Universidade de Aveiro
2023

Pedro
Amaral

Antecipação de Ações pelo Robô em Tarefas de Montagem Colaborativa

Robot Action Anticipation for Collaborative Assembly Tasks

DISSERTATION PROPOSAL

Keywords

Human-Robot Collaboration, Machine Learning, Action Anticipation, Predictive Model, Robot Controller, Collaborative Robot

Abstract

The increase in the diversity of products on sale due to the evolution of technology and standard of life results in a growing demand for flexible manufacturing that can meet the necessary production, especially in small companies. Although the usual solution for these needs is to use human operators, which provide the necessary flexibility and precision, this comes at a greater cost. In contrast, industrial robots offer a relatively smaller price and show more value in repetitive and heavy tasks. This is where Human-Robot Collaboration (HRC) comes into action since it complements the flexibility of a human worker with the strength and lower cost of the robotic worker in the same workspace. However, to achieve true collaboration it is not enough to react to the partner's movements and intentions, the robot must anticipate them. Inside HRC, Action Anticipation is a technique used to predict the actions of the human workers so that the robot can better plan its movements, increasing manufacturing efficiency and safety. This dissertation aims at the development of an anticipatory system that allows to enhance human-robot collaboration in an industrial setting. The collaborative scenario will be one in which the robot observes the actions of the human operator, makes predictions about the human's intention, and reacts accordingly by executing a physical action. This document reviews the research in this field, including the commonly used data sources and algorithms with a particular focus on machine learning methodologies. The nature of anticipation and the mechanisms that support it remains open questions in field of HRC. To this extent, this study may have an impact on how to model, implement and validate anticipatory processes in an assembly task.

Contents

List of Figures

Acronyms

AI	Artificial Intelligence
CNN	Convolutional Neural Network
EMG	Electromyography
HRC	Human-Robot Collaboration
HRI	Human-Robot Interaction
IMU	Inertial Measurement Unit
LSTM	Long Short-Term Memory
ML	Machine Learning
RNN	Recurrent Neural Network
RL	Reinforcement Learning
ROS	Robot Operating System

CHAPTER 1

Introduction

1.1 Background

The Third Industrial Revolution was characterized by a focus on automating repetitive and heavy tasks on the assembly lines. Still, this created a problem: whenever the manufacturers needed the robots to work in a different assembly process, they needed to be reprogrammed by an expert. The Fourth Industrial Revolution, also known as Industry 4.0, refers to the current trend of the manufacturing sector to become more intelligent and achieve greater automation. This trend takes advantage of the recent developments in artificial intelligence, the Internet of Things, and autonomous robots to pave the way for more efficient and flexible production processes. With Industry 4.0, robots are expected to be more adaptable and perform more actions without constant explicit programming.

The concept of Human-Robot Collaboration (HRC) emerges as part of Industry 4.0 and involves the research of mechanisms that allow humans and robots to work together to achieve a shared goal. Some of the most relevant topics in recent research include collision avoidance and human-aware planning of robot motions. However, to achieve true collaboration, it is not enough to react to the partner's movements and intentions, the robot must anticipate them.

Artificial Intelligence (AI) has significantly evolved in the last years. With the increase of computational power, Machine Learning (ML), a subset of AI, has become an increasingly promising method to deal with complex data like images and text, heavily contributing to areas such as visual perception and speech recognition. Machine Learning 's ability to learn from data with minimal human intervention and understand new data it has never seen before makes it a prime candidate to solve many problems in robotics and HRC in particular.

1.2 Problem Definition

The concept of anticipation has been studied in several research fields, such as biology, psychology, and Artificial Intelligence. In general terms, anticipation is viewed as the impact of predictions on the current behavior of a system, be it natural or artificial. A prediction model provides information about the possible future state of the environment and/or system. This perspective of looking to the future is related to the purpose of incorporating that information into a decision-making or planning process. Accordingly, the system becomes anticipatory when it incorporates such a model and, simultaneously, when it uses the model to change its current behavior.

Over the last few decades, experimental evidences of the existence of anticipatory biological processes at different levels of organization have been reported [1, 2]. The ability to modify behavior in anticipation of future events offers an adaptive advantage to living organisms with an impact on behavioral execution and learning. Anticipation is also considered one of the required abilities of cognitive robots operating in dynamically changing environments. The role of anticipation is to connect the robot's action in the present to its final goal, helping the design of robots with an increased level of autonomy and robustness.

The fundamental aspects of anticipation lie at the intersection of concepts such as time and information, involving abilities such as perception and prediction. The above definition of anticipation contains a temporal element that provides a key division between anticipatory and non-anticipatory robots. Anticipatory robots make decisions based on current states and predicted future states using predictive models of the environment. At the other extreme of the spectrum are the robots that live in the present based on the current state of the observed environment, which are usually called reactive robots (e.g., the Braintenberg's vehicles [3]). However, the behavior of a purely reactive robot is limited by its temporal horizon since they have no memory of the past to build a model of the world. Most of the current robots present a behavior influenced either by the current perception as well as by the memory of past perceptions but still lacking a perspective of the future.

Information provides another defining aspect of anticipation since the prediction of a future state depends on sensory data. The challenge arises from the moment that an anticipatory system operates based on a potential future state (even before it occurs) that can only be inferred from past and current information. The inherent uncertainty associated with prediction process can be reduced through the acquisition of information, namely by using different sensory modalities. In this context, sensory fusion is a process often adopted to merge data from multiple sensors such that to reduce the amount of uncertainty that may be involved to produce more reliable knowledge about the future.

The nature of anticipation and the mechanisms that support it are considered open questions in AI and robotics. Current research addresses fundamental questions such as: in which situations is anticipation useful? How can anticipatory processes be modeled and implemented in robotic systems? What are the impacts that may result from an anticipatory behavior? In the context of this dissertation proposal, we consider anticipation as a combination of prediction and decision-making, as illustrated by the blocks diagram in Fig. 1.1. The prediction model offers the possibility of incorporating action selection in their planning through a decision-making block, while the planning module relates to the robot's actions. These modules can be developed separately, or an end-to-end learning technique could be used where the model learns the different parts from the perception to the feedback control.

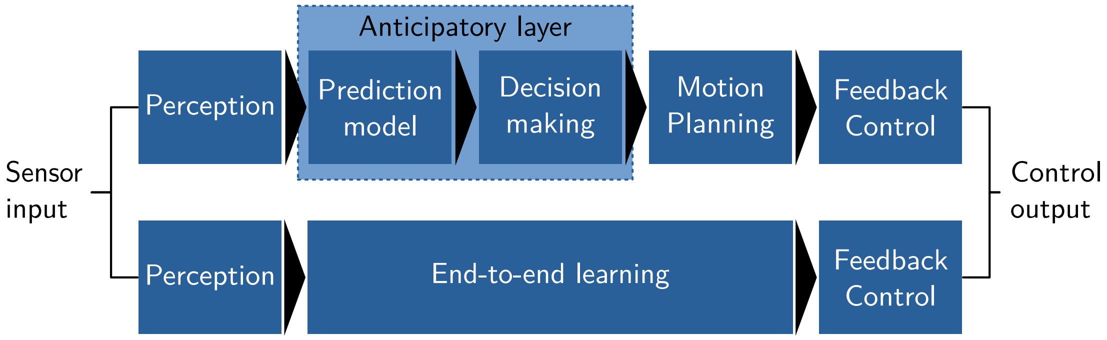


Figure 1.1: Functional blocks of an anticipatory robotic system considering two alternative approaches: modules developed separately vs end-to-end learning.

There are different situations in which an anticipatory response seems to be an essential ability for effective robot behavior. In an attempt to distinguish different types of anticipatory behaviors, three contexts in which a robot can operate are categorized below and the respective task requirements are presented as follows:

- **Time synchronization**. The interception of moving objects is central to several benchmark robotic tasks such as ball-catching and playing table tennis [4, 5]. These tasks are challenging due to the demanding spatial–temporal constraints, which require continuous coordination between visual, planning and control systems. On the one hand, frequent repredictions of the target location are required as new observations become available. On the other hand, this progressive refinement imposes an online re-planning of robot motion such that the goal is achieved in time.
- **Preventive safety**. Systems that manage risk require some form of anticipatory mechanism such that the robot can adapt its behavior when an undesired situation occurs. Autonomous driving is an example of how predicting future events and reacting properly are important abilities to mitigate risk. Modeling behavior and predicting the future intentions of pedestrians are core elements to ensure that the driver stops the car safely or avoids the pedestrian in time.
- **Coordinate joint activities in Human-Robot Interaction (HRI)**. Humans have the ability to coordinate their actions when carrying out joint tasks with other partners (Sebanz et al. [6] and Hoffman et al. [7]). In the same line of thought, anticipation can enhance the ability of a robot in its interaction with a human partner by predicting their actions (or intentions) before selecting its own action plan. In collaborative contexts such as those that occur during manufacturing or assembly tasks, the main challenge is combining anticipation and planning in a context of high uncertainty due to the variability of human behavior in complex industrial environments. Anticipation seems to have a significant potential for a more fluid and natural interaction with an impact on safety and cycle time.

This dissertation aims at the development of an anticipatory system that allows to enhance human-robot collaboration in industrial settings under the AUGMANITY mobilizing project[1].

[1] AUGMANITY website: `https://www.augmanity.pt`

The collaborative scenario will be one in which the robot observes the actions of the human operator, makes predictions about the human's intention and reacts accordingly by either waiting for more observations or executing a physical action. Fig. 1.2 illustrates the real workstation where the structure of a gas boiler for water heating is manually assembled. The collaborative robot's main function will be to assist with the assembly task by placing the parts in the jig while coordinating its actions with those of the human operator who is focused on the riveting process.

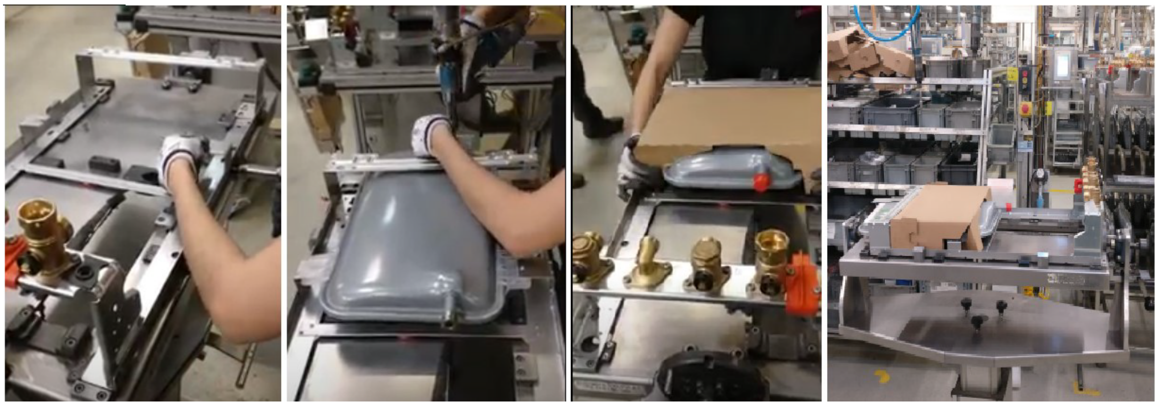


Figure 1.2: Different views of the current workstation used for the manual assemblage of the structure of a gas boiler for water heating.

1.3 Document Structure

The remainder of the document is organized into five chapters. Chapter 2 contains background material about anticipation, Machine Learning and collaborative robotics and a review of previous work on Action Anticipation in HRC including sensors and methods. Chapter 3 reviews tools that can be useful in the future implementation. Chapter 4 describes the progress made in the first semester. Chapter 5 portrays the planning of the second semester work and illustrates its calendarization. Chapter 6 concludes the document by restating the objectives of the work and the plan to achieve them.

CHAPTER 2

State of the Art

This chapter starts by covering background concepts about anticipation in biology, Machine Learning and collaborative robotics and then reviews previous work related to the dissertation theme, including sensors and methods.

2.1 Background Material

2.1.1 Anticipation in Biology

Anticipation is a research topic in many areas, such as biology, brain studies, psychology, social sciences, artificial intelligence, and engineering. One of the most cited definitions in the last decades and across the various fields is Rosen's [8]:

> An anticipatory system is a system containing a predictive model of itself and/or its environment, which allows it to change state at an instant in accord with the model's predictions pertaining to a later instant.

In the field of biology, Louie [9] claims that "Much, if not most, biological behavior is model-based ..." with the referred models being the "... internal predictive models of themselves and their environments ...". Poli [2] further claims that "... given that anticipatory behavior dramatically enhances the chances of survival, evolution itself may have found how to give anticipatory capacities to organisms, or to at least some of them.". For example, we can consider an animal predicting that it will be attacked by its predator and dodging said attack to survive.

In the case of humans, Louie [9] also stated, "We typically decide what to do now in terms of what we perceive will be the consequences of our action at some later time." alluding to our anticipatory behavior. Therefore, human actions can result from reactive behavior when they are based on the past, from anticipatory behavior when they are based on predictions of the future, or from a mix of both.

In particular, sports is a field where, according to Smith [10], "Proficiency in action anticipation is relevant in many performance contexts such as anticipating the direction of a shot (in soccer, hockey, tennis, volleyball, badminton, etc.), the deceptive movement of an opponent (in soccer, basketball, rugby, football, boxing, etc.), or the movement of a partner (in figure skating, dancing, etc.).".

2.1.2 Machine Learning (ML)

Machine Learning algorithms have been increasingly more common in the last years due to, for example, their ability to deal with multidimensional data. These algorithms can automatically learn from data and make predictions or decisions, which makes them a prime candidate to use in the context of human action anticipation in collaborative environments. The most common strategies in ML are Supervised Learning, Unsupervised Learning, and Reinforcement Learning.

Supervised Learning

In Supervised Learning, the models are trained using a dataset of labeled data. According to Sarker [11], these models must generalize the knowledge from the dataset's input-output pairs to correctly deal with a new input they have never seen before. The models from this group are further divided into classification, where the new input is assigned a discrete output class, and Regression, where it is returned a real number from the continuous output space. Currently, RNNs and CNNs are two of the most common classification approaches.

A Recurrent Neural Network (RNN) is a type of neural network where the output of each time step is fed back into the input at the next time step, allowing the network to remember and incorporate information from previous time steps into its processing of current and future data. This characteristic makes RNNs particularly well-suited to processing sequential data, such as text, speech, or time series data which require context or temporal dependencies. In particular, according to [12], Long Short-Term Memory (LSTM) is an RNN with a more complex architecture that gives it an improved ability to backpropagate the error, making it better to train a model that classifies sequences with several time steps.

A Convolutional Neural Network (CNN) is a type of neural network made up of several convolutional layers which apply a sliding filter over the input reducing its dimension and obtaining its features. Typically, these layers are followed by one or more fully connected layers that perform the prediction using the mentioned features. This architecture makes CNNs an excellent choice to deal with data in a matrix structure such as an image because this input is too massive for manual feature engineering.

In Supervised Learning, transfer learning is a technique that makes use of a trained external model. Depending on the goal of its use, these models can be entirely or partially used; optionally, they can also be trained partially or fully. A common use case for this technique is when a small dataset of images is used to obtain a classifier, and a standard model cannot generalize from that reduced amount of data. In this case, a model such as VGG-16 and ResNet-50 can be used partially to extract the features with one or more fully connected layers in the end, to perform the desired classification from those features.

Unsupervised Learning

In Unsupervised Learning, the datasets involved have no labels. According to Sarker [11], these algorithms aim to find patterns and structure in the data. This makes them valuable in tasks such as clustering based on common characteristics, density estimation, identifying anomalies and outliers, dimensionality reduction, feature learning and finding association rules.

Clustering is a technique used to create groups of points representing instances of the dataset to discover relevant trends or patterns. K-means clustering is one of the most common and simple clustering algorithms. It starts by creating k random centroids and assigns each instance of the data to the closest centroid by squared distance. This process can be repeated to achieve better results. This algorithm works well when the points from different sets are considerably separated from each other.

Dimensionality reduction is a technique that aims to reduce the number of features by selecting a subset of the original ones using algorithms such as Chi-squared test or extracting new features from the originals using algorithms such as Principal Component Analysis (PCA).

Reinforcement Learning (RL)

Reinforcement Learning is different from the previous approaches because it does not need a dataset. According to Alom et al. [13], the agent learns how to act in an unknown environment by interacting with it. After the agent's action, the environment returns an observation and a certain reward to the agent depending on the quality of the action. The agent uses the reward to update its internal model named policy improving its future performance and the cycle repeats, as shown in Fig. 2.1. This type of learning by trial and error has a certain resemblance to how humans gain knowledge, and it is useful when there is a need for an agent to make decisions in an environment that has considerable complexity, such as controlling a robot or playing a game.

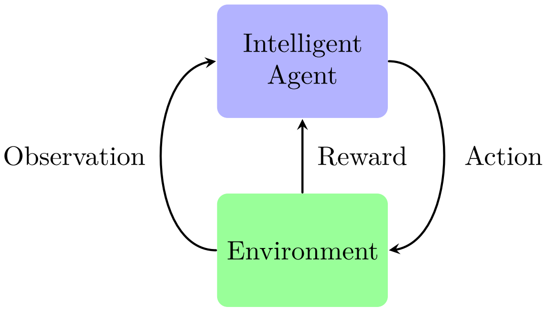


Figure 2.1: Interactions between the Agent and the Environment in RL [13]

In the workflow of Reinforcement Learning, the agent must associate an observation to an environment state. This is a simple process in a small discrete environment since there are fewer states. However, if the environment has many variables or it is not discrete then it becomes challenging to associate states with an observation. In these cases, it is necessary to use deep Reinforcement Learning, which is able to extract the relevant information from the observation and use it to associate the observation with a state.

Given that to train a Reinforcement Learning model, it is necessary for the agent to interact with the environment thousands of times, this process ends up needing a simulator. According to Li et al. [14], one of the major challenges in RL is transferring the knowledge learned in the simulator to a real-life environment. There may be a gap between the real and the simulated environment because the real world has more or less observable variables causing a drop in performance. Ahmed et al. [15] also claims that this may be due to an incorrect design of the reward function leading to over-fitting and sub-optimal policies.

2.1.3 Collaborative Robotics

Human-Robot Collaboration (HRC) consists of robots and humans working in the same workspace towards a common goal. Classical industrial robots are usually automated to perform repetitive tasks that require high physical strength. On the other hand, tasks that require cognitive knowledge, flexibility, and precision are better suited for humans, even if they are physically weaker. HRC aims to take advantage of both of their strengths and complement each others' weaknesses to increase manufacturing efficiency.

In a HRC scenario, robots need to be different from the traditional ones, given that they will work in the same workspace as humans. According to Castro et al. [16], "Collaborative robots need to be endowed with a set of abilities that enable them to act in close contact with humans, such as sensing, reasoning, and learning. In turn, the human must be placed at the centre of a careful design where safety aspects and intuitive physical interaction need to be addressed as well.". In [17], it is stated that nowadays, collaborative robots are developed to be compact, easy to install and program, flexible, mobile, consistent and precise. Additionally, they positively impact employees since they are responsible for monotonous and dangerous actions and reduce the production cost for the company.

Human-Robot Communication

Humans and robots can communicate through several methods, which can be direct such as using a console or a remote, or indirect, resulting from data captured from sensors. Based on [16, 18, 19], the main methods for indirect communication can be seen in the diagram in Fig. 2.2 and can be described as follows:

- **Gestures**: these are one of the main ways humans communicate, whether through simple movements or formal sign language. In the literature about HRC, gestures can also commonly be found since they have the advantage of resisting ambient noise. Usually, gestures are captured with vision-based methods with either an RGB or RGB-D camera, so there is no need for unnatural movements. With vision, it is possible to include markers, but these may lead to occlusions and hinder the worker's movements. Consequently, there is also work in the literature that uses markerless vision to allow more unrestricted movements. Another way to capture the movements of the human worker would be to use wearable inertial sensors, which contain accelerometers and gyroscopes, but, once again, wearables can hinder the worker's movements. Finally,

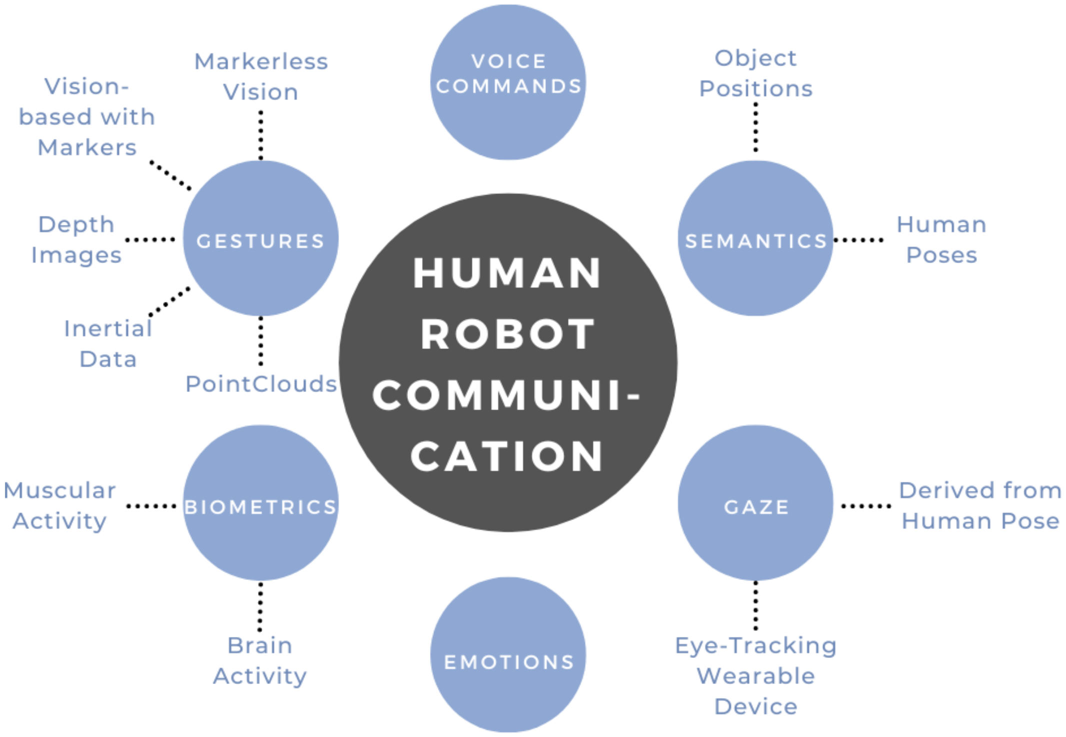


Figure 2.2: Data sources common in Human-Robot Collaboration

capturing point clouds using a LIDAR presents another possibility of capturing gestures without restricting the worker's motion.

- **Voice Commands**: Talking is the most intuitive way for humans to communicate with each other. The advances in voice recognition and natural language processing make this a possible communication solution with robots. However, despite being intuitive, simple, effective, and even robust against lighting variations, when it comes to an industrial setting that contains significant sound noise, it becomes less valuable than the alternatives.
- **Semantics**: semantic information about the objects can also help the global workflow. For example, suppose the robot is trained to recognize certain features in objects related to how it can pick them up. In this case, the robot can pick up a new object it has never seen before if it has a similar structure. Human actions can also be represented semantically by obtaining the poses of the human as a specific set of limbs, even if only partially. During action recognition, this can be used to know which objects the worker can interact with. Having semantic information about the pose of the human body also helps in the path-planning phase of the robot since it can use this information to avoid the worker and prevent collisions.
- **Gaze**: this can be used to determine where the user's attention resides, giving a considerable amount of information that can trigger some action. There are two options to obtain the user's gaze. Wearable sensors can provide better results but are expensive and intrusive. On the other hand, algorithms that detect head pose and assume the gaze from it can also be used, which is a cheaper and non-intrusive solution.

- **Emotions**: although this is a relatively new idea, some applications analyze the user's emotions from his facial expressions to have even more information in the algorithms.
- **Biometrics**: Electromyography (EMG) sensors can measure electrical signals generated by muscle contractions, while electroencephalography (EEG) signals are commonly used in brain-computer interfaces (BCIs).

Safety

Safety is one of the most critical topics in collaborative robotics and the first step toward establishing a collaborative environment. According to [17], collaborative robots are able to safely work with people because they have sensitive sensors that can detect the human interrupting them, causing them to stop their actions, while traditional robots would potentially injure the worker. However, given that there are tasks that require the robot to move very close to the worker, some norms were implemented: ISO 10218-1 and 10218-2. From these two standards, Castro et al. [16] and Villani et al. [20] describe the four criteria from which at least one must be met as:

1. **Safety-rated monitored stop**: when a human enters the cobot's workspace, it completely stops;
2. **Hand guiding**: when an operator manually moves the cobot, it is compliant;
3. **Speed and separation monitoring**: as the human moves closer to the cobot, it becomes gradually slower;
4. **Power and force limiting**: the cobot has its operation restricted in terms of force and torque.

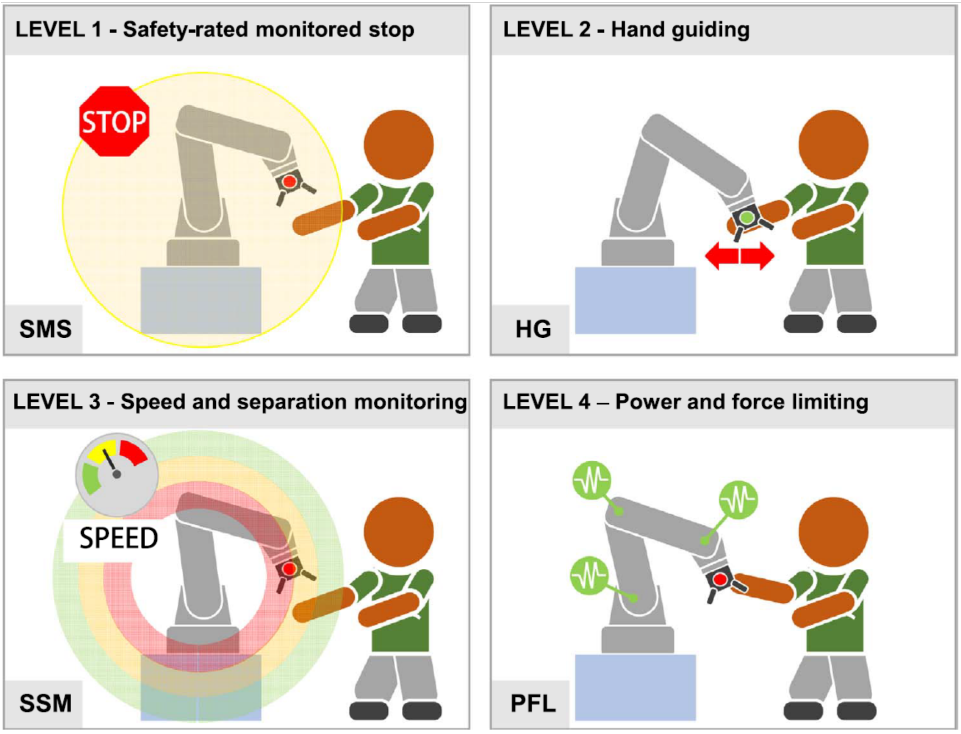

LEVEL 1 - Safety-rated monitored stop
STOP
SMS
LEVEL 2 - Hand guiding
HG
LEVEL 3 - Speed and separation monitoring
SPEED
SSM
LEVEL 4 – Power and force limiting
PFL

Figure 2.3: The four collaborative operative modes identified by robot safety standards ISO modes 10218-1/2 [20]

2.2 Data Sources and Sensors

The first step to anticipating the following action is to know which sensors should be used. Previously, several forms of communication between humans and robots were described. Still, these work in a more active way, and not all of them can be applied to action anticipation, where the user should not need to do anything for the robot to act. Essentially, there is a need to capture the human's body language or, in other words, his involuntary pose, gestures and gaze, which became some of the most commonly used data to perform action anticipation.

Regarding the sensors used to capture the raw data, most literature suggests using an RGB camera. However, the captured images may be used in the following different ways:

- directly used as input to models which can extract features from the images;
- used as input to frameworks that receive an image, process it, and return the key points, such as the skeleton joints of the person in the image; these key points can also then be used to assume the gaze of the human in the image such as in Canuto et al. [21] where the authors used OpenPose (explored in detail in Section 3.4) to obtain not only the skeleton joints but also the worker's gaze;
- used to process the optical flow[22, 23, 24, 25];
- if the human was wearing markers, the image can be used to obtain the positions of the markers obtaining gestures from the sequence of those positions [26];

Besides RBG cameras, some works, such as the one described in Moutinho et al. [27], indicate the use of an RGB-D camera to capture both the color and the depth images, which contain the gestures and pose of the worker. Other than cameras, in Tortora et al. [28] IMU and EMG data was used as input to capture the gestures and anticipate the worker's action. When it comes to obtaining the worker's gaze, it is possible to do so from the RGB images as mentioned above, but it is also possible to use wearable sensors to capture it, such as in Schydlo et al. [29].

2.3 Methods

After knowing which data is usually captured and provided to an algorithm, this section explores possible algorithmic solutions present in previous work starting by those that are only about predicting the action of the human worker and then those that go a step further and reference how to go from a prediction to the action that the robot must execute as a response.

Predictive Modeling Techniques

Predicting the next action of the worker can be represented as a classification problem since it is possible to use a sequence of images that must be classified as a particular future action class. Using Fig. 2.4 as an example, the high-five action should be predicted before the frames that contain it are captured. The previous work with this kind of algorithm mainly includes CNNs and RNNs, with the latter being the most common.

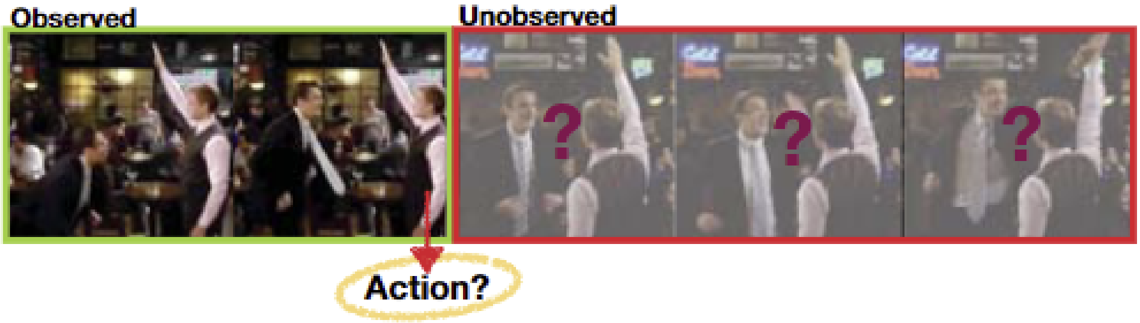


Figure 2.4: Action Anticipation using Supervised Learning diagram[22]

In Furnari et al. [25], the authors aimed to predict the subsequent actions that someone wearing a camera would perform and the objects he would interact with. They used three datasets containing RBG frames from which they derived the optical flow and the objects in the environment. This data is then passed on to a Rolling-Unrolling LSTM. The Rolling LSTM (R-LSTM) is a network that continuously encodes the received observations and keeps an updated summary of the past. When it is time to make predictions about future actions, the Unrolling LSTM (U-LSTM) is used with its hidden and cell states equal to the current ones of the R-LSTM.

In Schydlo et al. [29], the authors used an encoder-decoder recurrent neural network topology to predict human actions and intent where the encoder and the decoder are both LSTM cells. At each step, the decoder returns a discrete distribution of the possible actions making this algorithm able to consider multiple action sequences, which are then subject to a pruning method that reduces them to obtain the right action finally. In their work, these algorithms were tested in two different datasets, one containing RGB images with optical markers and gaze information from wearable sensors and another with RGB-D images.

In Moutinho et al. [27], the authors aimed to increase the natural collaboration between the robot and the human in an assembly station by interpreting implicit communication cues. The data related to the environment was captured using an RGB-D camera. This data was then passed on to a ResNet-34, a pre-trained neural network that extracted the features from the images. These features are used as the input to a LSTM to perform human action recognition.

In Gammulle et al. [22], the authors aimed to predict future frames while at the same time predicting the following action. In their implementation, they used public datasets with videos from which they obtained RGB images and optical flow streams. To consider both data sources, they also used two ResNet-50's, which are pre-trained networks, one to get the input features from the image and another from the optical flow, and 2 LSTMs to take into account both sequences of inputs. Then the two results are merged into a final classification. They also used two Generative Adversarial Networks (GAN) to generate the subsequent frames, but this is different from the focus of the analysis.

In Wang et al. [30], the authors used video datasets to train a model that would predict a future action from the observed frames. They used three pre-trained neural networks in their work: VGG-16, TS, and ConvNet, to extract features from the images. Then these features were aggregated using a Temporal Transformer module (TTM), and finally, a progressive prediction module (PPM) would anticipate the worker's future action. This article also addresses the issue of specifying what the algorithm should consider as an action. Although most of the literature often implies that the last frames captured by the camera are considered an action, given that those are the frames that contain the last action made by the user, the authors of this article go into greater detail. They tested and evaluated how many frames should be considered as the last action to obtain the best results using a metric from Geest et al. [31] named per-frame calibrated average precision (cAP) calculated with (2.1). In [30] it is defined with

$$cAP = \frac{\sum_k cPrec(k) * I(k)}{P}, \tag{2.1}$$

"... where calibrated precision $cPrec = \frac{TP}{TP+FP/w}$, $I(k)$ is an indicator function that is equal to 1 if the cut-off frame k is a true positive, P denotes the total number of true positives, and w is the ratio between negative and positive frames. The mean cAP over all classes is reported for final performance.".

In Rodriguez et al. [24], the authors aimed to predict the following action by first predicting the following motion images. They used datasets containing videos and then processed them to obtain motion images. These motion images become the input of a convolutional autoencoder network that generates the following motion images. These images are then passed to a CNN that processes them and makes action predictions for the future. The final action prediction is obtained from the results of the previous network and those of a second CNN, which analyzes the original RGB images.

In Wu et al. [23], the author's goal was to predict the following action someone wearing a camera would perform after some time. Initially, the optical flow was obtained from the captured images, and both were used as input to the model. The model is comprised of a Temporal Segment Networks (TSN), a CNN, and a LSTM to predict the future frame features and then use them to perform the required classification.

From Prediction to Planning

After predicting the next action of the worker, the robot must execute some action as a response to complete the anticipation process. This subsubsection contains articles that go beyond the predictive model and have relevant details for the integration of the model in a controller.

In Canuto et al. [21], the authors aimed to predict the following action using a LSTM, one of the most common RNNs. In their work, they used a dataset captured with an RGB camera. From these images, they obtained the objects in the environment, the human skeleton joints extracted over time using OpenPose, and the gaze derived from the joints. Then the three data sources were given to the LSTM as input to perform the desired classification. In this process, the authors use an adaptive threshold on the uncertainty of the recurrent neural network, which makes the model need a certain level of certainty to classify the action as a particular class. This creates a more robust solution since a standard supervised learning algorithm would predict the class with the highest probability even if the model has low certainty about every category.

In Maeda et al. [26], the authors aimed to reduce the delay in the robot's response by anticipating the human worker and providing a screw or a plate accordingly. They captured the environment using an RGB camera and tracked the hand using optical markers. Then they predicted the following human action using a look-up table containing different orders for assembly actions. With the nearest neighbor algorithm, the actions of the human would be matched with a particular order. The limitation of this method is that all possible sequences need to be on the table because if they are not there, then the robot will match with a different order which may be undesirable. If the robot eventually notices that it did the wrong action, it would then follow a hard-coded contingency trajectory to return to the pre-grasping position. When performing a handover action, the previously captured data is used to generate possible trajectories and this is given to the feedback controller as a reference.

In Zhang et al. [32], the authors aimed to predict the intention of the human worker to provide him with the required piece. To achieve this, they used an RGB camera to capture the data from the environment. Then the images are given to a convLSTM framework where the CNN part is in charge of extracting features from the input images, and these features are then passed on to the LSTM to predict the intention. This article also tackles the issue of having several possible assembly orders. It solves it by creating a phase at the beginning of the collaboration in which the robot learns the assembly actions and their order from a demonstration. After the prediction of the intention of the worker, the robot proceeds to fetch the required piece. It uses a CNN to recognize said piece and ROS Open Motion Planning Library (OMPL) to handle the trajectory planning jobs. In terms of safety, the authors defined speed limits for the robot and ensured that the robot would avoid the workspace of the human. Then when it needs to move closer to the user, its speed is reduced to guarantee the user's safety.

In Huang et al. [33], the authors' goal is to make the robot use the anticipated actions of the worker to decide its tasks. It monitors the worker's gaze using a wearable device and uses it to predict his intent using SVM. After predicting it, the robot uses an anticipatory motion planner named "MoveIt!" to plan its motion according to a certain confidence threshold. This means that while it is unsure of what the human wants, the robot starts to move towards the item it thinks he wants but only really moves completely when it surpasses the threshold.

CHAPTER 3

Tools Review

This chapter covers a review of the experimental setup and relevant software tools that may be helpful in developing the final solution.

3.1 Experimental Setup

The experimental part of this thesis will be developed using the setup available at the Laboratory for Automation and Robotics (LAR) located in the Department of Mechanical Engineering at the University of Aveiro. The available setup contains a collaborative robot surrounded by several sensors and a powerful computer that can be used to train machine learning models.

3.2 Robot Operating System (ROS)

ROS[34][1,2] is an open-source collection of tools and software libraries used to develop a robotics application. Its main features are:

- **message broker**: every process in the project is a node in the ROS network and communicates with the other nodes mainly through topics (asynchronous publish/subscribe streaming of data) or services (synchronous RPC-style communication);
- **code reuse**: executables and packages are written to be as independent as possible, making the developer able to reuse them in another project;
- **rich ecosystem**: there are several open-source packages available to the developer that can be easily integrated;
- **scalability**: given that the nodes are so loosely coupled, it allows for node distribution;
- **language independence**: nodes can be written in any language since communication is established through well-defined objects;
- **data visualization**: there are tools to visualize the data in real-time, such as Rviz;

[1] ROS 1 documentation: `https://wiki.ros.org`

[2] ROS 2 documentation: `https://docs.ros.org/en/humble`

- **simulator support**: ROS has support for simulators with Gazebo being the most common;
- **hardware abstraction**: contains driver packages to deal with some hardware devices;

ROS can be used by installing on Ubuntu or Mac OS X systems, and then nodes can be launched from existing repositories or can be programmed in Python, C++ or Lisp with more languages still under development.

3.3 Machine Learning Frameworks

This section contains a review of Tensorflow and Pytorch, which are two of the most popular machine learning frameworks that can be helpful in pre-processing data, building machine learning models and deploying said models.

Tensorflow

Tensorflow[3] is a platform that can be used for all steps of a machine learning project. Its main features are:

- **prepare data**: load data, data pre-processing and data augmentation;
- **build models**: design and train custom models with little code or use pre-trained ones (transfer learning);
- **deploy models**: helps using models in different platforms such as locally, in the cloud, in a browser, or in mobile;
- **implement MLOps**: run models in production, tracking their performance and identifying issues.

Tensorflow can be used through its APIs in several languages to adapt to every project but the Python API is recommended since it is the most stable.

Pytorch

Pytorch[4] is an open-source framework that can be used for all steps of a machine learning project. Its main features are:

- **distributed model training**: takes advantage of some Python and C++ features to optimize the performance in both research and production phases;
- **robust ecosystem**: there are tools and libraries that extend PyTorch to include features related to, for example, computer vision and natural language processing;
- **ready for production**: it has tools to deploy models with scalability in environments such as the cloud;
- **cloud support**: it has good support in the most common cloud platforms allowing an easier deployment in a production environment.

Pytorch can be used through its APIs in Python, Java, or C++.

[3] Tensorflow documentation: `https://www.tensorflow.org/api_docs`

[4] Pytorch documentation: `https://pytorch.org/docs`

3.4 Key Point Detection Frameworks

This section reviews OpenPose and OpenPifPaf which are two projects containing models to detect key points in images, such as the human skeleton joints.

OpenPose

OpenPose[35, 36, 37, 38][5] is an open-source project that aims to detect key points in the human body, face, hands, and feet from images. Its main features are:

- 2D real-time key point detection based on the body/foot, the hand, or the face of multiple people;
- 3D real-time key point detection based on images from multiple cameras of one person;
- estimation of camera calibration parameters;
- single-person tracking.

OpenPose can be used through the command-line or using an API for Python or C++.

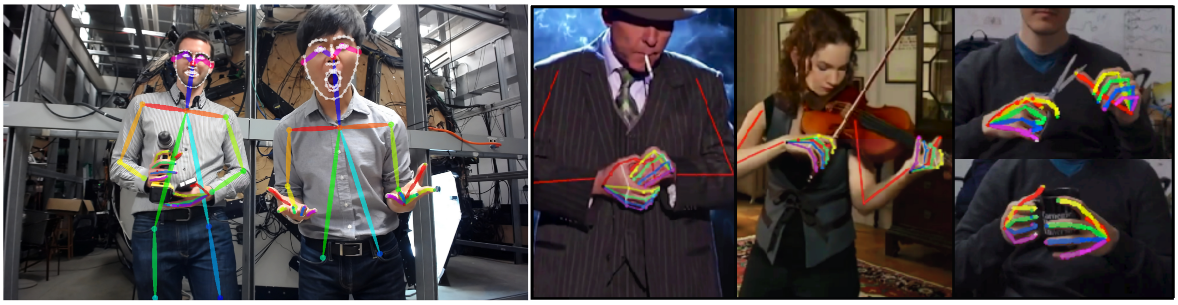


Figure 3.1: OpenPose Examples [35, 36]

OpenPifPaf

OpenPifPaf[39, 40][6] is an open-source project that aims to detect, associate and track semantic key points. Detecting human joints is an example of its usage but it is also able to generalize this detection to other classes such as cars and animals. It can be installed as a python package which can then be imported.

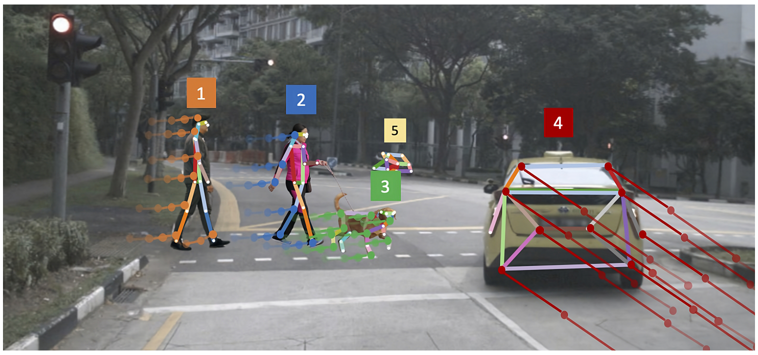


Figure 3.2: OpenPifPaf Example [39]

[5] OpenPose documentation: `https://cmu-perceptual-computing-lab.github.io/openpose`
[6] OpenPifPaf documentation: `https://openpifpaf.github.io`

CHAPTER 4

Work Progress

This chapter covers the work made in the first semester related to the testing of a smart inertial sensor, which is a common type sensor in HRC as seen in Subsection 2.1.3.

In the context of this dissertation, the idea was to take advantage of the features of this sensor as another possible source of data. This data can then be used to help detect gestures with the end goal of helping and automating the labeling of the model training data.

BHI260AP[1] is a smart sensor with integrated Inertial Measurement Unit (IMU) from Bosch Sensortec. According to [41], it includes several software functionalities, a 32-bit customer programmable microcontroller, and a 6-axis IMU. It is designed for always-on sensor applications such as fitness tracking, navigation, machine learning analytics and orientation estimation.

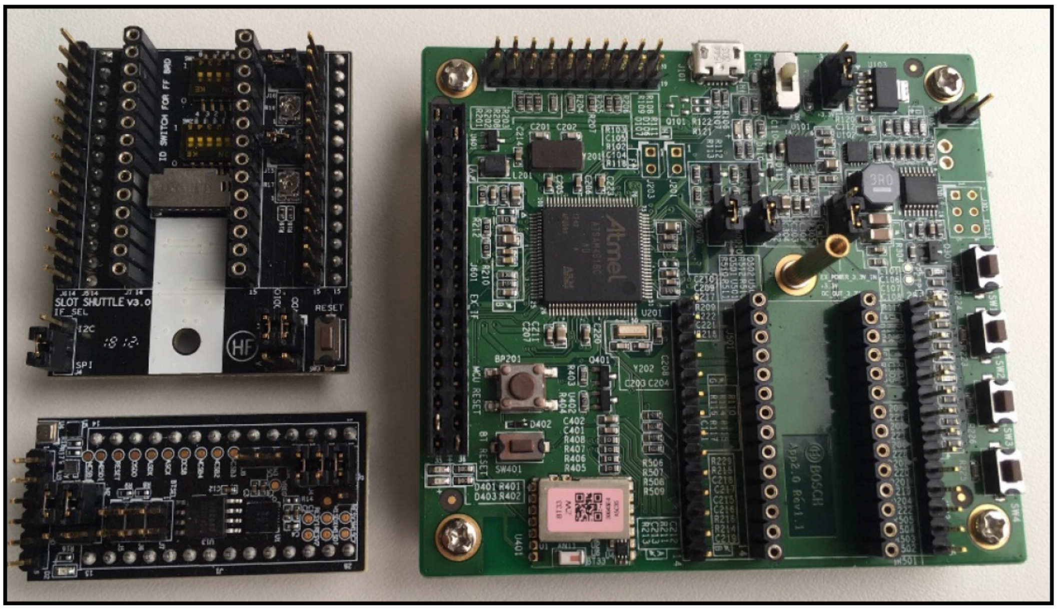
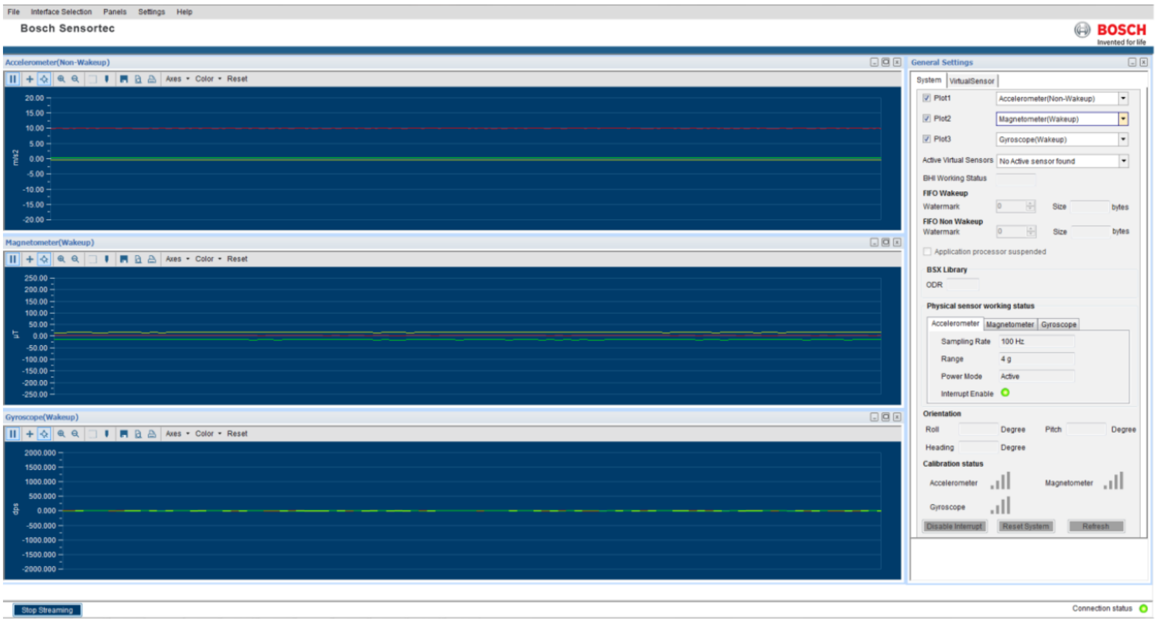


Figure 4.1: BHI260AP Sensor [42]

[1] Product Page: `https://www.bosch-sensortec.com/products/smart-sensors/bhi260ap`

During the testing of this sensor, three approaches were attempted:

- **Python API**: there was an attempt to use the Python library to communicate with the sensor but although communication was established, there was a lack of documentation and examples that allowed to create code that collected data;
- **C++ API**: there was also an attempt to use the C++ library to communicate with the sensor but the instructions available resulted in a compilation error;
- **Development Desktop 2.0**: this is a desktop application for Windows that allows communication with all Bosch Sensortec sensors and using it, it was possible to collect data to a file or display it in graphs such as in Fig. 4.2.



☑ Plot1 Accelerometer(Non-Wakeup)

☑ Plot2 Magnetometer(Wakeup)

☑ Plot3 Gyroscope(Wakeup)

Active Virtual Sensors No Active sensor found

BHI Working Status

Watermark 0 Size bytes

Watermark 0 Size bytes

□ Application processor suspended

ODR

Sampling Rate 100 Hz

Range 4 g

Power Mode Active

Interrupt Enable ●

Roll Degree Pitch Degree

Heading Degree

Accelerometer

Magnetometer

Gyroscope

Connection status ●

Figure 4.2: Development Desktop 2.0 Interface [43]

Therefore, collecting data was only possible with the Development Desktop 2.0. However, using the Python or the C++ APIs would be ideal since, as they work in Linux, it would be possible to integrate their usage with ROS.

CHAPTER 5

Planning

This chapter covers the planning of the work and the task scheduling to be accomplished during the second semester to achieve the proposed objectives. The main tasks to be carried out are described below (the Gantt diagram in Fig. 5.1 shows these activities displayed against time):

1. **Overview of the Experimental Setup**. Familiarization with the robot and tools that will be used throughout the dissertation.
2. **Development of Action Anticipation Models**. To formally define how to anticipate an action in the context of the collaborative task under study using RGB-D images as input. ML models, such as Recurrent Neural Networks (RNNs), are at the forefront of the algorithms to explore.
3. **Development of an Anticipatory Controller**. To develop robot controllers that consider the human partner's movements and intentions and use these inferences to make appropriate decisions during the execution of a sequential assembly task.
4. **Metrics and performance evaluation**. To provide performance metrics used to evaluate the action anticipation models and the add-value of the anticipatory controller (e.g., in terms of cycle time).
5. **Thesis Writing**. Writing the master dissertation and other detailed documentation.

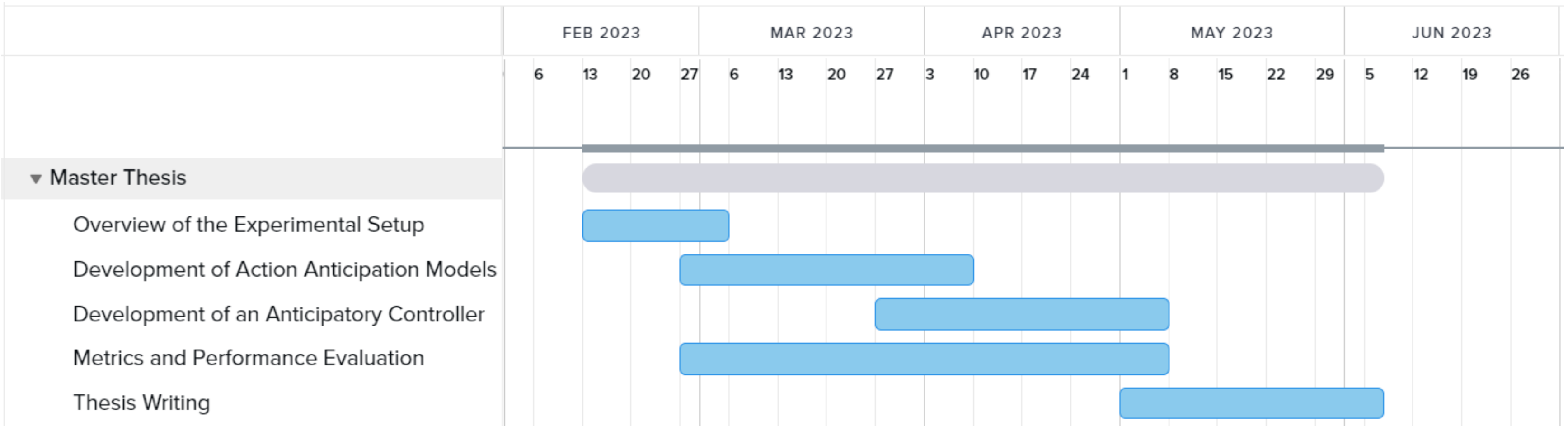


Figure 5.1: Gantt Diagram of the Second Semester Tasks

CHAPTER 6

Conclusion

This document presented the problem of anticipating human actions in collaborative environments with the goal of developing an anticipatory robot controller for an assembly task. Looking at previous work found in the literature, there is a clear predominance of perception using RGB cameras with different ways of preprocessing the captured images. When it comes to the methods, Machine Learning and, in particular, supervised learning techniques are predominant, given that most work nowadays takes advantage of the progress made in that field. With the continuous evolution of ML, it is expected that the algorithms related to the topic in this paper also evolve and, consequently, give rise to even better solutions.

To complete the study of previous work, some relevant tools were reviewed with a particular emphasis on two libraries that can detect key points in an image, such as skeleton joints which are very important to detect human poses. Regarding the practical side of the dissertation, an inertial sensor was tested in other to evaluate its inclusion as another source of data. Although it was possible to capture data, an additional effort will be required to demonstrate its usefulness for the proposed study. Furthermore, as a result of this work, the tasks for the second semester were delineated and scheduled.

In summary, the results of this study demonstrate that Action Anticipation is still a relatively new concept, but it has much potential to increase the efficiency and safety of collaborative tasks, revolutionizing the world of Human-Robot Collaboration.

References

[1] Carrie Deans. “Biological Prescience: The Role of Anticipation in Organismal Processes”. In: *Frontiers in Physiology* 12 (Dec. 2021). ISSN: 1664-042X. DOI: 10.3389/fphys.2021.672457. URL: https://www.frontiersin.org/articles/10.3389/fphys.2021.672457/full.

[2] Roberto Poli. “The many aspects of anticipation”. In: *Foresight* 12 (3 June 2010). Ed. by Riel Miller, pp. 7–17. ISSN: 1463-6689. DOI: 10.1108/14636681011049839. URL: https://www.emerald.com/insight/content/doi/10.1108/14636681011049839/full/html.

[3] Valentino Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. The MIT Press, Feb. 1986. ISBN: 9780262521123.

[4] Diogo Carneiro et al. “Robot Anticipation Learning System for Ball Catching”. In: *Robotics* 10 (4 Oct. 2021), p. 113. ISSN: 2218-6581. DOI: 10.3390/robotics10040113. URL: https://www.mdpi.com/2218-6581/10/4/113.

[5] Zhikun Wang et al. “Anticipatory action selection for human–robot table tennis”. In: *Artificial Intelligence* 247 (June 2017), pp. 399–414. ISSN: 00043702. DOI: 10.1016/j.artint.2014.11.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S0004370214001398.

[6] Natalie Sebanz et al. “Joint action: bodies and minds moving together”. In: *Trends in Cognitive Sciences* 10 (2 Feb. 2006), pp. 70–76. ISSN: 13646613. DOI: 10.1016/j.tics.2005.12.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S1364661305003566.

[7] Guy Hoffman et al. “Cost-Based Anticipatory Action Selection for Human–Robot Fluency”. In: *IEEE Transactions on Robotics* 23 (5 Oct. 2007), pp. 952–961. ISSN: 1552-3098. DOI: 10.1109/TRO.2007.907483. URL: https://ieeexplore.ieee.org/document/4339531/.

[8] Robert Rosen. *Anticipatory Systems: Philosophical, Mathematical and Methodological Foundations*. Elsevier, 1985, pp. 339–347. ISBN: 9780080311586. DOI: 10.1016/C2009-0-07769-1.

[9] A.H. Louie. “Robert Rosen’s anticipatory systems”. In: *Foresight* 12 (3 June 2010). Ed. by Riel Miller, pp. 18–29. ISSN: 1463-6689. DOI: 10.1108/14636681011049848. URL: https://www.emerald.com/insight/content/doi/10.1108/14636681011049848/full/html.

[10] Daniel M. Smith. “Neurophysiology of action anticipation in athletes: A systematic review”. In: *Neuroscience & Biobehavioral Reviews* 60 (Jan. 2016), pp. 115–120. ISSN: 01497634. DOI: 10.1016/j.neubiorev.2015.11.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S0149763415302360.

[11] Iqbal H. Sarker. “Machine Learning: Algorithms, Real-World Applications and Research Directions”. In: *SN Computer Science* 2 (3 May 2021), p. 160. ISSN: 2662-995X. DOI: 10.1007/s42979-021-00592-x. URL: https://link.springer.com/10.1007/s42979-021-00592-x.

[12] Tiago Miguel. *How the LSTM improves the RNN*. Last accessed 20 January 2023. 2021. URL: https://towardsdatascience.com/how-the-lstm-improves-the-rnn-1ef156b75121.

[13] Md Zahangir Alom et al. “A State-of-the-Art Survey on Deep Learning Theory and Architectures”. In: *Electronics* 8 (3 Mar. 2019), p. 292. ISSN: 2079-9292. DOI: 10.3390/electronics8030292. URL: https://www.mdpi.com/2079-9292/8/3/292.

[14] Chengxi Li et al. “Deep reinforcement learning in smart manufacturing: A review and prospects”. In: *CIRP Journal of Manufacturing Science and Technology* 40 (Feb. 2023), pp. 75–101. ISSN: 17555817. DOI: 10.1016/j.cirpj.2022.11.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S1755581722001717.

[15] Ossama Ahmed et al. "CausalWorld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning". unpublished. Oct. 2020. URL: `http://arxiv.org/abs/2010.04296`.

[16] Afonso Castro et al. "Trends of Human-Robot Collaboration in Industry Contexts: Handover, Learning, and Metrics". In: *Sensors* 21 (12 June 2021), p. 4113. ISSN: 1424-8220. DOI: `10.3390/s21124113`. URL: `https://www.mdpi.com/1424-8220/21/12/4113`.

[17] WiredWorkers. *Cobots*. Last accessed 3 January 2023. URL: `https://wiredworkers.io/cobot/`.

[18] Debasmita Mukherjee et al. "A Survey of Robot Learning Strategies for Human-Robot Collaboration in Industrial Settings". In: *Robotics and Computer-Integrated Manufacturing* 73 (Feb. 2022), p. 102231. ISSN: 07365845. DOI: `10.1016/j.rcim.2021.102231`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0736584521001137`.

[19] Francesco Semeraro et al. "Human–robot collaboration and machine learning: A systematic review of recent research". In: *Robotics and Computer-Integrated Manufacturing* 79 (Feb. 2023), p. 102432. ISSN: 07365845. DOI: `10.1016/j.rcim.2022.102432`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0736584522001156`.

[20] Valeria Villani et al. "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications". In: *Mechatronics* 55 (Nov. 2018), pp. 248–266. ISSN: 09574158. DOI: `10.1016/j.mechatronics.2018.02.009`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0957415818300321`.

[21] Clebeson Canuto et al. "Action anticipation for collaborative environments: The impact of contextual information and uncertainty-based prediction". In: *Neurocomputing* 444 (July 2021), pp. 301–318. ISSN: 09252312. DOI: `10.1016/j.neucom.2020.07.135`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0925231220317719`.

[22] Harshala Gammulle et al. "Predicting the Future: A Jointly Learnt Model for Action Anticipation". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019, pp. 5561–5570. ISBN: 978-1-7281-4803-8. DOI: `10.1109/ICCV.2019.00566`. URL: `https://ieeexplore.ieee.org/document/9009844/`.

[23] Yu Wu et al. "Learning to Anticipate Egocentric Actions by Imagination". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 1143–1152. ISSN: 1057-7149. DOI: `10.1109/TIP.2020.3040521`. URL: `https://ieeexplore.ieee.org/document/9280353/`.

[24] Cristian Rodriguez et al. "Action Anticipation by Predicting Future Dynamic Images". In: *Computer Vision – ECCV 2018 Workshops*. Springer, 2019, pp. 89–105. ISBN: 9783030110147. DOI: `10.1007/978-3-030-11015-4_10`. URL: `http://link.springer.com/10.1007/978-3-030-11015-4_10`.

[25] Antonino Furnari et al. "Rolling-Unrolling LSTMs for Action Anticipation from First-Person Video". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (11 Nov. 2021), pp. 4021–4036. ISSN: 0162-8828. DOI: `10.1109/TPAMI.2020.2992889`. URL: `https://ieeexplore.ieee.org/document/9088213/`.

[26] Guilherme J. Maeda et al. "Anticipative interaction primitives for human-robot collaboration". In: *AAAI Fall Symposium - Technical Report*. 2016. ISBN: 9781577357759.

[27] Duarte Moutinho et al. "Deep learning-based human action recognition to leverage context awareness in collaborative assembly". In: *Robotics and Computer-Integrated Manufacturing* 80 (Apr. 2023), p. 102449. ISSN: 07365845. DOI: `10.1016/j.rcim.2022.102449`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0736584522001314`.

[28] Stefano Tortora et al. "Fast human motion prediction for human-robot collaboration with wearable interface". In: *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, Nov. 2019, pp. 457–462. ISBN: 978-1-7281-3458-1. DOI: `10.1109/CIS-RAM47153.2019.9095779`. URL: `https://ieeexplore.ieee.org/document/9095779/`.

[29] Paul Schydlo et al. "Anticipation in Human-Robot Cooperation: A Recurrent Neural Network Approach for Multiple Action Sequences Prediction". In: IEEE, May 2018, pp. 1–6. ISBN: 978-1-5386-3081-5. DOI: `10.1109/ICRA.2018.8460924`. URL: `https://ieeexplore.ieee.org/document/8460924/`.

[30] Wen Wang et al. “TTPP: Temporal Transformer with Progressive Prediction for efficient action anticipation”. In: *Neurocomputing* 438 (May 2021), pp. 270–279. ISSN: 09252312. DOI: 10.1016/j.neucom.2021.01.087. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925231221001697.

[31] Roeland De Geest et al. “Online Action Detection”. unpublished. Apr. 2016. URL: http://arxiv.org/abs/1604.06506.

[32] Zhujun Zhang et al. “Prediction-Based Human-Robot Collaboration in Assembly Tasks Using a Learning from Demonstration Model”. In: *Sensors* 22 (11 June 2022), p. 4279. ISSN: 1424-8220. DOI: 10.3390/s22114279. URL: https://www.mdpi.com/1424-8220/22/11/4279.

[33] Chien-Ming Huang et al. “Anticipatory robot control for efficient human-robot collaboration”. In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, Mar. 2016, pp. 83–90. ISBN: 978-1-4673-8370-7. DOI: 10.1109/HRI.2016.7451737. URL: http://ieeexplore.ieee.org/document/7451737/.

[34] Steven Macenski et al. “Robot Operating System 2: Design, architecture, and uses in the wild”. In: *Science Robotics* 7.66 (2022), eabm6074. DOI: 10.1126/scirobotics.abm6074. URL: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.

[35] Zhe Cao et al. “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (1 Jan. 2021), pp. 172–186. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2019.2929257. URL: https://ieeexplore.ieee.org/document/8765346/.

[36] Tomas Simon et al. “Hand Keypoint Detection in Single Images using Multiview Bootstrapping”. unpublished. Apr. 2017. URL: http://arxiv.org/abs/1704.07809.

[37] Zhe Cao et al. “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”. unpublished. Dec. 2018. URL: http://arxiv.org/abs/1812.08008.

[38] Shih-En Wei et al. “Convolutional Pose Machines”. unpublished. Jan. 2016. URL: http://arxiv.org/abs/1602.00134.

[39] Sven Kreiss et al. “OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association”. unpublished. Mar. 2021. URL: http://arxiv.org/abs/2103.02440.

[40] Sven Kreiss et al. “PifPaf: Composite Fields for Human Pose Estimation”. unpublished. Mar. 2019. URL: http://arxiv.org/abs/1903.06593.

[41] *BHI260AP Flyer*. Bosch Sensortec.

[42] *BHI260AB/BHA260AB Evaluation Setup Guide*. Bosch Sensortec.

[43] *BHYxxx Desktop Development 2.0 User Manual*. Bosch Sensortec.