UNIVERSIDAD CARLOS III DE MADRID

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

TESIS DE MÁSTER

HUMAN-ROBOT REMOTE COLLABORATION AND LEARNING OF SKILLS

Autor: Daniel Hernández García
Director: Dr. Carlos Balaguer Bernaldo de Quirós

MÁSTER OFICIAL EN
ROBÓTICA Y AUTOMATIZACIÓN

LEGANÉS, MADRID
OCTUBRE 2010

UNIVERSIDAD CARLOS III DE MADRID
MÁSTER OFICIAL EN ROBÓTICA Y AUTOMATIZACIÓN

El tribunal aprueba la tesis de Máster titulada "**Human-Robot Remote Collaboration and Learning of Skills** " realizado por **Daniel Hernández García**.

Fecha: Octubre 2010

Tribunal:

Dr. Nestor Nava Rodríguez

Dr. Juan González Gómez

Dr. Manuel Desco Menendez

Contents

List of Tables

List of Figures

Acknowledgements

First I want to thank Dr. Carlos Balaguer, not only for his work as director of this thesis, but for granting me the opportunity to be a part of his research group and to work in such fascinating topics as the one's deal on this work.

I also want to thank my partners working with the HOAP-3 robot, Paolo and Miguel. And to the rest of the humanoids group, Santi, Concha, Juan and Alberto. I also want to extent my thanks to the RoboticsLab were I have work in the completion of this work.

Also a very special thanks to Prof. Aude Billard for receiving me in LASA laboratory at the EPFL where I studied the learning techniques presented in this work. I also must thank Elena, Mohammed and Eric Sauser from who's work mine leans on. I also extent my thanks to rest of the LASA laboratory who make me fell welcome at their lab.

I must always thank Prof. Gerardo Fernández who introduced me to the world of robotics. And to my friends Sylvia, Joshue, Idania, Luis, Leonel, Jorge and Sasha.

Finally, I want to thank all my family who always supports me and care about me.

Resumen

Este trabajo se centra en temas referentes a la interacción y la colaboración entre humanos y robots humanoides para realizar tareas en un entorno colaborativo de trabajo. En el marco de este trabajo se desarrollo una arquitectura de colaboración robot-humano-robot para que un operador humano y un robot local puedan colaborar con un robot situado en una ubicación remota. Se presentan tres formas de interacción, a) humano-robot colaborativa a distancia, en la que un operador y un robot en un entorno de trabajo remoto interactuan a través de una HRI en trabajos colaborativos. b) humano-robot cercana, donde un humano, maestro, enseña a un robot varias demonstraciones de una tarea. b) robot-robot, para transferir los modelos de las habilidades aprendidas entre un robot local, enseñado por un operador, y un robot remoto ocupado de realizar tareas en un entorno de colaboración a distancia. Para probar el sistema se presenta un escenario de trabajo collaborativo humano-robot en un entorno espacial. El operador se conecta al robot remoto a través de la HRI. El robot se mueve y funciona de forma autónoma de acuerdo a la solicitud del operador. Cuando surge una situacion nueva o desconocida el robot remoto pide al operador el modelo de la habilidad. El operador enseña al robot local, y se produce una interacción robot-robot para la transferencia de los modelos de conocimiento de la tarea. El robot remoto reproduce las habilidades aprendidas para completar la tarea.

Abstract

This work deals with issues of interaction and collaboration between humans and humanoid robots performing tasks in a collaborative working environment. A robot-human-robot collaboration architecture was developed for a human operator and a local robot to collaborate with a robot located at a remote location. In this work three forms of interaction where presented, a human-robot remote collaboration interaction where a human operator and a robot at a remote working environment interact through a HRI in achieving collaboratively a global goal. A close human-robot interaction where a human teacher present a robot several demonstration of a task. And a robot-robot interaction for transferring the learned skills models between a local robot, that is taught by a human operator, and a remote robot performing task autonomously in a remote collaboration environment. The system was tested in a scenario presenting a robot working in collaboration with a human in a space environment. The human operator connects to remote robot through the HRI. The robot moves and performs autonomously according to the request of the operator. When new or unknown request arise the remote robot ask the operator for the teaching of the skill. The human teaches the local robot, and a robot-robot interaction ensue to transfer the learned models of the task. The robot reproduce the learned skills to complete the task, with operator supervision.

Chapter 1

Introduction

While the modern conception of robotics comes from the science fiction books and movies, the obligatory mentions to Karel Capek "R.U.R. (Rossum's Universal Robots)"(Capek, 2004) and Isaac Asimov "I, Robot"(Asimov, 2008) needs to come here, technological advances throughout the 20th century has allowed for the development of robotic solutions, in industrial and manufacturing applications, as a reality.

Since 1980's robots has been progressively introduce in the industry for the automation of manufacturing process performing precise and repetitive task, handling delicate or dangerous substances, lifting heavy objects, etc. Robots has been use for tasks that can be accurately defined and must be performed the same way every time, in well know and highly structured environments, standing in the place of human workers for jobs that were considered as dull, dangerous or dirty (know as the 3 D's of robotics). Robotic systems are broadly employ in several areas such as the automotive, metal products, the chemical, the electronics and the food industries.

Recent developments and technological advances has allowed for robotics to expand its applications from a largely dominant industrial focus into the challenges of the human world. The new generation of robots should be able to interact with humans in homes, workplaces, and communities, providing support in services, entertainment, education, healthcare, manufacturing, and assistance (Siciliano & Khatib, 2008). The future robotic systems must depart from the simple and repetitive 3 D's tasks and evolve to more complex and dynamic 3 A's tasks (The 3 A's stand for Aware, Autonomous and Assistant) (Pierro, 2009).

Humanoid Robots are suitable for this tasks since they have a human shape design that will allow them to collaborate and work with humans in the home, office or workplace without the need to adapt the environment and it will allow a higher acceptance and a more intuitive and natural interaction between human operators and the robotic agents. Recent years have seen and increase in research of humanoid robots like the WABIAN-2 from the University of Waseda (Ogura et al., 2006), ASIMO of Honda (Sakagami et al., 2002), the HRP-2 from the National Institute of Advanced Industrial Science and Technology of Japan (Kaneko et al., 2004) or the RH-1 and RH-2 designed at the Universdad Carlos III de Madrid (Arbulú, Kaynov, Cabas, & Balaguer, 2009).

1.1 Robots in the Human Working Environment

Research on the subject of collaborative robots and the collaborative working environments has not receive extensive attention from the robotics community. The main fundamental target of research is toward specific applications in which the collaborative working is only a subsequent problematic. However there are various projects related to collaborative environments with robots financed under different European frameworks programmes.

- ACROBOTER : Autonomous collaborative robots to swing and work in everyday environment. The project aims to develop a radically new robot locomotion technology that can effectively be used in home and/or in office environments for manipulating small objects autonomously or in close cooperation with humans (*ACROBOTER*, 2010).

- PHRIENDS : Physical Human-Robot Interaction: depENDability and Safety. The project aims at developing robots that can co-exist and co-operate with people, enabling a physical human-robot interaction which is dependable and safe (*PHRIENDS*, 2010).

- SMErobot : The European Robot Initiative for Strengthening the Competitiveness of SMEs in Manufacturing. The project aims at creating robots capable of understanding human-like instructions (by voice, gesture, graphics). And a safe and productive human-aware space-sharing robot (*SMErobot*, 2010).

- Robot@CWE : Advanced robotic systems in future collaborative working environments. The Projects aims to research and demonstrate integrative concepts of advanced robotic systems, to be seen as collaborative agents, in various environments working together with humans. ROBOT@CWE will design suitable architectures and technologies to achieve this goal (*Robot@CWE*, 2010).

This work was carried out under the Robot@CWE framework. In Robot@CWE the future robotic system is envisaged as potential working agents collaborating with humans in different collaborative environment clusters. Human centred robotics poses several challenges, such as: acceptability in the society, autonomy, interactivity, flexibility, and versatility (Arbulu et al., 2007).

Robot@CWE focus on humanoid robots as robotic full agents capable of interacting with themselves, humans and their environments in various autonomy and operational modes. It extend from the concept of human centred design towards a CWE-centred robotic design. This requires a system that is capable of handling shared spaces, of using functional representations through virtualized and mixed environments, of tele-working in multi-operator control modes, of collecting information using human communication interfaces, etc. (Hernández et al., 2009). When several human operators and robotic agents are added in an open space collaborative working environment, eventually at a remote location communicating through possible virtualization of functional aspects, there are huge potentialities and challenges in the working architecture design (Stasse et al., 2008). Figure 1.1 illustrates the ROBOT@CWE framework.

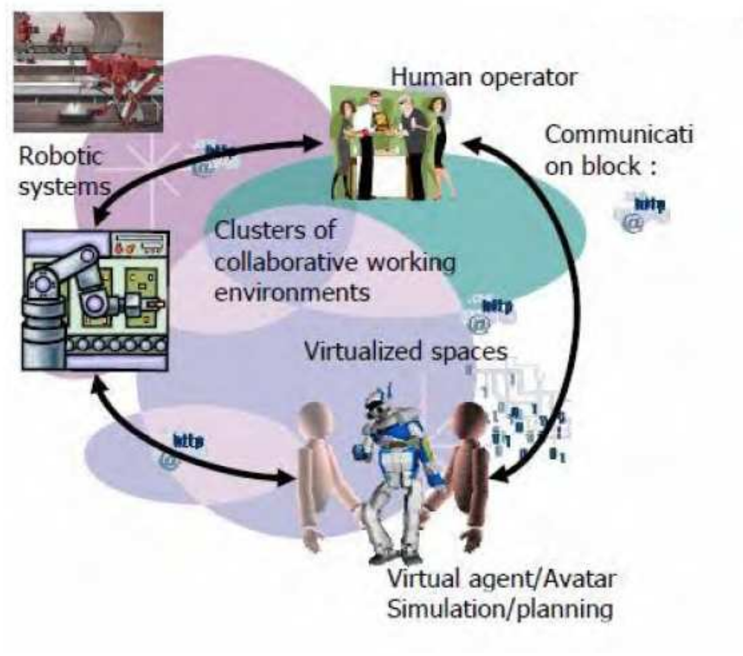


Figure 1.1: *A general framework architecture to realize task in a collaborative environment*

In the scope of Robot@CWE a functional collaborative working architecture was proposed. A Robot system must perform in an environment, where they communicate, interact, work, collaborate, and share resources with human partners. The technical requirements in terms of robots interfacing with various collaborative environments, working in collaboration with humans and sharing a common environmental working space, assumes robots to be flexible enough to adapt to different working styles, taxonomies and situations. Figure 1.2 describes an overall functional architecture to achieve collaborative work with a humanoid robot.

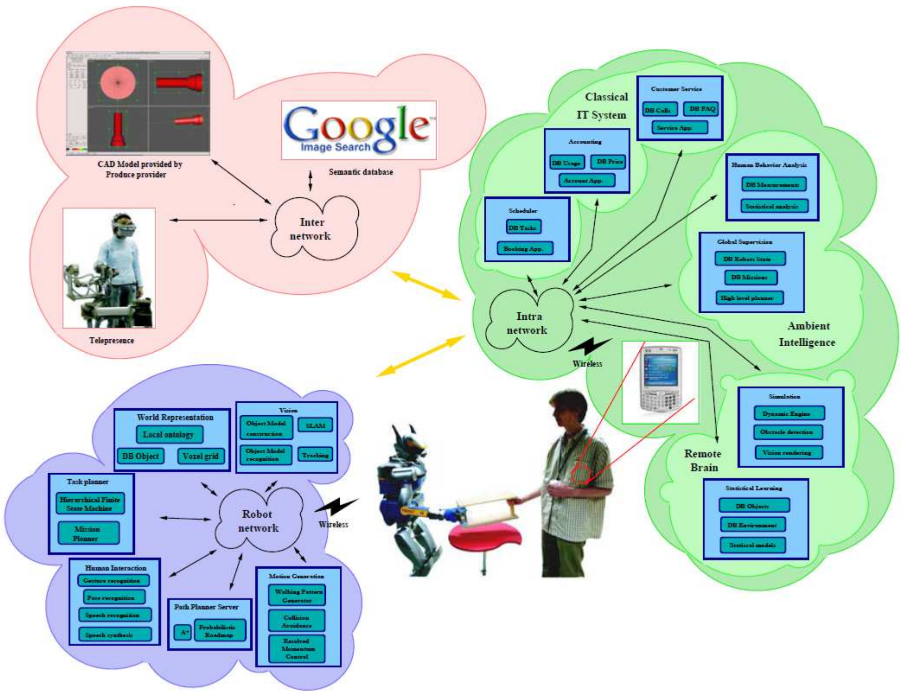


Figure 1.2: *Global Overview of the Architecture*

The aim was to design flexible architectures capable of realizing the framework for different working paradigms and applications. From this it was proposed a multi-layer architecture allowing humanoid robots to become CWE-robots with the following functionalities:

- Realize autonomously high-level behaviours such as:
  - Search for an object in an unknown environment.
  - Plan a trajectory from one point to another autonomously.
  - Recognize and identified objects.
  - Realize whole-body motion based on visual information.

- Act as a proxy for a remote operator to perform a collaborative task.
- Interact with an advanced Collaborative Working Environment (BSCW) which can specify high-level task to be realized.

1.2 Human-Robot Interactions

Humanoids robots are one of the main topics in service robots investigation. As humanoid robots are design to resemble a human shape and to poses human capabilities, they are capable of performing tasks in a world that is made for humans and to safely share the same space with people. Humanoid robots have many features that make them a very suitable partner in collaborative working environments. There are many experiments proving that humanoid robots can manipulate human tools (Ambrose et al., 2000) or even drive human vehicles (Yokoi et al., 2006), also, humanoid robots can enter environments in which simple mobile robots can hardly move and where humans are used to traverse (Yokoi et al., 2004).

Humanoid Robots are flexible and versatile machines able to help humans and work with them as an active agent. To achieve this goal, humanoid robots would have to interact not only with humans but with the environment as well. In order to do so, it is necessary to design new control architectures that allow them to perform all kind of tasks helping humans and sharing the same working environment.

The field of human-robot interaction is one of great interest for the robotic community. Many researchers are studying and developing several ways to permit robots to easily and explicitly communicate with a human by gesture or speech (Gorostiza et al., 2006). But this type of interaction - which is actually suitable for tasks in social robotics - may become inappropriate for human-robot cooperative work in terms of the user's cognitive load because it forces a user to become familiar with explicit communication protocols (Stasse et al., 2008).

We are still far from having a fully autonomous collaborative robot. This work concentrates on the robot as an intelligent tool but commanded and supervise by the human operator. A Telepresence system would require that a human operator control the actions of a remotely operated robot. While a completely autonomous robot will simply interact with the human, but never collaborate. Collaborative robots are different from industrial robots and computers or other technology typically found in the work environment, because they are mobile, semi-autonomous and interactive. In figure 1.3, (Pierro, 2009), one can graphically understand the trade-off between total control on the robot and its complete autonomy.

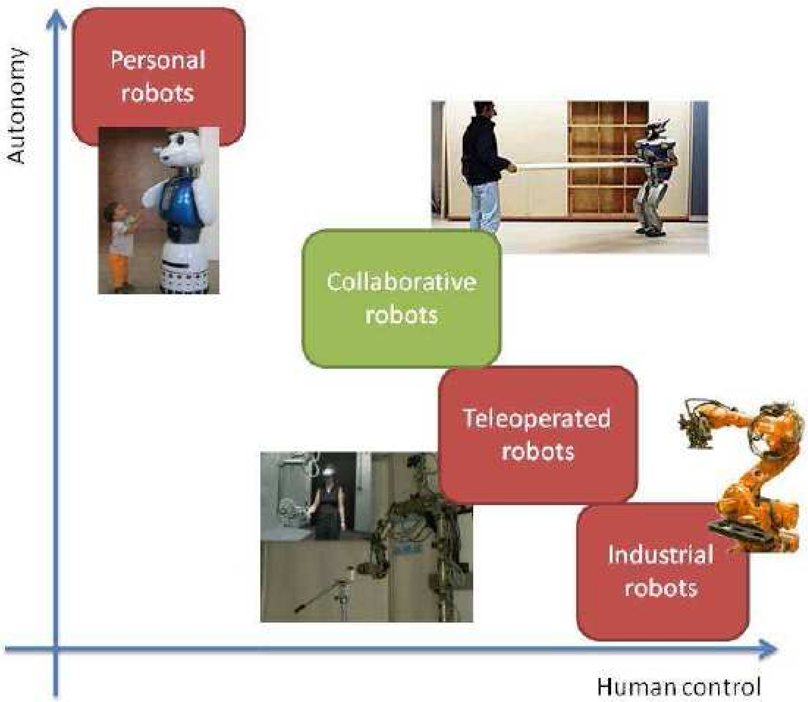


Figure 1.3: *Human Control vs. Robot Autonomy*

Humanoids robots working on a collaborative environment would present

	Direct Control	Visual and Vocal Interaction	Table PC	PDA System	Remote Control
Communication Security	high	low	medium	medium	low
Precision	high	low	high	medium	medium
Adaptability to different tasks	low	low	medium	medium	high
Composition of Robot Teams	homogeneous	heterogeneous	heterogeneous	heterogeneous	heterogeneous
Interaction Roles	operator	team-mate	team-mate	team-mate	operator
Human-Robot Physical Proximity	close	close	relatively-close	relatively-close	far
Interaction Roles	operator	team-mate	team-mate	team-mate	operator
Decision Support for Operators	high availability of sensors	no availability of sensors	medium availability of sensors	low availability of sensors	high availability of sensors
Time/Space Taxonomy	synchronous-collocated	synchronous-collocated	synchronous-non-collocated	synchronous-non-collocated	asynchronous-non-collocated
Required Autonomy Level of the Robot	no autonomy	high autonomy	semi-autonomy	high autonomy	reduced autonomy
Ease of Use	complex	easy	medium	easy	medium

Table 1.1: *Interaction Modalities in a Collaborative Working Environment*

different models of interaction, from direct control or teleoperation of the robot to robot with an autonomous and independent behaviour and ambient intelligence. In the Robot@CWE project (Stasse et al., 2008) a possible classification of the different types of interaction modalities in a collaborative context has been proposed. The classification, synthesized in Table 1.1, has been analysed for the following modalities:

- Direct Control: refers to interacting directly with the onboard Robot PC.
- Visual and Vocal Interaction: refers to a explicit interaction based on human gesture and/or speech.
- Tablet PC and PDA systems: refers to the possibility to communicate via a notebook, or a PDA equipped with a touch screen giving the possibility to work with a fingertip, instead of a keyboard or mouse.
- Remote Control: refers to the possibility of operating the robot remotely using teleoperation methodologies.

This work presents an interaction combining two of the modalities: Remote control and Tablet-PC HRI system, for a robot-human collaborative interaction for executing a task. It also present a modality not contemplate in the previous classification: Physical Interaction in the form of imitation learning and kinaesthetic teaching of a task skill.

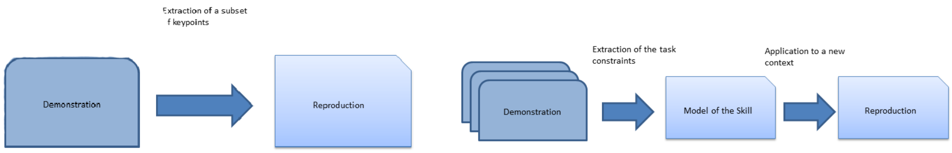
1.3 Learning Algorithms and Knowledge Representation

Current robot systems working in the industry perform repetitive tasks that are well known for the robot developers. The reproduction of this task required highly specific embedded controllers with an extensive knowledge of the robot's architecture and of its environment. For humanoid robots to collaboratively work with humans in and unstructured environment the robot must be able to perform dynamically changing tasks that require great adaptations to react to new constraints. To Foresee and program specialized controllers for every single task and situation that could be encounter seems like an impractical and unfeasible goal.

To develop the capacities expected for humanoid robots, flexible and generic control methods that can adapt to various tasks and robot's constraints are necessary. Robot Learning by Imitation, also referred to as Robot Programming by Demonstration, explores novel means of implicitly teaching a robot new motor skills. From (Billard, Calinon, Dillmann, & Schaal, 2008) some of the advantages that providing a robot with imitation abilities presents are:

- Provides a natural, user-friendly means of implicitly programming the robot.
- Constrains the search space of motor learning by showing possible and/or optimal solutions.

Robot Programming by Demonstration offers an implicit means of training a machine, such that explicit and tedious programming of a task by a human user can be minimized or eliminated. Studying and modelling the coupling of perception and action, helps to understand the mechanisms by which the self-organization of perception and action could arise during development.



(a) (b)

Figure 1.4: *Learning of a Skill: (a) Exact copy of a skill. (b) Generalization of a skill.*

A first approach, to PbD, the position of the end-effector and the forces applied on the object manipulated were stored throughout the demonstrations together with the positions and orientations of the obstacles and of the target. This sensorimotor information was then segmented into discrete sub-goals and into appropriate primitive actions to attain these sub-goals, Figure 1.4-a.

Generic approaches for learning a skill allow a robot to automatically extract the important features characterizing the skill. So first must be determine a metric of determining the weights one must attach to reproducing each components of the skill. Once the metric is determined, an optimal controller to imitate by trying to minimize this metric can be found. One common approach consists in creating a model of the skill based on several demonstrations of the same skill performed in slightly different conditions, Figure 1.4-b.

Robot systems need to make models of the representation of the learn skills. This models need to be generic and they should be accessible for itself and other robots sharing the working environment. Approaches to represent a skill can be broadly divided between two trends: a low-level representation of the skill, refer to as “trajectories encoding”, taking the form of a non-linear mapping between sensory and motor information. And, a high-level representation of the skill, refer to as “symbolic encoding”, that decomposes the skill in a sequence of action-perception units. Table 1.2, (Billard et al., 2008), summarizes the advantages and drawbacks of the different approaches.

	Span of the generalization process	**Advantages**	**Advantages Drawbacks**
Symbolic level	Sequential organization of pre-defined motion elements	Allows to learn hierarchy, rules and loops	Requires to pre-define a set of basic controllers for reproduction
Trajectory level	Generalization of movements	Generic representation of motion which allows encoding of very different types of signals/gestures	Does not allow to reproduce complicated high-level skills

Table 1.2: *Advantages and drawbacks of representing a skill at a symbolic/trajectory level*

1.4 Organization of the Document

This work deals with issues of interaction and collaboration between human and humanoid robots performing tasks in a collaborative working environment. The robotic system of the future is expected to be introduce into humans everyday lives, much in the same way as personal computers are a part of the present everyday human labor. Therefore the future robots must share the same workspace as men and it must assist them in performing their typical work, helping them to fulfil their common needs. All this means that human-robot teams must be form, working and collaborating to achieve shared goals in an effective and more productive way. In this work we present a framework for remote collaboration and learning of skills for human-robot teams working in a remote collaborative environment.

The general architecture and overview of the propose system is given in Chapter 2. The robotic platform use during this work is introduce as well as the other components of the system. Chapter 3 presents the modalities for human-robot collaborative interaction. Two types of collaboration can be identified, close collaboration and remote collaboration. The major part of Chapter 3 is dedicate on the concepts of human-robot remote collaborative work. Also the functionalities and use of the HRI employ for human interaction with the robot HOAP-3 is detailed.

Chapter 4 presents the learning algorithms and techniques implemented for a human operator to teach a robot new skills. On Chapter 5 the Shared Knowledge Database, developed on this work to provide a robot-robot communication to transfer the learned skill knowledge, is presented.

In Chapter 6 the experimental set-up and results on the demonstrators of the propose system for remote collaboration and learning of skills are expose. Finally conclusions of this work are presented in Chapter 7

Chapter 2

General System Architecture

Chapter 1 presented a general framework for human-humanoid work in collaborative environments. It shows general concepts for a functional architecture for autonomous behaviours in which robots interact with humans in the same environment and it can move and manipulate objects together with the human operator. Here a generic framework for a remote collaborative work environment its introduce.

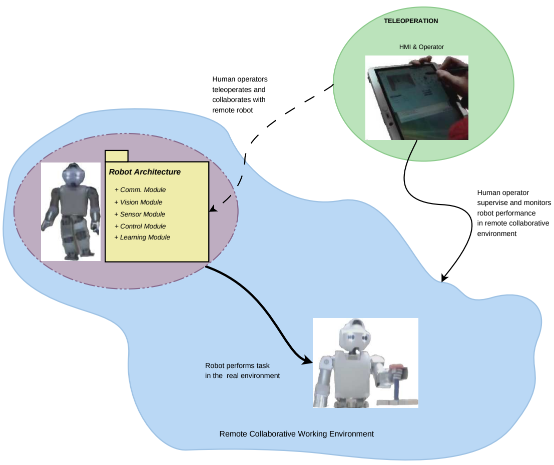


Figure 2.1: *Generic framework for human-humanoid remote collaborative work.*

In order to achieve such scenarios, the structure should aim at possessing five functional parts:

- an autonomous part
- an optional remote brain
- an intelligence shared with other robots so called ambient intelligence
- an interaction with the other services of the information system
- the Internet

In the framework presented in figure 2.1 the human operator would provide the remote intelligence. While a humanoid robot would perform task autonomously and interact with the environment and the remote human operator. For a humanoid robot, realizing collaborative work with a human, to have the capacity to achieve minimal autonomy the architecture must contemplate:

- Perception i.e., vision, sound, force, etc.
- Action, i.e., walk, manipulate, etc.
- Decision Making, i.e., computational services.

A robot-human-robot collaboration architecture was developed for a human operator and a local robot to interact with a robot located at a remote location to:

- Teleoperate and supervise remote robot performance.
- Collaborate between a robot-human team in execution of tasks.
- Allow a human operator to teach the performance of a task.
- Share skills knowledge between robots.

Figure 2.2 shows the propose architecture for the remote collaborative work and transfer of skills learning. A human operator monitors the state of the humanoid robot HOAP-3 in the remote collaborative task. By means of an human robot interface the human operator will supervise the robot and will send instructions for teleoperation to realize the tasks. HOAP-3 software server handles the robot movements and motions and maintains communication with the operator through the HRI on the completion state of the tasks.

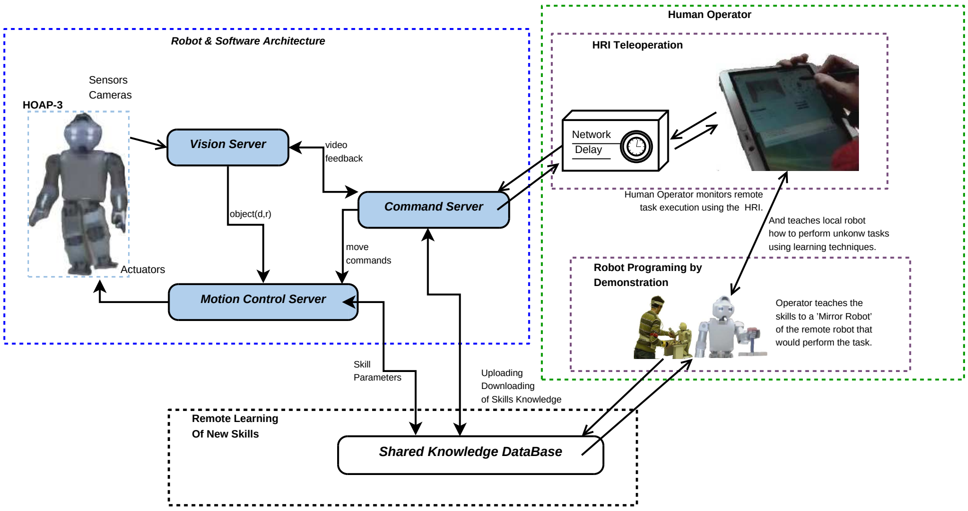


Figure 2.2: *The Collaborative Robot-Human-Robot Architecture.*

The humanoid robot would perform its requested task in an autonomous way, however during the execution of the requested task its possible that the robot encounter situations and objectives that it has not been previously trained to perform. When such situation arise the human operator will teach a HOAP-3 robot at its work site how to perform the desired task. Once the local robot has learned the task it will communicate the learned skill to the HOAP-3 robot at the remote working environment through a shared database. If the operator is not satisfied with the reproduction of the skill it can retrain the task until it is done in an appropriated way. The following sections introduce the subsystems belonging to the Collaborative Robot-Human-Robot Architecture.

2.1 HOAP-3 Robot

To test the propose systems the HOAP-3 Humanoid Robot was use as a test platform, Figure 2.3. The small humanoid robot HOAP-3 (Fujitsu: research and development, 2009) is about 60 cm in height, and weight about 8 kg. HOAP stands for "Humanoid for Open Architecture Platform" the model use in this work is an evolution from the previous HOAP and HOAP-2 robot family.

Figure 2.3: *The HOAP-3 Humanoid Robot*

The control architecture operates on RT-Linux mounted on a embedded PC-104 computer, Pentium 1.1 GHz processor with 512 Mb of RAM memory and a Compact Flash drive of 1 Gb capacity. Communications with the robot could be via a USB interface or with Wi-Fi IEEE802.11g communication. A 24V NiMH battery can be loaded for a 30 min autonomy operation.

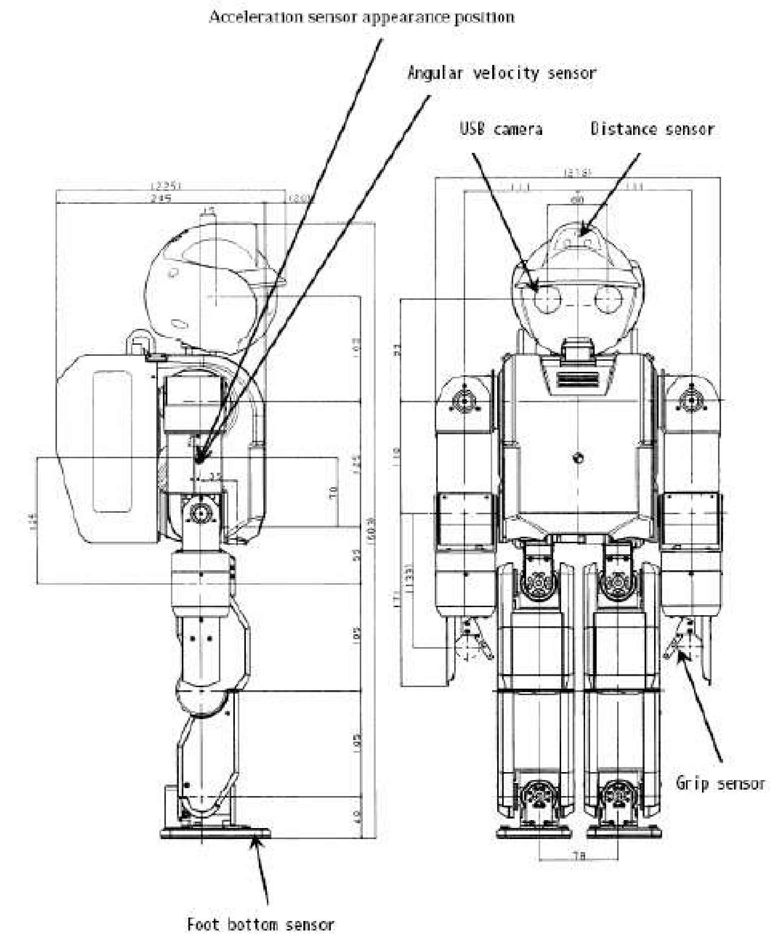


Figure 2.4: *The HOAP-3 Robot dimensions and sensors distribution*

HOAP-3 robot has has a total of 28 degrees of freedom, distributed like so:

- 6 DOFs for each robot arm, 4 DOFs for the arm, 2 DOFs for the hand.
- 6 DOFs for each leg.
- 3 DOFs in the head, for the pitch, yaw, and roll
- 1 DOF for the waist

Additionally the robot is equipped with the following sensors:

- Posture sensors (a gyroscope sensor and acceleration sensor).
- Contact sensors (in every corner of each foot).
- Grip sensor (in the thumb of the hands).
- Two USB cameras in the head

Figure 2.4 shows HOAP-3 Robot structure and sensors distribution. Its structure and sensor system permit to try different control architecture, thought to be used in a collaborative system.

2.1.1 HOAP-3 Software Architecture

The HOAP-3 software architecture is composed of three modules: A vision server for visual perception, a command server to handle the communication protocol with the HRI and the motion control server to control movements of the robot.

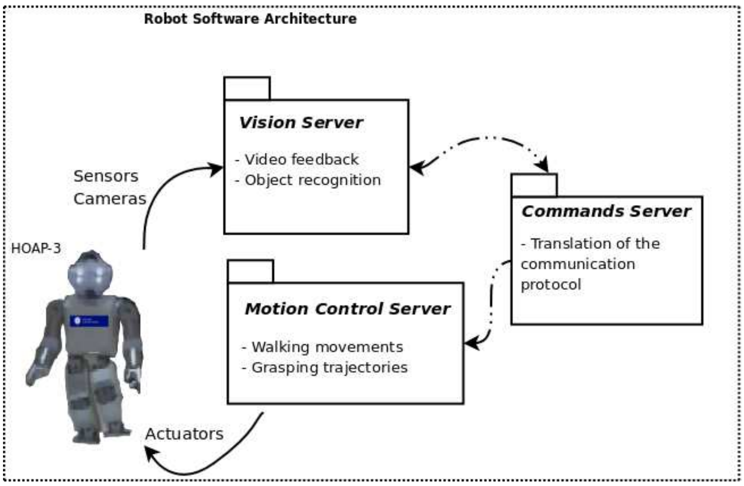


Figure 2.5: *The HOAP-3 Robot Software Architecture*

Figure 2.5 presents the HOAP-3 Robot Software Architecture. The functionalities of three modules are explained in the following sections:

(a) (b)

Figure 2.6: *Example of blob generation (a) and their color segmentation after thresholding (b).*

2.1.1.1 Vision Server

The Vision Server module handles the vision services for the HOAP-3 Robot. A software approach is adopted considering that techniques used should provide robust results satisfying the real-time restrictions of robotic applications. The vision module was implemented by D. Herrero-Pérez and is described in (Herrero-Perez et al., n.d.).

The color segmentation method consists of the selection of a prism for each channel in the HSV color domain. Instead of selecting all pixel for each channel, each channel is usually defined by the selection few pixels because other channel only provide information about saturation and brightness. The complexity of the $YCbCr$-HSV conversion is solved using a look-up table of variable resolution. In our application, a resolution of 7 bits is used as trade-off between memory used and computational cost saved.

When pixels are color labeled, similar regions are groped into blobs. Then, correspondences between blobs or their combination and possible color coded objects in the environment should be found. When blobs or their combination satisfy sanity checks and match with the color properties of some object, they are considered as the detection of an object. These detections are represented using bounding boxes around the object in Figure 2.6.

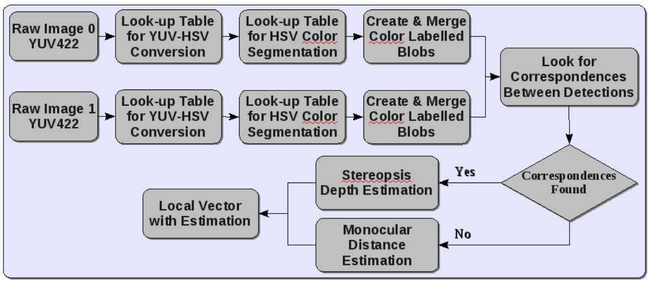


Figure 2.7: *Complete Vision Flowchart.*

The typical way for determining the depth using two-cameras vision systems is by stereopsis. The basis of stereopsis is epipolar geometry, which states that the line connecting optical centers of both cameras (baseline) intersects the image planes in the epipoles. A simplified case of stereopsis is the rectified configuration of cameras, which reduces the dimensionality of search space for a correspondence between perception in both cameras from 2D to 1D. This configuration consider both image planes are parallel, and hence, baseline is also parallel to image planes. This particular configuration sends the epipoles to infinity. In addition, epipolar lines of all possible detections coincide with the image's rows. Thus, correspondences between detection of both images can be found by matching pixels linewise (horizontal lines instead of general ones).

The image processing pipeline is shown in Figure 2.7.

2.1.1.2 Motion Control Server

The motion control server is in charge of controlling the joint actuators and of sending the appropriate commands for the walking, turning and grasping motions. The motion control server receives orders from the command server of what movements is require to perform; it also receives distance and orientation parameters of the tracked object from the vision server.

Once that a command has been received, the robot distinguishes if it is a

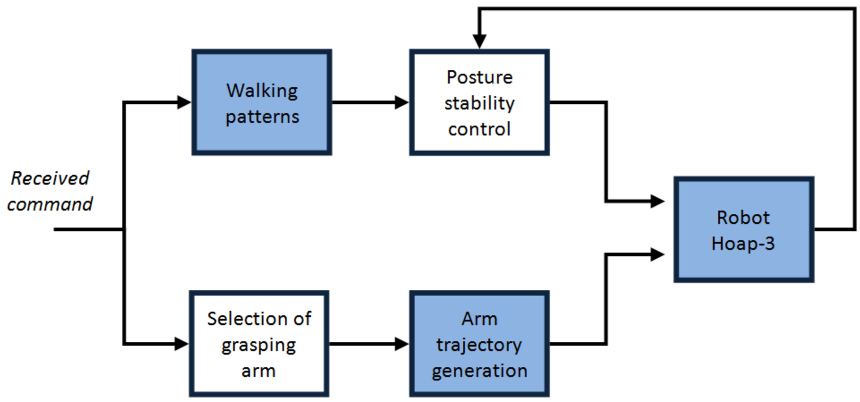


Figure 2.8: *Hoap-3 Motion Control Strategy*

command for the walking generation or for the arms movement. Figure 2.8 presents the control strategy.

The walking patterns of the robot have been designed base on the theory of the 3D Linear Inverted Pendulum Mode presented in (Kajita, Kanehiro, Kaneko, Yokoi, & Hirukawa, 2001). The posture stability control has not been implemented yet, but several studies are being done in order to accomplish it (Monje, Pierro, & Balaguer, 2008). The trajectory of the arm is evaluated online through the algorithm of kinematic inversion presented in (Siciliano, Sciavicco, Villani, & Oriolo, 2009), once that the vision server provides the distance and the orientation from the object. The orientation reference for the object is calculated with the support of the unit quaternion presented in (Chiaverini & Siciliano, 1999).

Also the motion control server communicates with the shared knowledge database to obtain the learned skill parameters necessary for its reproduction.

2.1.1.3 Command Server

The Command Server handles communications with the HRI. It is in charge of translating the Robot Command Protocol (RCP), Chapter 3, to forward the instructions from the operator back to the other services and gives the adequate responses to the operator and the HRI. RCP is a text-based protocol which has its roots in UNIX protocols like SMTP or FTP. Each RCP command is a text string terminated by a newline character, such as, `GOTO OBJECT(<object_id>)`, `GRAB OBJECT(<object_id>)`, etc. And give the adequate responses from robot execution back to the operator and the HRI, like `OK COMMAND <command_id> COMPLETED`. The commands server will receive and process all requests that must be handled by the robot like capture video frames, move or grasp, or use the learned skill with the shared knowledge database.

2.2 Human Robot Interface

A HRI was developed in the frame of this project to enable a human operator to control and monitor the Robot. The HRI is user friendly and it gives an intuitive way for a non expert user to interact with the humanoid robot HOAP-3. Its main functionalities are:

- Connect to one robot at a time via (Wireless) TCP/IP
- Display streaming video from the robot camera
- Drive robot's movements and speed
- Move robot's head (tilt and pan)
- Send high-level commands (tasks) to the robot

A Robot Command Protocol was design to allow the implementation of the HRI functionalities (Blasi & Stasse, 2008). The HRI will generate commands according to this Robot Command Protocol (RCP); these commands would be interpreted by the HOAP-3 robot command server module into the appropriate instructions for the robot to perform the requested task.

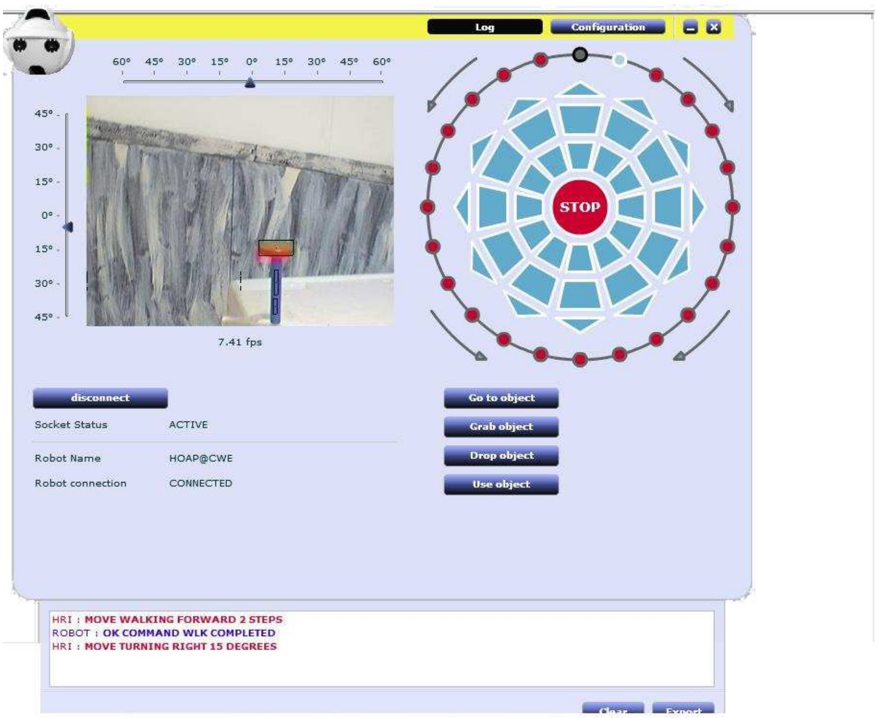


Figure 2.9: *The Human Robot Interface for the teleoperation of the HOAP-3 robot*

Figure 2.9 shows the main aspect of the Human Robot Interface. A more detailed explanation on the characteristics of the HRI and the RCP protocol can be found on Chapter 3

2.3 Learning Techniques

The skills learning module can be essentially divide in two major parts:

- Acquisition of skills from human teacher demonstrations

- Reproduction of those skills from learned models by a robot

The skills acquisition part accounts for gathering data demonstrated by a teacher and for further statistical processing of these data into a model of a skill. The model of the skill is represented by a set of parameters sufficient to reconstruct relevant trajectories of the task. After learning, the robot reproduces a task to obtain a confirmation from a human teacher that training was successful; once this confirmation is received, the robot submits the learned model to the shared knowledge database, where the model becomes accessible.

The reproduction part aims at generating motions from the learned tasks models. Once the operator sends a command to manipulate an object through the HRI interface, the reproduction algorithm retrieves a relevant model of a skill, reads position of the object from the vision server and gets motors feedback; based on this input information, the reproduction algorithm generates motions to accomplish the task.

A more detailed explanation on the learning techniques can be found on Chapter 4

2.4 Shared Knowledge Database

In order for robot systems to communicate with each other to transfer the learned models of the skills a "Shared Knowledge Database" was implemented. The shared knowledge database is an interface that allows robot-to-robot interaction, a robot working on the collaborative task would be able to access the shared knowledge database and upload or download the learned skill knowledge for the reproduction of a task.

The shared knowledge database communicates with the HOAP-3 server command server module and motion control server module for the learning of parameters to reproduce new skills.

The shared knowledge database holds commonly accessible data for that the Robots could learn and reproduce a skill. The robots can upload or download the learning of the skills when is available. They will also send or receive acknowledgement signals when new data of the skill is uploaded to the share database. The Robots communications through the shared knowledge database will use a TCP/IP private network. A more detailed explanation on the characteristics of Shared Knowledge Database can be found on Chapter 5

Chapter 3

Human Robot Interaction

In Chapter 1 we talk about the human centred and collaborative centred robotics of the future, where robotics would leave the structured, automated, industrial environments of the present day vehicle and manufacturing factories and humanoid robots will be deployed in the humans conventional environment, sharing the home and the workspace with human users, helping them to perform everyday work.

Clearly for this vision to be accomplished a major focus of research is in the interaction between robots and humans, as this presents one of the main tasks it has to be achieved if we want a world where humans and robots can work together.

The optimal ideal for the human-robot interaction is for the human operator to accept and recognize the robot system just as one more partner in a working team compose of multiple human and robotic agents. A human-robot team can present many advantages. Robots can be used in order to cover human limitations or to assist them in numerous tasks.

A robot can be deployed at sites that are too dangerous or inaccessible to humans, like in a disaster rescue mission or a space environment. Robots can also be of great assistant for a human worker at a construction scenario, taking of most of the workload in a transportation or an assembly task and performing more risky activities. A robot partner can also perform precise or sensible tasks in an industrial or factory scenario.

This Chapter focus on the human-robot interaction require to accomplish collaborative work in remote collaboration environments. It also demonstrate a conceptual application of a human-robot team performing task collaboratively in a space scenario.

3.1 Human Robot Collaboration

This work follows on the conceptualizations of collaboration propose by Paolo Pierro (Pierro, 2009). His work define two kinds of classifications of collaborative working environments, close collaboration and remote collaboration.

3.1.1 Close Collaboration

A close collaboration can be defined as the collaborative working environments where humans and robots work collaboratively in the same working cluster. This is the case when robots work in close area with humans, for instance when a robot-agent is handling the same object together with humans.

As an example of close collaboration task we can consider a coordinating task between a human and a humanoid robot such as transporting a long or a heavy object. This kind of collaboration can be applied to a building site scenario where clear benefits in the use of robots for unpleasant and dangerous work show the value of this technology (Pierro, 2009).

The human operator would generally be required to act as the master, this means that he takes the initiative of a task execution. The robot partner must be equipped with a sensor system, 6-axis force/torque sensors at the wrists and the feet, that allowed it to process the information of the reaction forces once it interacts with the human through the object they are carrying. A novel control scheme for the collaboration human-robot in manipulation tasks is proposed in (Pierro, Monje, & Balaguer, 2008) and (Monje et al., 2008).

A second example of close collaboration is a Human operator teaching a task to a robot by means of kinaesthetic teaching. Here the human operator, the teacher, would guide the robot arms through the motions that constitute a learned skill. The process of kinaesthetic teaching would be further detail in Chapter 4.

3.1.2 Remote Collaboration

Remote collaboration consist of the collaborative working environments where humans and robots work collaboratively in two separate spaces: a task working cluster and a supervision, planning working cluster. This case focuses on monitoring, mission planning and teleworking, for instance a teleoperation or telepresence task.

As example of remote collaboration we consider a collaboration when the human operator does not share the environment with the robot and human-robot interaction occurs through information technologies interfaces. This could be the case of a collaboration between humanoid robots and humans in order to achieve tasks in space environments.

The human operator would need to control the robot remotely using a control interface, also information on the physical situation of the environment must be readily available to provide the operator with some situation awareness. Similarly the robot must have the capacity to perceive its environment and it should be able to move along it, detect and interact with objects, etc. The robot would need to present a sufficient level of autonomy to perform in the environment on its own with only remote supervision from the human operator.

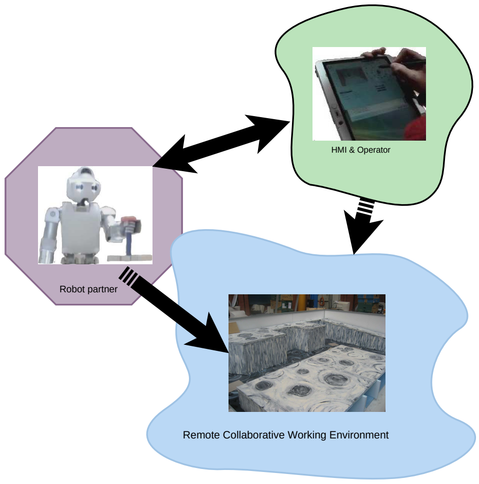


Figure 3.1: *Remote Collaboration Modalities.*

The following section would expand on the concepts on human-robot remote collaborative work presented here and in Chapter 2. To illustrate the questions of remote collaboration two examples are presented: teleoperation of a humanoid robot and remote learning of skills.

3.2 Remote Collaborative Interaction with a Humanoid Robot

As stated before a human-robot team could present several advantages. For (Fong, Thorpe, & Baur, 2002) this advantages could be particular beneficial if we treat a robot as a partner, this, however, need to enable humans and robots to collaborate, this is, to engage each other in dialogue and to assist each other to jointly solve problems. In the work of (Fong et al., 2002) a model for collaboration is proposed to address this needs. In their model instead of a supervisor dictating to a subordinate, the human and the robot engage in dialogue to exchange ideas, to ask questions, and to resolve differences. An important consequence of collaborative control is that the robot can decide how to use human advice: to follow it when available and relevant; to modify it when inappropriate or unsafe.

Following a similar approach in a previos work (Herrero-Perez et al., n.d.), a DH↔DR concept is introduced, DH↔DR means, what is Difficult for the Human will be Done by the Robot and what is Difficult for the Robot will be Done by the Human. This is advantageous because it works for the strengths of all partners and allows a human-robot team to be more productive. In the case of remote collaboration, were the human and the robot does not share the same space, this DH↔DR collaboration focuses on a human-robot interaction were the human is not a supervisor but a partner in which the robot can look for assistance in the decision making process. This level of collaboration allow a human operator to free itself from the high work load of requiring time-critical or situation-critical response, while the robot is able to perform with greater degree of autonomy. It also plays on the strength of the human partner in the use of the more powerful perception and cognition capacities of a human to recognize situations on the environment, and complement this with the higher computational capability of a robot processor to determine sizes, distance to object, physical constraints, etc.

3.2.1 Teleoperation of the Robot

Teleoperation is a problem that have long been a focus of research in the robotic community. (Chen, Haas, & Barnes, 2007) summarize the most common factors that can affect the remote perception and manipulation capacities of the operator when working with a teleoperation system.

- Limited FOV: The use of cameras to capture the environment in which the robot is navigating sometimes creates the so-called “keyhole” effect. Important distance cues may be lost and depth perception may be degraded when FOV is restricted.

- Orientation: In order to successfully navigate in the remote environment, the robotic operator needs to have a good sense of orientation, both globally and locally. Also the operator needs to be aware of the robot's attitude. Both which can be hard to estimate on a mobile platform, and require a lot of engagement and work load from the operator.

- Degraded Depth Perception: Projecting 3-D depth information onto 2-D display surface results in compressed depth perception. This is worse with the ground robots because of their low viewpoints. Degraded depth perception affects the teleoperator's estimates of distance and size and can have profound effects on mission effectiveness.

- Degraded Video Image: Teleoperation is often prone to poor spatial awareness of the remote environment due to the impoverished representations from video feeds, which could leave out essential cues for building teleoperator's mental models of the environment.

- Time Delay: Refers to the delay between input action and (visible) output response, and is usually caused by transmitting information across a communications network. Latency over 1s greatly affects the control strategy that can be use by a teleoperator.

- Motion: Performing computerized tasks or simulated teleoperation tasks on moving platforms is difficult. Besides perceptual and psycho-motor aspects, motion also made cognitive tasks more challenging.

Performing tasks as a robot-team in a remote collaborative frame, instead of a direct teleoperation task, allows to side step and overcome some of the issues mentioned above. For instance, by giving the remote robot greater autonomy the operator is free from some of the critical tasks that require of it to navigate the remote robot system, a problematic like keeping good orientation an altitude estimates of the robot is not an issue since the robot navigate itself. Also while limited FOV, depth perception and degraded video image still negatively impact the operator capacity to observe and understand the environment. Working as a team with the remote robot will provide the operator with better information from the robot computer vision algorithms while preserving the human operator greater cognitive capacities on the overall goals of the task.

A human-robot team could successfully engage in a remote collaborative working environment for the accomplishment of various task. A teleoperated remote collaborative interaction with a Humanoid Robot requires for a human operator, acting as a partner, with a communications interface that allows it to communicate with the robot partner, and offer support and assistance in the decision making process. It also is require for a remote robot to posses a minimum degree of autonomy and situation awareness of its environments, visual and sensory perception. The robot partner must be able on its own to realize a number of activities with out requiring a direct control operation from the human partner in order to be a meaningful collaboration. And also must relay back information to the operator of the remote environment state.

To achieve a functional remote collaboration, the human-robot interaction is a major aspect of the framework. In order to communicate and engaged in a productive dialogue it is necessary to develop interfaces with great functionality. Also a understandable and expandable communication protocol is vital for a useful human-robot communication. The following sections would deal with the HRI and the robot command protocol implemented in the framework of this thesis.

3.2.2 Remote Learning of Skills

As it has been discuss in previous sections, humanoid robots working in collaborative environment with human partners must be flexible enough to deal and solve a great number of tasks and scenarios, most of them who would have been difficult to foreseen during development. This presents a significant problem at the time of programming behaviours and controller for a robotic system of the characteristics and functionalities discuss here. In order to find a solution to this problem a series of learning algorithms has been developed within the robotics community to give a robot the capability to learn to execute new task and behaviours.

In this work we utilize learning by imitation techniques, also known as Robot Programming by Demonstration (Billard et al., 2008), to learned models of the task motion dynamics that must later be reproduce. To teach this models, the human operator must present the robot with various demonstration of the task motion, later the robot would learn a probabilistic model of the demonstrate models as Gaussian Mixture Models. Chapter 4 deals in more detail with how to learn and reproduce this models.

An additional problem for a Remote Learning of Skills collaboration presents in by which means would the knowledge of a task can be transfer remotely to a robot, that is not working in the same physical location. In this work a "proxy" robot of the remote robot is used to teach locally the task demonstrations. Once the local robot has learn the models of the task, and the operator is satisfied with its performance, the local robot needs to transfer this skills to the local robot. For transfering this skills knowledge between the robots a Robot-to-Robot collaboration is presented trough the use of a shared interface of skills knowledge, this interface will de denominate the Shared Knowledge Database and it will be expanded on Chapter 5.

3.3 Human-Robot Interface

In order for a remote operator to control and monitor the robot actions on the remote collaborative environment the human operator needs to connect to the environment trough a Human-Robot Interface. The HRI needs to provide the remote operator with the opportunity to request the robot to perform certain actions on the environment, also the HRI needs to present the human operator with relevant information on the state of the remote collaborative environment, such as video feedback from the remote environment, state of the robot process and sensors, etc.

3.3.1 Graphic Interface and Functionalities

During the Robot@CWE project a HRI was design with one of the project partners (Blasi & Stasse, 2008), (Blasi, Weiss, Stasse, & Hernández, 2009). The HRI user interface is shown in Figure 3.2. Its main functionalities are:

- Connect to one robot at a time via (Wireless) TCP/IP.
- Display streaming video from the robot camera
- Drive robot's movements and speed
- Body rotation controls
- Move robot's head (tilt and pan)
- Send high-level commands (tasks) to the robot

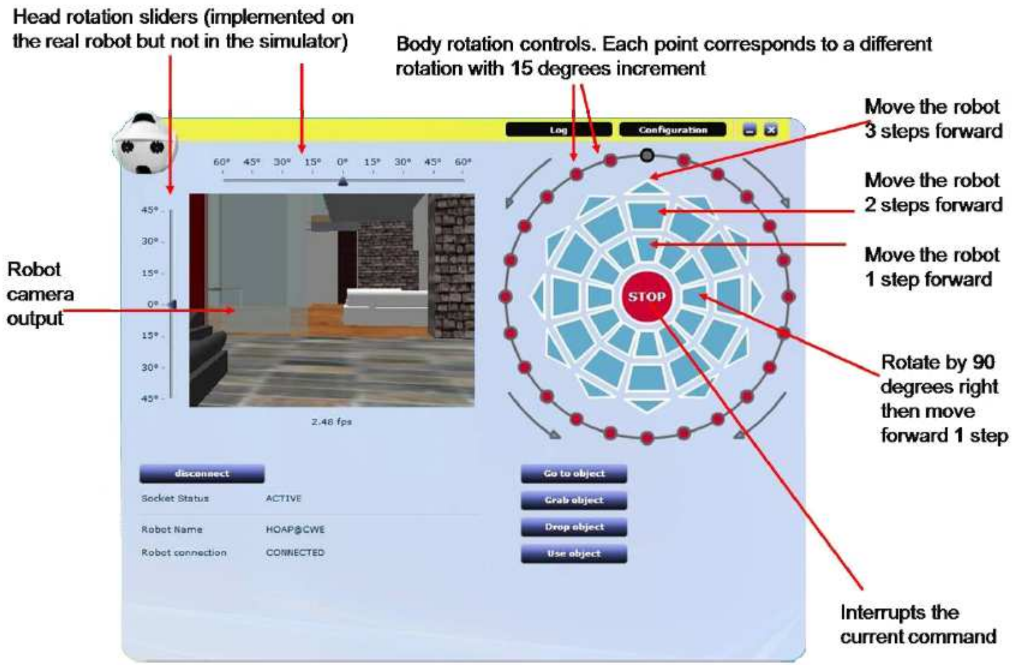
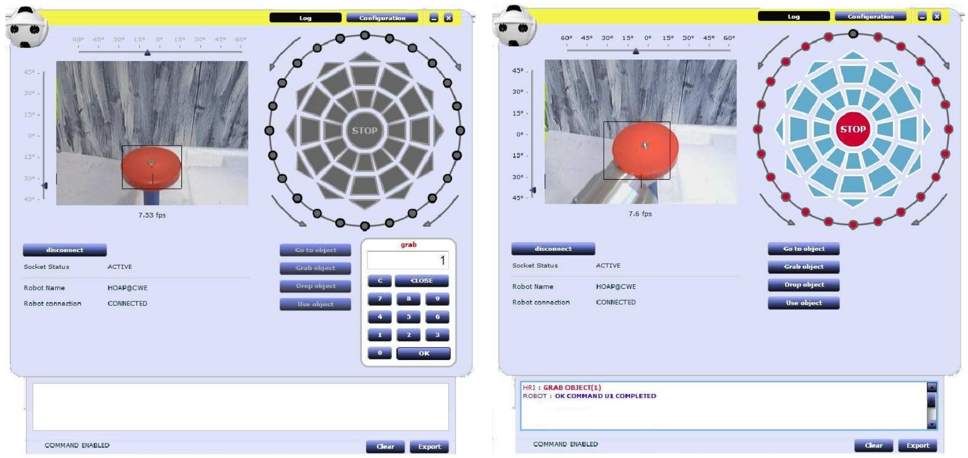


Figure 3.2: *HRI Interface and Main Components*

The main functionality in the HRI user interface is the graphical control component, shown in the upper right part of the window, which allows the operator to move the robot in several directions at different speeds, rotate it and stop it. And the high-level button controls which allows the operator to request the robot to perform several high-level actions, such as, go to a location and grab, drop or use an object. Further functionalities of the HRI, not shown in Figure 3.2, are the log window, where all the commands sent to the robot and related responses can be seen and the configuration window, which allows the specification of robot's IP address and ports, desired streaming speed, video encoding and network latency.

In the case of spatial applications, several problems have to be addressed. The main problem is the time delay. For a moon-earth communication the delay could be around 3 seconds and for mars-earth communication the delay could



(a) (b)

Figure 3.3: *Example of* GRAB OBJECT(<object_id>) *(a) HRI send request. (b) Robot perform action.*

be up to 10 minutes, as the time delay increases with the distance from the earth (Ferre, Buss, Aracil, Melchiorri, & Balaguer, 2007). In order to simulate the conditions and problems that arise in a real space communications application, a controllable time delay module has been added to the HRI. This module will permit to set a variable delay and to test the performance of the proposed teleoperated system in space environments.

3.3.2 Robot Command Protocol

Along side with the HRI a Robot Command Protocol a was design for the communication between the human an the robot (Blasi & Stasse, 2008), (Blasi et al., 2009). Design goals for this protocol were: simplicity, generality, flexibility and expressiveness. A powerful characteristic that leads to both flexibility and expressiveness can be identified as orthogonality, which can be achieved by clearly separating disconnected functionalities while at the same time allowing their combination without unneeded constraints (Pierro et al., 2009). **RCP** is a text-based protocol which has its roots in Unix protocols like SMTP or FTP. Each **RCP** command is a text string terminated by a newline character. The RCP protocol present various attractive characteristics:

- The resulting protocol is simple to understand and implement.
- Support for robot control can also be easily added to programs different from our HRI.
- The protocol is lightweight; since the robot has limited computational resources that can be dedicated to command parsing, this was an important design goal.
- The human-readable text commands make debugging easy.
- The protocol is also general in that it has not been designed for a specific target robot, but for a generic target robot described by a high-level robot model.

RCP was originally defined in (Blasi & Stasse, 2008) and can be decomposed into several sub-protocols, like the RoboLink protocol (*AIST presentation at DSIG Plenary Meeting*, 2005) is organized into "profiles". Each sub-protocol contains a set of commands used for a single purpose. The list of **RCP** sub-protocols is shown in Table 3.1.

As an example of a command we review the basic movement sub-protocol. The basic movement sub-protocol defines movements of the body and head for teleoperation of the robot. A general **MOVE** command presents the following structure:

```
MOVE <movement_type> <direction> <count> <unit>
```

Currently three movement types of the command are supported:

Name
Connection
Control negotiation
Basic movement
Direct command execution
Configuration
Sensor reading
Positioning
Notification
Goal-setting
Object grabbing
Strategy selection

Table 3.1: *RCP Sub-protocols*

```
MOVE WALKING [FORWARD|BACKWARD] <count> STEPS
MOVE TURNING [LEFT|RIGHT] <count> DEGREES
MOVE HEAD [UP|DOWN|LEFT|RIGHT] <count> DEGREES
```

The **MOVE** command is a good example of the flexibility of the protocol, in that its structure allows adding new movement types easily. For example we could add a new BOWING movement. As movements are not an instant action, the robot can send multiple replies in response to a **MOVE** command:

```
OK COMMAND <command_id> QUEUED
OK COMMAND <command_id> STARTED
OK COMMAND <command_id> COMPLETED
```

The goal-setting sub-protocol presents a more advanced way of controlling robot's movements. This sub-protocol allow for higher order request to be send to the robot with a task goal oriented movement in mind. In order to tell the robot to go towards, grab or drop an object the user issues the commands:

```
GRAB OBJECT(<object_id>)
```

```
GRAB OBJECT(<object_id>)
DROP OBJECT(<object_id>)
```

Also the user can send a command to “use” and object, the syntax of this command is the same as the previous ones,

```
USE OBJECT(<object_id>)
```

but this would involve some decision on the part of the robot about which strategy should be used for executing the operation considering the database of know learn skills of the robot. For this purpose a specific strategy selection sub-protocol has been defined with which both the robot and the user collaborate in deciding which strategy to use for the operation at hand. The strategy selection dialogue is initiated from the robot side with a request listing the possible strategies:

```
SELECT STRATEGY FOR <cmd_id> [<strategy_1>, <strategy_2>
                    ...  <strategy_n>]
```

Then the user chooses a strategy and communicates its decision with the command:

```
USE STRATEGY FOR <command_id> <strategy>
```

In this work a subset of the full protocol described above was implemented. A reference of all the commands currently defined in the RCP protocol is shown in Table 3.2.

Note from the above lines that IDs are used to refer to commands. Every command is assigned an ID by its receiver (i.e. the robot or the HRI). Then the receiver sends the counterpart a reply indicating whether the command has been accepted or not. Successful replies always start with ”OK”, while unsuccessful ones start with ”KO”.

Sub-protocol	**Command**
Connection	CONNECT <profile> DISCONNECT
Control negotiation	CONTROL BEGIN CONTROL END
Basic movement	MOVE <movement_type><direction><count><unit> STOP
Direct command execution	DIRECT <command>
Configuration	QUERY PARAM <parameter_name> SET <parameter_name> <parameter_value>
Sensor reading	QUERY SENSOR [<label>,... , <label>]
Positioning	QUERY POSITION POSITION <x > <y > <confidence>
Notification	tbd
Goal-setting	GOTO OBJECT(<object_id>) GOTO <x> <y>
	GRAB OBJECT(<object_id>) USE OBJECT(<object_id>) DROP OBJECT(<object_id>)
Strategy selection	SELECT STRATEGY FOR <cmd_id> [<strategy_1>, ..., <strategy_n>] USE STRATEGY FOR <cmd_id > <strategy>

Table 3.2: *RCP commands*

Chapter 4

Learning Algorithms

In this work we have presented a framework for human-robot collaborative work and remote collaboration environments, Chapter 2. As we have stated before, the humanoid robots must be able to adapt to the human environment, therefore not only the human appearance is important but the algorithms used for its control require flexibility and versatility. Robots working alongside humans means there will be continuously changing environments and a huge variability of tasks that the robot is expected to perform, thus the robot should have the ability to continuously learn new skills and adapt the existing skills to new contexts.

Programming by Demonstration (PbD, has appeared as one way to respond to this growing need for intuitive control methods. PbD formulates user-friendly methods by which a human user can teach to a robot how to accomplish a given task, simply by demonstrating this task (Gribovskaya, Zadeh, Mohammad, & Billard, 2010).

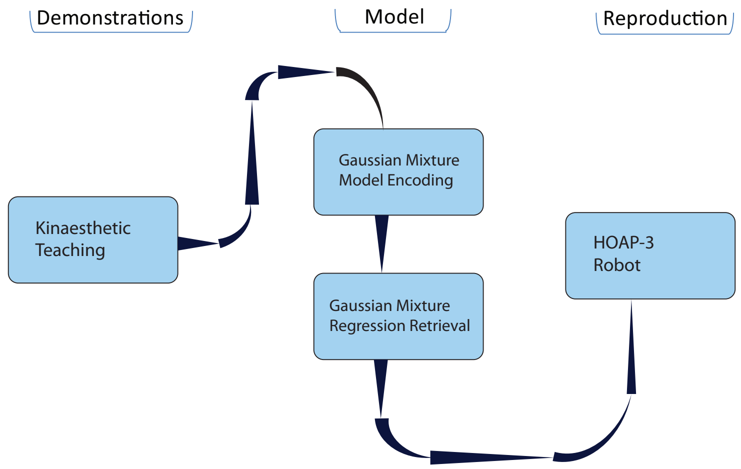


Figure 4.1: *Flowchart of the learning and reproduction of the skill process.*

The following sections described the methodology employ during this work for learning and teaching manipulation skills to the robot. This work follows on the Programming by Demonstration framework developed at the Laboratoire d'Algorithmes et Systemes d'Apprentissage (LASA) at École polytechnique fédérale de Lausanne (*LASA: Laboratoire d'Algorithmes et Systemes d'Apprentissage*, 2010).

4.1 Programming by Demonstration

Robot Programming by Demonstration, also know as Imitation Learning, appeared as a promising route to automate the tedious manual programming of robots and as way to reduce the costs involved in the development and maintenance of robots in a factory. The imitation learning approaches focuses on the development of algorithms that are generic in their representation of the skills and in the way they are generated (Billard et al., 2008).

Implementing robot PbD methods offers the possibility of making learning faster, in contrast to tedious reinforcement learning methods or trials-and-error learning. Also, the methods being user-friendly, makes PbD a important technique to facilitate and enhance the application of robots in human daily environments. Initial means of providing demonstration of the task to the robot implied the teleoperation of the end-effectors. Later more user-friendly interfaces were use, like vision recognition, data gloves, laser range finder or kinaesthetic teaching.

Robot PbD focus on developing algorithms that are generic in their representation of the skills and in the way they are generated. To reproduce a skill in a new situation, the robot can not simply copy an observed behaviour; it must have the capability to generalize. Current approaches to represent a skill can be broadly divided between two trends: a symbolic level representation, described by the sequential or hierarchical organization of a discrete set of primitives that are predetermined or extracted with predefined rules. Or a trajectory level representation, described by temporally continuous signals representing different configuration properties changing over time (Calinon, 2009).

Observing multiple demonstrations can help at generalizing a skill by extracting which are the task requisites. One trend of work investigates how statistical learning techniques deal with the high variability inherent to the demonstrations. (Calinon, Guenter, & Billard, 2007) used Gaussian Mixture Models (GMM) to encode a set of trajectories, and Gaussian Mixture Regressions (GMR) to retrieve a smooth generalized version of these trajectories and associated variabilities.

4.2 Learning of the Skill

In this work we focus on teaching, a humanoid robot, manipulation tasks that requires both coordinated motion of limbs and accurate positioning of an end-effector. The robots position and orientation control are learned as multivariate Dynamical Systems using a PbD framework. Dynamical Systems (DS) offer a particularly interesting solution to an imitation process aimed at being robust to perturbations which is robust to dynamical changes in the environment (Billard et al., 2008).

We follow a framework presented on (Gribovskaya & Billard, 2009) that allows learning non-linear dynamics of motion in manipulation tasks and generating dynamical laws for control of position and orientation. The strength of the method is three-fold: i) it extracts dynamical control laws from demonstrations, and subsequently provides concurrent smooth control of both position and orientation; ii) it allows to generalize a motion to unseen context; iii) it guarantees on-line adaptation of the motion in the face of spatial and temporal perturbations.

A motion in a manipulation tasks consists of two parts: a transport phase, which allows the hand to reach the object, and a grasping phase, which preshape the fingers. The transport phase is described by a translational and an orientational component. The translational component brings a robot's hand in the proximity of a manipulated object and the orientation component aligns the hand with the object. To successfully accomplish the manipulation task and generate smooth, natural-looking motions the humanoid robot should reproduce both of these components simultaneously, in a coordinated manner. Therefore, the learning and reproduction of the position and orientation motions should be encoded simultaneously, replicating a coordinated pattern. Figure 4.1 illustrate the process followed for the learning and reproduction of the task.

4.2.1 Kinaesthetic Teaching

For demonstrating the motions of the task to the robot we use kinaesthetic teaching. The kinesthetic teaching process (Calinon, 2009), consists in using the motor encoders of the robot to record information while the teacher moves the robot's arms.

For demonstrating the task the robot motors are set in passive mode, standing beside the robot a human demonstrator moves simultaneously the robot

(a) (b)

Figure 4.2: *Kinaesthetic Teaching of the Skill: (a) Teaching of a spoon in the cup task. (b) Teaching of a grasp task.*

arms. The kinematics of each joint motion are recorded at a rate of $1000Hz$ during the demonstrations and were then re-sampled to a fixed number of points. The robot is provided with motor encoders for every DOF, except for the hands and the head actuators. The process is illustrate in Figure 4.2 for the teaching of two task with the humanoid HOAP-3.

4.2.2 Learning the Motion Dynamics

After demonstrations of the task a model of the learn skill must be generated. A time independent model of the motion is estimate through a set of first order non-linear multivariate dynamical systems. DS provides an effective mean to encode trajectories through time-independent functions that define the temporal evolution of the motions.

We define a variable ξ that unambiguously describe the state of the robot end-effector. Let us assume that the state of our robotic system ξ can be governed by an Autonomous Dynamical System, define as the tuple $< X, f, T >$, with $f : t \to f^t$ a continuous map of X onto itself.

And further assume that the transition map function $f : R^n \rightarrow R^n$ is a non-linear, continuous, and continuously differentiable function, with a single equilibrium point $\dot{\bar{\xi}} = f(\bar{\xi}) = 0$.

Let the set **M** of N-dimensional demonstrate data points $\{\xi_i, \dot{\xi}_i\}_{i=0}^M$ be instances of a global motion governed by a first order autonomous ordinary differential equation (ODE):

$$\dot{\xi}(t)^M = f(\xi(t)^M), \tag{4.1}$$

where $\xi^M \in R^n$, and its time derivative $\dot{\xi}^M \in R^n$ are vectors that describe the robot motion. The problem then consists in building a stable estimate $\hat{f}$ of f based on the set of demonstrations. Without loss of generality, we can transfer the attractor $\bar{\xi}$ to the origin, $\bar{\xi} = 0$, so that $f(\bar{\xi}) = f(0) = 0$ and by extension $\hat{f}(\bar{\xi}) = \hat{f}(0) = 0$.

To build the estimate $\hat{f}$ from the set of demonstrated data points $\{\xi_i, \dot{\xi}_i\}_{i=0}^M$ we follow a statistical approach and define $\hat{f}$ through a Gaussian Mixture Model (Gribovskaya & Billard, 2009).

4.2.2.1 Gaussian Mixture Models

The GMMs define a probability distribution $p(\xi^i, \dot{\xi}^i)$ of the training set of demonstrated trajectories as a mixture of the K Gaussian multivariate distributions $\mathbf{N^k}$

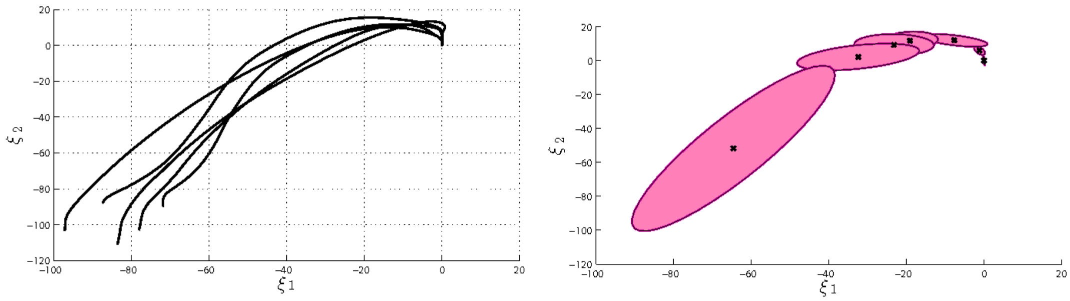
$$p(\xi^i, \dot{\xi}^i) = \frac{1}{K} \sum_{k=1}^{K} \pi^k N^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k) \tag{4.2}$$

Where π^k is the prior probability; $\mu^k = \{\mu^k_{\xi}; \mu^k_{\dot{\xi}}\}$ is the mean value; and

$$\Sigma^k = [\begin{array}{cc} \Sigma^k_{\xi} & \Sigma^k_{\xi\dot{\xi}} \\ \Sigma^k_{\dot{\xi}\xi} & \Sigma^k_{\dot{\xi}} \end{array}] \tag{4.3}$$

is the covariance matrix of a Gaussian distribution $\mathbf{N^k}$

The probability density function of the model $N^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k)$ is then given



(a) (b)

Figure 4.3: *Illustration of the learning process: (a)Training data of the task. (b) GMM of the learned motion.*

by:

$$N^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k) = \frac{1}{\sqrt{(2\pi)^d |\Sigma^k|}} \exp \frac{-1}{2}(([\xi^i, \dot{\xi}^i] - \mu^k)^T (\Sigma^k)^{-1}([\xi^i, \dot{\xi}^i] - \mu^k)) \tag{4.4}$$

By considering an adequate number of Guassians, and adjusting their means and covariances matrix parameters, almost any continuous density can be approximate to arbitrary accuracy. The form of the Gaussian mixture distribution is governed by the parameters π^k, μ^k, Σ^k. The model is initialized using the k-means clustering algorithm starting from a uniform mesh and is refined iteratively through Expectation-Maximization (EM) for finding the maximum likelihood function of equation 4.2.

$$\ln p(\xi^i, \dot{\xi}^i) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi^k N(\xi_n^i, \dot{\xi}_n^i | \mu^k, \Sigma^k)\} \tag{4.5}$$

The theoretical analysis of GMMs can be found on (Mclachlan & Peel, 2000), (Vlassis & Likas, 2002), (Dasgupta & Schulman, 2000). Figure 4.3 illustrates the encoding of a training data set $\{\xi_i, \dot{\xi}_i\}_{i=0}^M$ into a model of mixtures of Guassians.

To generate a new trajectory from the GMM, one then can sample from the probability distribution function $p(\xi^i, \dot{\xi}^i)$, this process is called Gaussian Mixture Regression.

4.2.2.2 Gaussian Mixture Regression

The GMM computes a joint probability density function for the input and the output so that the probability of the output conditioned on the input are a Mixture of Gaussians. So it is possible after training, to recover the expected output variable $\hat{\xi}$, given the observed input ξ. Taking the conditional mean estimate of $p(\dot{\xi}|\xi)$, the estimate of our function $\hat{\dot{\xi}} = \hat{f}(\xi)$ can be expressed as a non-linear sum of linear dynamical systems, given by:

$$\hat{\dot{\xi}} = \sum_{k=1}^{K} h_k(\xi)(\Sigma^k_{\dot{\xi}\xi}(\Sigma^k_{\xi})^{-1}(\xi - \mu^k_{\xi}) + \mu^k_{\dot{\xi}}) \tag{4.6}$$

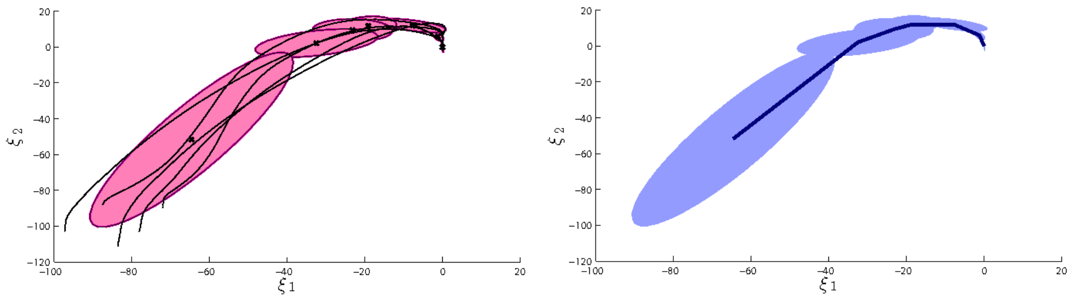
where

$$h_k(\xi) = \frac{p(\xi; \mu^k_{\xi}, \Sigma^k_{\xi})}{\sum_{k=1}^{K} P(\xi; \mu^k_{\xi}, \Sigma^k_{\xi})}, h_k(\xi)\, 0 \tag{4.7}$$

and $\sum_{k=1}^{K} h_k(\xi) = 1$

This process is called Gaussian Mixture Regression. A review of GMR can be found in (Sung, 2004). In this work the demonstrated tasks were learned using an iterative algorithm, Binary Merging (BM), which guarantees to produce stable non-linear dynamics (Khansari-Zadeh & Billard, 2010).

Figure 4.4 illustrates the process of GMM encoding of the demonstrations and GMR reproduction of the learned motions. To learn the model of the trajectories, first several demonstration of the task are presented and them the trajectory is encoded as a mixture of Gaussian distributions. To reproduce the trajectories one sample from the probability distribution of the GMM trough the



(a) (b)

Figure 4.4: *Illustration of the learning process: (a) Learned model of the motion (b) Reproduction of the GMR.*

Gaussian Mixture Regression process. The GMR approximates the dynamical systems through a non-linear weighted sum of local linear models.

4.2.3 Learning the Position and Orientation Components of the Motion

The learning algorithm described in this chapter is a generic framework and makes no assumption on the variable that is used for training. From (Gribovskaya & Billard, 2009), the task space trajectories of the robot's end-effector are chosen to learn control of the position and orientation of the motion. The variables in the training set were chosen as the translation component of a motion of the end-effector (a vector of Cartesian coordinates $x \in R^3$; and the orientation of the end-effector (a pair of variables $\{s, \phi\}$ - the axis and the angle of rotation). According to this representation, the orientation of a moving referential $x\prime y\prime z\prime$ with respect to a fixed referential xyz is described by the rotational axis $s \in R^3$ and the angle $\phi \in [0; 2\pi]$.

Therefore, the functions that the robot must learn from the demonstrations are:

$\dot{x} = \hat{f}_x(x)$, with $\xi = x \in R^3$ for the dynamics of the end-effector's position, and $[\dot{s}, \dot{\phi}] = \hat{f}_o(s, \phi)$, with $\xi = [s, \phi]$ for the dynamics of the end-effector orientation.

4.3 Reproduction of the Skill

For the reproduction of the learned trajectories it could be implemented two types of controllers. A decoupled controller of the position and orientation and a coupled controller of position and the orientation components of the motion.

In the decoupled controller, the position and the orientation components of the motion are learned separately, so we need to learn the following functions:

$$\dot{x} = \hat{f}_x(x), \text{ for the position, with } x \in R^3$$
$$\dot{o} = \hat{f}_o(0), \text{ for the orientation, with } o = [s, \phi] \text{ and } s \in R^3, \phi \in [0; 2\pi]$$

Then we infer the estimate $\hat{f}$ for the dynamics through GMR, as follows:

$$\dot{x} = \hat{f}_x = E[p(\dot{x}|x)], \text{ for the position, and}$$
$$\dot{o} = \hat{f}_o = E[p(\dot{o}|o)], \text{ for the orientation.}$$

In the fully coupled controller position and the orientation components are encoded in a single variable ξ, so we need to learn the following function:

$$\dot{\xi} = \hat{\xi}, \text{ where } \xi = [x, s, \phi] \text{ and } x \in R^3, s \in R^3, \phi \in [0; 2\pi]$$

Then the estimate $\hat{f}$ of the dynamics can be inferred through GMR, as follows:

$$\dot{\xi} = \hat{f}_\xi = E[p(\dot{\xi}|\xi)]$$

The process for on-line reproduction of the learned motion dynamics can be summarize as follows:

1. Learn the estimates, $\hat{\xi}$ of the dynamics underlying the position and orientation of the end-effector's motion.

2. Detect a target position in the global referential $\{xyz\}$.

3. Recompute the current position of the end-effector in the target referential $\{x\prime y\prime z\prime\}$: $\{x0, s0, \phi 0\}$.

4. LOOP from t = 0 until the target position is reached:

5. Infer the velocity at the next time step through GMR, equation 4.6.

6. Solve the Inverse Kinematics problem to find $\dot{\theta}$.

7. Send command $\dot{\theta}_t$ to robot and get motors feedback.

8. Compute the actual position and orientation of the end-effector x_t, s_t, ϕ_t.

9. END

Figure 4.5 and 4.6 summarize the results of encoding and reproducing the two manipulation task presented in Figure 4.2.

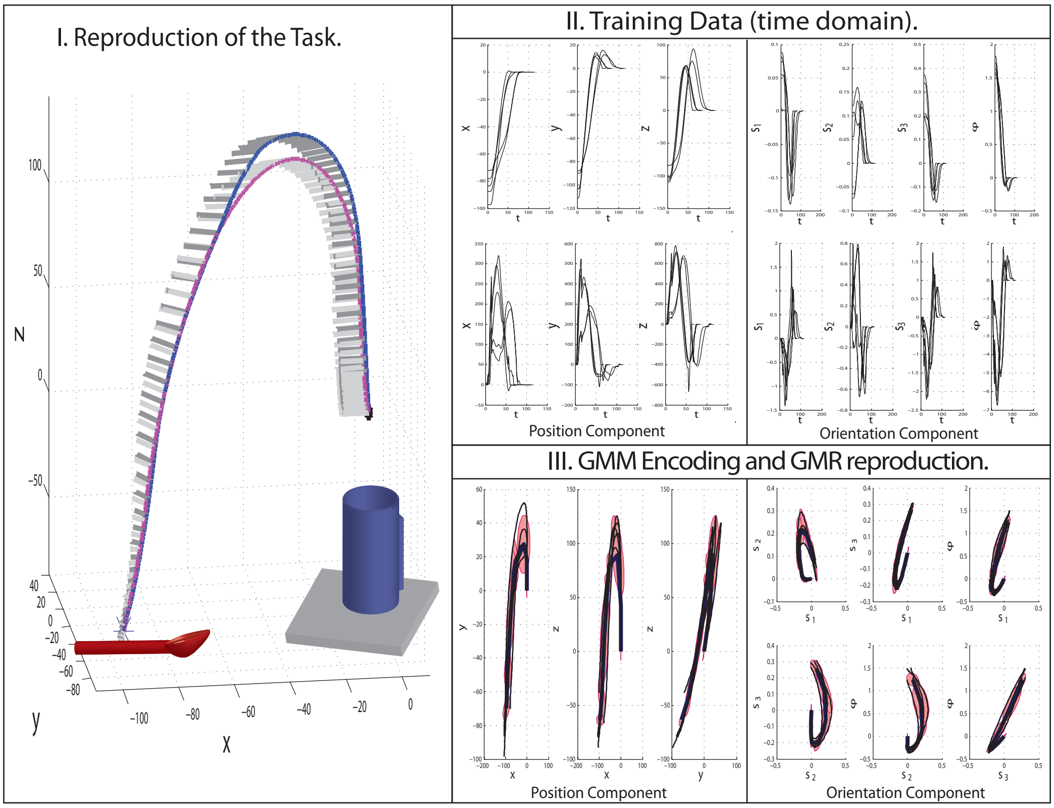


Figure 4.5: *Demonstrations and Reproductions of the spoon in the cup skill. I. Reproduction of the task, coupled controller (blue), decoupled controller (magenta). II. Training data of the demonstrated task. III. Encoded model of the task and reproduction trough GMR.*

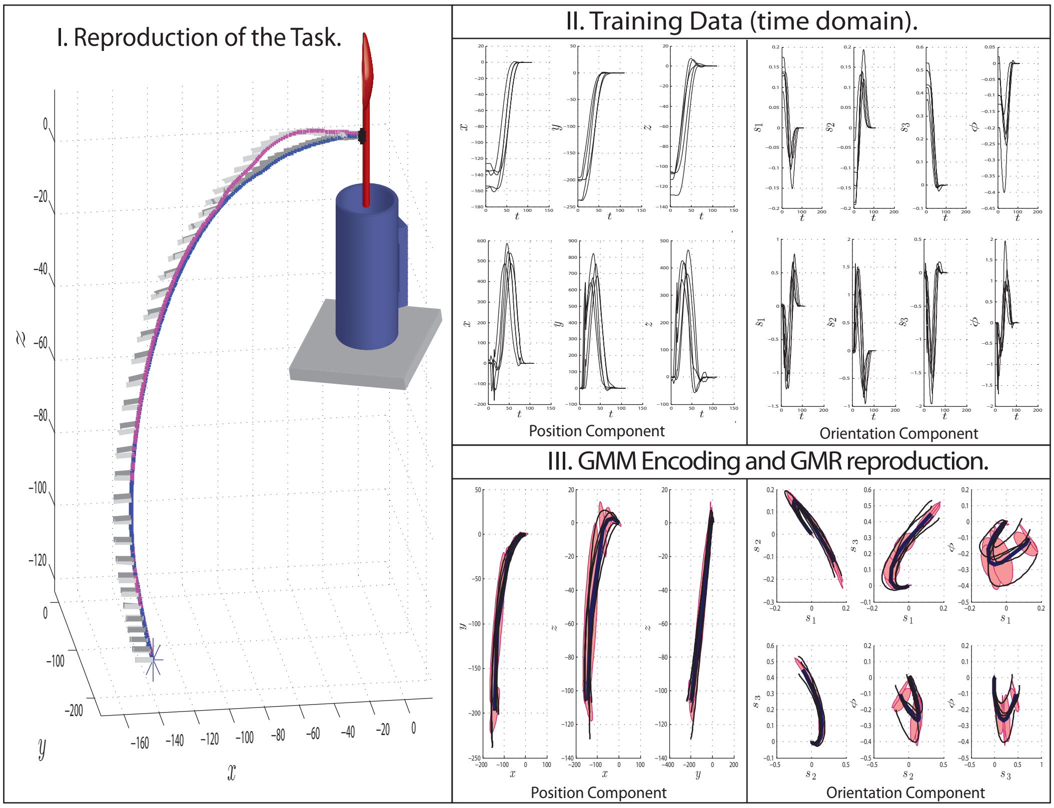


Figure 4.6: *Demonstrations and Reproductions of a grasp skill. I. Reproduction of the task, coupled controller (blue), decoupled controller (magenta). II. Training data of the demonstrated task. III. Encoded model of the task and reproduction trough GMR.*

Chapter 5

Shared Knowledge Database

In Chapter 3 a Remote Collaboration interaction with the HOAP-3 Humanoid Robot is presented. In this collaborative interaction is considered a case of transferring the models of a skill to increase the capabilities of a remote robot by providing it with the ability to learn new task motions.

Chapter 4 explain the learning algorithm and methodology implemented for teaching the robot the motion dynamics of a skill. The learning algorithms module can be divide in two parts: acquisition of skills from human teacher demonstrations and reproduction of those skills from learned models by a robot. The skills acquisition part accounts for gathering data demonstrated by a teacher and for further statistical processing of these data into a model of a skill. A probabilistic encoding of the motion is obtain by means of the Gaussian Mixture Models. The model of the skill is represented by a set of parameters sufficient to reconstruct relevant trajectories of the task.

Here we deal with how to transfer the learned models of the skill to a robot locate at a remote collaboration environment, where a human teacher would be unavailable to teach the robot. For this purpose a Shared Knowledge Database its propose, where the models of the skill task would reside and a remote collaborative robot could access and download the necessary learning. This Chapter presents the Robot-Robot interaction for transferring a skill and the Shared Knowledge Database functionalities and implementation.

5.1 Representation of Skills

All the task contemplated in this work presents a robot performing actions over an object that is found in the remote environment. Therefore a direct link between objects and task skills can be intuitively established.

The Shared Knowledge Database would hold common information of the learned skills models that must be reproduce by the robots. The learn information to be shared would primary concern and object and task skills models.

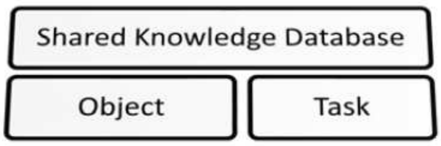


Figure 5.1: *Representation of task knowledge*

1. Objects: necessary information for the recognition and identification of the object, and any constraint relate to it. Tag, Color, Size, Shape, etc.

2. Task: necessary information to reconstruct the model of the skill for the task. Task constraint, generalization of the trajectory, GMM model.

Therefore the elements in the database could be considered, in an analogy to object-oriented programming, as instances of a class object.

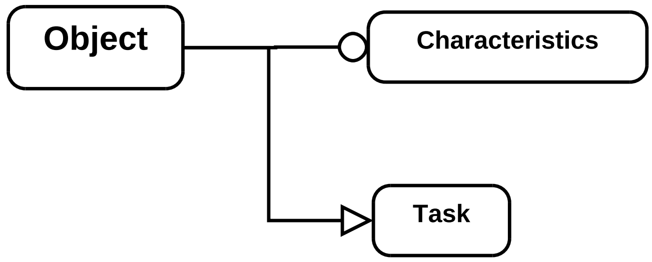


Figure 5.2: *Object instances in the Sharded Knowledge Database: Characteristics attributes and Task operations*

And Object instance in the database would be described by:

- Characteristic attributes, this could be color, shape, AR or RFID tags, size, and any other intrinsic property of the object that allow for its identification.

- Task operations, that act upon the specified object, this would described the learned models of the skill.

In this way the shared knowledge database would be populate by a series of known objects that the robot could identified in its environment. Linked to any instance of an object there could be one, several or no task model operation associated.

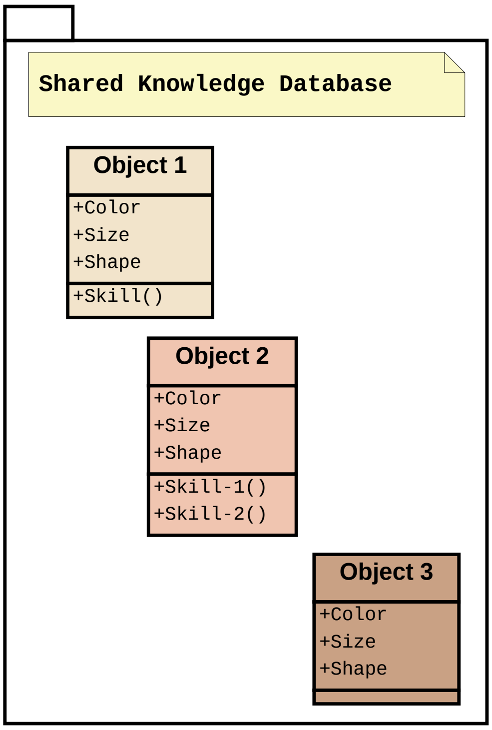


Figure 5.3: *Shared Knowledge Database fill with various object instances.*

To illustrate this, lets consider as example the case of a robot working on a kitchen. During its workday the robot would have to perform various task such as helping setting up the table, serving food plates, and cleaning the dishes. Therefore one would have for a single object *i.e.* a glass, or a plate, different action task to perform depending on the situation, such as, robot *clean plate* or robot *serve plate*. Mean while for an object, such as, a table, there would be no task, since there is no manipulation of the table object required.

Furthermore, to expand the functionalities of the Shared Knowledge Database a "behavioural" instance could be implemented. A behaviour, therefore, would consist of a list of task actions, with its associated object, that need to be executed to achieve a goal. Using this behavioural mode would increase the functionality of the shared knowledge database, this remains as future work that need to be implemented.

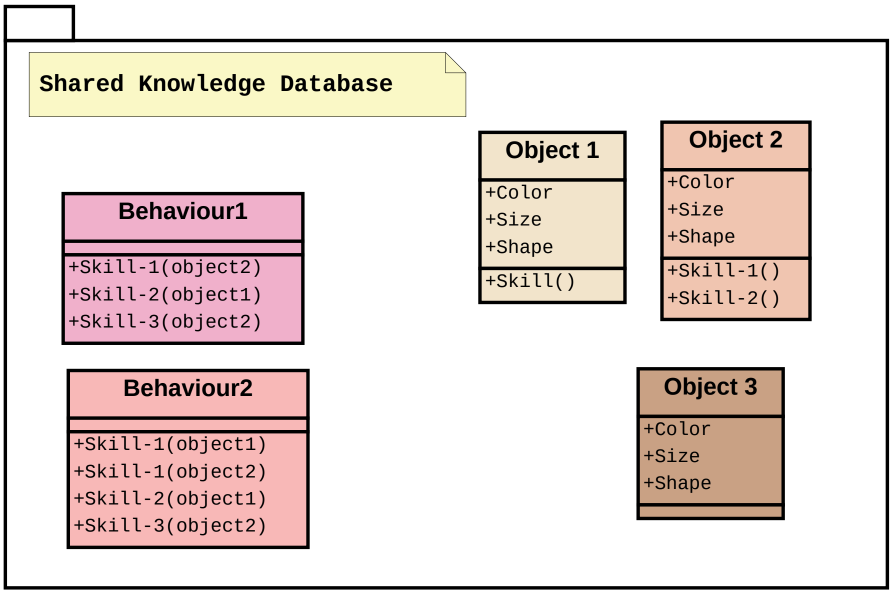


Figure 5.4: *Shared Knowledge Database fill with various object instances and behaviours.*

As example, lets reconsider the kitchen setting, a robot behaviour in this case could be to set-up the table for lunch. This behaviour would require the robot to reproduce several tasks, namely placing a plate, glass, and silverware. The setting the table behaviour would consist them of an sequence of tasks skills the robot must completed for completion of the behaviour task.

5.2 Robot-Robot Interaction Transfer of a Skill Knowledge

The shared knowledge database holds commonly accessible data for that the robots could learn and reproduce a skill. The robots can upload or download the learning of the skills when is available. They will also send or receive acknowledgement signals when new data of the skill is uploaded to the share database. The Robots communications through the shared knowledge database will use a TCP/IP private network.

In the scenario presented on Chapter 3 a human operator would collaborate with a robot partner performing several task. During the collaboration the human operator can request the robot to `USE OBJECT(<object_id>)`, as established in the robot command protocol. To respond to this command the robot would look up the shared knowledge database for task associated to object `<object_id>`. If no task have been associated with the object a request is made from the robot to learn a new skill.

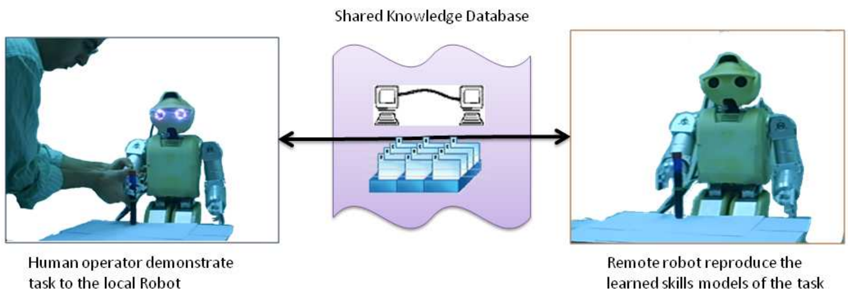


Figure 5.5: *Robot-Robot Interaction Transfer of a Skill Knowledge*

A human operator would demonstrate, by kinaesthetic teaching, the motion trajectories of the task to a robot agent. Using the algorithms and techniques presented in Chapter 4 the robot would learn a generalized model of the motions. This skill model would be associate to the actuated object and the local robot would upload a new instance to the database for the learning of the new skill. Them this model of the task would be available for the remote robot to download. Trough the shared knowledge database two robot agents could transfer the models of a skill.

5.3 Refinement of the Skills Models

In the framework presented in this work the reproduction of the task is executed by a remote robot. While the teaching of the skill is perform by an operator demonstrating the task on a local robot. As the teaching environment and the reproduction environment could potentially be different, this presents certain issues of adaptation and correspondence.

To tackled this issues in (Billard et al., 2008) suggest to combining PbD methods with other motor learning techniques, to allow a robot to learn how to perform a task in new situations. Reinforcement learning (RL), appeared particularly indicated for this type of problem.

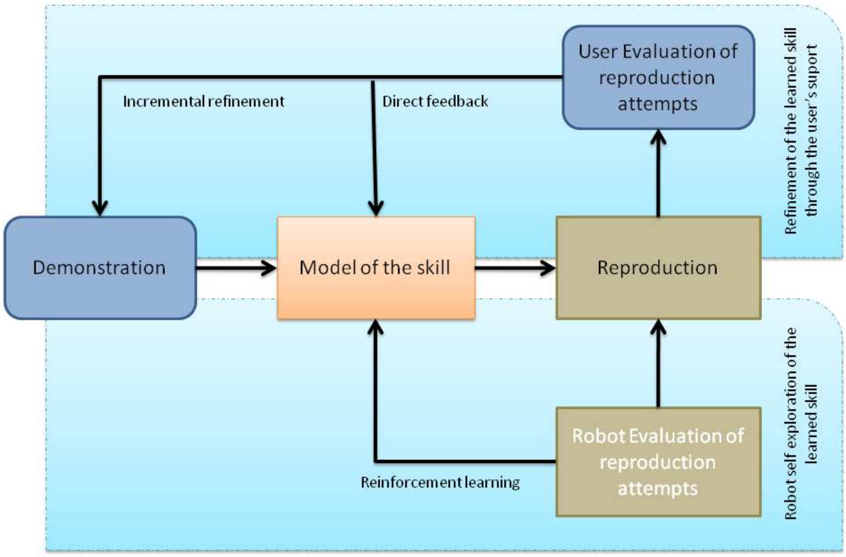


Figure 5.6: *Interactive refinement*

Refinement of the model can be done at two stages:

- By the human operator, when teaching the task to the local robot.
- Or by self exploration, when executing the task by the remote robot.

After refinement the model of the task needs to be update an reloaded on the shared knowledge database. The process of the remote robot evaluating the performance and refining the learned models of the task has not been implemented. This remain as a future work for this thesis.

Chapter 6

Experimental Results

To test the proposed systems and algorithms various experimental demonstrations were conducted with the humanoid robot HOAP-3, described in Chapter 2.

Chapter 3 presented the interaction modalities for a Remote Collaboration environment with the HOAP-3 Robot. For controlling and monitoring robot activities on a remote site, the propose system require a robot agent deployed at a remote setting and a human operator that communicates to it by means of a human-robot interface, the HRI functionalities has been detailed on Chapter 3. To increase the robot flexibility for performing a task it must have the capacity to learn new, previously untaught, skills. To teach a robot how to reproduce new skills remotely, learning algorithms, explained on Chapter 4, and a shared knowledge database, presented on Chapter 5 was needed.

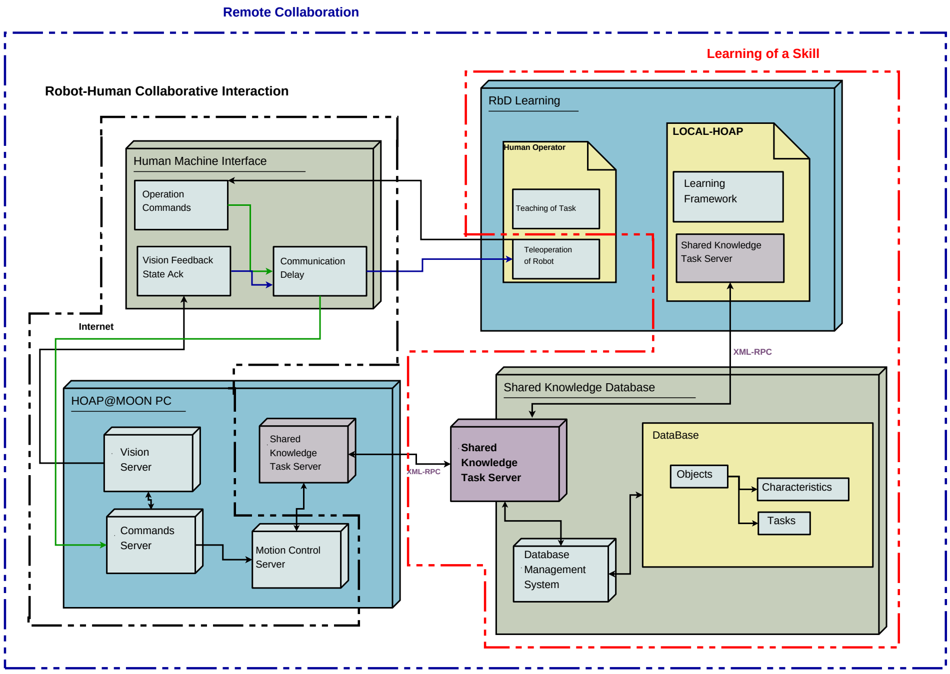


Figure 6.1: *Deployment Structure of the Experimental Set-Up*

Figure 6.1 presents the structure for the Remote Collaboration experimental set-up. To illustrate the system, experiments were first perform separately on the human-robot collaborative interaction, and the learning of a skill modalities. Later the experiment of the completed Remote Collaboration and Learning of Skills scenario is presented. The following section would describe the overall set-up for the demonstration of the remote collaborative environment and learning of skills. Later the evaluation of the experiments is presented.

6.1 Experimental Set-up

At the Robotics Lab on the Universidad Carlos III de Madrid a lunar scenario was built to simulate the operation of a robotic agent working in collaboration with a human in a space environment. It consists on a long corridor surrounded

Figure 6.2: *The Space Moon Scenario.*

by cliffs where the robot can walk and interact with the environment. The surface of the cliffs has been built with planes of polystyrene where it has been made holes to simulate craters. The floor of the scenario has been made of hard cardboard as the robot has to walk on it. To paint it, we have used a uniform grey to avoid interferences in the vision of the robot.

The overall story is: a *Robot at a remote space environment*, "the moon", will perform autonomously a manipulation task. The robot would not know how to perform this task at first and it will learn the task with the assistance of a Human-Robot team at "earth". An expert operator at a local site will teach a Robot the task and then this *local* robot would communicate the learned task to the *remote* robot.

There are three agents interacting in this scenario:

- The humanoid Robot HOAP-3 at the Moon (located at the moon scenario build at our RoboticsLab),
- The human operator, located at a "Earth center of operation", has two task:
  - By means of an HRI interface the operator will send instructions and monitor the state of the remote HOAP robot.
  - By means of the learning techniques the operator will teach a local robot how to perform the task.

- The local humanoid Robot HOAP-3 at the "Earth center of operation" will learn the task and transmit this knowledge to the remote robot HOAP at "moon" trough a Shared Knowledge Database.

The experimental scenario can be considered as one major task, human collaboration and supervision of the robot at remote environment, that latter contains three sub-parts:

- Robot autonomous operation.
- Teaching of the task skills.
- Robot reproduction of learned skills.

There are 4 main components in the system, that have been described in the previous chapters:

1. HOAP-3 Software Server: The Robot control software system.
2. The Human-Robot Interface (HRI): for the teleoperation of HOAP robot.
3. Learning Algorithms: The Robot learning techniques and algorithms system for teaching task to robot HOAP.
4. The Shared Knowledge Database: The Knowledge interface for transferring the learning of the tasks between the robots.

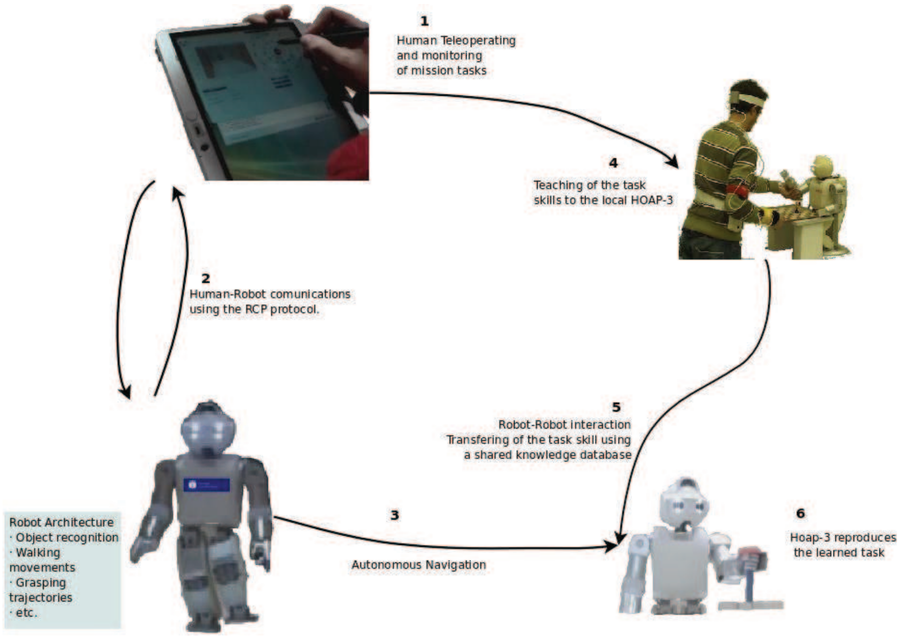


Figure 6.3: *Overview of the experimental demonstrator for the remote collaboration and skill learning environment.*

The HRI and the HOAP-3 at remote environment are the sole components involved in all segments of the demonstration. The HOAP-Server is in charge of realizing the second segment of the demonstration (Robot autonomous operation) after receiving request from the operator to perform. The learning algorithms module is mostly in charge of the fourth segment of the demonstration (Teaching of the task skills). HOAP-Server and the learning module are the main components in the fourth segment of the demonstration (Robot reproduction of learned skills). In the third and fourth segments, The Shared Knowledge Database is used to communicate and transfer the learning and task information between the HOAP robots. The overview of the experimental demonstrator can be seen on Figure 6.3.

6.2 Evaluation of the Experiments

6.2.1 Robot-Human Collaborative Interaction: Teleoperating a robot for executing a task

In figure 6.4 shows the global sequence of the teleoperation task scenario in the robot-human collaborative interaction.

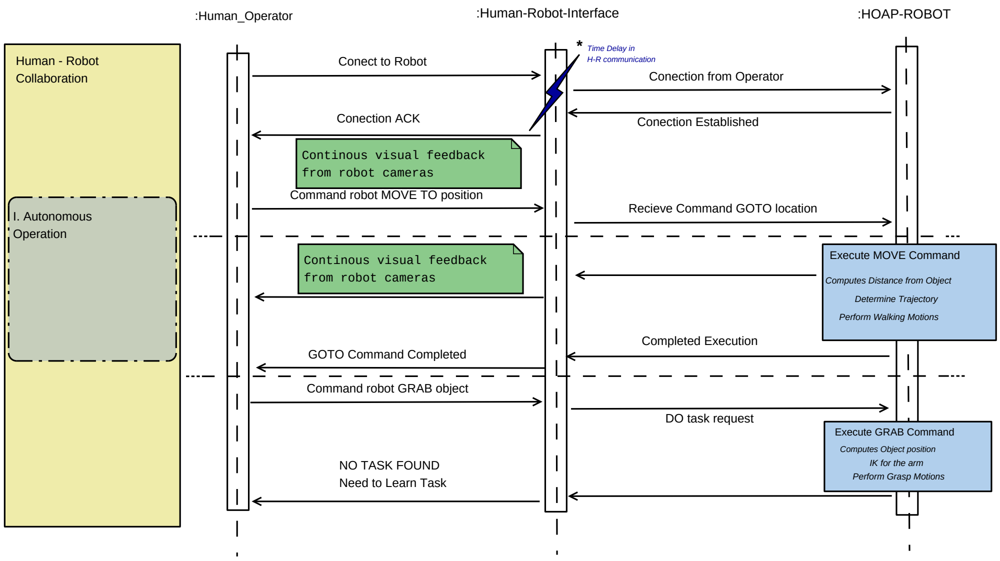


Figure 6.4: *Sequence diagram for the teleoperation scenario*

A first teleoperation experiment with the HOAP-3 robot and the HRI was conducted in the Robot@CWE project for the second year demonstrator. The experiment consisted on working collaboratively with a robot to perform a task by the teleoperation of the HOAP-3 robot. The human agent works collaboratively with the humanoid robot by supervising, controlling and helping in the decisions taken by the robot.

The task to be performed consists of teleoperating the HOAP-3 robot, inside the moon scenario, first walking through an enclosed hall and finding an object, in this case an "antenna", then grasping the object and placing it in a different location (Pierro et al., 2009).

(a) (b)

Figure 6.6: *Teleoperation of the robot: (a) The robot autonomously walks trough the scenario. (b) Robot grasp the "antenna', as requested by the operator.*

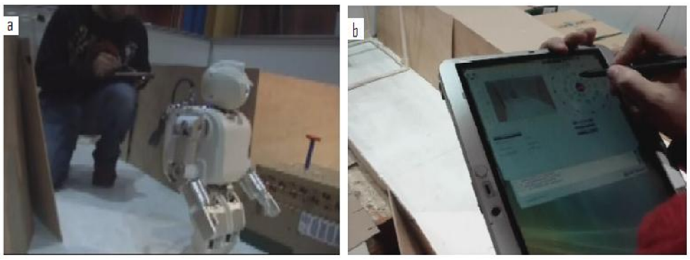


Figure 6.5: *Teleoperation of the robot.*

This scenario allows the human operator to interact with the robot through teleoperation by using the Human-Robot interface. First the operator must connect to the robot, and a connection negotiation according to predefined users and protocols ensue. Once the robot has granted connection to the robot, the operator would be able to see the remote scenario through the eyes of the robot, receiving continuous vision feedback from the HOAP-3 robot cameras. Figure 6.5 shows the human operator using the HRI and the HOAP-3 robot at the remote environment.

Whit the HRI the operator can send command request for the HOAP-3 robot that performs them autonomously. The operator can send walking and turning movements, grasping motions, or higher order commands. In the scenario, the human-robot team are requested to look for an "antenna" and restore communications. A efficient way to do this is for the operator to request the robot to walk around the environment until it sees an object, since the human vision capacities to recognize objects are greater than that can be implemented on a robot platform, it would be on the operator responsible to look on the video feedback and tell the robot when it has been found the "antenna". Once locate the target object the robot can autonomously walk towards the object to a close enough distance so that it can grab it when requested by the human operator. Figure 6.6 shows the robot HOAP-3 walking on the remote environment, and grasping an object when requested.

6.2.2 Learning of a Skill: Teaching a robot manipulation tasks

To test the learning algorithms presented on Chapter 4 a robot HOAP-3 was taught to perform several manipulation skills.

The robot was taught to perform two manipulation tasks. In all cases the training data were provided by a human operator guiding the robot arm through the tasks, kinaesthetic teaching. Between four to six demonstrations where provided for each task. The first task was chosen so as to require specific coordination between the position and orientation for successful task's accomplishment. For the other task the coordination between position and orientation component was not of principal importance.

Figure 6.7: *The HOAP-3 robot is teach to reproduce a "place spoon in a cup" task*

Figure 6.8: *The HOAP-3 robot is teach to reproduce a "reach and grasp spoon" task*

The first task (place spoon in a cup); see Figure 6.7, consists in putting an object inside a cup. To accomplish this task, the robot should adapt the orientation of the end-effector as it approaches to the cup so as not to hit it. The robot should also simultaneously converge to the demonstrated final position and orientation so to put the object in the container. The second task (reach and grasp the spoon); see Figure 6.8, consists in reaching and grasping an object. To accomplish this task, the robot should reach an object (the spoon), while adapting the orientation of the end-effector. The position and orientation components are not expected to be highly correlated.

During the demonstrations the kinematics data from the motors encoders of each joint are recorded at a rate of $1000Hz$. This is then re-sampled to a fixed number of points and the data is transform from the joint space to the task space for training the model of the motion. With the GMM and GMR algorithms the models of the position and orientation dynamics are learned. An iterative algorithm, Binary Merging (BM), which guarantees to produce stable non-linear dynamics (Khansari-Zadeh & Billard, 2010), was used to learned the models. Finally the robot is provide with the models of the motion dynamics for its reproduction of the task.

Reproductions experiments on both tasks were conducted under various spatial and temporal perturbations of the position and orientation component; and implementing the controllers presented in Chapter 4, decoupled and fully coupled controllers of the position and the orientation. The results are summarize in the Figures 6.9, 6.10, 6.11, 6.12.

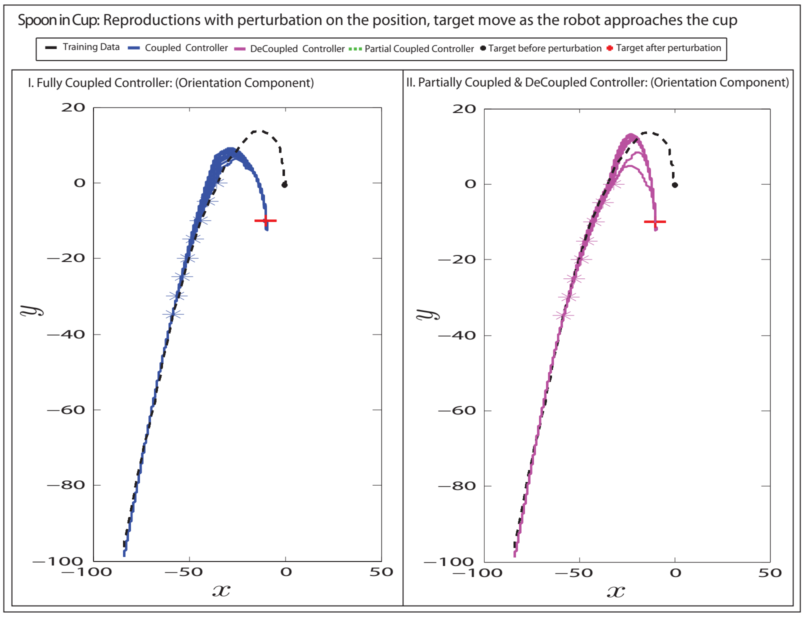


Figure 6.9: *Reproductions of task "place spoon in cup" with a perturbation on the target position*

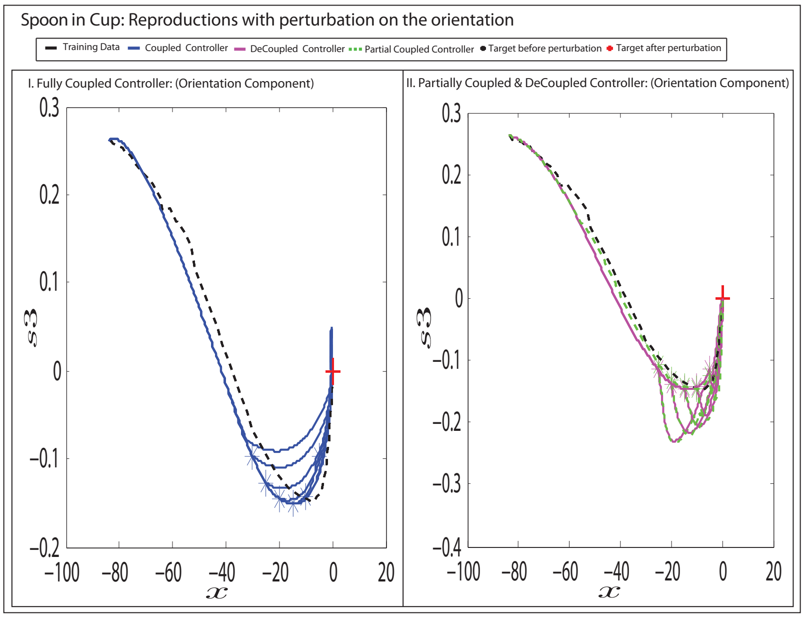


Figure 6.10: *Reproductions of task "reach and grasp spoon" with a perturbation on the target orientation*

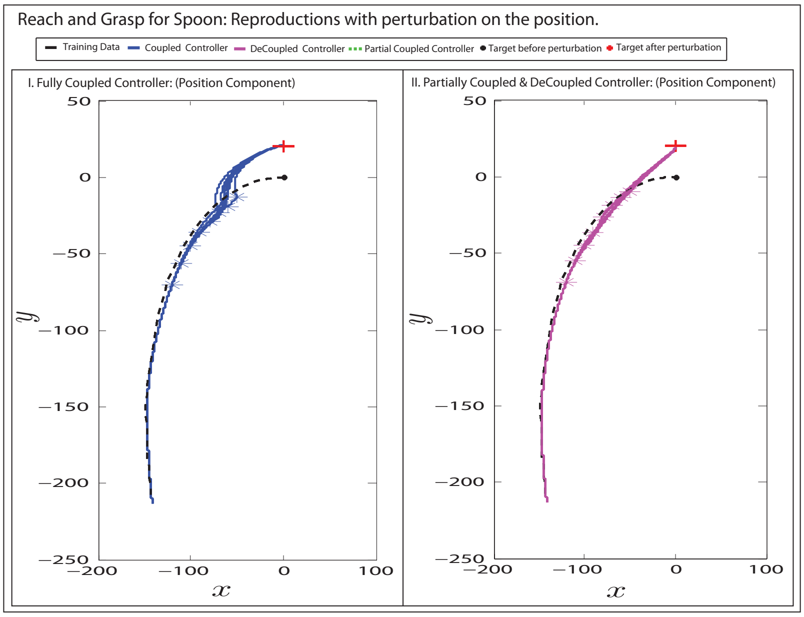


Figure 6.11: *Reproductions of task "reach and grasp spoon" with a perturbation on the target position*

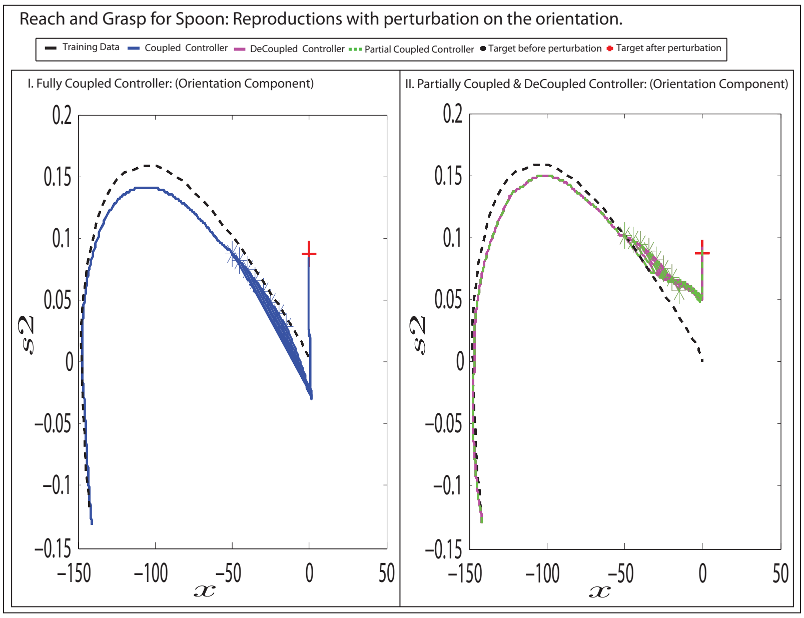


Figure 6.12: *Reproductions of task "reach and grasp spoon" with a perturbation on the target orientation*

6.2.3 Remote Collaboration: Human-robot collaborative working at a space scenario

Figure 6.13 shows the global sequence of the remote collaborative working task scenario.

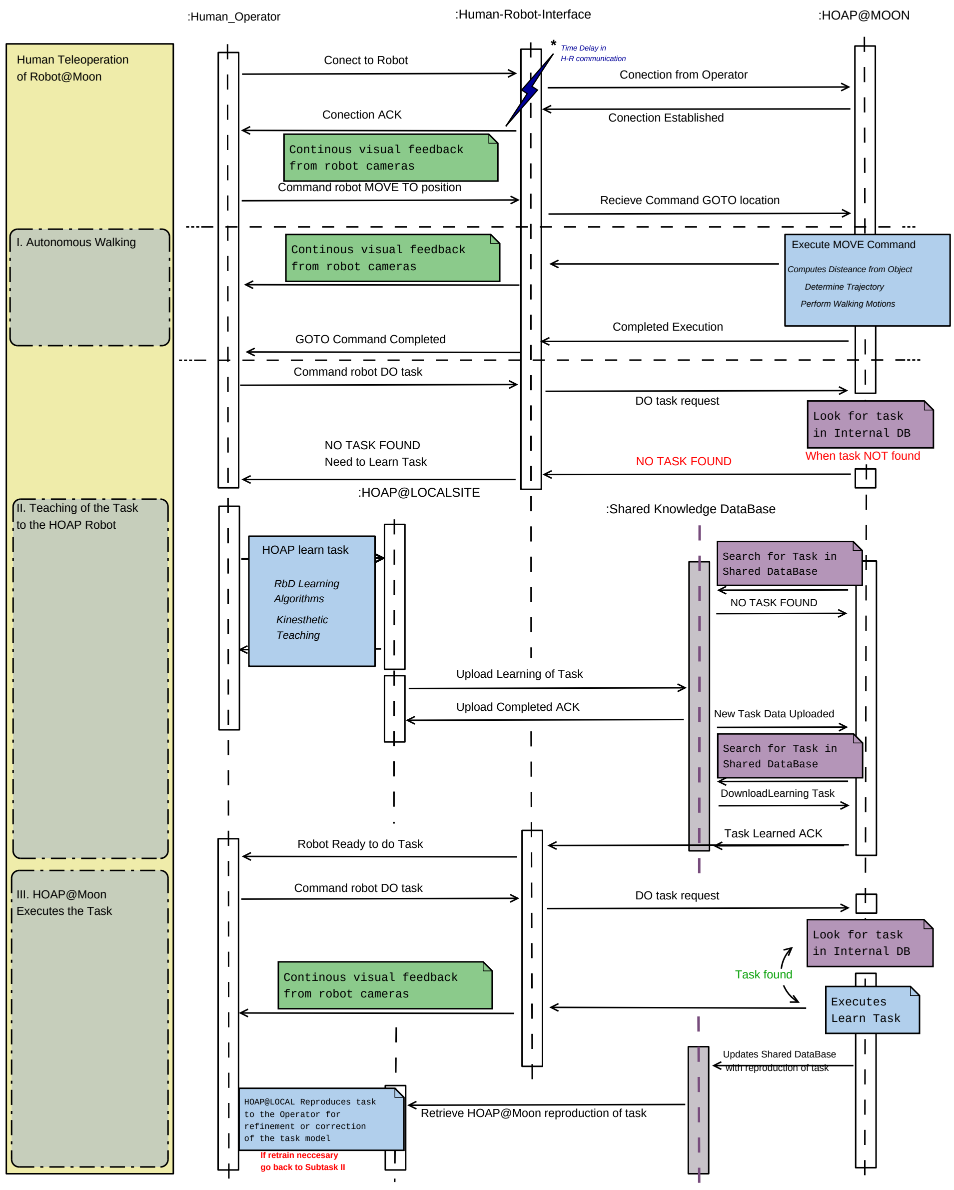


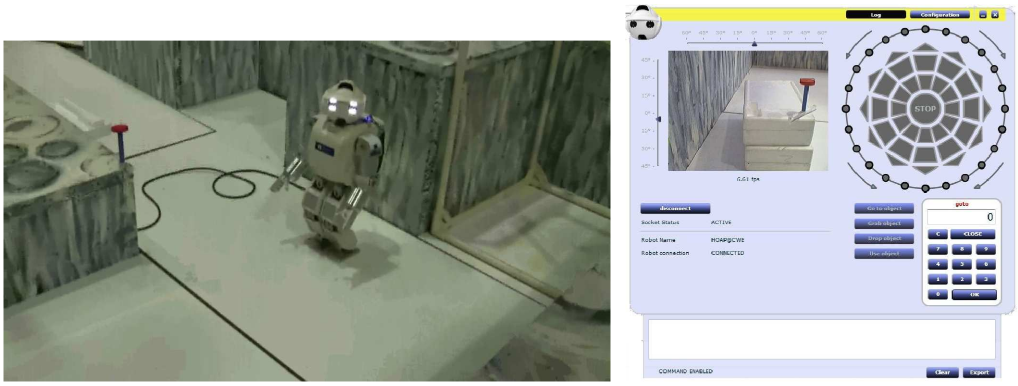
Figure 6.13: *Remote Collaboration Scenario Global Sequence Diagram*

A final experimental demonstrator combining the subsystems of the two previous sections is implemented for a remote collaboration and learning of skills, following the experimental set-up described at the beginning of this Chapter. The plan for the demonstrator involves three agents, a human operator, and two humanoid robots HOAP-3. One HOAP-3 robot will be at the remote location, which in this case represent robot working at a moon space scenario. While the human operator and the other HOAP-3 robot would be at the same work space.

The human operator has two tasks, using an HRI interface the operator will send instructions and monitor the state of the remote HOAP-3 robot. The Humanoid Robot and the Human Operator will work collaboratively at the remote environment achieving the task goals. The Operator will teach a robot the necessary skills needed to complete this task. With the teaching and learning techniques, previously presented, the operator will teach the local HOAP-3 robot, sharing the same workspace as the operator, how to perform the task. Once the local robot learns the task it will transmit the knowledge of the task to the remote HOAP-3 robot at the moon scenario.

The demonstrator would follow the general plan outline in Figure 6.13:

- Human Teleoperation of Robot at moon scenario.
  - The human operator connects to remote HOAP-3 through the HRI after a connection negotiation protocol.
  - The operator request the robot to go to a location where it should be to perform the task.
  - The remote robot moves autonomously to the location where the task must be performed (Start subtask I).
  - The human operator and the remote robot work collaboratively to achieve the task. If a new or unknown request arise the remote robot ask the operator for the teaching of a skill (Start subtask II).



(a) (b)

Figure 6.14: *Remote Collaboration: subtask I. Robot autonomous operation. (a) The robot autonomously walks to the object. (b) The human operator request a* `GOTO OBJECT` *command.*

- The robot reproduce the learning skills to complete the task, with operator supervision (Start subtask III).

The sub-tasks I, II, and III are described hereafter:

I. Subtask I. Robot autonomous operation. The robot starts the first part of the plan by going to the location specified by the human operator trough a HRI command:

- The robot calculates the distance to the location. Detects object with vision and computes the distance from its position.
- It determines the trajectory it must follow.
- It performs the necessary movement to reach the objective.
- Sends acknowledge to Operator that movement command have been execute in HRI.

II. Subtask II. Teaching of the task skills to the HOAP-3 robot. With the learning techniques, human operator teaches local robot the task skills:

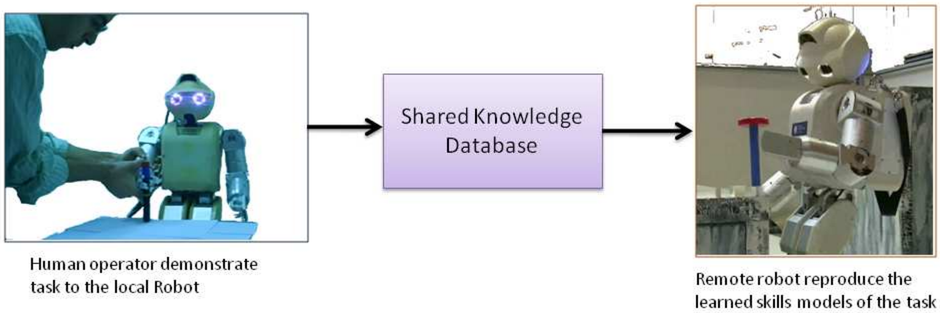
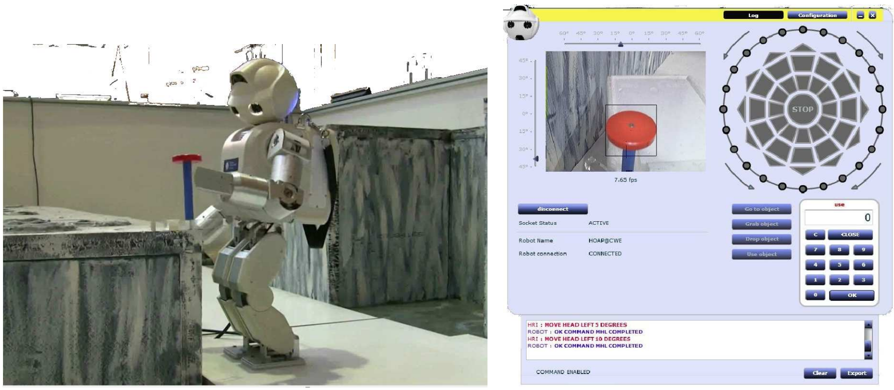


Figure 6.15: *Remote Collaboration: subtask II. Teaching of the task skills to the HOAP-3 robot. The human operator teaches a local robot the skills, later both robots interact trough the shared database to transfer the models of the skill.*

- The Operator will teach the local robot to perform the task require by the remote robot to complete the task goals. Using the learning by imitation techniques presented in Chapter 4.
- Once the task is learned the local robot uploads the skill knowledge to the shared knowledge database. Send acknowledgement that the task have been uploaded.
- The robot at the remote collaborative scenario downloads the skill model. Send an acknowledgement to Operator that task is ready to be performed.

III. Subtask III. Robot reproduction of learned skills (human operator supervision). The robot reproduce the learned models of the skill. The operator check performance of HOAP-3 robot, see if correction in the demonstrations are required:

- The remote robot executes the learned task, after confirmation from the human operator that its right to proceed, the human operator request a `USE OBJECT` command, this will trigger a selection of strategy dialog with the robot for selecting the type of task to reproduce with the object.



(a) (b)

Figure 6.16: *Remote Collaboration: subtask III. Robot reproduction of learned skills. (a) The robot reproduce the learned models of the skill. (b) The human operator request a* `USE OBJECT` *command, this will trigger a selection of strategy dialog with the robot for selecting the type of task to reproduce with the object.*

- Operator monitors execution with vision and state of the robot feedback trough the HRI.
- If the Operator is not satisfied with the reproduction it will repeat the teaching subtask with the local robot. If satisfied the demonstrator task are completed.

The experiment described here was perform as part of a final demonstrator to the Robot@CWE European project and the CARHU project funded CICYT.

Chapter 7

Conclusions

This work has focus on aspects of human-robot interaction and human-robot working in collaborative environments. In particular the collaboration between human and humanoid robots performing tasks in a remote collaborative working environment is studied.

In this work three forms of interaction where presented, a human-robot remote collaboration interaction where a human operator and a robot at a remote working environment interact through a HRI in achieving collaboratively a global goal. A close human-robot interaction where a human teacher present a robot several demonstration of a task for it to learn. And a robot-robot interaction for transferring the learned skills models of a task between a local robot, that is taught by a human operator, and a remote robot performing task autonomously in a remote collaboration environment.

A robot-human-robot collaboration architecture was developed for a human operator and a local robot to interact with a robot located at a remote location to:

- Teleoperate and supervise remote robot performance.
- Collaborate between a robot-human team in execution of tasks.
- Allow a human operator to teach the performance of a task.

- Share skills knowledge between robots.

By endowing a humanoid robot partner with sufficient navigation autonomy and situation awareness of its environments, visual and sensory perception. And providing a human-operator with a functional HRI and a versatile communication protocol. A human-robot team could successfully engage in a remote collaborative working environment for the accomplishment of various task.

Also we have implemented learning by imitation techniques, to learned models of the task motion dynamics with a local robot. The robot is presented with various demonstrations of the task motion, a model of the motion is estimate through a set of first order non-linear multivariate dynamical systems, to learned the motion dynamics the task are learned in a probabilistic approach as Gaussian Mixture Models. To reproduce the trajectories one sample from the GMM model through a Gaussian Mixture Regression. The GMR approximates the dynamical systems through a non-linear weighted sum of local linear models. To test the learning algorithms a robot HOAP-3 was taught to perform several manipulation skills. Controllers for decoupled and fully coupled controllers of the position and the orientation were implemented. Reproductions of the tasks were conducted under various spatial and temporal perturbations of the position and orientation component.

To allow robot-robot interaction and the transfer of the skills to a remote robot a Shared Knowledge Database was developed, where the models of the skill task would reside and the remote collaborative robot could access and download the necessary learning. The Shared Knowledge Database hold common information of the learned skills models, and the objects associated to the task, that must be reproduce by the robots. To expand the functionalities of the Shared Knowledge Database a "behavioural" instance could be implemented. A behaviour, therefore, would consist of a list of task actions, with its associated object, that need to be executed to achieve a goal. Also the process of the remote robot evaluating the performance and refining the learned models of the task has not been implemented. Both of this remain as a future work for this thesis.

The system was tested with a remote collaboration scenario of a robot working in space. A first demonstration consisted working collaboratively with a robot to perform a task by the teleoperation of the HOAP-3 robot. The human agent works collaboratively with the humanoid robot by supervising, controlling and helping in the decisions taken by the robot. The human teleoperated the robot looking for and object in its environment. After the object is locate the robot approach to it autonomously and grasp it.

A final demonstrator test was conducted, it illustrates the two examples of human-robot interaction presented in this work. One of remote control and collaboration. And other of cognitive interaction in which a human expert teaches a robot to accomplish parts of a global unknown task with the use of programming by demonstration and learning algorithms. The plan for the demonstrator involves three agents, a human operator, and two humanoid robots HOAP-3. One HOAP-3 robot will be at the remote moon space scenario. While the human operator and the other HOAP-3 robot would be at the same work space.

In the demonstrator the human operator connects to remote HOAP-3 through the HRI. The remote robot moves and performs autonomously according to the requested instructions by the operator with the HRI. When new or unknown request arise the remote robot ask the operator for the teaching of the skill. The human teaches the local robot, and a robot-robot interaction ensue to transfer the learned models of the task. The robot reproduce the learned skills to complete the task, with operator supervision.

The experiment results presented on this work has been implemented as part of a final demonstrator to the Robot@CWE European project and the CARHU project funded CICYT.

References

Acroboter. (2010). `http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=80213`.

Aist presentation at dsig plenary meeting. (2005). `http://robotics.omg.org/docs/robotics2005-04-Athens.pdf`.

Ambrose, R., Aldridge, H., Askew, R., Burridge, R., Bluethmann, W., Diftler, M., et al. (2000, Jul/Aug). Robonaut: Nasa's space humanoid. *Intelligent Systems and their Applications, IEEE*, *15*(4), 57-63.

Arbulu, M., Balaguer, C., Bauer, A., Evrard, P., Kheddar, A., Perez, C., et al. (2007). *Deliverable 1.1-2: Ist-robot concepts and preliminary integration* (Tech. Rep.). ROBOT@CWE. Advanced robotic systems in future collaborative working environments.

Arbulú, M., Kaynov, D., Cabas, L. M., & Balaguer, C. (2009). The rh-1 full-size humanoid robot: design, walking pattern generation and control. *Journal of Applied Bionics and Biomechanics.*, *6*(3), 301–344.

Asimov, I. (2008). *I, robot*. Random House Publishing Group.

Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Robot programming by demonstration. In B. Siciliano & O. Khatib (Eds.), *Handbook of robotics* (pp. 1371–1394). Secaucus, NJ, USA: Springer.

Blasi, L., & Stasse, O. (2008). *Deliverable 3.3: Design support software for robot@cwe* (Tech. Rep.). ROBOT@CWE. Advanced robotic systems in future collaborative working environments.

Blasi, L., Weiss, A., Stasse, O., & Hernández, D. (2009). *Deliverable 3.8: Final report on software concepts for robot@cwe* (Tech. Rep.). ROBOT@CWE. Advanced robotic systems in future collaborative working environments.

Calinon, S. (2009). *Robot programming by demonstration: A probabilistic approach*. EPFL/CRC Press.

Calinon, S., Guenter, F., & Billard, A. (2007). On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, *37*(2), 286–298.

Capek, K. (2004). *R.u.r. (rossum's universal robots)*. Penguin Group.

Chen, J., Haas, E., & Barnes, M. (2007, nov.). Human performance issues and user interface design for teleoperated robots. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *37*(6), 1231 -1245.

Chiaverini, S., & Siciliano, B. (1999). The unit quaternion: a useful tool for inverse kinematics of robot manipulators. *Syst. Anal. Model. Simul.*, *35*(1), 45–60.

Dasgupta, S., & Schulman, L. J. (2000). A two-round variant of em for gaussian mixtures. In *Uai '00: Proceedings of the 16th conference on uncertainty in artificial intelligence* (pp. 152–159). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Ferre, M., Buss, M., Aracil, R., Melchiorri, C., & Balaguer, C. (2007). *Advances in telerobotics*. Springer Publishing Company, Incorporated.

Fong, T., Thorpe, C., & Baur, C. (2002). Collaboration, dialogue, and human-robot interaction. In *In 10th international symposium of robotics research.* Springer-Verlag.

Gorostiza, J., Barber, R., Khamis, A., Pacheco, M., Rivas, R., Corrales, A., et al. (2006, sep.). Multimodal human-robot interaction framework for a personal robot. In (p. 39 -44).

Gribovskaya, E., & Billard, A. (2009). Learning Nonlinear Multi-Variate Motion Dynamics for Real- Time Position and Orientation Control of Robotic Manipulators. In *Proceedings of 9th IEEE-RAS International Conference on Humanoid Robots.*

Gribovskaya, E., Zadeh, K., Mohammad, S., & Billard, A. (2010). Learning Nonlinear Multivariate Dynamics of Motion in Robotic Manipulators [accepted]. *International Journal of Robotics Research.*

Hernández, D., Fierro, M. G., Weiss, A., Wurhofer, D., Gribovskaya, E., & Peer, A. (2009). *Deliverable 2.8-9: Final report on ist-robotic architectures and models in cwe* (Tech. Rep.). ROBOT@CWE. Advanced robotic systems in future collaborative working environments.

Herrero-Perez, D., Pierro, P., Hernandez Garcia, D., Gonzalez-Fierro, M., Blasi, L., & Balaguer, C. (n.d.). *A dh-dr collaborative architecture for a humanoid robot in an unstructured scenario.*

Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., & Hirukawa, H. (2001). The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Intelligent robots and systems, 2001. proceedings. 2001 ieee/rsj international conference on* (Vol. 1, p. 239-246 vol.1).

Kaneko, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., et al. (2004). Humanoid robot hrp-2. In (pp. 1083–1090). Proceedings of IEEE International Conference on Robotics and Automation.

Khansari-Zadeh, S. M., & Billard, A. (2010). BM: An Iterative Method to Learn Stable Non-Linear Dynamical Systems with Gaussian Mixture Models. In *In Proceeding of the International Conference on Robotics and Automation (ICRA 2010)* (pp. 2381–2388). Available from `http://icra2010.grasp.upenn.edu/`

Lasa: Laboratoire d'algorithmes et systemes d'apprentissage. (2010). `http://lasa.epfl.ch/`.

Mclachlan, G., & Peel, D. (2000). *Finite mixture models*. Wiley-Interscience. Hardcover. Available from `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0471006262`

Monje, C. A., Pierro, P., & Balaguer, C. (2008). Humanoid robot rh-1 for collaborative tasks: a control architecture for human-robot cooperation. *Applied Bionics and Biomechanics*, *5*(4), 225-234.

Ogura, Y., Aikawa, H., Kondo, H., Morishima, A., Lim, H. ok, & Takanishi, A. (2006). Development of a new humanoid robot wabian-2. In (pp. 76–81). International Conference on Proceedings of IEEE Robotics and Automation.

Phriends. (2010). `http://www.phriends.eu/index.html`.

Pierro, P. (2009). *Human-humanoid collaborative models*. Unpublished master's thesis, UNIVERSIDAD CARLOS III DE MADRID. DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA.

Pierro, P., Hernandez, D., Gonzalez-Fierro, M., Blasi, L., Milani, A., & Balaguer, C. (2009, jun.). Humanoid teleoperation system for space environments. In (p. 1 -6).

Pierro, P., Monje, C., & Balaguer, C. (2008, dec.). Modelling and control of the humanoid robot rh-1 for collaborative tasks. In (p. 125 -131).

Robot@cwe. (2010). `http://www.robot-at-cwe.eu/`.

Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., & Fujimura, K. (2002). The intelligent asimo: System overview and integration. In (pp. 2478–2483). IEEE/RSJ international conference on intelligent robots and systems.

Siciliano, B., & Khatib, O. (Eds.). (2008). *Handbook of robotics*. Springer.

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: Modelling, planning and control* (Second ed.). Springer.

Smerobot. (2010). `http://www.smerobot.org/`.

Stasse, O., Pierro, P., Perez, C., Balaguer, C., Evrard, P., Kheddar, A., et al. (2008). *Deliverable 2.2-3: Architectures and models of ist-robotics: Different integration models of collaborative work in different real environments* (Tech. Rep.). ROBOT@CWE. Advanced robotic systems in future collaborative working environments.

Sung, H. G. (2004). *Gaussian mixture regression and classification*. Unpublished doctoral dissertation, Rice University.

Vlassis, N., & Likas, A. (2002). A greedy em algorithm for gaussian mixture learning. *Neural Processing Letters*, *15*, 77-87. Available from `http://dx.doi.org/10.1023/A:1013844811137` (10.1023/A:1013844811137)

Yokoi, K., Kawauchi, Sawasaki, Nakajima, Nakamura, Sawada, et al. (2004). Humanoid robot applications in hrp. *Int. J. Rob. Res.*, *1*(3), 409–428.

Yokoi, K., Nakashima, K., Kobayashi, M., Mihune, H., Hasunuma, H., Yanagihara, Y., et al. (2006). A tele-operated humanoid operator. *Int. J. Rob. Res.*, *25*(5-6), 593–602.