${}^{Sin}\hat{q}_{in}$ is the I2S (inertial-to-segment) orientation, i.e., the orientation of the sensor with respect to the segment attached frame. The second orientation ${}^{H}\hat{q}_{Si0}$ is the initial sensor orientation with respect to the human reference frame.

The two orientations ${}^{H}\hat{q}_{Si0}$ and ${}^{Sin}\hat{q}_{in}$ as well as the sensor orientation from Algorithm 1 applied to the human model are necessary to transform the sensors data to a Cartesian position of the human hand. They constitute the record mapping which adapts the original dataset (angular rate and acceleration of the sensors) to encode a new information (hand trajectory). As the goal is to command the robot gripper trajectory, the task actions A_t encoded with the human hand trajectory should then be mapped to the robot actions A_l. The initialization procedure explained in this section is the method to identify the part of the record mapping: ${}^{H}\hat{q}_{Si0}$ and ${}^{Sin}\hat{q}_{in}$ (see Figure 4.2).

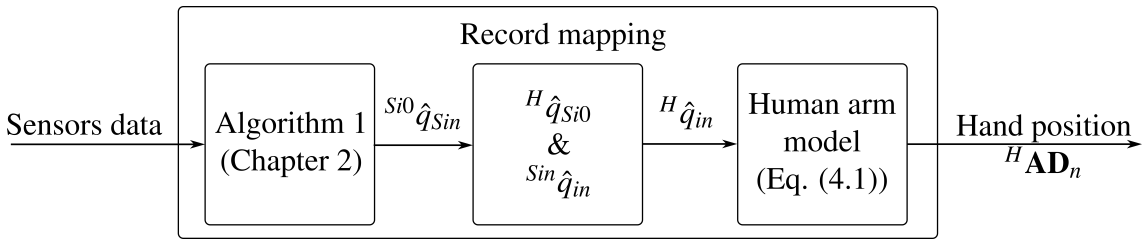

FIGURE 4.2: The elements of the record mapping.

4.2.1 Estimation of the I2S orientation

As said earlier, it is assumed that the motion of the skin and soft tissues can be neglected. The I2S (Inertial-to-Segment) orientation $^{Sin}\hat{q}_{in}$ is therefore constant, i.e.,

$$^{Sin}\hat{q}_{in} = {}^{Si0}\hat{q}_{i0} \ \forall n. \tag{4.3}$$

Furthermore, it is asked to the operator to start the trajectory with the arm horizontal, outstretched and in the sagittal plane as illustrated in Figure 4.3 a. This initial arm configuration is such that each segment is oriented ahead of the teacher. This orientation will serve as reference for the activation of the heading reset (see chapter 2). It is recalled that the application of the heading reset is based on the heading of each sensor independently. Therefore, it is not applied only when the operator arm returns to the initial configuration, but also in a diversity of other configurations. In this initial arm configuration, the segment frame $i0$ is defined as aligned with the inertial human reference frame H, i.e.,

$$^{Si0}\hat{q}_{i0} = {}^{Si0}\hat{q}_{H}. \tag{4.4}$$

Therefore, the I2S orientation and the initial sensor orientation are equal:

$$^{Sin}\hat{q}_{in} = {}^{Si0}\hat{q}_{H} \ \forall n. \tag{4.5}$$

The problem is then reduced to identifying only one orientation for each sensor: $^{Si0}\hat{q}_{H}$. Estimating it boils down to measuring two axes of the human frame H with respect to the sensor frames. The next section describes the procedure.

4.2.2 Estimating the initial sensor orientation

The **z**-axis of the reference frame H being along the gravity direction, its orientation with respect to the sensor frame is estimated through accelerometer measurement when the operator is in the initial position (see Figure 4.3 a):

$$^{Si0}\mathbf{z}_{H} = -\frac{\mathbf{a}}{\|\mathbf{a}\|} \tag{4.6}$$

Previously, the operator receives the instruction to perform a shoulder flexion in the sagittal plane (see Figure 4.3 b), and the **y**-axis of the reference frame H is then defined as the normalized

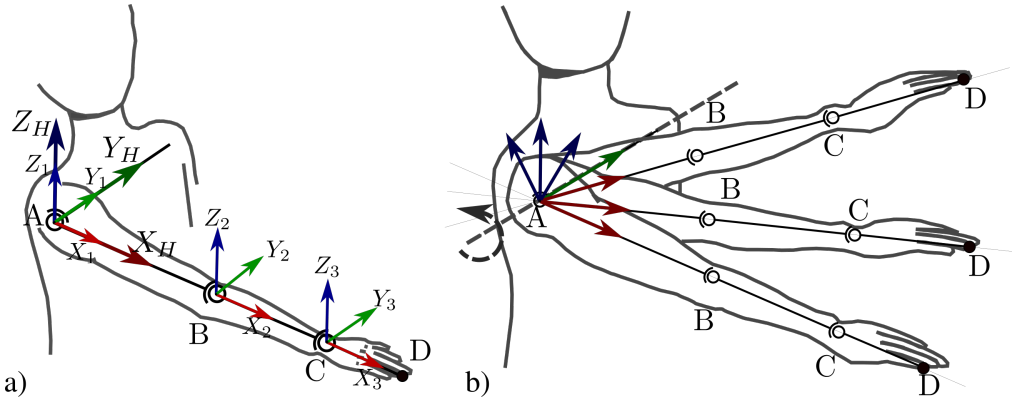

FIGURE 4.3: Initialization procedure: a) initial position ; b) rotation around shoulder-shoulder axis.

projection of the angular velocity in the plane perpendicular to the **z**-axis of H. In this way, the rotation ${}^{Si0}\hat{q}_H$ is obtained for each sensor i and consequently, the I2S orientation is obtained too. Table 4.1 details the initialization procedure for segment i.

1	Rotation around the shoulder-shoulder axis with a straight arm	N recording of $\boldsymbol{\omega}_k$
2	Computation of $\mathbf{y}_\omega$	$\mathbf{y}_\omega = \frac{1}{N}\sum_{k=1}^{N}\frac{\boldsymbol{\omega}_k}{\|\boldsymbol{\omega}_k\|}$
3	Arm locked in initial configuration	N recording of $\mathbf{a}_k$
4	Computation of ${}^{Si0}\mathbf{z}_H$	${}^{Si0}\mathbf{z}_H = -\frac{1}{N}\sum_{k=1}^{N}\frac{\mathbf{a}_k}{\|\mathbf{a}_k\|}$
5	Projection of $\mathbf{y}_\omega$	$\mathbf{y}'_\omega = -(\mathbf{y}_\omega - (\mathbf{y}_\omega \cdot {}^{Si0}\mathbf{z}_H){}^{Si0}\mathbf{z}_H)$
6	Computation of ${}^{Si0}\mathbf{y}_H$	${}^{Si0}\mathbf{y}_H = \frac{\mathbf{y}'_\omega}{\|\mathbf{y}'_\omega\|}$
7	Computation of ${}^{Si0}\mathbf{x}_H$	${}^{Si0}\mathbf{x}_H = {}^{Si0}\mathbf{y}_H \times {}^{Si0}\mathbf{z}_H$
8	Computation of rotation matrix	${}^{Si0}R_H = [{}^{Si0}\mathbf{x}_H\, {}^{Si0}\mathbf{y}_H\, {}^{Si0}\mathbf{z}_H]$
9	Conversion to quaternion	${}^{Si0}R_H \rightarrow {}^{Si0}\hat{q}_H$
10	Computation of ${}^{H}\hat{q}_{Si0}$ and ${}^{Sin}\hat{q}_{in}$	${}^{H}\hat{q}_{Si0} = ({}^{Si0}\hat{q}_H)^{-1}$; ${}^{Sin}\hat{q}_{in} = {}^{Si0}\hat{q}_H$

TABLE 4.1: Steps of the initialization of the human arm measurement process.

4.3 Human-robot transformation: the embodiment mapping

For the actions encoded in the human hand trajectory to be mapped to the robotic system, the hand trajectory has to be expressed with respect to the robot reference frame R. An embodiment mapping is then required and it consists in the transformation from H to R (see Figure 4.4). Identifying the embodiment mapping is estimating the transformation from the human inertial reference frame H with origin A to the robot based frame R with origin O, composed of a rotation part ${}^{R}\hat{q}_H$ and a translation part ${}^{R}\mathbf{AO}$.

Once the initialization procedure, described in the previous section, is finished, the human hand position with respect to H is tracked. The operator is asked to put her/his fingertip in contact

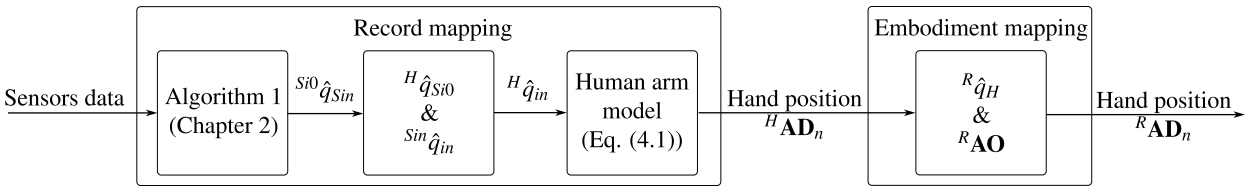

FIGURE 4.4: The record and embodiment mappings.

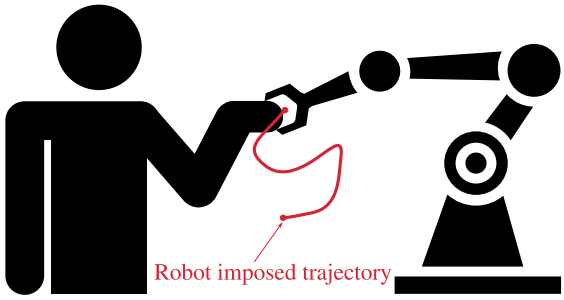

FIGURE 4.5: Illustration of the human-robot transformation procedure: the teacher follows a trajectory imposed by the robot.

with the robot end-effector and follows an arbitrary trajectory imposed by the robot as illustrated in Figure 4.5. During the trajectory, N human fingertip positions $^{H}\mathbf{AD}_k$ and robot end-effector positions $^{R}\mathbf{OP}_k$ are recorded. The rotation $^{R}\hat{q}_H$ is estimated using a least square method [9]. The first step is to compute the centroid of each data set $^{H}\mathbf{AD}$ and $^{R}\mathbf{OP}$ as

$$^{H}\mathbf{AD} = \frac{1}{N}\sum_{k=1}^{N} {}^{H}\mathbf{AD}_k \tag{4.7}$$

$$^{R}\mathbf{OP} = \frac{1}{N}\sum_{k=1}^{N} {}^{R}\mathbf{OP}_k. \tag{4.8}$$

The rotational matrix Q corresponding to $^{R}\hat{q}_H$ is estimated using a singular value decomposition of the matrix M

$$M = \sum_{k=1}^{N} ({}^{H}\mathbf{AD}_k - {}^{H}\mathbf{AD})({}^{R}\mathbf{OP}_k - {}^{R}\mathbf{OP})^T = USV^T \tag{4.9}$$

$$Q = VU^T. \tag{4.10}$$

The matrix Q is then converted into the unit quaternion $^{R}\hat{q}_H$. The translational part $^{R}\mathbf{AO}$ is obtained as

$$^{R}\mathbf{AO} = \frac{1}{N}\sum_{k=1}^{N} ({}^{R}\mathbf{OP}_k - Q\ {}^{H}\mathbf{AD}_k)\ . \tag{4.11}$$

4.4 Optimization of the parameters

The initialization procedure can suffer from parasitic motions as disturbances in rotation of the arm or the arm not completely outstretched as well as the human-robot transformation procedure if, for instance, the contact point between the human and the robot is not permanent. Such parasitic motions are denoted *calibration errors*. The *calibration errors*, with the *modelling errors* (see Section 4.1), lead to uncertainties in the parameters ${}^{H}\hat{q}_{Si0}$ and ${}^{Si}\hat{q}_i$ with $i \in [1,2,3]$ as well as in the parameters ${}^{R}\hat{q}_H$ and ${}^{R}\mathbf{AO}$. Furthermore, imperfect measurements of the segment length which induces some uncertainties in L_1, L_2 and L_3, also contributes to the error on the hand trajectory.

In order to partially reduce the error on the above-mentioned parameters and consequently on the hand trajectory, an optimization step is conducted. The optimization process takes advantage of the PbD context by using data recorded during the human-robot transformation estimation procedure. No extra step is introduced which maintains a sufficient easiness-of-use required by the context of PbD. The robot end-effector positions ${}^{R}\mathbf{OP}_k$ measured through robot encoders are used as reference. The human hand position with respect to the robot frame is computed from the estimation of the sensors orientation ${}^{S10}\hat{q}_{S1k}$, ${}^{S20}\hat{q}_{S2k}$ and ${}^{S30}\hat{q}_{S3k}$ recorded simultaneously with the robot end-effector positions. The optimization is made by minimizing the following error function f:

$$\begin{aligned} f({}^{1n}\mathbf{AB}, {}^{2n}\mathbf{BC}, {}^{3n}\mathbf{CD}, {}^{S1}\hat{q}_1, {}^{S2}\hat{q}_2, {}^{S3}\hat{q}_3, {}^{H}\hat{q}_{S10}, {}^{H}\hat{q}_{S20}, {}^{H}\hat{q}_{S30}, {}^{H}\hat{q}_R, {}^{R}\mathbf{AO}) \\ = \sum_{k=1}^{N} \| {}^{R}\hat{O}P_k - ({}^{R}\hat{q}_H \otimes {}^{H}\hat{AD}_k^* \otimes {}^{H}\hat{q}_R - {}^{R}\hat{AO}) \|^2 \end{aligned} \tag{4.12}$$

with

$${}^{H}\hat{AD}_k^* = {}^{H}\hat{q}_{1k}^* \otimes {}^{1k}\hat{AB} \otimes {}^{1k}\hat{q}_H^* + {}^{H}\hat{q}_{2k}^* \otimes {}^{2k}\hat{BC} \otimes {}^{2k}\hat{q}_H^* + {}^{H}\hat{q}_{3k}^* \otimes {}^{3k}\hat{CD} \otimes {}^{3k}\hat{q}_H^* \tag{4.13}$$

$${}^{H}\hat{q}_{1k}^* = {}^{H}\hat{q}_{S10} \otimes {}^{S10}\hat{q}_{S1k} \otimes {}^{S1}\hat{q}_1 \tag{4.14}$$

$${}^{H}\hat{q}_{2k}^* = {}^{H}\hat{q}_{S20} \otimes {}^{S20}\hat{q}_{S2k} \otimes {}^{S2}\hat{q}_2 \tag{4.15}$$

$${}^{H}\hat{q}_{3k}^* = {}^{H}\hat{q}_{S30} \otimes {}^{S30}\hat{q}_{S3k} \otimes {}^{S3}\hat{q}_3. \tag{4.16}$$

The optimization problem is solved using the *optimize.minimize* function from the SciPy library with the BFGS algorithm and a tolerance set to 10. The initial guesses for the segment lengths L_1, L_2, L_3 involved in ${}^{1n}\mathbf{AB}$, ${}^{2n}\mathbf{BC}$ and ${}^{3n}\mathbf{CD}$ are from a direct measurement on the operator arm. The initial values of the I2S orientation ${}^{S1}\hat{q}_1$, ${}^{S2}\hat{q}_2$ and ${}^{S3}\hat{q}_3$ come from the initialization as well as the orientations ${}^{S10}\hat{q}_{S1k}$, ${}^{S20}\hat{q}_{S2k}$ and ${}^{S30}\hat{q}_{S3k}$. Finally, ${}^{H}\hat{q}_R$ and ${}^{R}\mathbf{AO}$ initial guesses are the values from the human-robot transformation estimation process described in the previous section.

4.5 Summary

This chapter exposes the record (partially) and the embodiment mappings required to track the human hand with respect to the robot base frame, as well as the method to estimate them.

The record mapping includes the sensor orientation from Algorithm 1 (Chapter 2) and two constant orientations: the orientation from the sensor initial frame to the human reference frame and the orientation from the sensor frame to the human segment frame. These orientations are identified based on the initialization procedure.

The embodiment mapping is the transformation between the human reference frame and the robot base frame. During the procedure to estimate this mapping, the robot imposes a trajectory to the human and both agents trajectories are recorded simultaneously. A least square method is applied on the two trajectories to estimate the transformation from the human frame to the robot frame.

The IHMT method is affected by different sources of errors. Firstly, the sensor orientation errors evaluated in Chapter 3 have a direct influence on the estimation of the hand position. In addition, some *modelling errors* and *calibration errors* have been identified. They introduce uncertainties in some model parameters and degrade the hand trajectory. To limit the impact of these uncertainties, a parameters optimization is performed. This extra step is based on data from the human-robot transformation estimation procedure, thus, the method keeps its simplicity.

The identification of the mappings requires a protocol that the teacher should observe before performing the task. This protocol can be summarized as follows. The teacher stands in front of the task workspace. He/she then performs a rotation of his/her arm straight while the gyroscope data are recorded and complete his/her motion in the initial arm configuration (see Figure 4.3). The teacher remains in that position a short moment while the accelerometer data is recorded. The tracking of the human hand with respect to the human reference frame H starts and the teacher enters in contact with the robot and follows the imposed trajectories (both hand and gripper trajectories are recorded). Once the robot has completed this trajectory, the model parameters are optimized, the human hand is tracked with respect to the robot frame R and the teacher can begin the demonstration.

In the next chapter, several experiments are conducted in order to evaluate each step of the mappings identification, to assess the validity of the assumptions and to quantify the accuracy of the human hand trajectory estimated with the proposed IHMT method.

CHAPTER 5

EXPERIMENTAL TESTS ON THE IHMT METHOD

Contents

The experiments presented in this chapter are intended to evaluate the influence of the sensor orientation errors, the modelling errors (torso and shoulder displacements, soft tissues displacements, etc.) and the calibration errors (the deviation between the actual and the recommended motions in the initialization phases) of the IHMT method proposed in the previous chapter.

In order to evaluate the initialization procedure, the experiment is first conducted on a robot arm which is considered performing exactly the recommended motions. In this case, the initialization procedure and the trajectory are not affected by the modelling errors nor the calibration errors. The ability of the initialization procedure to manage unknown sensors positioning is evaluated.

Secondly, an experiment is conducted with a human operator. The objective is to quantify the error on the human hand trajectory with respect to the human reference frame when estimated by the proposed IHMT method. Compared to the error measured in the previous experiment, the contribution to the hand trajectory error of the calibration errors, as well as the modelling errors can be identified. This experiment allows us to assess the validity of the assumptions made for the initialization (see Section 4.1).

Finally, the hand trajectory accuracy with respect to the robot base frame is evaluated. Compared to the previous experiment, one can evaluate the error due to the human-robot transformation estimation process as well as the impact of the optimization of the parameters.

5.1 Initialization procedure assessment

This section presents the experiment conducted to evaluate the robustness of the initialization procedure which was proposed to estimate the positioning of the sensors on each segment. This experiment consists in tracking the motion of a robotic arm as in [42].

5.1.1 Experimental setup

Three Xsens MTw sensors have been mounted on an ABB IRB 120 robotic arm as shown in Figure 5.1. The support (right side of Figure 5.1) allows the sensors to be set in different configurations (heading angle and inclination angle) with respect to the segment frame. The robot has an anthropomorphic kinematics with one degree of freedom less than the human arm. Axes 1 and 2 of the robot mimic the human shoulder with the center of the joint at the intersection of the two axes. The human elbow is materialized by axis 3. Axes 4, 5 and 6 mimic the human wrist. So, the rotation of the human arm segment around the axis of the segment itself (3$^{\text{rd}}$ degree of freedom of the shoulder) has no equivalent in the robot kinematics. However, the human arm model used in this work, composed of 3 spherical joints, is also applicable to reconstruct the robot motions, so that our method can be applied to the robot directly. The lengths of the robot segments are taken from the robot data sheet: ${}^{1n}\mathbf{AB}_n = [270\ 0\ 0]^T$mm, ${}^{2n}\mathbf{BC}_n = [310.0065\ 0\ 0]^T$mm and ${}^{3n}\mathbf{CD}_n = [72\ 0\ 0]$mm $\forall n$.

The trajectory programmed on the robot has been arbitrarily defined with the intention to cover as much as possible the human hand reachable space which can be described, e.g., in [69].

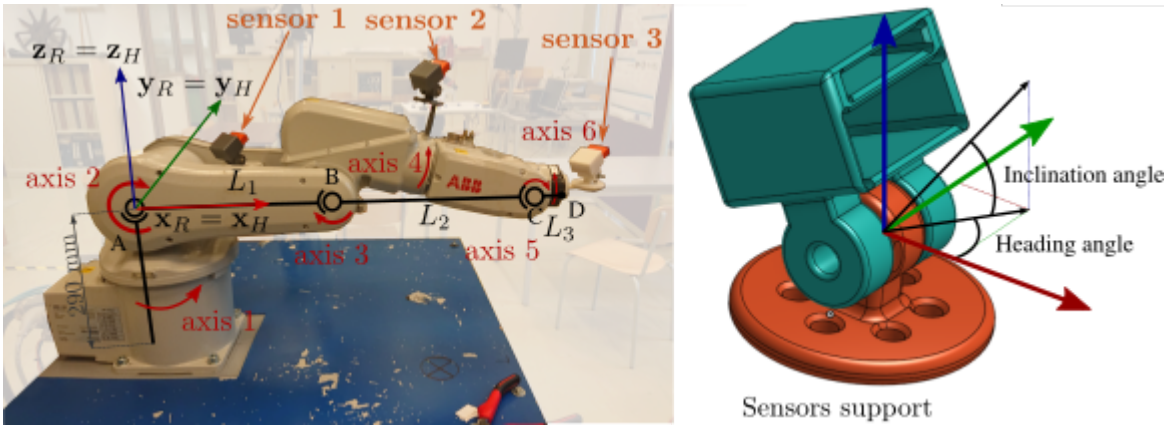

FIGURE 5.1: Experimental setup for the measurement of the error on the hand trajectory for the proposed IHMT method. The left figure shows the correspondence between the 6 robot axis and the human arm model and the location of the three IMUs sensors on the robot's segments.

Besides the human reachable space, no extra care was taken to make the trajectory humanly realistic. The evaluation of the method for human motion is made in the next experiment. The trajectory executed by the robot is shown in Figure 5.2. To build the trajectory, some positions have been set to make a lower and an upper "infinite symbol" shape trajectory. It is played first continuously and then with breaks of 1, 2 or 3 seconds at each position. In this second phase, the trajectory shows angular points at these positions. Furthermore, at some moments, the robot returns to its initial position to ensure that the heading reset is applied. The heading reset can also be applied at different positions along the trajectory if the configuration of the robot allows it. The complete described movement is executed twice, first at a maximum end-effector linear velocity of 400 mm/s and a maximum joint angular velocity at 400 °/s and secondly at a maximum velocity of 50 mm/s and 50 °/s. The trajectory lasts 6 minutes and 7 seconds. This trajectory is measured 19 times with different configurations of the sensors on the robot. The 19 configurations do not cover all possible ways to set up the sensors on the arm, but they represent a reasonable sampling for our purpose.

The trajectory of the end-effector ${}^{R}\mathbf{OP}_k$ measured by the robot through its encoders is used as reference. The absolute error on the hand position is computed as

$$\varepsilon_t = \frac{1}{N}\sum_{k=1}^{N} \|{}^{H}\mathbf{AD}_k - {}^{R}\mathbf{OP}_k\| \tag{5.1}$$

with N the number of points and ${}^{H}\mathbf{AD}$ the trajectory measured from inertial sensors and ${}^{R}\mathbf{OP}$ is the same trajectory recorded through robot encoders. The robot reference frame R is simply the base frame shifted of 290 mm along its $\mathbf{z}$-axis (direction of axis 1 according to the robot data sheet). The frame H is the reference frame for the inertial-based trajectory and is defined such that it coincides with R, which implies that the points O and A are at the same position.

The IHMT method exposed in the previous chapter (Chapter 4) is not applied in its entirety. Since only one agent is involved in this experiment, no human-robot transformation estimation procedure is conducted. The consequence is that the parameters optimization step (see Section

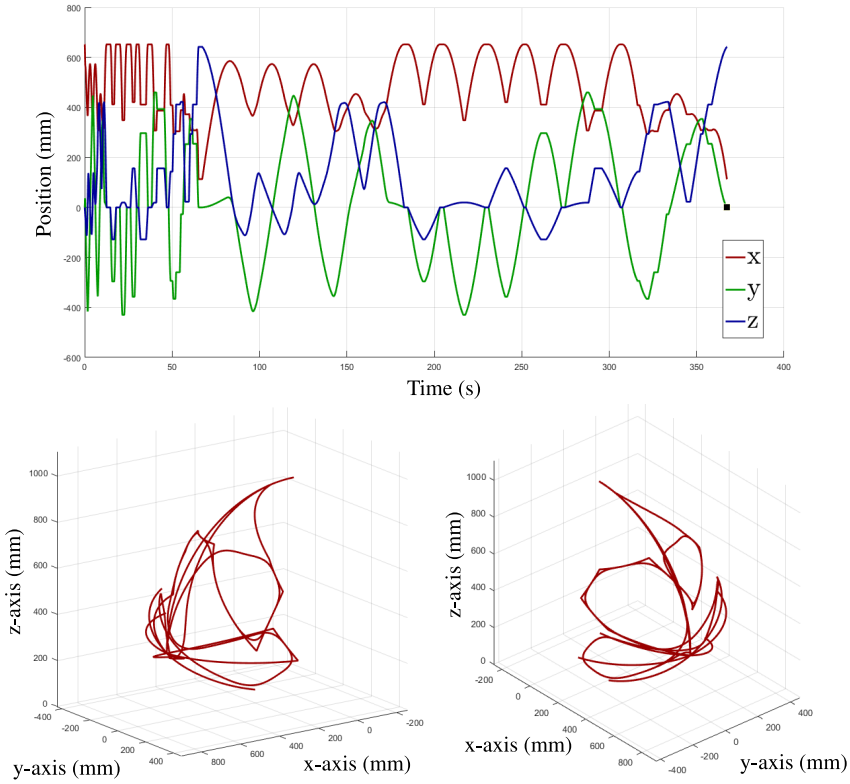

FIGURE 5.2: The 3D trajectory imposed to the robot and measured using robot encoders for reference.

4.4) cannot be performed. It can be recalled that the optimization step aims at catching uncertainties in some model parameters. However, by design, the robot joint motions are accurately captured by the model. Furthermore, the sensors are rigidly attached to the robot; the base of the robot (which mimics the human torso) does not move and the robot "hand" is rigid (no finger motion). The motions executed by the robot in the initialization procedure are considered undisturbed. Therefore, the parameters optimization is not relevant in the context of this experiment.

5.1.2 Results and discussion

The results are presented in Table 5.1. For each configuration, the mean error ε and the standard deviation σ are given. The mean error on the hand position is comprised between 27.7 mm and 47.7 mm. The 15^{th} measurement seems aberrant and is discarded. Figure 5.3 shows the measured trajectory for the best (N° 7) and the worse (N° 13) sensor configurations. Let us show that this error is mainly caused by the sensor orientation errors.

An expected error on the hand position is estimated for that purpose. It is defined as the mean error on the hand position that is expected if the only source of error is the error on sensor orientation. The sensor orientation error has been previously measured (see Section 3.4.1).

Three measurements $m = [1,2,3]$ of the same trajectory have been performed. For a given sensor i and a given measurement m, at time step k, the reference orientation obtained from

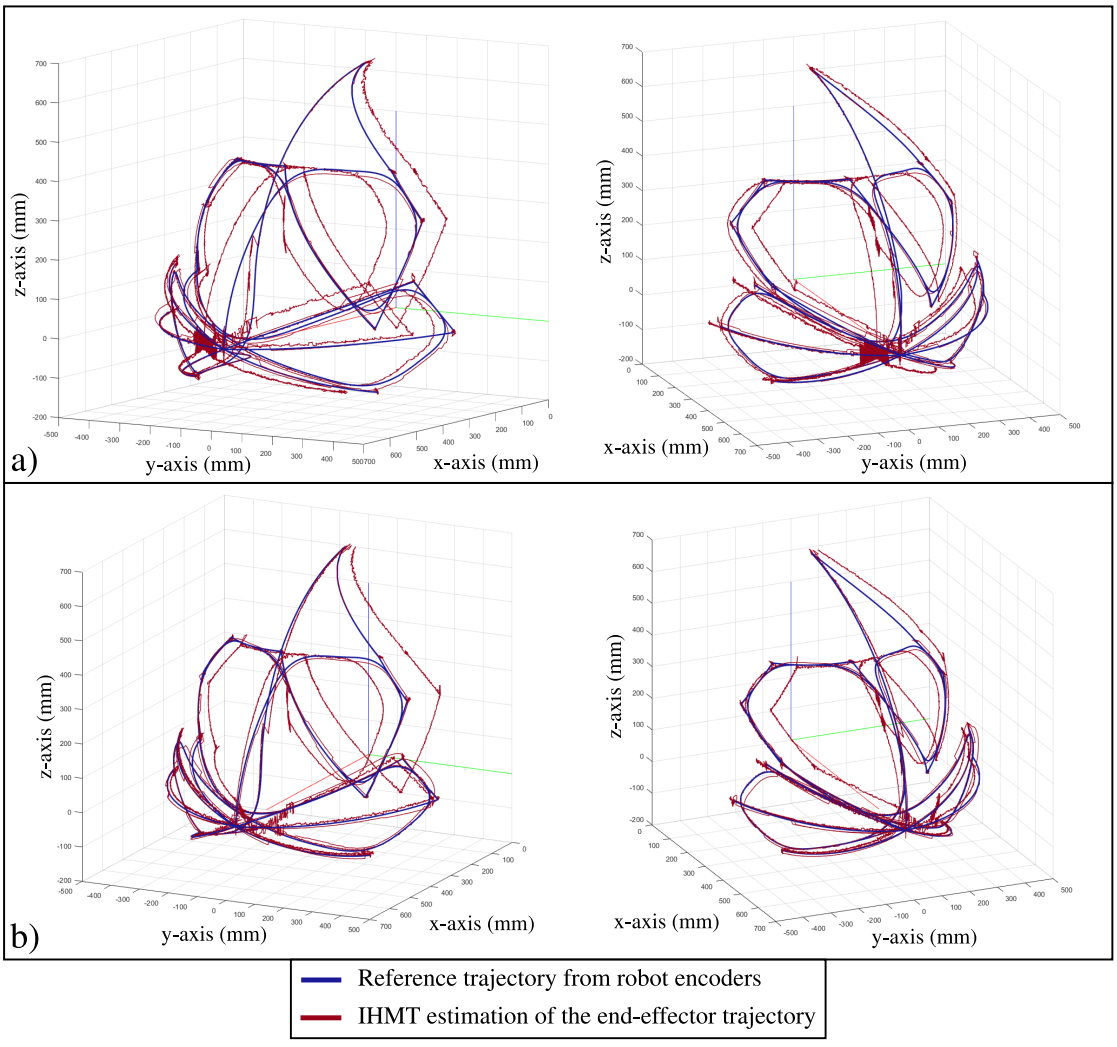

FIGURE 5.3: The trajectory reference from robot encoders and the estimated trajectory from the IHMT method for configuration 13 a) and for configuration 7 b).

Sensors configuration N°	ε (mm)	σ (mm)
1	42.2	30.1
2	41.7	31.3
3	40.2	30.8
4	41.4	30.9
5	41.8	32.5
6	38.2	30.7
7	27.7	31.0
8	32.0	35.9
9	29.3	33.3
10	44.1	32.8
11	40.8	29.9
12	42.2	29.3
13	47.7	33.3
14	44.5	30.8
15	157.1	61.2
16	44.5	33.8
17	43.9	32.3
18	42.3	31.8
19	40.0	30.1

TABLE 5.1: Mean error and standard deviation of the error on the end-effector position estimated from the proposed IHMT method for different configurations of the sensors on the robotic arm.

the robot encoders is $\hat{q}_{i,m,k}^{robot}$, the estimated orientation based on Algorithm 1 is $q_{i,m,k}^{IMU}$ and the quaternion representation $\hat{q}_{i,m,k}^{err}$ of the error on the sensor orientation estimation is

$$\hat{q}_{i,m,k}^{err} = \hat{q}_{i,m,k}^{IMU} \otimes (\hat{q}_{i,m,k}^{robot})^{-1} \tag{5.2}$$

with $k \in [1;N]$ and N the number of time steps in a measurement. The error $\hat{q}_{i,m,k}^{err}$ can be expressed as a rotation vector as

$$\boldsymbol{\theta}_{i,m,k}^{err} = \log(\hat{q}_{i,m,k}^{err}) \tag{5.3}$$

with the logarithm map log defined in Section 2.2. The mean error rotation vector $\bar{\boldsymbol{\theta}}_{i,m}^{err}$ is computed as

$$\bar{\boldsymbol{\theta}}_{i,m}^{err} = \frac{1}{N}\sum_{k=1}^{N} \boldsymbol{\theta}_{i,m,k}^{err} \tag{5.4}$$

The values of $\bar{\boldsymbol{\theta}}_{i,m}^{err}$ are presented in Table 5.2. Finally, for a sensor i, the mean error rotation vector is computed as

$$\bar{\boldsymbol{\theta}}_{i}^{err} = \frac{1}{3}\sum_{m=1}^{3} \bar{\boldsymbol{\theta}}_{i,m}^{err} \tag{5.5}$$

Measurement m	Sensors i	$\bar{\boldsymbol{\theta}}_{i,m}^{err}$ (deg)		
		x	y	z
1	1	0.77	0.70	1.67
	2	0.77	0.75	13.33
	3	0.75	0.66	1.35
2	1	0.56	1.44	1.41
	2	1.30	1.57	1.25
	3	2.00	1.49	2.50
3	1	0.58	0.47	1.60
	2	0.68	0.54	1.29
	3	0.61	0.48	1.09

TABLE 5.2: Mean error on the sensor orientation from Algorithm 1.

For the three sensors, the errors $\bar{\boldsymbol{\theta}}_1^{err}$, $\bar{\boldsymbol{\theta}}_2^{err}$ and $\bar{\boldsymbol{\theta}}_3^{err}$ are propagated in the model (see Equation 4.1, Chapter 4). Let us consider the arm model in a given configuration corresponding to the true segment orientation $\hat{q}_1$, $\hat{q}_2$, $\hat{q}_3$. Each segment orientation $\hat{q}_i$ can be affected by their respected sensor mean error $\bar{\boldsymbol{\theta}}_i^{err}$ as $\hat{q}_i \otimes \exp(\bar{\boldsymbol{\theta}}_i^{err})$. In the initial configuration $\hat{q}_1 = \hat{q}_2 = \hat{q}_3 = [1\,0\,0\,0]^T$, the expected error on the hand position is computed using Eq. 4.1 and is 39.4 mm.

The reliability of the initialization procedure to the sensor positioning can be evaluated by comparing this expected error to the error values exposed in Table 5.1. The error estimated experimentally is between 27.7 mm and 47.7 mm and the mean error over all measurements is 40.25 mm. It can be concluded that the observed errors in Table 5.1 are mostly attributed to the errors on the sensor orientation in this experiment. This confirms our assumptions that the modelling errors and the calibration errors have a limited impact. This also shows that the proposed IHMT method is robust to the unknown positioning of the sensors thanks to the initialization procedure.

5.2 IHMT method assessment on human motion

In this section, the method is implemented to track the human hand. The trajectory of the human hand is compared to the trajectory measured by an opto-electronic motion capture system with active markers (Codamotion, Charnwood Dynamics Ltd). The Codamotion system presents an error under 5 mm on the 3D positioning of a marker according to Schwartz et al. [115]. Compared to the previous results, the results of this experiment emphasize the influence of the modelling errors and the calibration errors.

5.2.1 Experimental setup

The markers of the Codamotion system emit infrared light illustrated in Figure 5.4. This light is perceived by four units. A unit is composed of three linear optical sensors. The geometric relation between these three sensors is known, so each unit can, by triangulation, estimate the 3D position of a marker. Four units are used to avoid occlusion and the transformation between each unit and the Codamotion reference frame C is known by a calibration procedure. The Codamotion reference frame C is built from three active markers set on the floor during the calibration phase. One marker is the origin O and the two others point the **x**-axis and the **y**-axis.

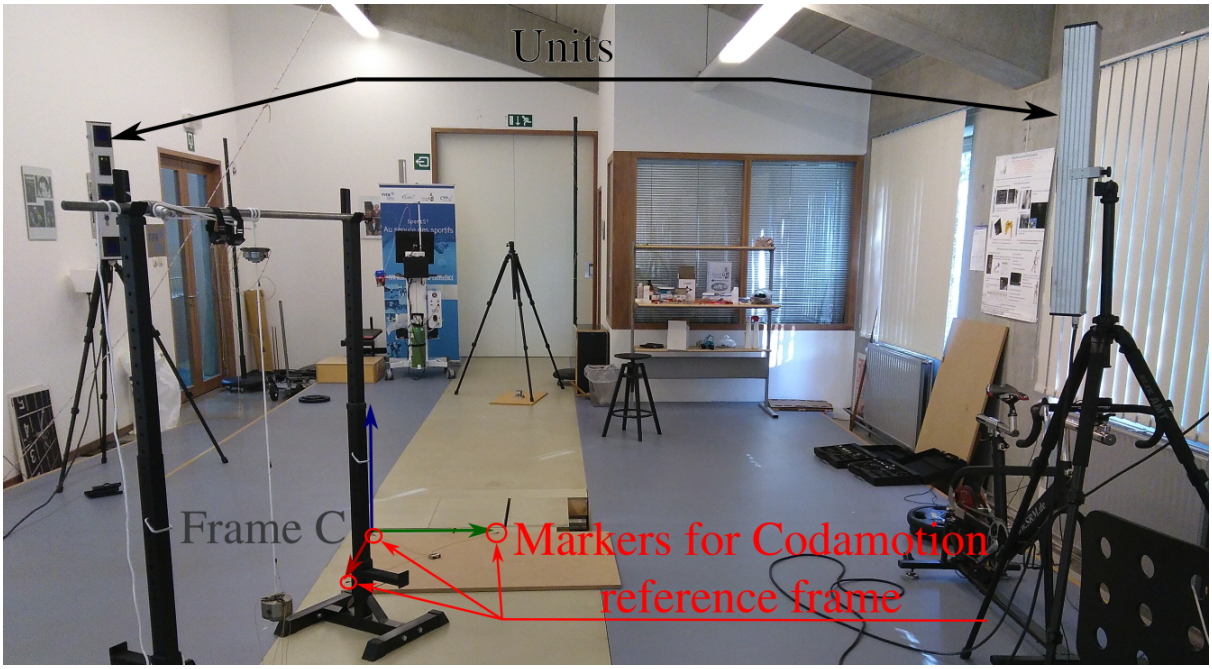

FIGURE 5.4: The Codamotion setup.

The **z**-axis of C is built from the **x**-axis and the **y**-axis and is therefore perpendicular to the ground and considered aligned with the gravity.

In order to limit fingers disturbing motions and to have a precise point to track, the subject holds a gun-shape tool (see Figure 5.5). The IMU related to the hand is fixed in this tool. The tip of the gun-shape tool is the point of interest to track. Two Codamotion markers are set on the gun-shape tool tip to limit occlusion issues during motion. The point of interest is the mid-point between the two markers when both markers are visible or just the visible one otherwise. Two other markers are set on the top of the gun-shape tool. Four markers are set on the bony landmarks of the wrist and elbow (the radial styloid, the ulnar styloid, the lateral and the medial epicondyle). Finally, markers are set on the end of the acromion, the incisura jugularis (IJ) and the seventh vertebra (C7), see Figure 5.5. The segment lengths for the IHMT method are measured directly on the subject: ${}^{1n}\mathbf{AB}_n = [330\ 0\ 0]^T$ mm, ${}^{2n}\mathbf{BC}_n = [280\ 0\ 0]^T$ mm and ${}^{3n}\mathbf{CD}_n = [270\ -20\ 100]$ mm $\forall n$.

In order to compare the estimation of the hand trajectory from the IHMT method with the measurements from the Codamotion system, the transformation between the two reference frames H and C has to be known. The **z**-axis of the human frame being, by definition, aligned with the gravity direction, it is aligned with the **z**-axis of the Codamotion frame C. To identify the **y**-axis of H with respect to C, the trajectories of the hand, the wrist and the elbow markers during the rotation movement of the initialization procedure are studied. A circle is fitted on each of them. The mean of the normal to the circles projected onto the plane perpendicular to the **z**-axis, constitutes the **y**-axis of the human reference frame H with respect to C. The **x**-axis is computed from the cross product of the **y** and **z**-axes. It is not possible to directly measure the origin of the human reference frame H with respect to the Codamotion system. In a similar experience, Filippeschi et. al [49], estimate the center of the shoulder using markers on bony landmarks: “the

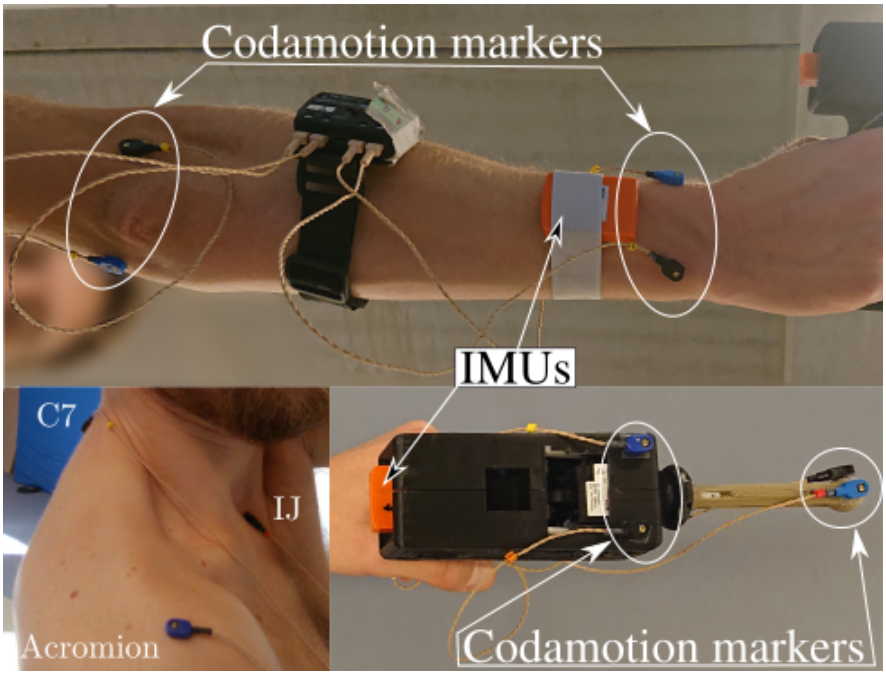

FIGURE 5.5: Markers of the Codamotion on the human arm.

marker on the acromion served to capture the shoulder joint center, which is assumed to be 4.5 cm under the acromion in the Zv direction". However, Amabile et al. show in [3] that the center of rotation of the shoulder is not isolated. Therefore, we propose to define the origin of H as the mean point of all the centers of the circles fitting the rotation motion of the markers on the arm during the rotation part of the initialization procedure. The transformation between H and C can then be estimated. It is composed of a rotation part ${}^{H}R_{C}$ and a translation part ${}^{H}\mathbf{AO}$. Figure 5.6 shows the markers trajectory, the fitting circles and the frames H and C. This transformation is estimated before each measurement.

Several trajectories are executed by a unique operator with different velocities and occasional breaks during each trajectory. Inspired from [96], the operator has the instruction to perform circular trajectories, 3D cross-shaped trajectories, cube edges or hexagonal trajectories. These trajectories imply rotations and translations in all directions of space. The error is evaluated as the distance between the estimated position ${}^{H}\mathbf{AD}_{k}$ of the gun tip by the proposed method and the reference one ${}^{C}\mathbf{OP}_{k}$ measured by the Codamotion system as

$$ {}^{H}\mathbf{AP}_{k} = {}^{H}R_{C}\,{}^{C}\mathbf{OP}_{k} + {}^{H}\mathbf{AO} \tag{5.6} $$

$$ \varepsilon_{t} = \frac{1}{N}\sum_{k=1}^{N} \|{}^{H}\mathbf{AD}_{k} - {}^{H}\mathbf{AP}_{k}\| \tag{5.7} $$

with N the number of points.

5.2.2 Results

Some examples of the trajectories followed during this experiment are shown in Figure 5.7. The results of the experiments are presented in Tables 5.3 and 5.4.

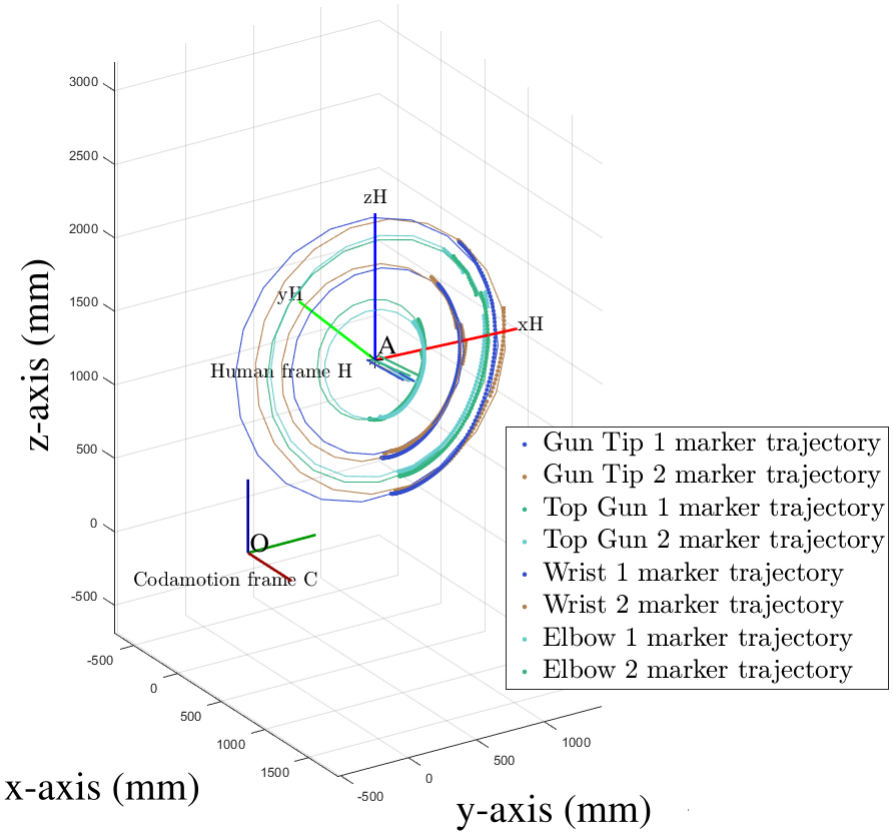

FIGURE 5.6: 3D representation of the human attached frame H and the reference Codamotion frame C. The Figure shows the positions of the Codamotion markers attached to the human arm when performing the initialization procedure and the fitting circles used to build the human reference H frame with respect to the Codamotion frame C.

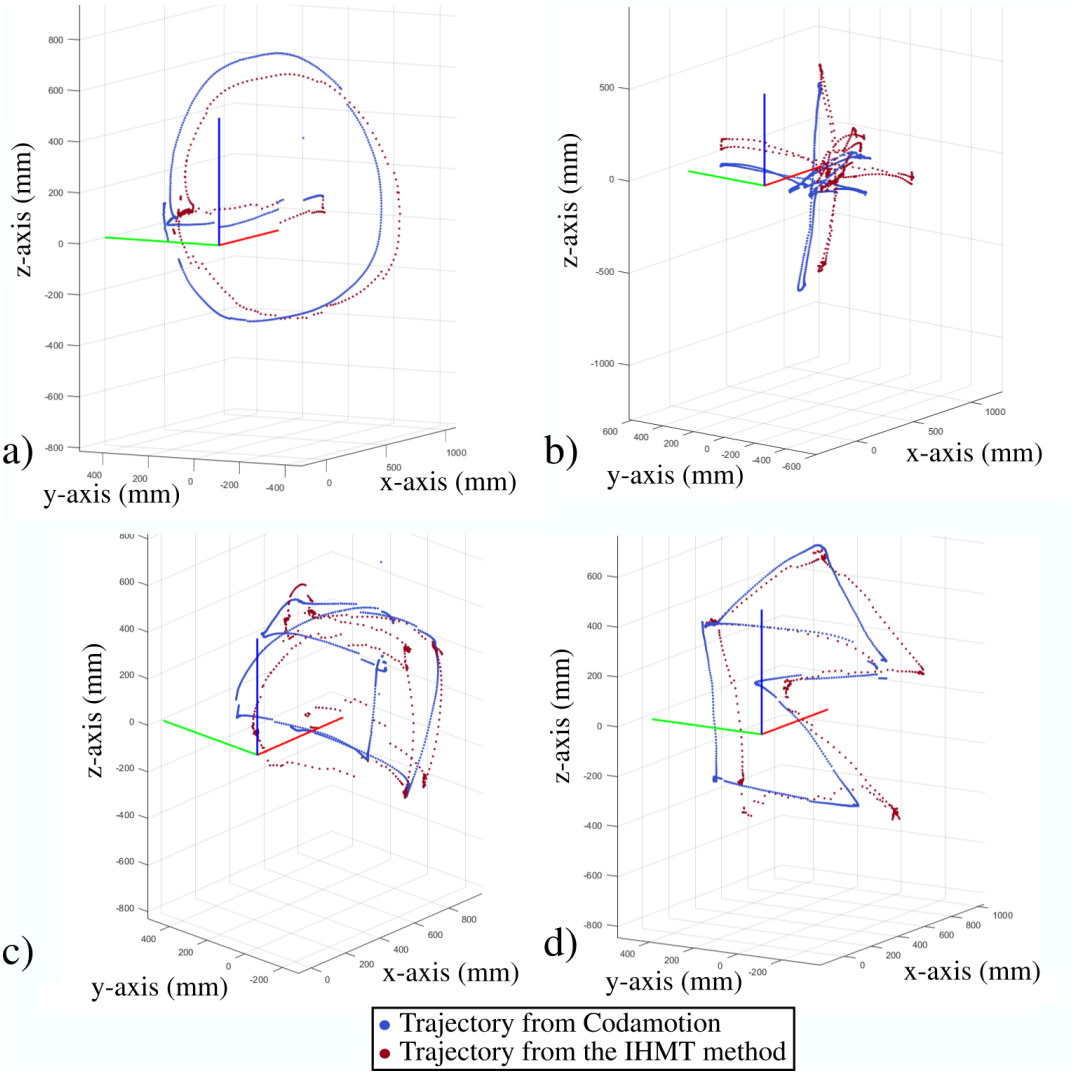

FIGURE 5.7: Examples of the different trajectories performed by the human operator: a) Circular shape trajectory; b) cross-shape trajectory; c) cube edges trajectory; d) hexagonal trajectory.

The errors for all trajectories are shown in Table 5.3. In this table, three indicators, defined in the list below, are used to quantify some contributions of the modelling errors. One indicator is used to evaluate the impact of the calibration errors. They are defined in the following list and their values are presented in Table 5.4. Moreover, some indicators are presented to quantify the quality of the transformation between the human frame H and the Codamotion frame C which is a sensitive point of this experiment.

- The **torso parasitic motion** is quantified as **displacement** and **orientation**. The position of the torso is calculated as the mid-point between C7 and IJ. The partial orientation of the torso is given by the direction (C7 IJ). It is then possible to estimate the variation of position and partially the orientation of the torso. Table 5.3 presents the torso displacement using 2 indicators: the maximum distance between any torso position and the mean torso position ("Torso disp max"); and the standard deviation of the torso position ("Torso disp std"). Similarly, the orientation of the torso is estimated using 2 indicators: the maximum angle ("Torso ori max") between the torso orientation vector and the mean torso orientation vector and its standard deviation ("Torso ori std").

- The **shoulder displacement** is represented by the marker set on the acromion. It is well known that the shoulder is a complex joint with a non-unique center of rotation which depends on the position of the arm [3, 32, 135]. The human arm model used in this work does not reflect this complexity. It is not possible to directly track the human shoulder center of rotation, but the acromion is usually exploited for the estimation of its position as in [49]. So, the acromion marker is used to quantify the contribution of the shoulder parasitic motion to the trajectory error. Two indicators are given, the maximum displacement of the acromion marker relatively to its mean position ("Acromion disp max"); and the standard deviation ("Acromion disp std") of the acromion marker position (see Table 5.3).

- The **misalignment of the arm during the static pose of the initialization** can be estimated. Let us recall that, according to the model, the segment frames are aligned with the human reference frame during this static pose. The misalignment ("Misalignment") in Table 5.4) is defined as the distance between the tip of the gun and the theoretical position of the gun according to the model. The theoretical position of the gun tip, corresponding to the initial arm configuration, is computed as ${}^{H}\mathbf{AB}_n + {}^{H}\mathbf{BC}_n + {}^{H}\mathbf{CD}_n$ with $n = 0$.

- The error on **the position of the origin of H with respect to the codamotion frame** C can be estimated. The origin of H is the mean point of the circle center that fit the trajectory of the markers on the gun-shape tool, the wrist and the elbow. The standard deviation ("Center std") and the maximum distance with mean position ("Center dist max") are used to quantify the quality of the estimation of the origin of H with respect to C.

- The error on **the estimation of the y-axis of H with respect to** C is quantified by the angle between the **y**-axis of H and the normals of the circles that fit the trajectory of the markers during the rotating part of the initialization. The maximum angle ("y-axis angle max") and its standard deviation ("y-axis angle std") are presented in Table 5.4.

5.2.3 Discussion

The mean error shown in Table 5.3 is comprised between 54.8 mm and 181.8 mm. It can be recalled that the error in the previous experiment with the robot was comprised between 27.7

Measurement	Time (s)	Error max (mm)	Error mean (mm)	Error std (mm)	Torso disp max (mm)	Torso disp std (mm)	Torso ori max (deg)	Torso ori std (deg)	Acromion disp max (mm)	Acromion disp std (mm)
Circle 1	8.8	183.3	112.2	27.4	10.0	2.6	3.6	0.6	178.8	15.7
Circle 2	5.9	215.3	128.9	32.0	23.0	5.1	9.8	2.2	85.7	14.2
Circle 3	9.5	139.5	54.8	23.9	27.0	2.6	14.4	0.7	116.6	13.2
Cross 1	21.8	373.1	154.9	39.2	23.7	4.2	11.9	1.9	73.1	21.0
Cross 2	27.4	272.6	123.4	40.6	23.2	3.8	11.1	2.9	147.6	10.9
Cross 3	26.8	204.7	127.7	29.1	20.1	4.8	13.2	2.0	129.5	11.2
Cube 1	14.8	259.8	145.3	38.5	26.9	4.6	12.1	4.2	59.0	28.8
Cube 2	19.5	213.2	111.9	52.2	19.1	3.3	8.3	2.2	81.1	14.3
Cube 3	25.4	456.8	181.8	84.6	39.8	4.5	26.4	3.7	67.9	9.1
Hexagon 1	11.2	222.1	123.2	57.5	15.4	2.5	6.9	2.3	94.6	17.4
Hexagon 2	14.1	228.4	137.3	54.0	29.1	5.1	7.8	1.8	98.7	18.3
Hexagon 3	13.0	229.0	92.2	59.0	20.3	4.2	6.5	1.8	122.9	37.5

TABLE 5.3: Human arm trajectory error, modelling error indicators and parasitic motions indicators.

Measurement	Misalignment (mm)	Center dist max (mm)	Center std (mm)	y-axis angle max (deg)	y-axis angle std (deg)
Circle 1	149.3	95.9	19.5	3.1	0.6
Circle 2	104.1	42.1	12.2	4.7	1.2
Circle 3	100.4	118.1	29.1	3.4	0.8
Cross 1	101.2	73.2	22.8	11.6	2.9
Cross 2	115.0	147.5	39.0	11.6	3.2
Cross 3	152.0	90.1	20.7	7.9	1.7
Cube 1	132.4	75.7	17.6	5.6	1.3
Cube 2	84.9	66.1	16.8	13.5	3.9
Cube 3	81.8	35.4	6.1	21.5	5.1
Hexagon 1	157.7	78.4	15.5	4.3	1.1
Hexagon 2	144.4	42.7	11.4	4.3	1.2
Hexagon 3	142.4	87.2	26.1	3.7	0.7

TABLE 5.4: Initialization parasitic motion indicators.

mm and 47.7 mm and was mostly attributed to sensors orientation errors since the initialization is not prone to significant uncertainties in this case. Therefore, in the present experiment it can be considered that between $\frac{54.8-27.7}{54.8} = 49.4\,\%$ and $\frac{181.8-47.7}{181.8} = 73.8\,\%$ of the error do not come from the sensors orientation errors but from the modelling errors and the calibration errors. This is discussed in the following.

First, the parasitic torso and shoulder motions during the trajectory contribute to the hand position error:

- The standard deviation of the torso displacement is between 2.6 mm and 5.1 mm ("Torso disp std"). This indicates that the torso is fairly steady during the trajectory. However, the maximum displacement observed is 39.8 mm ("Torso disp max"), which means that a torso displacement can significantly contribute to the error sporadically.

- The standard deviation of the torso orientation is comprised between 0.6° and 4.2° ("Torso ori std"). Considering the arm in the initial configuration (straight and horizontal), the distance from the human reference frame origin and the fingertip is $L = \|{}^H\mathbf{AB}_n + {}^H\mathbf{BC}_n + {}^H\mathbf{CD}_n\|$ with $n = 0$, such rotating torso parasitic motions could involve a displacement between $\frac{0.6\pi L}{180} = 9.3$ mm and $\frac{4.2\pi L}{180} = 63.4$ mm.

- The shoulder displacement ("Acromion disp std") is between 9.1 mm and 37.5 mm. This displacement is partly induced by the torso parasitic motion.

The torso and the shoulder parasitic motions contribute in a non-negligible proportion to the error on the trajectory of the hand. In addition, some other parasitic motions related to the initialization procedure can be quantified from the values in Table 5.4.

- The column "Misalignment" of Table 5.4 clearly shows a misalignment of the arm during the **z**-axis identification. According to the values presented in the column "Misalignment", a potential error approximately between 5° and 10° could be induced in the parameters ${}^{Si}q_i$ and ${}^H q_{Si}$.

- The last two columns of Table 5.4 quantify the quality of the rotating motion of the initialization. The "y-axis angle" is the angle between the **y**-axis of H and the normal of the circle that fits the trajectory of a marker during this rotation of the arm. A low value of the maximum and the standard deviation of this parameter means that the different points of the arm follow a trajectory in parallel planes which implies a good execution of this step. A high value of this parameter reveals disturbances in the execution of this motion. The disturbances can come from the shoulder complex kinematics. But it can also reflect a bad execution of the recommended motion from the human during this step. If the operator does not maintain his/her arm straight during the motion, the axis of rotation of the trajectory is disturbed. The recommended motion is well executed in most cases since the "y-axis angle std" is often below 2°. But for 4 trajectories this parameter value is higher, which indicates this motion is not systematically executed well.

This analysis shows that the operator is not always able to perform the recommended motion with sufficient accuracy during the calibration phase which induces some uncertainties on the estimated model parameters. These uncertainties can be later compensated in the optimization of the parameters step. The main parasitic motion during the initialization procedure is the disturbed static pose which induces a misalignment of the human arm segment frames with respect to the human reference frame.

An additional part of the error is due to the Codamotion since it is difficult to identify the transformation between human reference frame and the Codamotion reference frame: ${}^H R_C$ and ${}^H\mathbf{AO}$. The error mainly comes from the identification of the origin of the frame H. The "Center

std" column shows that the centers of the circles that fit the trajectories of the different markers on the arm are scattered (from 6.1 mm to 39.0 mm). It can be estimated that this contribution could be responsible of 3.3 % to 71.3 % of the error in the worse case.

Finally, other sources of error may affect the hand positioning accuracy but their contributions have not been quantified:

- the length of the segments are measured from distances between anatomical landmarks which might present inaccuracies;
- even limited by the use of a gun-shape tool, finger motion could contribute to the error by introducing a displacement of the sensor with respect to the hand segment;
- the inertial sensors of the arm and forearm are in contact with soft tissues that might induce the displacement of the sensors with respect to the segment frame during the trajectory.

The results of this experiment show important sources of errors in the estimation of the trajectory of the human hand. Regarding the inaccurate execution of the initialization procedure, this leads to an error in the estimation of ${}^{Si}q_i$ and ${}^{H}q_{Si0}$. The optimization step, presented in chapter 4, was proposed to overcome this issue. The impact of this step is addressed in the next experiment.

5.3 IHMT method with respect to the robot base frame assessment

The aim of this section is to evaluate the accuracy of the complete method which evaluates the human hand trajectory with respect to the robot frame. This section studies the application of the proposed IHMT method with a human operator and a 7-DoF Sawyer collaborative robot from Rethink Robotics. The robot is programmed from its SDK interface based on ROS [112].

In the procedure to estimate the human-robot transformation, the robot imposes a trajectory to the operator. The trajectory measured by the robot encoders is used as the reference to estimate the accuracy of the IHMT estimation of the hand trajectory. Furthermore, comparing the accuracy of the complete method to the previous results enables to quantify the impact of the human-robot transformation estimation as well as the optimization of the parameters.

5.3.1 Experimental setup

Two IMUs are set up on the arm and forearm with reflective metallic strips (see Figure 5.8). The IMU for the hand is set in a gun-shape tool. The gun-shape tool is used to limit the impact of potential disturbing finger motions. Also, this tool includes a trigger button to communicate with the robotic system. The spherical tip of the gun fits in a receiver mounted at the robot end-effector to enforce the contact during the human-robot transformation estimation procedure and the recording of the human trajectory. The trajectory recorded through the robot encoders is used as reference. The measured segment dimensions of the subject are, in millimetres, ${}^{1n}\mathbf{AB}_n = [330\ 0\ 0]^T$, ${}^{2n}\mathbf{BC}_n = [280\ 0\ 0]^T$, ${}^{3n}\mathbf{CD}_n = [270\ -20\ 100]^T\ \ \forall n$. The minimization process is conducted by the *optimize.minimize* function from the SciPy library with the BFGS algorithm. Different trajectories are executed by a unique operator. The trajectories are similar than those

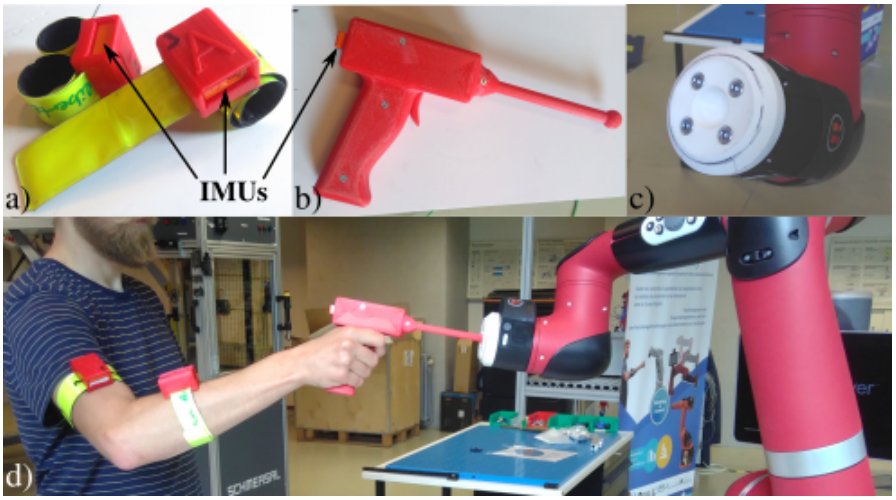

FIGURE 5.8: Experimental setup: a) IMUs set on the arm and forearm, b) gun-shaped tool with the trigger button, c) end-effector hole to receive the tip of the tool, d) global view of the setup.

from the previous experiment (see Figure 5.9). The mean error ε is computed from the distance between the hand position ${}^{R}\mathbf{OD}_k$ with respect to the robot base frame R estimated from the proposed IHMT method and the measured robot end-effector position ${}^{R}\mathbf{OP}_k$ with O the origin of the robot base:

$$\varepsilon = \frac{1}{N}\sum_{k=1}^{N} \|{}^{R}\mathbf{OP}_k - {}^{R}\mathbf{OD}_k\| \tag{5.8}$$

with N the number of points.

5.3.2 Results and discussion

The results are exposed in Table 5.6. The table shows the mean ε of the error as well as the standard deviation σ. The mean error ε with non-optimized parameters is comprised between 47.2 mm and 96.2 mm and is lower than the values from the previous experiment (between 54.8 mm and 181.8 mm). This difference can be explained by the difficulty to identify the transformation between the human reference frame and the Codamotion frame. It seems that the errors on the human-robot transformation is negligible compared to the errors caused by the parasitic motions exposed in the previous section.

With optimized parameters, the error on the hand trajectory with respect to the robot frame falls down between 28.5 mm and 61.8 mm. The benefit of the optimization step is clear. In the survey [49], Filippeschi et al. evaluated the accuracy of 5 different IHMT methods between 23.6 mm and 89.2 mm. Our method has a similar level of accuracy but it offers the advantage that it does not rely on magnetometer measurements. Let us mention that our method takes advantages of the PbD context, e.g., for the heading reset and the optimization of the parameters. In addition, the effect of the sensor orientation errors measured in the first experiment conducted on a robotics arm (see Section 5.1) is between 27.7 mm and 47.7 mm. The level of accuracy of the complete method is close to this error interval. Therefore, the optimization step catches up a large part of the calibration errors. The modelling errors (the parasitic motions of the torso and

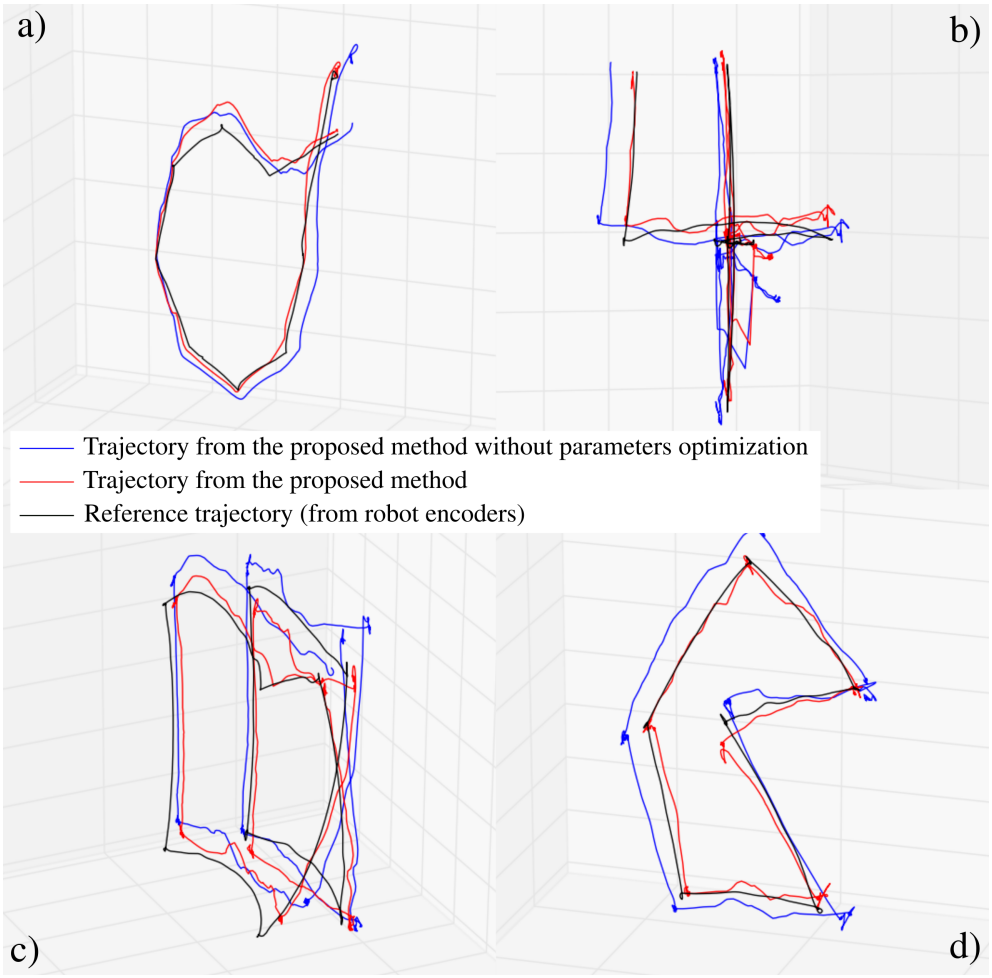

FIGURE 5.9: Trajectory imposed by the robot: a) “Circle” trajectory, b) “Cross” trajectory, c) “Cube” trajectory, d) “Hexagon” trajectory.

the shoulder, soft tissue displacement, finger motions, etc., see Section 4.4) still affect the hand trajectory accuracy.

The difference between non-optimized and optimized values of each parameter is computed for all trials. Table 5.5 gives the maximum and the minimum of these differences. The differences are low, which indicates that the parameters conserve a physically plausible value. It can also be observed that the optimization step mainly corrects the orientation parameters and in particular the I2S segment orientation.

Parameters	Maximum difference	Minimum difference
${}^{1n}\mathbf{AB}$	0.53 mm	0.012 mm
${}^{2n}\mathbf{BC}$	0.44 mm	0.012 mm
${}^{3n}\mathbf{CD}$	0.29 mm	0.003 mm
${}^{S1}\hat{q}_1$	21.8 °	13.0 °
${}^{S2}\hat{q}_2$	32.7 °	20.0 °
${}^{S3}\hat{q}_3$	32.9 °	20.3 °
${}^{H}\hat{q}_{S10}$	27.1 °	8.6 °
${}^{H}\hat{q}_{S20}$	23.4 °	7.8 °
${}^{H}\hat{q}_{S30}$	33.3 °	6.2 °
${}^{H}\hat{q}_{R}$	23.6 °	3.2 °
${}^{R}\mathbf{AO}$	1.2 mm	0.015 mm

TABLE 5.5: Different of values before and after optimization.

Measurement	Time (s)	non-optimized parameters		optimized parameters	
		ε (mm)	σ (mm)	ε (mm)	σ (mm)
Circle 1	8.55	52.0	14.7	32.9	20.5
Circle 2	7.91	53.8	12.7	28.8	10.8
Circle 3	8.25	50.1	19.7	29.4	12.8
Cross 1	27.99	55.5	32.7	39.9	26.0
Cross 2	27.71	47.2	28.4	42.8	20.1
Cross 3	27.08	77.0	35.0	61.8	27.4
Cube 1	44.24	85.8	33.5	45.0	13.4
Cube 2	39.19	96.2	53.7	50.4	20.5
Cube 3	39.07	83.8	43.4	39.8	17.7
Hexagon 1	28.47	55.4	16.4	28.5	16.7
Hexagon 2	27.77	68.5	26.5	40.9	20.5
Hexagon 3	27.09	70.7	26.4	52.8	22.1

TABLE 5.6: Human arm trajectory in robot base frame error

Even if the results are encouraging, this level of accuracy is still not sufficient for the acquisition of demonstration in a PbD context. For a pick-and-place task, an error of 28.5 mm when picking the object is too high. Several approaches could be investigated to improve accuracy. For

instance, instead of being modelled as a spherical joint, the elbow could be modelled as a hinge joint. This more restricted kinematics may help to improve the segment orientation estimation. For example, Seel et. al [80, 117] propose a method to estimate the elbow joint axis with respect to the arm and the forearm frames in a first step. Then, during the trajectory, the elbow joint axis can be estimated with respect to the IMUs attached to the arm and forearm. Comparing these two values enables to partially correct IMUs orientation. Such approach might require an extra procedure step to estimate the elbow joint axis with respect to the segment frame. Regarding the torso and shoulder parasitic motions and the complexity of the shoulder joint, some enhancement could be investigated such as the development of a more detailed arm model or the use of extra IMU sensors to track parasitic motions. However, our goal is to develop simple robot programming methods. Therefore, one should then evaluate the increased complexity of the procedure with respect to the potential gain in accuracy. One may also keep in mind that the same level of accuracy is not always required all along the trajectory in a pick-and-place task.

5.4 Summary

In this chapter, several experiments are conducted in order to evaluate the different elements of the proposed IHMT method presented in Chapter 4.

Firstly, the method is applied to a robotic arm. The robot is not affected by parasitic motions during the trajectory or during the initialization phase, i.e., the modelling and calibration errors are negligible in this case. The results show that the method is robust to sensors positioning on the arm.

Secondly, the method is applied to a real human arm. The reference motion is measured using an optoelectronic device. The error is significantly higher than in the previous experiment due to parasitic motions, i.e., modelling and calibration errors have a significant influence in this case. Some sources of error are identified and their contribution quantified. This experiment reveals a certain difficulty to perform the recommended motions of the initialization procedure and therefore justifies the optimization step of the parameters of the proposed IHMT method.

Thirdly, the complete method is applied with a Sawyer robot. This experiment shows the positive effect of the parameter optimization step. The level of accuracy of the complete IHMT method is then evaluated between 28.5 mm to 61.8 mm. The IHMT method gives reliable results. The impact of the calibration errors is limited by the optimization step. However, the sensor orientation errors and the modelling errors still degrade the IHMT estimation of the hand position making the level of accuracy still too low for an application in a PbD process. The following chapter proposes a solution to reach a sufficient level of accuracy.

CHAPTER 6

CORRECTION OF THE IHMT HAND TRAJECTORY WITH ENVIRONMENT OBSERVATION

Contents

As discussed in the introduction (see Section 1.2.3), and considering the amplitude of the sensors orientation errors and the modelling errors discussed in the last chapter (see Section 5.2.2), the IHMT method may hardly reach a sufficient level of accuracy for PbD. The method proposed in this chapter aims at improving the accuracy of the human hand trajectory estimation without increasing the complexity and the user difficulty in the context of the programming by demonstration of a pick-and-place task. It consists in correcting the hand trajectory using additional information. Vision-based technologies appear as a good candidate for that purpose. As presented in the introduction, the development of a combined approach is motivated by the complementarity between vision and inertial technologies. Vision-based technologies are often more accurate than inertial sensors, but inertial-based technologies are easier to implement and use. For the sake of simplicity, it is proposed to track by vision the environment instead of the teacher. In this way, we avoid many difficulties encountered with the use of vision systems for human motion tracking (see Section 1.1.3). In addition, a vision system is often necessary in PbD applications to track some elements in the task workspace. Hence, such approach does not necessarily require additional hardware. Pick-and-place applications being the targeted context in this work, tracking the environment consists at least in tracking the objects to be manipulated in the environment. These objects can be the manipulated objects or the tools used by the teacher like pliers or screwdrivers.

Vision-based methods for objects tracking have been recently reviewed in [24]. This subject is not the main focus of this work and we shall simply assume that a vision system gives the reliable positions of the objects but with a lower acquisition frequency than inertial data and with occasional concealment. We also assume that the positions of the objects can be expressed with respect to the same frame as the hand trajectory.

The method proposed in this chapter relies on positions of interest (POIs). A POI corresponds to a static position linked to an action during the task. In the context of the demonstration of a pick-and-place task, the action of picking an object is expected to generate two POIs: one POI corresponding to the static hand position when picking the object and another POI corresponding to the object position before the grasp. The correction of the hand POI – based on inertial sensors – with the object POI – based on vision sensors – is expected to increase the accuracy of the human hand trajectory when picking up the object. Thus, the human hand trajectory can be corrected in this way to ensure a sufficient accuracy for specific actions. The rest of the trajectory (which could include avoidance motion) may have a lower accuracy. The proposed method, which shall be described in the second section of this chapter, relies on the 3 following steps:

- Identification of positions of interest (POIs) in the hand trajectory and the object positions. Since POIs correspond to static positions, they form a dense group of data points with close spatiotemporal coordinates. They can be identified by searching clusters in their spatiotemporal representations using a clustering method.
- Pairing hand POIs from inertial data to objects POIs from vision data corresponding to the same action.
- Correcting the hand trajectory with the matched object POIs.

The first two steps can be solved using stat-of-the-art methods as explained in the section 6.1. The third step involves more personal developments which will be presented in Section 6.4.

6.1 State-of-the-art and selection of algorithms

The first two steps of the correction method are based on methods found in the literature and adapted to our purpose. The first step, consisting in extracting the POIs from both hand and object dataset, is based on clustering.

6.1.1 Clustering

The objective of a clustering method is to identify elements in a dataset that should belong to the same group. Various solutions have been proposed in the literature and can be classified into two categories: the hierarchical clustering and the partitional clustering [45, 103].

In *hierarchical clustering* methods, the data are "partitioned into levels in a hierarchical format" [45]. Two main approaches can be distinguished. The first one divides the complete set of data into smaller clusters [66, 68]. The second one agglomerates data into clusters [119]. According to Ezugwu et al. [45], hierarchical clustering algorithms can be applied to any data type but they present a lack of robustness to noise and outliers.

In *partitional clustering* methods the data are "organized into a nested sequence of groups without any hierarchical structure". This category includes methods such as EM-algorithms [77] or the K-means algorithm [40] and its derivatives [18, 23]. The number of clusters to be detected should be specified in advance. These methods are less sensitive to the initial conditions and they are robust to noise but they are generally computationally expensive.

The IHMT hand trajectory is considered noisy due to the non-negligible level of errors measured in Section 5.3. It also present outliers since some points of the trajectory correspond to non-static phases (like avoidance motions) and should not be involved in any POI and thus in any cluster. Moreover, the demonstrations are supposed to last a few seconds or minutes, leading to a moderate amount of data so that the computational cost of partitional clustering algorithms will remain small. Therefore, partitional clustering algorithms appear more suitable for our application.

Among the partitional clustering category, so-called density-based algorithms isolate the dense areas that are separated by sparse areas as defined in [4, 19]. According to [45], "generation of the cluster this way enables the building of clusters with arbitrary shapes. Outliers or noisy data points are naturally eliminated". In particular, "the DBSCAN has a well-defined cluster model with fairly low complexity", still according to [45]. A number of interesting properties are confirmed by [14]. For this reason, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm [38] is selected in this work. This algorithm creates a cluster from one or several core points (see Figure 6.1). A *core point* p_1 is a point with at least $MinPts$ points within distance ε. The two parameters of the algorithm are then ε and $MinPts$. In our case, the Euclidean distance is chosen to compute the distance between points. A point p_2 within the distance ε to p_1 is called *directly reachable* from p_1. A point p is considered *reachable* from p_1 if a path $[p_1, ..., p_i, p_{i+1}, ..., p]$ exists in which p_{i+1} is directly reachable of p_i. All core points, directly reachable points and reachable points form a cluster. The points that are not reachable from any core point are outliers and are not part of any cluster.

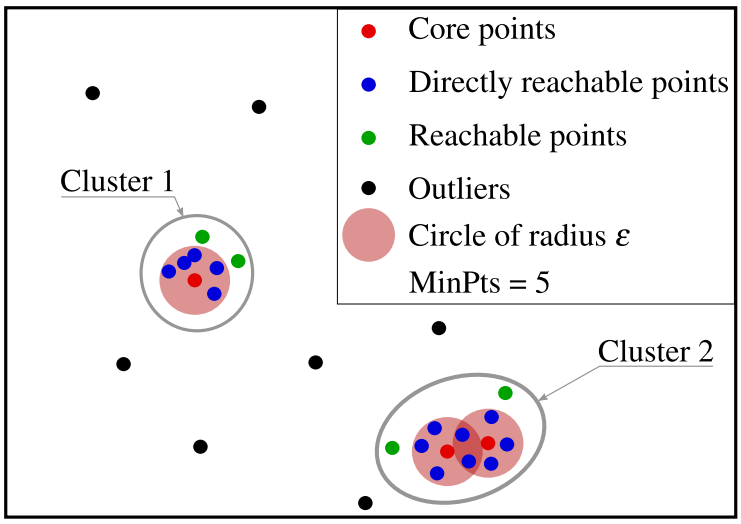

FIGURE 6.1: Illustration of the principle of the clustering DBSCAN algorithm.

6.1.2 Pairing

Once the POIs are extracted thanks to the clustering algorithm, the second step consists in associating some objects POIs to some hand POIs. During the demonstration, some objects can be present in the scene but not manipulated, which could lead to POIs unrelated to any actions. The teacher may observe stops during the demonstration that could also lead to POIs unrelated to any relevant actions. Such POIs should not be associated with another POI. The pairing method should be able to distinguish such cases.

The problem of pairing or matching points among two sets is largely documented in computer vision with applications involving 2D points, 3D points, or spatiotemporal points. In [61], the matching is solved by using the Haussdorff distance to quantify the similarity between two different sets of points. In [105], the matching problem is solved by reordering a sequence of key points thanks to a genetic algorithm. This method relies on the condition that the relative position of any two points in adjacent area from two images is almost unchanged. It is therefore not applicable in our context. In [131], for video treatment, each space-time key point is described based on its surroundings. Thanks to this description, a putative set of pairs is created and the challenge consists in eliminating the non-desired pairs. This approach can also be seen in [118] which compares different algorithms to eliminate false pairs (RANSAC, K-nearest-neighbour graph, Graph Transformation Matching); or in [108], where a convolutional neural network is used to characterize 3D points. These methods are specific to image treatment (describing key points thanks to their neighbourhood in the image for instance) and are difficult to transpose to the problem of matching POIs. In [16], for shape recognition, the exact sets of points to pair are known so that the problem boils down to an assignment problem solved by the shortest-augmenting-path-based algorithm [65]. In our context, the exact number of pairs is not known, so that this method is not applicable.

A variety of matching methods exist and the choice is generally guided by the application and the final goal (geometry transformation in stereo vision, pattern recognition ...). Therefore, we propose here a specific method for our application, which is inspired by elements seen in the

other works. A distance function is used to estimate the similarities between POIs as in [61]. The matching problem is reduced to an assignment problem as in [16] by eliminating POIs that should not be involved in any pairs.

The approach proposed in this work is based on an iterative process. At each step a hand or an object POI is eliminated. The criterion to eliminate a POI is based on a specific time-space distance function. The distance is computed from the modified spatial part of the hand POIs. The spatial part of the hand POIs is modified by a geometric transformation. This geometric transformation aims at bringing the subsets of hand POIs to pair closer to the object POIs to pair. It is detailed later in Section 6.3. Identifying this transformation is a point-set registration (PSR) problem widely documented, see Zhu et al. [144] and Maiseli et al. [90]. Since both sets of hand and object POIs contain outliers (POIs that should not be involved in any pairs) and noise (error on the IHMT hand trajectory), the kernel correlation based method [124] is selected for our application.

The following three sections present the three steps of the method to correct the IHMT hand trajectory with object positions measurements from the vision system.

6.2 POIs identification

A POI $\mathcal{P}$ is defined as a stationary position during the task. It is represented by a 3D position $\mathbf{P}$ and two temporal components indicating the beginning s and the end e of the POI: $\mathcal{P} = [s, \mathbf{P}, e]$. A POI is supposed to correspond to an action. It is considered that 2 actions cannot occur simultaneously. Thus, the time interval of two POIs from the same trajectory (hand or object trajectory) cannot overlap.

6.2.1 Time-space scaling

For POIs identification (as well as POIs matching), it is necessary to compare data spatially and temporally. Two different actions during the task, which are represented by two different POIs, could be at the same 3D position on the scene, but they should differ in time. Spatial and temporal data having different physical units, their scaling should be carefully selected.

Considering a dataset Γ of N points $\mathbf{V}_i$ with $i \in [1,N]$, where each point $\mathbf{V}_i$ is a vector of n temporal and spatial components. For example, if a point $\mathbf{V}_i = (t_i, \mathbf{P}_i)$ has one temporal and 3 spatial components, $n = 4$. The gravity center $\mathbf{G}$ and the scale vector $\mathbf{S}$ are computed component-wise as:

$$\mathbf{G} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{V}_i \tag{6.1}$$

$$\mathbf{S} = [\max_i |\mathbf{V}_i(1) - \mathbf{G}(1)| \ , \ ... \ , \ \max_i |\mathbf{V}_i(n) - \mathbf{G}(n)|] \tag{6.2}$$

Then, the scaled dataset $\gamma = [\mathbf{v}_1 \ , \ ... \ , \ \mathbf{v}_N]$ is composed of scaled points $\mathbf{v}_i$ computed as

$$\mathbf{v}_i = [\frac{\mathbf{V}_i(1) - \mathbf{G}(1)}{\mathbf{S}(1)} \ , \ ... \ , \ \frac{\mathbf{V}_i(n) - \mathbf{G}(n)}{\mathbf{S}(n)}] \ . \tag{6.3}$$

The object data recorded from the camera and the hand data recorded from IMUs present different features. IMU sensors transmit a continuous flow of data but suffer from noise and errors. On the other hand, the camera could be occluded during the execution of the task, resulting in an irregular data stream. In addition, objects being more static than human hand during the demonstration, the recorded data is less noisy. Therefore, the DBSCAN algorithm to extract POIs is applied differently to each dataset.

6.2.2 The object POIs

It is considered that the object detection and positioning have been obtained using a suitable method. The 3D object positions with respect to the robot base frame are therefore known. It can be recalled that the hand trajectory is also expressed with respect to the robot base frame.

Let us consider that N^{obj} objects are identified. For each object j ($j \in [1, N^{obj}]$), its trajectory Γ^j is recorded. The trajectory Γ^j, represented by N^j 3D positions $\mathbf{P}_i$ and times t_i with $i \in [1, N^j]$, is noted:

$$\Gamma^j = [\ (t_1\ \ \mathbf{P}_1)\ ,\ \ldots\ ,\ (t_i\ \ \mathbf{P}_i)\ ,\ \ldots\ ,\ (t_{N^j}\ \ \mathbf{P}_{N^j})\ ]. \tag{6.4}$$

Each trajectory Γ^j is scaled and noted γ^j. For each γ^j, the DBSCAN algorithm is applied on the scaled data and returns N^c clusters. The results of the DBSCAN being a list of indexes attached to γ^j, it can be directly transposed to the non-scaled data Γ^j. So a cluster C^k ($k \in [1, N^c]$ is made of N^k points of Γ^j:

$$C^k = [\ (t_u\ \ \mathbf{P}_u)\ ,\ \ldots\ ,\ (t_v\ \ \mathbf{P}_v)\ ] \tag{6.5}$$

An object POI $\mathbb{O}_k$, associated to the cluster C^k, is computed as

$$\mathbb{O}_k = [\min([t_u, \ldots, t_v])\ ,\ \frac{1}{N^k}\sum_{j=u}^{v}\mathbf{P}_j\ ,\ \max([t_u, \ldots, t_v])]. \tag{6.6}$$

The set of extracted POIs depends on the value of the DBSCAN algorithm parameters (see Section 6.1). A too high value of ε might miss some POIs and a too low value might create unnecessary POIs. For a better robustness, a low value of ε is preferred and a post-treatment to merge close POIs is added. Let us consider two consecutive POIs $\mathbb{O}_j = [s_j, \mathbf{P}_j, e_j]$ and $\mathbb{O}_k = [s_k, \mathbf{P}_k, e_k]$ such as $s_j < s_k$. These consecutive POIs are temporarily close if $s_k < e_j + t_{limit}$, and spatially close if $\|\mathbf{P}_j - \mathbf{P}_k\| < d_{limit}$. So, two consecutive POIs $\mathbb{O}_j = [s_j, \mathbf{P}_j, e_j]$ and $\mathbb{O}_k = [s_k, \mathbf{P}_k, e_k]$ spatially or temporally close are merged as

$$\mathbb{O}_j = [\min([s_k, s_j])\ ,\ \frac{\mathbf{P}_k + \mathbf{P}_j}{2}\ ,\ \max([e_k, e_j])]. \tag{6.7}$$

The motivation to merge the POIs when they are only spatially close comes from limitations of the vision-based positioning method. Indeed, two POIs might be spatially close but not temporally close due to concealment issues. An object, manipulated later in the task, might be occluded during the early actions. Thus, two POIs with the same 3D position but with distant temporal

boundaries may correspond to the same action and should be merged. Finally, in the case of an object manipulated twice, one POI is created corresponding to two actions: the first placing and the second picking. In that case the POI is split.

6.2.3 The hand POIs

The clustering of the hand trajectory with the method described in Section 6.2.2 would be too sensitive to the parameters ε and $MinPts$ and would then detect unwanted POIs or miss others. In order to make the POIs detection more robust, the DBSCAN algorithm is applied twice in a different way.

First, the DSBCAN algorithm is applied on the scaled human hand trajectory N^{ε} times with different values of $\varepsilon \in E$ with $E = [\varepsilon_1, ..., \varepsilon_{N^{\varepsilon}}]$, while $MinPts$ is fixed. To each value ε_i ($i \in [1, N^{\varepsilon}]$), a set of potential POIs $\mathcal{P}^i$ is formed from the output of the DBSCAN algorithm in the same way as for the object POIs as described in the previous section 6.2.2. The only difference is that no post-treatment is applied. Then, the complete set $\mathcal{P}$ of all potential POIs is constructed as

$$\mathcal{P} = [\mathcal{P}^1, ..., \mathcal{P}^{N^{\varepsilon}}]. \tag{6.8}$$

Secondly, $\mathcal{P}$ is scaled and noted $\not{\mathcal{P}}$. The DBSCAN algorithm is applied again on $\not{\mathcal{P}}$ where a point is then a 5-components vector. The clustering results are less sensitive to the parameters values of the DBSCAN algorithm. A final hand POI $\mathcal{H}_k$, corresponding to the k^{th} cluster, is computed from the N^k POIs $[\mathcal{P}_u, ..., \mathcal{P}_v]$ belonging to this cluster. $\mathcal{H}_k$ is computed as

$$\mathcal{H}_k = [\min([s_u, ..., s_v]) \ , \ \frac{1}{N^k} \sum_{k=u}^{v} \mathbf{P}_k \ , \ \max([e_u, ..., e_v])]. \tag{6.9}$$

Finally, a post-treatment is applied. First, the POIs from the set $\mathcal{H}$ are reordered consecutively, i.e., they are sorted chronologically according to their start time. Then, the POIs are merged or separated according to their spatiotemporal distance. Considering two consecutive POIs $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ temporally close, i.e., $s_{i+1} < e_i + t_{limit}$:

- If $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ are also spatially close, i.e., $\|\mathbf{H}_i - \mathbf{H}_{i+1}\| < d_{limit}$, the POIs are considered representing the same action. Both POIs are merged as

$$\mathcal{H}_i = [\min(s_i, s_{i+1}) \ , \ \frac{\mathbf{H}_i + \mathbf{H}_{i+1}}{2} \ , \ \max(e_i, e_{i+1})]. \tag{6.10}$$

- If $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ are distant, i.e., $\|\mathbf{H}_i - \mathbf{H}_{i+1}\| > d_{limit}$, the POIs represent different actions. Since two actions cannot occur at the same moment, the POIs are differentiated. To differentiate two POIs, two cases appear:

  - If $\mathcal{H}_{i+1}$ is temporally included in $\mathcal{H}_i$, only e_i is modified as $e_i = s_{i+1} - t_{limit}$ to eliminate the temporal overlap and prevent the suppression of the included POI $\mathcal{H}_{i+1}$ as illustrated in Figure 6.2 a.

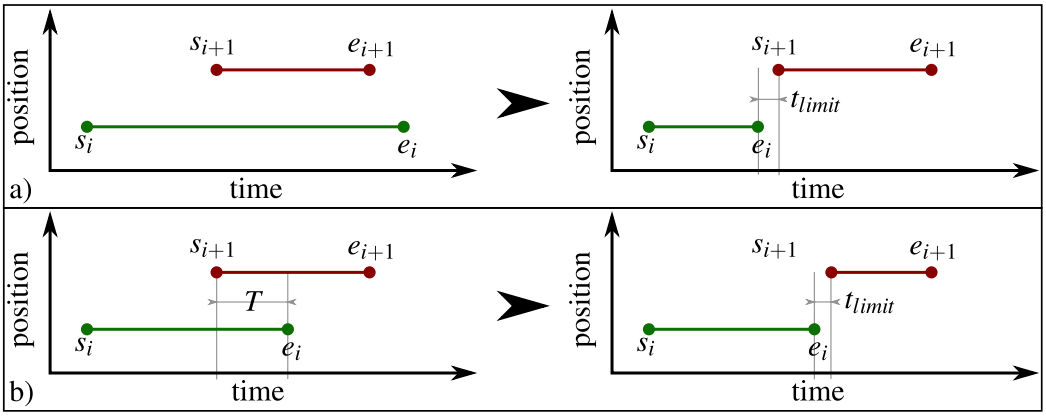

FIGURE 6.2: Illustration of the differentiation of the hand POIs.

- If the time intervals of $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ are overlapping, i.e., $s_{i+1} < e_i < e_{i+1}$, POIs are modified proportionally to their duration (see Figure 6.2 b) as

$$T = e_i - s_{i+1} \tag{6.11}$$

$$\alpha = \frac{e_i - s_i}{(e_i - s_i) + (e_{i+1} - s_{i+1})} \tag{6.12}$$

$$\mathcal{H}_i = [s_i \, , \, \mathbf{H}_i \, , \, e_i - \alpha T - \frac{t_{limit}}{2}] \tag{6.13}$$

$$\mathcal{H}_{i+1} = [s_{i+1} + (1-\alpha)T + \frac{t_{limit}}{2} \, , \, \mathbf{H}_{i+1} \, , \, e_{i+1}] \tag{6.14}$$

This procedure is applied iteratively to the complete set $\mathcal{H}$ until the number of hand POIs can no longer be modified. This post-treatment is detailed in Algorithm 2. Finally, as for object POIs, POIs whose time interval $e_j - s_j$ is lower than t_{limit} are discarded.

Algorithm 2 : Post treatment of hand POIs

$\mathcal{H}$ is sorted
N^H :=Number of POIs in $\mathcal{H}$
$N^h := 0$
while $N^h \neq N^H$ **do**
  $i := 1$
  N^H :=Number of POIs in $\mathcal{H}$
  while $i < N^H - 1$ **do**
    if $e_i > s_{i+1} - t_{limit}$ **and** $\|\mathbf{H}_i - \mathbf{H}_{i+1}\| < d_{limit}$ **then**
      $\mathcal{H}_i := [\min([s_i, s_{i+1}])\ ,\ \frac{\mathbf{H}_i + \mathbf{H}_{i+1}}{2}\ ,\ \max([e_i, e_{i+1}])]$
      $\mathcal{H}_{i+1}$ is discarded
      $i := i + 1$
    else if $e_i > s_{i+1} - t_{limit}$ **and** $e_i > e_{i+1}$ **then**
      $\mathcal{H}_i := [s_i\ ,\ \mathbf{H}_i\ ,\ e_{i+1} - t_{limit}]$
    else if $e_i > s_{i+1} - t_{limit}$ **and** $e_i < e_{i+1}$ **then**
      $T = e_i - s_{i+1}$
      $\alpha = \frac{e_i - s_i}{(e_i - s_i) + (e_{i+1} - s_{i+1})}$
      $\mathcal{H}_i := [s_i\ ,\ \mathbf{H}_i\ ,\ e_i - \alpha T - \frac{t_{limit}}{2}]$
      $\mathcal{H}_{i+1} := [s_{i+1} + (1-\alpha)T + \frac{t_{limit}}{2}\ ,\ \mathbf{H}_{i+1}\ ,\ e_{i+1}]$
    end if
    $i := i + 1$
  end while
  N^h :=Number of POIs in $\mathcal{H}$
end while

6.2.4 The parameters

The presented method requires the setting of several parameters values:

- DBSCAN algorithm parameters for objects POIs detection : ε_o and $MinPts_o$,
- DBSCAN algorithm parameters for the first step of hand POIs detection : E and $MinPts_{h1}$,
- DBSCAN algorithm parameters for the second step of hand POIs detection: ε_h and $MinPts_{h2}$,
- the two thresholds for post treatment: t_{limit} and d_{limit}.

The selected values are presented in the next chapter (Chapter 7) describing the experimental implementation of the method. Several considerations regarding the values of the parameters are exposed in this section. The DBSCAN algorithm is applied on scaled data so that the same values for the ε parameters can be used:

$$\varepsilon_o = \varepsilon_h = \varepsilon \tag{6.15}$$

For the same reason, E is a list of value centered on ε. The number of values in E depends on the quality of the hand trajectory. For a hand trajectory with low noise, the range of values of ε_i in E can be small.

The values of the parameters $MinPts$ mainly depend on the data acquisition frequency. It is assumed that the object detection and positioning method presents a frequency of acquisition lower than the inertial system due to the possible concealment. So, we require $MinPts_o < MinPts_{h1}$. Since the second part of the hand POI detection is not applied on the inertial data but on a set of scattered POIs that presents similarities with vision data, the value of $MinPts_o$ can be used for $MinPts_{h2}$:

$$MinPts_{h2} = MinPts_o. \tag{6.16}$$

The value of the parameter d_{limit} can be adjusted based on the quality of the data. The noisier the data, the higher d_{limit} should be. The parameter t_{limit} can be interpreted as the human reaction time. It is commonly admitted that the mean human reaction time is approximately 250 ms. It is considered that the minimum time interval between two actions is twice the mean human reaction time, so t_{limit} is tuned to 0.5 s.

The pairing process, described in the next section, aims at pairing the POIs corresponding to the same significant action of the task from each dataset $\mathcal{H}$ and $\mathcal{O}$ respectively from the hand and objects trajectories.

6.3 Pairing hand and object POIs

Let us consider two sets $\mathcal{H}$ (for the hand) and $\mathcal{O}$ (for the objects) containing respectively N^H and N^O POIs. These two sets contain the POIs corresponding to the significant actions of the demonstrated task: picking, placing, dropping, etc. They can also include POIs corresponding to less relevant actions: initial or final hand positions, pauses observed by the teacher during the demonstration in $\mathcal{H}$ or non-manipulated objects in $\mathcal{O}$. The method described in this section aims at identifying the set $\mathfrak{P}$ of N^P pairs of POIs corresponding to the significant actions of the task (see Figure 6.3). A pair $\mathfrak{P}_i$ with $i \in [1, N^P]$ contains a hand and an object POIs: $\mathfrak{P}_i = [\mathcal{H}_j\ ,\ \mathcal{O}_k]$ with $j \in [1, N^H]$ and $k \in [1, N^O]$. Since a POI may at most be involved in only one action, it can belong to maximum one pair. The first step of this procedure consists in identifying the two subsets of hand POIs and object POIs that are to be involved in the pairs. The second step creates the pairs by assignment.

6.3.1 Identification of POIs to pair

The two sets of hand POIs $\mathcal{H}$ and object POIs $\mathcal{O}$ contain the subsets $\mathcal{H}^*$ and $\mathcal{O}^*$ and some outliers $\mathcal{H}^{out}$ and $\mathcal{O}^{out}$. $\mathcal{H}^*$ and $\mathcal{O}^*$ correspond to the significant actions and must be involved in the pairs. The outliers $\mathcal{H}^{out}$ and $\mathcal{O}^{out}$ should not be in any pair. Recall that the associated set of the 3D positions of the hand POIs and the objects POIs are denoted respectively H and O. Similarly H^*, O^*, H^{out} and O^{out} are the sets of 3D positions of $\mathcal{H}^*$, $\mathcal{O}^*$, $\mathcal{H}^{out}$ and $\mathcal{O}^{out}$. The objective of this step is then to identify $\mathcal{H}^*$ and $\mathcal{O}^*$. The process is iterative, at each step, either a hand POI or an object POI is removed from $\mathcal{H}$ and $\mathcal{O}$ until a certain condition is met.

The criterion to eliminate a POI is based on a specific space-time distance function d described later. To facilitate the identification of the POIs to be discarded, d is applied on the

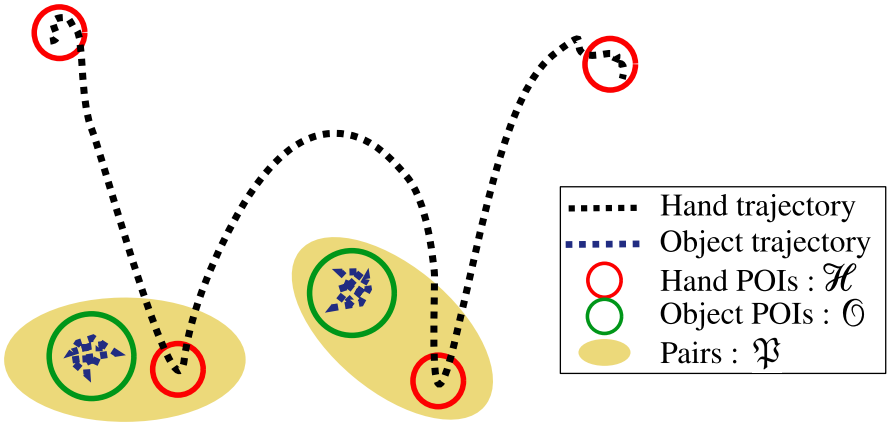

FIGURE 6.3: Illustration of the pairing of POIs.

modified POIs $\mathscr{H}'$. The modification concerns the 3D positions of the POIs to partially compensate the error on the IHMT measurement. H is modified into H' with a global geometric transformation T_{HO}. T_{HO} is the estimated transformation between H^* and O^*. H^* and O^* correspond to the same actions and are expected to have similar but noisy shapes. T_{HO} is estimated from the sets H and O which contain some outliers H^{out} and O^{out}. Thus, the estimation of T_{HO} is a point-set registration problem with noisy datasets that contain outliers. As presented in Section 6.1.2, the kernel correlation based method [124] is used to estimate T_{HO}. Once H' is computed from H with T_{HO}, it is expected that the hand POIs $\mathscr{H}^*$ are now spatially closer to their corresponding object POIs $\mathscr{O}^*$. Therefore, the space-time distance d between POIs is more relevant to determine which POI should be discarded.

In the case of a task involving only one pick and place, a small amount of POIs is extracted. With such a low amount of POIs, two subsets of POIs might have a more similar shape than H^* and O^*. In this case, the kernel correlation based PSR method returns an irrelevant transformation as illustrated in Figure 6.4. To avoid this problem, some *a priori* information about the error can be exploited. As explained in Chapter 2 and seen in Table 5.2, the sensor orientation errors are mainly around the gravity direction which is the **z**-axis of the reference frame. Thus, the transformation T_{HO} is limited to a 3D translation $\mathbf{T} = [T_x\ T_y\ T_z]^T$ and a 1-DoF rotation $R_z(\theta)$ of angle θ around the **z**-axis.

One iteration of the process of identifying the POIs to pair then involves two steps: the estimation of T_{HO} and the elimination of a POI based on the specific distance function d. These two steps are detailed in the following.

Estimating T_{HO}

Let us introduce the kernel correlation function KC, based on a Gaussian kernel. Considering the spatial component $\mathbf{O}_i$ and $\mathbf{H}_j$ of two POIs $\mathscr{O}_i$ and $\mathscr{H}_j$, the Gaussian kernel correlation function $KC(\mathbf{O}_i\,,\mathbf{H}_j)$ is

$$KC(\mathbf{O}_i\,,\mathbf{H}_j) = (2\pi\sigma^2)^{-N/2}\exp\left(\frac{-\|\mathbf{O}_i-\mathbf{H}_j\|^2}{2\sigma^2}\right) \tag{6.17}$$

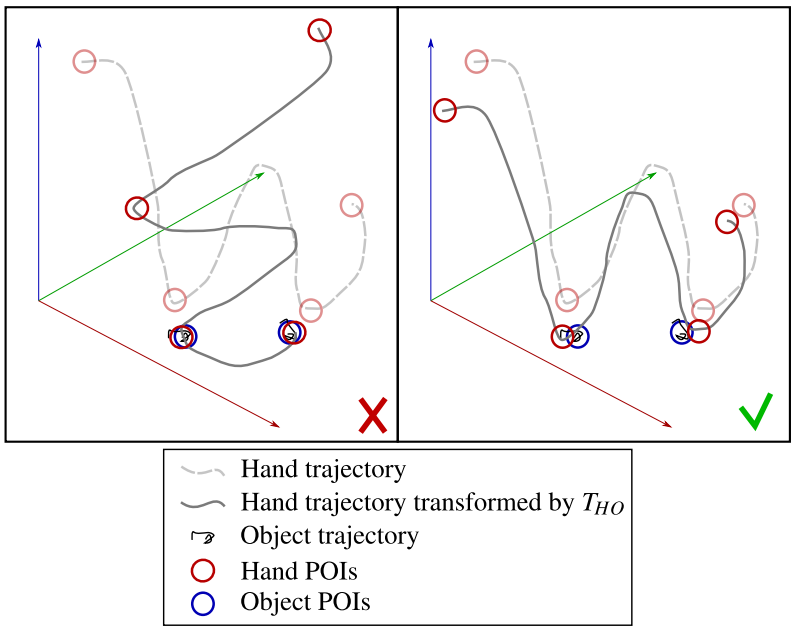

FIGURE 6.4: Irrelevant vs relevant estimation of T_{HO} with a low number of POIs.

with $\|.\|$ the Euclidean norm and σ^2 the variance. The transformation T_{HO} is estimated through the minimization of the following cost function as in [124]

$$COST(O,H,T_{HO}) = \sum_{i=1}^{N^O}\sum_{j=1}^{N^H} -KC(\mathbf{O}_i, T_{HO}(\mathbf{H}_j)) \tag{6.18}$$

with

$$T_{HO}(\mathbf{H}_j) = R_z(\theta)\mathbf{H}_j + \mathbf{T}. \tag{6.19}$$

Then, the transformed hand POIs set $\mathcal{H}'$ is computed by modifying the spatial component parts of each POI as

$$\mathbf{H}'_j = T_{HO}(\mathbf{H}_j). \tag{6.20}$$

Discarding a POI

The set of transformed hand POIs $\mathcal{H}'$ and the set of object POIs $\mathcal{O}$ are scaled to $\mathcal{H}'$ and $\mathcal{O}$. Let us introduce the distance function $d(o_i, h'_j)$ between two POIs, illustrated in Figure 6.5 and computed as

$$D1 = \|[s_i\ ,\ \mathbf{o}_i] - [s_j\ ,\ \mathbf{h}'_j]\| \tag{6.21}$$

$$D2 = \|[e_i\ ,\ \mathbf{o}_i] - [s_j\ ,\ \mathbf{h}'_j]\| \tag{6.22}$$

$$D3 = \|[e_i\ ,\ \mathbf{o}_i] - [e_j\ ,\ \mathbf{h}'_j]\| \tag{6.23}$$

$$D4 = \|[s_i\ ,\ \mathbf{o}_i] - [e_j\ ,\ \mathbf{h}'_j]\| \tag{6.24}$$

$$d(o_i, h'_j) = \min([D1, D2, D3, D4]) \tag{6.25}$$

An appropriate distance function is needed because of the intrinsic differences between the time intervals of the hand and object POIs (see Figure 6.5). For example, the hand POI associated

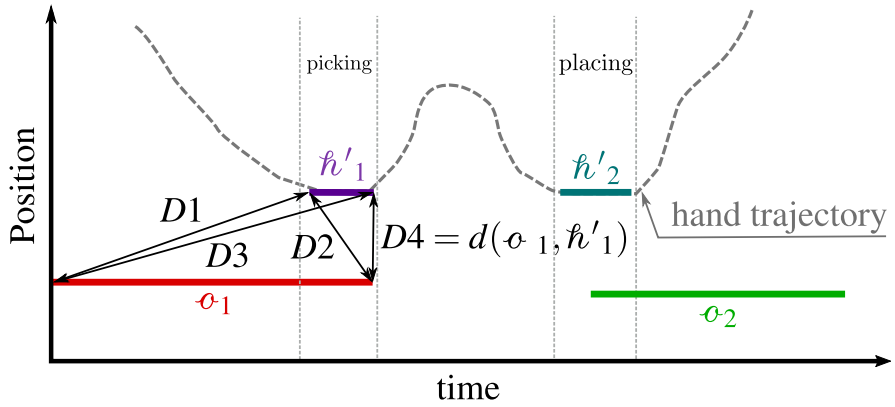

FIGURE 6.5: Illustration of the distance function d between the scaled POIs $\hbar'_1$ and o_1. The POIs are represented with only one spatial component.

to a pick action will have a time interval approximately as long as the action and centered on it. However, the object POI could have a time interval greater than the action. If an object has been stationary since the beginning of the demonstration, the POI time interval starts at the beginning of the task and ends with the pick action. In this case, the object POI start time is irrelevant for POI matching. Using the distance d and considering a scaled hand POI $\hbar'_j$, the value $d_{mean}(\hbar'_j)$ is computed as

$$d_{mean}(\hbar'_j) = \frac{1}{N^O}\sum_{i=1}^{N^O} d(o_i, \hbar'_j) \tag{6.26}$$

A hand POI whose d_{mean} value is higher than k_d is a candidate for elimination. Among all candidates, only the POI with the highest d_{mean} value is discarded in the current step of the iterative process. When no more hand POIs are candidates to elimination and if the two remaining subsets of hand and object POIs have the same size, the iterative process stops. Otherwise, the POIs elimination process continues. If $N^H > N^O$, the hand POI with the highest d_{mean} value is discarded (even if it is lower than k_d). If $N^O > N^H$, for each remaining object POI o_i, the d_{mean} value is computed similarly:

$$d_{mean}(o_i) = \frac{1}{N^H}\sum_{j=1}^{N^H} d(o_i, \hbar'_j). \tag{6.27}$$

The object POI with the highest d_{mean} value is discarded. Once the two subsets of POIs have the same dimension, the process stops. The complete process of discarding POIs is detailed in Algorithm 3.

Algorithm 3 : Discarding outliers POIs

N^H :=Number of POIs in $\mathscr{H}$
N^O :=Number of POIs in $\mathscr{O}$
while $N^H \neq N^O$ **or** $D_h > k_d$ **do**
  $T_{HO} = [R_z(\theta = 0), \mathbf{T} = \frac{1}{N_0}\sum_{i=1}^{N_O}\mathbf{O}_i - \frac{1}{N^H}\sum_{i=1}^{N^H}\mathbf{H}_i]$
  Optimization of T_{HO} by minimizing the function $COST$
  for $j \leftarrow 1$ to N^H **do**
    $\mathbf{H}'_j = T_{HO}(\mathbf{H}_j)$
  end for
  Scaling $\mathscr{O}$ to $\mathscr{o}$
  Scaling $\mathscr{H}'$ to $\mathscr{h}'$
  $[D_h, I_h] = \max([d_{mean}(\mathscr{h}'_1), \ldots, d_{mean}(\mathscr{h}'_{N^H})])$
  $[D_o, I_o] = \max([d_{mean}(\mathscr{o}_1), \ldots, d_{mean}(\mathscr{o}_{N^O})])$
  if $D_h > k_d$ **then**
    $\mathscr{H}_{I_h}$ is discarded from the set of hand POIs $\mathscr{H}$
  else
    if $N^H > N^O$ **then**
      $\mathscr{H}_{I_h}$ is discarded from the set of hand POIs $\mathscr{H}$
    else if $N^O > N^H$ **then**
      $\mathscr{O}_{I_o}$ is discarded from the set of object POIs $\mathscr{O}$
    end if
  end if
  N^H :=Number of POIs in $\mathscr{H}$
  N^O :=Number of POIs in $\mathscr{O}$
end while

Only 1 parameter, the threshold k_d, intervene in the identification of the subsets to pair. It can be tuned depending on the quality of the hand trajectory. If the hand trajectory presents a low accuracy, k_d has to be high enough to avoid rejecting some desired POIs. Conversely, the more accurate the hand trajectory, the closer the hand POIs are to the object POIs and the lower k_d can be.

6.3.2 Assignment of hand and objects POIs

The previous step identifies the two subsets of hand and object POIs which has to be paired. These subsets are noted $\mathscr{H}$ and $\mathscr{O}$ and contain the same number of POIs: $N^H = N^O$. The next problem is an assignment problem. In this work, the Hungarian method [76] is used to solve it. The method requires a cost matrix. The cost matrix M of size $N^H \times N^O$ (or $N^P \times N^P$ where N^p is the number of pairs) represents the temporal distance between POIs and is computed as

$$M_{i,j} = d^t(\mathscr{H}_i, \mathscr{O}_j). \tag{6.28}$$

with d^t the temporal distance function between 2 POIs

$$D1 = \|s_i - s_j\| \tag{6.29}$$
$$D2 = \|e_i - s_j\| \tag{6.30}$$
$$D3 = \|e_i - e_j\| \tag{6.31}$$
$$D4 = \|s_i - e_j\| \tag{6.32}$$
$$d^t(\mathcal{H}_i, \mathcal{O}_j) = \min(D1, D2, D3, D4). \tag{6.33}$$

At the end of this procedure, a set of pairs $\mathfrak{P}$ has been formed. This set of pairs can then be used to correct the hand trajectory, as explained in the next section.

6.4 Correcting trajectory

The correction of the hand trajectory has two steps: a global correction of the trajectory to partially catch up the error on the position of the human with respect to the robot, and a local correction to catch up the errors due to the modelling errors and the calibration errors. The global correction is applied on the whole trajectory. The local correction is applied at the specific moments of the trajectory corresponding to the actions as picking or placing.

6.4.1 The global correction

The global correction consists in applying a transformation to the hand trajectory (see Figure 6.6). The transformation is represented by a 3D rotation R and a 3D translation $\mathbf{T}$ and is estimated through the minimization of the following function e

$$e(R, \mathbf{T}) = \sum_{k=1}^{N^P} \|\mathbf{O}_k - (R\,\mathbf{H}_k + \mathbf{T})\|^2 \tag{6.34}$$

with N^P the number of pairs (let us recall that, at this step, $N^P = N^O = N^H$). Then the transformation is applied to the complete human hand trajectory. Then, a local correction of the trajectory is applied.

6.4.2 The local correction

This step consists in the local modification of the hand trajectory at and around the paired POIs. The aim is to make the hand trajectory fit the object positions measured using the vision-based system (see Figure 6.7).

The IHMT hand trajectory Γ is modified according to the pairs of POIs $\mathfrak{P}$. For each pair $\mathfrak{P}_i = [\mathcal{H}_j, \mathcal{O}_k]$, the 3D positions of the trajectory Γ between the temporal components s_j and e_j of the hand POI $\mathcal{H}_j$ is replaced by the spatial part $\mathbf{O}_k$ of the object POI $\mathcal{O}_k$.

In a second step, the hand trajectory is smoothed around each POIs. Let us consider that Γ_k is the part of the trajectory from the object POI $\mathcal{O}_a$ belonging to the pair $\mathfrak{P}_c = [\mathcal{H}_b, \mathcal{O}_a]$ and which

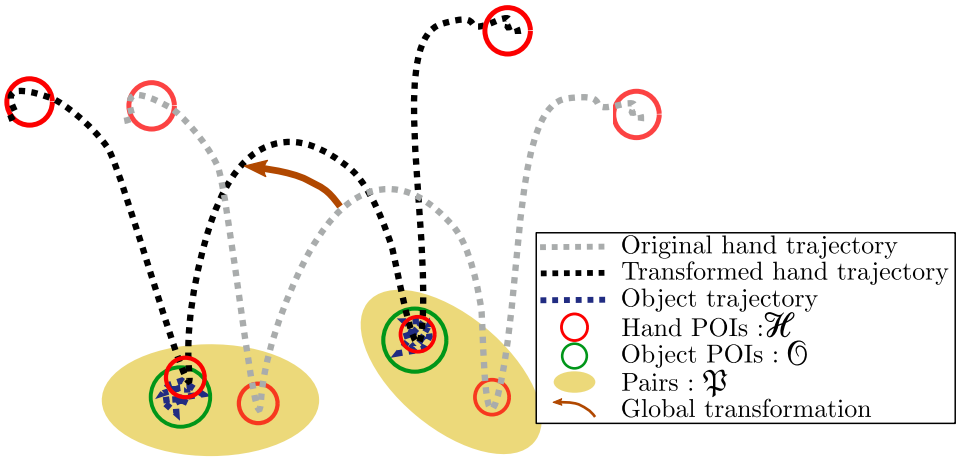

FIGURE 6.6: Illustration of the global correction of the hand trajectory based on pairs of POIs.

finishes at the next object POI $\mathscr{O}_u$ from the pair $\mathfrak{P}_w = [\mathscr{H}_v, \mathscr{O}_u]$. Γ_k contains N 3D points between $\mathbf{O}_a$ and $\mathbf{O}_u$:

$$\Gamma_k = [(e_b\,\mathbf{O}_a)\,,\,(t_1\,\mathbf{P}_1)\,,\,(t_2\,\mathbf{P}_2)\,,\,\ldots\,,\,(t_{N-1}\,\mathbf{P}_{N-1})\,,\,(t_N\mathbf{P}_N)\,,\,(s_v\,\mathbf{O}_u)]^T. \tag{6.35}$$

To avoid abrupt motions between $\mathbf{O}_a$ and $\mathbf{P}_1$, and between $\mathbf{P}_n$ and $\mathbf{O}_u$, f and g points are discarded in Γ_k:

$$\Gamma_k^* = [(e_b\,\mathbf{O}_a)\,,\,(t_{f+1}\,\mathbf{P}_{f+1})\,,\,(t_{f+2}\,\mathbf{P}_{f+2})\,,\,\ldots\,,\,(t_{n-g-1}\,\mathbf{P}_{n-g-1})\,,\,(t_{n-g}\,\mathbf{P}_{n-g})\,,\,(s_v\,\mathbf{O}_u)]^T. \tag{6.36}$$

The values for f and g are proportional to N and computed as

$$f = \lfloor k_f \times N \rfloor \tag{6.37}$$

$$g = \lfloor k_g \times N \rfloor \tag{6.38}$$

where k_f and k_g are two coefficients lower than one. The two coefficients k_f and k_g are selected based on the expected local corrections, e.g., for a high difference between the spatial parts of the hand POIs and the object POIs, a high value for these parameters would be selected. Finally, an interpolation is made on the motion part Γ_k^* to create a smooth trajectory Γ_k^{smooth}. The locally corrected trajectory Γ^{Cor} is then the sequence of the interpolated motion part $\Gamma^{Cor} = [\Gamma_1^{smooth}\,,\,\ldots\,,\,\Gamma_k^{smooth}\,,\,\ldots\,,\,\Gamma_K^{smooth}]$.

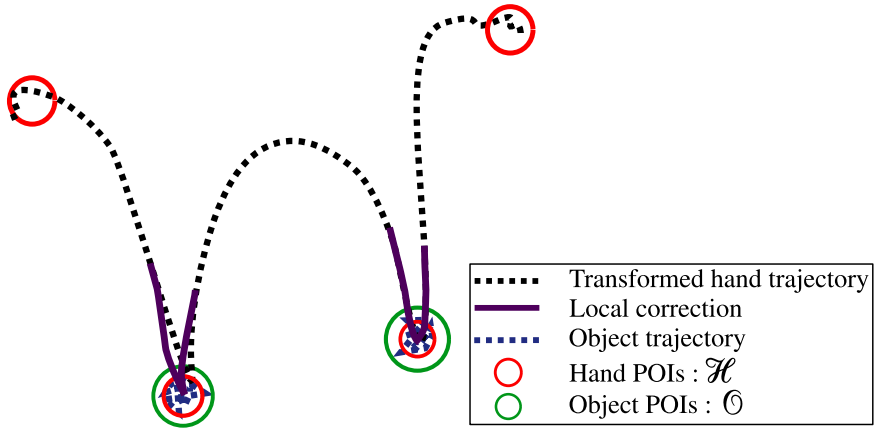

FIGURE 6.7: Illustration of the local correction of the hand trajectory.

6.5 Summary

In the context of pick-and-place tasks, this chapter proposes a method to correct the estimation of the hand trajectory from the proposed IHMT method thanks to vision-based measurements of the manipulated objects. The process relies on positions of interest (POIs). A POI is a 5-component vector: 3 spatial components for a 3D position and 2 temporal components for the beginning and end of the POI. The method presents 3 steps: extracting POIs, pairing POIs and correcting the hand trajectory.

Extracting POIs in both inertial and vision datasets is performed using the density-based clustering algorithm DBSCAN. Since the vision and inertial sensor data have different characteristics, the application of the DBSCAN algorithm is different for each dataset. The pairing step starts with an iterative process to discard POIs in both sets of object and hand POIs that should not be involved in any pair. The second step is then an assignment problem solved by the Hungarian method. Finally, the hand trajectory is corrected, first globally, then locally. The global correction aims at catching up the error in the positioning of the human with respect to the robot. A transformation is computed from the pairs and applied to the hand trajectory. Then the hand trajectory is locally corrected to fit to the position of the objects measured by the vision system.

The next chapter aims at evaluating the correction method presented here. It can be recalled (see Section 1.1.2) that a demonstration is valid if it can be reproduced by the learner. Therefore, one criterion to validate the correction method is the success rate of the robotic system to re-execute the demonstration.

CHAPTER 7

EXPERIMENTAL EVALUATION OF THE CORRECTION METHOD

Contents

This chapter presents the implementation and the assessment of the correction method presented in the previous chapter. The method corrects the IHMT estimation of the hand trajectory with object positions measured from vision sensors.

7.1 Experiment on robot imposed trajectories

In this experiment, the method is applied on data already recorded for the experiment described in Section 5.3.2. The human hand is guided by a robot undergoing imposed trajectories (circular, 3D cross-shape, cube edges or hexagonal shape). The end-effector position estimated from robot encoders measurement is considered as the environment observation and thus replaces the vision-based estimation of the object positions. In this way, the measurement uncertainties of a vision system are avoided. This experiment aims at a first quantification of the benefits of the method by comparison with the previous results from Chapter 5.

7.1.1 Experimental setup

The reader is referred to Section 5.3 for a description of the experimental protocol. Some pauses are present during the imposed trajectories which should be detected as POIs in both the IHMT trajectory and the reference trajectory, making the application of the correction method possible. Any object detection and positioning system has a certain level of inaccuracy that would contribute to the error on the corrected trajectory. Using the robot's trajectory as a reference and as an "objects observation" avoids the potential measurement errors of the vision system. The accuracy of the corrected hand trajectory can be quantified and the benefit of the correction method can be evaluated. It is recalled that the mean error values of the uncorrected hand trajectory with respect to the robot base frame range from 28.5 mm to 61.8 mm.

Compared to Section 5.3, the sampling rate of the reference trajectory is reduced by a factor of 10 for the present study. In this way, the resulting "objects observation" better match the characteristics of a vision-based system.

The parameters of the correction method are defined as $\varepsilon = 0.02$ and E =[0.010, 0.012, 0.014, 0.016, 0.018, 0.020, 0.022, 0.024, 0.026, 0.028, 0.030]; $MinPts_{h1} = 10$ and $MinPts_{h2} = MinPts_o = 2$. The parameter t_{limit} is tuned to 0.5 s according to the method. The parameter d_{limit} is chosen equal to 50 mm which is in the error range of the hand trajectory estimation. The optimization problem is solved using the *scipy.optimize* library. The parameter k_d is equal to 1, and k_f and k_g are both equal to 0.1.

7.1.2 Results and discussion

The mean error on the corrected trajectory is computed as

$$\varepsilon = \frac{1}{N}\sum_{k=1}^{N} \| {}^{R}\mathbf{OP}_k^{cor} - {}^{R}\mathbf{OP}_k^{robot} \| \tag{7.1}$$

with R the robot base frame, ${}^{R}\mathbf{OP}_k^{cor}$ the 3D positions of the corrected IHMT hand trajectory and ${}^{R}\mathbf{OP}_k^{robot}$ the reference 3D positions recorded by the robot's encoders. The results are presented

in Table 7.1 and Figure 7.1 shows the corrected trajectory for the “Cube 3” trajectory.

The circular trajectory should not be taken into account because no pairs of POIs were extracted (see the last 3 columns of Table 7.1). This is because no pauses are observed during these trajectories unlike the other trajectories.

If the circular trajectory is thus excluded, Table 7.1 shows that the corrected trajectory has a mean error between 11.9 mm and 20.8 mm which corresponds to an improvement between 58 % and 66 % with respect to the IHMT trajectory. Furthermore, the minimum level of error is brought to 0.0 mm, which means that at some places of the trajectory, the corrected trajectory perfectly matches the reference, as expected at the POIs. For 7 cases over 9, the maximum error is smaller after the correction meaning that the global correction has improved the whole trajectory.

It can be mentioned that the mean error here is lower than the mean error estimated when tracking the motion of a robot arm (see Section 5.1.2). During this experiment, the robot was considered to be performing an undisturbed motion and the error on the trajectory was measured between 27.7 mm and 47.7 mm and was attributed to the sensor orientation error. This shows that the correction method is able to catch up not only the modelling errors and the calibration errors but also the sensor orientation errors.

Trajectories	IHMT hand trajectory error (mm)			Corrected trajectory error (mm)			POIs in the reference	POIs in the hand trajectory	Pairs of POIs
	max	mean	min	max	mean	min			
Circle 1	82.6	46.5	9.3	82.6	46.5	9.3	1	0	0
Circle 2	110.6	41.1	12.9	80.0	41.1	12.9	0	0	0
Circle 3	106.4	49.0	10.8	106.4	49.0	10.9	0	0	0
Cross 1	99.8	53.1	3.0	86.5	19.8	0.0	9	9	9
Cross 2	96.7	83.9	8.0	101.4	20.5	0.0	9	9	9
Cross 3	95.9	44.5	12.2	74.4	19.0	0.0	9	9	9
Cube 1	148.9	76.1	5.1	124.1	20.0	0.0	8	8	8
Cube 2	140.2	60.8	11.2	114.3	17.7	0.0	8	8	8
Cube 3	128.5	51.4	5.0	186.7	20.8	0.0	8	8	8
Hexagone 1	139.0	52.8	15.0	132.6	11.9	0.0	7	7	7
Hexagone 2	137.5	61.9	23.4	112.15	14.5	0.0	7	7	7
Hexagone 3	124.0	48.3	14.2	118.6	14.3	0.0	7	7	7

TABLE 7.1: Mean errors on the IHMT hand trajectory before and after correction.

In the following sections, the proposed method is no longer evaluated in terms of accuracy but in terms of success in the robot’s execution of the demonstrated task, which is a more relevant criterion when evaluating a demonstration method for PbD.

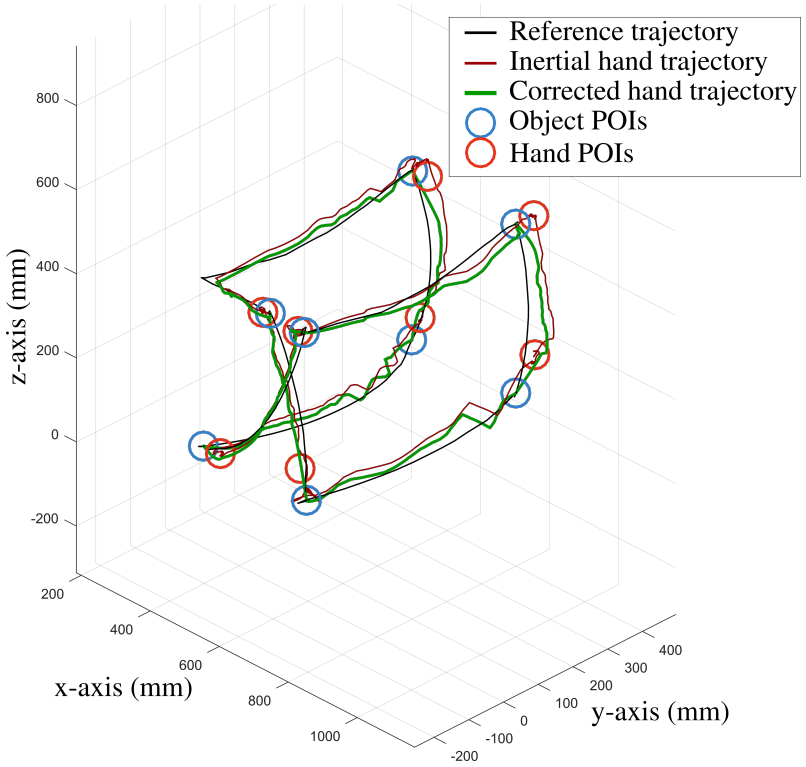

FIGURE 7.1: 3D representation of the hand trajectory before and after the application of the proposed correcting method for the trajectory "Cube 3".

7.2 Experiment on pick and place tasks

This experiment is conducted to evaluate the correction method in the context of pick-and-place tasks targeted in this work. The correction method is now applied with real vision data. The objects are equipped with ArUco markers to facilitate the estimation of their position. After the execution of the task by the human, the manipulated objects are returned to their original position. If the robot successfully reproduces the demonstrated task, the measured human trajectory is considered sufficiently accurate.

7.2.1 Experimental setup

The equipment

The method is applied to a Sawyer robot from Rethink Robotic. A gun-shape tool is used to reduce potentially disturbing fingers motion (see Figure 7.2 a). This tool includes an IMU for hand orientation tracking and a button, activated by a trigger, to communicate with the robot during the initialization procedure for inertial human motion tracking. The trigger also controls a pinch at the tip of the tool to manipulate the objects during the demonstration. With the trigger pressing the integrated button, the pinch states corresponding to the human hand states are recorded. The robot end-effector is a 2-fingers gripper (see Figure 7.2 b) with an opening going from 26 mm to 70 mm. The IMUs used in this work are the Xsens MTw Awinda sensors [15].

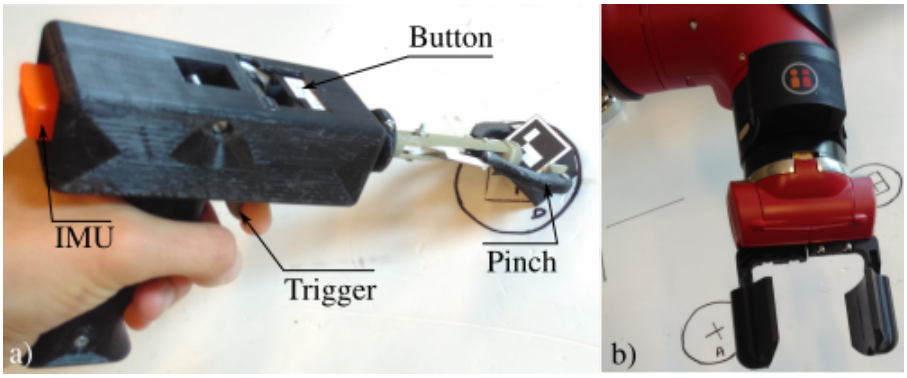

FIGURE 7.2: Experimental setup: a) gun-shape tool, b) robot end-effector

The objects are 30 mm side cubes with an ArUco marker on the top side. The position of the object is tracked based on the camera images using the library *openCV* [56]. The camera used in this experiment is the black and white camera embedded in the wrist of the Sawyer robot. Its resolution is 752 x 480 pixels. The camera inner parameters are directly given by the robot software as well as the position of the camera with respect to the robot base frame. The positions of the objects are first evaluated with respect to the camera frame and then easily transformed with respect to the robot base frame.

The vision system accuracy

The manipulated objects being 30 mm wide and the open gripper being 70 mm wide. Such configuration may tolerate up to a 20 mm inaccuracy on the positioning of the object. The accuracy of the object positions estimation has been estimated. Two objects (A and B) placed on the table are observed by the camera during approximately 60 seconds. The two sets p^A and p^B of N measured positions, for objects A and B respectively, are recorded. The reference distance d_{AB}^{ref} between the objects is measured with a ruler. The distance between the objects estimated from the vision system is computed as

$$d_{AB} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i^A - \mathbf{p}_i^B\| \tag{7.2}$$

and the standard deviation of each object position dataset is also computed:

$$\sigma_A = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i^A - \bar{\mathbf{p}}^A\|^2} \tag{7.3}$$

$$\sigma_B = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i^B - \bar{\mathbf{p}}^B\|^2} \tag{7.4}$$

with $\bar{\mathbf{p}}^A$ and $\bar{\mathbf{p}}^B$ the mean of the sets $\mathbf{p}^A$ and $\mathbf{p}^B$. The error ε_{AB} of the distance between the 2 objects is computed as

$$\varepsilon_{AB} = \frac{|d_{AB} - d_{AB}^{ref}|}{d_{AB}^{ref}} \,. \tag{7.5}$$

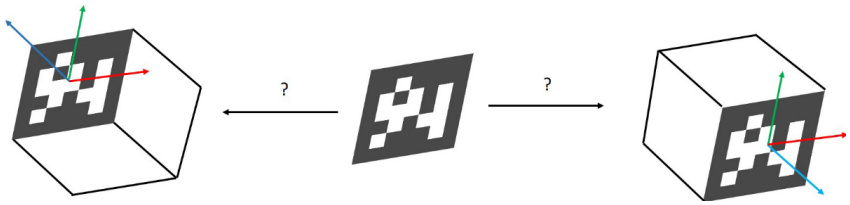

FIGURE 7.3: Illustration of the ArUco pose ambiguity with perspective projection [39].

The objects are arranged on the table in 22 different configurations with distances ranging from 50 mm to 230 mm and covering as much of the observable area as possible (approximately a 800 mm side square). Table 7.2 shows the error and the standard deviation. The maximum value of the error ε_{AB} is 18.8 mm. As the columns σ_A and σ_B reveal, the estimation of the object positions is noisy. A standard deviation between 3.6 mm and 23.1 mm is observed. This is mainly due to a pose ambiguity with perspective projection of the ArUco markers [116] as illustrated in Figure 7.3. This analysis shows that the object positioning algorithms are also expected to introduce some errors, which are nevertheless reasonably small for the targeted application.

d_{AB}^{ref} (mm)	σ_A (mm)	σ_B (mm)	d_{AB}	ε_{AB} (mm)	ε_{AB} (%)
200.0	5.2	3.6	206.0	6.0	3.0
200.0	10.1	10.1	205.5	5.5	2.7
200.0	4.8	4.5	204.4	4.4	2.2
230.0	9.8	9.0	241.4	11.4	5.0
230.0	12.1	13.8	241.6	11.6	5.0
150.0	15.9	13.7	144.6	5.4	3.6
150.0	18.8	16.8	140.1	9.9	6.6
150.0	21.1	19.6	137.7	12.3	8.2
150.0	19.3	19.3	135.1	14.9	9.9
100.0	15.7	17.4	106.4	6.4	6.4
100.0	16.5	16.5	107.0	7.0	7.0
50.0	19.0	18.4	49.6	0.4	0.7
50.0	10.9	9.9	49.9	0.1	0.2
50.0	10.8	10.3	44.8	5.2	10.4
50.0	6.8	5.8	44.4	5.6	11.2
50.0	16.3	16.6	44.9	5.1	10.2
50.0	21.2	19.3	45.7	4.3	8.7
230.0	16.8	17.2	234.2	4.2	1.8
230.0	19.2	19.7	234.4	4.4	1.9
230.0	19.5	20.0	234.6	4.6	2.0
140.0	19.2	23.4	158.8	18.8	13.4
140.0	23.1	22.3	142.6	2.6	1.9

TABLE 7.2: Error on the object positions estimation method.

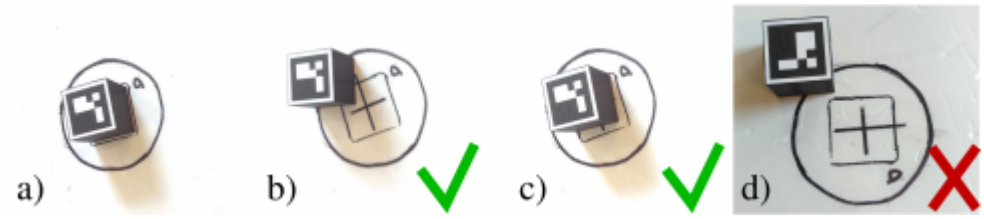

FIGURE 7.4: Criterion for the success of the task: a) targeted position, b) success, c) success, d) fail.

The tasks

The tasks executed by the operator consists in moving one or more objects one or more times in the observable area of the camera. After the demonstration of the task by the operator, the objects are returned to their original positions marked on the table. Then the robot executes the demonstrated task. The task is considered successful if the footprint of the objects (30 mm side square) covers a part of their targeted final position, as illustrated in Figure 7.4. Such a tolerance is reasonable considering to the level of accuracy of the vision-based estimation of the object positions (see Table 7.2). The different tasks are listed in Table 7.3. The last column gives the expected number of pairs or, in other words, the number of POIs corresponding to the relevant actions. These values are deduced from the actions performed during the demonstration. For instance, the first task consists in moving one object from a place to another. Thus, the task contains one "pick" and one "place" which is expected to lead to 2 pairs.

Task number	Tasks properties			Expected number of pairs
	objects on table	objects moved once	objects moved twice	
1	1	1	0	2
2	2	1	0	2
3	2	2	0	4
4	3	0	1	4
5	3	2	0	4
6	3	2	1	8
7	4	2	0	4
8	4	4	0	8
9	4	3	1	10
10	4	2	2	12

TABLE 7.3: Tasks properties.

The parameter values are similar to those presented in the previous Section 7.1.1: $\varepsilon = 0.02$; E =[0.010, 0.012, 0.014, 0.016, 0.018, 0.020, 0.022, 0.024, 0.026, 0.028, 0.030]; $MinPts_{h1} = 10$; $MinPts_{h2} = MinPts_o = 2$; $t_{limit} = 0.5$ s. Only the parameter d_{limit} is increased to 55 mm in order to absorb the error from object positioning, and the two gains k_f and k_g are both chosen equal to 0.30.

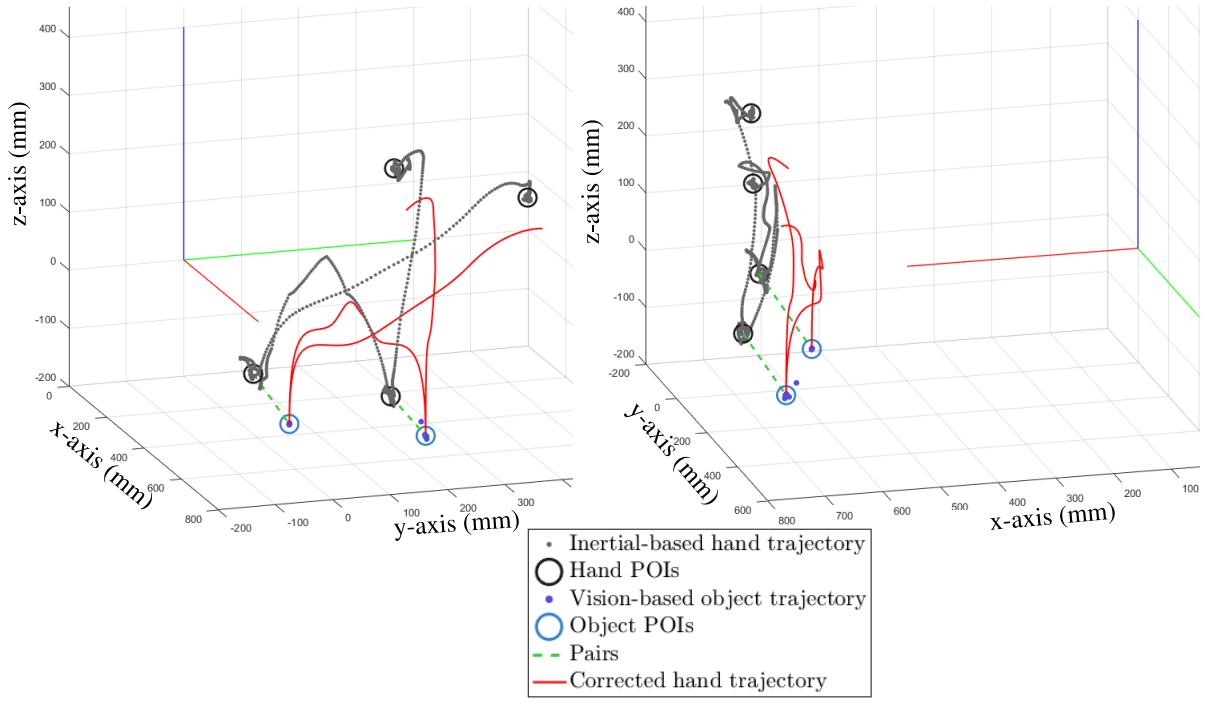

FIGURE 7.5: Hand and object trajectories for task 1.

7.2.2 Results and discussion

The results are presented in Table 7.4. The amount of extracted hand and object POIs is given as well as the number of pairs. A "1" in the column "$\mathscr{H}$" and "$\mathfrak{O}$" means that no necessary POI is missing. A necessary POI corresponds to a relevant action for the task (picking, placing, etc.) that should be corrected. The column "$\mathfrak{P}$" represents the success or failure of the pairing. If a failure was observed in the columns "$\mathscr{H}$" or "$\mathfrak{O}$" for a given task, the pairing is necessarily wrong. The column "IHMT trajectory" represents the successes or failures of the robot to execute the task from the human trajectory obtained only from inertial sensors, thus uncorrected. The last column "Corrected trajectory" represents the same thing but using the corrected trajectory. Figure 7.5 shows the modified trajectory for the task number 1.

All executions of the task based on the non-corrected IHMT hand trajectory failed when almost all executions with the corrected trajectory succeeded. The correction is required to obtain an exploitable estimation of the human motion.

With the corrected trajectory, all tasks have been successfully reproduced by the robot, with the exception of task 8. For this task, a hand POI corresponding to the second picking action was not extracted. Figure 7.6 illustrates this issue. It shows the 2D representation of the hand trajectory on the first part a). The second part b) presents the POIs initially extracted by the DBSCAN algorithm. The lower part c) shows the remaining POIs after the post-treatment. The z-component of the hand trajectory between 17 and 18 seconds is low which indicates a picking action (motion towards the table). However, no POIs has been created as can be seen in part b) and c). The non-creation of the POIs at this moment may come from a perturbation in the trajectory: the z-component and the y-component are not steady (see ellipses in Figure 7.6 a)). This

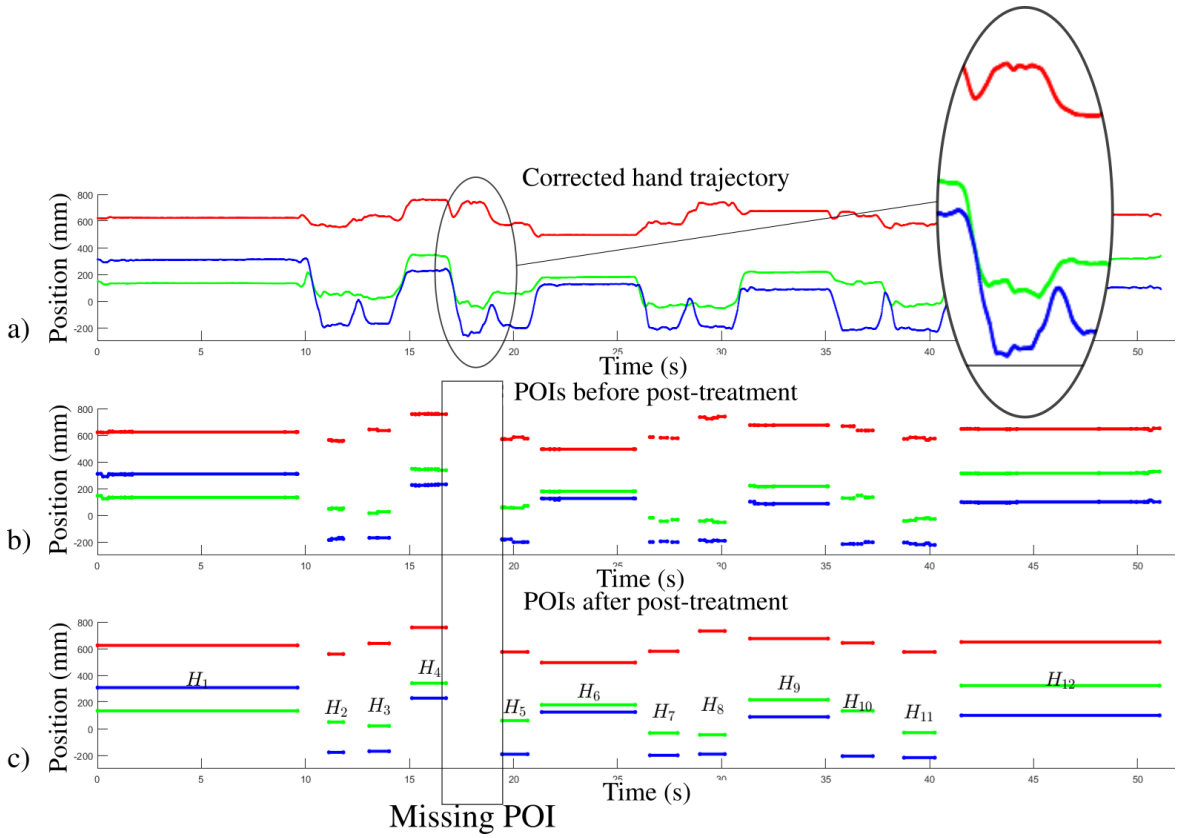

FIGURE 7.6: a) 2D representation of the hand trajectory for the task 8, b) the extracted POIs before post-treatment and c) after post-treatment. Focus on the second picking action (ellipses in a)) and the corresponding missing POI (in b) and c)).

parasitic motion may come from a displacement of the human hand or may be artificially introduced by the IHMT method. By decreasing the parameter of the DBSCAN algorithm $MinPts_{h1}$ to 8 (instead of 10), fewer data points are required to consider a cluster, and then a POI for this picking action is created, leading to a successful execution of the task. This phenomenon reveals a potential lack of robustness to parasitic motions during the demonstration.

Task number	N_H	N_O	N_P	$\mathscr{H}$	$\mathfrak{O}$	$\mathfrak{P}$	IHMT trajectory	Corrected trajectory
1	4	2	2	1	1	1	0	1
2	7	4	4	1	1	1	0	1
3	7	4	4	1	1	1	0	1
4	7	6	4	1	1	1	0	1
5	7	5	4	1	1	1	0	1
6	14	8	8	1	1	1	0	1
7	8	6	4	1	1	1	0	1
8	12	8	8	0	1	0	0	0
9	17	10	10	1	1	1	0	1
10	18	12	12	1	1	1	0	1

TABLE 7.4: Successes and failures of the execution of the demonstration by the robot.

7.3 Experiment with tool tracking

Pick-and-place tasks cover a wide range of applications and many of them involve a tool whose trajectory may be tracked by the vision system. Sometimes, the task requires to drop an object. In this case, the object tracked by a vision device is attached to the hand at the drop position. This drop position must be accurately estimated. It is therefore relevant to track the tool or object attached to the hand using the vision system. We propose an experiment where the tracked object is the gun-shape tool. In such applications, the data from the vision system represent the same motion as the data from the inertial sensors: the position of the tip of the gun-shape tool. The position of the tool from the vision system is more accurate but suffers from concealment and presents a lower frequency of acquisition. In such cases, the problem is to merge the inertial-based and the vision-based estimation of the same element. Some method has been proposed for this specific problem and their suitability for PbD could be investigated.

7.3.1 Experimental setup

The position of the tip of the gun-shape tool is estimated from the detection of 4 ArUco markers added on the tool (see Figure 7.7). Both the ArUco markers and the IHMT method are used to localize the ball tip of the gun. The hand segment ${}^{3n}\mathbf{CD}$ of the human arm model takes into account the geometry of the human hand holding the gun-shape tool. In case different tools are used for different demonstrations – for instance if one switches from a drill to a screwdriver between tasks – a discrepancy would occur between the tracked point by the IHMT method and the tracked point by the vision system. In this case, the hand segment geometry ${}^{3n}\mathbf{CD}$ can be automatically adapted by the optimization of the model parameters if the human-robot transformation estimation procedure is executed by the teacher holding the new tool. Furthermore, the correcting method has also the potential to catch up this discrepancy.

A ZED camera from Stereolabs is used to track the tool (see Figure 7.8). The resolution is set to its maximum: 2K, and the acquisition rate is 15 Hz. In order to estimate the position of the tool with respect to the robot base frame, the positioning of the ZED camera with respect to the robot base frame is necessary. In order to estimate it, before starting the recording of a

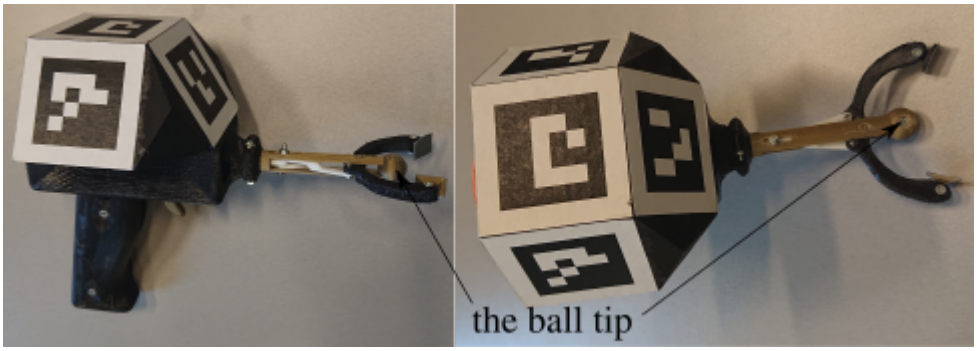

FIGURE 7.7: Gun-shape tool with ArUco marker for tracking.

task, both the ZED camera and the robot via its wrist camera observe a large (200 x 200 mm) ArUco marker placed on the table. Therefore, the transformation between the ZED camera and the large ArUco marker as well as the transformation between the robot base frame and the large ArUco marker are estimated. The transformation from the ZED camera to the robot base frame is computed from these transformations. The tool being tracked by ArUco markers, the error on its position is expected to be similar to the position of the cube evaluated in the previous experiment (see Section 7.2.1 Table 7.2). The manipulated objects are the 30 mm side cubes described in the previous section but without ArUco markers to avoid disturbing the tracking of the tool.

The parameter values of the method are the same as in the previous experiment: $\varepsilon = 0.02$; E =[0.010, 0.012, 0.014, 0.016, 0.018, 0.020, 0.022, 0.024, 0.026, 0.028, 0.030]; $MinPts_{h1} = 10$; $MinPts_{h2} = MinPts_o = 2$; $t_{limit} = 0.5$ and $d_{limit} = 55$ mm and the two gains $k_f = k_g = 0.30$.

The features of the demonstrated tasks are detailed in Table 7.5. As in the previous experiment, some objects are moved once or twice. The criterion for a successful task execution by the robot is similar to that of the previous experiment (see Figure 7.4). In the previous experiment, the non-manipulated cubes are detected by the vision system which can induce outliers POIs. In this experiment, the non-manipulated cubes are replaced by larger objects (cans, drill) as obstacles. Obstacle avoidance is then a new constraint to be fulfilled during the execution of the task by the robot. If an obstacle is moved, the execution of the task is failed. The different tasks are executed 3 times.

7.3.2 Results and discussion

The results are gathered in Table 7.6. The number of successes is again large in this context (tracking tool and obstacle avoidance) which further demonstrates the efficiency of the proposed method. Only, 3 executions failed, which are discussed below.

The 3rd attempt of task 4 failed due to a wrong set of object POIs. Figure 7.9 shows the POIs for the hand and the objects before and after the post-treatment. The POIs are represented as 3 horizontal lines. Each line represents one spatial component and the extremities of the lines are the temporal components. The element "Px" ("x" being a number) is noted at the location of a hand POI and an object POI to represent the x^{th} pair. As it can be seen on this figure, two temporally distinct object POIs are close spatially and merged as they should not be. The

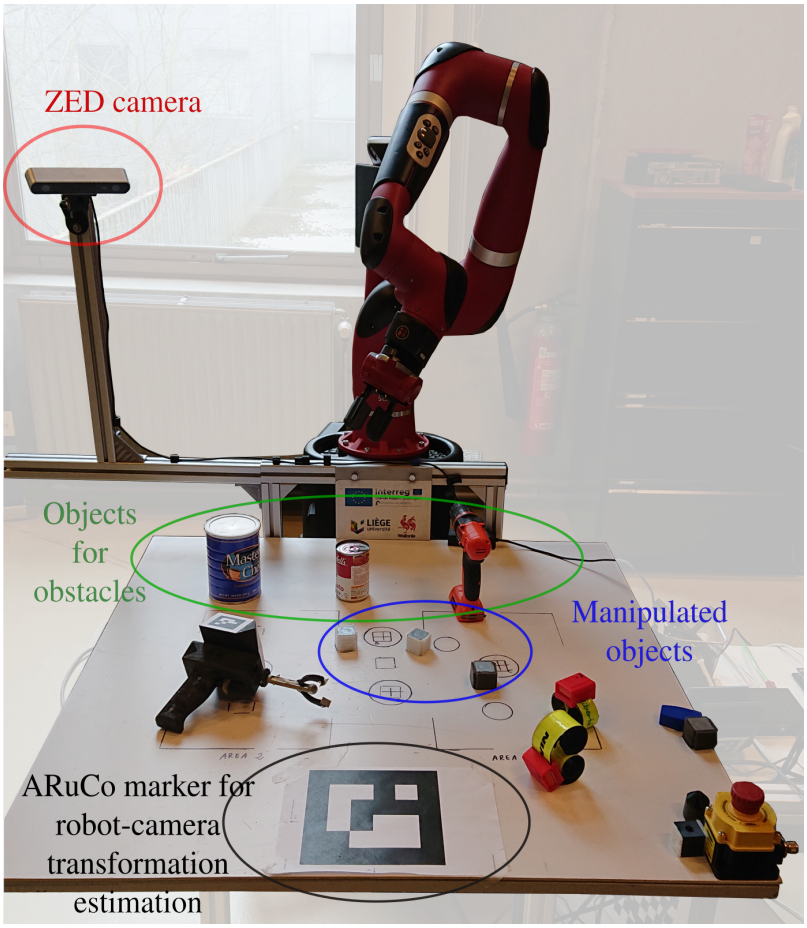

FIGURE 7.8: Setup for tool tracking experiment.

Task number	Tasks properties			Expected number of pairs
	objects as obstacles	objects moved once	objects moved twice	
1	0	1	0	2
2	1	1	0	2
3	0	2	0	4
4	2	0	1	4
5	1	2	0	4
6	0	2	1	6
7	2	2	0	4
8	0	4	0	8
9	0	3	1	10
10	0	2	2	12

TABLE 7.5: Tasks properties for pick-and-place tasks with tool tracking.

sequence of actions consists in placing an object (20-23 seconds), move the tool to an avoidance position above the table and hold that position (25-35 seconds), and pick up the same object at the same position (35-38 seconds). In this case, the avoidance position is such that the tool is not seen by the camera. As expected, one hand POI is created at the avoidance position but no object POI. Consequently, two consecutive object POIs present a very close spatial position (their distance is less than d_{limit} since the position is the same). These POIs are merged by the object post-treatment leading to the failure of the pairing.

The 2nd attempt of the task 7 failed because no object POI is formed. This comes from the failure of the DBSCAN algorithm to create clusters with the chosen parameters values. It can be seen in Figure 7.10 as the distance with the next point on the scaled data is never under 0.02 (no-unit), the value of the parameter ε_o. By modifying the value of ε_o to 0.024 (no-unit) the clustering method extracts 5 objects POIs, the pairing is successful as well as the execution of the task. An explanation could come from the noise on the measurement from the vision system. A high noise could create artificial distance between two positions of the gun tip which is higher than the parameter ε of the DBSCAN algorithm. However, it reveals a sensitivity of the method to this parameter.

The 2nd attempt of task 9 failed because of a missing hand POI as seen in Figure 7.11. The 10^{th} pair associates the wrong POIs. The 9^{th} hand and object POIs correspond to a placing action. Then a clearing movement with a pause in the middle is observed before picking up another object (pair P11). However, a disturbing motion during the placing action results in an unwanted hand POI at approximately 53 seconds. The pause during the clearing motion results in an object POI and no hand POI. The pairing method creates the 10^{th} pair by associating the hand POI corresponding to the disturbing motion with the object POI corresponding to the clearance position. If the 10^{th} pair is manually discarded, the robot executes the task successfully. The method fails to create a hand POI associated to the clearance position yet visible in the hand POIs before post-treatment (see first and second draw of Figure 7.11). It can also be considered that the method fails to discard the short unwanted POI resulting from the disturbed motion during

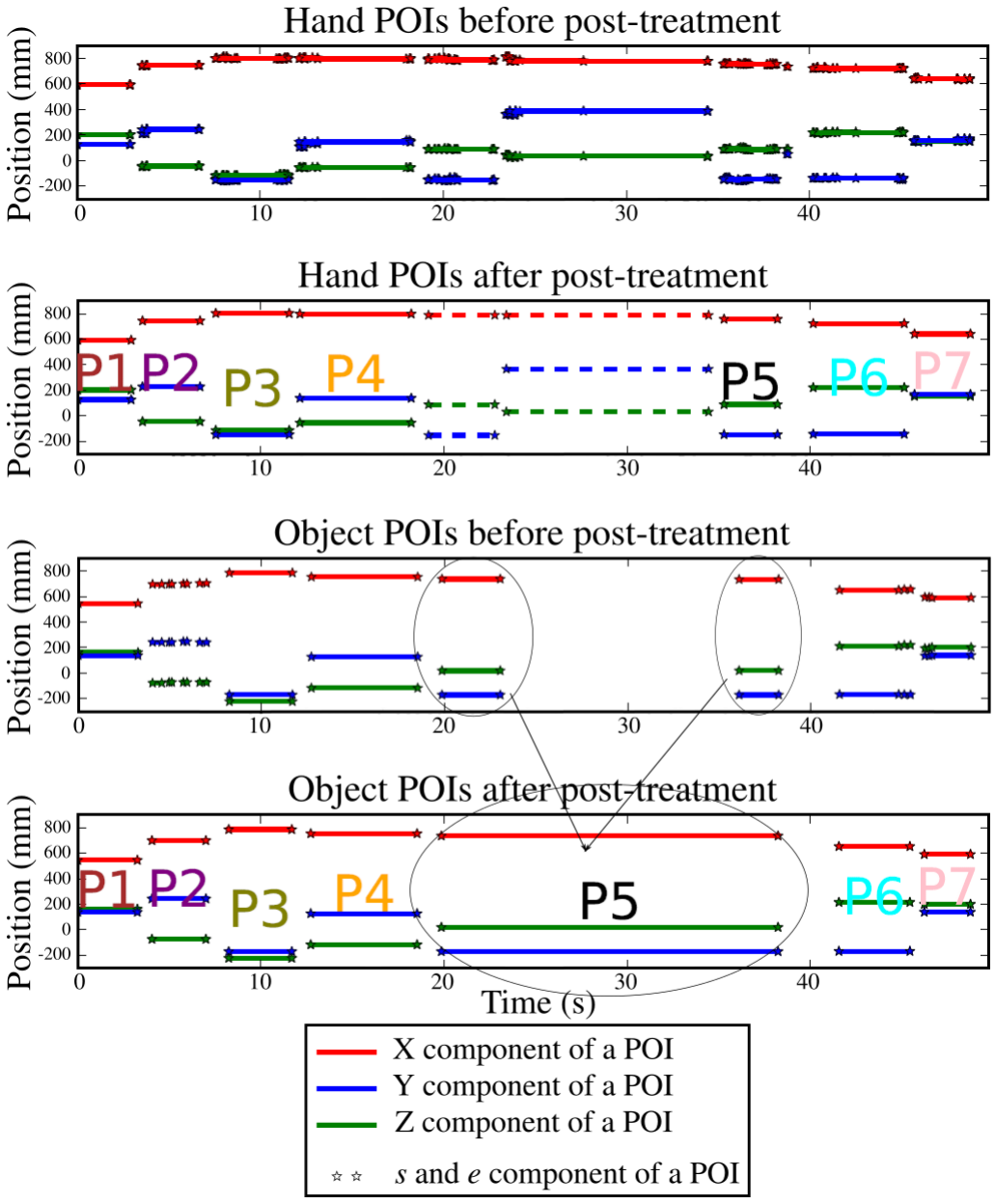

FIGURE 7.9: Representation of the POIs (before and after post-treatment) for the hand and the objects trajectories for attempt 3 of task 4. "Px" represent the pairs of POIs. In the bottom figures, the post-treatment merges 2 POIs mistakenly.

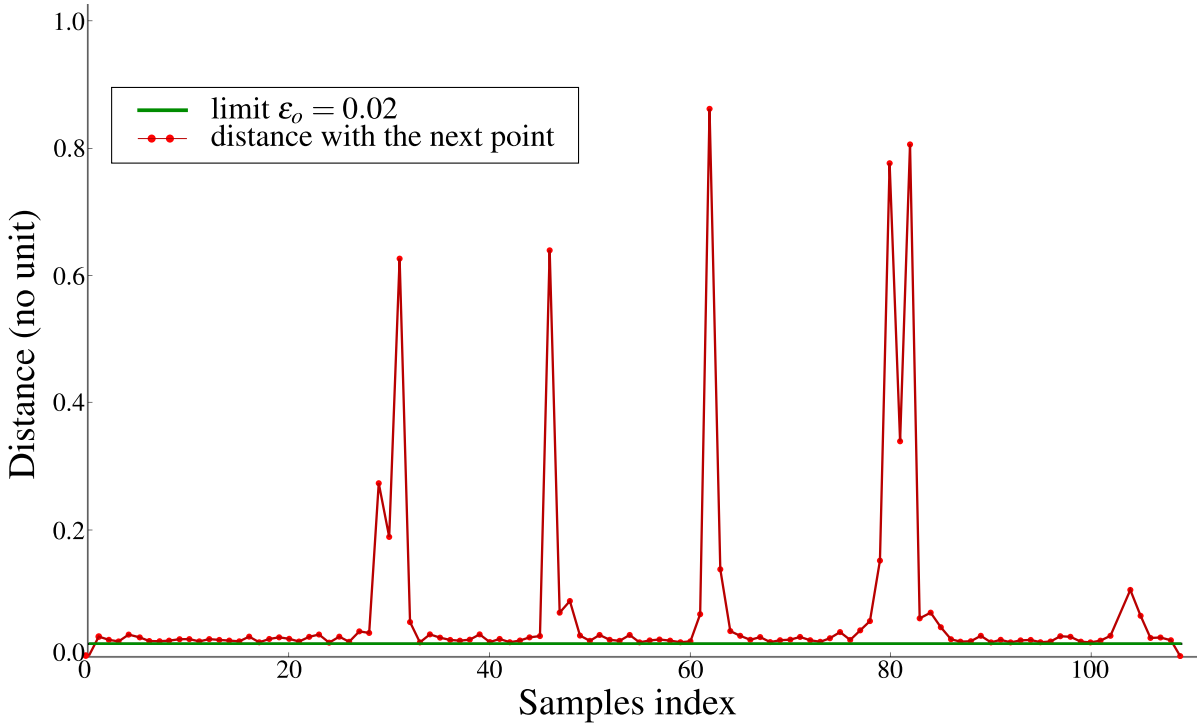

FIGURE 7.10: Distance to the nearest neighbour between data points of the scaled object trajectory for the 2nd attempt of task 7. The green line represents the threshold parameter ε_o of the DBSCAN algorithm.

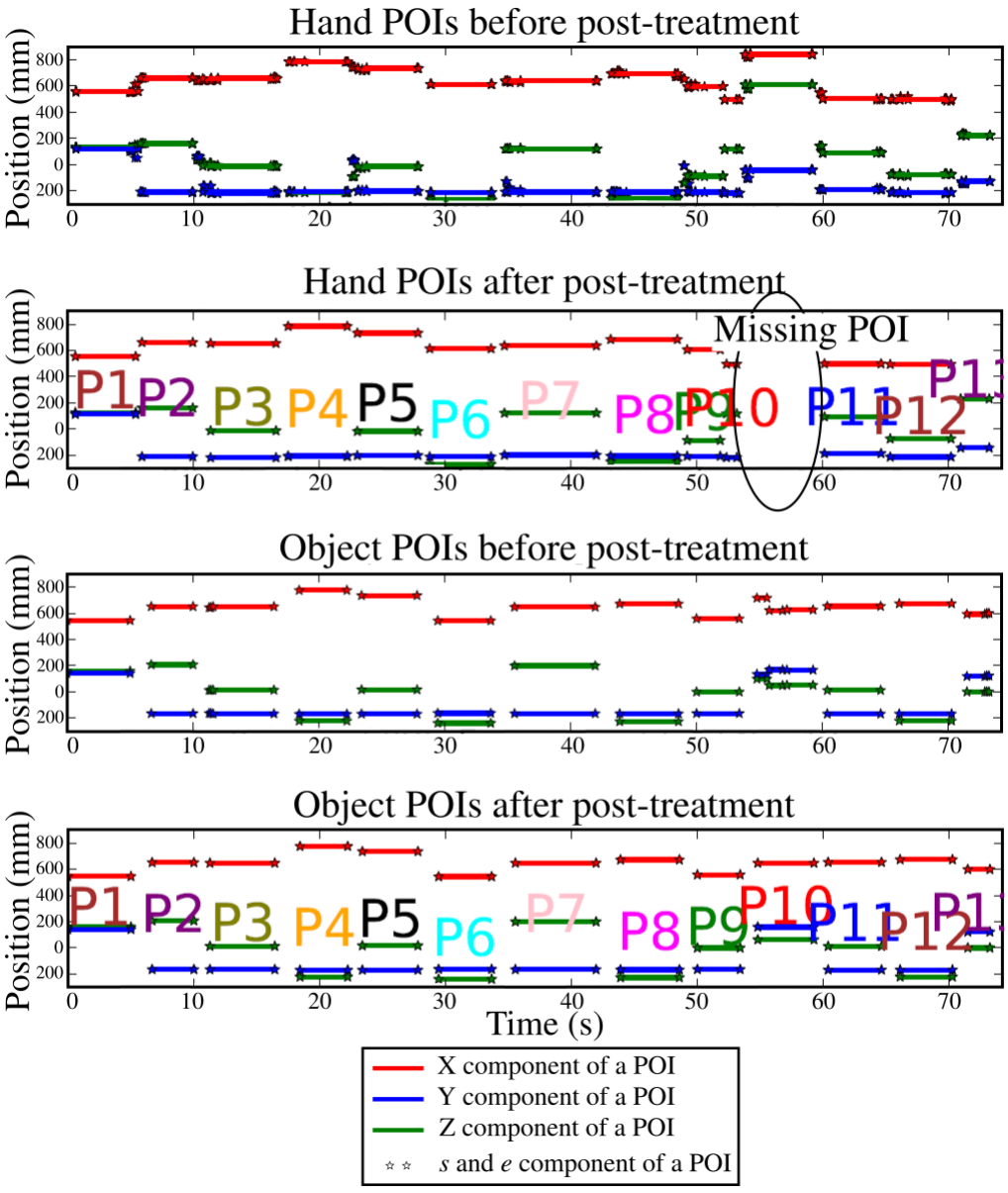

FIGURE 7.11: Representation of the POIs before and after post-treatment for the hand and the object POIs for attempt 2 of task 9. "Px" represent the pairs of POIs. The second figure shows the missing POI due to a post-treatment issue.

Task number	Attempt	N_H	N_O	N_P	success "1" or failure "0"			
					$\mathscr{H}$	$\mathfrak{O}$	$\mathfrak{P}$	task
1	1	4	4	4	1	1	1	1
	2	4	5	4	1	1	1	1
	3	4	4	4	1	1	1	1
2	1	4	4	4	1	1	1	1
	2	4	4	4	1	1	1	1
	3	4	3	3	1	1	1	1
3	1	6	6	6	1	1	1	1
	2	6	5	5	1	1	1	1
	3	6	6	6	1	1	1	1
4	1	8	8	8	1	1	1	1
	2	8	6	6	1	1	1	1
	3	9	7	7	1	0	0	0
5	1	6	5	5	1	1	1	1
	2	10	8	8	1	1	1	1
	3	6	6	6	1	1	1	1
6	1	11	11	11	1	1	1	1
	2	11	11	11	1	1	1	1
	3	11	11	11	1	1	1	1
7	1	6	6	6	1	1	1	1
	2	5	0	0	0	0	0	0
	3	8	8	8	1	1	1	1
8	1	10	10	10	1	1	1	1
	2	10	10	10	1	1	1	1
	3	14	11	11	1	1	1	1
9	1	15	13	13	1	1	1	1
	2	12	12	12	0	1	0	0
	3	14	11	11	1	1	1	1
10	1	17	17	17	1	1	1	1
	2	19	19	19	1	1	1	1
	3	16	16	16	1	1	1	1

TABLE 7.6: Successes and failures of the execution of the demonstrated tasks by the robot

the pairing step. The disturbing motion can be observed in Figure 7.12. This motion may come from an IMU sensor issue and not from a human disturbing motion since it is not present in the data from the vision device.

The analysis of the failures in this experiment reveals a lack of robustness in some special cases: two POIs can be merged by mistake, the method can be sensitive to the parameters values, and some POIs can be mistakenly eliminated. However, the success rate is significant enough to confirm the usefulness of the method in the context of tool tracking.

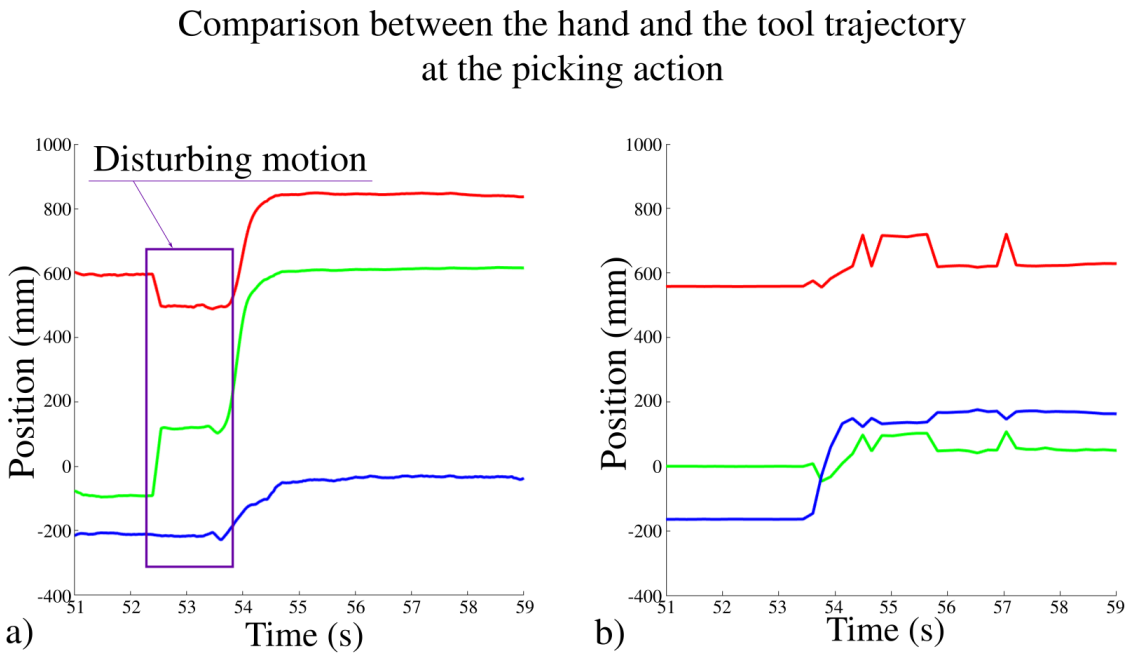

FIGURE 7.12: a) The IHMT trajectory and b) the vision-based tool trajectory at the picking action. Figure a) shows the disturbing motion (only present in the IHMT trajectory) in the IHMT trajectory that lead to the creation of an extra hand POI.

7.4 The correcting method and the actions

The correcting method successfully improves the accuracy of the IHMT method and belongs to the step of **gathering the dataset** in the PbD process (see the introduction Section 1.1.1). The correcting method also extracts some information about the actions of the task and thereby contributes to the second step of PbD called **learning the task**. This second step consists in "understanding" the task from the demonstrations. This includes the extraction of information about the actions from the dataset [11, 20, 21, 31]. Let us develop this idea.

What does it mean to extract the actions for a pick-and-place problem? A picking or placing action can be described as the specific position of the hand regarding an object state coupled with a change of the gripper state (closing or opening). A pair of POIs associates an object key position with a hand key position. It only misses the gripper change of state to fully describe a picking or placing action. Therefore, a pair of POIs can be considered as a partial representation of an action A_l.

Thus, the proposed correction method has a double impact: it makes the estimation of the hand trajectory precise enough to be used in the PbD process (as seen in this chapter) and it emphasizes the actions in the demonstration dataset. The correction of the hand trajectory forces the hand position to coincide with a specific object position. The quality of the dataset d_l is improved, i.e., the actions A_l are better encoded into the dataset d_l.

The proposed correction method is presented here as a part of the step of gathering the dataset. However, it can also be considered that this method extracts task features and can be considered

as part of the step of learning the task. In the next chapter, the proposed demonstration acquisition method is evaluated in a complete PbD process. For a greater robustness, the implementation of the learning the task algorithm will rely on the POIs when it is possible.

7.5 Summary

The approach for correcting the IHMT hand trajectory with vision-based object positioning consists in 3 steps : extracting POIs (positions of interest), pairing POIs and modifying the human trajectory. The POIs correspond to the stationary moment during the demonstration due to actions such as picking up and placing objects. This chapter presents the experimental evaluation of the proposed correction method.

First, the correction method is applied without any vision device. The objective is to evaluate the gain in accuracy of the corrected trajectory of the human hand. The robot imposes different trajectories to the human based on the contact interface between its end-effector and the gun-shape tool held by the human. The estimation of the robot trajectory through its encoders is used as reference and also as input data for the correction. This experiment shows that the correction method has a beneficial impact on the accuracy of the human hand trajectory estimation. The error is between 28.5 mm and 61.8 mm without any correction and between 11.9 mm and 20.8 mm after correction. It can be mentioned that the IHMT method applied on a robot arm performing undisturbed motions presents an error level between 27.7 mm and 47.7 mm (see Chapter 5). It can be concluded that the correction method catches up some part of the error on the hand trajectory not only due to the modelling errors and the calibration errors but also due to the sensor orientation errors.

Then, the method is tested with several pick-and-place tasks. The objects used are 30 mm side cubes. An ArUco marker is placed on each cube to facilitate their tracking. The camera used is a black and white camera integrated in the robot. The method has been tested with different pick-and-place tasks where one or several objects are manipulated once or several times. The robot was able to execute all but one of the demonstrated tasks, which demonstrates that the method is suitable in the context targeted in this work.

Finally, the method is exploited to track the gun-shape tool carried by the human. Indeed, some pick-and-place tasks may involve tools or dropping positions where the object to track is held by the human. For 27 demonstrations over 30, the robot was able to successfully reproduce the task. The 3 failures reveal that the method might be sensitive to some parameters values, and can mistakenly manage POIs and pairs in some cases.

The potential of this approach based on the complementarity of vision and inertial data was demonstrated. Furthermore, it was shown that the robot is able to reproduce the demonstrated task making the complete method suitable for PbD.

The correction method can be seen as a way to extract partial representation of actions in the demonstrations. It is then a step towards the second part of a PbD process: *learning the task*. The different steps of a PbD process (*gathering the dataset* and *learning the task*) can influence each other. The next chapter aims then at evaluating the proposed method in a complete PbD process.

CHAPTER 8

IMPLEMENTATION OF THE CORRECTED IHMT METHOD IN A PBD PROCESS

Contents

The objective of this chapter is to implement and evaluate the proposed corrected IHMT method in a complete PbD process, i.e., from the acquisition of demonstrations to the robot execution of the task in new configurations. Several methods have been proposed to solve the *learning the task* part and could have been considered here [8, 29, 34, 109]. The probabilistic approach developed by Calinon et al. [29] is selected and detailed in this chapter. Then the setup and the taught task are described. The same task is also taught using kinesthetic demonstrations. The results of the experiment are discussed and the proposed corrected IHMT method is compared to kinesthesy.

8.1 Learning the task: the probabilistic approach

The probabilistic approach developed by Calinon et al. [29] is selected. The authors test their method on tasks for the manipulation of objects that present similarities with the applications targeted in this work. This approach relies on several demonstrations encoded using a Gaussian Mixture Model (GMM). From the GMM, the task is then generalized using a Gaussian Mixture Regression (GMR). Then, a metric of imitation is determined to measure the performance of a behaviour to reproduce the task. It is based on the results of the GMR and the initial conditions of the new task configuration. From the optimization of the metric of imitation, a command is computed and sent to the robotic system in order to perform the task. The process is summarized in Figure 8.1. The implementation in this work of the probabilistic approach contains several differences from the original work. They are explained throughout the presentation of the method in this section. For a detailed explanation of the probabilistic approach, the reader may refer to [29].

The task studied in this chapter is to pick up a cube from the table and drop it into a jar. This task features a dropping position that has not yet been encountered in the previous experiments in this work.

8.1.1 Adapting the dataset

The first step of the probabilistic approach is to adapt the dataset used in the Gaussian process. Consider that Nd demonstrations d have been performed by the teacher. The k^{th} demonstration is denoted d^k with $k \in [1, Nd]$. Each demonstration d^k consists of Nt^k time steps t_i^k with $i \in [1, Nt^k]$, with $k \in [1, Nd]$. Nt^k may vary with k since each demonstration does not last the same amount of time. During the demonstrations, for each time step t_i, Nv variables Vj with $j \in [1, Nv]$ are recorded. Each variable contains one or more components. For instance, if Vj is a position, it contains 3 components noted Vj_x, Vj_y and Vj_z. In the following, the matrix Vj designates

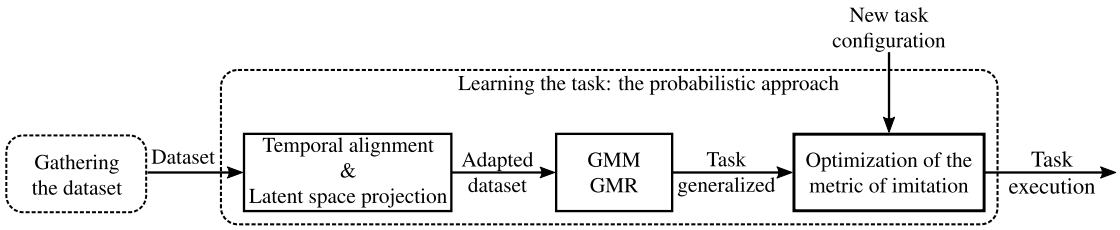

FIGURE 8.1: Graphical summary of the probabilistic approach [29] for the step of learning the task in a PbD process.

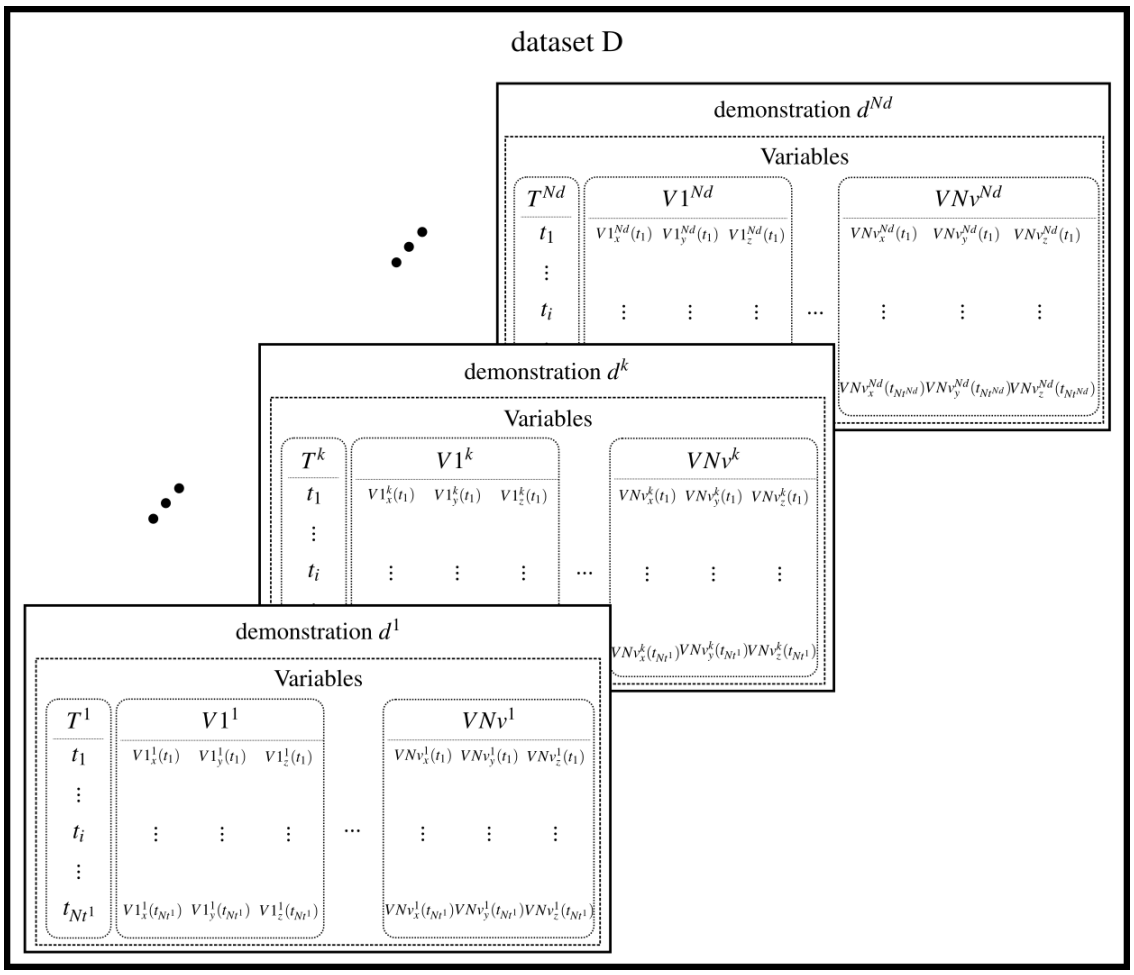

FIGURE 8.2: Structure of the dataset for several demonstrations used for the probabilistic approach [29].

the variable values for all the time steps. The values of the components of a variable Vj for a given time step is noted by the vector $Vj(t_i)$. Let us consider that No objects are involved in the task. The initial object positions O_l with $l \in [1, No]$ are recorded. In addition, some variables are computed from the recorded variables and the initial positions of the objects. Figure 8.2 represents the demonstrations and the variables. In their work, Calinon et al. [29] typically perform the demonstrations by kinesthesy, therefore, one of the recorded variables is the position of the robot in the form of joint angles. For the targeted pick-and-drop task, each demonstration d^k has the following variables:

- T : the time which is a $Nt^k \times 1$ vector. Each time step is noted t_i with $i \in [1, Nt^k]$.
- H : the IHMT estimation of the position of the human hand with respect to the robot frame and corrected by the object observations using the correction method. The variable H^k having 3 components, it is a $Nt^k \times 3$ matrix.
- HC : the distance between the hand position during the trajectory and the initial position of the cube $C0^k$. HC^k is computed as $HC^k(t_i) = H^k(t_i) - C0^k \; \forall t_i \in T^k$. HC is a $Nt^k \times 3$ matrix.

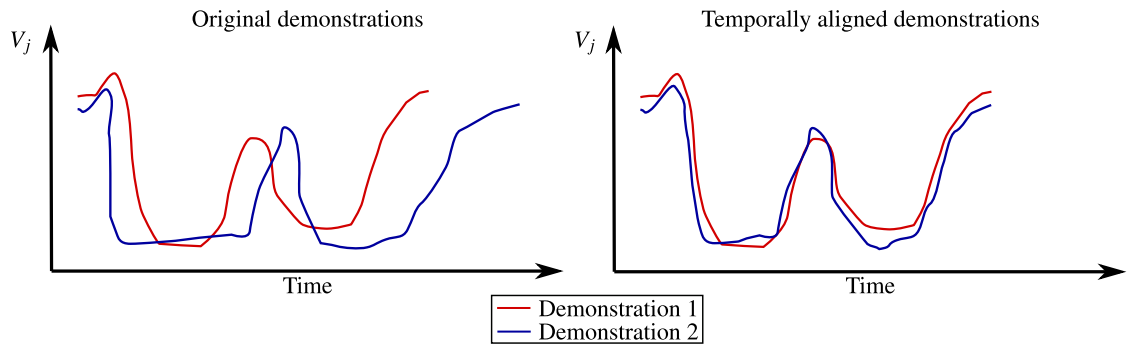

FIGURE 8.3: Illustration of the temporal alignment of 2 demonstrations with 1 variable of 1 dimension.

- HJ : similarly, the distance of the hand to the initial position $J0$ of the jar. HJ^k is a $Nt^k \times 3$ matrix.
- G : The hand state which is a 1-component variable of Nt^k time steps. $G^k(t_i)$ is either 0 or 1 corresponding respectively to the gripper open or close.

The dimensionality of the dataset can be high and difficult to manage. For instance, the configuration of a dual-armed robot in the form of joint angles may contain up to 15 components. Moreover, some components of a variable can be less relevant. For instance, the torso of a humanoid robot can be barely used if the objects are close enough to the robot. Calinon et al. [29] then propose to project the dataset into a latent space. The authors recommend applying a principal component analysis (PCA). Applying a PCA to the variables Vj with $j \in [1,Nv]$ transforms them into the variables $\mathcal{V}j$ ($j \in [1,Nv]$) of a lower dimension. The relation between Vj and $\mathcal{V}j$ is written as

$$Vj - \bar{V}j = Aj\mathcal{V}j \tag{8.1}$$

with $\bar{V}j$ the means of Vj and Aj represents the transformation matrix. In [29], the Gaussian process is applied on the projected variables $\mathcal{V}j$. The computed robotic command is transformed back into the interpretable variables thanks to Aj. However, in our context, the 4 variables H, HJ, HC and G contain together 10 components which is an acceptable dimensionality. The projection into the latent space is then not necessary and simply skipped. The Gaussian process is applied on the regular variables Vj.

Finally, the dataset is temporally aligned as illustrated in Figure 8.3. Since the demonstrations are performed without any temporal constraints, the same action among the demonstrations does not necessarily correspond to the same time steps. In order for the Gaussian process to be relevant, the demonstrations must be temporally aligned. To do so, a Dynamic Time Warping (DTW) is performed. During this step, only the temporal variable T^k with $k \in [1,Nd]$ of the demonstrations is modified and become all identical. It is proposed here to take advantage of the POIs for the temporal alignment. All the extracted hand POIs are considered and not only the POIs involved in the pairs. It is considered that all the demonstrations have the same amount of hand POIs N_{POI}. For each demonstration d^k, the temporal components of the POIs are gathered into the time actions vector $\mathbf{T}^k_{actions}$ as

$$\mathbf{T}^k_{actions} = [s_1, e_1, ..., s_{N_{POI}}, e_{N_{POI}}]. \tag{8.2}$$

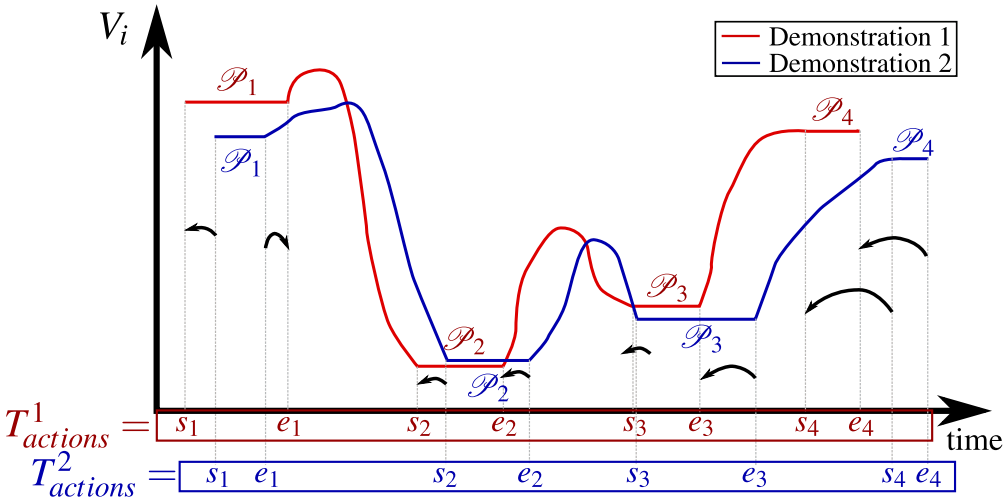

FIGURE 8.4: Illustration of the temporal alignment method based on the POIs.

The time steps in $\mathbf{T}_{actions}$ with the same index correspond to the same moment in all demonstrations. For instance, let us consider the moment when the teacher's hand arrives at the drop position. The time step corresponding to this moment is called t_{drop}. It is present in all the demonstrations but has a different value: $t^1_{drop} \neq t^2_{drop} ... \neq t^k_{drop} ... \neq t^{Nd}_{drop}$. The time t_{drop} is the first temporal boundary of the POI created at the drop position. It can be found in all the $\mathbf{T}^k_{actions}$ vectors at the same index u: $\mathbf{T}^k_{actions}(u) = t^k_{drop} \forall k \in [1, Nd]$. The temporal variable T^1 and the time actions vector $\mathbf{T}^1_{actions}$ of the first demonstration are used as reference and the other demonstrations are aligned to it. The temporal alignment of the demonstration d^k with $k \in [2, Nd]$ consists in modifying the temporal variables T^k as illustrated in Figure 8.4. Let us consider two consecutive elements of $\mathbf{T}^k_{actions}$: $\mathbf{T}^k_{actions}(u-1)$ and $\mathbf{T}^k_{actions}(u)$. The temporal variable T^k is "stretched" or "compressed" between the time step $t^k_i = \mathbf{T}^k_{actions}(u-1)$ and $t^k_j = \mathbf{T}^k_{actions}(u)$ in order for t^k_i and t^k_j to fit with the time steps $t^1_m = \mathbf{T}^1_{actions}(u-1)$ and $t^1_n = \mathbf{T}^1_{actions}(u)$. Algorithm 4 describes this process.

Algorithm 4 : POIs based temporal alignment of demonstrations

```
for k ← 2 to N_d do
    for u ← 2 to 2 × N_POI do
        gap = T^k_actions(u) − T^1_actions(u)
        I = indexes of T^k such as T^k_actions(u−1) ≤ T^k(I) ≤ T^k_actions(u)
        for i ← 1 to Card(I) do
            T^k(I(i)) = T^k(I(i)) + i × gap / Card(I)
        end for
    end for
end for
```

After the temporal alignment, each demonstration d^k has its temporal variable T^k starting and ending with the same values and the actions (picking, dropping ...) start and end at the same time. However, the number of time steps Nt^k in each demonstration remains unchanged. For the next step, the demonstrations and variables are collected in a $M \times N$ matrix D. D is constructed by concatenating all the aligned demonstrations as described in Figure 8.5. M is the number of

$$
D = \overbrace{\begin{array}{|ccc|ccc|ccc|}
\multicolumn{3}{c|}{1^{st}\text{ demonstration}} & \multicolumn{3}{c|}{k^{th}\text{ demonstration}} & \multicolumn{3}{c|}{Nd^{th}\text{ demonstration}} \\
t_1 & \cdots & t_{Nt^1} & t_1 & \cdots & t_{Nt^1} & t_1 & \cdots & t_{Nt^1} \\
H_x^1(t_1) & \cdots & H_x^1(t_{Nt^1}) & H_x^k(t_1) & \cdots & H_x^{Nd}(t_{Nt^1}) & H_x^{Nd}(t_1) & \cdots & H_x^{Nd}(t_{Nt^1}) \\
H_y^1(t_1) & \cdots & H_y^1(t_{Nt^1}) & H_y^k(t_1) & \cdots & H_y^{Nd}(t_{Nt^1}) & H_y^{Nd}(t_1) & \cdots & H_y^{Nd}(t_{Nt^1}) \\
H_z^1(t_1) & \cdots & H_z^1(t_{Nt^1}) & H_z^k(t_1) & \cdots & H_z^{Nd}(t_{Nt^1}) & H_z^{Nd}(t_1) & \cdots & H_z^{Nd}(t_{Nt^1}) \\
HJ_x^1(t_1) & \cdots & HJ_x^1(t_{Nt^1}) & HJ_x^k(t_1) & \cdots & HJ_x^{Nd}(t_{Nt^1}) & HJ_x^{Nd}(t_1) & \cdots & HJ_x^{Nd}(t_{Nt^1}) \\
HJ_y^1(t_1) & \cdots & HJ_y^1(t_{Nt^1}) & HJ_y^k(t_1) & \cdots & HJ_y^{Nd}(t_{Nt^1}) & HJ_y^{Nd}(t_1) & \cdots & HJ_y^{Nd}(t_{Nt^1}) \\
HJ_z^1(t_1) & \cdots & HJ_z^1(t_{Nt^1}) & HJ_z^k(t_1) & \cdots & HJ_z^{Nd}(t_{Nt^1}) & HJ_z^{Nd}(t_1) & \cdots & HJ_z^{Nd}(t_{Nt^1}) \\
HC_x^1(t_1) & \cdots & HC_x^1(t_{Nt^1}) & HC_x^k(t_1) & \cdots & HC_x^{Nd}(t_{Nt^1}) & HC_x^{Nd}(t_1) & \cdots & HC_x^{Nd}(t_{Nt^1}) \\
HC_y^1(t_1) & \cdots & HC_y^1(t_{Nt^1}) & HC_y^k(t_1) & \cdots & HC_y^{Nd}(t_{Nt^1}) & HC_y^{Nd}(t_1) & \cdots & HC_y^{Nd}(t_{Nt^1}) \\
HC_z^1(t_1) & \cdots & HC_z^1(t_{Nt^1}) & HC_z^k(t_1) & \cdots & HC_z^{Nd}(t_{Nt^1}) & HC_z^{Nd}(t_1) & \cdots & HC_z^{Nd}(t_{Nt^1}) \\
G^1(t_1) & \cdots & G^1(t_{Nt^1}) & G^k(t_1) & \cdots & G^k(t_{Nt^1}) & G^{Nd}(t_1) & \cdots & G^{Nd}(t_{Nt^1})
\end{array}}^{\text{N}} \Bigg\} \text{M}
$$

FIGURE 8.5: Structure of the dataset D after temporal alignment of the demonstrations.

components of the variables, including time. The 3 variables H, HJ and HC have 3 components and the time and G are both one-dimensional vectors, so $M = 11$. N is the amount of data points in all the demonstrations: $N = \sum_{k=1}^{Nd} Nt^k$.

8.1.2 Modeling and generalizing the task

First, a model of the task, composed of a mixture of Gaussians, is computed from the temporally aligned demonstrations. This **Gaussian Mixture Model (GMM)** consists in several multivariate Gaussians representing the dataset D and thus reflecting the task constraints. It is illustrated in the upper part of Figure 8.6. Ng Gaussians are considered in the GMM. The number Ng of Gaussians can be optimized by trial and error. In their work, Calinon et al. [29] conducted experiments where the values of Ng are comprised between 4 and 7. The means of the Gaussians $\boldsymbol{\mu} = [\mu_1\ ,\ ...\ ,\ \mu_u\ ,\ ...\ ,\ \mu_{Ng}]$ and their covariances $\boldsymbol{\sigma} = [\sigma_1\ ,\ ...\ ,\ \sigma_u\ ,\ ...\ ,\ \sigma_{Ng}]$ are estimated. A first guess of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ is given by clustering the dataset D. The authors advise using a *k-means* algorithm. Then, the values of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are optimized using the *Expectation-Maximization* algorithm. It is proposed here to take advantage of the POIs again to estimate the GMM parameters. The temporal action vector $\mathbf{T}^1_{actions}$ already segments all the demonstrations in a relevant way. It is recalled that $\mathbf{T}^1_{actions}$ is consistent with all the demonstrations after temporal alignment. Since $\mathbf{T}^1_{actions}$ is built from POIs and POIs are key positions related to actions in the task, the part of the demonstrations between two elements of $\mathbf{T}^1_{actions}$ is considered as an action (motion toward a position, picking up, dropping, avoidance motion...). Let us consider the u^{th} action between the moments $\mathbf{T}^1_{actions}(u)$ and $\mathbf{T}^1_{actions}(u+1)$ (see Figure 8.6). The list I gathers the Nu indices of D corresponding to this action in all the demonstrations. The mean $\boldsymbol{\mu}_u$ is a M-component vector. It corresponds to the u^{th} Gaussian and it is computed from the dataset D as

$$\boldsymbol{\mu}_u = \frac{1}{Nu} \sum_{i \in I} D[:, i]\ . \tag{8.3}$$

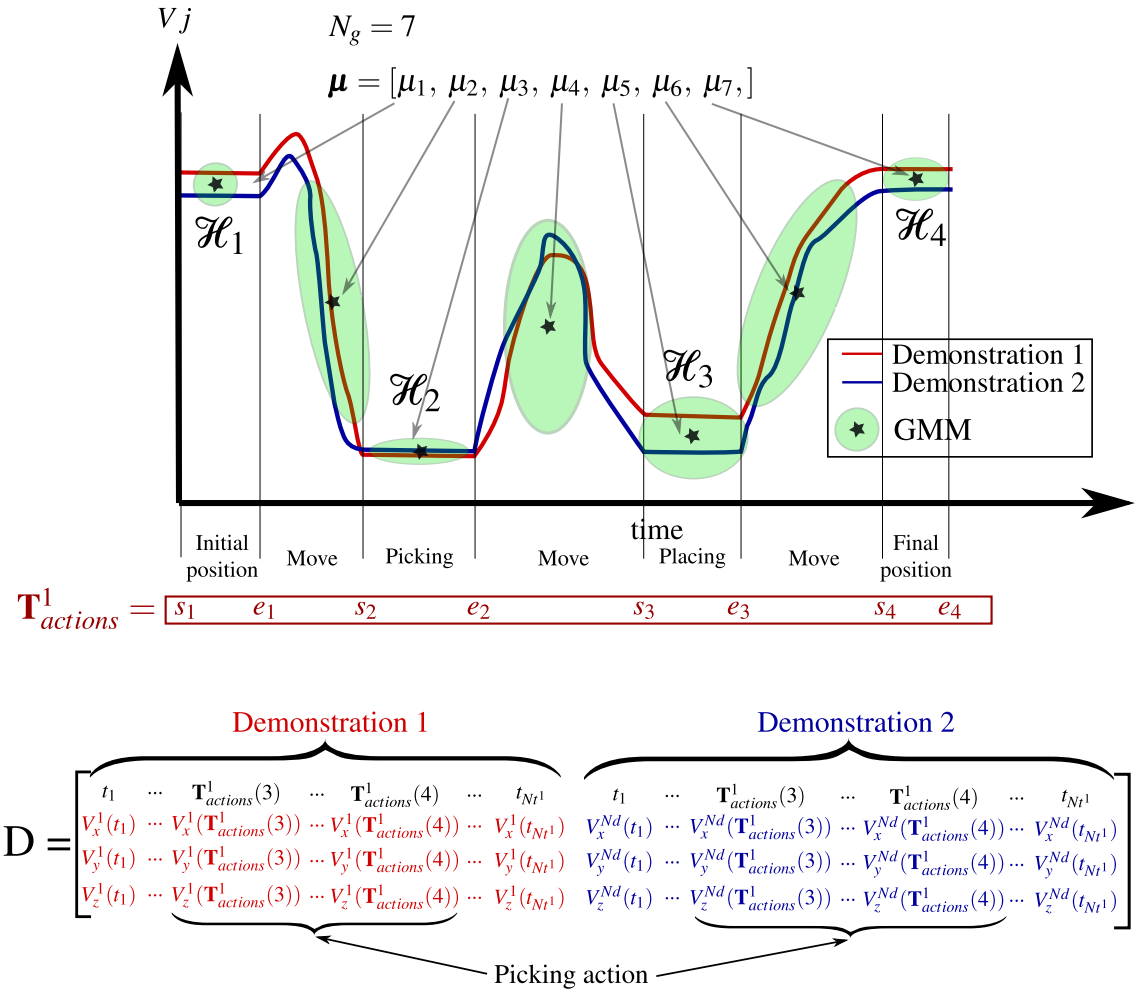

FIGURE 8.6: Illustration of the Gaussian Mixture Model.

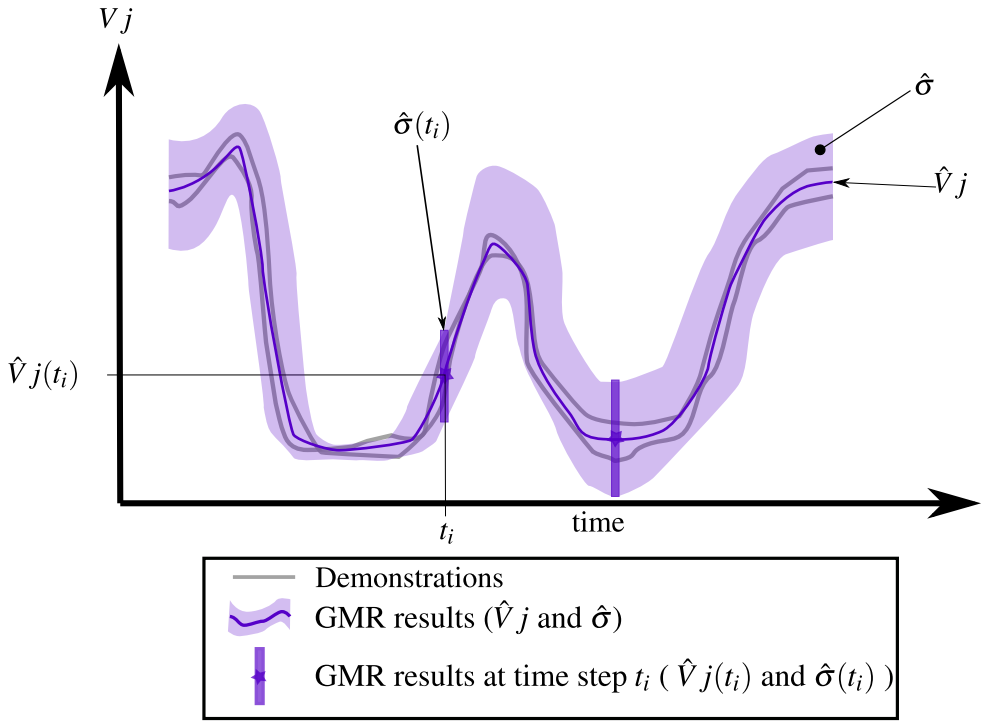

FIGURE 8.7: Illustration of the Gaussian Mixture Regression.

The covariance matrix σ_u is computed as

$$\sigma_u = cov(D[:,I]) \; . \tag{8.4}$$

For each action u, a Gaussian is identified in this way. This set of Gaussians constitutes the GMM. The number of Gaussians Ng in the GMM is then $Ng = Card(\mathbf{T}^1_{actions}) - 1 = 2 \times N_{POI} - 1$. No optimization of the parameters $\boldsymbol{\mu}$ and σ is necessary since the time action vector $\mathbf{T}^1_{actions}$ already segments the task in a relevant way.

Secondly, from the GMM, the task is generalized through a **Gaussian Mixture Regression (GMR)**. The GMR returns a single generalized time-dependent signal for each variable of the task as well as the time-dependent covariance matrices representing the relation across the different variables. The results of the GMR encapsulates the essential task features. The results of the GMR is illustrated in Figure 8.7. In this regression, the query points are the time steps of the temporal variable T^1. The results of the GMR are the means $\hat{V}(t_i)$ and a set of covariance matrices $\hat{\sigma}(t_i)$ corresponding to a set of multi-variable Gaussians at each time step t_i of T^1. Thus, the resulting variables $\hat{V}$ at time step t_i in T^1, is $\hat{V}(t_i) = [\hat{V}1(t_i), \; ... \; , \hat{V}j(t_i), \; ... \; , \hat{V}Nv(t_i)]^T$ and represents the means of the multi-variables Gaussian. In our context the GMR returns $\hat{H}$, $\hat{HC}$, $\hat{HJ}$ and $\hat{G}$ as well as the covariance matrices $\hat{\sigma}$. For each time step t_i from the time vector T^1, a multi-variable Gaussian is associated for which the means of the multi-variables Gaussian is

$$\hat{V}(t_i) = [\hat{H}_x(t_i) \; \hat{H}_y(t_i) \; \hat{H}_z(t_i) \; \hat{HC}_x(t_i) \; \hat{HC}_y(t_i) \; \hat{HC}_z(t_i) \; \hat{HJ}_x(t_i) \; \hat{HJ}_y(t_i) \; \hat{HJ}_z(t_i) \; \hat{G}(t_i)]^T . \tag{8.5}$$

The covariance matrix for these 10 components is given by $\hat{\sigma}(t_i)$.

Once the task is generalized by the GMR, the command for the execution of the task by the robotic system under a new configuration can be computed. The command is computed by an optimization process described in the next section.

8.1.3 Execution of the task

The goal is to compute the command sent to the robot to execute the task in a new configuration. It consists of a Cartesian trajectory of the gripper and a command to close or open it. The command is computed from the minimization of a metric of imitation given the object positions in the scene. This metric of imitation is a measure that evaluates the task reproduction performance of a command and should take into account the new task configuration and the task constraints encapsulated in the GMR results. The metric of imitation takes the form of a cost function $COST$.

Let us considered a new task configuration never encountered in the demonstrations. According to the current context, a new configuration corresponds to the new initial positions O_l^* with $l \in [1,No]$. These positions are assumed to be known by the robotic system. The objective is to compute the desired variables Vj^* with $j \in [1,Nv]$ that can be used to command the robotic system to execute the task in the new configuration. The cost function $COST$ is defined as

$$\hat{V}j = [\hat{V}j(t_1)\ ,\ ...\ ,\ \hat{V}j(t_{Nt^1})]^T \tag{8.6}$$

$$COST = \sum_{j=1}^{Nv} (Vj^* - \hat{V}j)^T\, W_{Vj}\, (Vj^* - \hat{V}j) \tag{8.7}$$

It can be recalled that $\hat{V}$ and $\hat{\sigma}$ are the generalized variables resulting from the GMR and which encapsulate the task constraints. W_{Vj} for $j \in [1,Nv]$ are the weighting matrices which represent the time-varying relative importance of the variables during the task. The desired variables Vj^* are computed through the minimization of the cost function $COST$ under some constraints.

In our context, the computed variables to command the robot are the hand/gripper H^* trajectory and the gripper state G^*. Since the gripper state is a discrete variable, the value for the task execution is computed directly from $\hat{G}$ as

$$G^*(i) = \begin{cases} 1\ , & \text{if } \hat{G}(i) > k_g \\ 0\ , & \text{if } \hat{G}(i) \leq k_g \end{cases} \tag{8.8}$$

with k_g is arbitrarily set at 0.5. The initial object positions $C0^*$ and $J0^*$ of the new task configuration intervene in the computation of H^* developed in the following. To compute H^*, the goal is to minimize the gap between the generalized (" ˆ ") and the desired (" * ") variables for the position of the hand H, the distance HC between the hand and the cube and the distance HJ between the hand and the jar and taking into account the new initial object positions. In the context of this work, the cost function $COST$ is

$$\begin{aligned} COST = {} & (H^* - \hat{H})^T W_H (H^* - \hat{H}) + (HJ^* - \hat{HJ})^T W_{HJ} (HJ^* - \hat{HJ}) \\ & + (HC^* - \hat{HC})^T W_{HC} (HC^* - \hat{HC}) \end{aligned} \tag{8.9}$$

As advised in [29], W_H, W_{HJ} and W_{HC} are the weights computed from the covariances $\hat{\sigma}$ resulting from the GMR. $\hat{\sigma}(t_i)$ is a $M-1 \times M-1$ matrix (the temporal variable is the query points of the GMR). The weights at time step t_i are computed as

$$W_H(t_i) = \frac{1}{tr(\hat{\sigma}(t_i))} \begin{bmatrix} \hat{\sigma}(t_i)_{11} & 0 & 0 \\ 0 & \hat{\sigma}(t_i)_{22} & 0 \\ 0 & 0 & \hat{\sigma}(t_i)_{33} \end{bmatrix} \tag{8.10}$$

$$W_{HJ}(t_i) = \frac{1}{tr(\hat{\sigma}(t_i))} \begin{bmatrix} \hat{\sigma}(t_i)_{44} & 0 & 0 \\ 0 & \hat{\sigma}(t_i)_{55} & 0 \\ 0 & 0 & \hat{\sigma}(t_i)_{66} \end{bmatrix} \tag{8.11}$$

$$W_{HC}(t_i) = \frac{1}{tr(\hat{\sigma}(t_i))} \begin{bmatrix} \hat{\sigma}(t_i)_{77} & 0 & 0 \\ 0 & \hat{\sigma}(t_i)_{88} & 0 \\ 0 & 0 & \hat{\sigma}(t_i)_{99} \end{bmatrix}. \tag{8.12}$$

The desired variables are computed from the minimization of the metric COST under constraints related to the definition of the variables HC and HJ: $HJ^* = H^* - J0^*$ and $HC^* = H^* - C0^*$. As proposed by Calinon et al. [29], the metric $COST$ can be analytically minimized using the Lagrange multipliers method. So, the Lagrangian L is defined as

$$L(H^*, HJ^*, HC^*, \lambda_1, \lambda_2) = COST + \lambda_1^T (HJ^* - H^* + J0^*) + \lambda_2^T (HC^* - H^* + C0^*) \tag{8.13}$$

In our context, only the variable H^* is useful for the execution of the task by the robot. H^* can be computed by finding a minimum of L which leads to the system

$$\begin{cases} \frac{\partial L}{\partial H^*} = 0 \\ \frac{\partial L}{\partial HJ^*} = 0 \\ \frac{\partial L}{\partial HC^*} = 0 \\ \frac{\partial L}{\partial \lambda_1} = 0 \\ \frac{\partial L}{\partial \lambda_2} = 0 \end{cases}. \tag{8.14}$$

So,

$$\begin{cases} -2W_H(H^* - \hat{H}) - \lambda_1 - \lambda_2 = 0 \\ -2W_{HJ}(HJ^* - \hat{HJ}) + \lambda_1 = 0 \\ -2W_{HC}(HC^* - \hat{HC}) + \lambda_2 = 0 \\ HJ^* - H^* + J0^* = 0 \\ HC^* - H^* + C0^* = 0 \end{cases}. \tag{8.15}$$

Thus, the Lagrange multipliers are

$$\lambda_1 = 2W_{HJ}(HJ^* - \hat{HJ}) \tag{8.16}$$

$$\lambda_2 = 2W_{HC}(HC^* - \hat{HC}) \tag{8.17}$$

which leads to,

$$W_H(H^* - \hat{H}) + W_{HJ}(HJ^* - \hat{HJ}) + W_{HC}(HC^* - \hat{HC}) = 0 \tag{8.18}$$

Furthermore, using $HJ^* = H^* - J0^*$ and $HC^* = H^* - C0^*$, we obtain

$$W_H(H^* - \hat{H}) + W_{HJ}(H^* - J0^* - \hat{HJ}) + W_{HC}(H^* - C0^* - \hat{HC}) = 0 \tag{8.19}$$

and finally,

$$H^* = (W_H + W_{HJ} + W_{HC})^{-1}(W_H\hat{H} + W_{HJ}(J0^* + \hat{HJ}) + W_{HC}(C0^* + \hat{HC})) \tag{8.20}$$

Consequently, the complete gripper trajectory H^* taking into account the new task configuration is computed from Eq. (8.20). The trajectory H^* and the gripper change of state G^* are used to command the Sawyer robot. The complete setup of this experiment is described in the next section.

8.2 The experiment

8.2.1 The setup

In this study, the task taught by PbD is to pick up a cube from the table and drop it into a jar. The object dropped in the jar is a 50 mm side cube with an ArUco marker on it. The distance between the fingers of the gripper is 60 mm when it is open and 40 mm when it is closed. The jar is an empty 80 mm diameter cylinder with 2 ArUco markers attached to it (see Figure 8.8. The two markers locate the center of the top face of the cylinder. One marker is in principle sufficient but a second one is added for a better robustness on the jar position. The vision system used in this experiment is the same as described in Section 7.3 based on a ZED camera. The setup is illustrated in Figure 8.9.

This task involves a dropping position, i.e., a position where the cube is held then dropped from above the jar. At this position, the object must be seen by the camera in order to correct the IHMT hand position. The gun-shape tool described in the previous chapter limits disturbing finger motion but often hinders the detection of the ArUco marker when the object is held. Therefore, the gun-shape tool is abandoned and the teacher manipulates the object directly with his/her own hand. The cube can be held in a way that the ArUco marker is seen at the drop position. The sensor to estimate the hand orientation is directly attached to the top of the human hand (see Figure 8.9). The tracked point is the contact point between the thumb and the index

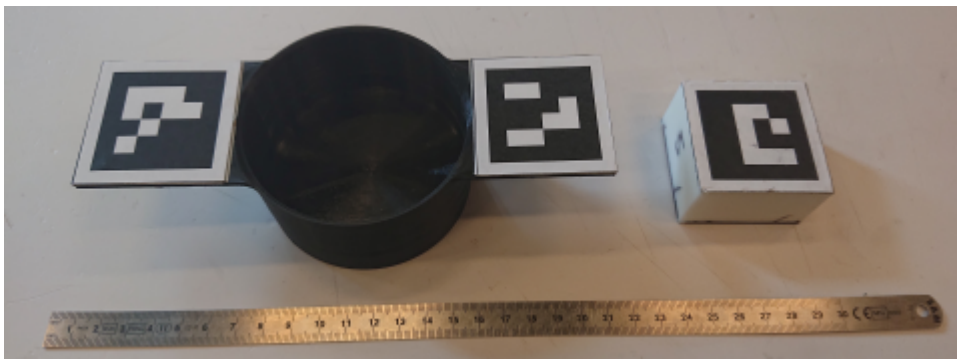

FIGURE 8.8: The objects used in the task: the jar (left) and the cube (right).

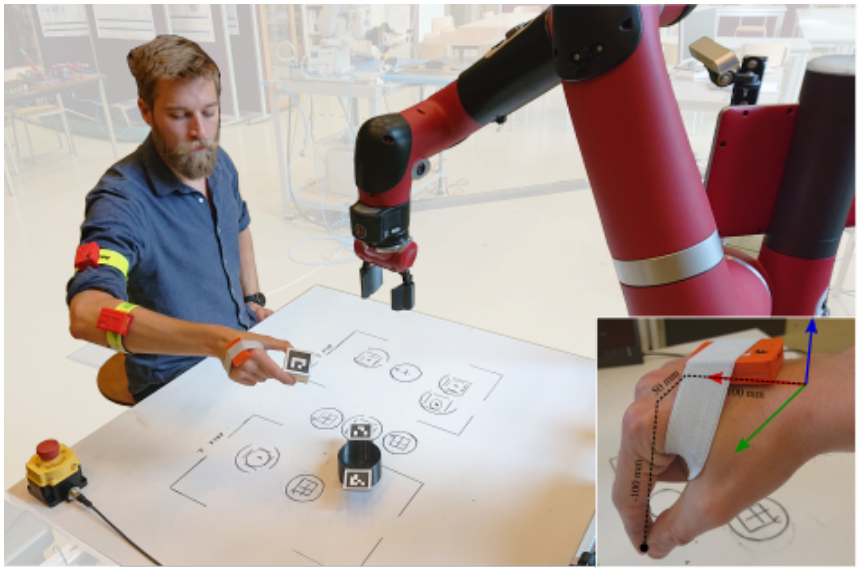

FIGURE 8.9: Setup of the demonstrations by the proposed corrected IHMT method.

finger. The human arm segment vectors, directly measured on the teacher, are ${}^{1n}\mathbf{AB}_n = [330\,0\,0]$, ${}^{2n}\mathbf{BC}_n = [280\,0\,0]$, ${}^{3n}\mathbf{CD}_n = [100\,50\,-100]$.

8.2.2 The demonstrations

Six demonstrations, shown in Figure 8.10, have been recorded with the proposed corrected IHMT method, as illustrated in Figure 8.9. The different characteristics of the demonstrations are shown in Table 8.1. All the demonstrations present four hand POIs corresponding to the initial position, the picking up position, the drop-off position and the final position. Two POIs related to the cube are extracted and used to correct the IHMT hand position at the picking and the dropping positions. The last column of Table 8.1 presents the duration of the demonstrations without including the initialization procedure nor the procedure to estimate the human-robot transformation. All demonstrations are successfully reproduced by the robotic system after their recording to ensure their adequacy for PbD. The 6 positions (A, B, C, D, E and F) marked on the table (see Figure 8.11) are used for the purpose of reproducibility and verification. These marks are not known and are not exploited by the robotic system.

Demonstrations	Cube position	Jar position	$N_{\mathscr{H}}$	$N_{\copyright}$	$N_{\mathfrak{P}}$	time (s)
1	A	C	4	2	2	7.99
2	C	A	4	2	2	16.23
3	B	D	4	2	2	7.41
4	D	B	4	2	2	9.67
5	E	F	4	2	2	8.90
6	F	E	4	2	2	9.07

TABLE 8.1: Features of the demonstrations from the corrected IHMT method.

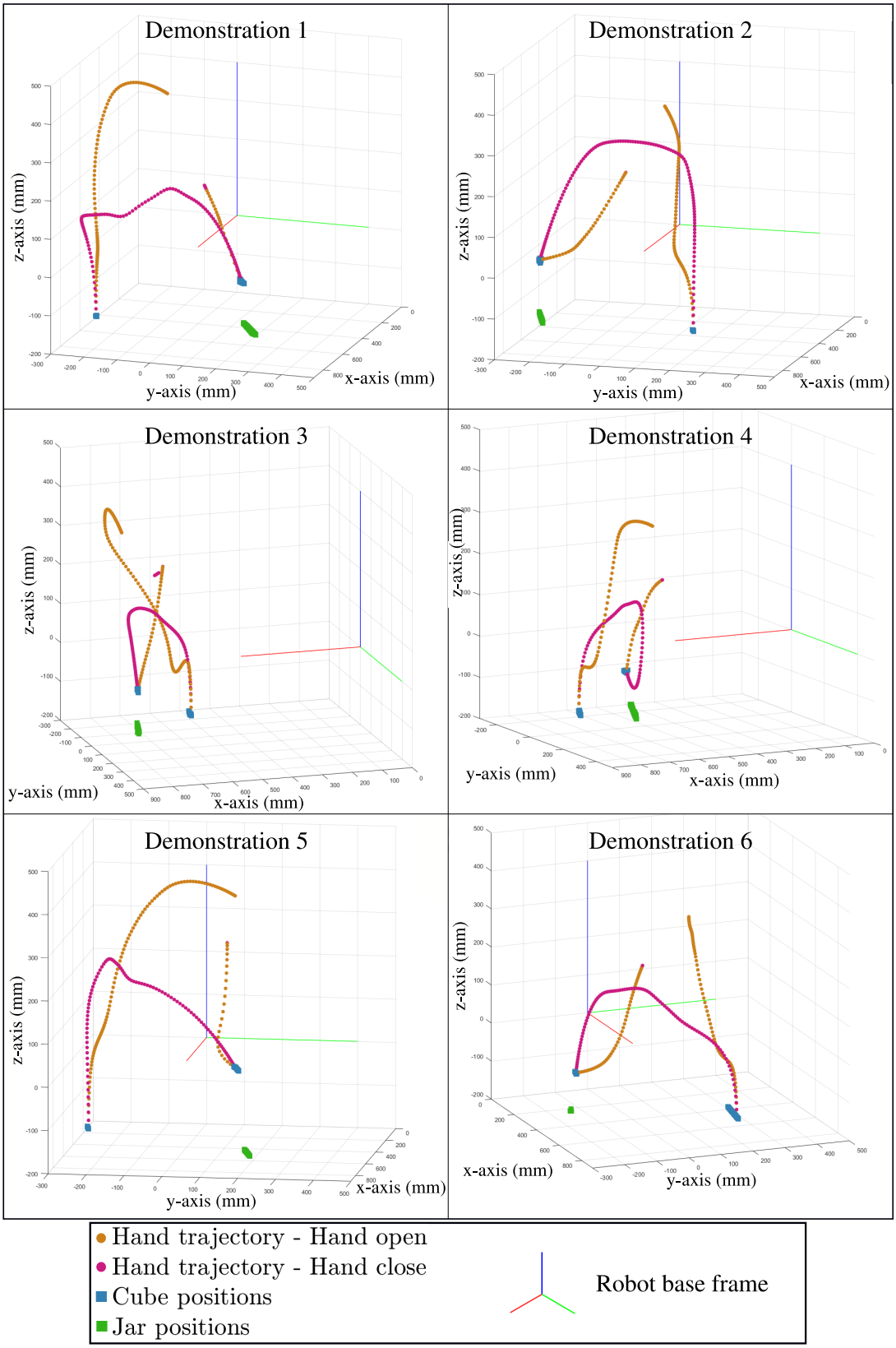

FIGURE 8.10: Demonstrations from the corrected IHMT method.

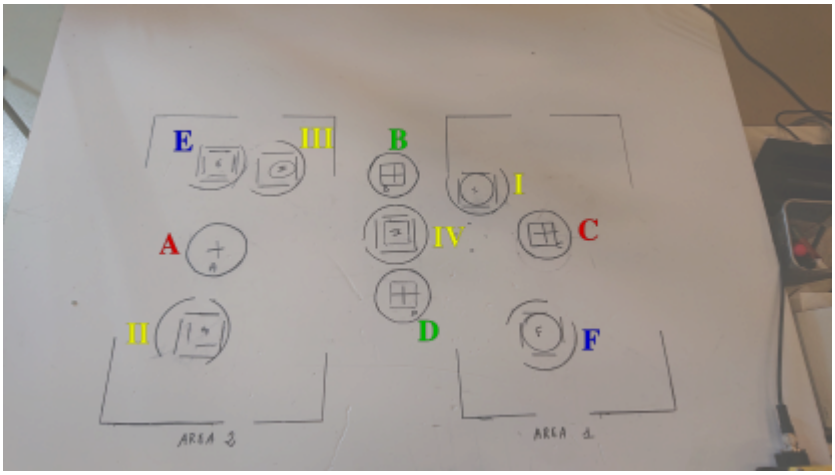

FIGURE 8.11: Initial positions of the objects for the demonstrations (A, B, C, D, E and F) and the task executions (I, II, III and IV).

In order to compare the proposed corrected IHMT method with kinesthesy, six additional demonstrations (see Figure 8.12) were recorded by kinesthesy as illustrated in Figure 8.13. The characteristics of each demonstration are listed in Table 8.2. For comparison purpose, the *learning the task* algorithm needs to be implemented in the same way for both demonstration methods. Therefore, POIs are extracted from the Cartesian gripper trajectory recorded during the demonstrations. The gripper trajectory is not corrected, only the first step of the correction method is applied to extract hand/gripper POIs. For each demonstration, four hand/gripper POIs have been extracted which allows the implementation of the *learning the task* algorithm as described in the previous section. The gripper state (open or close) is also recorded as well as the object initial positions using the same vision system. The last column of Table 8.2 presents the recording times.

Demonstrations	Cube position	Jar position	$N_{\mathcal{H}}$	Time (s)
1	A	C	4	16.87
2	C	A	4	19.53
3	B	D	4	14.70
4	D	B	4	17.05
5	E	F	4	18.28
6	F	E	4	16.55

TABLE 8.2: Features of the kinesthetic demonstrations.

8.2.3 Execution of the task

The task is taught and executed according to 4 different configurations. The task is considered successful if the cube is inside the jar at the end. The configurations consist of 4 object positions (I, II, III and IV) not encountered during the demonstrations (see Figure 8.11). As the vision system has a non-trivial error, these positions have been measured by robot palpation in order to avoid any bias in the comparison between the kinesthetic demonstrations and the demonstrations

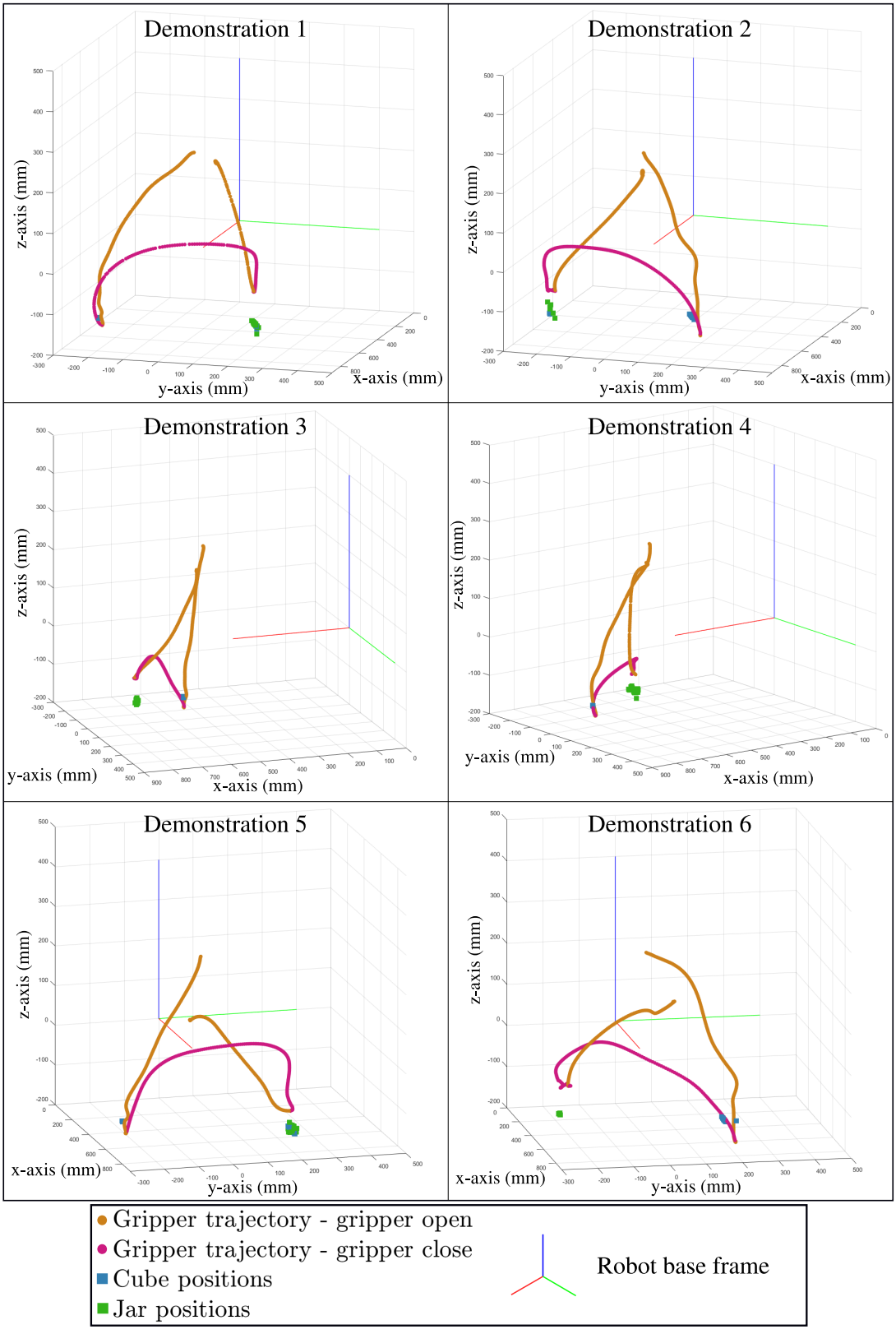

FIGURE 8.12: Demonstrations from kinesthesy.

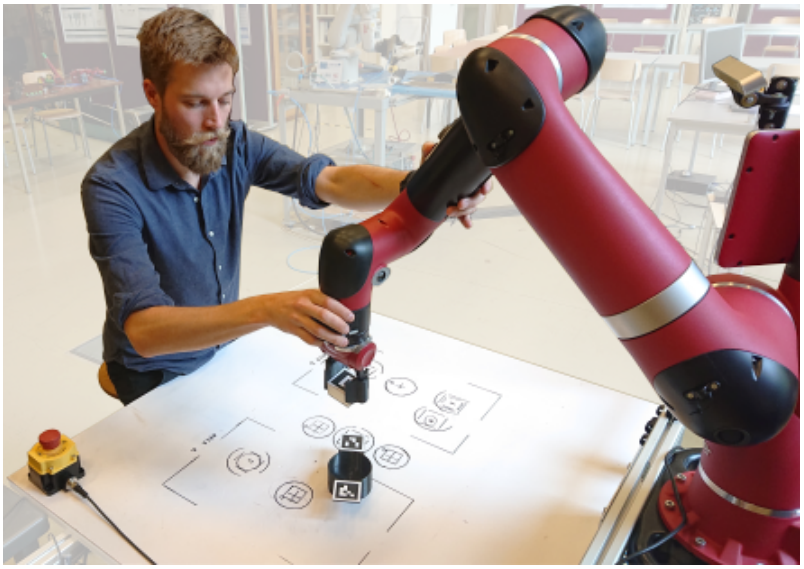

FIGURE 8.13: Setup of the demonstrations by kinesthesy.

from the proposed method. The results regarding the success of the execution of the task are presented in Table 8.3. The 3D representations of the executed task are shown in Figure 8.14. It represents the computed trajectory H^* used as command to the robot and the object positions. We assume that the robot precisely follows the trajectory H^* (according to the manufacturer, its accuracy is 0.1 mm). For each task configuration, the computed trajectory is run with 2 different gripper orientations. The gripper orientation is fixed to be vertical (**z**-axis pointing downwards) but the heading orientation around the vertical axis has 2 different angles: 90° or 0°. This aspect is discussed in the next section.

Task configuration	orientation 90°		orientation 0°	
	proposed method	kinesthesy	proposed method	kinesthesy
Cube I - Jar II	Success	Fail	Fail	Fail
Cube II - Jar I	Success	Fail	Success	Fail
Cube III - Jar IV	Success	Fail	Success	Fail
Cube IV - Jar III	Success	Fail	Fail	Success

TABLE 8.3: Results of the execution of the task for the different configurations.

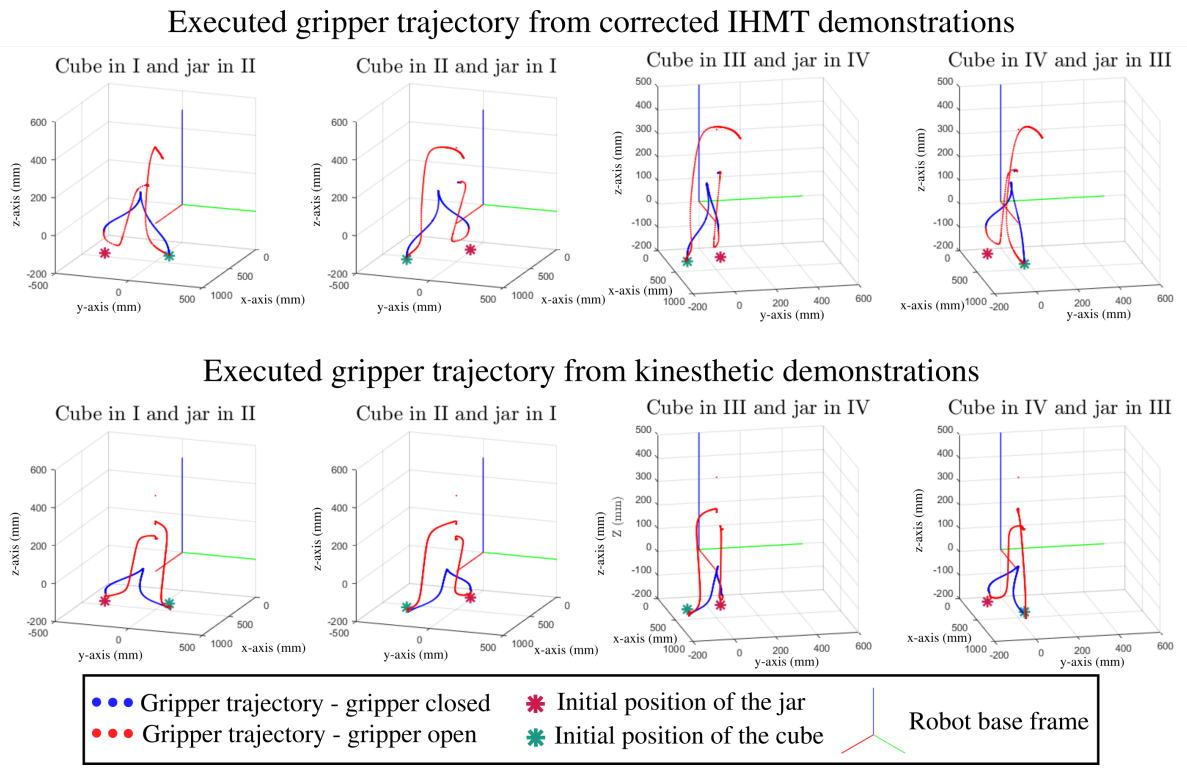

FIGURE 8.14: Executed tasks from corrected IHMT and kinesthetic demonstrations.

8.3 Discussion and perspectives

For 6 out of 8 cases (see Table 8.3), the task was successfully taught by demonstrations from the proposed corrected IHMT method. Considering only the results when the gripper orientation is 90°, the success rate is 100 %. The results reveal that the proposed corrected IHMT method can be used for PbD.

Furthermore, the time required to perform the demonstrations is shorter with the proposed method than with kinesthesy. The values in Table 8.1 and Table 8.2 only consider the demonstration time, not the time of the procedures before the demonstrations. The demonstration times tend to confirm a better handiness of the proposed corrected IHMT method compared to the kinesthetic method in a PbD process. However, additional experiments including several subjects could be conducted to further validate this point.

From Table 8.3, the proposed method is more robust (6 out of 8 successes) than the kinesthetic method (1 out of 8 success) in the experiment. It is shown in Figure 8.14 that all the computed trajectories – from both demonstrations method – reach the pick and the drop positions and only differ slightly. The task execution failures from both demonstration methods seem to stem from a disturbing horizontal motion during the approach of the picking position. The different executions of the task can be watched by following this link `https://youtu.be/KJXpP2PRmkk`. Figure 8.15 shows a more detailed view on this part for a new task configuration. Due to this disturbing horizontal motion, the robot pushes the cube before reaching the picking position. Thus, when the robot closes its gripper, there is no object to grasp or the cube is grasped incorrectly causing the task to fail. With a different orientation of the gripper, the object is not or less

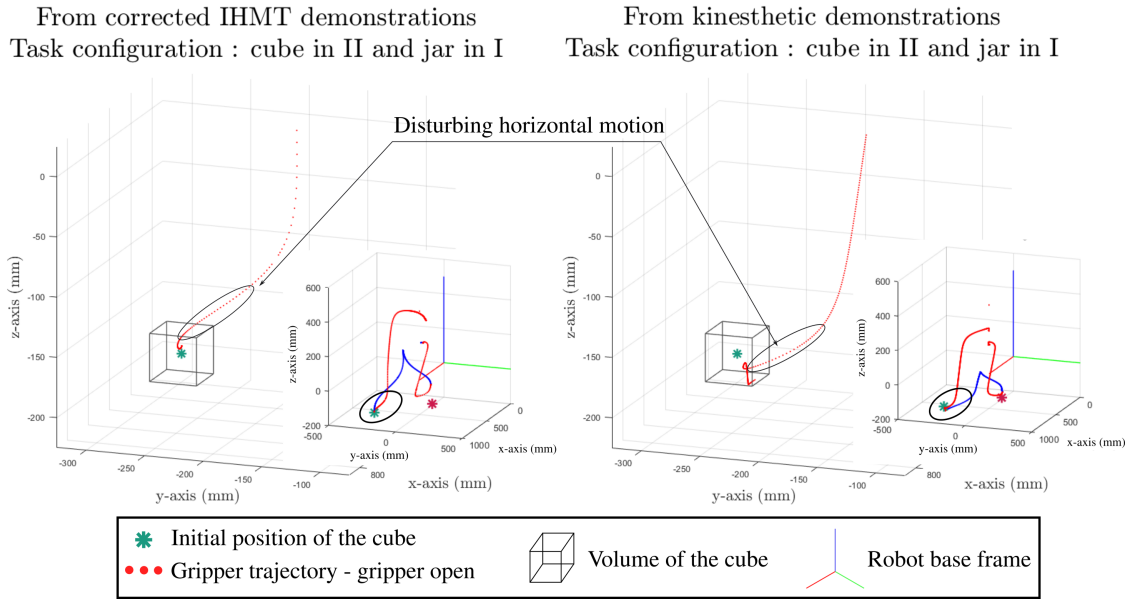

FIGURE 8.15: 3D representation of the resulting trajectories from IHMT and kinesthesy demonstrations with a focus on the disturbing horizontal motion during the picking approach.

pushed away and the task may be successful. The disturbing horizontal motion is more or less important depending on the demonstration method. This motion can be quantified as follows: it starts when the gripper is low enough to be able to touch the cube and it ends when the gripper is close to grab the cube. The TCP of the gripper is the point located 1 cm before the extremity of the gripper fingers (see Figure 8.16). Thus, the gripper is able to touch the cube when the z-component of its position is 1 cm above the z-component of the position of the cube. During this time interval, the horizontal displacement d_h is estimated as

$$d_h = \sum_{i=2}^{N^h} \sqrt{(Hx(t_i) - Hx(t_{i-1}))^2 + (Hy(t_i) - Hy(t_{i-1}))^2} \tag{8.21}$$

with N^h the number of points corresponding to this motion. The different values of d_h are shown in Figure 8.17. The task command computed from kinesthetic demonstrations presents a higher d_h value than from the proposed method. The fact that this element is not present in any of the demonstrations (see Figures 8.10 and 8.12) suggests that it is introduced by the *learning the task* algorithm and/or its implementation. The study of the impact of the *learning the task* algorithm is beyond the scope of this work. Nevertheless, some explanations concerning the demonstration acquisition methods can be given.

As for the differences of values of d_h, some differences in the GMR results can be observed regarding the demonstration acquisition methods. Figure 8.18 shows the $\hat{HC}$ variable (distance from the hand/gripper to the cube) computed from the set of demonstrations from the corrected IHMT method and from kinesthesy. The picking action occurs between 2.0 and 2.7 seconds for the corrected IHMT method and between 5.5 and 7.5 seconds with kinesthesy. At this moment, the $\hat{\sigma}$ values are lower than at other moments of the task. This corresponds to a representation of an action: the hand has to be at the cube position to pick it up. The values of $\hat{\sigma}$ are lower in the GMR results from the corrected IHMT method than from kinesthesy. This means that this action

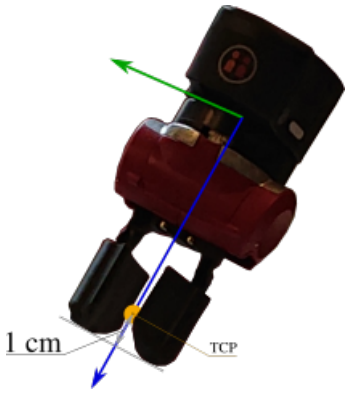

FIGURE 8.16: TCP of the gripper.

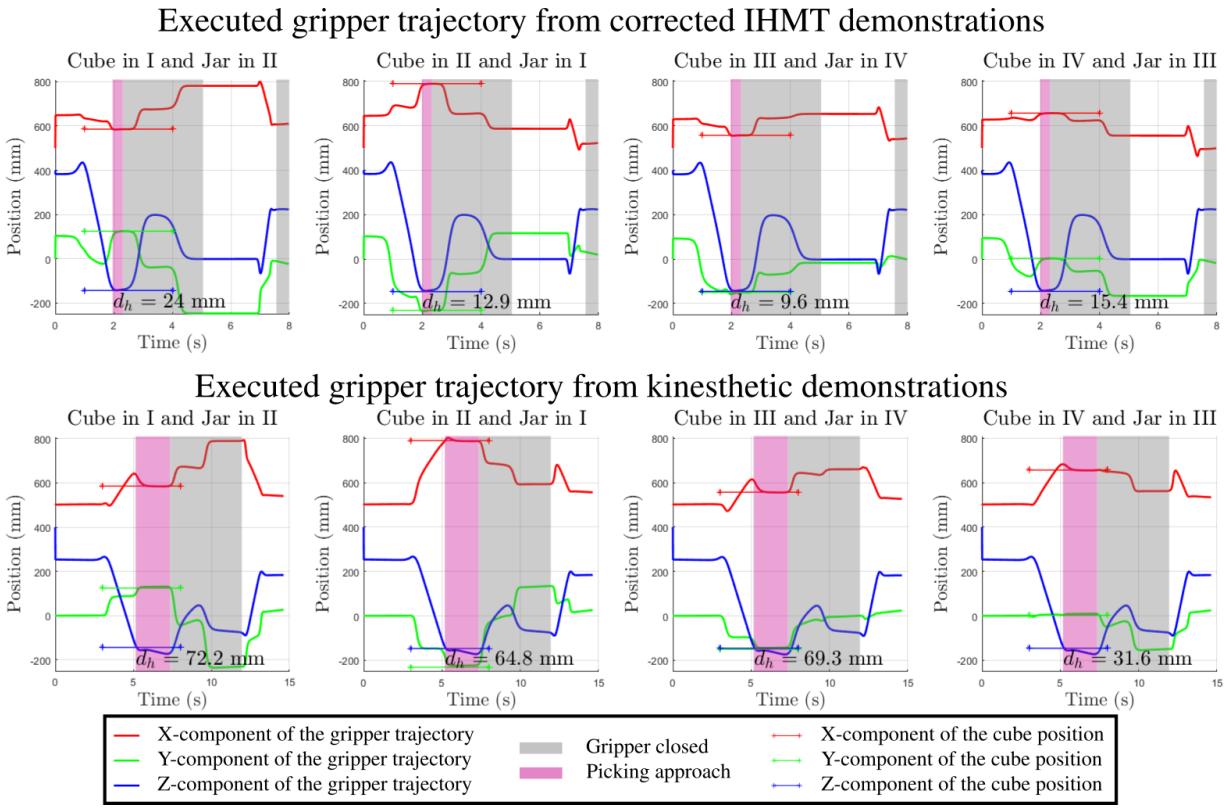

FIGURE 8.17: Gripper trajectory for the different task configurations and the d_h values.

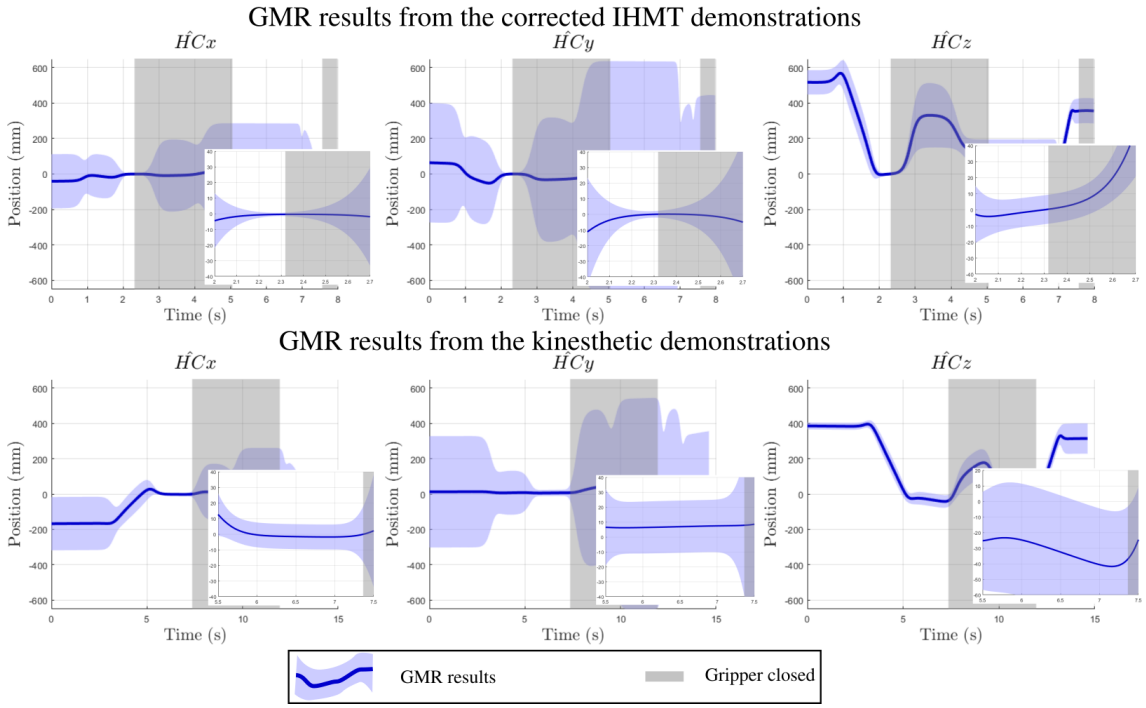

FIGURE 8.18: GMR results for the variable $\hat{HC}$ for both demonstration methods with zoom on the picking action.

is better encoded with the demonstrations from the proposed method than from kinesthesy. It seems that a better encoding of the action limits the intensity of the disturbing horizontal motion in the picking approach, which can explain the higher success rate for the proposed method.

The difference in the encoding of actions between the corrected IHMT method and kinesthesy is directly linked to the quality of the demonstrations. Some additional elements can be given to explain the differences between the corrected IHMT method and kinesthesy. Figure 8.19 shows the temporally aligned demonstration variables: the hand/gripper trajectory with respect to the robot base frame, the distance between the hand/gripper and the initial cube position and the distance between the hand/gripper and the initial jar position. The demonstrations from kinesthesy show a larger dispersion at the picking position (lower part of Figure 8.19, between 5 and 8 seconds) than those from the corrected IHMT method (upper part of Figure 8.19, between 1.8 and 3 seconds). This can be attributed to the difficulty for the teacher to manipulate the robot. The teacher was able to easily move the robot close to the cube position, but had difficulties to accurately place the robot gripper around the cube. Such lack of dexterity may contribute to deteriorate the quality of the demonstrations and increase the disturbing horizontal motion. In addition, the initial position of the cube measured by the vision system presents a certain error. The robot gripper being brought by the teacher to the physical cube position, this error propagates in the HC variable. On the other hand, the correction method modifies the IHMT hand positions at the picking position (POIs corresponding to the picking action). Therefore, the HC variable is forced to be 0 during the picking action (see the upper part of Figure 8.19 between 2 and 3 seconds). The correction method improves the encoding of the picking action.

The double impact of the correction method introduced in Section 7.4 is further illustrated

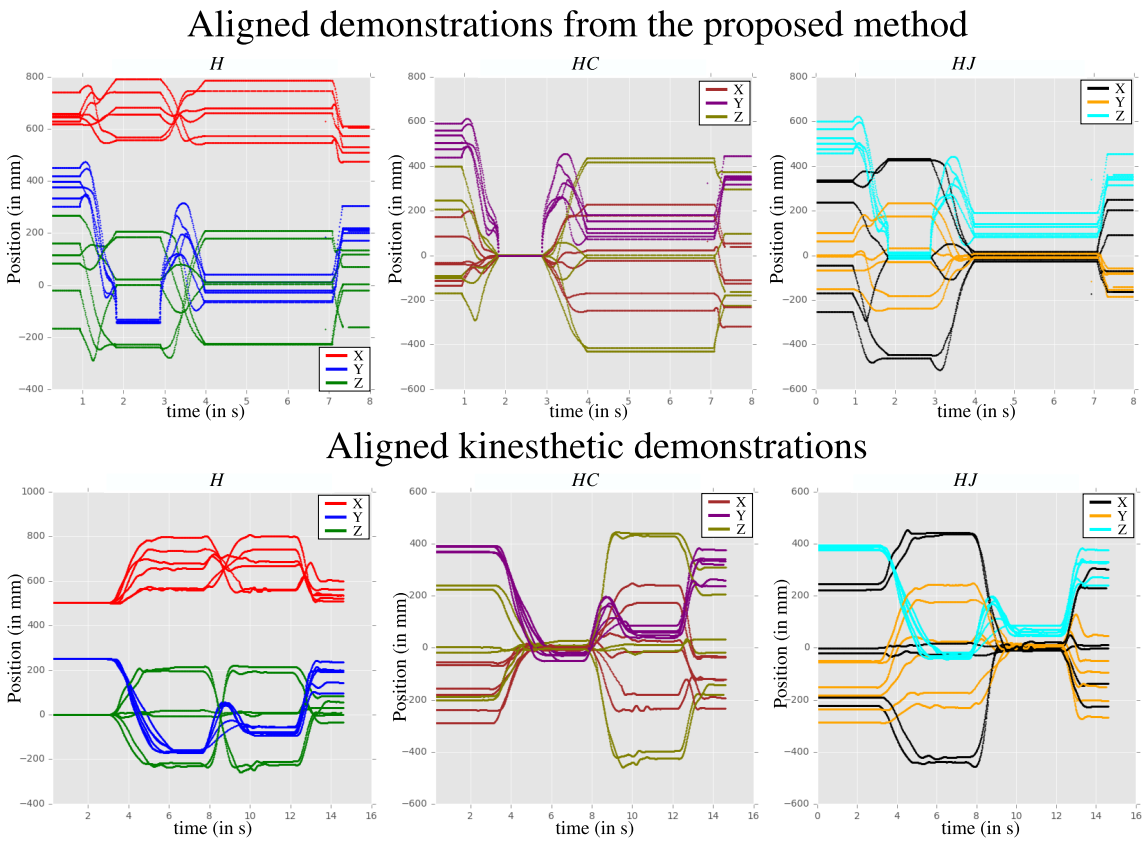

FIGURE 8.19: Aligned demonstrations per variables.

here. The correction method improves the accuracy of the hand trajectory to makes the demonstration suitable for PbD and it improves the encoding of the actions of the task by absorbing the error on the vision-based measurement of the object positions. It can be mentioned that the correction method, in the context of this experiment, is hardly applicable on the kinesthetic demonstrations. The bulkiness of the gripper and the lack of dexterity for the teacher prevent the cube from being detected at the drop position.

From the best of our knowledge, only a few studies compare different demonstration acquisition methods. In [50], Fischer et al. compares kinesthetic guidance with teleoperation from data gloves and teleoperation joint by joint from remote controller. Their study shows that kinesthetic guidance has better usability. However, Fischer et al. compare methods (kinesthesy and teleoperation) that present similarities as explained in the introduction Section 1.1.2. Kinesthesy can present advantages over teleoperation but the same is not necessarily true regarding observational methods. Muxfeldt et al. [98] compare demonstrations by kinesthesy with demonstrations where the human motion is captured from a vision system for an assembly task. Four tasks are considered: peg-in-hole, inserting a shaft into a base, mounting a clip socket on a rail, assemble a bracket to a base. The different methods are not compared in a complete programming by demonstration process but only on a contact time basis. The contact phases start when the part to be assembled is in contact with the other element until the assembly is completed. The detection of these moments is done by force analysis. A short contact time is considered evidence of the superiority of the demonstration method. Muxfeldt et al. state that kinesthetic guidance can be used for such tasks. Regarding the experiment described in this chapter, the success rate of each

method depends on the disturbing horizontal motion during the picking approach introduced by the *learning the task* algorithm and emphasized by the demonstration acquisition method. Therefore, as in this study, the evaluation of a demonstration acquisition method should be considered in a complete PBD process.

8.4 Summary

This chapter details the implementation of the proposed method in a complete PbD process. The selected *learning the task* algorithm is the probabilistic approach from Calinon et al. [29]. The implementation of the probabilistic approach sometimes differs from their original work. In particular, some elements of the implemented *learning the task* algorithm exploit the POIs extracted during the correction of the IHMT hand trajectory.

A pick-and-drop task is taught to a robotic system by the proposed corrected IHMT method and by kinesthetic demonstrations. The taught task consists of picking up a cube from the table and dropping it into a jar. The robot was able to successfully perform the task in configurations non encountered during the demonstrations. The proposed method is suitable for PbD.

The results reveal a higher success rate in the execution of the task from the demonstrations acquired with the proposed method than from the kinesthetic demonstrations. The failures are attributed to a disturbing horizontal motion when approaching the position of the cube to be picked. This disturbing motion is introduced by the *learning the task* algorithm but it is more important when the demonstrations are performed by kinesthesy. A detailed study of this problem reveals that a lack of dexterity of the teacher in kinesthetic demonstrations can be a source of failure in the execution of the task. On the other hand, the correction method applied to the IHMT hand trajectory limits the impact of the error of the vision-based measurement of object positions and emphasize the task constraint, thus improving the behaviour of the robot in the picking approach when executing the task in new configurations.

CHAPTER 9

CONCLUSION

Contents

9.1 Summary and contributions

Programming by demonstration (PbD), also referred to as learning from demonstrations (LfD), aims at teaching new skills or tasks to a robotic system through human demonstrations. This process is typically divided into two main steps: *gathering the dataset* and *learning the task*. The second step is to "understand" the task from the set of demonstrations so that the robot can execute the task in a configuration not encountered in the demonstrations. The first step consists in recording some elements during the teacher's demonstrations. A demonstration is considered suitable for PbD if it can be reproducible by the learner. The teacher and the learner may differ in sensing and actuation and the elements recorded during the demonstration may not be usable by the learner. In that case, a correspondence problem should be solved. According to the demonstration acquisition method, up to two mappings may be necessary to adapt the recorded dataset and overcome this issue: a record mapping and/or an embodiment mapping. One largely used method to acquire demonstrations is kinesthetic guidance where the robot is manipulated by the teacher and the robotic system records the demonstrations with its own sensory system. With such a method, the correspondence issue is circumvented. The recorded elements, such as robot joints angle states, can be used by the learner to mimic the demonstration. However, this method presents disadvantages mainly due to limitations of the teacher's dexterity which can impact the whole PbD process. Thus, some applications as fine manipulations or painting tasks cannot be transmitted by kinesthesy. On the other hand, the demonstrations can be encoded using sensors allowing the teacher to move freely. In this case, the correspondence issue must usually be solved, i.e., the record mapping and/or the embodiment mapping need to be identified. The choice of the demonstration acquisition method requires a compromise between constraints on the teacher's motion and easy identifiable mappings with appropriate levels of accuracy and ease of use and implementation. This problem is addressed in this work which aims at developing a method to acquire demonstrations in a way that is convenient for the teacher. For their easiness of use, inertial sensors have been selected to measure the human hand motion during the demonstrations. In order to solve the correspondence issue, the teacher hand trajectory should ne represented in the robot frame. The targeted applications are pick-and-place tasks in an industrial context.

The inertial human hand motion tracking (IHMT) method proposed in this work first requires the estimation of each sensor orientation with respect to its original orientation. It is largely documented that such an estimation suffers from a drift due to the gyroscope bias and noise integration. The drift is partially compensated using an estimation of the gravity vector when the sensor is not undergoing linear acceleration. **A new criterion based on the derivative of the accelerometer is proposed in this work (see Chapter 2) to better identify such stationary moments**. To reduce the remaining drift, inertial sensors can be combined with a magnetometer. However, such a sensor is highly sensitive to disturbances especially in an indoor environment and near a robot. Instead, **a new method, called the heading reset, is proposed to limit the drift.** In a PbD context, the teacher, facing the workspace, is expected to perform movements where the sensor passes through its original heading several times with stationary phases during the demonstration. At these moments, a heading reset is applied to catch up the drift. The heading reset is implemented in an integration and vector observation algorithm and evaluated in a robot experiment (see Chapter 3). Three sensors are mounted on a robot end-effector. The robot follows a set of human realistic orientations. The recorded orientation by the robot encoders is used as reference. The results of this experiment reveal a positive impact of the heading reset.

The orientation error is evaluated between 1.6° and 13.6° after a one-hour measurement while it is between 5.1° and 35.1° without the heading reset.

Another contribution of this work is the development of an inertial human hand motion tracking (IHMT) method designed for this context that solves the correspondence problem (see Chapter 4). This means that the hand trajectory is automatically obtained in the robot frame. In this work, the human arm is modelled with 3 spherical joints for the wrist, the elbow and the shoulder, and 3 rigid segments (arm, forearm and hand). The estimation of the human hand trajectory relies on the estimation of the human segments orientations. To estimate these orientations, an inertial sensor is attached to each segment. The orientation of the sensor with respect to the segment frame needs to be estimated. A specific procedure is proposed for this purpose. It consists of a rotational motion of the arm straight followed by a static pose. In addition, the hand position has to be expressed with respect to the robot base frame in order to be meaningful for the robotic system. A method to estimate the transformation between the human and the robot is proposed. It consists in asking the human to follow a trajectory imposed by the robot. The data recorded during this part is also used to optimize some parameters involved in the method. Different experiments have been conducted to evaluate the different elements of the proposed method (see Chapter 5). A first experiment performed on a robot arm such that the IMUs are used to estimate the robot end-effector position, showed that the initialization procedure is robust to sensor positioning. The error measured during this experiment is between 27.7 mm and 47.7 mm regardless of the sensor positioning. A second experiment is conducted on a human arm. The error measured during this experiment is comprised between 54.76 mm and 181.84 mm. Compared to the previous result, this last result reveals that the human hand position estimation is disturbed by modelling errors (e.g., the model does not take into account that the shoulder joint centre of rotation depends on the human arm poses, the finger motion is not modelled, torso displacement) and calibration errors (e.g., deviation from the recommended movement of the initialization). A third experiment is conducted to evaluate the human hand trajectory with respect to the robot base frame. The error has been evaluated between 28.5 mm 61.8 mm. The gain in accuracy compared to the previous result comes from the optimization of the system parameters. It partially compensates the error due to the calibration errors.

Due to the sensor orientation errors, the modelling errors and the calibration errors, it seems difficult to achieve a sufficient level of accuracy for PbD only based on IMU signals. **A method is proposed to correct the IHMT estimation of the hand trajectory by a vision-based estimation of objects positions (see Chapter 6).** This method exploits the complementarity between vision and inertial sensors. The proposed correction method is based on 3 main steps. The first step consists in extracting positions of interest (POIs) from vision and inertial data. POIs are static positions that correspond, among others, to picking, placing, initial pose or avoidance pose. More precisely, a POI is a 3D position and two temporal boundaries. The extraction of POIs is based on the clustering algorithm DBSCAN. The relevant actions such as picking and placing induct a POI in both vision data and inertial data. The second step is to pair these POIs. The pairing is a two-step process. The first step iteratively eliminates the POIs that should not be involved in any pairs. The second step is to assign the remaining POIs using the Hungarian method with a cost matrix based on temporal distances between POIs. Once the pairs of POIs are constituted, the inertial hand trajectory is corrected. Firstly, a global correction is applied on the whole trajectory with a transformation estimated from the pairs of POIs. Secondly, the IHMT hand trajectory is locally modified to fit the positions of the object POIs and is smoothed around

these positions. This method has been implemented and evaluated for different pick-and-place tasks (see Chapter 7). It has also been evaluated in a context where the tracked object is a tool held by the teacher. The robot succeeds to mimic the demonstrated tasks in the majority of cases, making the demonstrations appropriate for PbD.

In the last chapter (see Chapter 8), the entire method is evaluated in a complete PbD process. The task to be taught is a pick-and-drop task. An object is picked on the table and dropped from above a jar. The method used to *learn the task* is the probabilistic approach from Calinon et al. [29]. Our corrected IHMT method creates qualitative demonstrations used by the robotics system to execute the task in non-encountered task configurations. A comparison is also made between the proposed method and the kinesthetic method. For this case, the kinesthetic demonstrations lead to a lower success rate. It seems that the lower dexterity due to kinesthesy disturbs the resulting trajectory. It is also seen that the vision-based correction method has a combined impact. By associating a hand POI and an object POI, the task constraints are emphasized which improves the quality of the demonstrations. Such high quality demonstrations contribute to the better success rate of the proposed method in the task execution.

9.2 Conclusions

The present doctoral thesis proposes a corrected IHMT method to acquire demonstrations of pick-and-place tasks for PbD. The method meets the requirement of PbD and is convenient for the teacher. The method has been developed to teach a pick-and-drop task. In this context, the proposed method performs better demonstrations than kinesthesy. The proposed corrected IHMT method is a relevant alternative to kinesthesy for applications that require a certain dexterity.

The heading reset presents a positive impact on the sensor orientation estimation. The heading reset achieves a compensation of the drift in the orientation sensor around the gravity direction when it is not possible to use a magnetometer. This original approach takes advantage of some particular properties of the human motion for task demonstrations.

The IHMT method proposed in this work reaches a similar level of accuracy than other proposed methods in the literature [49] and has also been developed to meet specific requirements of the targeted application. The proposed IHMT method is thus adapted to PbD. It does not require magnetometer data that are too sensitive to disturbances encountered in industrial environments and it exploits some interactions with the robot to improve the accuracy.

The experimental tests conducted to evaluate different factors influencing the IHMT method reveal that the human motion is affected by parasitic motions. Such motions can be torso or shoulder displacements, a deviation from the recommended motion for the initialization, or soft tissue artefacts. Such parasitic motions induce important measurement inaccuracies for the demonstration acquisition method solely based on inertial sensors. However, it is shown that once corrected using a vision-based object tracking, the corrected IHMT demonstrations are suitable for PbD. The proposed corrected method consists in merging inertial data and vision data. The problem that is mostly addressed in the literature is to merge vision measurement and inertial measurement of the same element, for instance the estimation of the drone position from inertial data (double integration of the linear acceleration) with the estimation of the same position from landmark observations. The approach presented here to merge inertial and vision data

differs by merging measurements of different elements: the hand position from inertial measurements and the objects positions from vision measurements. This approach has been developed for programming by demonstration of pick-and-place tasks but it may potentially be applied to other contexts as well.

The impact of the proposed correction method goes beyond improving the level of accuracy of the hand trajectory from the IHMT method. The correction method improves the encoding of the actions of the task by associating a hand position with an object position. The use of pairs of POIs is then an appropriate approach for PbD.

9.3 Perspectives

Programming by demonstration aims at developing intuitive methods to transmit tasks or skills to a robot in a similar way humans interact, i.e., by demonstrating the desired behaviour. PbD has the potential to address challenges of industry 4.0 which is faced with the automation of short series requiring frequent robot programming. With the increasing number of robots in our daily lives, PbD also has the potential to address challenges in the service sector.

The heading reset appears as a promising approach to reduce the drift around the gravity direction. The key aspect of this element is the detection of the moments when it should be applied. The criterion to determine these moments can be further studied as well as its relevance for different applications. Furthermore, one could wonder if some other natural features in human behaviour during demonstrations can be exploited to detect and compensate the drift of the heading angle.

The proposed method has been evaluated on pick-and-place tasks and compared to kinesthetic guidance. Demonstrations where human motion is measured have the potential to reach more complex applications where kinesthesy cannot be applied, in particular, applications where fine manipulation is involved. The proposed method might be adapted to track fingers motions (with extra sensors on the phalanges) and could be tested for such manipulation tasks. In addition, it would be interesting to evaluate the proposed method with other classes of objects and not only cubes. A general object positioning algorithm would then be required.

The torso motion could also be tracked using an additional IMU for a better user-friendliness for the teacher. This has been tested in an early attempt. The teacher sits and the model is enriched to account for the rotation of the torso considered as a rigid segment connected to a fixed support by a spherical joint. An IMU, set on the chest of the teacher, is used to track the torso segment orientation. Unfortunately, the results were not conclusive and further investigations could be conducted to better model and measure the torso motion.

One taxonomy of the *learning the task* methods opposes low-level and high-level representations of the task [34]. The high-level representation deals with a set of symbolic elements of a task such as “pick-up”, “screw” or “move”. The method developed and implemented in this work belongs to the low-level category. Methods in such categories describe the task with, for instance, trajectories. On the other hand, inertial sensors seem particularly suitable for human gestures recognition as in [84, 134]. Without modifying the IHMT equipment, additional high-level information could be extracted from IMUs signals in order to complement the low-level

FIGURE 9.1: Robot teleoperation using the proposed IHMT method.

representation of a demonstration. An early work on this problem is presented in [93]. A first step could be to manage pointing gestures as in [25] during demonstrations to indicate a pick or a place position non-reachable by the teacher.

The POIs extracted to correct the hand trajectory are also used in the implementation of the *learning the task* algorithm. In [62], the demonstrations only include key positions. The representation of the demonstrations with specific positions and how they are related to each other could be investigated for PbD.

Our IHMT method was developed for the purpose of demonstration acquisition. Nevertheless, it may also be used in a wider context. For example, some initial tests have been performed to exploit the method for teleoperation. As can be seen in Figure 9.1, the motions of the operator are recorded and transmitted to the robot which replicates in real-time these motions. In the illustration, the operator drives the robot to grasp a ball and insert it in a funnel. More details on this application are given in Appendix C. Such teleoperation setups can be exploited for various purposes including demonstration acquisition in a PbD context.

APPENDIX A

THREE ALGORITHMS FOR IMU ORIENTATION ESTIMATION

Contents

A.1 Notations

The algorithms presented in the following have two versions: the MARG version when considering magnetometer data and the IMU version when considering only the gyroscope and the accelerometer data. Frame 0 denotes the inertial reference frame for the orientation of the sensor. It follows the ENU convention (**y**-axis pointing East, **x**-axis pointing North and **z**-axis pointing Up) for the MARG version. In the case of the IMU version of the algorithm, the **z**-axis of the reference frame is pointing up and the **y**-axis and the **x**-axis are defined by the initial sensor frame. Frames n and $n-1$ are the sensor frame at time step n and $n-1$ respectively.

${}^n\boldsymbol{\omega}_n^{\text{gyro}}$, ${}^n\mathbf{a}_n$ and ${}^n\mathbf{m}_n^{\text{mag}}$ denote respectively the gyroscope, accelerometer and magnetometer measurement at the time step n with respect to the sensor frame n.

The algorithms presented here use the vector ${}^0\mathbf{g}$, a unit vector representing the gravity direction with respect to the reference frame. The **z**-axis of the reference frame being defined as aligned with the gravity, ${}^0\mathbf{g} = [0\ 0\ -1]$. ${}^n\mathbf{g}$ is a unit vector estimating the gravity direction with respect to the sensor frame at time step n. g is the gravity constant at the Earth surface. $\mathbf{m}^{earth}$ is the local Earth magnetic vector and δ is the inclination angle of the Earth magnetic field. $\mathbf{m}^{earth}$ and δ are determined by the World Magnetic Model from the National Oceanic and Atmospheric Administration. It can be mentioned that $\mathbf{m}^{earth}$ is only irrelevant for the MARG version of the algorithms. ${}^n\mathbf{m}$ and ${}^0\mathbf{m}$ are unit vectors estimating the direction of the local magnetic vector.

The unit quaternion ${}^0\hat{q}_n$ denotes the rotation from frame n to frame 0 and ${}^n\hat{q}_0 = ({}^0\hat{q}_n)^{-1}$. The cross product is noted $\times$. The quaternion product is represented by $\otimes$. The Euclidean norm is noted $\|.\|$. $\hat{v} = [0\ v_x\ v_y\ v_z]^T$ is the quaternion representation of a vector $\mathbf{v} = [v_x\ v_y\ v_z]^T$. The operator $\tilde{\mathbf{v}}$ represents the skew symmetric matrix associated to the vector $\mathbf{v}$.

$$\tilde{\mathbf{v}} = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix}.$$

A.2 Fourati's Algorithm

The Fourati's algorithm is presented in Algorithm 5. The values for the parameters in the Fourati's algorithm are defined in Algorithm 6 for the MARG version and in Algorithm 7 for the IMU version.

Algorithm 5 : Fourati's algorithm [53].

$^{n}\mathbf{g} = \frac{^{n}\mathbf{a}_n}{\|^{n}\mathbf{a}_n\|}$
$^{n}\mathbf{m} = \frac{^{n}\mathbf{m}_n^{\text{mag}}}{\|^{n}\mathbf{m}_n^{\text{mag}}\|}$
$^{0}\mathbf{m} = [\cos(\delta) \;\; 0 \;\; -\sin(\delta)]^T$
$^{n-1}\hat{g} = {}^{0}\hat{q}_{n-1}^{-1} \otimes {}^{0}\hat{g} \otimes {}^{0}\hat{q}_{n-1}$
$^{n-1}\hat{m} = {}^{0}\hat{q}_{n-1}^{-1} \otimes {}^{0}\hat{m} \otimes {}^{0}\hat{q}_{n-1}$
$X = -2\left(^{n-1}\tilde{\mathbf{g}} \;\; ^{n-1}\tilde{\mathbf{m}}\right)^T$ for the MARG version
$X = -2\left(^{n-1}\tilde{\mathbf{g}}\right)^T$ for the IMU version
$W = -2\left(k_a\,^{n-1}\tilde{\mathbf{g}} \;\; k_m\,^{n-1}\tilde{\mathbf{m}}\right)^T$ for the MARG version
$W = -2\left(k_a\,^{n-1}\tilde{\mathbf{g}}\right)^T$ for the IMU version
$K = (X^T X + 10^{-5} I_3)^{-1} W^T$
$e = \begin{pmatrix} 1 \\ K \begin{pmatrix} ^{n-1}\mathbf{g} - {}^{n}\mathbf{g} \\ ^{n-1}\mathbf{m} - {}^{n}\mathbf{m} \end{pmatrix} \end{pmatrix}$ for the MARG version
$e = \begin{pmatrix} 1 \\ K \begin{pmatrix} ^{n-1}\mathbf{g} - {}^{n}\mathbf{g} \end{pmatrix} \end{pmatrix}$ for the IMU version
$^{0}\hat{q}_n = {}^{0}\hat{q}_{n-1} + \frac{1}{2}{}^{0}\hat{q}_{n-1} \otimes {}^{n}\hat{\omega}_n^{\text{gyro}} \otimes e$
$^{0}\hat{q}_n = \frac{^{0}\hat{q}_n}{\|^{0}\hat{q}_n\|}$

Algorithm 6 : parameters values for the Fourati's algorithm (MARG version).

if $|\|^{n}\mathbf{a}_n\| - 1| > 0.1$ **and** $|\|^{n}\mathbf{m}_n^{\text{mag}}\| - \|\mathbf{m}^{earth}\|| < 5$ **then**
  $k_a = 0.1$, $k_m = 0.9$
else if $|\|^{n}\mathbf{a}_n\| - g| < 0.1$ **and** $|\|^{n}m_n^{mag}\| - \|\mathbf{m}^{earth}\|| > 5$ **then**
  $k_a = 0.9$, $k_m = 0.1$
else
  $k_a = 0.5$, $k_m = 0.5$
end if

Algorithm 7 : parameter values for the Fourati's algorithm (IMU version).

if $|\|^{n}\mathbf{a}_n\| - 1| > 0.1$ **then**
  $k_a = 0.1$
else
  $k_a = 0.9$
end if

A.3 Mahony's algorithm

Algorithm 8 : Mahony's algorithm [89].

$^{n}\mathbf{g} = \frac{^{n}\mathbf{a}_n}{\|^{n}\mathbf{a}_n\|}$
$^{n}\mathbf{m} = \frac{^{n}\mathbf{m}_n^{\text{mag}}}{\|^{n}\mathbf{m}_n^{\text{mag}}\|}$
$^{n-1}\hat{g} = {}^{0}\hat{q}_{n-1}^{-1} \otimes {}^{0}\hat{g} \otimes {}^{0}\hat{q}_{n-1}$
$^{n-1}\hat{m} = {}^{0}\hat{q}_{n-1}^{-1} \otimes {}^{0}\hat{m} \otimes {}^{0}\hat{q}_{n-1}$
$\mathbf{e}_p = k_1(^{n}\mathbf{g} \times {}^{n-1}\mathbf{g}) + k_2(^{n}\mathbf{m} \times {}^{n-1}\mathbf{m})$
$\mathbf{e}_i = \int_0^n \Delta t \mathbf{e}_p$
$^{n}\boldsymbol{\omega} = {}^{n}\boldsymbol{\omega}_n^{gyro} + k_p\mathbf{e}_p + k_i\mathbf{e}_i$
${}^{0}\hat{q}_n = {}^{0}\hat{q}_{n-1} + \frac{1}{2}{}^{0}\hat{q}_n \otimes {}^{n}\hat{\boldsymbol{\omega}}$
$\hat{q}_n = \frac{\hat{q}_n}{\|\hat{q}_n\|}$

The proportional k_p and integral k_i gains of the Mahony's algorithm have been respectively set up to 2 and 0.005 according to author recommendations [89]. In order to consider magnetometer measurements, the values for k_1 and k_2 have to be set to 1. But to use this algorithm only with accelerometers and gyroscope, the values should be $k_1 = 1$ and $k_2 = 0$.

A.4 Madgewick's algorithm

Algorithm 9 : Madgwick's algorithm [88].

$^{n}\mathbf{g} = \frac{^{n}\mathbf{a}_n}{\|^{n}\mathbf{a}_n\|}$
$^{n}\mathbf{m} = \frac{^{n}\mathbf{m}_n^{\text{mag}}}{\|^{n}\mathbf{m}_n^{\text{mag}}\|}$
$^{n-1}\hat{g} = {}^{0}\hat{q}_{n-1}^{-1} \otimes {}^{0}\hat{g} \otimes {}^{0}\hat{q}_{n-1}$
$^{n-1}\hat{m} = {}^{0}\hat{q}_{n-1}^{-1} \otimes {}^{0}\hat{m} \otimes {}^{0}\hat{q}_{n-1}$
$F = \begin{pmatrix} ^{n-1}\mathbf{g} - {}^{n}\mathbf{g} \\ ^{n-1}\mathbf{m} - {}^{n}\mathbf{m} \end{pmatrix}$ for the MARG version
$F = \begin{pmatrix} ^{n-1}\mathbf{g} - {}^{n}\mathbf{g} \end{pmatrix}$ for the IMU version
$\mathbf{e} = J^T F$ with J the jacobian matrix of F
$\mathbf{e} = \frac{\mathbf{e}}{\|\mathbf{e}\|}$
$\hat{q}_n = \hat{q}_{n-1} + \frac{1}{2}\hat{q}_{n-1} \otimes (^{n}\boldsymbol{\omega}_n^{gyro} - \beta\hat{e})$
$\hat{q}_n = \frac{\hat{q}_n}{\|\hat{q}_n\|}$

According to the author recommendation $\beta = \sqrt{\frac{3}{4}} \times \omega_\beta$ with ω_β the norm of the zero mean gyroscope measurement error.

APPENDIX B

IMU ORIENTATION ESTIMATION WITH AND WITHOUT A MAGNETOMETER IN A ROBOTIC ENVIRONMENT

Contents

B.1 Experimental Setup

Three Xsens MTw sensors have been mounted on an ABB IRB 120 robotic arm, see Figure B.1. The three algorithms presented in Appendix A are applied on the calibrated data of the Xsens sensors. The orientation of the end-effector of the robot measured by encoders is used as reference. The trajectory applied to the robot is composed of 4 rotations of +100°, +60°, -60° and -100° around the **x**, **y** and **z** axes of the robot base frame, 3 translations along these axes and a random motion. This trajectory is repeated 5 times at different maximum speeds of the robot end-effector: 500 mm/s, 500 °/s; 300 mm/s, 300 °/s; 50 mm/s, 50 °/s; 300 mm/s, 300 °/s and 500 mm/s, 500 °/s. Each measurement lasts 10 minutes and 39 seconds.

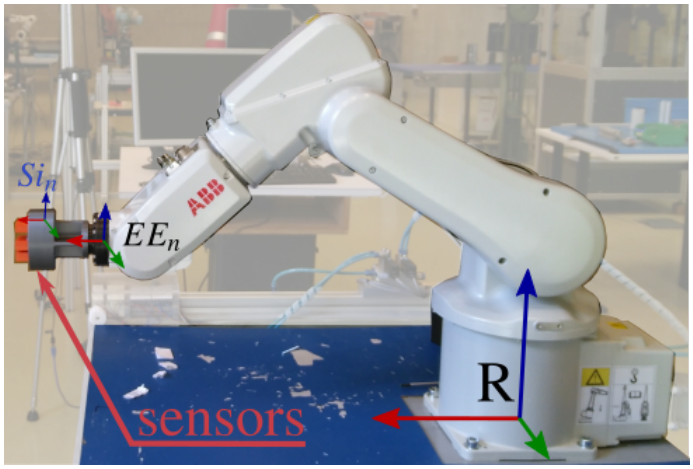

FIGURE B.1: Experimental setup.

B.2 Results

The results are shown in Table B.1. The table presents the results of both IMU and MARG version of each algorithm. The error ε is the norm of the rotation vector θ describing the rotation from the estimated orientation from sensor data $\hat{q}^{inertial}$ and the robot's end-effector orientation measured from encoders $\hat{q}^{robot}$:

$$\theta = \hat{q}^{inertial} \otimes (\hat{q}^{robot})^{-1} \tag{B.1}$$

$$\varepsilon = \|\theta\| \tag{B.2}$$

The green and the red cells show respectively the best and the worst estimation between the MARG and the IMU versions of the same algorithm. Regarding the Mahony's algorithm, it is clear that the use of the magnetometer has a negative impact on the sensor estimation orientation close to a robot. Regarding the Madgwick's and the Fourati's algorithm, the contribution of the magnetometer sometimes degrade the sensor orientation estimation. It can be considered that using a MARG algorithm for sensor orientation estimation close to a robot does not give a reliable result.

Measurement	Sensors	The Mahony's algorithm		The Madgwick's algorithm		The Fourati's algorithm	
		MARG	IMU	MARG	IMU	MARG	IMU
1	1	26.2	5.8	15.8	8.6	7.9	15.9
	2	21.9	2.5	4.3	3.3	7.3	4.6
	3	21.3	12.0	20.8	21.6	28.1	17.4
2	1	21.8	2.3	4.4	6.9	7.3	6.3
	2	26.2	3.6	15.9	8.3	8.0	16.8
	3	21.4	9.1	17.4	20.1	24.1	14.6
3	1	21.8	3.9	5.5	12.5	7.0	8.4
	2	21.4	8.7	18.9	20.7	26.7	15.8
	3	26.2	5.5	14.3	8.1	6.0	13.0
4	1	23.0	3.9	19.1	20.1	20.7	15.8
	2	27.0	12.9	15.9	9.4	11.0	14.1
	3	23.6	7.0	8.3	14.7	4.9	7.2
5	1	27.0	8.7	15.8	9.0	11.9	16.5
	2	23.0	3.9	20.6	24.0	20.8	15.9
	3	23.6	9.2	8.6	16.2	4.5	6.4
6	1	27.0	8.4	15.6	9.3	12.8	14.5
	2	23.6	3.9	11.0	19.7	7.2	6.6
	3	23.0	3.7	17.4	19.1	19.7	13.8

TABLE B.1: Comparison between the IMU and the MARG version of three algorithms.

APPENDIX C

EVALUATION OF THE HANDINESS OF THE IHMT METHOD FOR ROBOT TELEOPERATION

Contents

C.1 Experimental setup

The experimental setup is based on the Sawyer robot from Rethink Robotics (see Figure 9.1). The three IMU sensors transmit data by radio frequency to a regular desktop computed, which host the numerical implementation of the method. The desktop computer and the robot's controller communicate via a local ROS-based network.

The task considered is a simple pick-and-place, which consists in picking a squash ball from a little container and drop it inside a funnel. After the funnel, a tunnel guide the ball back to its original container. The task is successfully completed if the ball goes back at its original location in the container.

Seven subjects are asked to perform this task in two manners, with the keyboard and with the IHMT teleoperation method. Each subject is asked to do 5 successes with each method. The number of trials is recorded as well as the time used to complete each individual success. The subjects are aged from 29 to 36 years old. The group is constituted of 2 women and 5 men with robotics expertise from "none" to "expert".

For IHMT teleoperation, the subject faces the robot and stands approximately 0.5 m away from the table. Once the subject has performed the rotating motion and the static pose of the initialization procedure, the robot's gripper is controlled to follow the subject's hand. If the subject performs a motion of 10 cm on the left, the robot does the same with a negligible delay. The orientation of the gripper is also controlled by the orientation of the subject's hand. Once the robot is teleoperated, the subject must stand upright, only his/her arm is moving. For keyboard teleoperation, the subject sits at the right front of the robot. The robot starts in a central position, gripper facing down. The subject can move freely.

C.2 Results and discussion

Figure C.1 presents the duration of each of the 5 successes for both teleoperation methods. Globally, the IHMT teleoperation method requires less time to complete the task which tends to confirm that this method is more convenient for robot teleoperation than the conventional teleoperation methods. However, the results for keyboard teleoperation reveal a learning effect as identified in [13, 98]. For almost all subjects, the time to complete task decreases with the number of trials. It is not the case for the IHMT teleoperation method. So the advantage of such method could be less marked after some training.

Table C.1 shows the number of trials to reach 5 successes with both methods. The number of failures is higher for the IHMT teleoperation method for all but 2 subjects for which it is the same. In addition, the number of trials for the keyboard method is 5 or 6, meaning that almost no failure occurred when using this method. This tends to reveal a difficulty to handle the proposed IHMT method. However, as with the time analysis, a learning effect can be identified regarding the success rate. The success rate is defined as the number of successes divided by the number of trials. It can be computed after each trial. Figure C.2 presents the evolution of the success rate for each subject. Subjects 4, 5 and 7 clearly show a learning effect since their success rate increases with the number of trials. Subjects 1 and 2 also present a learning effect except for their first trial which is a success. Subject 6 does not present a learning effect but no failure occurs. Subject 3

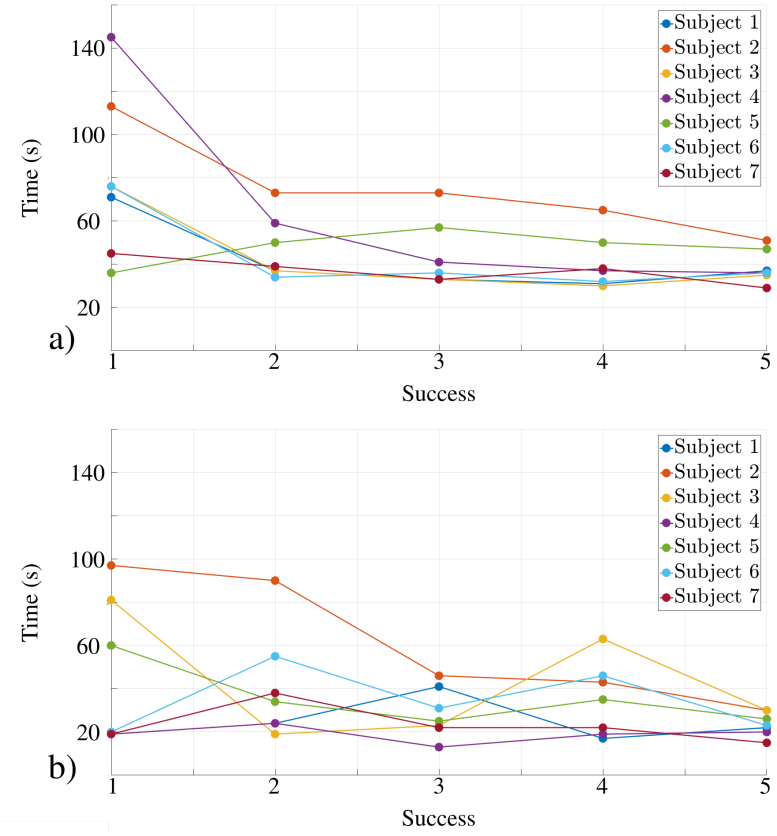

FIGURE C.1: Time for each individual success for the keyboard method a) and for the IHMT method b).

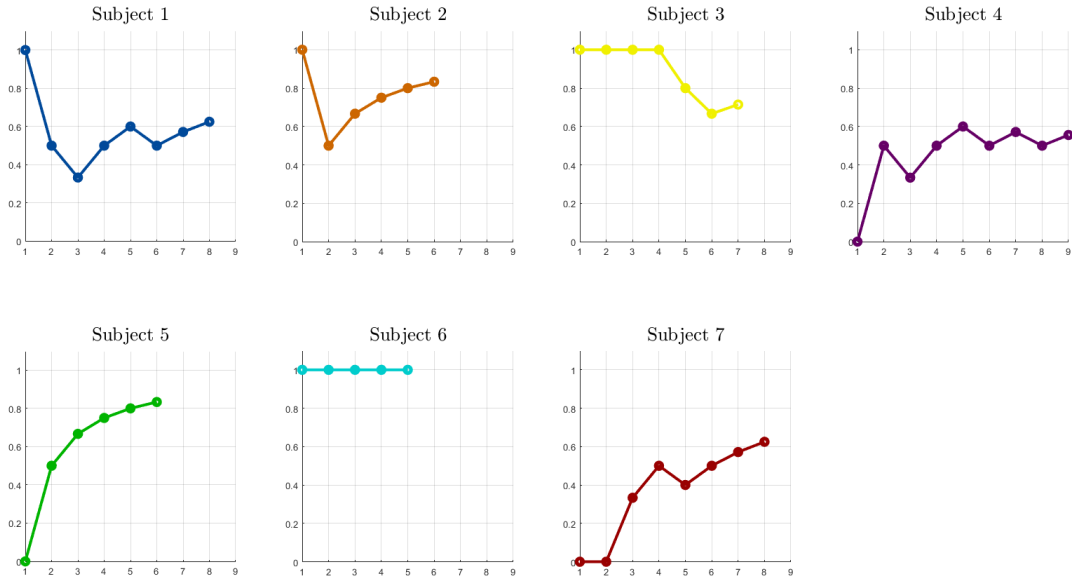

FIGURE C.2: Success rate for each subject using the IHMT method.

does not present any learning effect. It can be expected that a potential difficulty to handle the IHMT method could be overcome with some practice.

Subject	Methods	Number of trials	Success 1 time (s)	Success 2 time (s)	Success 3 time (s)	Success 4 time (s)	Success 5 time (s)
1	keyboard	6	71	37	33	31	37
	IHMT	8	19	24	41	17	22
2	keyboard	6	113	73	73	65	51
	IHMT	6	97	90	46	43	30
3	keyboard	5	76	37	33	30	35
	IHMT	7	81	19	23	63	30
4	keyboard	5	145	59	41	37	36
	IHMT	9	19	24	13	19	20
5	keyboard	5	36	50	57	50	47
	IHMT	6	60	34	25	35	26
6	keyboard	5	76	34	36	32	36
	IHMT	5	20	55	31	46	23
7	keyboard	5	45	39	33	38	29
	IHMT	8	19	38	22	22	15

TABLE C.1: Results of the execution of the pick-and-place by teleoperation from the IHMT and from the keyboard method.

References

[1] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. “Autonomous helicopter aerobatics through apprenticeship learning”. In: *International Journal of Robotics Research* 29.13 (2010), pp. 1608–1639. ISSN: 02783649. DOI: 10.1177/0278364910371999.

[2] Shaikh Shakil Abdul Rajjak and A. K. Kureshi. “Recent advances in object detection and tracking for high resolution video: Overview and state-of-the-art”. In: *Proceedings - 2019 5th International Conference on Computing, Communication Control and Automation, ICCUBEA 2019* (2019). DOI: 10.1109/ICCUBEA47591.2019.9128812.

[3] Celia Amabile, Anthony M.J. Bull, and Angela E. Kedgley. “The centre of rotation of the shoulder complex and the effect of normalisation”. In: *Journal of Biomechanics* 49.9 (2016), pp. 1938–1943. ISSN: 18732380. DOI: 10.1016/j.jbiomech.2016.03.035.

[4] Amineh Amini, Teh Ying Wah, and Hadi Saboohi. “On density-based data streams clustering algorithms: A survey”. In: *Journal of Computer Science and Technology* 29.1 (2014), pp. 116–141. ISSN: 10009000. DOI: 10.1007/s11390-014-1416-y.

[5] António Amorim et al. “Robust human position estimation in cooperative robotic cells”. In: *Robotics and Computer-Integrated Manufacturing* 67 (2021). ISSN: 07365845. DOI: 10.1016/j.rcim.2020.102035.

[6] Michael S. Andrle and John L. Crassidis. “Geometric integration of quaternions”. In: *Journal of Guidance, Control, and Dynamics* 36.6 (2013), pp. 1762–1767. ISSN: 15333884. DOI: 10.2514/1.58558.

[7] Brenna D. Argall, Eric L. Sauser, and Aude G. Billard. “Tactile guidance for policy refinement and reuse”. In: *2010 IEEE 9th International Conference on Development and Learning, ICDL-2010 - Conference Program* (2010), pp. 7–12. DOI: 10.1109/DEVLRN.2010.5578872.

[8] Brenna D. Argall et al. “A survey of robot learning from demonstration”. In: *Robotics and Autonomous Systems* 57.5 (2009), pp. 469–483. ISSN: 09218890. DOI: 10.1016/j.robot.2008.10.024.

[9] K. S. Arun, T. S. Huang, and S. D. Blostein. “Least-Squares Fitting of Two 3-D Point Sets”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9.5 (1987), pp. 698–700. ISSN: 01628828. DOI: 10.1109/TPAMI.1987.4767965.

[10] Eric R. Bachmann et al. “Inertial and magnetic posture tracking for inserting humans into networked virtual environments”. In: *ACM Symposium on Virtual Reality Software and Technology, Proceedings, VRST* November (2001), pp. 9–16. DOI: 10.1145/505009.505011.

[11] Jordi Bautista Ballester, Jaume Vergés Llahí, and Domènec Puig. “Programming by demonstration: A taxonomy of current relevant methods to teach and describe new skills to robots”. In: *Advances in Intelligent Systems and Computing* 252 (2014), pp. 287–300. ISSN: 21945357. DOI: 10.1007/978-3-319-03413-3_21.

[12] Andrea Bandini and Jose Zariffa. "Analysis of the hands in egocentric vision: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1. ISSN: 0162-8828. DOI: 10.1109/tpami.2020.2986648. arXiv: 1912.10867.

[13] Aljaz Baumkircher, Marko Munih, and Matjaz Mihelj. "Performance Analysis of Learning from Demonstration Approaches during a Fine Movement Generation". In: *IEEE Transactions on Human-Machine Systems* 51.6 (2021), pp. 653–662. ISSN: 21682305. DOI: 10.1109/THMS.2021.3107523.

[14] Asma Belhadi et al. "Space–time series clustering: Algorithms, taxonomy, and case study on urban smart cities". In: *Engineering Applications of Artificial Intelligence* 95 (2020). ISSN: 09521976. DOI: 10.1016/j.engappai.2020.103857.

[15] Giovanni Bellusci, Fred Dijkstra, and Per Slycke. "Xsens MTw : Miniature Wireless Inertial Motion Tracker for Highly Accurate 3D Kinematic Applications". In: *Xsens Technologies*. April. 2013, pp. 1–9. DOI: 10.13140/RG.2.2.23576.49929.

[16] Serge Belongie, Jitendra Malik, and Jan Puzicha. "Shape context: A new descriptor for shape matching and object recognition". In: *Advances in Neural Information Processing Systems* 13.July (2001), pp. 831–837. ISSN: 10495258.

[17] Tadej Beravs, Janez Podobnik, and Marko Munih. "Three-axial accelerometer calibration using kalman filter covariance matrix for online estimation of optimal sensor orientation". In: *IEEE Transactions on Instrumentation and Measurement* 61.9 (2012), pp. 2501–2511. ISSN: 00189456. DOI: 10.1109/TIM.2012.2187360.

[18] James C. Bezdek. "Pattern Recognition with Fuzzy Objective Function Algorithms". In: *Pattern Recognition with Fuzzy Objective Function Algorithms* (1981). DOI: 10.1007/978-1-4757-0450-1.

[19] Panthadeep Bhattacharjee and Pinaki Mitra. "A survey of density based clustering algorithms". In: *Frontiers of Computer Science* 15.1 (2021). ISSN: 20952236. DOI: 10.1007/s11704-019-9059-3.

[20] Aude Billard et al. "Robot Programming be Demonstration". In: *Siciliano, B., Khatib, O. (eds) Springer Handbook of Robotics*. Springer, 2008, pp. 1371–1394. DOI: 10.1007/978-3-540-30301-5_60.

[21] Aude G. Billard, Sylvain Calinon, and Rüdiger Dillmann. "Learning from Humans". In: *Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. 2nd. Berlin: Springer, 2016, pp. 1–2227. ISBN: 9783319325521. DOI: 10.1007/978-3-319-32552-1.

[22] Gabriele Bleser, Gustaf Hendeby, and Markus Miezal. "Using egocentric vision to achieve robust inertial body tracking under magnetic disturbances". In: *2011 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011* October (2011), pp. 103–109. DOI: 10.1109/ISMAR.2011.6092528.

[23] Hans-Hermann Bock. "Clustering Methods: A History of k-Means Algorithms". In: *Selected Contributions in Data Analysis and Classification*. Ed. by Springer. Springer, Berlin, Heidelberg, 2007, pp. 161–172. ISBN: 978-3-540-73558-8. DOI: 10.1007/978-3-540-73560-1_15.

[24] Sara Bouraya and Abdessamad Belangour. "Multi object tracking: a survey". In: *Proceedings of SPIE - The International Society for Optical Engineering*. Vol. 11878. 2021, p. 96. ISBN: 9781510646001. DOI: 10.1117/12.2602901.

[25] Denis Broggini et al. “Learning to detect pointing gestures from wearable IMUS”. In: *32nd AAAI Conference on Artificial Intelligence, AAAI 2018* 0 (2018), pp. 8051–8052. ISSN: 2159-5399. DOI: 10.1609/aaai.v32i1.12143.

[26] Hans-Peter Brückner et al. “Evaluation of inertial sensor fusion algorithms in grasping tasks using real input data: Comparison of computational costs and root mean square error”. In: *Proceedings - BSN 2012: 9th International Workshop on Wearable and Implantable Body Sensor Networks* (2012), pp. 189–194. DOI: 10.1109/BSN.2012.9.

[27] M. A. Buckley et al. “Dynamics of the upper limb during performance of the tasks of everyday living - A review of the current knowledge base”. In: *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*. Vol. 210. 4. 1996, pp. 241–245. DOI: 10.1243/pime_proc_1996_210_420_02.

[28] Sylvain Calinon. “Learning from Demonstration (Programming by Demonstration)”. In: *In: Ang, M., Khatib, O., Siciliano, B. (eds) Encyclopedia of Robotics.* Springer, Berlin, Heidelberg, 2018, pp. 1040–1046. DOI: 10.1007/978-3-642-41610-1_27-1.

[29] Sylvain Calinon. *Robot programming by demonstration : a probabilistic approach*. Ed. by EPFL Press English Imprint. 1ère éditi. 2009, p. 222. ISBN: 1439808678.

[30] Sylvain Calinon and Aude Billard. “Incremental Learning of Gestures by Imitation in a Humanoid Robot”. In: *2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2007, pp. 255–262. DOI: 10.1017/CBO9780511489808.012.

[31] Sylvain Calinon, Florent Guenter, and Aude Billard. “On learning, representing, and generalizing a task in a humanoid robot”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37.2 (2007), pp. 286–298. ISSN: 10834419. DOI: 10.1109/TSMCB.2006.886952.

[32] Amity Cree Campbell et al. “Effects of different technical coordinate system definitions on the three dimensional representation of the glenohumeral joint centre”. In: *Medical and Biological Engineering and Computing* 47.5 SPEC. ISS. (2009), pp. 543–550. ISSN: 01400118. DOI: 10.1007/s11517-009-0467-7.

[33] Peng Cheng and Bengt Oelmann. “Joint-angle measurement using accelerometers and gyroscopes - A survey”. In: *IEEE Transactions on Instrumentation and Measurement* 59.2 (2010), pp. 404–414. ISSN: 00189456. DOI: 10.1109/TIM.2009.2024367.

[34] Sonia Chernova and Andrea L. Thomaz. *Robot learning from human teachers*. Vol. 28. 2014, pp. 1–121. DOI: 10.2200/S00568ED1V01Y201402AIM028.

[35] Daniel Choukroun, Itzhack Y. Bar-Itzhack, and Yaakov Oshman. “Novel quaternion Kalman filter”. In: *IEEE Transactions on Aerospace and Electronic Systems* 42.1 (2006), pp. 174–190. ISSN: 00189251. DOI: 10.1109/TAES.2006.1603413.

[36] Julien Clément, Maxime Raison, and Dominique M. Rouleau. “Reproducibility analysis of upper limbs reachable workspace, and effects of acquisition protocol, sex and hand dominancy”. In: *Journal of Biomechanics* 68 (2018), pp. 58–64. ISSN: 18732380. DOI: 10.1016/j.jbiomech.2017.12.010.

[37] Camilo Cortés et al. “Upper limb posture estimation in robotic and virtual reality-based rehabilitation”. In: *BioMed Research International* 2014.3 (2014). ISSN: 23146141. DOI: 10.1155/2014/821908.

[38] M. Daszykowski and B. Walczak. “Density-Based Clustering Methods”. In: *Comprehensive Chemometrics*. Vol. 2. 2009, pp. 635–654. ISBN: 9780444527011. DOI: 10.1016/B978-044452701-1.00067-3.

[39] Nathan Deom and Olivier Brüls. “Programmation robotique par démonstration Enregistrement optique de trajectoires par marqueurs”. Matser Thesis. Université de Liège, 2018, p. 101.

[40] E. Forgy. “Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications”. In: *Biometrics* 21 (1965), pp. 768–780.

[41] M. Ehrenmann et al. “Programming service tasks in household environments by human demonstration”. In: *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication* (2002), pp. 460–467. DOI: 10.1109/ROMAN.2002.1045665.

[42] Mahmoud El-Gohary and James McNames. “Human Joint Angle Estimation with Inertial Sensors and Validation with A Robot Arm”. In: *IEEE Transactions on Biomedical Engineering* 62.7 (2015), pp. 1759–1767. DOI: 10.1109/TBME.2015.2403368.

[43] Sarah Elliott, Zhe Xu, and Maya Cakmak. “Learning generalizable surface cleaning actions from demonstration”. In: *RO-MAN 2017 - 26th IEEE International Symposium on Robot and Human Interactive Communication* 2017-Janua (2017), pp. 993–999. DOI: 10.1109/ROMAN.2017.8172424.

[44] Mark Euston et al. “A complementary filter for attitude estimation of a fixed-wing UAV”. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* (2008), pp. 340–345. DOI: 10.1109/IROS.2008.4650766.

[45] Absalom E. Ezugwu et al. “A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects”. In: *Engineering Applications of Artificial Intelligence* 110.January (2022), p. 104743. ISSN: 09521976. DOI: 10.1016/j.engappai.2022.104743.

[46] Marcos Ferreira et al. “Stereo-based real-time 6-DoF work tool tracking for robot programing by demonstration”. In: *The International Journal of Advanced Manufacturing Technology* 85 (2014), pp. 1–4. DOI: 10.1109/OFC.2003.1248319.

[47] Matthew Field et al. “Motion capture in robotics review”. In: *2009 IEEE International Conference on Control and Automation, ICCA 2009* (2009), pp. 1697–1702. DOI: 10.1109/ICCA.2009.5410185.

[48] Nadia Figueroa, Ana Lucia Pais Ureche, and Aude Billard. “Learning complex sequential tasks from demonstration: A pizza dough rolling case study”. In: *ACM/IEEE International Conference on Human-Robot Interaction* 2016-April (2016), pp. 611–612. ISSN: 21672148. DOI: 10.1109/HRI.2016.7451881.

[49] Alessandro Filippeschi et al. “Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion”. In: *Sensors (Switzerland)* 17.6 (2017), p. 1257. ISSN: 14248220. DOI: 10.3390/s17061257.

[50] Kerstin Fischer et al. “A comparison of types of robot control for programming by demonstration”. In: *ACM/IEEE International Conference on Human-Robot Interaction* 2016-April (2016), pp. 213–220. ISSN: 21672148. DOI: 10.1109/HRI.2016.7451754.

[51] Hassen Fourati. "Heterogeneous data fusion algorithm for pedestrian navigation via foot-mounted inertial measurement unit and complementary filter". In: *IEEE Transactions on Instrumentation and Measurement* 64.1 (2015), pp. 221–229. ISSN: 00189456. DOI: 10.1109/TIM.2014.2335912.

[52] Hassen Fourati et al. "A Nonlinear Filtering Approach for the Attitude and Dynamic Body Acceleration Estimation Based on Inertial and Magnetic Sensors: Bio-Logging Application". In: *IEEE Sensors Journal* 11.1 (2011), pp. 233–244. ISSN: 15581748. DOI: 10.1109/JSEN.2010.2053353.

[53] Hassen Fourati et al. "Complementary observer for body segments motion capturing by inertial and magnetic sensors". In: *IEEE/ASME Transactions on Mechatronics* 19.1 (2014), pp. 149–157. ISSN: 10834435. DOI: 10.1109/TMECH.2012.2225151.

[54] Eric Foxlin. "Pedestrian tracking with shoe-mounted inertial sensors". In: *IEEE Computer Graphics and Applications* 25.6 (2005), pp. 38–46. DOI: 10.1109/MCG.2005.140..

[55] Anthony Gallagher, Yoky Matsuoka, and Wei Tech Ang. "An efficient real-time human posture tracking algorithm using low-cost inertial and magnetic sensors". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 3 (2004), pp. 2967–2972. DOI: 10.1109/iros.2004.1389860.

[56] S. Garrido-Jurado et al. "Automatic generation and detection of highly reliable fiducial markers under occlusion". In: *Pattern Recognition* 47.6 (2014), pp. 2280–2292. ISSN: 00313203. DOI: 10.1016/j.patcog.2014.01.005.

[57] Yu He and Shengyong Chen. "Advances in sensing and processing methods for three-dimensional robot vision". In: *International Journal of Advanced Robotic Systems* 15.2 (2018). ISSN: 17298814. DOI: 10.1177/1729881418760623.

[58] John Henawy et al. "Accurate IMU Factor Using Switched Linear Systems for VIO". In: *IEEE Transactions on Industrial Electronics* 68.8 (2021), pp. 7199–7208. ISSN: 15579948. DOI: 10.1109/TIE.2020.3000097.

[59] Chao Hong et al. "Interactive humanoid robot arm imitation system using human upper limb motion tracking". In: *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2017, pp. 2746–2751. ISBN: 9781538637418. DOI: 10.1109/ROBIO.2017.8324706.

[60] Wenzheng Hu et al. "An inertial sensor system for measurements of tibia angle with applications to knee valgus/varus detection". In: *IEEE ISSNIP 2014 - 2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Conference Proceedings* (2014). DOI: 10.1109/ISSNIP.2014.6827603.

[61] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge. "Comparing Images Using the Hausdorff Distance". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.9 (1993), pp. 850–863. DOI: 10.1109/34.232073..

[62] Julius Jankowski, Mattia Racca, and Sylvain Calinon. "From Key Positions to Optimal Basis Functions for Probabilistic Adaptive Control". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 3242–3249. ISSN: 23773766. DOI: 10.1109/LRA.2022.3146614.

[63] Prayook Jatesiktat and Wei Tech Ang. “Recovery of forearm occluded trajectory in Kinect using a wrist-mounted Inertial Measurement Unit”. In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (2017), pp. 807–812. ISSN: 1557170X. DOI: 10.1109/EMBC.2017.8036947.

[64] Abhishek Jha, Shital S. Chiddarwar, and Mayur V. Andulkar. “An integrated approach for robot training using Kinect and human arm kinematics”. In: *2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015*. Kerala, 2015, pp. 216–221. ISBN: 9781479987917. DOI: 10.1109/ICACCI.2015.7275612.

[65] R. Jonker and A. Volgenant. “A shortest augmenting path algorithm for dense and sparse linear assignment problems”. In: *Computing* 38.4 (1987), pp. 325–340. ISSN: 0010485X. DOI: 10.1007/BF02278710.

[66] George Karypis and Vipin Kumar. “Multilevel k-way hypergraph partitioning”. In: *VLSI Design* 11.3 (2000), pp. 285–300. ISSN: 1065514X. DOI: 10.1155/2000/19436.

[67] Anthony Kim and M. F. Golnaraghi. “A quaternion-based orientation estimation algorithm using an inertial measurement unit”. In: *Record - IEEE PLANS, Position Location and Navigation Symposium* (2004), pp. 268–272. DOI: 10.1109/plans.2004.1309003.

[68] Jaejik Kim and L. Billard. “A polythetic clustering process and cluster validity indexes for histogram-valued objects”. In: *Computational Statistics and Data Analysis* 55.7 (2011), pp. 2250–2262. ISSN: 01679473. DOI: 10.1016/j.csda.2011.01.011.

[69] Nives Klopčar and Jadran Lenarčič. “Kinematic model for determination of human arm reachable workspace”. In: *Meccanica* 40.2 (2005), pp. 203–219. ISSN: 00256455. DOI: 10.1007/s11012-005-3067-0.

[70] Futoshi Kobayashi et al. “Human motion caption with vision and inertial sensors for hand/arm robot teleoperation”. In: *International Journal of Applied Electromagnetics and Mechanics* 52.3-4 (2016), pp. 1629–1636. ISSN: 13835416. DOI: 10.3233/JAE-162140.

[71] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. “An optimization-based approach to human body motion capture using inertial sensors”. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)* 19.3 (2014), pp. 79–85. DOI: 10.3182/20140824-6-za-1003.02252.

[72] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. “Using inertial sensors for position and orientation estimation”. In: *Foundations and Trends in Signal Processing* 11.1-2 (2017), pp. 1–153. ISSN: 19328354. DOI: 10.1561/2000000094. arXiv: 1704.06053.

[73] Petar Kormushev, Sylvain Calinon, and Darwin G. Caldwell. “Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input”. In: *Advanced Robotics* 25.5 (2011), pp. 581–603. DOI: 10.1163/016918611X558261.

[74] Bernhard Krach and Patrick Robertson. “Integration of foot-mounted inertial sensors into a bayesian location estimation framework”. In: *5th Workshop on Positioning, Navigation and Communication 2008, WPNC’08* 2 (2008), pp. 55–61. DOI: 10.1109/WPNC.2008.4510357.

[75] Edgar Kraft. "A quaternion-based unscented Kalman filter for orientation tracking". In: *Proceedings of the 6th International Conference on Information Fusion, FUSION 2003*. Vol. 1. Cairns, QLD, Australia: IEEE, 2003, pp. 47–54. ISBN: 0972184449. DOI: 10.1109/ICIF.2003.177425.

[76] Harold W. Kuhn. *The Hungarian method for the assignment problem*. 1955. DOI: 10.1007/978-3-540-68279-0_2.

[77] Debashis Kushary, Geoffrey J. McLachlan, and Thriyambakam Krishnan. "The EM Algorithm and Extensions". In: *Technometrics* 40.3 (1998), p. 260. ISSN: 00401706. DOI: 10.2307/1271189.

[78] Maria Kyrarini et al. "Human-Robot Synergy for Cooperative Robots". In: *Facta Universitatis, Series: Automatic Control and Robotics* 15.3 (2016), p. 187. ISSN: 1820-6417. DOI: 10.22190/fuacr1603187k.

[79] Daniel Laidig, Philipp Müller, and Thomas Seel. "Automatic anatomical calibration for IMU-based elbow angle measurement in disturbed magnetic fields". In: *Current Directions in Biomedical Engineering* 3.2 (2017), pp. 167–170. ISSN: 23645504. DOI: 10.1515/cdbme-2017-0035.

[80] Daniel Laidig, Thomas Schauer, and Thomas Seel. "Exploiting kinematic constraints to compensate magnetic disturbances when calculating joint angles of approximate hinge joints from orientation estimates of inertial sensors". In: *IEEE International Conference on Rehabilitation Robotics* (2017), pp. 971–976. ISSN: 19457901. DOI: 10.1109/ICORR.2017.8009375.

[81] Jihong Lee and Insoo Ha. "Sensor fusion and calibration for motion captures using accelerometers". In: *Proceedings - IEEE International Conference on Robotics and Automation* 3.May (1999), pp. 1954–1959. ISSN: 10504729. DOI: 10.1109/robot.1999.770394.

[82] Michel A. Lemay and Patrick E. Crago. "A dynamic model for simulating movements of the elbow, forearm, and wrist". In: *Journal of Biomechanics* 29.10 (1996), pp. 1319–1330. ISSN: 00219290. DOI: 10.1016/0021-9290(96)00026-7.

[83] Gabriele Ligorio and Angelo Maria Sabatini. "Dealing with magnetic disturbances in human motion capture: A survey of techniques". In: *Micromachines* 7.3 (2016), p. 43. ISSN: 2072666X. DOI: 10.3390/mi7030043.

[84] João Lopes et al. "Hand/arm Gesture Segmentation by Motion Using IMU and EMG Sensing". In: *Procedia Manufacturing* 11.June (2017), pp. 107–113. ISSN: 23519789. DOI: 10.1016/j.promfg.2017.07.158.

[85] Irvin Hussein Lopez-Nava and Munoz Melendez Angelica. "Wearable Inertial Sensors for Human Motion Analysis: A review". In: *IEEE Sensors Journal* PP.99 (2016), pp. 7821–7834. ISSN: 1530437X. DOI: 10.1109/JSEN.2016.2609392.

[86] H.J. Luinge and P.H. Veltink. "Measuring orientation of human body segments using miniature gyroscopes and accelerometers". In: *Medical and Biological Engineering and Computing* 43.2 (2005), pp. 273–282. ISSN: 01400118. DOI: 10.1007/BF02345966.

[87] H.J. Luinge, P.H. Veltink, and C.T.M. Baten. "Ambulatory measurement of arm orientation". In: *Journal of Biomechanics* 40.1 (2007), pp. 78–85. ISSN: 00219290. DOI: 10.1016/j.jbiomech.2005.11.011.

[88] Sebastian O.H. Madgwick. "An efficient orientation filter for inertial and inertial/magnetic sensor arrays". In: *Report x-io and University of ...* (2010), p. 32. ISSN: 1945-7901.

[89] Robert Mahony, Tarek Hamel, and Jean Michel Pflimlin. "Nonlinear complementary filters on the special orthogonal group". In: *IEEE Transactions on Automatic Control* 53.5 (2008), pp. 1203–1218. ISSN: 00189286. DOI: 10.1109/TAC.2008.923738.

[90] Baraka Maiseli, Yanfeng Gu, and Huijun Gao. "Recent developments and trends in point set registration methods". In: *Journal of Visual Communication and Image Representation* 46 (2017), pp. 95–106. ISSN: 10959076. DOI: 10.1016/j.jvcir.2017.03.012.

[91] F. Landis Markley, John L. Crassidis, and Yang Cheng. "Nonlinear attitude filtering methods". In: *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference* 1 (2005), pp. 753–784. DOI: 10.2514/6.2005-5927.

[92] Philippe Martin and Erwan Salaün. "Design and implementation of a low-cost observer-based attitude and heading reference system". In: *Control Engineering Practice* 18.7 (2010), pp. 712–722. ISSN: 09670661. DOI: 10.1016/j.conengprac.2010.01.012.

[93] Franziska Meier, Evangelos Theodorou, and Stefan Schaal. "Movement segmentation and recognition for imitation learning". In: *Journal of Machine Learning Research* 22 (2012), pp. 761–769. ISSN: 15337928.

[94] Thibaud Michel et al. "A comparative analysis of attitude estimation for pedestrian navigation with smartphones". In: *2015 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2015* (2015). DOI: 10.1109/IPIN.2015.7346767.

[95] Markus Miezal, Bertram Taetz, and Gabriele Bleser. "On inertial body tracking in the presence of model calibration errors". In: *Sensors (Switzerland)* 16.7 (2016), p. 1132. ISSN: 14248220. DOI: 10.3390/s16071132.

[96] Nathan Miller et al. "Motion capture from inertial sensing for untethered humanoid teleoperation". In: *2004 4th IEEE-RAS International Conference on Humanoid Robots* 2 (2004), pp. 547–565. DOI: 10.1109/ichr.2004.1442670.

[97] Mousa Moradi et al. "Integrating Human Hand Gestures with Vision Based Feedback Controller to Navigate a Virtual Robotic Arm". In: *2020 23rd IEEE International Symposium on Measurement and Control in Robotics, ISMCR 2020*. 2020, pp. 1–6. ISBN: 9781665404792. DOI: 10.1109/ISMCR51255.2020.9263771.

[98] Arne Muxfeldt, Jan Henrik Kluth, and Daniel Kubus. "Kinesthetic teaching in assembly operations – a user study". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8810.October (2014), pp. 533–544. ISSN: 16113349. DOI: 10.1007/978-3-319-11900-7_45.

[99] Gabriel Nützi et al. "Fusion of IMU and vision for absolute scale estimation in monocular SLAM". In: *Journal of Intelligent and Robotic Systems: Theory and Applications* 61.1-4 (2011), pp. 287–299. ISSN: 09210296. DOI: 10.1007/s10846-010-9490-z.

[100] Hanna E. Nyqvist et al. "Pose estimation using monocular vision and inertial sensors aided with ultra wide band". In: *2015 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2015* (2015). DOI: 10.1109/IPIN.2015.7346940.

[101] Lauro Ojeda and Johann Borenstein. "Non-GPS navigation for security personnel and first responders". In: *Journal of Navigation* 60.3 (2007), pp. 391–407. ISSN: 03734633. DOI: 10.1017/S0373463307004286.

[102] Alberto Olivares et al. "Detection of (In)activity periods in human body motion using inertial sensors: A comparative study". In: *Sensors (Switzerland)* 12.5 (2012), pp. 5791–5814. ISSN: 14248220. DOI: 10.3390/s120505791.

[103] Mahamed G.H. Omran, Andries P. Engelbrecht, and Ayed Salman. "An overview of clustering methods". In: *Intelligent Data Analysis* 11.6 (2007), pp. 583–605. ISSN: 15714128. DOI: 10.3233/ida-2007-11602.

[104] Eduardo Palermo et al. "Experimental evaluation of accuracy and repeatability of a novel body-to-sensor calibration procedure for inertial sensor-based gait analysis". In: *Measurement: Journal of the International Measurement Confederation* 52.1 (2014), pp. 145–155. ISSN: 02632241. DOI: 10.1016/j.measurement.2014.03.004.

[105] Jun Jun Pan and Yan Ning Zhang. "Corresponding points matching based on position similarity". In: *Proceedings of the Conference on Computer Graphics, Imaging and Vision: New Trends 2005* 2005 (2005), pp. 33–37. DOI: 10.1109/CGIV.2005.29.

[106] Lorenzo Peppoloni et al. "A novel 7 degrees of freedom model for upper limb kinematic reconstruction based on wearable sensors". In: *SISY 2013 - IEEE 11th International Symposium on Intelligent Systems and Informatics, Proceedings* (2013), pp. 105–110. DOI: 10.1109/SISY.2013.6662551.

[107] Vitor H. Pinto et al. "Enhanced performance real-time industrial robot programming by demonstration using stereoscopic vision and an IMU sensor". In: *2020 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2020* (2020), pp. 108–113. DOI: 10.1109/ICARSC49921.2020.9096200.

[108] Alba Pujol-Miró, Josep R. Casas, and Javier Ruiz-Hidalgo. "Correspondence matching in unorganized 3D point clouds using Convolutional Neural Networks". In: *Image and Vision Computing* 83-84 (2019), pp. 51–60. DOI: 10.1016/j.imavis.2019.02.013.

[109] Harish Ravichandar et al. "Recent Advances in Robot Learning from Demonstration". In: *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020), pp. 297–330. ISSN: 25735144. DOI: 10.1146/annurev-control-100819-063206.

[110] Niyati Rawal and Ruth Maria Stock-Homburg. "Facial Emotion Expressions in Human–Robot Interaction: A Survey". In: *International Journal of Social Robotics* 2020.January (2022). DOI: 10.1007/s12369-022-00867-0.

[111] Xavier Robert-Lachaine et al. "Accuracy and repeatability of single-pose calibration of inertial measurement units for whole-body motion analysis". In: *Gait and Posture* 54 (2017), pp. 80–86. DOI: 10.1016/j.gaitpost.2017.02.029.

[112] Open Robotics. *ROS*. URL: https://www.ros.org/.

[113] I. Rodriguez et al. "Bringing the human arm reachable space to a virtual environment for its analysis". In: *Proceedings - IEEE International Conference on Multimedia and Expo* 1.August (2003), pp. 229–232. DOI: 10.1109/ICME.2003.1220896.

[114] Korbinian Schmid and Heiko Hirschmuller. “Stereo vision and IMU based real-time ego-motion and depth image computation on a handheld device”. In: *Proceedings - IEEE International Conference on Robotics and Automation* (2013), pp. 4671–4678. DOI: 10.1109/ICRA.2013.6631242.

[115] Cédric Schwartz et al. “Merging multi-camera data to reduce motion analysis instrumental errors using Kalman filters”. In: *Computer Methods in Biomechanics and Biomedical Engineering* 18.9 (2015), pp. 952–960. DOI: 10.1080/10255842.2013.864640.

[116] Gerald Schweighofer and Axel Pinz. “Robust pose estimation from a planar target”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.12 (2006), pp. 2024–2030. ISSN: 01628828. DOI: 10.1109/TPAMI.2006.252.

[117] Thomas Seel, Thomas Schauer, and Jorg Raisch. “Joint axis and position estimation from inertial measurement data by exploiting kinematic constraints”. In: *Proceedings of the IEEE International Conference on Control Applications* October (2012), pp. 45–49. DOI: 10.1109/CCA.2012.6402423.

[118] X. J. Shan and P. Tang. “A robust false matching points detection method for remote sensing image registration”. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* 40.7W3 (2015), pp. 699–702. DOI: 10.5194/isprsarchives-XL-7-W3-699-2015.

[119] R. Sibson. “SLINK: An optimally efficient algorithm for the single-link cluster method”. In: *The Computer Journal* 16.1 (1973), pp. 30–34. ISSN: 0010-4620. DOI: 10.1093/comjnl/16.1.30.

[120] Tasbolat Taunyazov, Bukeikhan Omarali, and Almas Shintemirov. “A novel low-cost 4-DOF wireless human arm motion tracker”. In: *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*. Vol. 2016-July. 2016, pp. 157–162. ISBN: 9781509032877. DOI: 10.1109/BIOROB.2016.7523615.

[121] David Tedaldi, Alberto Pretto, and Emanuele Menegatti. “A robust and easy to implement method for IMU calibration without external equipments”. In: *Proceedings - IEEE International Conference on Robotics and Automation*. 2014, pp. 3042–3049. DOI: 10.1109/ICRA.2014.6907297.

[122] Yi Lung Tsai et al. “EcoIMU: A dual triaxial-accelerometer inertial measurement unit for wearable applications”. In: *2010 International Conference on Body Sensor Networks, BSN 2010* (2010), pp. 207–212. DOI: 10.1109/BSN.2010.47.

[123] Panagiota Tsarouchi, Sotiris Makris, and George Chryssolouris. “Human–robot interaction review and challenges on task planning and programming”. In: *International Journal of Computer Integrated Manufacturing* 29.8 (2016), pp. 916–931. DOI: 10.1080/0951192X.2015.1130251.

[124] Yanghai Tsin and Takeo Kanade. “A correlation-based approach to robust point set registration”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3023 (2004), pp. 558–569. ISSN: 16113349. DOI: 10.1007/978-3-540-24672-5_44.

[125] Aleksandar Vakanski et al. “Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. Vol. 42. 4. 2012, pp. 1039–1052. DOI: 10.1109/TSMCB.2012.2185694.

[126] P.H. Veltink et al. “Detection of static and dynamic activities using uniaxial accelerometers”. In: *IEEE Transactions on Rehabilitation Engineering* 4.4 (1996), pp. 375–385. ISSN: 10636528. DOI: 10.1109/86.547939.

[127] Valeria Villani et al. “Survey on Human-Robot Interaction for Robot Programming in Industrial Applications”. In: *IFAC-PapersOnLine* 51.11 (2018), pp. 66–71. ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.08.236.

[128] W.H.K. de Vries et al. “Functionally interpretable local coordinate systems for the upper extremity using inertial and magnetic measurement systems”. In: *Journal of Biomechanics* 43.10 (2010), pp. 1983–1988. ISSN: 00219290. DOI: 10.1016/j.jbiomech.2010.03.007.

[129] W.H.K. de Vries et al. “Magnetic distortion in motion labs, implications for validating inertial magnetic sensors”. In: *Gait and Posture* 29.4 (2009), pp. 535–541. ISSN: 09666362. DOI: 10.1016/j.gaitpost.2008.12.004.

[130] Corrin P. Walmsley et al. “Measurement of Upper Limb Range of Motion Using Wearable Sensors: A Systematic Review”. In: *Sports Medicine - Open* 4.1 (2018), pp. 53–75. DOI: 10.1186/s40798-018-0167-7.

[131] Daniel Wedge, Du Huynh, and Peter Kovesi. “Using space-time interest points for video sequence synchronization”. In: *Proceedings of IAPR Conference on Machine Vision Applications, MVA 2007* (2007), pp. 190–194.

[132] Samuel Wilson et al. “Formulation of a new gradient descent MARG orientation algorithm: Case study on robot teleoperation”. In: *Mechanical Systems and Signal Processing* 130 (2019), pp. 183–200. DOI: 10.1016/j.ymssp.2019.04.064.

[133] Michael T. Wolf et al. “Gesture-based robot control with variable autonomy from the JPL BioSleeve”. In: *Proceedings - IEEE International Conference on Robotics and Automation* (2013), pp. 1160–1165. DOI: 10.1109/ICRA.2013.6630718.

[134] Liukai Xu et al. “Gesture recognition using dual-stream CNN based on fusion of sEMG energy kernel phase portrait and IMU amplitude image”. In: *Biomedical Signal Processing and Control* 73 (2022), p. 103364. DOI: 10.1016/j.bspc.2021.103364.

[135] Katsu Yamame. *Dance Notations and Robot Motion*. Ed. by Jean-Paul Laumond and Naoko Abe. Springer Cham, 2016, p. 430. DOI: 10.1007/978-3-319-25739-6.

[136] Qing Quan Yang, Ling Ling Sun, and Longzhao Yang. “A Fast Adaptive-Gain Complementary Filter Algorithm for Attitude Estimation of an Unmanned Aerial Vehicle”. In: *Journal of Navigation* 71.6 (2018), pp. 1478–1491. DOI: 10.1017/S0373463318000231.

[137] Xiaoping Yun and Eric R. Bachmann. “Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking”. In: *IEEE Transactions on Robotics* 22.6 (2006), pp. 1216–1227. DOI: 10.1109/TRO.2006.886270.

[138] Mahdi Zabat et al. “IMU-based sensor-to-segment multiple calibration for upper limb joint angle measurement—a proof of concept”. In: *Medical and Biological Engineering and Computing* 57.11 (2019), pp. 2449–2460. DOI: 10.1007/s11517-019-02033-7.

[139] Angeliki Zacharaki et al. “Safety bounds in human robot interaction: A survey”. In: *Safety Science* 127 (2020). DOI: 10.1016/j.ssci.2020.104667.

[140] Heng Zhang et al. “A Feasibility Study on an Intuitive Teleoperation System Combining IMU with sEMG Sensors”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10984 LNAI (2018), pp. 465–474. DOI: 10.1007/978-3-319-97586-3_42.

[141] Tianhao Zhang et al. “Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 5628–5635. DOI: 10.1109/ICRA.2018.8461249.

[142] Zhi Qiang Zhang and Jian Kang Wu. “A novel hierarchical information fusion method for three-dimensional upper limb motion estimation”. In: *IEEE Transactions on Instrumentation and Measurement* 60.11 (2011), pp. 3709–3719. DOI: 10.1109/TIM.2011.2135070.

[143] Huiyu Zhou and Huosheng Hu. “Reducing drifts in the inertial measurements of wrist and elbow positions”. In: *IEEE Transactions on Instrumentation and Measurement* 59.3 (2010), pp. 575–585. ISSN: 00189456. DOI: 10.1109/TIM.2009.2025065.

[144] Hao Zhu et al. “A review of point set registration: From pairwise registration to groupwise registration”. In: *Sensors (Switzerland)* 19.5 (2019). DOI: 10.3390/s19051191.

[145] Shaghayegh Zihajehzadeh and Edward J. Park. “A Novel Biomechanical Model-Aided IMU/UWB Fusion for Magnetometer-Free Lower Body Motion Capture”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.6 (2017), pp. 927–938. DOI: 10.1109/TSMC.2016.2521823.