Imperial College
London

ONLINE MULTIMODAL AUTONOMOUS LEARNING OF ROBOTS INTERNAL MODELS

MARTINA ZAMBELLI

Thesis submitted for the degree of Doctor of Philosophy

Supervised by PROFESSOR YIANNIS DEMIRIS
Personal Robotics Lab
Department of Electrical and Electronic Engineering
Imperial College London

September 2017

Copyright declaration

Originality declaration

I hereby declare that this thesis and the work herein detailed, was composed and originated by myself, except where appropriately referenced and credited.

London, September 2017

Martina Zambelli

Abstract

Robots can learn new skills by autonomously acquiring internal models that can be used for action planning and control. The ability of learning internal models with no prior information allows robots to be fully autonomous not only in the acquisition of such models and motor skills, but also in adapting to new environments and working set-ups. This is particularly important for robots interacting with humans in unconstrained environments. Autonomous learning eases the engineering work of pre-programming each robotic system for each particular task, while endowing robots with flexibility, adaptability and versatility.

This thesis investigates how the use of multiple sources of information can influence such autonomous learning process. In particular, multiple prediction hypotheses provided by different prediction models, as well as information available to a robot from multiple sensory modalities (such as vision, touch, proprioception) are leveraged to enhance the learning process.

Through autonomous exploration a robot can bootstrap internal models of its own sensorimotor system that enable it to predict the consequences of its actions (forward models) or to generate new actions to reach target states (inverse models). This thesis studies how multiple information can enhance the bootstrapping process of these models or their use in environments and tasks that involve integration of different types of data.

It is shown that the use of multiple sources of information benefits the learning process. The combination of multiple predictors allows to enhance forward models' accuracy. The use of multiple sensory modalities is fundamental to perform tasks that are inherently multimodal, such as playing a piano keyboard. Also, multimodal integration allows a versatile applicability of the model learned. Furthermore, the learned multimodal model can be deployed in learning and control frameworks to predict the robot and other agents' motion, and to plan the robot's actions.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Prof. Yiannis Demiris, for giving me the opportunity of developing my research in the Personal Robotics laboratory at Imperial College London, and, more importantly, for the continuous support and guidance, the wise advice, the constructive and motivational conversations, the invaluable and uncountable opportunities to learn and grow as a researcher.

I would also like to thank my friends and members of the Personal Robotics Lab, who made my PhD journey so much more enjoyable, active and fun: Ayse, Miguel, Yixing, Tobias, Maxime, Theodosis, Hyung Jin, Antoine, Oya, Mark, Caterina, Fan, Ahmed, Josh, Ruohan, Dimitrios, Kyuhwa. I loved all the lunches, coffee breaks and discussions, the karaoke and Catan nights, the operas, squash, the conference times shared. Many thanks go also to the very good friends I met during these years: Debashis, Patric, Alessandro, Olivia, Nima, Emma, Dip, Francesca. I had the best time in the trips, climbing, Diwali festivals, dinners (with tiramisù!) and adventures we spent together.

During these past years I have been able to maintain myself thanks to the support of the Engineering and Physical Sciences Research Council and of Prof. Demiris. I am also grateful to Dr. Steven Right and Dr. Daniel Nucinkis, who gave me the opportunity to teach Maths to undergraduate students. I also want to acknowledge Patrick, the

group administrator who made it so easy to go through all the PhD bureaucracy.

Thanks to my parents, and deep gratitude to Marco and Alberto. I am very much grateful to my brother, Marco, always open to listen to my troubles, to support me and to encourage me to be stronger. Finally, I am deeply grateful to Alberto Padoan, for his love and patience, for tirelessly believing in me, for always supporting me in good and bad times, for being a thoughtful, focused, frank and loving partner, and for his continuous effort in encouraging me to win my fear of jumping from cliffs (real and metaphorical ones).

"Sai chi è che non sbaglia mai?... Chi non fa niente"
("Do you know who is never wrong?... Those who do nothing")
- Marino (my grandfather)

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Complex robots rely on internal models to achieve motor control and planning. Classical approaches to robot control and motion planning include formulating these models analytically, by using predefined parameters and structures. These approaches, however, present several drawbacks. For example, the mechanical structure of the robot may vary over time due to wear or damages. Also, the tasks that a robot will perform over its life time may change. A fixed and pre-programmed behaviour is effective for industrial robots, which are involved in repetitive actions and do not interact with human users. On the contrary, robots interacting with humans in their everyday activities must be able to adapt to new environments and to learn quickly new skills. For this type of robots, a fixed and pre-programmed approach to design their behaviour is not ideal.

The focus of this thesis is to endow robots with learning capabilities, with the goal of making them learn their sensorimotor capabilities in an autonomous way. This knowledge can then be used to accomplish complex cognitive tasks involving the interaction with human users, such as imitation and prediction of others' actions.

Learning processes can effectively be used to acquire models of actions, sensory systems, skills. Humans are vivid examples of successful applications of autonomous learning approaches. There are

neuroscientific evidences that humans develop their internal models through a learning process which starts in the first months of their infants' life. Based on such examples and findings, a branch of robotics, namely developmental robotics, aims to acquire internal models for robots by designing learning mechanisms to let the robots build their own perceptive and behavioural repertoires. The focus is to investigate the acquisition of motor skills from sensorimotor interaction with the environment. As a result, the developmental approach aims to endow robots with all the learning capabilities that may be necessary to build rich and flexible sensorimotor representations. These representations generally correspond to internal models. There are two types of internal models: the forward and the inverse models. The forward model can predict the next sensory state, given the current sensory state and the motor command. The inverse model is a mapping in the opposite direction: given a target (goal) state and the current state, it provides the motor commands needed to reach the goal. These models have been related to the central nervous system that internally simulates the motor system in planning, control and learning. Also, interestingly humans can use their internal models not only to predict their own actions, but also to make predictions about others' actions.

This thesis investigates how the use of multiple sources of information can influence autonomous learning processes on humanoid robots. The research questions addressed in this thesis are the following:

- Can the use of multiple predictors enhance autonomous learning of robots forward models, and how?
- Can multiple sensory modalities be integrated and combined in the learning process of inverse models and in imitation tasks, and how?
- Can multimodal self-learned internal models be used by a robot to predict and imitate others' actions, and how?

In this thesis, a humanoid iCub robot has been used to demonstrate the methods proposed to learn internal models from self-experience and through the integration of information from multiple sources. Online ensemble learning algorithms have been developed to learn for-

ward models, and a multimodal least square regression approach has been adopted to learn multimodal inverse models. Furthermore, a deep neural network has been proposed to predict and imitate other agents' motion. The methods developed and the results presented in this thesis have implications for researchers interested in continuous online autonomous learning. Also, the use of multiple sensory modalities in the learning process is a fundamental contribution in the field of autonomous model learning. The approaches considered in this thesis make no assumptions on prior knowledge available to a robot regarding its kinematic/dynamic structure. This hypothesis allows the use of the same methods on different robotics platforms. Finally, the use of the self-learned internal models on tasks involving a human teacher is demonstrated, thus showing the potential of this approach also in contexts where the robot can learn and interact with other agents.

1.1 CONTRIBUTIONS OF THIS THESIS

The main contribution of this thesis is the integration of multiple sources of information in the autonomous learning process that a robot can use to acquire its own internal models. Different approaches are proposed to acquire models from multiple sensorimotor information in a fully autonomous manner, assuming no prior knowledge on the robot kinematic or dynamic structure. The proposed approaches are based on learning internal forward and inverse models in a fully autonomous way, using robot motor babbling. In this thesis, it is shown that multiple sources of information benefit the learning process deployed to acquire internal models, either by exploiting and combining multiple predictor models, or by using and integrating multiple sensory modalities (*e.g.* vision, touch, proprioception). The self-learned models allow robots not only to predict and plan their own motion, but also to learn new skills from imitating human demonstrators, and to make predictions of the motion of another agent (*e.g.* a human). The novelties of the proposed approaches are the use and integration of multiple sensory modalities for learning, and the exploit-

ation of the learned models to go beyond the self and to effectively interact, interpret and learn from others.

The original contributions of this thesis can thus be summarised as follows.

- *Autonomous learning of forward models using ensembles of different types of predictors.* The anticipation capabilities of an autonomous robot must be accurate in order to achieve a proper understanding of its own motion capabilities and of its environment. The problem of assuring accuracy of the self-learned forward models has thus been considered. Ensemble methods have been proved effective to enhance the accuracy of classifiers and predictors. This is obtained by combining strategically multiple predictors so that the final prediction benefits from each predictor contribution in the best possible way. In order to realise a method that allows the robot to learn continuously as new data are available, online ensemble algorithms have been developed. The proposed methods leverage the diversity among different types of predictors. The methods enhance the prediction accuracy of self-learned forward models, and can update online as the robot collects new information from self-exploration and interaction with human demonstrators.

- *Autonomous learning of multimodal inverse models and imitation learning exploiting information from multiple senses.* The integration of multiple modalities has been considered in the learning process to build multimodal internal models able to integrate data from different sensors. This is an important aspect in the learning process, which has rarely been taken into consideration. Humans make extensive use of different senses in order to capture information about their body and environments. Analogously, in this thesis multimodal data have been considered to learn internal models. By integrating and leveraging multimodal information, the learned internal models can take into consideration different types of constraints, which in turns allow to perform imitation tasks exploiting different sensory modalities during the execution of a multimodal task. The method proposed

to address multimodal imitation is based on least square regression and allows to achieve on-the-fly imitation of multimodal references.

- *Use self-learned models to predict and imitate others through reconstructing non-observable sensory modalities.* It has been shown that humans use their internal models not only to predict and plan their own actions, but also to make predictions of actions of others. This skill is fundamental for autonomous robots that are required to interact with human users or more generally with other agents. Being able to make predictions about others' motion, is likely to be fundamental in order to achieve a smooth interaction. The main challenge in using self-learned internal models to predict others, is the difference of the data available to the robot to make predictions. While self-data are usually completely available at every time for the robot, only the visual information can be captured when observing others. To solve this issue, a multimodal deep variational autoencoder has been proposed not only to reconstruct the missing sensory modalities, but also to generate new data, such as motor commands needed for example to imitate the observed movements.

1.2 ROADMAP

Figure 1.1 provides an overview of the robotics system, its environment, sensorimotor system and internal models as used in this thesis.

The rest of this thesis is organised as follows:

- Chapter 2 sets the overall scene by reviewing the relevant literature on internal model and model learning. Internal models are introduced both from a developmental psychology and from a robotics perspective, while model learning is presented in light of recent advances in the robotics field.

- Chapter 3 presents how the use of multiple predictors allows to improve the prediction accuracy of forward models learned from self-exploration, when no prior knowledge is assumed for

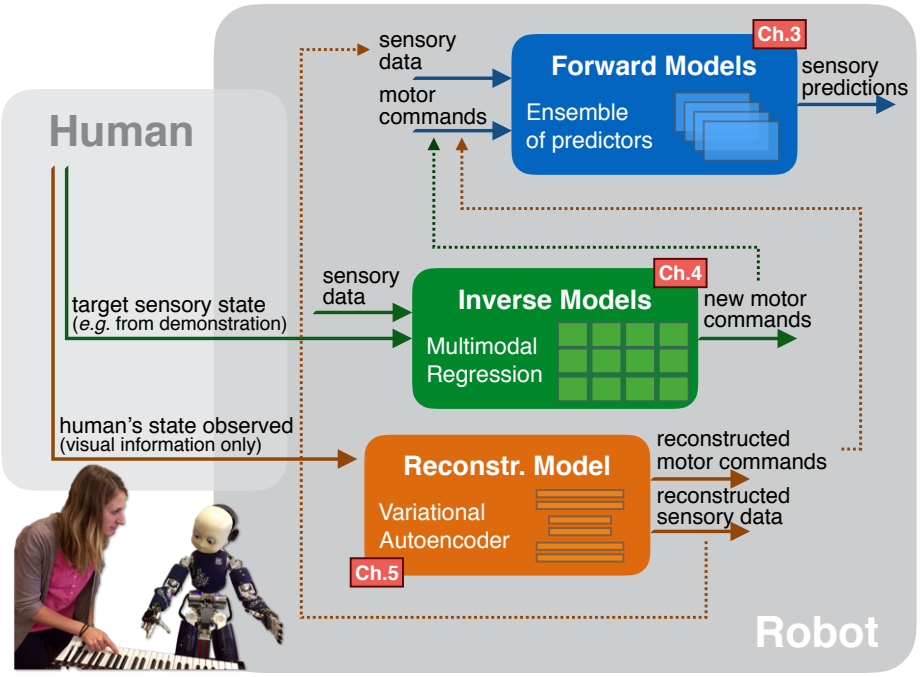


Figure 1.1: Thesis roadmap. A robot can learn internal models from autonomous exploration. A human can provide the robot with target state or observations, that the robot can use to achieve complex cognitive tasks, such as imitation and prediction of others' motion. The red rectangles refer to the associated chapter where each component is discussed.

the robot. The ensemble learning methods proposed are based on online learning algorithms that also leverage the multimodal sensorimotor data available to the robot.

- Chapter 4 addresses the problem of using self-learned internal models to achieve complex cognitive tasks, such as imitation, only based on self-acquired sensorimotor mappings. The method proposed is based on a least square regression formulation and allows to achieve on-the-fly imitation of multimodal references.

- Chapter 5 illustrates the implementation of a more complex learning architecture based on a self-learned multimodal deep variational autoencoder, which allows to reconstruct data from partial observations, to predict and imitate others' motion only based on self-learned mappings.

- Chapter 6 summarises the important points of this thesis and suggests directions for future research.
- In Appendix, details are given on the specific learning algorithms used to construct the ensemble method, and on the experimental platforms (including the iCub humanoid robot, the YARP middleware, and the piano keyboard used in the experiments). Additional experiments conducted on a Baxter robot are also reported.

1.3 PUBLICATIONS

The contributions and results of this thesis have been published.

Most relevant publications

- M. Zambelli, A. Cully and Y. Demiris, "A multimodal deep model for prediction and imitation of self and others' sensorimotor states", IEEE Transactions on Robotics (under review).
- M. Zambelli and Y. Demiris, "Online Multimodal Ensemble Learning using Self-learnt Sensorimotor Representations", IEEE Transactions on Cognitive and Developmental Systems, vol. 9(2), pp.113-126, 2016.
- M. Zambelli, T. Fisher, M. Petit, H.J. Chang, A. Cully and Y. Demiris, "Towards Anchoring Self-Learned Representations to Those of Other Agents", Workshop on Bio-inspired Social Robot Learning in Home Scenarios of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.
- M. Zambelli and Y. Demiris, "Multimodal Imitation using Self-learned Sensorimotor Representations", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.
- M. Zambelli and Y. Demiris, "Online Ensemble Learning of Sensorimotor Contingencies", Workshop on Sensorimotor Contin-

gencies for Robotics, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015.

Other publications

- H.J. Chang, T. Fischer, M. Petit, M. Zambelli, Y. Demiris, "Learning Kinematic Structure Correspondences Using Multi-Order Similarities", IEEE Transactions on Pattern Analysis and Machine Intelligence (minor revisions).

- C. Moulin-Frier, T. Fischer, M. Petit, G. Pointeau, J.-Y. Puigbo, U. Pattacini, S. Ching Low, D. Camilleri, P. Nguyen, M. Hoffmann, H.J. Chang, M. Zambelli, A.-L. Mealier, A. Damianou, G. Metta, T. J. Prescott, Y. Demiris, P. F. Dominey, P. F. M. J. Verschure, "DAC-h3: A Proactive Robot Cognitive Architecture to Acquire and Express Knowledge About the World and the Self", IEEE Transactions on Cognitive and Developmental Systems, 2017.

- H.J. Chang, T. Fischer, M. Petit, M. Zambelli, Y. Demiris, "Kinematic Structure Correspondences via Hypergraph Matching", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

CHAPTER 2

Background and literature review

In this chapter, the main notions of internal models and model learning are recalled, and the relevant literature is presented. Internal models are illustrated from the perspective of both human cognitive development and robotics. Studies from the developmental psychology and neuroscience fields are revised to give an account of the fundamental notions on which the thesis is based on. The robotics literature is presented, with particular focus on the areas of model learning and developmental robotics, including classical approaches as well as recent advances in these fields.

2.1 INTERNAL MODELS

Studies in the field of neuroscience and neurophysiology have provided evidence of the existence of multiple representations of the body in the human brain, associated with different objectives and functionalities (Marshall and Meltzoff, 2015; Wolpert, Diedrichsen and Flanagan, 2011; Miall and Wolpert, 1996; Wolpert and Flanagan, 2001; Ishikawa et al., 2016). Internal models of the body have emerged in different experiments, and used to process sensory inputs such as proprioceptive

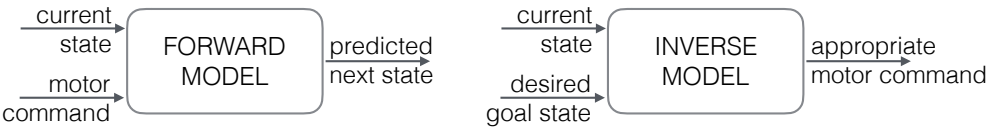


Figure 2.1: A forward model (left) and an inverse model (right).

tion from muscle and joints, vision, tactile perception (Marshall and Meltzoff, 2015).

The body maps existing between motor and sensory variables are also termed *internal models* as they represent features of the body or the environment (Wolpert, Diedrichsen and Flanagan, 2011). There are two types of internal model: *forward models* and *inverse models*. The former provide predictions of the state of the agent given the current state and an action, while the latter provide mappings in the opposite direction: given a target state and the current state, they retrieve the action to bring the system from the current state to the target. A schematic representation of these models is shown in Figure 2.1. These two internal models play important roles in human brains and are involved in action planning, prediction, control (Wolpert and Kawato, 1998; Wolpert and Flanagan, 2001).

Forward models: Forward models have been related to the central nervous system which internally simulates the motor system in planning, control and learning (Miall and Wolpert, 1996). Several studies in human and primates (Miall et al., 2007; Nowak et al., 2007; Izawa, Criscimagna-Hemminger and Shadmehr, 2012; Popa, Hewitt and Ebner, 2013; Ishikawa et al., 2016) suggested that the cerebellum is related to the forward model, which plays a fundamental role by providing predictions of sensory consequences of motor commands and allowing the discrimination between external sensory input and input deriving from motor action. They are also strictly related to anticipation, a fundamental aspect that distinguishes cognitive agents from adapted reactive systems, which do not perform explicit predictions (Pezzulo, 2007). Anticipation presents several advantages: it enhances adaptability and play a role not only in increasing individual cognitive functions, but also in extending the capabilities to learn complex and abstract concepts, as well as in developing efficient social interac-

tions. Furthermore, the use of forward models for internal feedback allows to estimate the outcome of an action before sensory feedback is available. In other words, they make possible to realise mental practice, that is predicting the outcomes of actions without actually executing them (Wolpert, Ghahrmani and Jordan, 1995).

Motor prediction and mirror neurons: Motor prediction plays an important role in motor control but also in high level cognitive functions such as observation and understanding of others (Wolpert and Flanagan, 2001), mental practice, imitation and social cognition (Wolpert and Kawato, 1998). Studies have shown that people can make predictions of their own actions as well as others' behaviour through the use of internal models of the central nervous system (Blakemore, Decety and Albert, 2001; Decety and Sommerville, 2003; Wolpert and Flanagan, 2001; Kilner, Friston and Frith, 2007; Pickering and Clark, 2014; Wolpert, Doya and Kawato, 2003). More interestingly, people predict others' behaviours using the same internal models that they use for predicting their own behaviour, through a process called *simulation* (Gallese and Goldman, 1998; Cruz and Gordon, 2003; Hesslow, 2012). Assuming existing similarities between agents, the internal model used to predict one's own actions can be instrumental to predict the (*visual*) consequences of someone else's actions (Pickering and Clark, 2014). The mirror neuron system plays a key role in this framework. Mirror neurons are special brain cells which activate both when producing own actions and when observing similar actions executed by other individuals (Rizzolatti, 2005; Cattaneo and Rizzolatti, 2009). This indicates that perception and action generation share the same neural structure (Gallese and Goldman, 1998; Fabbri-Destro and Rizzolatti, 2008). The discovery of mirror neurons had a great influence in understanding how people perceive and potentially imitate others (Demiris, Aziz-Zadeh and Bonaiuto, 2014a). A mirror neuron is a neuron that "mirrors" the behaviour of the other, as if the observer were itself acting. In other words, when we are looking at someone performing an action, the same motor circuits that are recruited when we ourselves perform that action are concurrently activated. This led to the hypothesis that one possible function of the mirror neuron sys-

tem could be to promote learning by imitation: when new motor skills are learned, some time is often spent in the first training phases to replicate the movements of an observed instructor (Gallese and Goldman, 1998). These observations link directly to the ability of understanding others' actions as well as the ability of simulating them using our own internal models.

Inverse models: Inverse models can also be referred to as behaviours or controllers (Narendra and Balakrishnan, 1997; Lazkano et al., 2007). While the main objective in learning forward models is to obtain predictions that are as close as possible to the actual state of the agent, specifications to learn inverse models are less straightforward. This is related to the fact that there exist multiple actions an agent may take to achieve a goal. There is physiological evidence showing that parts of the cerebellum represent inverse models and output directly to the controller (Kawato, 1999; Wolpert, Miall and Kawato, 1998). Furthermore, a number of cortical areas (*e.g.* the primary motor cortex, the premotor contex, the parietal cortex and the prefrontal cortex) contribute to the voluntary control of arm movements, and several studies (Haruno, Wolpert and Kawato, 2001; Kawato and Gomi, 1992) suggested that the cerebellum plays a role in the generation of motor commands as an inverse model.

Experience: Finally, internal models are not fixed entities: as the body of the agent changes and experience is accumulated, new models need to be acquired and existing models need to be updated. Internal models must hence be learned and updated throughout experience (Wolpert and Flanagan, 2001). Studies have argued that infants use self-exploration and self-stimulation to "calibrate" their sensorimotor and body representations (Spencer and Quinn, 1997). The role of exploration is thus fundamental to form and test models.

2.2 MODEL LEARNING IN ROBOTICS

Several robotic systems use manually designed mathematical models, such as kinematic and dynamic models (Siciliano and Khatib, 2016), based on the assumption that the motor system of such robotic systems is known beforehand. However, learning approaches present numerous benefits: they can cope with changes of the robot morphology over time, they allow to ease the programming effort of modelling complex robotic structures from sensor data, they enable the use of the same software on different robots, they can adapt to different types of interaction with environments and users. Learning algorithms are effective in building internal models for robots. They achieve flexibility and adaptability in building robots' kinematic and dynamic models, by incorporating uncertainties and nonlinearities, as well as dynamical changes due to wear, and in limiting the influence of specific engineered settings.

Several studies addressed the problem of model learning in robotics. Typically, learning forward models has been less investigated (compared to learning inverse models) in traditional robotics because they can be directly defined based on the kinematic structure of the robot. For example, forward models of serial robots can easily be computed analytically. However, forward models of complex, advanced robots may be difficult to formulate. Also, learning forward models is fundamental for robots to be able to make accurate predictions not only about their own actions but also about others' actions. Learning methods based on multilayer perceptrons networks have been used to learn forward kinematic models of a number of simple systems (Nguyen, Patel and Khorasani, 1990; Sadjadian and Taghirad, 2005). In (Nguyen, Patel and Khorasani, 1990), it was shown that a three-layer back-propagation network was capable of learning the forward kinematics of a simulated two degrees-of-freedom planar arm without any knowledge of the manipulator's kinematic structure. In (Sadjadian and Taghirad, 2005), a more complex robotic system was considered, namely a three degrees-of-freedom actuator redundant hydraulic parallel manipulator, involving highly coupled nonlinear equations. Different neural network structures were studied to learn

the forward kinematics of such system, showing satisfactory performance on tracking and error estimation. A method based on Bayesian networks has been proposed in (Dearden and Demiris, 2005), where a robot was able to autonomously learn forward models of its motor system and then use it to imitate human movements.

While forward models are uniquely determined, inverse models are generally not and do not always exist. Three main approaches have been used to learn inverse models (Wolpert and Kawato, 1998): direct inverse modelling (Miller, 1987; Kuperstein, 1988), distal supervised learning (Jordan and Rumelhart, 1992) and feedback error learning (Kawato, 1990). Direct inverse modelling treats the problem of learning an inverse model as a classical supervised learning problem. Distal supervised learning and feedback-error-learning instead rely on the ability to convert errors at the trajectory level into error at the motor level. Unlike the direct inverse modelling approach, they can acquire accurate inverse models even for redundant systems, while the direct approach can fail in learning control of redundant robots because of the existence of a one-to-many inverse kinematics problem (Jordan and Rumelhart, 1992).

A comprehensive survey focused on model learning for robot control was presented in (Nguyen-Tuong and Peters, 2011). Many approaches to learn controllers for robots have been proposed, including for example reinforcement learning (Sutton and Barto, 1998; Abbeel et al., 2007) and learning by demonstration (Argall et al., 2009; Billard et al., 2008; Atkeson and Schaal, 1997).

Several approaches have been proposed in which the inverse model is directly learned from data, by adopting machine learning techniques. Gaussian processes have been used in (Deisenroth and Rasmussen, 2011) to learn a probabilistic dynamics model (*i.e.* PILCO - probabilistic inference for learning control) where the model uncertainty was explicitly incorporated into long-term planning, and in (Williams et al., 2009), where the multi-task learning problem rising from solving the inverse dynamic problem for a robotic manipulator holding different loads was addressed. In (Hunt et al., 1992), a survey on the use of neural networks from a classical control theoretic point of view, addressing problems of modelling, identification and con-

trol of nonlinear systems was presented. Neural networks were also considered in (Kawato et al., 1988), where a hierarchical model was introduced to deal with computational problems at different levels, including determination of desired trajectories in the visual coordinates, transformation of the trajectory from visual coordinates to body coordinates, and generation of motor commands. Applications to the robotics field were also outlined in that work. The Locally Weighted Projection Regression (LWPR) algorithm has been used in (D'Souza, Vijayakumar and Schaal, 2001) to learn the inverse kinematics of a humanoid robot, while the Infinite Mixture of Linear Experts algorithm has been used in (Damas, Jamone and Santos-Victor, 2013) to solve the same problem. Both these approaches allow to learn the model while performing a task, and are based on finding an approximation of the Jacobian.

A broadly used class of methods to learn inverse models is the Learning by Demonstration, or Programming by Demonstration framework (Billard et al., 2008). Seminal works on this class of approaches are for example (Billard, 2001; Calinon, Guenter and Billard, 2007). In (Billard, 2001), a biologically inspired model for motor skills imitation was presented. Inspired by the control of movements in primates and the functions of different areas of the brain, that study proposed a model based on modules corresponding to different functionalities and level of abstractions. A framework for extracting relevant features of a given task and for addressing the problem of generalizing the acquired knowledge to different contexts was instead proposed in (Calinon, Guenter and Billard, 2007). There, a method was proposed to extract the important features of a task based on spatio-temporal correlations across a multivariate dataset, to determine a generic metric to evaluate the robot's imitative performance, and to optimize the robot's reproduction of the task, according to the metric of imitation performance in new contexts. In order to learn inverse models from data, approaches based on learning by demonstration typically rely on the acquisition of target trajectories, that can be collected from multiple demonstrations of each task. This approach has been explored for example in (Korkinof and Demiris, 2013; Chatzis et al., 2012; Soh and Demiris, 2013; Soh and Demiris, 2015), where statistical meth-

ods have been proposed to infer models from multiple demonstrated trajectories. However, multiple demonstrations might be difficult to obtain. Other methods are based on a single demonstration instead, *e.g.* in (Wu, Su and Demiris, 2014; Wu and Demiris, 2010b; Wu and Demiris, 2010a; Wu and Demiris, 2009). In this case, however, it is assumed that all input features are observable. A diverse approach was proposed in (Lee et al., 2013; Lee, Kim and Demiris, 2012), where context free grammars were used in order to learn sequences of demonstrated actions.

A considerable number of works has recently addressed the problem of task learning and policy search adopting the reinforcement learning approach (Andrew Bagnell, 2014) to learn complex movement tasks in robotics. In (Kormushev, Calinon and Caldwell, 2013), the expectation-maximization-based reinforcement learning approach was applied to real robots ranging from bipedal robots to planar robotic arms. In (Deisenroth, Fox and Rasmussen, 2015), an approach based on a probabilistic, non-parametric Gaussian process transition model was proposed to speed up learning by extracting more information from data. In (Levine, Wagener and Abbeel, 2015) the guided policy search approach was used to learn a set of trajectories for a desired motion skill then unified into a single control policy that can generalize to new situations. An automatic method for interactive control of physical humanoid robots based on high-level tasks has been proposed in (Mordatch et al., 2016). The method proposed in that study is based on the combination of a model-based policy that is trained off-line in simulation and sends high-level commands to a model-free controller that executes these commands on the physical robot. Although these approaches achieve high performance on a number or different tasks, they typically require a large amount of data and training time. Another crucial aspect of these approaches is the definition of a reward/cost function to be optimise, which usually depends on the specific task.

Finally, deep learning methods have recently received increasing attention and have proven successful in several robotics applications (LeCun, Bengio and Hinton, 2015; Hinton, Osindero and Teh, 2006; Levine et al., 2016; Sigaud and Droniou, 2016). These methods are

based on deep artificial neural networks consisting of more than one hidden layer. They represent an effective approach to learn data representations, in particular when a large number of inputs or dimensions are involved. In (Ngiam et al., 2011), a multimodal deep learning approach was presented, able to cope with data of different types, such as visual and audio data, with cross-modal learning and reconstruction. Deep learning has been previously exploited also in the context of developmental learning, for example in (Droniou, Ivaldi and Sigaud, 2015). In that study, an architecture based on deep networks was proposed to make a humanoid robot iCub learn a task from multiple perceptual modalities (namely proprioception, vision, audition). Two recent works (Baraglia et al., 2015; Copete, Nagai and Asada, 2016) have applied deep autoencoders to make a robot predict others' actions through predictive learning. In these studies, it was shown how a robot can use a self-acquired model to make predictions of others' goals.

DEVELOPMENTAL LEARNING "Developmental robotics is the *interdisciplinary approach to the autonomous design of behavioral and cognitive capabilities in artificial agents (robots) that takes direct inspiration from the developmental principles and mechanisms observed in the natural cognitive systems of children*" (Cangelosi, Schlesinger and Smith, 2015).

In developmental robotics, robots' internal models are acquired by designing learning mechanisms to let a robot build its own perceptive and behavioural repertoire. The focus is to investigate the acquisition of motor skills from sensorimotor interaction with the environment (Lungarella et al., 2003). As a result, the developmental approach aims to endow robots with all the learning capabilities that may be necessary to build rich and flexible sensorimotor representations (Sigaud and Droniou, 2016).

Biological systems are a good example of the importance of sensorimotor learning to the system's adaptability and robustness, and a general agreement also exists on the importance of learning internal models in robotics frameworks (Bays and Wolpert, 2007; Wolpert, Miall and Kawato, 1998; Hoffmann et al., 2010). Sensorimotor learning involves learning the mappings between motor and sensory variables.

Well-established learning systems for motor prediction and control (Wolpert and Kawato, 1998; Kawato, 1999; Demiris and Khadhouri, 2006) are based on internal models. Inspired by findings in human brain and cognitive development, robotics control architectures have been proposed and implemented using forward and inverse models. Notable examples are the HAMMER (Hierarchical Attentive Multiple Models for Execution and Recognition) architecture (Demiris and Khadhouri, 2006), and the MOSAIC (MOdular Selection And Identification for Control) architecture (Sugimoto et al., 2012; Haruno, Wolpert and Kawato, 2003).

Exploration: Inspired by infants' developmental steps, biologically inspired architectures are based on self-exploration strategies to collect the information used to learn internal models. Analogously, a robot can explore its sensorimotor capabilities through self-exploration, or *motor babbling* (Demiris and Meltzoff, 2008). The motor babbling strategy has been adopted in different works, *e.g.* (Demiris and Dearden, 2005; Dearden and Demiris, 2007; Schillaci, 2013; Ramırez-Contla, Cangelosi and Marocco, 2012; Zhong, Cangelosi and Wermter, 2014). Other studies, however, claim that early movements are already goal-directed (Hofsten, 2004). A variation of the motor babbling approach, called *goal babbling*, has been proposed in (Rolf, Steil and Gienger, 2010) to learn inverse models from self-exploration, a strategy that has been used for example also in (Rolf, Steil and Gienger, 2011; Jamone et al., 2011).

ONLINE LEARNING A real challenge in robot learning is devising algorithms that allow online adaptation. The importance of online learning in robotic applications is related to the fact that robots are required to interact in a continuously evolving environment. Also, changing of contexts, such as tools and other agents, make online strategies attractive (Jamone et al., 2012; Hersch, Sauser and Billard, 2008; Jamone et al., 2013). Another motivation for online model learning is that it is difficult if not impossible to cover the complete state space with data beforehand (Nguyen-Tuong and Peters, 2011). Although many state-of-the-art learning paradigms are based on batch

processing, several works have proposed online strategies to learn robot internal models.

Several studies have explored online learning methods on humanoid robots. A biologically inspired model for online and continuous learning of visuo-motor coordination has been proposed in (Schillaci, Hafner and Lara, 2014), where dynamic self-organising maps associated through Hebbian links have been adopted for learning the visuo-motor coordination online on a Nao humanoid robot. An online learning approach to achieve reaching behaviour in a humanoid robot has been proposed in (Jamone et al., 2012), where the receptive field weighted regression algorithm (Vijayakumar and Schaal, 2000b) has been employed to learn online a representation of the robot's reachable space. In (Hersch, Sauser and Billard, 2008) an online strategy has been implemented to learn the kinematic structure of a humanoid robot, yielding the position of each segment and computing the associated Jacobians.

Studies on mobile robots also addressed online learning. In (Hoffmann, 2007), authors showed that the ability to anticipate acquired by a mobile robot can be exploited for perceptual judgement. In particular, the forward model learned autonomously by the mobile robot has been effectively used to realise mental simulation, hence achieving tasks such as distance estimation and recognition of a dead end. In (Tani and Nolfi, 1999), an online learning scheme was developed using a mixture of recurrent neural networks, and examined through simulation experiments involving the online navigation learning problem. The learning architecture proposed in that study is based on a set of modules (recurrent neural networks) that can self-organise on multiple levels of abstraction in order to account for different categories of sensorimotor information.

Online algorithms have been also used to learn inverse dynamic models for anthropomorphic arms: the locally weighted projection regression (Vijayakumar and Schaal, 2000b) has been used in (Schaal, Atkeson and Vijayakumar, 2002), a support vector regression approach has been used in (Choi, Cheong and Schweighofer, 2007), and the local Gaussian process regression approach has been illustrated in (Nguyen-Tuong, Seeger and Peters, 2009). Although these methods

have been shown to be effective in learning control policies online, their ability of scaling on high dimensional spaces to describe the robotic system is usually limited.

MULTIMODAL LEARNING Vision perception plays a key role in cognitive tasks and in the execution of everyday activities. Works in the fields of cognitive science, neuroscience, as well as robotics and computer vision, have studied vision perception from different perspectives. In (Harding et al., 2012; Leitner et al., 2012) the objective was to learn to locate objects from vision using artificial neural networks and a genetic programming method. In (Borji and Itti, 2013) the focus was put on visual attention modelling, thus linking vision with a notion of relevance. A large number of works in the literature (*e.g.* (Sturm, Plagemann and Burgard, 2009) and (Kajic et al., 2014)) have employed visual markers or color blobs to identify and track the object of interest in the visual frames, *e.g.* to identify and learn the position of the hand or end-effector of a robot in the visual space. Although this is a practical way of approaching visual learning, it also implies an augmentation of the reality of a robot, by adding external elements to the original structure. More recent works, *e.g.* (Broun et al., 2014), have used optical flow and recent techniques developed in computer vision fields to build kinematic model for robots. An interesting branch of computer vision has developed methods to segment figures and motion and to identify skeletons of moving agents (*e.g.* (Ross, Tarlow and Zemel, 2010)). These techniques represent a promising way to recognise limbs or body structure, without relying on external clues. Whereas a variety of different sensors has become available on advanced robots, most of the approaches to model learning are still based on the use of data from a single modality, usually vision.

Despite the key role of vision, however, many of the tasks performed during everyday activities intrinsically involve the use of different sensory feedback from multiple senses. For example, to draw on a board we engage vision to check the result of our drawing, proprioception to perform smooth movements with our arm, and touch to sense the contact with the board. Playing an instrument also

involves sound, in addition to the visual, tactile and proprioceptive feedback.

The problem of integrating inputs of different types has been studied in the research areas of sensor fusion and pattern recognition. Several works have addressed the problem of learning representations from multiple sources, *e.g.* text and audio or text and images (Ramisa et al., 2017; Yan et al., 2014; Poria et al., 2016), sensor networks or signals from different robots in multi-robot set-ups (Olfati-Saber and Shamma, 2005; Gravina et al., 2017; Dietrich et al., 2016; Cho et al., 2014; Chen, Jafari and Kehtarnavaz, 2016). Recent advances on deep neural networks have allowed the development of several effective solutions, for example based on convolutional neural networks (CNN). Studies in multimedia research has also focussed on the problem of learning representations from multimedia contexts, such as scene descriptions, event detection and cross-domain feature learning (Cho, Courville and Bengio, 2015; Gan et al., 2015; Kang et al., 2015; Chang et al., 2017; Yang, Zhang and Xu, 2015). Solutions in this field take advantage of the progress in deep feature representations and unsupervised learning, and are often based on measures of the correlation between different sources of information.

In robotics, the approach of multimodal learning can be used to leverage the different sensing capabilities of robots to build a more complete and effective representation of their structure. Recent work has been developed in this direction. Vision and proprioception were considered *e.g.* in (Sturm, Plagemann and Burgard, 2009; Schillaci, 2013; Dearden and Demiris, 2007). The touch sense was used *e.g.* in (Roncone et al., 2014; Yoshikawa et al., 2002; Fuke, Ogino and Asada, 2007). Vision and audition were used in (Martinez, Lungarella and Pfeifer, 2008). Multiple modalities have been taken into consideration also in (Tidemann, Öztürk and Demiris, 2009; Schmidts, Lee and Peer, 2011; Stepanova et al., 2017; Zhong, Cangelosi and Ogata, 2017). A combination of sound and movements has been adopted in (Tidemann, Öztürk and Demiris, 2009) to imitate human drumming behaviours, while motion and force data have been used to teach grasping gestures to a simulated manipulator in (Schmidts, Lee and Peer, 2011). In (Stepanova et al., 2017) simultaneous tactile and

linguistic inputs were used to cluster body parts. In (Zhong, Cangelosi and Ogata, 2017), experiments have been presented to study the abstraction capabilities of networks from multimodal time series. Other studies, *e.g.* (Fadlil et al., 2013; Araki et al., 2011; Batula et al., 2013; Johnsson and Balkenius, 2011), have presented solutions for merging different sensors' data to address classification-type problems, such as object/gesture recognition or speaker identification/spatial localisation, rather than predicting or imitating demonstrated behaviours. Recently, studies have presented advances in integrating language in a multimodal framework for robot learning. For example, in (Zhong et al., 2017; Zhong, Cangelosi and Ogata, 2017), different types of recurrent neural networks have been deployed to address the problem of learning and performing abstraction using multiple modalities, including language. Finally, information from different sensors have also been used to solve imitation learning using different sensor informations, such as hierarchical architectures based on multiple internal models (Wu and Demiris, 2010a; Demiris and Dearden, 2005; Lopes and Santos-Victor, 2005; Tidemann, Öztürk and Demiris, 2009), and Gaussian Mixture Regression together with Hidden Markov Model (Schmidts, Lee and Peer, 2011).

2.3 SUMMARY

In this chapter, several studies relevant to the contributions of this thesis have been revised. The work of this thesis is fundamentally grounded in the developmental learning approach to robotics, based on learning internal (forward and inverse) models from experience derived from the interaction of a robot with its environment. The studies and approaches presented in Section 2.1 are thus adopted in this thesis as a guideline and as fundamental references. Works presented in Section 2.2, relative to model learning in robotics, vary with respect to applications, hardware, goals. The set-up and goals of this thesis relate particularly to previous work on learning forward models (Dearden and Demiris, 2005), to principles of developmental robotics (Cangelosi, Schlesinger and Smith, 2015), to fundamental studies

on multimodal learning (Ngiam et al., 2011) and approaches based on deep networks applied to robotics (Baraglia et al., 2015; Copete, Nagai and Asada, 2016). One of the challenges left open in learning internal models in a developmental fashion is how to integrate the information available from the different robot's sensors. While the majority of studies focussed on visual data and joint angles, there are many more sources of information that can be taken into consideration during learning, such as sound, touch, as well as the sensorimotor representations that can be synthesised by different learning algorithms. All this information can contribute to the learning process by introducing additional clues or to achieve more accurate representations. On the other hand, several studies on multimodal learning were only tested on datasets that were not directly related to a dynamical system (*e.g.* a robot) interacting with an environment and acting its own sensorimotor system. This scenario poses many challenges including the noise in the data collected, the misalignment of data streams, and the interdependencies among data generated by a possibly redundant system. This thesis set out to develop methods to learn internal models from multiple sources of information, through ensemble methods and multimodal learning. One of the working assumptions was to account for as little prior knowledge as possible at the beginning of the learning process. Unlike more classical approaches to control and planning (*e.g.* (Hunt et al., 1992; Siciliano and Khatib, 2016)), where typically several assumptions are made on the robot's structure and model, the methods proposed in this thesis contribute to the research area of autonomous developmental learning by including learning strategies based on self-experience that rely on information from multiple sources. Another working paradigm considered in this thesis has been to leverage the multiple different sensors available on the iCub, including motor encoders for proprioception, RGB cameras for vision, artificial tactile skin for touch. In addition to these data sources, a piano keyboard was used, from which sound data could also be collected. For such a complex scenario, methods including motor babbling and imitation learning have been considered to collect data for learning, while methods based on reinforcement learning where not due to the fact that they typically require very large amount of data

for learning. Only a small number of methods revised in the previous sections have been shown to effectively manage learning using multimodal data and using a developmental approach. The learning methods proposed in this thesis take advantage of the progresses in the area of ensemble learning and deep learning. While ensemble methods (Dietterich, 2000) allow effective combinations of predictors, deep neural networks such as variational autoencoders (Kingma and Welling, 2013), allow to capture complex representations of data. In conclusion, this thesis addresses the challenges of integrating multimodal learning and learning from multiple sources of information into a developmental framework on a physical iCub robot.

Background sections, focussing on more specific aspects of the thesis subjects, are included in the following chapters in order to provide a more specific contextualisation for each of the research challenges addressed.

CHAPTER 3

Online Ensemble Learning of Forward Models

This chapter addresses the first research question:

> Can the use of multiple predictors enhance autonomous learning of robots forward models, and how?

In this chapter, it is shown that the use of multiple predictors benefits the robot learning process to acquire a forward model by enhancing the model accuracy significantly.

An introduction on forward model learning on robots and humans and the relevant background are summarised in Section 3.1. The solution proposed to implement autonomous online learning of forward models on robots is then illustrated (Section 3.2). This is the first block of the learning architecture anticipated in Chapter 1.

3.1 Learning Forward Models from Self-Exploration

Complex robots rely on internal models describing the kinematics and dynamics for controlling and planning actions. Constructing analytical models of complex robotic platforms, however, often presents critical difficulties and costs. In particular, analytical models might be inaccurate because they are based on assumptions that are not realistic, such as the complete rigidity of the links. Also, nonlinearities

and uncertainties of the system are often difficult to include. Another critical problem related to the use of analytical models is that they are usually highly specific for a particular robotic platform. This also implies a reformulation of the model in case of modifications of the platform, thus adding time, computational and economical costs.

These observations motivate the interest in endowing robots with learning capabilities, in order to enable them to build their internal models through learning processes (Kawato, 1999; Haruno, Wolpert and Kawato, 2001; Demiris and Khadhouri, 2006) in which relations between actions and associated changes in sensory inputs, are involved. Another benefit of self-learned models is that, in principle, they can update over the life-time of the robot, avoiding re-modelling in case, for example, of damaged parts or hardware failures. If internal models are learned by the robot continuously, they can handle changes in the robot morphology or in the robot sensory system, while eliminating the need for explicit analytical model formulation and dealing with model drifts (Cully et al., 2015). Contrary to more classical control approaches based on hand-crafted kinematic and dynamic models, methods based on autonomous learning of sensorimotor representations can achieve complex behaviours, such as imitation, without the need of explicit model formulation.

A vivid and successful example of systems where autonomous learning of motor skills is continuously used is represented by humans. It is argued that infants use self-exploration and self-stimulation to "calibrate" their sensorimotor and body representations (Spencer and Quinn, 1997). Studies on motor behaviour of infants have shown that mobility at early age is characterised by variations in movement trajectories and in temporal and quantitative aspects of mobility, that are not neatly tuned to environmental conditions (Hadders-Algra, 2000). Analogously, a robot can explore its sensorimotor capabilities through self-exploration, or *motor babbling* (Spencer and Quinn, 1997; Demiris and Meltzoff, 2008; Demiris and Dearden, 2005; Mochizuki et al., 2013). This approach allows a robot to autonomously explore its own sensorimotor capabilities, avoiding the use of pre-programmed engineered behaviours or the analytical formulation of kinematic and dynamic models.

ONLINE ENSEMBLE LEARNING The focus of this chapter is to learn forward models with high prediction accuracy online. Ensemble learning is an effective approach to improve the classification or prediction performance of a model. In ensemble learning systems, multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular problem. The key for the success of an ensemble system and its ability to correct the errors of some of its members is the diversity among the classifiers of the ensemble. The intuition is that if each model makes different errors, then a strategic combination can reduce the total error.

Ensemble methods are popular research directions in machine learning and pattern recognition (Ranawana and Palade, 2006). Several methods (Dietterich, 2000; Ren, Zhang and Suganthan, 2016; Kuncheva and Rodríguez, 2014; Breiman, 1996a; Breiman, 2001) have been extensively studied and applied in the research areas of pattern recognition and sensor fusion. Several approaches combine feature selection algorithms with ensemble strategies, and evaluate popular ensemble approaches in different domains, spanning from text classification to biology, from concept detection to emotion recognition or land classification (Awais et al., 2011; Yijing et al., 2016; Schuller et al., 2005; Sun, Zhang and Zhang, 2007; Gislason, Benediktsson and Sveinsson, 2006; Whitehead and Yaeger, 2010; Tang et al., 2012; Yan et al., 2016; Sun et al., 2015; Onan, Korukoğlu and Bulut, 2016). Many ensemble learning algorithms have been used in robotics to solve problems such as localization, detection, recognition, decision making (Nyga, Balint-Benczedi and Beetz, 2014; Heinrich et al., 2013). However, the use of combinations of different predictors has rarely been deployed to build internal models or to learn and produce motor behaviours on robots. Nonetheless ensemble methods have been shown effective in different learning frameworks to achieve high prediction accuracy (Choraś et al., 2009; Valentini and Masulli, 2002; Mendes-Moreira et al., 2012).

Several approaches exist to solve offline regression problems (Mendes-Moreira et al., 2012), however, it is desirable that robots update their internal model in an online fashion, as new data become available from experience and explorations. Online ensemble learning algorithms for regression have recently been advanced (Ikonomovska, Gama and

Džeroski, 2015; Soares and Araújo, 2015). The online learning problem of predicting with experts was first presented in (Littlestone and Warmuth, 1994; Freund and Schapire, 1997; Cesa-Bianchi et al., 1997), and many algorithms have been developed in order to address difficult objectives such as learning with unknown number of experts, coping with different combinations of experts, achieving smaller regret while ensuring robustness. The AdaNormalHedge algorithm (Luo and Schapire, 2015) has been recently proposed in order to achieve all the diverse goals of online learning simultaneously, without using any prior information on the learners. Other well-known algorithms to build ensembles of multiple learners are the Bagging algorithm (Breiman, 1996a) and Random forests (Breiman, 2001). These algorithms have been used in many different applications, including computer vision and robotics ones. Finally, a very recent study has addressed the problem of online learning using multiple sliding Gaussian Processes (Meier and Schaal, 2016). This method has proved to be successful in building accurate predictors in robotics applications, such as solving the inverse dynamics problem.

3.2 ONLINE HETEROGENEOUS ENSEMBLES OF EXPERTS

In this section, the proposed methods to realise ensembles of different types of predictors are illustrated. The ensemble models presented are learned by a robot from data only in an online fashion. The learned models provide predictions of the multimodal sensory state of the robot, given the current states and motor commands. Hence, they are effectively forward models for the robot.

3.2.1 *Model formulation*

The formalisation of the proposed method is general and not confined to specific robotic platforms. Denote the sensory state vector containing different sensory modalities (note that some modalities can be multidimensional) as $x = [x_1\ x_2\ \dots\ x_N]^T$ and the motor commands applied to the M motors of the body part as the vector $u = [u_1\ u_2\ \dots\ u_M]^T$.

Following seminal works such as (Wolpert and Kawato, 1998; Jordan and Rumelhart, 1992) and (Maye and Engel, 2011), a sensorimotor representation mapping is formulated as a system that can be described by the equation

$$x' = f(x, u) \tag{3.1}$$

where x' is the sensory state after applying the motor command u from state x. The unknown map f is learned incrementally throughout the experience accumulated by the robot itself. The sets where x and u take values are denoted as $\mathcal{X}$ and $\mathcal{U}$, respectively.

Exploration steps are performed to acquire a set of data points used to learn online an estimate of the sensorimotor representation mapping f, used as a forward model that predicts the next sensory data x', given the current state x and a motor command u. To learn this forward mapping, a new method is proposed based on an online ensemble of different online regression algorithms. This method allows to obtain accurate predictions by leveraging the properties of the ensemble strategy adopted.

3.2.2 *Online heterogeneous ensemble*

In this section, an Online Heterogeneous Ensemble (OHE) is presented. The method illustrated is novel to solving online regression problems using an ensemble of predictors of different types.

THE ENSEMBLE LEARNING STRATEGY: Main ingredients in ensemble learning methods are ensemble generation and ensemble integration. *Ensemble generation* refers to the generation of base models (or experts), where the objective is to build a set of N_e base models, also called *pool of models* ($\mathcal{F}_{N_e} = \{\hat{f}_i,\ i = 1, \ldots, N_e\}$), to approximate a true function f. If the models in $\mathcal{F}_{N_e}$ are all generated using the same induction algorithm, the ensemble is called *homogeneous*, while if more than one algorithm is used to build $\mathcal{F}_{N_e}$, the ensemble is *heterogeneous*. Less work exists on heterogeneous ensembles than in homogeneous ones (Mendes-Moreira et al., 2012), however, combining

different algorithms is a promising strategy to obtain diversity, which has been shown of great importance in enhancing prediction accuracy (Kuncheva and Whitaker, 2003).

The ensemble integration step can be realised in a number of different ways. A common solution is to compute the ensemble estimate by taking the weighted average of the base models: $\hat{f}_E = \sum_{i=1}^{N_e} w_i \hat{f}_i$, where $w_i \in [0, 1]$, $\sum_{i=1}^{N_e} w_i = 1$, are the weights assigned to each base model $\hat{f}_i$. The weights state the importance of the single base models in building the ensemble, according to some application-dependent or optimality criterion. The weights can be constant or dynamically calculated according to each data sample. Popular algorithms to obtain ensemble weights are stacked regression (Breiman, 1996b) and dynamic weighting (Rooney et al., 2004). Given a learning set $\mathcal{L}$ with K data samples, the stacked regression approach calculates the weights by minimising $\sum_{k=1}^{K} [f(\mathbf{x}_k) - \sum_{i=1}^{N_e} w_i \hat{f}_i(\mathbf{x}_k)]^2$, while the dynamic weighting method sets the weights according to performance measurements of the predictors.

POOL OF MODELS: In this thesis, a heterogeneous online ensemble learning method is proposed. This method combines predictors of different types in an online manner. The following four algorithms (see Fig. 3.1) have been considered to populate the pool of models:

1. Echo State Networks (ESN) (Jaeger, 2002), which are a class of recurrent neural networks,
2. Online Echo State Gaussian Processes (OESGPs) (Soh and Demiris, 2015), which combine ESN with sparse Gaussian Processes,
3. Locally Weighted Projection Regression (LWPR) (Vijayakumar and Schaal, 2000a), which exploits piecewise linear models to realise an incremental learning algorithm,
4. Recursive ARX models (RARX) identified using the recursive least square method (Ljung, 1998; Ljung, 1983).

These four algorithms can all update in an online manner (*e.g.* incrementally or recursively). However, they differ from each other in

several aspects: firstly, while the ESN, OESGP and LWPR are non-parametric approaches, the RARX is parametric and fits the data by finding polynomial coefficients. Also, the chosen algorithms rely on different structures, *i.e.* neural networks, Gaussian processes, piecewise linear models, polynomial transfer functions. Contrary to a large number of other learning algorithms, these are online algorithms, which are able to update as new data is available, iteratively or recursively. Moreover, an advantage of using these different algorithms is that their dissimilarities guarantee the necessary diversity between the base models that constitute the ensemble, particularly in terms of prediction errors (*e.g.* overshoot vs. undershoot, offsets).

The base models are trained separately and in parallel, and the update step is different for each of the diverse models. In the ESN model, only the output weights ($\mathbf{w}^{out}$) of the recurrent neural network are updated, and the prediction is obtained by computing $\tanh(\mathbf{w}^{out}x(t))$ (Jaeger, 2002). The OESGP model's prediction is obtained through the Gaussian predictive distribution $\mathcal{N}(\mu, \sigma^2)$, where the mean μ and the variance σ^2 are estimated incrementally during training (Soh and Demiris, 2015). The RARX model updates the parameter estimates $\hat{\theta}$ at each iteration, while the prediction is calculated as $\psi^T(t)\hat{\theta}(t)$ where ψ represents the gradient of the predicted model output (Ljung, 1983). The LWPR model updates local models' parameters by minimising a predicted residual sums of squares function and then produces an estimated $\hat{f}$ as a weighted combination of local models, (Vijayakumar and Schaal, 2000a). A more detailed description of each model is presented in Appendix A.

Multiple models for each learning algorithm can be instantiated by using different initialisation parameters: for example, different numbers of internal units in the ESNs, different length scales for the Gaussian distributions in the OESGPs, different orders for the polynomials of the RARX models and different weight activation thresholds for new local models to be generated for LWPR. One of the advantages of the ensemble approach is also the flexibility allowed by the possibility of instantiating several different predictors without the need of fine parameter tuning.

WEIGHTS COMPUTATION: Each base model provides an estimate $\hat{f}_i$ of the true function f, that can be used to compute predicted values of $\hat{x}'_i$. The ensemble prediction is obtained by combining the base models' estimates through the ensemble integration. The subscript E will be used to indicate the ensemble estimates in the following. The ensemble estimate $\hat{x}_E$ is computed online at each time step t as:

$$\hat{x}_{t,E} = \sum_i w_{t,i} \hat{x}_{t,i}(t) \tag{3.2}$$

where $w_{t,i}$ are the normalised ensemble weights ($\sum_i w_{t,i}=1$). The aim is to calculate the ensemble weights so that the combination of models gives the closest estimate to the true value to be predicted.

A new method to update the ensemble weights is proposed, which takes into account both the cumulative and instantaneous base models' performance, so that both the overall behaviour and the accuracy at the last data point available are used to evaluate the base models.

At each time step t, that is for each new data point available, each base model i produces an estimate $\hat{x}'_i$. The performance scores obtained are evaluated through the following two values:

$$e^c_{t,i} = \frac{1}{t} \sum_{\tau=1}^{t} \left(x'_\tau - \hat{x}'_{\tau,i} \right)^2, \quad e^s_{t,i} = \left(x'_t - \hat{x}'_{t,i} \right)^2 \tag{3.3}$$

representing the cumulative mean squared error and the instantaneous squared error at the current time step, respectively, for all the base models $i = 1, ..., N_e$.

A convex combination of the instantaneous and cumulative errors is then used to compute the score

$$e_{t,i} = \alpha \left(e^s_{t,i} \right)^{-1} + (1 - \alpha) \left(e^c_{t,i} \right)^{-1}, \tag{3.4}$$

where α is a non-negative parameter, that can be chosen so that more importance can be given either to the inverse of the cumulative error or to the inverse of the instantaneous error.

The final ensemble weights are then computed as

$$w_{t,i} = \frac{e_{t,i}}{\sum_i e_{t,i}}. \tag{3.5}$$

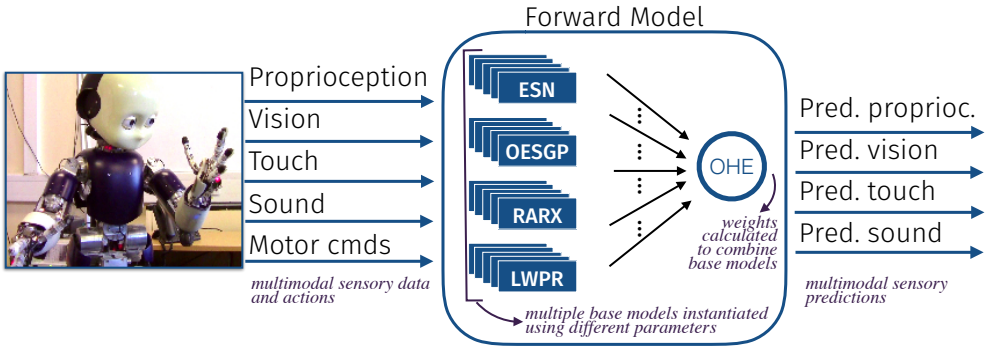


Figure 3.1: Multimodal forward model: predict the multimodal sensory state of the robot. The proposed online heterogeneous ensemble (OHE) approach (as well as the AdaNormalHedge variant) allows to learn forward models online by combining multiple learning algorithms, *i.e.* Echo State Networks (ESNs), Online Echo State Gaussian Processes (OESGPs), Recursive ARX (RARX) and Locally Weighted Projection Regression (LWPR) models.

Note that at each time step the quality of each base model is reassessed and the ensemble weights change dynamically. The final ensemble predictions are obtained by combining the base models according to the weights computes in eq. 3.5 using eq. 3.2.

A visualisation of the method illustrated in this chapter is shown in Figure 3.1. The ensemble model acts as a forward model, providing the predictions of the sensory state, given the current state and a motor command.

3.2.3 *Online ensemble learning using AdaNormalHedge with confidence-rated experts*

The computation of the ensemble weights proposed in the previous section (eq. 3.3-3.5) is very effective, as it will be shown in the experiment section. The selection of the base models using these weights is typically aggressive: very large weights are assigned to better performing models while very small weights are assigned to models performing worse. This approach may however harm the final ensemble by discarding models that could be accurate just locally.

In order to enhance the update of the forward models maintaining an online approach, a state-of-the-art algorithm has been adopted,

namely the AdaNormalHedge (Luo and Schapire, 2015). This section illustrates how the AdaNormalHedge method has been integrated with the ensemble presented in the previous paragraphs. AdaNormalHedge has been chosen among several online learning algorithms because it is a parameter-free algorithm, it does not require to know the number of experts, it presents nice properties in terms of regret bounds, and allows online update of the predictions. It also allows to mitigate the aggressive ensemble weight update proposed previously. Moreover, it can accommodate variants in order to inject confidence measures in the computation of the ensemble weights, while preserving properties in terms of regret bounds. This characteristic is leveraged and details about the proposed variant are presented in the following parts of this section.

The problem of online learning with experts is usually formulated as follows. At each time t, the player first chooses a distribution p_t over N_e experts, the the adversary decides each expert's loss $\ell_{t,i} \in [0,1]$, and reveals the losses of the player. The player then suffers the weighted average loss $\hat{\ell}_t = p_t \cdot \ell_t$, where $\ell_t = (\ell_{t,1}, ..., \ell_{t,N_e})$. The instantaneous regret to expert i at time step t is denoted by $r_{t,i} = \hat{\ell}_t - \ell_{t,i}$, the cumulative regret is denoted by $R_{t,i} = \sum_{\tau=1}^{t} r_{\tau,i}$, and the cumulative loss by $L_{t,i} = \sum_{\tau=1}^{t} \ell_{\tau,i}$. The goal of the player is to minimise the cumulative regret.

AdaNormalHedge uses a potential function $\Phi(R, C)$ and a weight function $\Omega(R, C)$, dependent on the regret and on a scale parameter C. These two functions are defined as $\Phi(R, C) = \exp\left(\frac{[R]_+^2}{3C}\right)$, with $\Phi(0,0) \triangleq 1$, and $\Omega(R, C) = \frac{1}{2}\left(\Phi(R+1, C+1) - \Phi(R-1, C+1)\right)$. The prediction step consists in setting $p_{t,i}$ to be proportional to $\Omega(R_{t-1,i}, C_{t-1,i})$ where $C_{t,i} = \sum_{\tau=1}^{t} |r_{\tau,i}|$, that is the cumulative magnitude of the instantaneous regrets up to time t. The player is allowed to have a prior distribution q over the experts, which can be simply set as a uniform distribution if no prior knowledge is available.

A generalisation of AdaNormalHedge to deal with confidence-rated experts is formulated as follows. At each time t, each expert produces a prediction and a corresponding confidence $w_{t,i} \in [0,1]$. The player then predicts p_t as usual, and still suffer the loss $\hat{\ell}_t = p_t \cdot \ell_t$. The instantaneous regret is though redefined to be $r_{t,i} = w_{t,i}\left(\hat{\ell}_t - \ell_{t,i}\right)$, that is

Algorithm 1: Final ensemble algorithm using AdaNormalHedge and ensemble weights as confidence scores

Initialize: $\forall i \in \{1, 2, ..., N_e\},\ R_{0,i} = 0,\ C_{0,i} = 0$
for $t = 1$ **to** T **do**
  Compute error scores $e^c_{t,i}, e^s_{t,i}, e_{t,i}$ (eq. 3.3-3.4)
  Compute ensemble weights $w_{t,i}$ (eq. 3.5)
  Predict $p_{t,i} \propto \frac{1}{N_e} w_{t,i} \Omega(R_{t-1,i}, C_{t-1,i})$
  Adversary reveals loss vector ℓ_t and
    player suffers loss $\hat{\ell}_t = p_t \cdot \ell_t$
  Set $\forall i \in \{1, 2, ..., N_e\},\ r_{t,i} = \hat{\ell}_t - \ell_{t,i}$,
    $R_{t,i} = R_{t-1,i} + r_{t,i},\ C_{t,i} = C_{t-1,i} + |r_{t,i}|$.
  Compute the ensemble prediction $\hat{x}_{t,E} = \sum_i p_{t,i} \hat{x}_{t,i}(t)$.

the difference between the loss of the player and expert i weighted by the confidence. The prediction step is now defined by setting

$$p_{t,i} \propto \frac{1}{N_e} w_{t,i} \Omega(R_{t-1,i}, C_{t-1,i}), \tag{3.6}$$

where the term $\frac{1}{N_e}$ indicates a prior uniform distribution among the base models. This is consistent with an agnostic approach regarding the performance of each base model.

The confidence rate plays a key role for the performance of the ensemble. It is proposed to formulate the confidence measure $w_{t,i}$ as the ensemble weights calculated as in equation 3.5. The ensemble estimate $\hat{x}_E$ is finally computed online at each time step t as:

$$\hat{x}_{t,E} = \sum_i p_{t,i} \hat{x}_{t,i}(t). \tag{3.7}$$

The final algorithm is summarised in Algorithm 1.

3.3 EXPERIMENTS AND DISCUSSION

In this section, the results achieved by the proposed methods are presented. It will be shown that the proposed methods achieve the best prediction accuracy over different datasets compared to other methods.

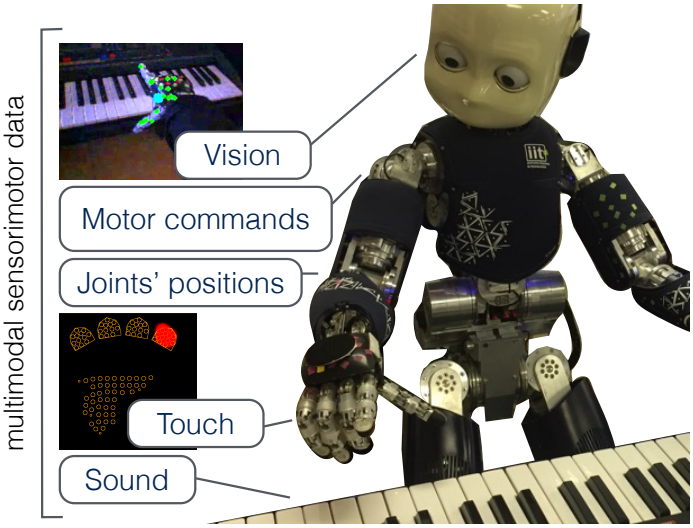


Figure 3.2: Scenario: the iCub humanoid robot interacting with a piano keyboard, learning internal models from multimodal sensory signals. The enclosed pictures show the view from the robot's left eye camera (top) and the hand taxels activated when keys are pressed (bottom).

3.3.1 *Datasets for performance evaluation*

The proposed algorithms have been evaluated on synthetic data and exploration data collected from the robots motor babbling.

Synthetic data: A synthetic dataset has been produced. To create the synthetic data, 1000 samples have been drown from a function $f(x) = \sin(2x) + 2\exp(-16x^2)$ to which Gaussian noise with variance 0.1^2 has been added (the same test data were used for example in (Meier and Schaal, 2016) to test a model performance).

iCub data: This dataset has been collected and built from a pseudo-random motor babbling of one of the arms of a humanoid iCub robot interacting with a piano MIDI keyboard[1] (see Figure 3.2).

Note that the iCub dataset used to test the proposed method is multimodal, that is it includes multiple sensory modalities used as input to the forward model. It is argued that multimodal information fosters the prediction accuracy of the learned forward model. This hypothesis is analysed later in this section.

1 MIDI is a symbolic representation of musical information incorporating both timing and velocity for each note played, which is associated to a specific integer number.

SELF-EXPLORATION ON THE ICUB ROBOT As infants in their first months perform series of gross movements of variable speed and amplitude, which lack distinctive sequencing (Hadders-Algra, 2000), analogously a robot can explore its sensorimotor representations through self-generated movements. Random sinusoidal motor commands are sent to the robot's joint actuators as velocity commands defined, for each joint j, as $u_j(t) = \alpha_j \sin(2\pi\omega t)$, where the amplitudes α_j are sampled for each joint at each cycle from a uniform distribution $\mathcal{U}(-\bar{u}, \bar{u})$, and the frequency ω is fixed so that each cycle starts and terminates at zero (*i.e.* null velocity). The choice of the velocity control compared to alternatives, such as position control, is motivated by the fact that no assumptions are made for the robot motion capabilities, *e.g.* joints' range of motion. The use of the velocity control allows to avoid encoding this prior knowledge during exploration.

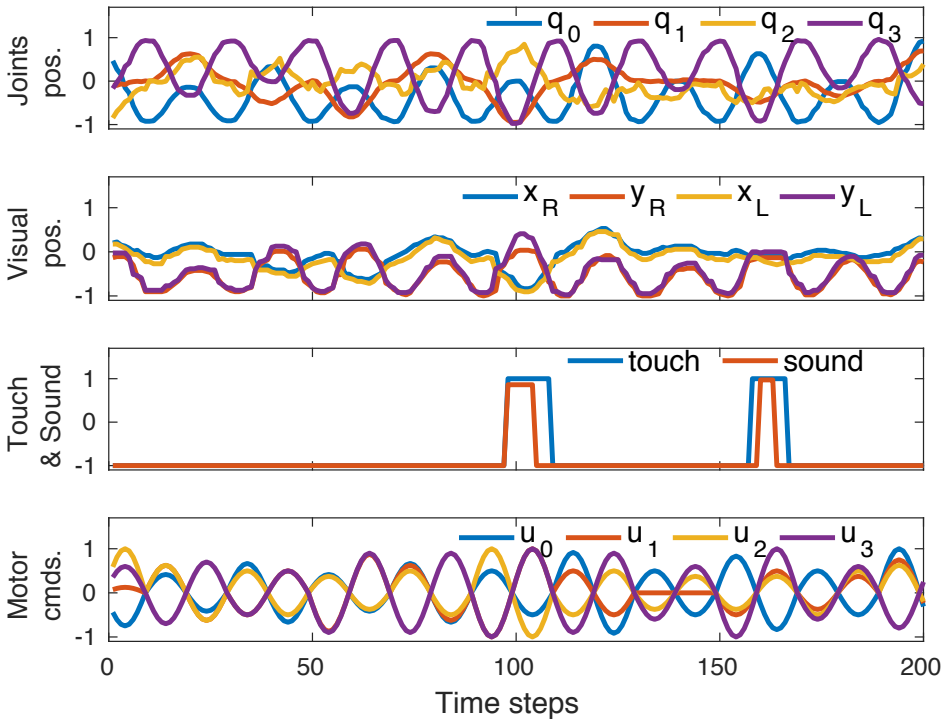


Figure 3.3: Data from self-exploration (representative examples). Joints positions $(q_0, ..., q_3)$ are recorded from the motor encoders, visual positions (x_R, y_R, x_L, y_L) are acquired from the RGB cameras of the robot's eyes. Tactile and sound signals are acquired from the skin and the keyboard, respectively. Random sinusoidal velocity commands $(u_0, ..., u_3)$ are issued to the arm joints to perform motor babbling.

Sensory feedback is simultaneously collected from different sensors of the robot:

- positions of the four arm joints (shoulder and elbow joints) from the corresponding motor encoders,
- position of the hand of the robot in the visual space from the robot's RGB cameras (4D position obtained by using the 2D positions from the two robot's eyes),
- tactile signal (one-dimensional binary data) from the robot's artificial skin present on the hand fingertips,
- musical signal from the MIDI keyboard used (one-dimensional data encoding the note played).

The dataset collected contains 7380 data points, corresponding to approximately 30 minutes. All data collected were normalised to range between -1 and 1, and representative examples of the data used are shown in Fig. 3.3. The task is to learn a model that predicts all the different sensory states at the next step, given the current sensory state and the motor commands.

COMPARISON MEASURES In order to compare the results obtained by diverse solutions the mean squared error (MSE) score has been used. Denoted $\hat{x}$ as the vector of n predictions, and x as the vector of observed values, the MSE of the predictor can be computed as $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2$. For better visualization of the results, the root MSE (RMSE) score will be also used ($\text{RMSE} = \sqrt{\text{MSE}}$).

3.3.2 *Experiments on synthetic data*

The first experiment on synthetic data has been carried out using different numbers of base models instantiated to build the ensembles. With this experiment, it is possible to examine the performance of the proposed ensemble under different conditions. Each base model forming the ensemble is initialised with different parameters, namely different numbers of internal nodes for the ESNs, different length

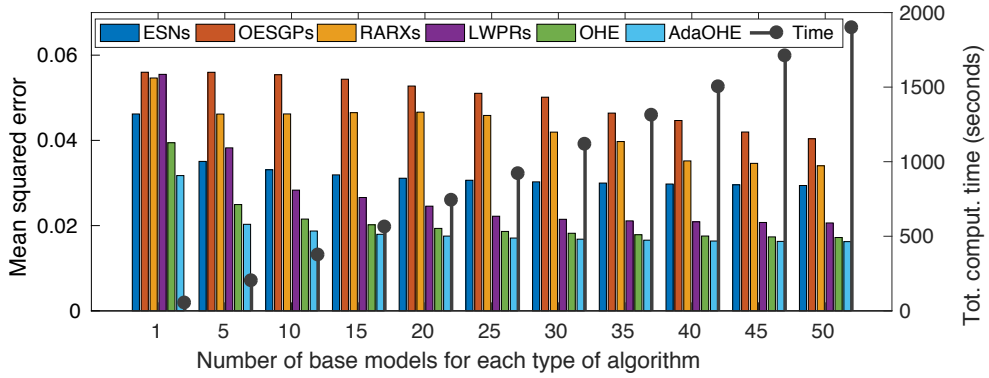


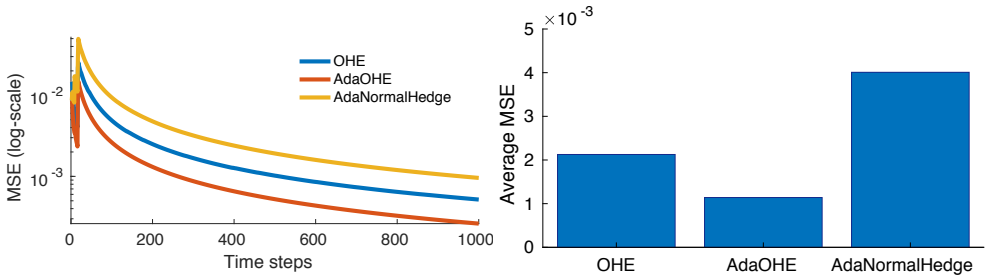
Figure 3.4: Ensemble performance on synthetic data using different numbers of base models. Coloured bars illustrate the mean squared error scores achieved by homogeneous ensembles of ESNs (blue), OESGPs (orange), RARX models (yellow), LWPR models (purple), and by the proposed OHE (green) and AdaOHE (light blue). An increase of the number of models instantiated corresponds to a decrease of the mean squared error, and to an increase of the total computational time. In all cases the proposed ensembles (green and light blue bars) outperforms homogeneous ensembles. The AdaNormalHedge variant (light blue) achieves improved accuracy compared to the proposed OHE (green).

scales for the GPs, different polynomial orders for the RARX models, and different activation thresholds for the LWPR models (details on the parameters used are reported in Appendix A). This approach allows to avoid fine tuning of the parameters, as well as providing diversity among the base models. Results are shown in Figure 3.4. In this figure, the mean squared error scores achieved on synthetic data by homogeneous ensembles and by the proposed OHE and AdaOHE are illustrated. Homogeneous ensembles are obtained by applying the same proposed ensemble methods on instances of models of the same type, thus obtaining four ensembles, one for each learning algorithm used. In this experiment, different numbers of base models have been tested: for each type of algorithm, different numbers of base model have been instantiated, namely $N_e = \{1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$. The total number of models constructing the ensemble is four times each number, that is from 4 up to 200 models instantiated. Further details on the parameters used to instantiate the base models are reported in Appendix A.

It can be noted that an increase of the number of models instantiated corresponds to a decrease of the mean squared error, and to an increase of the computational time up to even 30 minutes for the whole computation using 200 models on the 1000 points of the synthetic dataset. The proposed AdaOHE outperforms all the alternatives. In particular, AdaOHE achieves better prediction accuracy also compared with the OHE.

In order to provide evidence to support the proposed AdaOHE approach, a comparison has been conducted to evaluate the performance of the proposed OHE, AdaOHE and the basic AdaNormalHedge algorithm. The OHE acts in the most aggressive way in selecting the base models to compose the ensemble. The weights assigned to each base model following the OHE strategy strongly depend on the direct measure of the current and instantaneous mean squared errors. This results in a strong polarisation of large weights to the better performing models and of small weights to the worse performing models. The basic AdaNormalHedge algorithm, on the contrary, applies a softer selection among the models, assigning weights that are distributed more uniformly among base models. The proposed AdaOHE sits in a position in between these two approaches, balancing the two effects. Plots in Figure 3.5 show that the proposed AdaOHE outperforms the other two approaches, achieving the best prediction accuracy. The temporal profiles of the mean squared error obtained at each time step, and the average mean squared error scores achieved by each approach are depicted. In both representations, the results obtained with the AdaOHE are superior to those obtained with the OHE and the basic AdaNormalHedge.

Figure 3.6 shows the predictions and the MSE temporal profiles obtained using the proposed AdaOHE algorithm. Note that the proposed online ensemble achieves a prediction error score lower than 0.01 in less than 100 samples. Note also that the achieved performance, with a MSE score of 0.0065, is also more than two times more accurate than results obtained by using the drifting Gaussian processes reported in (Meier and Schaal, 2016), which achieved a MSE score of 0.0148 on the same task.



(a) Mean squared error temporal profiles (b) Average mean squared errors

Figure 3.5: Comparison between three ensemble strategies, namely the proposed OHE, the basic AdaNormalHedge and the proposed AdaOHE (confidence-rated AdaNormelHedge using proposed ensemble weights). (a) shows the temporal profile of the mean squared error at each time step (a logarithmic scale is used for the vertical axis to obtain a better visualisation of the results). (b) shows the average mean squared error scores for each variant. OHE acts in the most aggressive way to weight the base models of the ensembles, while the basic AdaNormalHedge acts in the least aggressive way. The proposed AdaOHE approach balances these two approaches achieving the best prediction accuracy.

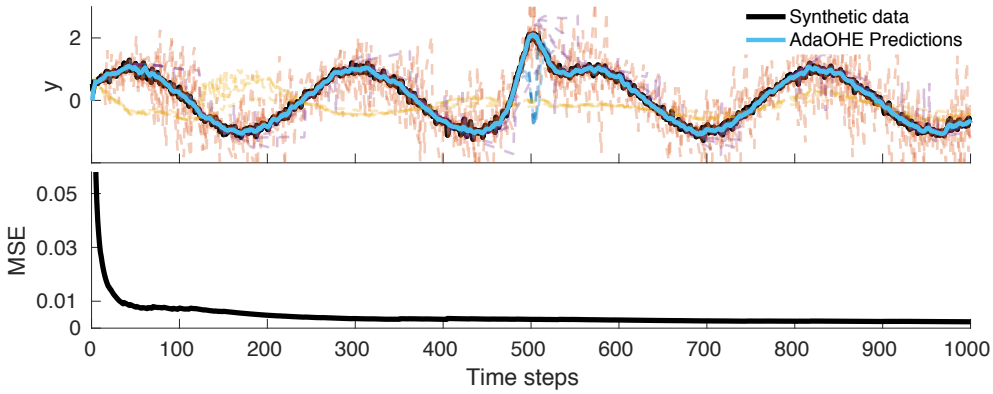


Figure 3.6: Prediction and mean squared error (MSE) temporal profile obtained on synthetic data by the proposed Online Heterogeneous Ensemble using AdaNormalHedge (AdaOHE). In the upper figure, the base models' predictions are represented with transparent dashed lines, while the AdaOHE predictions are represented by the solid light blue line. While the single base models' predictions can be inaccurate, their combination through the proposed method achieves high prediction accuracy. The bottom figure shows that the proposed method achieves a prediction error score lower than 0.01 in less than 100 iterations.

3.3.3 *Experiments on iCub data*

The proposed ensemble methods have then been tested on datasets collected from the iCub humanoid robot.

First the methods have been evaluated on the dataset consisting of data collected from pseudo-random motor babbling of the robot arms. Then the methods have been applied to predict trajectories of the robot imitating a human teaching playing a piano keyboard.

Similarly to the previous experiment, the ensemble has been tested using different numbers of base models allocated. Specifically, 1, 5, 10, 15, 20, 25, 30, 35, 40, 45, or 50 models have been allocated for each type of algorithm, leading to a total of 4 to 200 models instantiated. The models are allocated with increasing complexity: while the first base models are initialised with parameters in order to obtain a low complexity (*e.g.* small polynomial orders for RARX models, or lower activation rates for LWPR models), the base models allocated when the number of models increases are more complex models (*e.g.* higher polynomial orders for RARX models). More details about the parameters used to initialise base models are provided in Appendix A. Results are reported in Figure 3.7. Note that the proposed AdaOHE outperforms the other combinations in all cases. However, by increasing the number of models, the overall ensemble performance does not improve, due to the diminished performance of certain base models when increasing their complexity, such as the case of RARX models.

The results presented in the following part of this section refer to the combination of five base models instantiated for each type of expert. This choice has been made in light of the results presented in Figure 3.4 and Figure 3.7, so to achieve a compromise between diversity among models and computational time, while achieving high prediction accuracy.

In Fig. 3.8 the root mean squared error (RMSE) scores over the learning time steps are shown. The scores depicted in this figure are the average scores obtained over all the sensory modality dimensions, that is averaging the results obtained for the different sensory modalities. The ensemble achieves the best accuracy, that is the lowest RMSE

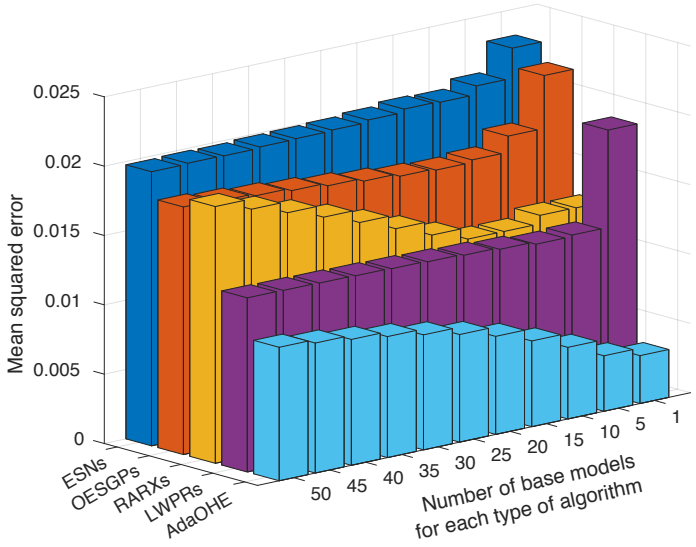


Figure 3.7: Average mean squared error scores obtained by the homogeneous and heterogeneous ensembles on the iCub motor babbling data, using different numbers of base models. The scores represent the average MSE scores over the 10 dimensions of the robot multimodal sensory space. Coloured bars illustrate the mean squared error scores achieved by homogeneous ensembles of ESNs (blue), OESGPs (orange), RARX models (yellow), LWPR models (purple), and by the and AdaOHE (light blue). The proposed ensemble outperforms alternatives in all cases. Also, increasing the number of models does not improve the ensemble performance, due to the diminished performance of certain base models when increasing their complexity, such as the case of RARX models.

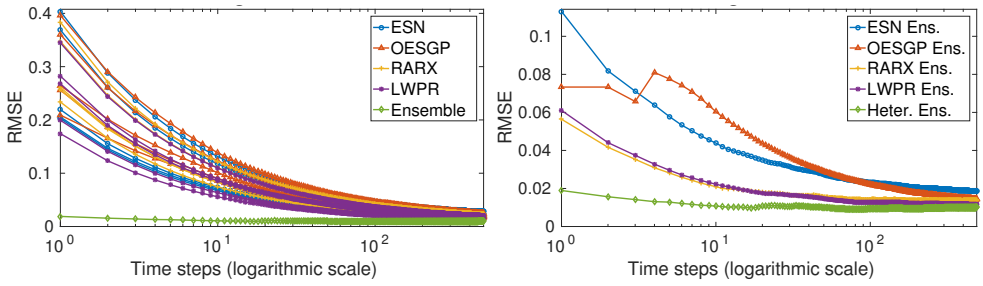
curve outperforming all the other single predictors as well as other homogeneous ensembles.

A comparison of the proposed heterogeneous online ensemble with other ensemble combinations, as well as with the single base models, has also been carried out. Results are shown in Fig. 3.9. The proposed heterogeneous ensemble scores the best accuracy compared to the alternatives not only on training data, that is on the data collected during self-exploration, but also on test datasets. Fig. 3.9b shows the RMSE scores on a validation dataset consisting of the trajectories executed during the imitation task on the piano keyboard (presented in Chapter 4). These results not only confirm the trend observed on training data, but also show that the learned forward models are able to generalise on novel uncharted gesture executions. Fig. 3.10 shows

the predicted trajectories (dashed) in the proprioception, vision and touch spaces. Accurate results are achieved on all the multimodal dimensions. In the touch space, predictions capture the overall behaviour but, compared to the other modalities, the accuracy is lower due to the binary nature of this data.

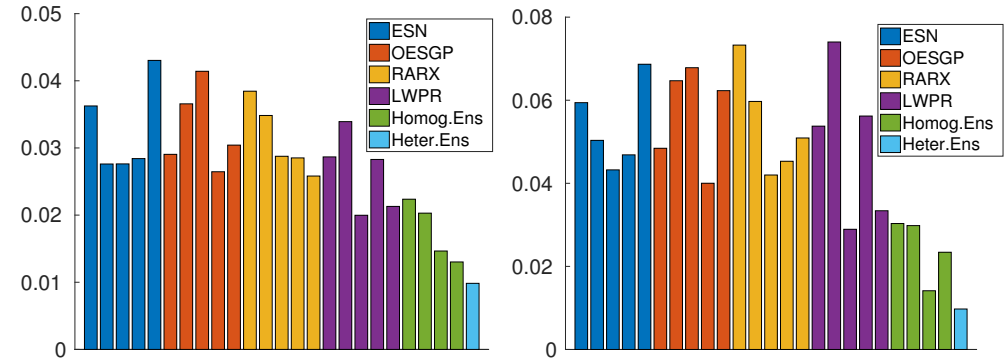
Figure 3.11 shows how the ensemble weights (computed as in equation 3.6) evolve over time. Recall that for the experiment carried out here, five instances for each type of algorithm were instantiated. The average values of the weights assigned to each type of algorithm over the learning period are shown in the figure. Higher weights are assigned to better performing base models. For example, the LWPR models are here assigned higher weights on average, compared to other models, coherently to the results shown in the previous figures. All models contribute to the ensemble, since none of the ensemble weights is null.

The proposed ensemble has been compared with other methods, including bagging trees (Breiman, 1996a), random forest (RF) (Breiman, 2001), and artificial neural networks (ANN) (Fabisch, A. Kassahun, Y. Wöhrle, H. Kirchner, 2014). The results obtained on this task show that



(a) Comparison with all base models (b) Comparison with homogeneous ens.

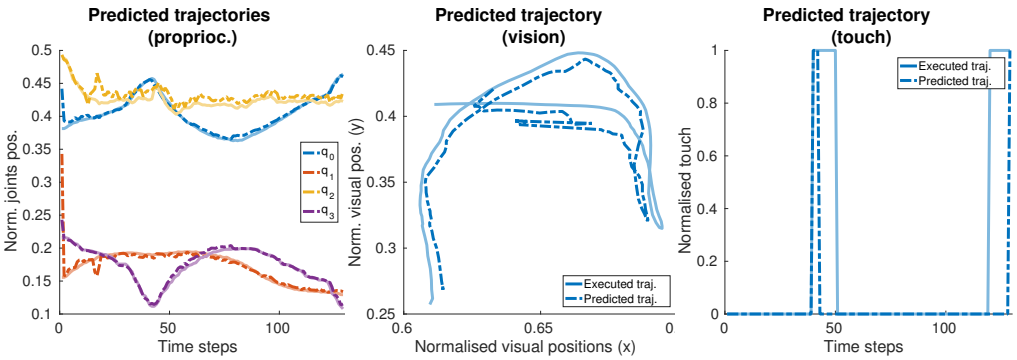
Figure 3.8: Average root mean squared error (RMSE) scores over iterations (logarithmic scale is used for the horizontal axis). (a) shows the comparison between all the base models (lines of the same colour correspond to the results obtained with base models of the same type) and the proposed heterogeneous ensemble. (b) shows the comparison between the homogeneous ensembles and the proposed heterogeneous ensemble. The ensemble predictions are more accurate than predictions obtained using all other alternatives. The error also decreases monotonically as more data are acquired over time.



(a) Comparison with all base models on exploration (training) data

(b) Comparison with all base models on imitation (test) data

Figure 3.9: (a) RMSE scores (average over different multimodal dimensions) obtained using single base modes (5 instances for each learning model), homogeneous ensembles (using the ensemble method on models of the same type: in the order ESN ensemble, OESGP ensemble, RARX ensemble and LWPR ensemble), and the proposed heterogeneous ensemble (light blue). The proposed heterogeneous ensemble scores the best accuracy compared to the alternatives. (b) RMSE scores on test data: the data for this validation experiment consists of the trajectories executed during the imitation task on the piano keyboard.



(a) Proprioception space

(b) Vision space

(c) Touch space

Figure 3.10: Predicted (dashed) and executed (solid) trajectories during the imitation task. The ensemble predictions are accurate on unforeseen data.

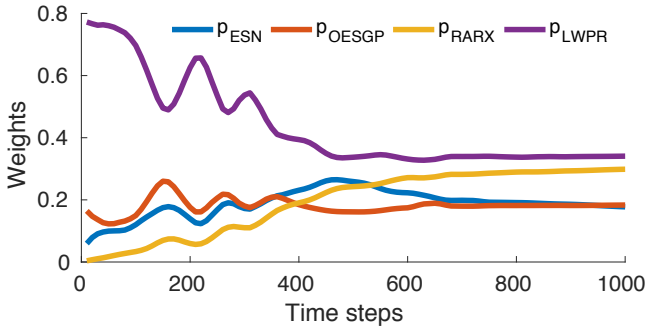


Figure 3.11: Ensemble weights (see eq. 3.6) over time steps. In the experiment, five instances for each type of algorithm were instantiated. The average values of the weights assigned to each type of algorithm over the learning period are shown. Higher weights are assigned to better performing base models (*e.g.* LWPR models are here assigned higher weights on average, compared to other models). All models contribute, since none of the ensemble weight is null.

Table 3.1: Predictive performance on iCub data, comparison with other methods. MSE scores are shown. The proposed ensembles outperform alternatives on all the data dimensions. Also, the AdaNormalHedge variation achieves improved accuracy compared to the proposed OHE.

Modality	Bagging	RF	ANN	OHE	**AdaOHE**
q_0	0.0089	0.0148	0.0136	0.0017	0.0016
q_1	0.0078	0.0101	0.0091	0.0027	0.0027
q_2	0.0161	0.0262	0.0231	0.0073	0.0068
q_3	0.0105	0.0188	0.0218	0.0023	0.0022
x_R	0.0088	0.0113	0.0102	0.0066	0.0028
y_R	0.0078	0.0178	0.0139	0.0072	0.0037
x_L	0.0042	0.0123	0.0114	0.0039	0.0024
y_L	0.0075	0.0157	0.0128	0.0075	0.0040
touch	0.2071	0.3165	0.1864	0.1791	0.1763
sound	0.3524	0.3681	0.2175	0.2113	0.2034

the proposed method achieves the best predictive performance on the multimodal dataset, outperforming other methods. Quantitative results are summarised in Table 3.1. The mean squared error scores of the predicted sensory data have been reported. Note that the tactile and the sound signals are the most difficult to predict. This is due to the fact that while all the other modalities are continuous signals, the touch and MIDI signals are discrete (binary in the case of the tactile signals). Also, these two signals are the least accurate among the recorded data, due to the little reliability of the tactile sensors, and to the delays and noise present in the MIDI data. For example, some of the tactile sensors may blink during the experiment even without any actual touch event. Similarly, the MIDI information acquired from the keyboard may be inaccurate with respect for example to the timing of the data recorded compared to the actual execution. These problems cause misleading signals that make predicting the touch and sound discrete trajectories given the current robot state and velocity motor commands significantly more difficult compared to the case of the other modalities. More details about the data and the experimental set-ups are provided in Appendix C.

3.3.4 *Multimodality influence on prediction accuracy*

To test the hypothesis that multimodal data favour the prediction accuracy in the task of learning forward models, results obtained by performing predictions of the visual and tactile modalities while including different numbers and types of inputs have been compared.

Prediction of visual data: The following cases have been compared:

1. learning a forward model for the x visual coordinate taking as input the motor commands together with the current position of x only;

2. learning a forward model for the x visual coordinate taking as input the motor commands together with the positions of the robot's joints and the current position of x only;

3. learning a forward model for the x visual coordinate taking as input the motor commands together with the positions of the robot's joints and the current position of x, y.

The MSE scores obtained for the three cases were 0.0072, 0.0066 and 0.0032, respectively, showing an improvement of the predictive performance of the learned forward model when multimodal data are taken into account. The analogous cases of forward models for the y visual coordinates have been tested. For the y coordinate, MSE $= 0.0074$, MSE $= 0.0068$, MSE $= 0.0040$ have been obtained for the three cases, respectively. These results confirm those obtained for the x coordinate. It is thus possible to conclude that the predictive accuracy is enhanced when using signals from multiple sources.

Prediction of touch data: The following cases have been compared:

1. learning a forward model for the tactile data taking as input the motor commands together with the current tactile signal only;

2. learning a forward model for the tactile data taking as input the motor commands together with the positions of the robot's joints and the current tactile signal only;

3. learning a forward model for the tactile data taking as input the motor commands together with the positions of the robot's joints, the current tactile sensation, the current visual position and the sound data.

The third case was already implemented in the previous experiments, and the corresponding MSE is 0.1763 (see Table 3.1). The MSE scores obtained for the first and second cases were 0.1899 and 0.1859, respectively, demonstrating that using multimodal data has a positive impact on the predictive performance of the learned forward models.

3.4 SUMMARY

In this chapter, a method to learn a forward model leveraging multiple sources of information in an online manner has been presented.

It has been shown that the combination of multiple predictors fosters the prediction accuracy of the forward model learned.

Results obtained on the different datasets have shown that the proposed heterogeneous ensemble of experts outperforms other algorithms in prediction accuracy. This is important in order to learn forward models that are accurate and that can be used effectively in complex control and learning architectures.

It is interesting to note that the implementation of the AdaNormalHedge algorithm to combine the predictors of the ensemble allows to achieve even improved accuracy. This result is supported by properties analysed and proofed in detailed in (Luo and Schapire, 2015), such as the regret bounds, which guarantees that the algorithm cannot do worse than the single best component. Also, importantly this method is parameter free, thus suitable for the "sleeping expert problem", a case where the number of experts is not defined *a priori*.

Furthermore, because the proposed solution is based on an online learning procedure, the learned forward models are continuously updated as new data are available from multiple different sensors present on the robot. This allows to obtain a system that continuously adapts to new data, which might come for example from the exploration of different areas of the reachable space of the robot.

Finally, the proposed system can easily incorporate multiple different modalities, so that data from different sensors can play a role in the learning of the robot's forward models. The results presented show that the multimodal nature of the input data can improve the accuracy of the prediction of the forward models, highlighting another important aspect of the proposed method.

CONTRIBUTIONS The main contributions illustrated in this chapter have been published/presented in:

- M. Zambelli and Y. Demiris, "Online Multimodal Ensemble Learning using Self-learnt Sensorimotor Representations", IEEE Transactions on Cognitive and Developmental Systems, vol. 9(2), pp.113-126, 2016.

- M. Zambelli and Y. Demiris, "Online Ensemble Learning of Sensorimotor Contingencies", Workshop on Sensorimotor Contingencies for Ro-

botics, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015.

CHAPTER 4

Multimodal Imitation using Self-learned Representations

This chapter addresses the second research question:

> Can multiple sensory modalities be integrated and combined in the learning process of inverse models and in imitation tasks, and how?

An introduction on inverse model learning and multimodal learning is presented in Section 4.1, together with a summary of the relevant background. The solution proposed to implement an inverse model which integrates different sensory modalities to achieve multimodal tasks is then illustrated in Section 4.2. The multimodal inverse model is the second block of the learning architecture anticipated in Chapter 1. This model is linked to the forward model presented in Chapter 3: the new motor commands produced by the inverse model can feed the forward model to realise mental simulation and prediction of the robot's own actions.

4.1 INVERSE MODEL AND MULTIMODAL LEARNING

An inverse model provides actions that, given a target (goal) state and the current state, allow to reach the goal. While forward models are uniquely determined, inverse models are generally not and do not always exist. Direct inverse modelling treats the problem of learning an inverse model as a classical supervised learning problem (Jordan and Rumelhart, 1992). Other methods to learn inverse models rely on reinforcement learning (Andrew Bagnell, 2014; Stulp and Sigaud, 2013; Kormushev, Calinon and Caldwell, 2013; Deisenroth, Fox and Rasmussen, 2015; Levine, Wagener and Abbeel, 2015; Mordatch et al., 2016). Another popular approach is learning by demonstration (Billard et al., 2008; Billard, 2001; Calinon, Guenter and Billard, 2007; Hayes and Demiris, 1994), which rely on the acquisition of target trajectories, that can be collected from multiple demonstrations. With the goal of reproducing a trajectory of a human motion, motion capture systems, kinaesthetic and teleoperation have often been used (Tidemann, Öztürk and Demiris, 2009; Schmidts, Lee and Peer, 2011). However, manual design of the system and usually a certain number of demonstrations are required by these approaches. Motor babbling and self-exploration, as opposite, are bottom-up approaches (Demiris and Dearden, 2005; Mochizuki et al., 2013), which require less prior design and leverage the advantages of a developmental approach to learning, such as more autonomy, incremental learning and adaptability to new conditions.

MULTIMODAL LEARNING AND IMITATION In this chapter, the focus is put on the integration of multiple sensory modalities to acquire sensorimotor representations from self-exploration data. These representations are then applied as inverse models which can be used to achieve multimodal imitation tasks, such as playing a piano keyboard. Multimodal learning leverages the different sensing capabilities of robots to build a more complete and effective representation of their structure. Recent work has been developed in this direction, *e.g.* in (Schillaci, 2013; Baraglia et al., 2015; Copete, Nagai and Asada, 2016; Roncone et al., 2014; Yoshikawa et al., 2002; Fuke, Ogino and Asada,

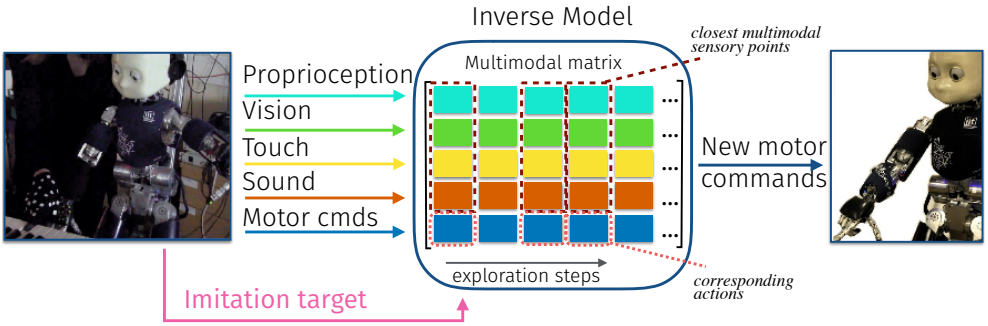


Figure 4.1: Multimodal inverse model: multiple modalities describe the sensorimotor state of the robot. Imitation targets are demonstrated by a human. The multimodal inverse model generates a new motor command to meet concurrent multimodal task constraints using multimodal sensorimotor representations. Multimodal representations are stored as a multimodal matrix, where each column corresponds to the information relative to the execution of an exploration step. The multimodal sensory points that are closest to the imitation target can be selected from the columns of the multimodal matrix, and the corresponding actions can be combined to generate new motor commands.

2007; Martinez, Lungarella and Pfeifer, 2008). Multimodal learning is a powerful strategy to enhance robot learning, in particular in the application to imitation. Imitation is considered a fundamental part of learning in humans and has been successfully used as a mechanism of learning in robots.

4.2 INVERSE MODEL USING MULTIMODAL DATA

The objective of this chapter is to show how information from multiple modalities can be integrated in learning an inverse model. A visualisation of the method that will be illustrated in this chapter is shown in Figure 4.1. The approach adopted here follows the same approach presented in the previous chapter: learning from self-exploration. This is achieved by letting the robot explore its multimodal sensorimotor space in order to collect data that can be used to learn multimodal representations and internal models.

4.2.1 *Model formulation*

Inverse model learning can also be reformulated as the problem of learning the motor commands required to achieve a target state x^* given the current state. With this formulation, it is thus straightforward to associate the inverse model learning problem with imitation tasks. The mathematical formalization of the method proposed here is general, and not specific to particular robot or sensors set-ups. Hence the method is illustrated using a general notation in the following.

Consider multiple modalities, yielding a state space of dimension N. The data collected during n self-exploration movements can be stacked to form a $N \times n$ *multimodal sensory matrix*

$$S = \begin{bmatrix} S_1 \\ \vdots \\ S_N \end{bmatrix} = \begin{bmatrix} \Delta_{x_1,1} & \cdots & \Delta_{x_1,n} \\ \vdots & \cdots & \vdots \\ \Delta_{x_N,1} & \cdots & \Delta_{x_N,n} \end{bmatrix}, \tag{4.1}$$

where $\Delta_{x_\nu,i}$ represents the change of the sensory state x_ν that has been observed during the execution of the exploration movement i. More specifically $\Delta_{x_\nu,i}$ encodes the change of the sensory state relative to the starting position, due to the application of the motor command u_i. Analogously the motor commands used to perform n self-exploration movements can be stacked to form a $J \times n$ *actuation primitives' matrix*

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_J \end{bmatrix} = \begin{bmatrix} u_{1,1} & \cdots & u_{1,n} \\ \vdots & \vdots & \vdots \\ u_{J,1} & \cdots & u_{J,n} \end{bmatrix}. \tag{4.2}$$

Target trajectories defining an imitation task can be expressed as reference trajectories (functions of time) $r_1(t), r_2(t), ..., r_N(t)$. At each time, the imitation error ε, defined on each modality space as the difference between the reference $r_\nu(t)$ and the current state $x_\nu(t)$, is defined as

$$\varepsilon(t) = \begin{bmatrix} \varepsilon_1(t) \\ \vdots \\ \varepsilon_N(t) \end{bmatrix} = \begin{bmatrix} r_1(t) - x_1(t) \\ \vdots \\ r_N(t) - x_N(t) \end{bmatrix}. \tag{4.3}$$

At each step during the execution of the imitation task, the robot moves towards the next reference point, using a combination of the primitives explored. The velocity commands to apply to the motors in order to achieve the multimodal target, defined as the vector $u^* = [u_1^*, u_2^*, \cdots, u_J^*]^T$, is obtained as a combination of those primitives that led to sensory results which are close to the current target. The experience accumulated during the exploration and the learned sensorimotor representations can then be scanned to search for those states that are closest to the vector ε. This corresponds to a search on the rows of the *multimodal sensory matrix* S containing the sensory dimensions on which a reference/target trajectory is defined.

A range search strategy has been used to find all neighbours within a designed distance. A kd-tree (Berg et al., 2008), created from the multimodal sensory matrix S was used in order to optimize the search. The kd-tree algorithm partitions the multimodal sensory matrix S by recursively splitting points in k-dimensional space into a binary tree. The nearest neighbours of the query observation is then found by restricting the training data space to the training observations in the leaf node that the query observation belongs to. The kd-tree algorithm is particularly useful when k, which represents the number of constrained modalities, is smaller than the available dimensions, that is the number of samples n in the exploration dataset. The condition $n \gg k$ is always satisfied, thus assuring the efficiency of the kd-tree implementation.

The range search gives as result the columns of the multimodal sensory matrix S containing the closest points to the query (ε). The indices of those columns are then used to select the corresponding columns in the actuation primitives' matrix. Denote the matrices obtained by selecting the indexed columns of S and A as $\tilde{S}$ and $\tilde{A}$, respectively. The sensory states contained in $\tilde{S}$ can now be associated to the current state. To achieve this correspondence, a least square regression problem can be defined as follows:

$$\tilde{S}w = \varepsilon, \tag{4.4}$$

where w is a weighting vector. The solution of this equation gives the solution for the control problem in the task (sensory) space. The best approximate solution, also the minimum norm solution of equation (4.4), is given by $w = \tilde{S}^{\dagger}\varepsilon$, where $\tilde{S}^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of the matrix $\tilde{S}$. Since each column in $\tilde{S}$ is directly related to a particular column in $\tilde{A}$, the same vector w can be used to generate new motor commands as combinations of the primitives recorded during exploration:

$$u^* = \tilde{A}w. \tag{4.5}$$

Equation (4.5) defines the desired motor command vector as a combination of the nearest primitives previously observed through the weight vector w. Note that the desired motor command vector u^* is found without requiring access to the Jacobian or any kinematic model of the robot.

4.2.2 *Robot's multimodal data*

The approach carried out throughout this thesis is based on the robot's self-exploration of its sensorimotor space. Pseudo-random control signals, also referred to as actuation primitives (see also (Kormushev, Demiris and Caldwell, 2015)), are issued to the robot's arm joints to generate exploratory movements. These control signals that populate the actuation primitives' matrix (eq. 4.2), are designed as velocity commands, in the same way as presented in the previous chapter. Data from multiple modalities are acquired during the execution of the exploratory movements, including the joints positions from the motor encoders, the position of the hand in the vision field through the robot's eye cameras, the tactile information through the tactile sensors placed on the robot's skin, and the sound data from a MIDI keyboard. The duration of the primitives, which is the period of the sinusoidal waves designed, is denoted as D in the following. The peaks correspond to the furthest reached position, *e.g.* in the visual or proprioceptive spaces peaks correspond to the samples at $D/2$. The initial time of each primitive i is denoted as t_{0_i}.

Proprioception data: Proprioception information is acquired from the motor encoders. The positions $q_1, \ldots, q_J$ of the J joints are acquired and normalised according to each joint's limits. A $J \times n$ matrix

$$S_P = \begin{bmatrix} \Delta_{q_1,1} & \cdots & \Delta_{q_1,n} \\ \vdots & \ldots & \vdots \\ \Delta_{q_J,1} & \cdots & \Delta_{q_J,n} \end{bmatrix}, \tag{4.6}$$

is then built, where $\Delta_{q_j,i} = q_j(t_{0_i} + D/2) - q_j(t_{0_i})$ denotes the relative position of joint j from the starting point of execution of primitive i.

Vision data: The robot's eye cameras are used to acquire visual data. The position of the hand in the visual space is represented by the four-dimensional vector $[x_L, y_L, x_R, y_R]^T$ of the coordinates of the centre of the hand in the 2D image frames of the left and right camera, respectively. The centre of the hand is computed as the average of the feature points detected by using the OpenCV optical flow algorithm (Bradski and Kaehler, 2008), and then normalised according to the frame dimensions. A $4 \times n$ matrix

$$S_V = \begin{bmatrix} \Delta_{x_L,1} & \cdots & \Delta_{x_L,n} \\ \Delta_{y_L,1} & \cdots & \Delta_{y_L,n} \\ \Delta_{x_R,1} & \cdots & \Delta_{x_R,n} \\ \Delta_{y_R,1} & \cdots & \Delta_{y_R,n} \end{bmatrix}, \tag{4.7}$$

is then built, where the relative displacements of the hand coordinates from the starting point of execution of primitive i is contained in $\Delta_{x_{L/R},i} = x_{L/R}(t_{0_i} + D/2) - x_{L/R}(t_{0_i})$ and $\Delta_{y_{L/R},i} = y_{L/R}(t_{0_i} + D/2) - y_{L/R}(t_{0_i})$.

Touch data: The iCub robot's skin consists of a network of tactile sensors (*taxels*), from which tactile information is recorded. In the presented experiments, the main focus is on the hand skin, which contains 60 taxels, including the fingertips. For each taxel l of the hand ($l = \{1, 2, ..., 60\}$), a binarised pressure output can be read. Each signal

is then normalised and the average pressure sensed on the hand is calculated as $p = \frac{1}{60}\sum_{l=1}^{60} p_l$. A $1 \times n$ vector

$$S_T = \begin{bmatrix} \Delta_{p,1} & \cdots & \Delta_{p,n} \end{bmatrix} \tag{4.8}$$

is then built, where $\Delta_{p,i} = p_i(t_{0_i} + D/2) - p_i(t_{0_i})$ contains the tactile feedback (on/off) during the execution of primitive i.

Sound data: Sound information is acquired using a MIDI keyboard. MIDI is a symbolic representation of musical information incorporating both timing and velocity for each note played. In this work, the information encoding the note played has been used, so that each key pressed is associated to a specific integer number. Similarly to the touch case, a single value is associated to each primitive execution, which is the code of the note if a note was played, or zeros if not. A $1 \times n$ vector

$$S_K = \begin{bmatrix} s_1 & \cdots & s_n \end{bmatrix} \tag{4.9}$$

is then built, where $s_1, \ldots, s_n$ are normalised integer numbers encoding the note played or zeros.

4.2.3 *Combining and updating internal models*

The forward model presented in Chapter 3 and the inverse model presented in the previous sections can be combined to build a learning architecture that allows to continuously learn and update the internal models. A representation is shown in Fig. 4.2. The learning architecture proposed in this study retraces well-known learning architectures proposed for example in (Haruno, Wolpert and Kawato, 2001; Demiris and Khadhouri, 2006), in which coupled forward and inverse models are used to achieve learning and motion behaviour on robots. A comprehensive discussion on the biological plausibility of this general learning scheme has been presented in (Demiris, Aziz-Zadeh and Bonaiuto, 2014b). It is worth remarking that, although the learning architecture and the learning strategy are biologically

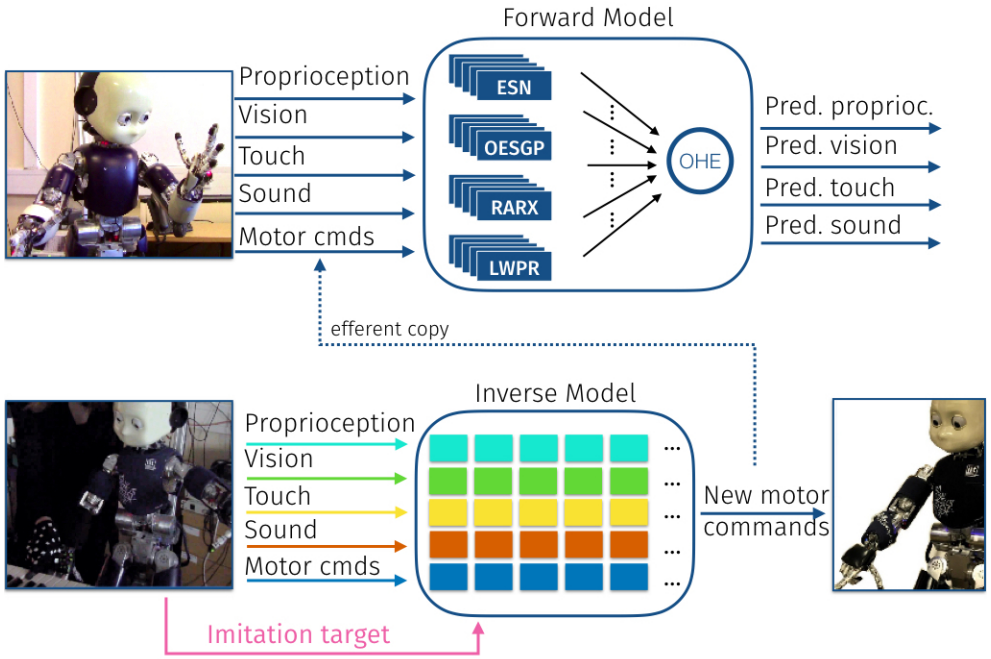


Figure 4.2: Architecture summary: forward models produce predictions of the sensory state given the current state and a motor command. This motor command can be the efferent copy of the signal calculated by the inverse model, which produces a new motor command given a target state. *(Upper part)* Online ensemble learning of forward models (see Fig. 3.1). *(Lower part)* Multimodal inverse model (see Fig. 4.1).

inspired, the proposed methods are implemented as engineered solutions that resort machine learning approaches to build forward and inverse models.

In order to refine the internal models, data points need to be acquired, either through exploration steps or from query points.

When a new control command is applied, the new reached state x_r can be observed and used to calculate $\Delta x = x_r - x$, that is the state update caused by the new experienced motor commands. The new control command u^* and Δx can then be added to the robot experience. If the error $e_r = x^* - x_r$ obtained at the reached position exceeds a predefined tolerance threshold, then more exploration is required for the robot to refine its internal models.

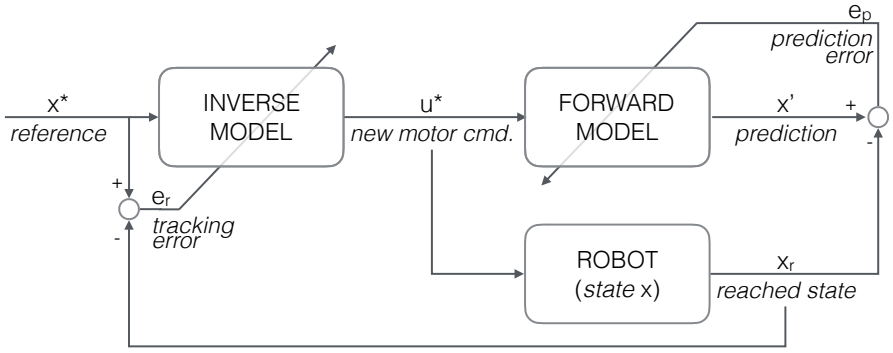


Figure 4.3: Learning architecture scheme: the inverse model is responsible for producing new motor commands u^* given a reference x^*. Motor commands are sent to the robot (characterised by the multimodal state x), and, as an efferent copy, to the forward model, which produces a prediction x' of the next state. The prediction error (e_p) and the tracking error (e_r) are then computed and used to refine the internal models.

In parallel to the execution of the calculated motor command u^*, an efferent copy of the same command is sent to the forward model, which performs an internal simulation of the action taken. The error $e_p = x' - x_r$ between the prediction obtained by the forward model and the actual reached position is then evaluated. If the prediction error e_p exceeds a tolerance threshold, then the forward model needs to be refined.

The tracking error e_r and the prediction error e_p can be used to define confidence measures for the inverse and forward models, respectively. Confidence values, C_r and C_p for e_r and e_p respectively, can be calculated as functions of the corresponding errors. Note that, since the sensory data is normalised, x, x_r, x', x^* take values in $[-1, 1]$, and $e_r, e_p \in [-2, 2]$. The confidence has been designed as a normal distribution over the error, $C \sim \mathcal{N}(\mu, \sigma^2)$, with $\mu = 0$ and $\sigma = 0.4$, so that when the error is close to zero, the confidence is approximately equal to 1 and when the error increases (symmetrically towards 2 or -2) the confidence values tend to zero. A threshold $\bar{c}$ for the confidence is set to 0.6, corresponding to a deviation of approximately 1σ.

Depending on the confidence value, more exploration might be required. For every new query point, the inverse model produces a new motor command u^*, which can be passed as efferent copy to the for-

Algorithm 2: Internal models update.

Initialise : $\kappa_r = 0$, $\kappa_p = 0$
Update :
if $C_r < \bar{c}$ **then** $\kappa_r = \kappa_r + 1$
if $C_p < \bar{c}$ **then** $\kappa_p = \kappa_p + 1$
Execute u^*
Update forward and inverse models
if $\kappa_r \geqslant \bar{\kappa}$ *or* $\kappa_p \geqslant \bar{\kappa}$ **then**
    Explore and update forward model
    Set $\kappa_r = 0$ and $\kappa_p = 0$
    Update inverse model

ward model. After the execution of u^* and after the prediction x' is obtained, both the forward and the inverse model are updated with the new data point. Meanwhile, counters κ_r and κ_p are kept to check how many times the conditions $C_r > \bar{c}$ and $C_p > \bar{c}$ are violated. If κ_r or κ_p overstep a predetermined limit, $\bar{\kappa}$, then more data points are required to refine the internal models and exploration steps are triggered. The procedure is summarised in Algorithm 2.

The state predictions provided by the forward models can be used to support the robot behaviour during imitation by giving the robot an anticipation of the tracking performance. If both the forward and inverse models are accurate enough, then the forward model can represent a considerable support to the imitation behaviour. For this reason, an important contribution to the overall behaviour performance is given by the prediction error and by the tracking error (see Fig. 4.3 for a schematic representation). In the implementation presented here, these measurements are used to trigger model refinement, according to the procedure described in Algorithm 2.

4.3 EXPERIMENTS AND DISCUSSION

The proposed approach to multimodal learning has been demonstrated on an iCub humanoid robot. The iCub first learns its sensorimotor representations models while interacting with a piano keyboard engaging vision, touch, proprioception and sound, while executing motor-babbling (see Fig. 3.2).

After exploration, a demonstrator shows the robot how to play a sequence of notes. The task assigned to the robot is to imitate the demonstrator execution based on the visual trajectory demonstrated. Touch and sound are fundamental in order to successfully play the piano keys. The task is multimodal, that is it forces constraints on different modalities, namely vision, touch and sound. In order to demonstrate the method including also the proprioception space, a constraint on proprioception has been added by fixing one degree of freedom of the arm, so that the robot is forced to execute the imitation task without actually exploiting one of the arm's degrees of freedom. This constraint can also be seen as simulating a faulty joint: the robot is required to complete the task nonetheless, while its operational space is reduced.

Experiments show that the robot is able to leverage the multimodal data acquisition and the self-learned multimodal sensorimotor representations to complete the multimodal imitation task. The stacking method used to build the multimodal matrices benefits multimodal imitation tasks by allowing requirements defined on different sensory data to be met concurrently.

4.3.1 *Experimental setup: data and exploration*

A humanoid iCub robot and a MIDI keyboard have been used (see Fig. 3.2). The exploration data are the same used in the previous chapter, and shown in Fig. 3.3. Four of the robot's arm joints have been used for motor babbling and imitation, namely the shoulder pitch, roll and yaw, and the elbow. The visual information both of the robot's motion and of the teacher demonstrations has been extracted from feature points found by using the OpenCV optical flow algorithm on the image frames acquired from the robot's on-board 2D RGB cameras (with resolution 320×240 pixel). Representative trajectories are shown in Fig. 4.4. It is worth noting that the experimental set-up, mostly concentrated on the piano keyboard and the moving hand, allows to avoid problems caused for example by different portions of arm visible in the image, or moving background. Also, the

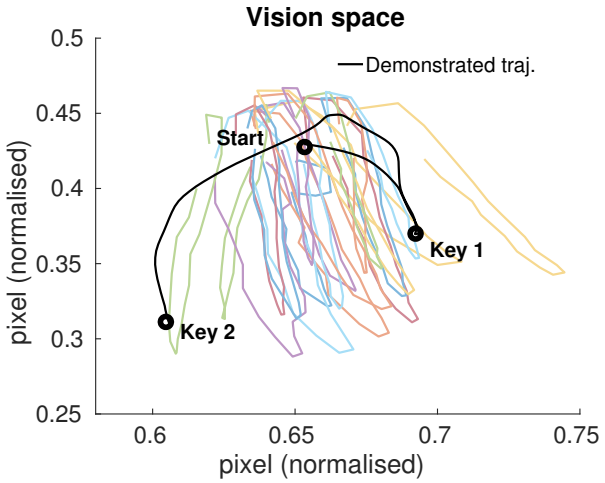


Figure 4.4: Visual trajectories: explored (coloured lines), demonstrated (black lines). The exploration is based on up-and-down motions, that correspond to movements of the robot arm aiming for the piano keys. There is a common starting point for all explored trajectories (close to the labelled 'Start' point). The demonstrated trajectory is not necessarily completely contained within the same ranges of explored visual space. The points marked on the demonstrated trajectory help visualising the sequence of movements demonstrated.

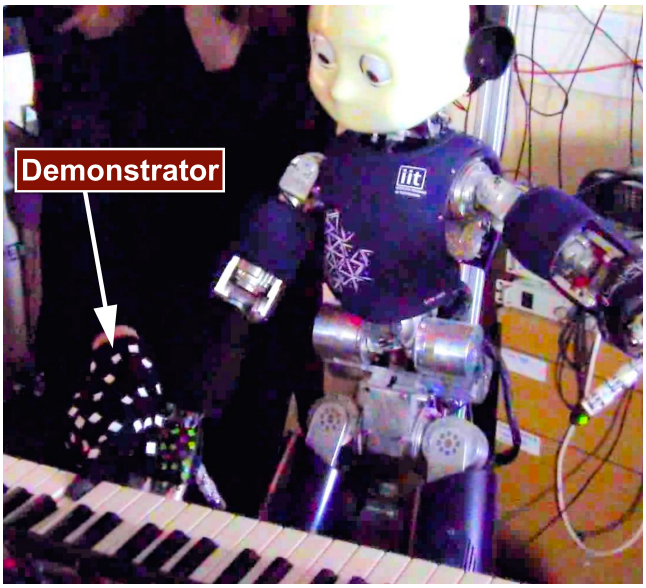


Figure 4.5: Demonstrating visual trajectories to the iCub robot. The human teacher and the iCub robot share the same point of view. The glove wore by the teacher was instrumental for feature tracking, however alternative tracking methods can be used instead.

robot and the teacher share the same point of view during the demonstrated execution (see Fig. 4.5). This is a common assumption, while alternative solutions could rely on perspective taking (Fischer and Demiris, 2016). Proprioceptive references (joint angle data) could be acquired for example from motion capture systems, as in (Schmidts, Lee and Peer, 2011; Tidemann, Öztürk and Demiris, 2009), or using more elaborate vision processing (Chang and Demiris, 2015). For the purpose of demonstrating the effectiveness of the proposed method, synthetic target trajectories have been used instead. The tactile reference is also synthetically provided, that is it is not acquired from the human demonstrator but it is designed as a piece-wise constant reference. More specifically, the target tactile reference p^* is defined as $p^* = 1$ when a key should be hit, and $p^* = 0$ during transition movements. The sound information collected during the imitation task execution is compared with the demonstrator one, so to assess if the completion of the task was successful.

4.3.2 *Multimodal imitation on piano keyboard*

The task designed for this experiment is to follow a demonstrated execution of a sequence of notes on the piano keyboard, exploiting the trajectory demonstrated on the vision space, while pressing the piano keys and without using one of the shoulder joints (constrained to remain fixed in a certain position). Note that the task can present some difficulties related to the extent to which the robot explored its sensorimotor system in the first exploration phase. This reflects in the number of times the robot touched the keyboard, in the extent of the region of the visual field where positions of the hand were registered, and, related to this, in the possibility that the demonstrated trajectory covers parts of the visual field that were not explored. Furthermore, due to the constraint forced on the proprioceptive space, the operational space of the robot's arm is effectively reduced: the robot needs to solve on-the-fly the imitation task using less degrees of freedom than the degrees available during the exploration.

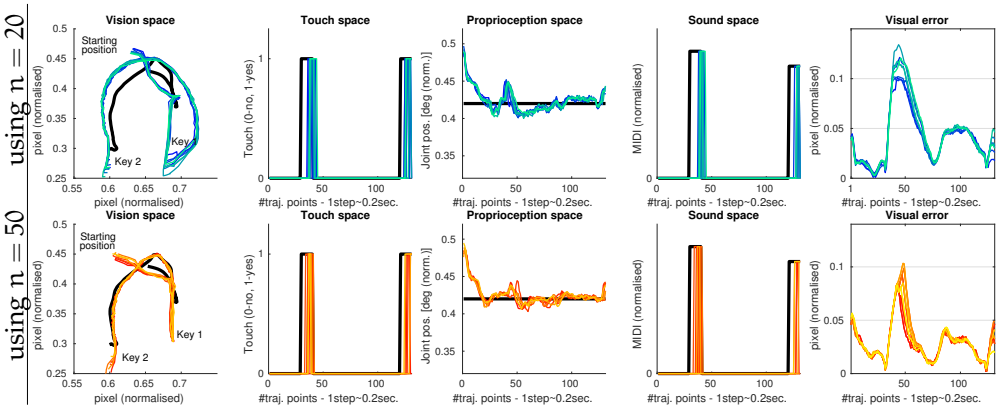


Figure 4.6: Multimodal imitation: results on the vision, touch, proprioception and sound spaces, using $n = 20$ (top row - blue coloured lines) and $n = 50$ (bottom row - yellow/red coloured lines). Reference trajectories are depicted with black lines. A constraint on the shoulder yaw is forced, so that the robot should complete the imitation while keeping it fixed to a certain position. The temporal profile of the error, evaluated as the Euclidean distance of each point to the corresponding point in the target trajectory, is depicted in the right most pictures. The improvement in the performance achieved using more exploration steps can be noted especially in the vision and proprioception cases. All figures show the results of 10 repetitions performed on the real robot.

Experimental results show that these issues can be effectively handled by the proposed method. The search on the multimodal space, rather than on single modalities, plays a fundamental role. A total of 50 repetitions of the experiment have been run on the real robot, where the iCub robot is required to imitate a demonstrator playing consequentially two notes. The success achieved, that is the successful execution of the two notes, was 45 over 50 (90%). In the failed attempts the pressure applied in order to play the piano keys was not sufficient, due to the fact that the tactile data acquired from the robot's fingertip were sometimes imprecise.

EXPERIMENTAL ANALYSIS Fig. 4.6 shows the results of the multimodal imitation task for 10 repetitions of the task performed on the real robot. It is possible to note that the robot aims at achieving a multimodal target: while following the demonstrated visual trajectory, it also moves in order to satisfy the touch modality requirement, that is

actually touching the piano keys, and also trying to avoid moving the constraint joint.

The demonstration can include any number of keys at different positions on the keyboard that is contained in the robot's visual field. However, since no prior information is assumed, the information on the sensorimotor representations, the learned models and the data used to learn them, have a notable impact on the imitation outcome. The experimental results show that the proposed method allows to effectively combine previous information to reach target points in the multimodal space, although a considerable difference can be noted between the executions obtained after $n = 20$ or $n = 50$ exploration primitives. The parameter n indicates the number of exploration primitives executed, hence the duration of the exploration phase and the number of sensorimotor samples gathered. The two values (20 and 50) have been chosen to provide representative behaviours of the imitation task on the real robot. Ten repetitions have been performed for both cases on the real robot. Note that the number of touch and sound events in just 20 primitives is considerably small (in these experiments only 6 touch events were present in the 20 primitives used). Also, the overall visual space explored in 20 primitives' executions is significantly smaller than the one obtained with 50. These two factors have a visible influence on the overall imitation performance. This effect is also confirmed by the results shown in Fig. 4.7, where the results obtained by using different amount of exploration data ($n = 20$ and $n = 50$), for 10 repetitions performed on the real robot, are reported. It can be noted that a considerable improvement is achieved by increasing the number of primitives used from $n = 20$ to $n = 50$ especially in the vision and proprioception space, while the touch space seams instead to be less influenced. In particular, the movement performed using $n = 50$ is more precise (see Fig. 4.6). In both cases, nonetheless, it is possible to note that the most critical moment in the imitation can be identified around time step 50, when the first key is reached and then the movement to reach the second key starts. The temporal profile of the vision error in Fig. 4.6 presents a maximum at this point, also because of the constraint forced on the proprioception space: note the pick of

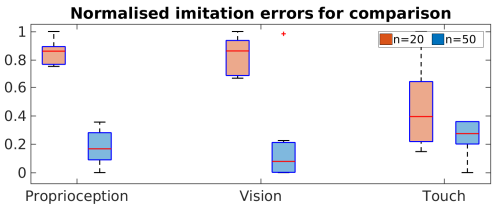
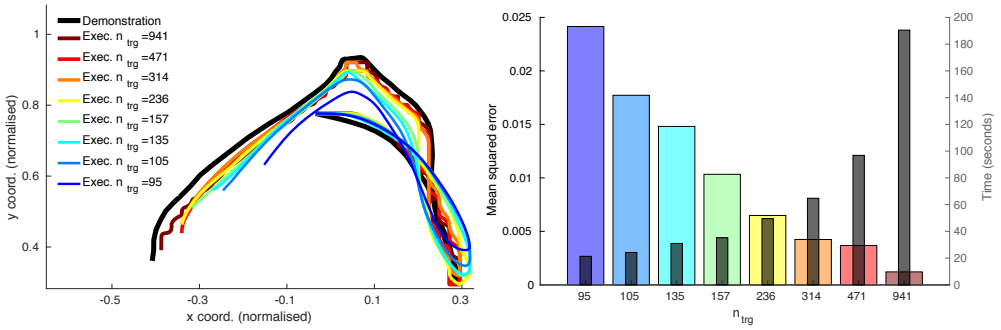


Figure 4.7: Effect of the parameter n on multimodal imitation performance. On the horizontal axis the multimodal sensory dimensions are reported, while the mean squared error scores are reported on the vertical axis.

the proprioception error in correspondence to the pick in the vision space.

The number of points defining the target trajectories (n_{trg}) also plays an important role in the imitation results. It reflects on the time taken to complete the imitation task and on the quality of the imitation: the more points are acquired, the more refined is the trajectory path, the more accurate is the tracking result, the slower the execution. The imitation obtained with increasing values of n_{trg} results in more accurate outcomes. In order to show the effect of the parameter n_{trg}, experiments have been performed on a visual imitation task. The original number of points (95 points) of the visual trajectory used as reference has been interpolated in order to obtain an incrementally finer granularity of the trajectory. Results are reported in Fig. 4.8: note that both the accuracy of the executed trajectories and the execution time increase significantly as more data points are considered. A trade-off should then be considered. In the experiments presented in this section $n_{trg} = 135$ was used.

Another important parameter to analyse is the parameter r, used to define the width of the range search on columns of S . This parameter effectively impacts on the number of column vectors used to build the matrices $\tilde{S}$ and $\tilde{A}$. It can be noted that increasing values of r might potentially cause higher computational complexity, since $\tilde{S}$ must be inverted to find the weight vector w. However, in practice, the inversion of $\tilde{S}$ is always easily computable, as the number of neighbours found remains limited. Unlike a k-nearest-neighbour search, with the range search it is possible to choose the maximum distance allowed from the query points, without the need of a prior *ad hoc* specification of



(a) Executed trajectories (b) Mean squared error and execution time

Figure 4.8: Effect of the number of target points on imitation performance. (a) Trajectories executed by the real robot using target trajectories defined by different number of points n_{trg}. The black line represents the demonstrated trajectory, while the coloured lines represent the trajectories executed by the robots. Different colours refer to different number of points used to define the target trajectory. (b) Effect of n_{trg} on the mean squared error and execution time (in seconds): the more points used, the smaller the imitation error (coloured bars), but the larger the execution time (grey bars).

a certain number of neighbours, which in turn could include vectors that are actually far from the query point. Using the range search, the number of neighbours varies for every query point. Notably, when no neighbours are found, this corresponds to a situation in which the robot has never experienced anything sufficiently similar to the query. A void search would then correspond to the robot staying still, as no motor commands could be chosen among the columns of the matrix A. This behaviour should not be seen as a limitation, instead it reflects directly the influence of the previous experience on the imitation task. Therefore, either a larger r is allowed, or more exploration should be performed. The first case would push the robot to try and combine the learned sensorimotor representations to reach for the target anyway. The second case would instead lead the robot to collect more data from self-exploration and thus refine the internal models. Also, because the number of neighbours in fact depends on the number of samples n collected in the exploration phase and the portion of the multimodal space actually explored, experimental results show that if r is chosen so that there exist neighbours almost at all times, the imitation performance does not improve sensibly by increasing r. Fig-

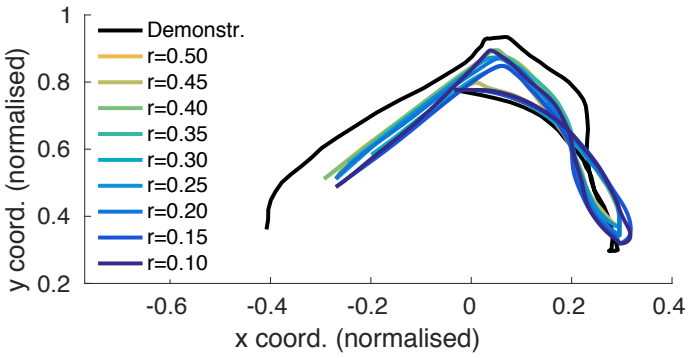


Figure 4.9: Trajectories executed to perform the imitation task on the visual space obtained by using a range of values between 0.1 and 0.5 for the range search parameter r. As long as r is chosen so that there exist neighbours almost at all times, the obtained imitation performances do not present significant improvements.

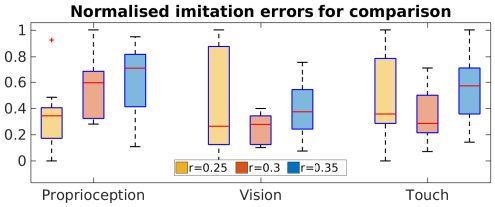


Figure 4.10: Effect of the parameter r on multimodal imitation performance. On the horizontal axis the multimodal sensory dimensions are reported, while the mean squared error scores are reported on the vertical axis.

ure 4.9 shows the results obtained by the robot imitating the visual trajectory only. Note that the performance obtained using a range of values between 0.1 and 0.5 for the parameter r does not present significant changes. The values chosen for this evaluation are so that the nearest neighbour search is not void: the minimum value used ($r = 0.1$) allows to have non-empty matrices $\tilde{S}$ and $\tilde{A}$ at all time steps, while the maximum value used ($r = 0.5$) makes the search resulting in the selection of all the columns of S. To evaluate the multimodal imitation on the real robot, three representative values have been chosen, *i.e.* $r = 0.25$, $r = 0.3$,$r = 0.35$. Figure 4.10 confirms the previous results: as long as r is chosen so that there exist neighbours almost at all times, the obtained imitation performances do not present significant improvements.

Table 4.1: Comparison with ANN and LWPR: MSE scores and standard deviation obtained on the imitation task. Best results are shown in bold.

	Vision - x coord.	Vision - y coord.	Touch
ANN	0.0053 ± 0.0047	0.0116 ± 0.0064	0.55 ± 0.24
LWPR	0.0023 ± 0.0001	0.0087 ± 0.0022	0.48 ± 0.33
Proposed	$\mathbf{0.00037 \pm 0.00004}$	$\mathbf{0.00050 \pm 0.00017}$	$\mathbf{0.121 \pm 0.054}$

SCALABILITY: Experiments have been carried out using different constraint setups in order to evaluate the behaviour of the robot on the imitation task when different constraints are forced on the proprioception space. Forcing a constant reference on the robot joints simulates the case of a faulty joint. Results are reported in Fig. 4.11, where four cases are represented: the first is obtained when no constraints are forced, the second and third when forcing constraints to one joint at time (the shoulder pitch q_0 and the shoulder yaw q_2, respectively), and the fourth when two joints (both q_0 and q_2) are forced to remain fixed. These results show that despite the task becoming increasingly difficult as more joints are constrained (the tactile space presents increasing shifts in time), the imitation task is successfully completed in all cases.

COMPARISONS: The proposed method has been compared against two well-known algorithms that have been broadly used in the framework of model learning in robotics, namely artificial neural networks (ANN) and locally weighted projection regression (LWPR). The comparison has been performed on the behaviours obtained when imitating the visual trajectory to play two keys of the keyboard while satisfying the tactile constraint (no restrictions are imposed to the arm joints). In Table 4.1 the average mean squared error (MSE) and the standard deviation over 10 repetitions of the task using the real robot have been reported. For these experiments, a network with 10 hidden layers and a LWPR model have been trained to learn the mapping between the motor commands and the target space consisting of the visual coordinates and touch. The results obtained show that the

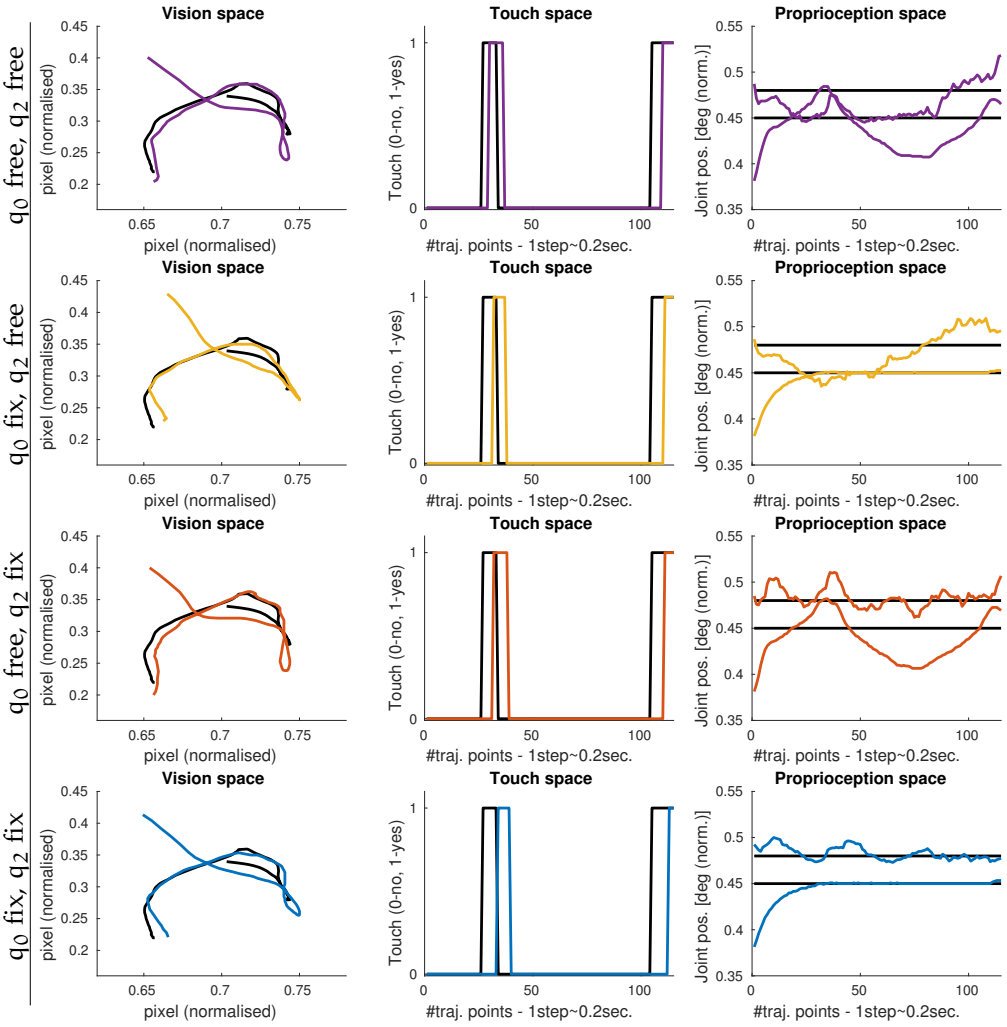


Figure 4.11: Comparison between behaviours obtained forcing different constraints on the proprioception space. These pictures show that the imitation task is achieved when no constraints are forced (first row, purple lines), as well as when forcing constraints to the shoulder pitch q_0 (second row, yellow lines), to the shoulder yaw q_2 (third row, orange lines), and to both (fourth row, blue lines). Proprioception trajectories present significant changes. Although the results on the tactile space present increasing delays when more joints are constrained, the imitation task is fulfilled in all cases.

proposed method achieves the smallest imitation errors on the target space.

LIMITATIONS: One issue that can rise at different level in the proposed architecture is how to deal with lack of information or incomplete data. The hypothesis considered in the proposed implementation is that the sampling times for all the different sensors are synchronised, that is all the information is gathered at a predefined rate (*i.e.* approximately 5 Hz), which allows to align the data collected. Concerning the implementation of the proposed inverse model used to achieve imitation, stacking unimodal data allows to include only the desired modality references while letting others free. This type of solution thus directly includes the possibility of dealing with incomplete information in terms of number of modalities considered for imitation. Finally, the problem of making predictions with missing data points is a well-known problem in many machine learning applications. A solution to tackle this issue is presented in the next Chapter.

4.4 SUMMARY

In this chapter, a method to endow robots with multimodal learning skills enabling imitation learning has been presented. The proposed method is based on multimodal sensorimotor representations which are learned during exploratory actions, and it has been shown to be effective in performing on-the-fly multimodal imitation by combining the knowledge acquired during the multimodal learning steps. The proposed approach to multimodal imitation benefits from learning sensorimotor representations using data from multiple sensors from self-exploration. The multimodal sensory matrices and the range search on a multimodal space are key aspects of the proposed method to achieve successful imitation of multimodal tasks. The formulation of the proposed method is general and allows to accommodate different modalities. Although the experiments presented have been conducted on an iCub humanoid robot, since no prior knowledge is as-

sumed on the kinematic and dynamic models of the robot, the proposed method can be applied to different robotic platforms. The proposed method is scalable and accurate. It has been shown that it outperforms neural networks and the LWPR method on the multimodal imitation task, achieving more accurate tracking performance. Overall, it has been shown how the use of multiple modalities can be used effectively in an imitation learning scenario, where a robot can learn *multimodal* sensorimotor representations.

The structure of the proposed method is simple and effective, however retaining large amounts of experience and learning flexible and generalizable multimodal representations may be difficult with the rigid matrix formulation. Also, the method proposed in this Chapter allows to achieve accurate multimodal imitation, but all the reference trajectories should be provided. However, typical scenarios would only allow a robot to access visual clues, for example when observing other agents' motions. The method proposed in this Chapter does not provide a way to directly reconstruct missing information from the one observed. These issues are resolved by the architecture presented in the next chapter, which allows to overcome the mentioned problems, and to learn more flexible and complex multimodal representations from self-exploration.

CONTRIBUTIONS The main contributions illustrated in this chapter have been published in:

- M. Zambelli and Y. Demiris, "Online Multimodal Ensemble Learning using Self-learnt Sensorimotor Representations", IEEE Transactions on Cognitive and Developmental Systems, vol. 9(2), pp.113-126, 2016.
- M. Zambelli and Y. Demiris, "Multimodal Imitation using Self-learned Sensorimotor Representations", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.

CHAPTER 5

Using Self-learned Internal Models to Predict and Imitate Others' Sensorimotor States

This chapter addresses the third research question:

> Can multimodal self-learned internal models be used by a robot to predict and imitate others' actions, and how?

An introduction on the use of self-learned models to predict and imitate others' actions and the relevant background are summarised in Section 5.1. The solution proposed to implement a self-learned model able to reconstruct missing data and generate actions is then illustrated (Section 5.2). This part builds up from the principles underling the forward model presented in Chapter 3, and the multimodal inverse model presented in Chapter 4.

The architecture presented in this Chapter allows to overcome issues related to the rigid model presented in the previous chapter, by learning flexible and complex multimodal representations as probability distributions rather than storing directly sensorimotor information. The learned distributions allow to generalize the experience accumulated through the robot's self-exploration. The architecture proposed in this Chapter is also a versatile model that allows to achieve the functions of both a forward and an inverse model.

5.1 PREDICT OTHERS USING SELF-LEARNED INTERNAL MODELS

One of the challenges that need to be addressed to allow robots to be a significant part of our everyday life is to improve the quality of their interactions with human users. Studies have revealed that the ability of humans to make predictions is not only essential for motor control, but it is also fundamental for high level cognitive functions including action recognition, understanding, imitation, mental replay, and social cognition (Wolpert and Flanagan, 2001). Robots thus need to develop a learning system which enables not only motor control and prediction of their own body, but also fosters the understanding of others' actions.

People can predict both themselves and others by using internal models (Blakemore, Decety and Albert, 2001). More interestingly, people predict others' behaviours using the same internal models that they use for predicting their own behaviour, through a process called *simulation* (Gallese and Goldman, 1998; Cruz and Gordon, 2003; Hesslow, 2012). Assuming existing similarities between agents, the internal model used to predict one's own actions can be instrumental to predict the (*visual*) consequences of someone else's actions (Pickering and Clark, 2014).

The problem of understanding trajectories of motion have been addressed in different ways, either by predicting actions of the agent (*e.g.* learning a forward model), or by predicting actions of others (*e.g.* human trajectories from images or videos) (Shotton et al., 2013; Kuderer et al., 2012; Hu et al., 2015). However, the combination and synergy of these two approaches are rarely considered in the literature.

DEEP ARCHITECTURE TO LEARN INTERNAL MODELS The main goal of this chapter is to develop an architecture to learn a model of the self that can be applied to interpret (*e.g.* predict and imitate) actions of others. The architecture presented in this chapter is based on a self-learned model, which is built, trained and updated only according to the experience accumulated by the agent. One of the major obstacles in using self-learned internal models to predict motion of others is the intrinsic difference between the available data. While the

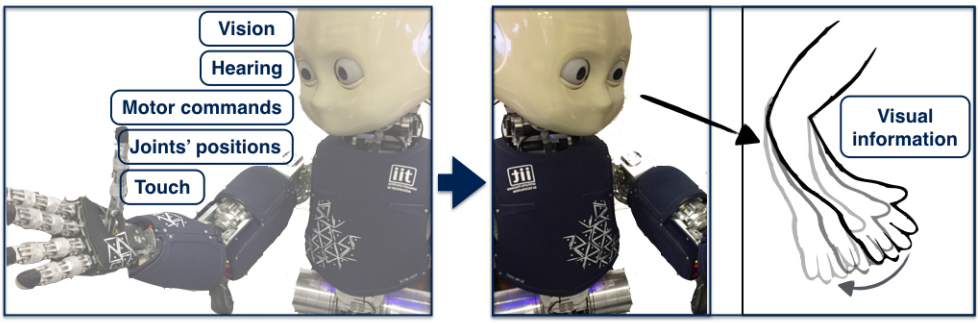


Figure 5.1: Predicting and imitating sensory trajectories using self-learned models. Internal models can be used to make predictions not only on actions performed by the agent itself, but also on actions observed from other agents. However, only visual information is available when observing someone else's motion. A multimodal deep learning approach is proposed to tackle this challenge by reconstructing the missing modalities to make predictions. The reconstructed motor information can also be used to directly control the robot, thus realizing, for example, imitation tasks.

model is learned and exploited by the agent using a whole range of available sensory modalities, only the visual information is available when observing someone else's motion. This challenge is overcome by implementing a model which is able to retrieve the missing sensory information and motor commands needed for simulation and prediction of others' motion. In this Chapter, a multimodal deep variational autoencoder is introduced. This model can be used in a versatile manner to (1) reconstruct missing sensory modalities, (2) predict the sensorimotor state of self and others, and (3) imitate the observed agent. This architecture represents a unified representation of the traditional forward and inverse models, leveraging their synergy to implement functions that are fundamental for autonomous systems. An overview of the proposed learning architecture is shown in Fig. 5.2.

A relevant study for the problem addressed in this chapter was presented in (Ngiam et al., 2011). There, a multimodal deep learning approach was proposed to cope with data of different types, such as visual and audio data, with cross-modal learning and reconstruction. The deep network that will be illustrated in this chapter, takes inspiration from the principles presented in (Ngiam et al., 2011) to deal with multimodal data, such as the strategy to train the network with

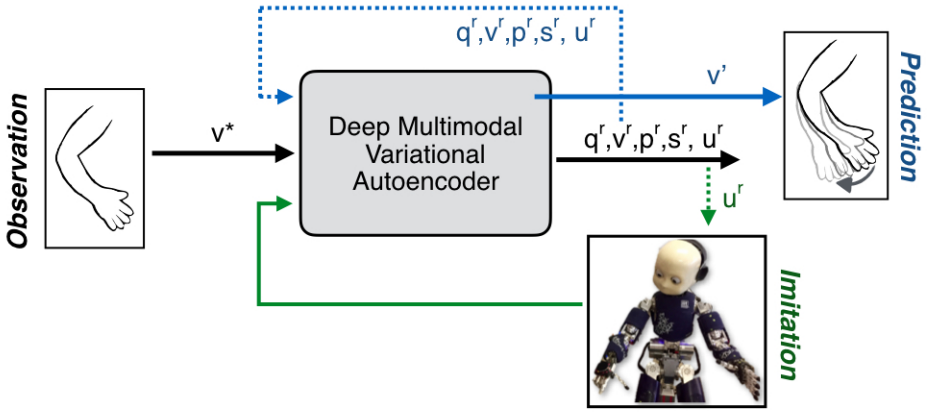


Figure 5.2: Overview of the learning architecture. The self-learned deep model can be used to (1) reconstruct missing data (black), (2) make predictions (blue), (3) control the robot's motion (green). When observing others, only the visual information is available (v^*). The model learned can reconstruct the multimodal state of the robot, including the proprioceptive (q^r), visual (v^r), tactile (p^r), acoustic (s^r) data and the motor commands (u^r). The reconstructed data can then be fed back to the network to make predictions (v'). The reconstructed motor commands can be issued directly to the robot's motor joints to perform, for example, imitation tasks.

partial unimodal data to achieve cross-modal learning and reconstruction. However, while in (Ngiam et al., 2011) only the visual and audio data were considered, in this thesis the implementation has been expanded to accommodate several different sensory modalities. In particular, the network is applied to sensorimotor modelling on physical robots, thus considering a much more complex system, in the context of developmental learning. Deep learning architectures have been exploited in the context of developmental learning to learn tasks from multiple perceptual modalities (namely proprioception, vision, audition) (Droniou, Ivaldi and Sigaud, 2015), or to predict others' goal through predictive learning (Baraglia et al., 2015; Copete, Nagai and Asada, 2016). Compared to those studies, the architecture presented in this chapter not only learns the cross-modal relationships between sensory modalities, but also retrieves missing information and uses the reconstructed data to both predict and control motion. Also, contrary to the mentioned works, in this thesis a fully autonomous exploration is used by a robot to acquire its own sensorimotor data.

Finally, the variational autoencoder that is proposed here can be seen as a more general and versatile model for robots not only to predict self and others' motion, but also to perform imitation tasks. It also presents one major advantage compared to the models presented in (Droniou, Ivaldi and Sigaud, 2015; Baraglia et al., 2015; Copete, Nagai and Asada, 2016), namely the ability to capture the redundancy of the robotic system.

5.2 MULTIMODAL DEEP VARIATIONAL AUTOENCODER MODEL

The architecture introduced in this chapter relies on a recent variant of autoencoder, named variational autoencoder (Kingma and Welling, 2013; Doersch, 2016).

Variational autoencoders (Kingma and Welling, 2013) have recently emerged as one of the most popular approaches for unsupervised learning of complex distributions of data. One of their key characteristics is that they can model the probability distribution of the reconstructed data and its distribution in the latent space. The use of variational autoencoders provides two main features: 1) like traditional (de-noising) autoencoders, the variational autoencoders is able to reconstruct missing parts of only partially observed data; 2) in contrast with traditional autoencoders, variational autoencoders are generative models that reconstruct the probability distribution of the data. These two features have been used to reconstruct the probability distribution of non-observed modalities (*e.g.* joint positions and velocities) given observed modalities (*e.g.* visual position of the end-effector). Using probability distributions is particularly important in the case of robotics applications, as it allows the system to take into account the redundancy of the system. Typically, several joint positions lead to the same end-effector position, and such relationships can be captured by the learned conditional probability distribution.

The main concept of a traditional autoencoder is to reconstruct a signal $\mathbf{y}$ provided as input by sequentially encoding and decoding it, in order to re-generate the initial signal. The encoding-decoding procedure presents several interesting properties. First, this operation can

be robust to noise or corruption in the data. Typically, if the input signal is noisy or corrupted (*e.g.* with some part of the data missing), the autoencoder can retrieve the noiseless (or original) data. Second, the encoded representation of the input signal (noted $\mathbf{z}$) is often of lower dimensionality. Thanks to this property, an autoencoder is a promising approach to reduce the dimensionality of datasets by projecting it into a latent space.

Variational autoencoders extend this concept to the domain of probabilistic distributions. In this case, the encoder determines the probability distribution $p(\mathbf{z}|\mathbf{y})$ which represents the latent distribution associated with input signal. Similarly to traditional autoencoders, this latent representation can be used to reconstruct (decode) the signal by defining the probability distribution $p(\mathbf{y}|\mathbf{z})$, which captures the potential values of the signal $\mathbf{y}$ given $\mathbf{z}$. The encoder and decoder are most of the time defined using neural networks that learn the unknown (and intractable) functions $\mathbf{z} = \text{encoder}(\mathbf{y})$ and $\mathbf{y} = \text{decoder}(\mathbf{z})$. More details can be found in (Kingma and Welling, 2013).

In this thesis, the concept of variational autoencoders is applied to multimodal sensorimotor data. A multimodal deep learning architecture has been proposed in (Ngiam et al., 2011), based on the use of Restricted Boltzmann Machines to learn a shared representation of the different modalities. The main difference that characterizes the *multimodal* deep learning approach compared to a standard deep network is that sub-networks can be used to pre-process each modality and then learning cross-modal relations through the shared hidden layers (see Fig. 5.3).

This type of variational model presents various advantages in a robotic framework. First, the ability of variational autoencoders to learn the distribution of a dataset in latent space is a powerful feature to generate a shared representation of the different modalities. For instance, the latent representation can be used to learn relationships and dependencies present in the sensorimotor experience of robots. This can be used to generate new artificial perception by sampling from the latent distribution in the latent space. Second, this shared latent representation also allows the robot to reconstruct missing modalities. For example, if data from a sensor is unavailable, this model can be

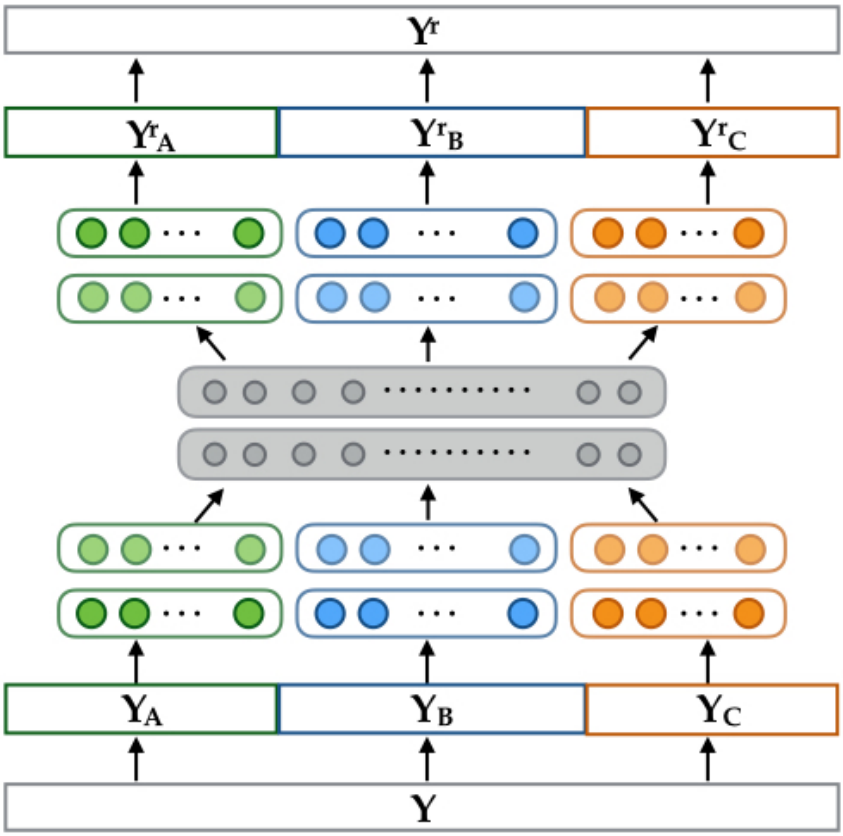


Figure 5.3: Multimodal Deep Variational Autoencoder. The input (Y) is composed by different sensorimotor data (Y_A, Y_B, Y_C), including multiple sensory modalities and motor commands. Each modality is encoded and decoded by separate autoencoders (shown with different colours). A shared layer (in grey, in the centre) allows to learn a shared representation among different dimensions. This architecture is trained with complete as well as partial data. Each uni-modal autoencoder can be trained separately, allowing for single modality learning. The cross-modality representations are also learned through the shared layer. The output of the network consists of the reconstruction of each different data part (Y^r_A, Y^r_B, Y^r_C), thus composing the reconstructed data (Y^r)

used to model the probability distribution of the data that should be observed from this sensor conditioned on the data from other sensors of the robot. Finally, their ability to predict probability distributions is fundamental to take into account the redundancy of complex robots, such as the iCub humanoid robot used in this study. With this property, the model can capture the fact that for a given end-effector position, several joint configurations are possible. These three properties are combined into a single framework for the prediction and imitation of other agents, based only on the experience of the robot.

5.2.1 *Variational autoencoder formalism*

Variational autoencoders are generative models, which combine ideas from deep learning with statistical inference (Kingma and Welling, 2013), and can be used to learn a low dimensional representation z of high dimensional data y. In contrast to standard autoencoders, y and z are random variables. The encoder is an approximation to the intractable true posterior $p_\theta(z|y)$ implemented as a multilayer perceptrons, and is represented by the model $q_\phi(z|y)$: given a data point y the encoder produces a distribution (*e.g.* a Gaussian) over the possible values of the latent variable z from which the data point y could have been generated. Analogously, the decoder is implemented as a multilayer neural network, and is represented by the model $p_\theta(y|z)$: given a latent variable z the decoder produces a distribution over the possible corresponding values of y. The variational approximate posterior is chosen to be a multivariate Gaussian with a diagonal covariance structure: $q_\phi(z|y_i) = \mathcal{N}(z; \mu_i, \sigma_i^2 \mathbf{I})$ where the mean and standard deviation of the approximate posterior, μ_i and σ_i, are outputs of the encoding multilayer perceptrons, *i.e.* nonlinear functions of data point y_i and the variational parameters ϕ. Data are then sampled from the posterior $z_{i,l} \sim q_\phi(z|y_i)$ using $z_{i,l} = \mu_i + \sigma_i \odot \epsilon_l$, where $\epsilon_l \sim \mathcal{N}(0, \mathbf{I})$, and $\odot$ indicating the element-wise product.

LOSS FUNCTION The loss function used to train this model is defined as follows:

$$\mathcal{L}(\theta, \phi, \mathbf{y}_i) = -D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{y}_i)||p_\theta(\mathbf{z})\right) + \frac{1}{L}\sum_{l=1}^{L}\log\left(p_\theta(\mathbf{y}_i|\mathbf{z}_{i,l})\right) \tag{5.1}$$

where D_{KL} is the Kullback-Leibler (KL) divergence, which can be computed and differentiated without estimation (see (Kilner, Friston and Frith, 2007) for full details). This first term on the right-hand side of equation (5.1) represents the latent loss, that is the cost associated to the encoding part building the latent space. The second term on the right-hand side of equation (5.1) represents instead the reconstruction loss, that is the cost associated to the decoding part, responsible of reconstructing the input data through sampling from the latent space.

The output distributions have been set as normal distributions, hence the term in the loss function can be written in closed form:

$$\log p(\mathbf{y}|\mathbf{z}) = \log \mathcal{N}(\mathbf{y}^*; \mu, \sigma^2\mathbf{I}), \tag{5.2}$$

where $\mathbf{y}^*$ represents the complete data (*i.e.* with all the modalities being visible, see (1) in Table 5.1), and σ and μ are the outputs of the neural network that define the normal distribution of the reconstructed data.

In order to put more emphasis on modalities with lower dimensionality, independent loss values for each modality have been computed, and the dimensionality of each modality has been used to weight each corresponding loss value. This is achieved by creating independent variational networks for each modality and corresponding independent reconstruction loss scores. Each cost is then weighted according to the number of dimensions of the corresponding modality. Finally, the sum of the costs is optimized. This approach has an interesting effect, that is it emphasizes the contribution of the worst modalities. This is due to the fact that costs are log-likelihoods, and hence summing the costs corresponds to multiplying the likelihoods. This approach in turn helps the network to learn even the most difficult parts of the state space, such as discrete or binary dimensions of the sensory space (see tactile example in Section 5.3).

5.2.2 *Training the architecture*

The training dataset contains multimodal sensorimotor data collected during a self-exploration phase (more details in Section 5.3.1). Data can be captured from different sensors of the robot, such as the position of the hand in the robot's visual space, tactile and sound data, and proprioception (joint positions) from the motor encoders. In particular, the position of the hand in the visual space is extracted by considering the centre point of a tracking window around the moving hand. All data are then normalized to take values in the range $[-1, 1]$.

Time series data from the self-exploration dataset recorded, and also used in the previous chapters, have been shown in Fig. 3.3. Denote by $\mathbf{u}_t$ the vector of velocity commands issued at time t, $\mathbf{q}_t$ the vector of joints position (proprioception) at time t, $\mathbf{v}_t$ the vector of the visual position at time t, $\mathbf{p}_t$ the tactile signal at time t and $\mathbf{s}_t$ the sound signal at time t. Note that other modalities can also be included. The input of the architecture is a multi-dimensional vector

$$\mathbf{y}_t = [\mathbf{q}_t, \mathbf{q}_{t-1}, \mathbf{v}_t, \mathbf{v}_{t-1}, \mathbf{p}_t, \mathbf{p}_{t-1}, \mathbf{s}_t, \mathbf{s}_{t-1}, \mathbf{v}_t, \mathbf{v}_{t-1}]. \tag{5.3}$$

The input vector contains both data from time t and $t-1$ to capture the temporal relationship between the different modalities.

The network is trained on both complete and partial data of the training dataset collected during the robot self-exploration. This dataset is augmented with samples that require the network to reconstruct the missing modalities given only one of them. This is realized by duplicating the dataset, while using a flag value (namely the arbitrary value -2, which is outside the range of any sensorimotor signal after normalization) to denote the non-observable modalities. The training dataset follows the structure in Table 5.1 to enable the network to perform predictions and reconstruction in multiple conditions of missing information. The different conditions will be recalled and referred to in the following paragraphs.

Table 5.1: Training dataset structure.

(1)	$\mathbf{q}_t$	$\mathbf{q}_{t-1}$	$\mathbf{v}_t$	$\mathbf{v}_{t-1}$	$\mathbf{p}_t$	$\mathbf{p}_{t-1}$	$\mathbf{s}_t$	$\mathbf{s}_{t-1}$	$\mathbf{u}_t$	$\mathbf{u}_{t-1}$
(2)	-	$\mathbf{q}_{t-1}$	-	$\mathbf{v}_{t-1}$	-	$\mathbf{p}_{t-1}$	-	$\mathbf{s}_{t-1}$	-	$\mathbf{u}_{t-1}$
(3)	-	$\mathbf{q}_{t-1}$	$\mathbf{v}_t$	$\mathbf{v}_{t-1}$	-	-	-	-	-	-
(4)	-	-	$\mathbf{v}_t$	$\mathbf{v}_{t-1}$	-	-	-	-	-	-

5.2.3 *The different usages of the architecture*

One of the major assets of the proposed model is its versatility. More specifically the possibility of using the same learned model to achieve different goals. This section illustrates how the learned multimodal deep variational autoencoder can be deployed to achieve three different functions:

1. reconstructing missing data;
2. predicting both the robot and others' sensorimotor state;
3. controlling the robot in an online control loop.

In these three cases, the training, structure, and parameters of the neural network remain the same. Details for each of the aforementioned functions that the model can achieve are given in the remaining part of the section.

RECONSTRUCTING MISSING DATA This first capability of the proposed architecture is relatively straightforward as it is the main purpose of de-noising autoencoders. For this application, the missing modalities of the input fed to the network are set to -2 (as explained in Section 5.2.2), while the network outputs the probability distribution of the reconstructed inputs.

This capability of the network is fundamental to address the problem of predicting others' motion from only visual information by relying on internal models of the self. In such an application, an agent learns internal representations of its sensorimotor space, in particular relating motor actions with multimodal sensory effects. However, when observing someone else performing an action, only the visual

information is available. The agent, which relies on full information from all its senses (*i.e.* sensors), must then be able to retrieve the missing information and interpret the observed motion in relation with its own internal representations. The architecture proposed in this chapter allows robots to achieve this by reconstructing the missing sensorimotor information (*i.e.* joint configuration, touch, sound and motor information) from observations of the visual input (see (4) in Table 5.1).

PREDICTING BOTH THE ROBOT AND OTHERS' SENSORIMOTOR STATE

People can predict others' behaviours using the same forward models that they use for predicting their own behaviour, through a process called *simulation* (Gallese and Goldman, 1998; Cruz and Gordon, 2003; Hesslow, 2012). This can be applied to robots, to enable better and smoother interactions. An underlying assumption is that there exist similarities between agents and their actions. If this assumption is true, then the model learned to predict the consequences of the robot's own actions becomes the resource to predict the (*visual*) consequences of another's action.

There exists a number of challenges to achieve the goal of having robots predict a human action by using their own internal models. The first one concerns the correspondence problem that might exist between the morphological and kinematic structures of a robot and a human user. Various techniques have been proposed to tackle this issue in contexts such as Learning by Demonstration (Argall et al., 2009; Billard et al., 2008; Atkeson and Schaal, 1997). To resolve this issue, the results obtained in (Chang and Demiris, 2015; Chang et al., 2016) on finding kinematic structure correspondences can be leveraged. Another challenge is the difference in the spatial perspective that the robot acquires of its own and of others' actions. A solution to solve this problem has been proposed in (Fischer and Demiris, 2016). In this work, it is assumed that agents share the same perspective (the same assumption is generally made in similar applications, (Baraglia et al., 2015; Copete, Nagai and Asada, 2016)), but the method developed in (Fischer and Demiris, 2016) can be applied to generalize this aspect.

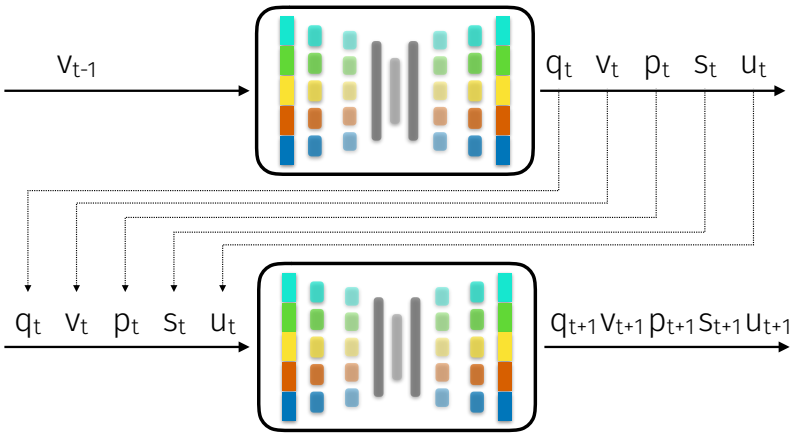


Figure 5.4: Using the learned neural network to make predictions. The reconstructed state $(\mathbf{q_t}, \mathbf{v_t}, ..., \mathbf{u_t})$ can be fed back as input of the network to predict the sensorimotor state of the agent at the next step $(\mathbf{q_{t+1}}, \mathbf{v_{t+1}}, ..., \mathbf{u_{t+1}})$. This is realised by exploiting two consecutive times the configuration (2) presented in Table 5.1.

Finally, the difference between the data that the agent has used to learn its sensorimotor representations, and the data that is available to the agent when observing others, is the last major challenge that needs to be addressed in order to achieve the goal. More specifically, while data from all sensory modalities is available to the agent when learning the models, only the visual input, from an ego-centric perspective is available when observing others. This implies that only data referring to the visual input are available in $\mathbf{y}$ (see (4) in Table 5.1).

In this respect, the reconstruction of missing modalities described above plays a key role. The neural network can act as a forward model to predict the next sensorimotor state $\mathbf{y_{t+1}}$ from the current state of the agent $\mathbf{y_t}$ (see line (2) in Table 5.1). However, when observing someone else, the current state of the agent is not fully available, as only vision information can be observed. To address this issue, the robot employs the reconstruction abilities of its model (described above) to retrieve the missing parts of the agent's state. This reconstructed state $\mathbf{y_t^r}$ can then be fed back to the input of the network to predict the current sensorimotor state of the agent at the next step $\mathbf{y_t}$. In summary, this capability enables the network to first reconstruct the current sensorimotor perceptions of the observed agent and then to use these re-

constructed perceptions to predict the next state of the agent. This approach is illustrated in Figure 5.4.

A possible different solution to predict sensorimotor states is to use the ability of the network to reconstruct missing modalities and then use them as input (efferent copy) to the learned forward model presented in Chapter 3. Hence, the visual prediction of another agent's motion, denoted by $\mathbf{v}'$, can be written as $\mathbf{v}' = f(g(\mathbf{v}^*))$, where the mapping f is the learned forward model (*e.g.* the ensemble of predictor presented in Chapter 3), and the mapping g is the multimodal variational autoencoder. More specifically, g can reconstruct the complete sensorimotor state ($\mathbf{y}^\mathbf{r}$) from just the visual information observed from someone else's motion ($\mathbf{v}^*$): $\mathbf{y}^\mathbf{r} = g(\mathbf{v}^*)$. The reconstructed signal can then be used by f: $\mathbf{v}' = f(\mathbf{y}^\mathbf{r})$.

CONTROLLING THE ROBOT IN AN ONLINE CONTROL LOOP In addition to the abilities of the architecture to reconstruct and predict the perceptions of other agents, the learned model can be used as a controller for the robot. In particular, the model can be placed in a control loop to regulate the sensory state of the robot given a target state. This approach can be used for instance in imitation learning scenarios, where the robot imitates a target trajectory. In the scenario presented above, the robot observes someone else's visual trajectory and use the learned model to replicate such trajectory.

The control loop is depicted in Fig. 5.5. Notably, the joint and visual configurations ($\mathbf{q_{t-1}}, \mathbf{v_{t-1}}$) of the robot are fed back to the network in order to provide the correct current state at each time. This prevents the network from drifting during the online cycles of the control loop, due to the dependencies between different input modalities. For example, areas of the sensory space that lie far from the training space have increased uncertainty. This condition is made more severe by the multimodal nature of the data, which come independently from diverse sensors. The feedback loop implemented to provide the network with the real current data from the robot helps preventing the accumulation of errors in different dimensions of the sensorimotor state.

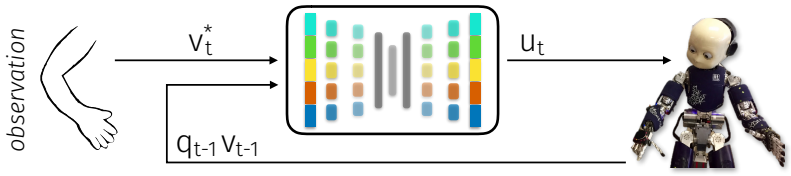


Figure 5.5: Using the learned neural network in an online control loop. The current joint and visual configuration ($\mathbf{q_{t-1}}, \mathbf{v_{t-1}}$) of the robot is fed back to the network at each time step in order to close the control loop. The network then uses the current robot configuration and the target visual position observed from other agents to reconstruct missing data, in particular the motor commands ($\mathbf{u_t}$), which is then used to control the robot motion. This is realised by exploiting the configuration (3) presented in Table 5.1.

It is also important to emphasize that using the learned network as a controller for the robot is not a trivial application, since the network itself represents a model of the robotic system. The ability of the network to produce motor commands is then key to achieve a controller behaviour, but this is not sufficient to implement an effective controller. It is important to provide the network with all the sensory information that can help the model to learn the kinematics and dynamics of the system, in particular the sensory states at two consecutive time steps. This is key for the network to build meaningful representations of the robot kinematics and dynamics, and in turn to generate sensible motor commands.

5.3 EXPERIMENTS AND DISCUSSION

5.3.1 *Experimental setup*

The proposed approach has been demonstrated using a humanoid iCub robot. The architecture is trained using data collected from the robot through experience, by performing pseudo-random self-exploratory movements (*i.e.* motor babbling). Similarly to the experiments illustrated in the previous chapters, in the scenario considered in these experiments, the robot is interacting with a MIDI keyboard (Fig. 3.2). Then, the robot uses the learned architecture to (1) reconstruct miss-

ing sensory modalities, (2) predict the sensorimotor state of self and other, and (3) control itself to imitate the observed agent.

Four joints of one of the robot's arms ($q_0, ..., q_3$) are used during motor babbling. Visual information encoding the position of the hand in the 2D visual field of the robot is acquired from the robot's eye cameras, with coordinates x_R, y_R and x_L, y_L for the right and left eye, respectively. A binary one-dimensional tactile signal is acquired from the robot's artificial skin, which consists of a network of taxels ("tactile pixels"). Sound data is also acquired from the MIDI keyboard, in the form of a one-dimensional vector containing the information related to the key played. The commands sent to the robot's motors ($u_0, ..., u_3$) to perform autonomous self-exploration (motor babbling) are velocity references, defined for each joint j as $u_j(t) = \alpha_j \sin(2\pi\omega t)$, where the amplitudes α_j are sampled for each joint at each cycle from a uniform distribution $\mathcal{U}(-\bar{u}, \bar{u})$, and the frequency ω is fixed so that each cycle starts and terminates at zero (*i.e.* null velocity). No prior knowledge is assumed on the robot's kinematic or dynamic structure. The choice of using velocity commands aims to keep this prior knowledge to a minimum by avoiding to rely on the inverse kinematic of the robot. However, the proposed method can accommodate other implementation choices, such as position or torque control. Normalization is applied to all data to obtain signals in the range $[-1, 1]$ (see Figure 3.3).

The input fed to the network is a 28-dimensional vector, including two four-dimensional joint position vectors ($\mathbf{q_t}, \mathbf{q_{t-1}}$), two four-dimensional visual position vectors ($\mathbf{v_t}, \mathbf{v_{t-1}}$), two one-dimensional tactile vectors ($\mathbf{p_t}, \mathbf{p_{t-1}}$), two one-dimensional sound vectors ($\mathbf{s_t}, \mathbf{s_{t-1}}$), and two four-dimensional motor commands vectors ($\mathbf{u_t}, \mathbf{u_{t-1}}$).

Extensive validation tests of the proposed method have been performed. Three different datasets have been used: (1) test data from the robot self-exploration, (2) data from a RGB-D camera of a human playing a piano keyboard, and (3) data from a RGB-D camera of the Imperial-PRL KSC Dataset (data used in (Chang and Demiris, 2015; Chang et al., 2016) to validate kinematic structure correspondences methods). Finally, to demonstrate the proposed method in practice, it is shown that the iCub robot is able to leverage its capability of pre-

dicting another agent's actions and to plan its own actions to imitate a human on the piano keyboard.

5.3.2 *Architecture structure*

Tensorflow (Abadi et al., 2015) has been used for the implementation of the multimodal deep variational autoencoder. The network implemented consists of five unimodal sub-networks, for the proprioceptive (joint positions), visual, tactile, sound and motor modalities, respectively. The encoders of each unimodal sub-network consist of two layers, while the decoders consist of three layers. For the proprioception, visual and motor networks, the two encoder layers consist of 40 and 20 units, respectively, and the three decoder layers consist of 40, 8 and 8 units. For the tactile and sound networks, the two encoder layers consist of 10 and 5 units, respectively, and the three decoder layers consist of 10, 2 and 2 units. The difference in the number of units is to take into account that tactile and sound data are two-dimensional vectors, while the other modalities consist of eight-dimensional vectors. The outputs of all the unimodal encoders are concatenated to feed into the shared network, which consists of a two-layer encoder with 100 and 28 units, and a two-layer decoder with 100 and 70 units, respectively.

A comparison among different structures of Multimodal Deep Variational Autoencoders (MDVAE) has been carried out to evaluate alternatives. Parameters of the different structures tested are reported in Table 5.2, and results are shown in Fig. 5.6. This figure shows the mean squared error scores obtained on each sensorimotor modality for each structure. Each colour corresponds to one of the sensorimotor modalities, with darker tones corresponding to the results obtained when reconstructing with complete data available, and lighter tones corresponding to the results obtained when reconstructing with only visual information available. Note that no variation is registered for the visual data, since this information is available in both tests. Note also that the lowest accuracy is obtained for the discrete tactile and sound data: not only the absolute error scores obtained in the two

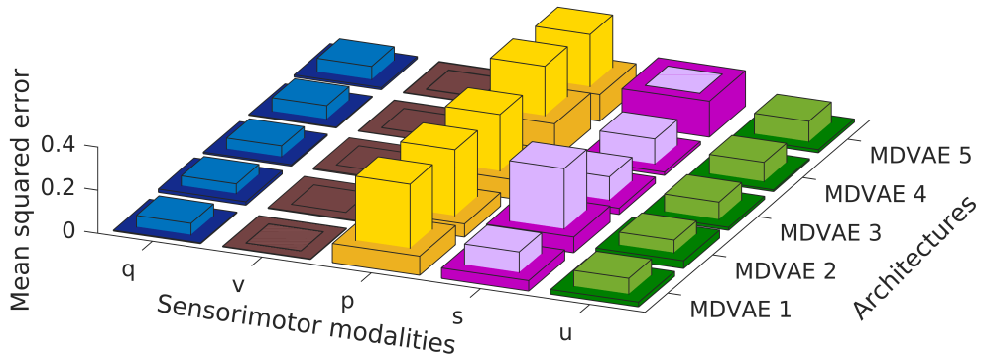


Figure 5.6: Architecture structures comparison. Different structures are compared in terms of mean squared error. Each colour corresponds to one of the sensorimotor modalities: q - joint positions (blue), v - visual positions (brown), p - touch (yellow), s - sound (pink), u - motor commands (green). Darker tones correspond to the results obtained when reconstructing with complete data available, while lighter tones correspond to the results obtained when reconstructing with only visual information available (note that no variation is rightly registered for the visual data). The first structure MDVAE 1 allows to achieve the best overall behaviour among all the modalities, and particularly on the discrete tactile and sound ones. On the contrary, the worst overall performance is obtained using the shallower and smaller network MDVAE 5. All parameters are specified in Table 5.2.

tests is greater than that of the other modalities, but also the difference between the scores obtained in the two experiments is significantly bigger. This is due to the fact that these data are considerably more difficult to infer when only visual information is provided. The quality of the MDVAEs can be evaluated not only in terms of absolute mean squared error scores, but also noting the difference between the reconstruction results obtained when complete data are provided and when only visual information is given. The structure described above corresponds to the MDVAE 1, which is shown to achieve the best overall performance among all the modalities, and particularly on the discrete tactile and sound ones. The chosen structure presents the best behaviour not only in terms of absolute error scores but also in terms of such relative error score.

Table 5.2: Comparison of different structures of the Multimodal Deep Variational Autoencoder (MDVAE). The number of units for each layer is reported in square brackets. Shallow layers are represented by empty brackets. Details for each modality network and for the shared network are given. The architecture used in the following experiments is MDVAE 1. MDVAE 4 and 5 are shallow networks consisting of the shared network only.

		q	v	p	s	u	shared
MDVAE 1	Enc	[40,20]	[40,20]	[10,5]	[10,5]	[40,10]	[100,28]
	Dec	[40,8,8]	[40,8,8]	[10,2,2]	[10,2,2]	[40,8,8]	[100,70]
MDVAE 2	Enc	[20,10]	[20,10]	[5,3]	[5,3]	[20,10]	[50,28]
	Dec	[20,8,8]	[20,8,8]	[5,2,2]	[5,2,2]	[20,8,8]	[50,36]
MDVAE 3	Enc	[40]	[40]	[10]	[10]	[40]	[100,28]
	Dec	[8,8]	[8,8]	[2,2]	[2,2]	[8,8]	[100,140]
MDVAE 4	Enc	[]	[]	[]	[]	[]	[70,28]
	Dec	[]	[]	[]	[]	[]	[70]
MDVAE 5	Enc	[]	[]	[]	[]	[]	[50,28]
	Dec	[]	[]	[]	[]	[]	[50]

5.3.3 *Reconstruction of sensorimotor data*

The multimodal deep variational autoencoder is first trained using data points explored during motor babbling. The dataset collected during babbling is split into a training dataset and a validation dataset. As described in Section 5.2, the network is trained on both complete and partial data of the training set.

In order to evaluate the reconstruction ability of the network, it is first assessed whether the encoding and decoding of the variational autoencoder manages to retrieve complete input data (when all the modalities are present). Then the model is validated on the reconstruction of missing modalities, using only the visual information as input.

The experiments conducted showed that the learned network achieves considerable results in terms of reconstruction and beyond that in terms of capturing the complexity of the system. The network is able to provide an estimate of the input reconstructed even when the majority of the modality dimensions are missing. Importantly, the model is also able to provide a measure of the uncertainty due, for example,

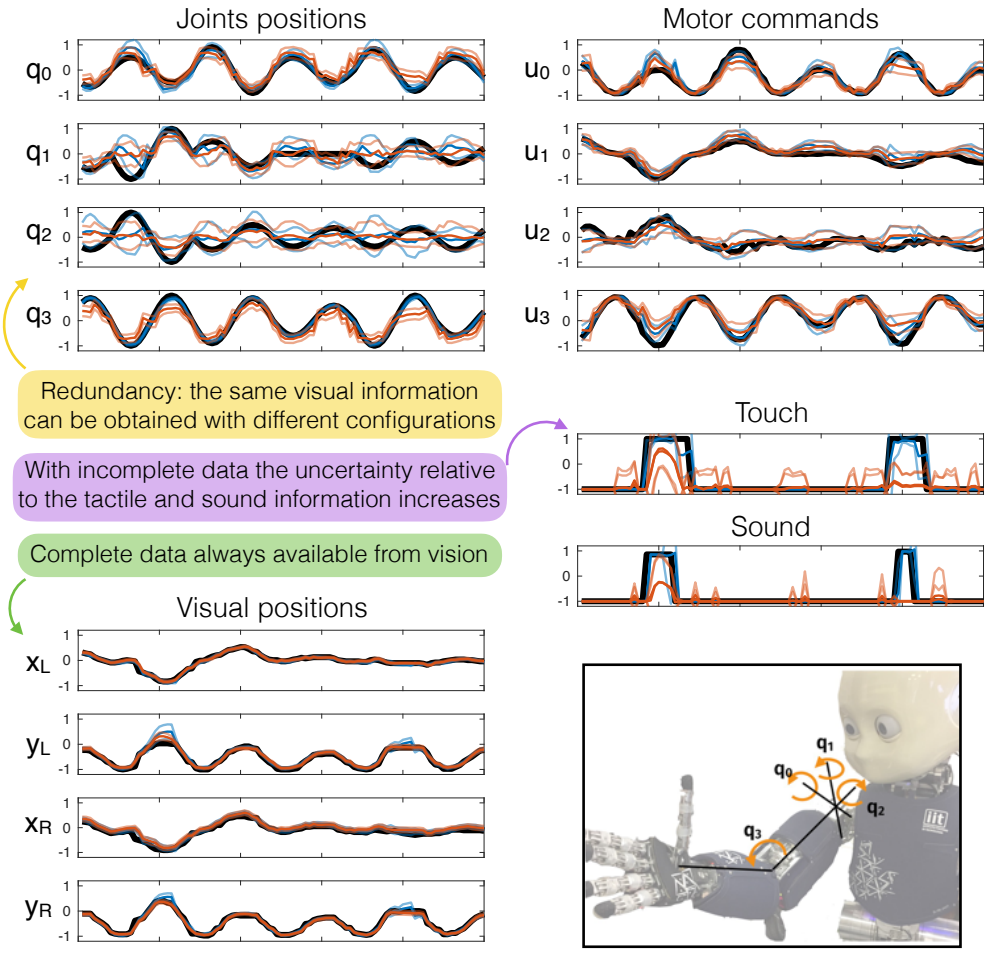


Figure 5.7: Reconstruction results on joints' positions, motor commands, visual positions, touch and sound, given complete and partial input data (blue and orange lines, respectively). Reconstruction results on the joint and motor spaces display the effect of the redundancy of the robot's arm: the same visual position can be reconstructed using diverse configurations, and applying diverse motor commands. Reconstruction errors occur simultaneously on different degrees of freedom, according to the robot's kinematic structure. The redundancy effect is particularly evident for the second and third joints (q_1, q_2). A representation of the degrees of freedom of the iCub arm is depicted in the lower-right picture. When only visual information is provided, reconstructing the discrete tactile and sound data is more difficult, thus the uncertainty of these reconstructed signals is higher compared to the results obtained when complete information is provided.

to the redundancy of the system. Results of the reconstruction obtained using the multimodal variational autoencoder are shown in Fig. 5.7. This figure shows the reconstruction results on the joints, motor, tactile, sound and visual spaces, obtained with both complete and partial input data. It is possible to note that while the reconstruction of the visual signals is very accurate, the reconstruction of the joints' positions and of the motor commands presents a peculiar behaviour. In particular, reconstruction errors occur simultaneously for diverse degrees of freedom. A closer analysis of these results shows that the joints where the reconstruction errors are simultaneous are related to the kinematic structure of the robot. The results shown in Fig. 5.7 demonstrate how this redundancy is captured by the multimodal variational autoencoder, thus demonstrating the power of this type of network on such difficult tasks. More specifically, the multimodal variational autoencoder allows to learn the general sensorimotor structure underlying the robot's movements rather than single trajectories or single motion sequences. In other words, a robot learns that there can be diverse configurations to achieve a target (for example a visual target). For instance, it can be seen that for q_1 and q_2 the variance of the reconstruction is particularly large. This comes from the fact that several joint configurations can explain the visual information provided to the architecture. Note that the true data to be reconstructed remains most of the time within the confidence range of the reconstruction. This phenomenon can also be interpreted as the ability of the network to capture the overall complexity of the robotic system, in particular the redundancy of its forward kinematics. The data used for this experiment belongs to the dataset collected from the robot self-exploration phase, but have not been used during the training of the model. It can be observed that the reconstruction obtained with partial data is very accurate. These results show that the network is able to correctly retrieve missing information, reconstructing them as accurately as when complete data is available. Note that although the number of dimensions to be reconstructed is relatively high compared with the number of available modalities, accurate results are still obtained.

Table 5.3: Mean squared error scores for each dimension of the multimodal reconstructed signal on validation data (babbling data that have not been used from training the network). The error scores achieved with partial data are comparable to those obtained when feeding complete data to the network, showing the ability of the network to learn cross-relations between the modalities.

	q	**v**	**p**	**s**	**u**
Rec. complete data	0.0396	0.0056	0.1818	0.2081	0.0663
Rec. partial data	0.0573	0.0058	0.1847	0.2137	0.0985

The results obtained show another interesting capability of the learned network, namely the ability of learning a forward kinematics only using 2D images from the robot's cameras, while not having direct access to the 3D position of the hand in the robot's operational space. This allows the system to avoid the use of stereo vision algorithms (with the related calibration and matching issues), while having the possibility to rely on the on-board 2D RGB cameras. The successful reconstruction of the robot joints' configuration when only visual information is provided shows the effectiveness of the learned model in achieving such forward kinematic mappings.

The mean squared errors of the reconstructed sensorimotor signals on validation data for each modality have been computed to provide a quantitative account of the network performance. In Table 5.3, the error scores obtained both when complete and partial data are provided to the network have been reported. Note that the error scores achieved with partial data are comparable to those obtained when feeding complete data to the network. This shows that the performance of the network is not degraded significantly when the input data consists only of partial data (*i.e.* vision only). This also shows that the network has successfully learned not only a direct reconstruction of each single modality but also cross-relations between the modalities and the way to reconstruct one of them provided only visual data are available.

5.3.4 *Prediction of sensorimotor states of self and others*

Using the multimodal deep variational autoencoder trained on data of the robot itself, the robot is able to make predictions also of others' motion trajectories in the visual space. When observing others, the robot only has access to the visual information. The learned model is then used to retrieve the motor commands (together with the other missing sensory modalities) that would enable the robot to reproduce the trajectory observed to perform mental simulation of the observed action.

First, the proposed architecture has been evaluated using test data from the robot's own data collected from motor babbling. Results of the predictions of the visual trajectories obtained on data explored during motor babbling are shown in Fig. 5.8. The mean squared prediction error score obtained on this experiment is 0.0083. The data

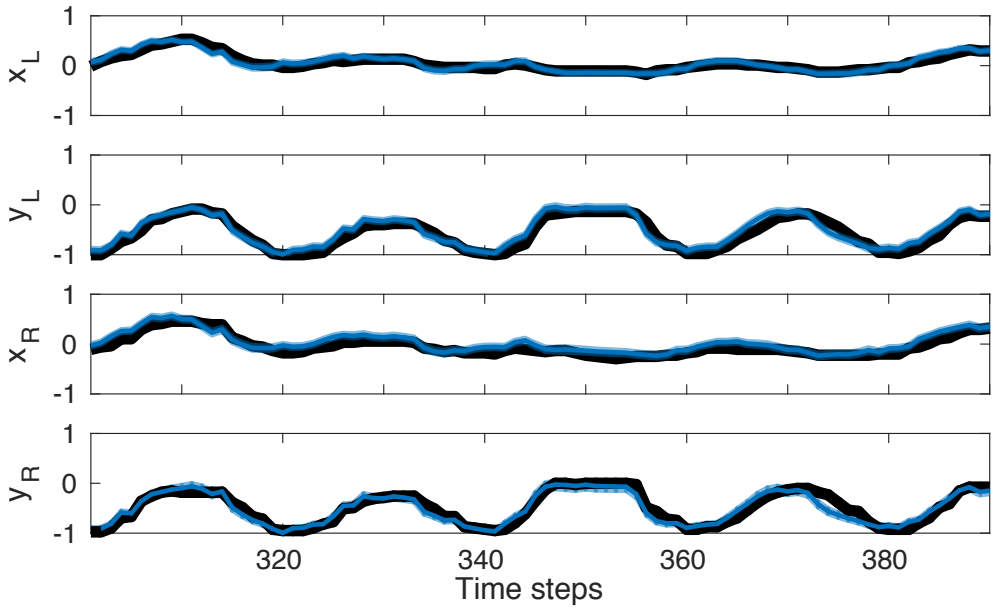
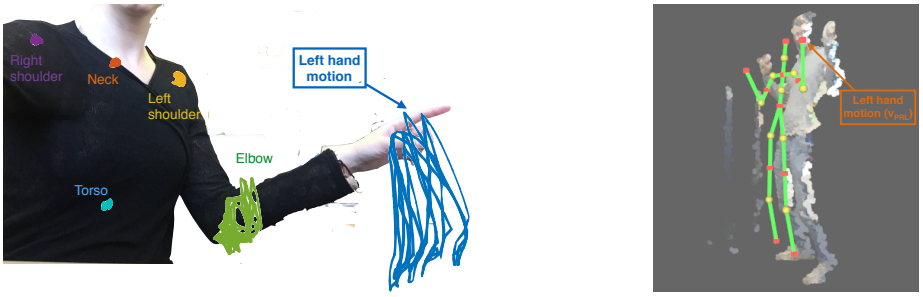


Figure 5.8: Prediction results using the learned model to predict the visual trajectories (with coordinates x_L, y_L, x_R, y_R) of the robot's own motion (a representative part of the trajectories is depicted). Solid black lines represent the real data (part of the test database), while blue lines represent the predicted mean and the shaded light blue areas the predicted variance (uncertainty) of the model. In each plot, the time steps are represented on the horizontal axis, while the magnitude (normalised) of each of the four dimensions of the visual state are represented on the vertical axis.



(a) Human piano playing. (b) Imperial-PRL KSC Dataset.

Figure 5.9: (a) Kinect data of a human upper-body movements while playing a piano keyboard with one hand. (b) Kinect data from the Imperial-PRL KSC Dataset. The trajectory of the left hand $\mathbf{v}_{\text{PRL}}$ has been used as test dataset.

on which the experiment is carried out is the test database, that is a part of the data from the robot's self-exploration which was not used for training the model. These results show that the network is able to effectively make accurate predictions by first reconstructing missing data from visual positions only, and then iterating the process for a second time in order to achieve the next step prediction.

Then the architecture has been evaluated on data collected from the observation of other agents. Experiments have been carried out using two different datasets. The first test dataset consists of movements of a human playing a piano keyboard, that has been recorded by the authors using a RGB-D camera (Fig. 5.9a). The second test dataset is part of the Imperial-PRL KSC Dataset (data used in (Chang and Demiris, 2015; Chang et al., 2016) to validate kinematic structure correspondences methods). It contains kinect data of a human moving his hands (represented in Fig. 5.9b). The 3D visual positions of these two datasets are then translated into 2D data by using two of the three available dimensions. This corresponds to a coarse approximation of the projection of the 3D trajectories onto the two eyes' cameras of the robot.

While the first dataset is similar to the self-exploration dataset in terms of scenario and application, the second one is significantly different, involving the free motion of the human arms, which are not confined within the scope of a keyboard. The first test dataset allows us to demonstrate that the robot can effectively reconstruct and pre-

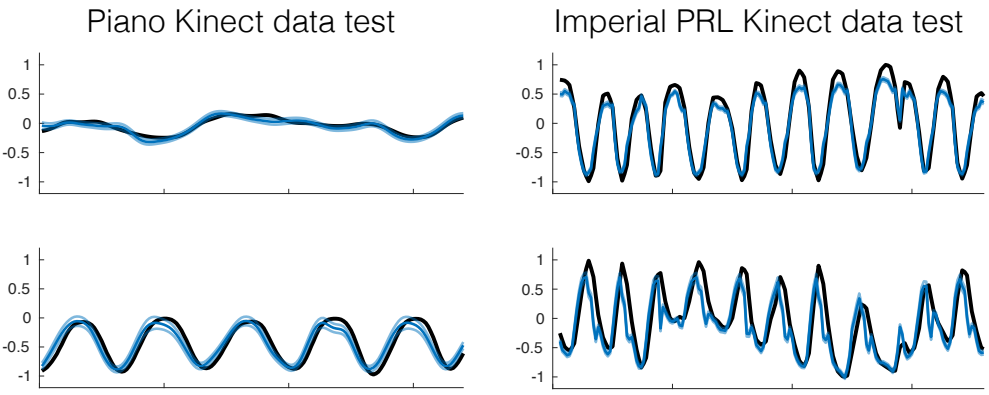


Figure 5.10: Predictions of others' trajectories. Solid black lines represent the real data, while blue lines represent the predicted mean and the shaded light blue areas the predicted variance (uncertainty) of the prediction model. Prediction of human playing a piano keyboard (left) and prediction of the left hand motion $\mathbf{v}_{\text{PRL}}$ (right). In each plot, the time steps are represented on the horizontal axis, while the magnitude (normalized) of two visual coordinates are represented on the vertical axis.

dict another agent's performing a sequence of motions that are similar to those performed in the motor babbling phase by using the learned internal models. The second test dataset allows us to demonstrate that the robot is able to reconstruct and predict others' motion using the learned models also when the type of motion is significantly different from the data acquired by the robot from self-exploration.

Results are shown in Fig. 5.10. In this figure, the results obtained for the two datasets are reported. More specifically the left graphs show the prediction performance on the kinect data collected from a human playing a piano keyboard (see Fig. 5.9a), and the right graphs show the prediction performance on the kinect data from the Imperial-PRL dataset (specifically on $\mathbf{v}_{\text{PRL}}$, see Fig. 5.9b). The corresponding mean squared error scores obtained are 0.0258 and 0.0278 (without unit because of the normalization applied to the data) for the two datasets, respectively. Note that these results are one order of magnitude higher than the scores obtained on data of the robot's self-exploration. This result is in line with results on cognitive development in humans (Wolpert and Flanagan, 2001; Decety and Sommerville, 2003), explaining how the prediction discrepancy between the predicted and the actual consequences of movements is related to whether the action was

self-produced or generated by another agent. The results achieved thus demonstrate that the proposed architecture obtains predictions of others' sensorimotor data by only making use of internal models of self.

5.3.5 *Control to imitate other observed agents*

The learned model can also be used in a control loop (Fig. 5.5). This corresponds to using the model as a controller for the robot's motion. By deploying the learned model as a controller, it is possible to implement, for example, imitation tasks, where the robot tracks trajectories in the sensory space. The learned model is able to reconstruct the motor commands necessary to achieve reference trajectories. The retrieved motor commands can then be issued to the robot's motors. For this experiment, two datasets have been used: (1) target trajectories from motor babbling, and (2) data observed from the human playing a piano keyboard (dataset used also for predictions in the experiment presented in the previous paragraph). The first dataset consists of trajectories from the part of the babbling dataset that has not been used for training the network. This test dataset thus contains data that have not been seen by the network before, though they are similar to the data used for training. In particular, the associations between positions in the sensory space and corresponding values of the velocity motor commands are similar. The second dataset is more challenging, particularly because it may contain visual positions that were not contained in the training set, and this can in turn lead to combinations of the multimodal dimensions of the input that the network was never presented before. The objective is for the robot to imitate the observed target trajectories. The target trajectory is used as reference and fed to the network in place of $\mathbf{v_t}$, while the current visual position of the robot and the current joint configuration of the robot ($\mathbf{v_{t-1}}$ and $\mathbf{q_{t-1}}$) are fed back to the network. All the other modalities are considered missing, in particular the motor commands that are produced by the network online after each new observation.

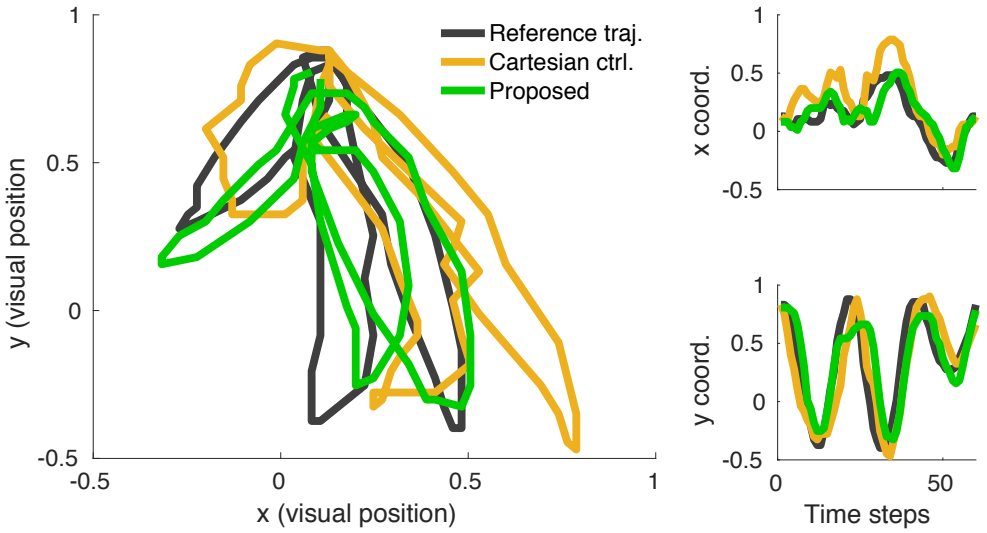


Figure 5.11: Results of the imitation task on robot's own movements realized by using the built-in Cartesian controller (yellow line) and the learned model (green line) to control online the robot's movements. The proposed method outperformed the built-in model, achieving higher accuracy in tracking the reference visual trajectory (grey line). The left plot shows the 2D visual position representation of the reference and executed trajectories, while the right plots show the corresponding temporal profiles of the positions (x and y coordinates). For clarity of the representation, only the trajectories acquired from the left eye camera of the robot are depicted, while similar results were obtained from the right camera.

In the first experiment, the proposed method has been compared with the Cartesian controller available on the iCub. The stereo vision system of the iCub is used to determine the 3D position in the Cartesian space associated with 2D visual inputs. This information is then used by the Cartesian controller to reach the target positions. Results obtained on the first dataset are represented in Fig. 5.11. It can be noted that the robot is able to reproduce target trajectories using the learned model, which outperformed the built-in Cartesian controller. The mean squared error score achieved by the proposed model on this task on the four-dimensional visual data is only 0.0195, a very low value considering the resolution of the image and the precision of the visual data encoding the hand position throughout the experiments. The built-in Cartesian controller achieved a less accurate tracking of the reference visual trajectory, with a mean squared error score on the

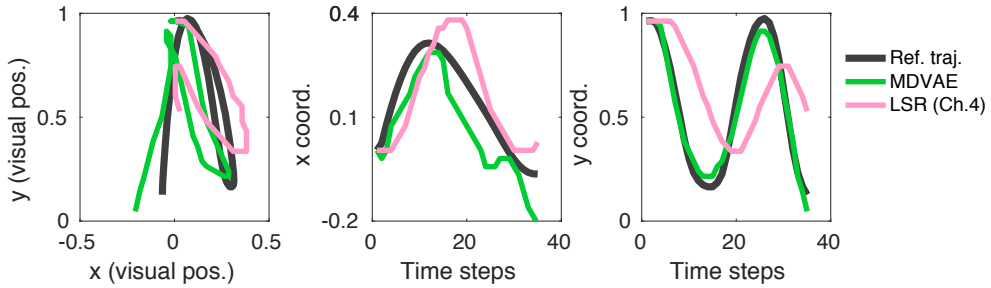


Figure 5.12: Results of the imitation task on the data collected from a human playing a piano keyboard. The proposed multimodal deep variational autoencoder (MDVAE) model (green lines) allows the robot to effectively track the reference visual trajectories (grey lines), outperforming the least square regression (LSR) method presented in Chapter 4 (pink lines) of one order of magnitude. The leftmost plot shows the 2D visual position representation of the reference and executed trajectories, while the central and rightmost plots show the corresponding temporal profiles of the positions (x and y coordinates). For the clarity of the representation, only the trajectories acquired from the left eye camera of the robot are depicted, while similar results where obtained from the right camera.

four-dimensional visual data of 0.0431, that is more than double the error achieved with the proposed method.

The experiments on the second dataset are also instrumental to show that the proposed method allows a robot to use data observed from another agent and imitate them. Results are represented in Figure 5.12. Here, the results obtained with the proposed multimodal deep variational autoencoder (MDVAE) are also compared against the results obtained on the same task by the least square regression (LSR) method presented in Chapter 4. The proposed MDVAE model outperforms the LSR model, achieving more accurate imitation results. The mean squared error scores achieved on this task on the four-dimensional visual data are 0.0054 and 0.0572 for the MDVAE and LSR models, respectively. The proposed MDVAE thus outperforms the previous method by one order of magnitude. It is also possible to note that the results on the y coordinate are more accurate than those obtained on the x coordinate. This reflects the structure of the actions performed during the exploration, which are used for training the model. While the exploratory movements spanned a wide range on

the vertical direction, a smaller part of the space was explored on the horizontal direction. This bias on the data acquired through the exploration results in a bias in the network performance observed in this experiment.

5.4 SUMMARY

This work takes inspiration from cognitive studies showing that humans can predict others' actions by using their own internal models. Following this direction, a new architecture that allows a robot to predict sensory consequences of other agents' actions by using only self-learned internal models has been proposed. A multimodal deep learning architecture has been introduced that allows a robot to (1) reconstruct missing sensory modalities, (2) predict the sensorimotor state of itself and others, and (3) control its motion to imitate the observed agent. This versatility represents a major advantage of the proposed approach, that can thus be applied in different applications to address different objectives (*e.g.* prediction, control, etc.). This architecture leverages advantages of developmental robotics and of deep learning, and has been evaluated extensively on different datasets and set-ups.

The results presented in this study show that a robot can learn to predict others' actions by exploiting only self-learned internal models. In this study, it has been argued that one of the main challenges in achieving predictions of others only based on internal models of self is the difference of the available data. While the whole set of sensorimotor data is available when the robot is acting and exploring, only visual information is available when the robot observes another agent. This motivated the proposed strategy to reconstruct and infer the missing information. In particular, the proposed variational autoencoder allows a robot to learn probability distributions among different sensorimotor modalities that allows to capture the kinematic redundancy of robot and human motions.

The proposed approach can be enhanced by enforcing the variational autoencoder to learn a latent space of a certain shape, for

example a Gaussian, from which inputs can be sampled in a more meaningful manner.

A limitation of the current implementation is the dependence of the reconstruction accuracy on the explored sensorimotor space. In particular, it is possible that combinations of sensory states reached during an imitation task are far from the training set of states used in the training of the network. In this case the network "guesses" motor commands by sampling from the learned distribution, but the reconstruction accuracy is usually poor due to the lack of samples resembling the observed new sensory state. A promising direction to overcome this issue is the implementation of more sophisticated exploration strategies, for instance curiosity-based strategies (Maestre et al., 2015; Baranes and Oudeyer, 2010).

CONTRIBUTIONS The main contributions illustrated in this chapter have been submitted:

- M. Zambelli, A. Cully and Y. Demiris, "A multimodal deep model for prediction and imitation of self and others' sensorimotor states", IEEE Transactions on Robotics (under review).

CHAPTER 6

Conclusions

6.1 OVERVIEW OF THE THESIS

This thesis set out to endow a robot to learn internal models in an autonomous manner by exploiting and leveraging information from multiple sources. This has been achieved either by considering combinations of multiple predictors to form ensembles of experts, and by integrating information available to the robot from multiple sensory modalities, such as vision, touch, proprioception. The ability of integrating multiple information has shown to benefit the learning process in several ways.

First, it has been shown that the forward model learned through the ensemble of predictors achieved higher prediction accuracy compared with alternatives and compared with any other single predictor model. Experiments presented in Chapter 3 showed that the proposed heterogeneous ensembles is an effective solution to achieve high prediction accuracy on validation and test datasets, without the need of fine tuning of the base models used. The proposed algorithm can update online by relying on the combination of online, recursive or incremental algorithms of different types, thus allowing for the forward model learned to adapt over time. Results have shown that significantly different algorithms can be effectively combined to achieve a

final accurate prediction model. Finally, the proposed method proved successful on real robot data, including multiple sensory inputs. This is in line with the overall contribution of the thesis and contribute to the learning of multimodal representation from self-acquired data.

Second, the integration of multiple sensory modalities in learning an inverse model from self-acquired data has proven fundamental for imitation tasks inherently involving multiple targets, such as playing a piano keyboard. Furthermore, multimodal integration has been shown to be fundamental when predicting observed actions. Experiments presented in Chapter 4 and Chapter 5 showed that by learning multimodal representations a robot can achieve successful execution of multimodal tasks as well as prediction of trajectories on different sensory spaces. The methods presented have been shown successful on imitation tasks involving multiple modalities, as well as in learning a representation that could be used by the robot to complete other tasks, such as predicting others' motion. Experiments involving the multimodal variational autoencoder have further shown that the learned model outperforms the robot's built-in controller on imitation tasks, and successfully predicts others' motion only based on internal representations. Both methods presented in Chapter 4 and Chapter 5 achieved the goal of learning representations across sensory modalities and, at the same time, anchoring this representation to the actions (motor commands). This feature allows the robot to use the learned multimodal representation in different ways, *e.g.* control and prediction.

The proposed learning approaches can constitute building blocks in a learning architecture for an autonomous robot. Using the proposed frameworks, a robot can acquire internal models from self-exploration, using no prior knowledge on its kinematic or dynamic structure, while leveraging information from different senses.

While many parts of the proposed approaches are biologically inspired, no claims are made as to the biological plausibility. The goal was to enable robots to use important properties of human development, such as prediction and learning abilities, not to verify or suggest that the proposed approaches are models of the process occurring in human brains, at either neuroscientific or psychological level.

6.2 FUTURE DIRECTIONS

The methods presented in this thesis are all based on a self-exploration (motor babbling) phase that is used by the robot to acquire sensorimotor data. A pseudo-random motor babbling was exploited in this thesis to produce the exploration motions. An interesting way to enhance this process is to apply more advanced exploration strategies, such as strategies based on curiosity. These strategies can improve the solutions proposed by allowing the acquisition of a self-perception database that covers the robot and the environment states as much as possible. Another improvement of the current work is represented by applying the exploration to the full body, to obtain a full body babbling. This would open further interesting questions relative to the organization of possible multiple models learned for each body part, their organization and inter-dependency.

The ensemble learning methods presented in Chapter 3 allowed the learner to continuously update the internal model as new data became available. The proposed method could be enhanced by using the input as a gate to update the ensemble weights. Characteristics of the input signals such as magnitude or frequency can have a direct impact on the single base models' performance, and can thus be used to modulate how the ensemble weights update. Moreover, the input space can be further scaled up, enlarging the sensorimotor space to include more dimensions (*e.g.* pixel information from the visual modality, or more granular representations of the pressure sensed by the tactile skin).

The multimodal nature of the sensorimotor information available to the robot was also considered in this thesis as a fundamental aspect in developing autonomous learning approaches. The method proposed in Chapter 4 based on a rigid least square regression formulation took into consideration multiple modalities, but the representations learned were static. The major problems of this method were addressed by the deep learning architecture proposed in Chapter 5. For both cases, the multimodal space can be enlarged. For example, the full images acquired from the robot's cameras can be considered instead of the visual coordinates of the tracked limb. Convolutional

neural networks have been proven to be successful in extracting valuable information from raw images. A larger range of visual information could thus be used in the learning process. Another interesting way to expand the multimodal information is to use language and verbal clues (Cangelosi, 2010; Lyon, Nehaniv and Saunders, 2012; Lyon et al., 2016; Zhong et al., 2017; Zhong, Cangelosi and Ogata, 2017).

Finally, the potential of the presented multimodal deep variational autoencoder can be further exploited. First, the generative capability of the model can be used to generate new unobserved sensorimotor state by sampling the learned latent space. Second, the cross-modality relations learned by the network can be further analysed, and exploited in order to learn, in addition to the sensorimotor trajectories, abstract representations of the interactions with the environment (*e.g.* properties of the different senses in correspondence to different objects/scenarios).

6.3 EPILOGUE

The chapters in this thesis illustrated and examined methods for a humanoid robot to learn its own internal models in a completely autonomous fashion, using as little prior knowledge as possible, and exploiting as much sensory information as possible. Inspired by findings in developmental psychology, this thesis showed that self-learned models can be used by the robot not only to make predictions about its own sensorimotor states, but also to develop complex cognitive tasks such as imitation, and to make predictions about other agents' actions. The benefits of this approach are multiple, including less programming effort for programmers/engineers, adaptability of the robot to new environments, increased versatility for the robot to learn new skills in a more intuitive manner.

APPENDIX A

Base models

In Chapter 3, online heterogeneous ensembles of experts have been presented. The pool of models used to build the ensemble includes four different types of models, namely Echo State Networks (ESN), Online Echo State Gaussian Processes (OESGPs), Locally Weighted Projection Regression (LWPR) models and Recursive AutoRegressive models with eXternal inputs (RARX). Here, each model is presented in more details.

A.1 ECHO STATE NETWORKS

Echo State Networks (ESN) have been introduced by Jaeger (Jaeger, 2002). They are a class of recurrent neural network (RNN), consisting of a large, fixed, recurrent network, where only the output weights are trained instead of adapting all network weights. The architecture is based on a randomly generated fixed RNN, called the reservoir. The network is driven with the input signal, thus inducing a response in each neuron of the reservoir. The output is then obtained as a trainable linear combination of the responses of the reservoir units (*e.g.*, using standard linear regression). Instead of adapting all network weights,

only the output weights are trained. More precisely, the state of the reservoir is updated during training as follows

$$\mathbf{x}_{t+1} = (1-\gamma)h(\mathbf{W}\mathbf{x}_t + \mathbf{W_i}\mathbf{u}_{t+1} + \mathbf{W_b}\mathbf{y}_t) + \gamma\mathbf{s}_t \tag{A.1}$$

where $\mathbf{x}_t$ is the state of the reservoir units at time t, $\mathbf{u}_t$ is the input, $h(\cdot)$ is the activation function, y_t is the desired output, $\mathbf{W}$ is reservoir weight matrix, $\mathbf{W_i}$ is the input weight matrix, $\mathbf{W_b}$ is the output feedback weight matrix, and γ is the leak rate. After training, the update equation is the following:

$$\mathbf{x}_{t+1} = (1-\gamma)h(\mathbf{W}\mathbf{x}_t + \mathbf{W_i}\mathbf{u}_{t+1} + \mathbf{W_b}\hat{\mathbf{y}}_t) + \gamma\mathbf{x}_t \tag{A.2}$$

and the predicted outputs $\hat{y}_t$ are obtained using

$$\hat{y}_t = \mathbf{W_o}\psi_t \tag{A.3}$$

where $\mathbf{W_o}$ is the linear output weight matrix and $\psi_t \triangleq [\mathbf{x}_t; \mathbf{u}_t]$ is the augmented reservoir state and input vector.

For online training, stochastic gradient descent can be used to update the output weights iteratively:

$$\mathbf{W}_{\mathbf{o},t+1} = \mathbf{W}_{\mathbf{o},t} + \eta(y_t - \hat{y}_t)\mathbf{u}_t \tag{A.4}$$

where η is the learning rate. To enable better convergence, Jaeger (Jaeger, 2002) proposed using the recursive least squares algorithm, which for ESN consists of the following iterative updates:

$$\rho_{t+1} = y_{t+1} - \psi_{t+1}^T \mathbf{w}_{\mathbf{o},t} \tag{A.5}$$

$$\mathbf{g}_{t+1} = \mathbf{P}_t\psi_{t+1}\left(\lambda + \psi_{t+1}^T\mathbf{P}_t\psi_{t+1}\right)^{-1} \tag{A.6}$$

$$\mathbf{P}_{t+1} = \lambda^{-1}\mathbf{P}_t - \mathbf{g}_{t+1}\psi_{t+1}^T\lambda^{-1}\mathbf{P}_t \tag{A.7}$$

$$\mathbf{w}_{\mathbf{o},t+1} = \mathbf{w}_{\mathbf{o},t} + \rho_t\mathbf{g}_{t+1} \tag{A.8}$$

where $\mathbf{w}_{\mathbf{o},t}$ is a row of the output weights matrix at time t and λ is the forgetting factor.

A.2 ONLINE ECHO STATE GAUSSIAN PROCESSES

Online Echo State Gaussian Processes (OESGPs) have been introduced in (Soh and Demiris, 2012). OESGPs combine ESN with Bayesian online learning for Gaussian processes, using sparse GP approximations to maintain the computational and storage costs fixed.

The OESGPs extend the Echo State Gaussian Process (ESGP) proposed by Chatzis and Demiris (Chatzis and Demiris, 2011), which is a Bayesian formulation of the standard echo-state network, based on Gaussian processes (GPs). Given an observation space with elements $\mathbf{x}$, a GP is a set of random variables whereby any finite subset has a joint Gaussian distribution (Rasmussen and Williams, 2006). A GP is defined as

$$f(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{A.9}$$

and it is completely specified by its mean function, $m(\mathbf{x}) = E[f(\mathbf{x})]$, and its covariance function $k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$.

Denoted $\mathbf{u}_{t+1}$ as the input into the reservoir, $\tilde{y}_{t+1}$ as the observed output signal, d_{t+1} as the true noise-free output, $\mathbf{x}_{t+1}$ as the updated reservoir state and $\psi_{t+1} \triangleq [\mathbf{x}_{t+1}; \mathbf{u}_{t+1}]$, and the reservoir states retained, called the basis vectors, as $\mathbf{b} \in \mathcal{B}$, the training of an OESGP consists of the following four steps.

1. Update the ESN state using eq. (A.1) to derive the new composite state ψ_{t+1}.

2. Compute a score γ of ψ_{t+1} that accounts for the "novelty" of the state ψ_{t+1}: $\gamma(\psi_{t+1}) = k_r(\psi_{t+1}, \psi_{t+1}) - \mathbf{k}_{\mathcal{B},t+1}^T \mathbf{K}_{\mathcal{B},t}^{-1} \mathbf{k}_{\mathcal{B},t+1}$, where k_r denotes the reservoir kernel function, $\mathbf{k}_{\mathcal{B},t+1} = [k_r(\mathbf{b}_i, \psi_{t+1})]_{\mathbf{b}_i \in \mathcal{B}}$, and $\mathbf{K}_{\mathcal{B},t}^{-1} = [k_r(\mathbf{b}_i, \mathbf{b}_j)]_{\mathbf{b}_i, \mathbf{b}_j \in \mathcal{B}}$.

3. If the score is higher than a pre-defined threshold, perform a full update of the model posterior: $m_t(\psi) = \alpha_t^T \mathbf{k}_r(\psi)$, $k_t(\psi, \psi') = k_r(\psi, \psi') + \mathbf{k}_r(\psi)^T C_t \mathbf{k}_r(\psi')$, where α and C are parameter updated according to unit vector e_t; otherwise perform an approximate update using $\hat{e}_t = \mathbf{K}_{\mathcal{B},t-1}^{-1} \mathbf{k}_{r,t+1}$ in place of e_t.

4. Maintain the size of $\mathcal{B}$ by removing the lowest scoring basis vector if $|\mathcal{B}|$ exceeds some predefined capacity.

Making predictions with the OESGP is then achieved by using the mean and the variance of the predictive distribution:

$$\mu_* = \mathbf{k}_{\mathcal{B},t}(\psi_{t*})^T \alpha_t \tag{A.10}$$

$$\sigma_*^2 = k_r(\psi_{t*}, \psi_{t*}) + \mathbf{k}_{\mathcal{B},t}(\psi_{t*})^T C_t \mathbf{k}_{\mathcal{B},t}(\psi_{t*}). \tag{A.11}$$

A.3 RECURSIVE AUTOREGRESSIVE MODELS WITH EXTERNAL INPUTS

A general way of illustrating this recursive algorithm is the recursive least square (RLS) method (Ljung, 1998; Ljung, 1983). More in particular, the recursive AutoRegressive model with eXternal inputs (ARX) has been considered. The ARX model structure is described by the equation:

$$y_t + a_1 y_{t-1} + \cdots + a_{n_a} y_{t-n_a} = b_1 u_{t-n_k} + \cdots + b_{n_b} u_{t-n_k-n_b+1} + e_t, \tag{A.12}$$

where y_t is the output at time t, $y_{t-1}, \ldots, y_{t-n_a}$ are the previous outputs on which the current output depends, $u_{t-n_k}, \ldots, u_{t-n_k-n_b+1}$ are the previous and delayed inputs, e_t is a white-noise disturbance value, the parameters n_a and n_b are the orders of the ARX model, and n_k is the delay. A more compact formulation of the model equation is

$$A(q)y_t = B(q)u_{t-n_k} + e_t \tag{A.13}$$

where q is the delay operator. Specifically,

$$A(q) = 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a} \tag{A.14}$$

$$B(q) = b_1 + b_2 q^{-1} + \ldots + b_{n_b} q^{-n_a+1}. \tag{A.15}$$

This is a parametric approach, where the goal is to fit the data with the fixed-structured model represented by the previous equa-

tions. The user is required to choose the polynomial orders, so that the model is in fact manually specified.

The parameters of the models can be gathered into a parameter vector $\theta = [a_1, ..., a_{n_a}, b_1, ..., b_{n_b}, n_k]^T$. The general form of the recursive estimation algorithm is as follows:

$$\hat{\theta}_t = \hat{\theta}_{t-1} + K_t (y_t - \hat{y}_t), \tag{A.16}$$

where $\hat{\theta}_t$ is the parameter estimate at time t, $\hat{y}_t$ is the prediction of the output y_t based on observations up to time $t-1$, and the gain K_t determines how much the current prediction error $y_t - \hat{y}_t$ affects the update of the parameter estimate. This gain is given by $K_t = Q_t \psi_t$, where ψ_t is the gradient of the predicted model output $\hat{y}_{t|\theta}$ with respect to the parameters θ, and the term Q_t is determined using an optimisation algorithm (such as the gradient method or the forgetting factor method) (Ljung, 1983).

A.4 LOCALLY WEIGHTED PROJECTION REGRESSION

Locally Weighted Projection Regression (LWPR) is an algorithm for incremental real-time learning of nonlinear functions. It was introduced in (Vijayakumar and Schaal, 2000a; Vijayakumar and Schaal, 2000b), and then used to solve different learning problems, such as learning in high dimensional space and learning of robot models (Vijayakumar, D 'souza and Schaal, 2005; Klanke, Vijayakumar and Schaal, 2008). The key idea behind this algorithm is to approximate nonlinear functions by exploiting piecewise linear models, whose region validity (called *receptive field*) is computed from a Gaussian function. Then, the model final output consists of the weighted mean of all linear models.

Consider a standard regression model:

$$y = f(\mathbf{x}) + \varepsilon, \tag{A.17}$$

where $\mathbf{x}$ denotes the N-dimensional input vector, y the output, and ε a zero-mean random noise term. When only a local subset of data in the vicinity of a point $\mathbf{x}_c$ is considered and the locality is chosen

appropriately, a low-order polynomial can be employed to model this local subset. A linear model can be considered to this end: $y = \beta^T \mathbf{x} + \varepsilon$. Nonlinear function approximations can thus be found by means of piecewise linear models. The LWPR regression function is constructed by combining local linear models $\psi_k(\mathbf{x})$ in the form

$$f(\mathbf{x}) = \frac{1}{W(\mathbf{x})} \sum_{k=1}^{K} w_k(\mathbf{x}) \psi_k(\mathbf{x}), \quad W(\mathbf{x}) = \sum_{k=1}^{K} w_k(\mathbf{x}). \tag{A.18}$$

The term $w_k(\mathbf{x})$ is a locality kernel that defines the area of validity of the local models which is usually modelled by a Gaussian

$$w_k(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c}_k)^T D_k (\mathbf{x} - \mathbf{c}_k)\right), \tag{A.19}$$

where c_k is the centre of the k-*th* linear model and D_k is its distance metric. During training, all updates to the local models are weighted by their activation $w_k(\mathbf{x})$, facilitating fully localised and independent learning. A new local model is created if no existing local model yields an activation above a certain threshold. The number K of local models is thus adapted automatically. For learning the linear models $\psi_k(\mathbf{x})$, LWPR employs an online formulation of weighted partial least squares regression. In particular, within each local model, the input data $\mathbf{x}$ is projected along selected directions $\mathbf{u}_i$, yielding "latent" variables $\mathbf{s}_i$ with

$$\mathbf{s}_i = \mathbf{u}_i^T \mathbf{x}_{i-1}, \quad \mathbf{x}_i = \mathbf{x}_{i-1} - \mathbf{p}_i \mathbf{u}_i^T \mathbf{x}_{i-1} \tag{A.20}$$

where the vectors $\mathbf{p}_i$ ensures orthogonality of the projections. The output of the local model is then formed by a linear combination of the latent variables:

$$\psi_k(\mathbf{x}) = \beta_0 + \sum_{i+1}^{R} \beta_i \mathbf{s}_i. \tag{A.21}$$

The number R of regression directions is automatically adapted to the local dimensionality of the training data, and the parameters $\mathbf{u}_i$, $\mathbf{p}_i$, and β_i can be robustly estimated. Similarly, the distance metrics D can be adapted using stochastic cross-validation, such that the input

space is covered by wide receptive fields in regions of low curvature, and narrow receptive fields where the curvature is high.

A.5 PARAMETERS USED TO INSTANTIATE BASE MODELS

In Chapter 3, experiments have been presented where different numbers of base models have been instantiated.

In the case of echo state networks, different numbers of internal nodes have been used. These were initialised as $\{1, 5, 10, 15, ..., 5 + 10(n - 2)\}$, where n indicates the number of base models instantiated for each type of algorithm. For example, when $n = 50$ the set of internal nodes used to initialise different ESNs is $\{1, 5, 10, 15, ..., 485\}$.

In the case of online echo state Gaussian processes, different length scales have been used. These were initialised as $\{0.15, 0.35, ..., 0.15n\}$ for synthetic data, and as $\{1.2, 1.35, ..., 1.2 + 0.15(n - 1)\}$ for iCub babbling data. For example, when $n = 50$ the sets of length scales used to initialise different OESGPs are $\{0.15, 0.35, ..., 7.5\}$ and $\{1.2, 1.35, ..., 8.55\}$, respectively.

In the case of RARX models, different polynomial orders have been used. Note that since each model is MISO (multiple input single output) the order n_b is always necessarily equals to 1, while the order of the A polynomial, *i.e.* n_a, can vary. The order n_a was thus initialised as $\{1, 2, ..., n\}$, where n indicates the number of base models instantiated for each type of algorithm. For example, when $n = 50$ the set of polynomial orders used to initialise different RARX models is $\{1, 2, ..., 50\}$. The parameter n_k was set equal to 1, to account for one step delay in the data observed.

In the case of LWPR models, different activation thresholds have been used. These were initialised as $\{0.05, 0.1, ..., 0.05n\}$. For example, when $n = 50$ the set of internal nodes used to initialise different LWPR models is $\{0.05, 0.1, ..., 2.5\}$.

Note that in all cases parameters were chosen in order to generate models of increased complexity, starting from the instantiation of small models first.

APPENDIX B

Further experiments on the Baxter robot

The experiments carried out on the Baxter robot and presented here demonstrate that the proposed ensemble learning method used to learn internal forward models can be applied to different robotic platforms. This is because the proposed method does not make any prior assumptions on the robot structure and is only based on data collected by the robot from self-exploration.

Baxter is a humanoid robot with two 7 degrees-of-freedom arms and equipped with a RGB camera on the head. The right arm has been used to perform motor babbling, visual data have been acquire from the head camera, and proprioceptive data (joints' positions) have been recorded from the motors' encoders. Motor babbling was performed engaging four of the seven degrees of freedom of one of the Baxter arms, and was realised by issuing pseudo-random sinusoidal velocity commands, programmed in order for the arm to move within the field of view of the head camera. Visual information, consisting of the positions of the lower and upper arm in the visual space (2D-coordinates computed from visual features) was collected from the head camera. Joints' encoders values (measured in radiants $[\mathrm{rad}]$) have been used for the proprioception modality, while pixel positions have been used for the visual modality. In particular, for each frame the 2D feature points tracked as part of the moving robot limb have been collected.

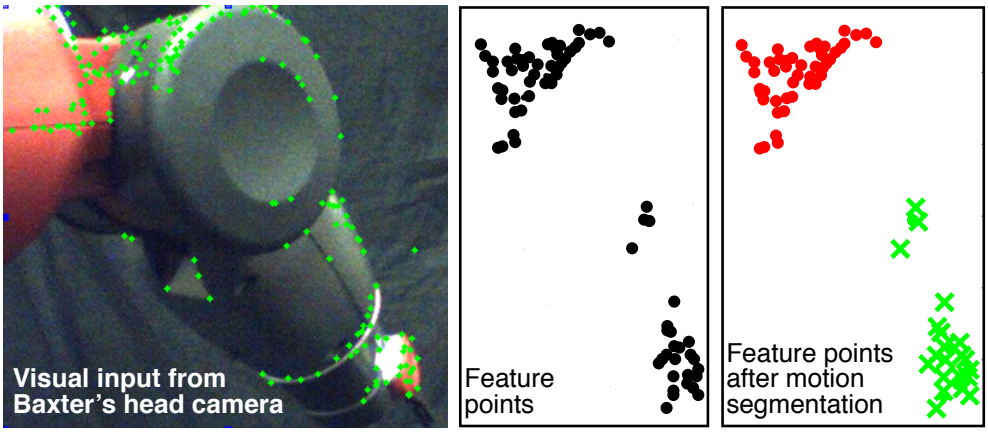


Figure B.1: Visual information (Baxter experiment). OpenCV feature detection and segmentation from motion are performed on the frames recorded from the head camera of the Baxter. Clusters identifying limb parts are represented.

A segmentation algorithm proposed in (Jung, Ju and Kim, 2014) has then been applied to identify parts of limbs in the visual frames. This algorithm is able to automatically cluster parts of the image according to motion, so that feature points moving together are clustered as one limb part. The frame size of the images acquired from the robot camera is 480×300 pixels. Independent movements were identified for the upper and lower arm of the Baxter, as shown in Figure B.1. The position of the limb in the vision space is computed as the position in the 2D image space of the centre of the clusters identified in the visual frames. This approach is thus robust against differences in shape, dimension, specific morphological characteristics.

The base models and the ensemble models are evaluated in terms of root mean squared error. The ensemble predictors are compared against the single base models, against an offline implementation of a standard tree-based bagging ensemble for regression (Breiman, 1996a), and against the homogeneous online ensemble obtained by applying the same ensemble integration algorithm but adopting homogeneous structures as base models (that is ensembles of ESNs only, of OESGPs only, of RARXs only, and of LWPRs only).

Experiments have been carried out also on the task of making multiple-step-ahead predictions. To realise a multiple-step-ahead predictor from

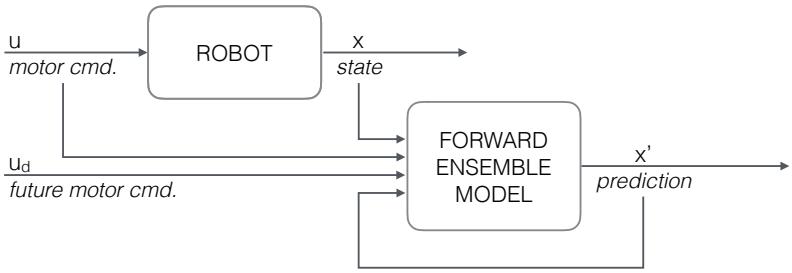


Figure B.2: Block scheme of the "naive" iterative k-step-ahead prediction model. Motor commands u, and future motor commands u_d (assumed known) are used on the robot's motors and as input to the forward (ensemble) model. The state x includes the proprioception and visual information. The multiple-step ahead prediction x' is obtained through the ensemble model.

the learned forward model, a simple implementation has been considered, based on feeding one-step-ahead predictions as input to the models that produced them. This implementation is sometime referred to as "naive", due to the fact that uncertainties generated in each iteration step of prediction are not considered. Denoted by y the sensory state of the robot (in this case formed by the visual and proprioceptive information), and by u the velocity commands issued to the robot joints, it is assumed that l previous states $x(i)$, for $i = \{(t-l), ..., t\}$, as well as k future velocity commands $u_d(i)$, $i = \{t+1, \ldots, t+k\}$, are known. This assumptions are realistic, since the previous outputs and inputs are available from the past experience of the robot, while the future commands (up to a certain number of steps ahead) can be thought of as already planned. In this implementation, the one-step-ahead predictions, performed by the base learners, are fed back as input for the next prediction together with the next motor command. To achieve the k-step-ahead prediction, this iteration is repeated k times. A representation of this implementation is depicted in Fig. B.2.

Results are reported in Table B.1. Each predictor has been evaluated both in short-term and long-term prediction performance. The one-step-ahead predictions obtained with the proposed online heterogeneous ensemble are in all the cases more accurate than those of all the other alternative solutions. The accuracy obtained by the proposed heterogeneous online ensemble method is approximately 20 to 85% higher compared to single predictors and homogeneous en-

Table B.1: RMSE scores for one-step-ahead and 30-step-ahead predictions. Scores in the proprioceptive space are given in [deg], while scores in the visual space are given in [pxl]. The proposed online heterogeneous ensemble achieves the best accuracy (scores in bold).

	Proprioceptive or Visual Element	Single base models				Offline	Online Homogeneous Ensembles				Proposed Online Heterog. Ensemble Method
		ESN	OESGP	RARX	LWPR	Tree-based Bagging	Homog. ESN	Homog. OESGP	Homog. RARX	Homog. LWPR	Heterog. Ensemble
1-step-ahead pred.	Shoulder Pitch	0.1222	0.1157	0.0419	0.0447	0.0442	0.0502	1.7683	0.0414	0.0413	**0.0356**
	Shoulder Roll	0.0449	0.0733	0.0429	0.0505	0.0438	0.0650	1.7072	0.0424	0.0456	**0.0351**
	Shoulder Yaw	0.0508	0.0803	0.0440	0.0460	0.0452	0.0787	1.9564	0.0435	0.0438	**0.0365**
	Elbow Flexion	0.0439	0.1089	0.0421	0.0431	0.0437	0.0981	2.0010	0.0417	0.0421	**0.0341**
	End-Eff. Cl., x	0.7773	0.8174	0.4862	1.2963	0.7996	0.4368	0.8178	0.4348	0.5812	**0.4340**
	End-Eff. Cl., y	0.7656	0.7735	0.6853	7.4233	1.1394	0.7496	0.7762	0.5997	0.8709	**0.5909**
	Arm Cluster, x	0.5028	0.5741	0.4809	2.1376	0.9796	0.5029	0.5586	0.4692	0.6012	**0.4566**
	Arm Cluster, y	0.2035	0.2134	0.2035	0.2095	0.2137	0.2036	0.2129	0.2024	0.2063	**0.2019**
30-step-ahead pred.	Shoulder Pitch	0.2079	2.7314	0.5012	0.2315	0.5165	0.1865	1.0378	0.5011	0.2109	**0.0732**
	Shoulder Roll	0.1244	9.8553	0.4373	0.1419	0.4509	0.0949	0.8140	0.4372	0.1352	**0.0779**
	Shoulder Yaw	0.1626	2.9569	0.5053	0.2227	0.5217	0.1494	2.8191	0.5050	0.1995	**0.0932**
	Elbow Flexion	0.1941	2.3872	0.5130	0.2102	0.5290	0.1838	0.7002	0.5127	0.1899	**0.0867**
	End-Eff. Cl., x	2.0561	4.3716	6.1392	6.4310	6.3767	1.9003	2.6667	6.1233	5.6672	**1.8755**
	End-Eff. Cl., y	2.7494	3.2962	10.0669	9.8536	10.4510	2.6049	3.1516	9.9824	9.6420	**2.4407**
	Arm Cluster, x	2.4029	4.8824	6.2943	6.8732	6.5788	2.4016	2.8459	6.2938	6.0330	**2.1155**
	Arm Cluster, y	1.3193	4.9293	1.3833	1.4951	1.3963	1.3187	4.4614	1.3829	1.2383	**1.0501**

sembles, and 50 to 60% higher compared to the accuracy obtained with the tree-based model.

The proposed online ensemble achieves the best performance also in the multiple-step-ahead prediction task, outperforming single predictors, the tree-based offline ensemble and homogeneous ensembles. In this case, the proposed heterogeneous online ensemble method outperforms single models and homogeneous ensembles by approximately 30 to 98% in accuracy, and the offline tree-based ensemble by approximately 30 to 60%. In this experiment, the prediction horizon of $k = 30$ has been taken into consideration, corresponding to roughly $\Delta T = 3$ seconds in the future. This time horizon is usually the time within small base actions take place.

The high prediction accuracy achieved by the heterogeneous ensemble is useful, for practical purposes, *e.g.* to improve a robot's performance in control tasks involving precise positioning of the end-effector or localisation of the end-effector in the robot's vision space. The heterogeneous ensemble allows to achieve the highest performance both in short and long term predictions, providing an accurate model for sensorimotor representations involving both proprioception and vision.

APPENDIX C

Robots

This thesis has been developed using mainly the iCub robot (Metta et al., 2010) (Fig. C.1, left). The Baxter robot (*Baxter - Redefining Robotics and Manufacturing - Rethink Robotics*) (Fig. C.1, right) has also been considered for some experiments (see Appendix B). Experiments on the iCub have been carried out using some of the open source C++ libraries relative to the robotics platform and relying on the middleware YARP (Yet Another Robot Platform) (Metta, Fitzpatrick and Natale, 2006). Experiments on the Baxter have been programmed using Python based libraries of the robotic platform, and relying on ROS (Robot Operating System) (Quigley et al., 2009). Finally, a MIDI keyboard has also been used. All these tools are described in more details below.

C.1 THE ICUB ROBOT

The iCub is a humanoid robot originally developed as part of the EU project RobotCub. It is a 1 metre high humanoid robot, with dimensions similar to that of a 3.5 year old child, and it represents a testbed for research into human cognition and artificial intelligence.

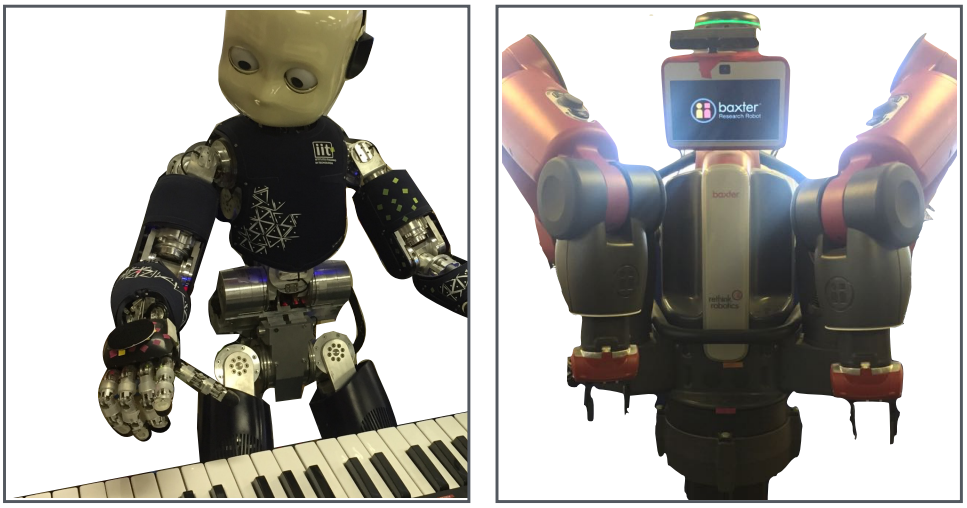


Figure C.1: The iCub (left) and the Baxter (right), humanoid robots used in the experiments presented in this thesis.

The robot is controlled by an on-board PC104 controller which communicates with actuators and sensors using CANBus. It utilises tendon driven joints for the hand and shoulder, with the fingers flexed by teflon-coated cable tendons running inside teflon-coated tubes, and pulling against spring returns. It has 53 motors that move the head, arms and hands, waist, and legs. It is endowed with multiple sensors, including RGB-cameras for vision, motor encoders for the sense of proprioception (body configuration), tactile sensors for the sense of touch, microphones for audition.

The tendon system allows to embed a considerable number of degrees-of-freedom, however the response of the joints to motor commands can present delays and imprecisions in positioning. This can result for example from wear or different temperature/humidity conditions.

Such environment conditions also influence the tactile sensors. The tactile sensor network (*i.e.* skin) of the iCub robot is organized into patches. A patch is a set of tactile sensors that are physically connected one to the other and that are read by the same microcontroller. Each patch is composed by triangular modules, each consisting of 10 tactile sensors (taxels), with two additional small ones that are thermal pads. Compensated tactile data were used in the experiments presented in this thesis. These values are floating point numbers in $[0, 255]$, where 0 means no pressure. The size of the vector of tactile

signals acquired depends on the skin part. The hand, and in particular the fingertip, was considered in the experiments presented. Hand data are described by 192 taxels, where the first 60 refer to the fingertips data (12 taxels per fingertip).

Visual data used in the experiments presented in this thesis was acquired from the robot's on-board cameras. A resolution of 320×240 pixels was used, so that the position of the hand in the visual field was defined in pixel coordinates. Although the resolution of the visual data could allow to achieve high accuracy in locating the hand of the robot in the visual field, the precision of the visual positions was limited by the tracking algorithm used. The final information obtained was nonetheless sufficient to achieve the goal of the studies presented.

The software libraries run on the iCub are largely written in C++ and use YARP for external communication with off-board software implementing higher level functionalities. The software developed to perform the presented experiment was implemented using different languages. The exploration policies, the forward models and the multimodal imitation based on multimodal matrices were implemented using Matlab. The variational autoencoder was implemented in Python using Tensorflow. The modules responsible to issuing motor commands to the robot's joints and to read sensor data from the different sensors were implemented in C++. The communication and synchronisation of the different code parts have been realised relying on the YARP middleware.

C.2 THE BAXTER ROBOT

Baxter is a humanoid robot developed by Rethink Robotics. It is approximately 1.8 meters in height and it has two 7 degrees-of-freedom arms and various sensors including force, position, and torque sensing for each joint. There are two RGB cameras at the endpoints of two limbs. Another RGB camera is present on the robot's head. Only the head camera has been used in the experiments presented in this thesis. A resolution of 480×300 was used to acquire visual data. A sig-

nificant limitation in using the head camera is the restricted field of view that can be captured from it. In order to collect data of the arm babbling, limited movements could be performed. An external camera could resolve this issue, however one of the goals and challenges of the experiments presented in this thesis was to rely on robots' onboard sensors.

The interfaces to control the robot using ROS are provided by the robot's SDK. The software developed to realise the experiments presented was implemented using Python, and linked to the libraries responsible for the basic control functions on the robot. The communication between the modules developed to send babbling motor commands to the robot's joints and to read sensor data relied on ROS nodes and topics.

C.3 YARP

YARP is a free and opensource middleware supporting a set of libraries, protocols, and tools to keep modules and devices cleanly decoupled. While it is not an operating system for the robot, it supports building a robot control system as a collection of programs communicating in a peer-to-peer way, with an extensible family of connection types (tcp, udp, multicast, local, MPI, ...).

YARP consists of three main libraries: (i) a library interfacing with the operating system(s) to support easy streaming of data across many threads across many machines, (ii) a library performing common signal processing tasks in an open manner easily interfaced with other commonly used libraries (*e.g.* OpenCV), and (iii) a library interfacing with common devices used in robotics, such as digital cameras, motor control boards, etc.

The YARP middleware was extensively used in the experiments performed and illustrated in this thesis. It was responsible mainly for the communications between different modules related either to the control of the robot or to the acquisition of sensor data from it. The main means used to implement such communications were YARP ports,

utilised to send messages of various type, such as control signals, sensor data, signals to trigger different behaviours.

C.4 ROS

The Robot Operating System (ROS) is a widely known and used framework for developing robot software. It consists of a collection of tools, libraries, and conventions that simplify the programming process of robots' behaviours, and that can be used for different robotic platforms.

At the lowest level, ROS represents a middle-ware, which deals with a message passing interface providing inter-process communication. This feature is the most relevant for the experiments developed in this thesis.

ROS was used mainly to manage the communication between different modules responsible to run the experiments on the Baxter robot. In particular, ROS nodes and topics were extensively utilised to send control commands to control the robot's joints, or to read sensor data from the robot's encoders or from its head camera.

C.5 MIDI KEYBOARD

An Akai MIDI keyboard has been used for the experiments carried out with the iCub. The MIDI keyboard is a piano-style user interface keyboard device used for sending MIDI signals or commands over a MIDI cable to a computer operating on the same MIDI protocol interface. MIDI information is sent to an electronic module capable of reproducing an array of digital sounds or samples, also referred to as voices or timbres. MIDI is a symbolic representation of musical information, encoding a specific instrument sample and sound parameters such as note volume (velocity), pitch bend and modulation controls.

MIDI data were acquired during the experiments performed with the iCub robot. The data collected were post-processed in order to obtain one-dimensional signals where the amplitudes and the dur-

ations of each note played was encoded. A C++ module was implemented in order to acquire the MIDI signals from the keyboard and transfer them through YARP. The signals thus acquired presented several issues, such as delays between the moment when a key was pressed and the moment when the signal registered a change of the sound produced, noise in the signals encoding a note played. The post-processing applied thus helped cleaning the data used in the robot learning processes.

Bibliography

Abadi, M. et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: http://tensorflow.org/ (cit. on p. 107).

Abbeel, P., A. Coates, M. Quigley and A. Y. Ng (2007). 'An application of reinforcement learning to aerobatic helicopter flight'. In: *Advances in neural information processing systems* 19, p. 1 (cit. on p. 30).

Andrew Bagnell, J. (2014). 'Reinforcement Learning in Robotics: A Survey'. In: *Springer Tracts in Advanced Robotics*. Vol. 97. Springer, pp. 9–67. DOI: 10.1007/978-3-319-03194-1_2 (cit. on pp. 32, 68).

Araki, T., T. Nakamura, T. Nagai, K. Funakoshi, M. Nakano and N. Iwahashi (2011). 'Autonomous acquisition of multimodal information for online object concept formation by robots'. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. IEEE, pp. 1540–1547. DOI: 10.1109/IROS.2011.6048422 (cit. on p. 38).

Argall, B. D., S. Chernova, M. Veloso and B. Browning (2009). 'A survey of robot learning from demonstration'. In: *Robotics and autonomous systems* 57.5, pp. 469–483. DOI: 10.1016/j.robot.2008.10.024 (cit. on pp. 30, 102).

Atkeson, C. G. and S. Schaal (1997). 'Robot learning from demonstration'. In: *Proceedings of the International Conference on Machine Learning*. Vol. 97, pp. 12–20 (cit. on pp. 30, 102).

Awais, M., F. Yan, K. Mikolajczyk and J. Kittler (2011). 'Novel fusion methods for pattern recognition'. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 140–155 (cit. on p. 43).

Baraglia, J., J. L. Copete, Y. Nagai and M. Asada (2015). 'Motor experience alters action perception through predictive learning of sensorimotor information'. In: *Proceedings of the Joint IEEE International Conference on Development and Learning and Epigenetic Robotics*. IEEE, pp. 63–69. DOI: 10.1109/devlrn.2015.7346116 (cit. on pp. 33, 39, 68, 94, 95, 102).

Baranes, A. and P.-Y. Oudeyer (2010). 'Intrinsically motivated goal exploration for active motor learning in robots: A case study'. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1766–1773. DOI: 10.1109/iros.2010.5651385 (cit. on p. 120).

Batula, A. M., M. Colacot, D. K. Grunberg and Y. E. Kim (2013). 'Using Audio and Haptic Feedback to Detect Errors in Humanoid Musical Performances'. In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. Vol. 2, pp. 295–300 (cit. on p. 38).

Bays, P. M. and D. M. Wolpert (2007). 'Computational principles of sensorimotor control that minimize uncertainty and variability.' In: *The Journal of physiology* 578.Pt 2, pp. 387–96. DOI: 10.1113/jphysiol.2006.120121 (cit. on p. 33).

Berg, M. de, O. Cheong, M. van Kreveld and M. Overmars (2008). 'Orthogonal Range Searching'. In: *Computational Geometry Algorithms and Applications*, pp. 95–120. DOI: 10.1007/978-3-540-77974-2_5 (cit. on p. 71).

Billard, A. (2001). 'Learning motor skills by imitation: a biologically inspired robotic model'. In: *Cybernetics and Systems* 32.1-2, pp. 155–193. DOI: 10.1080/019697201300001849 (cit. on pp. 31, 68).

Billard, A., S. Calinon, R. Dillmann and S. Schaal (2008). 'Robot Programming by Demonstration'. In: *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1371–1394. DOI: 10.1007/978-3-540-30301-5_60 (cit. on pp. 30, 31, 68, 102).

Blakemore, S.-J., J. Decety and C. Albert (2001). 'From the Perception of Action to the Understanding of Intention'. In: *Nature Reviews Neuroscience* 2.August, pp. 561–567. DOI: 10.1038/35086023 (cit. on pp. 27, 92).

Borji, A. and L. Itti (2013). 'State-of-the-art in visual attention modeling'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1, pp. 185–207. DOI: 10.1109/TPAMI.2012.89 (cit. on p. 36).

Bradski, G. and A. Kaehler (2008). *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc." (cit. on p. 73).

Breiman, L. (1996a). 'Bagging predictors'. In: *Machine learning* 24.2, pp. 123–140. DOI: 10.1007/BF00058655 (cit. on pp. 43, 44, 60, 134).

Breiman, L. (1996b). 'Stacked regressions'. In: *Machine Learning* 24, pp. 49–64. DOI: 10.1007/BF00117832 (cit. on p. 46).

Breiman, L. (2001). 'Random forests'. In: *Machine learning* 45.1, pp. 5–32 (cit. on pp. 43, 44, 60).

Broun, A., C. Beck, T. Pipe, M. Mirmehdi and C. Melhuish (2014). 'Bootstrapping a robot's kinematic model'. In: *Robotics and Autonomous Systems* 62.3, pp. 330–339. DOI: 10.1016/j.robot.2013.09.011 (cit. on p. 36).

Calinon, S., F. Guenter and A. Billard (2007). 'On learning, representing, and generalizing a task in a humanoid robot'. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37.2, pp. 286–298. DOI: 10.1109/TSMCB.2006.886952 (cit. on pp. 31, 68).

Cangelosi, A. (2010). 'Grounding language in action and perception: from cognitive agents to humanoid robots'. In: *Physics of life reviews* 7.2, pp. 139–151. DOI: 10.1016/j.plrev.2010.02.001 (cit. on p. 124).

Cangelosi, A., M. Schlesinger and L. B. Smith (2015). *Developmental robotics: From babies to robots*. MIT Press (cit. on pp. 33, 38).

Cattaneo, L. and G. Rizzolatti (2009). 'The mirror neuron system'. In: *Archives of neurology* 66.5, pp. 557–560. DOI: 10.1001/archneurol.2009.41 (cit. on p. 27).

Cesa-Bianchi, N., Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire and M. K. Warmuth (1997). 'How to use expert advice'. In:

Journal of the ACM 44.3, pp. 427–485. DOI: 10.1145/258128.258179 (cit. on p. 44).

Chang, H. J. and Y. Demiris (2015). 'Unsupervised learning of complex articulated kinematic structures combining motion and skeleton information'. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3138–3146. DOI: 10.1109/CVPR.2015.7298933 (cit. on pp. 80, 102, 106, 114).

Chang, H. J., T. Fischer, M. Petit, M. Zambelli and Y. Demiris (2016). 'Kinematic Structure Correspondences via Hypergraph Matching'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4216–4425. DOI: 10.1109/cvpr.2016.457 (cit. on pp. 102, 106, 114).

Chang, X., Z. Ma, Y. Yang, Z. Zeng and A. G. Hauptmann (2017). 'Bi-level semantic representation analysis for multimedia event detection'. In: *IEEE transactions on cybernetics* 47.5, pp. 1180–1197 (cit. on p. 37).

Chatzis, S. P. and Y. Demiris (2011). 'Echo state Gaussian process'. In: *IEEE Transactions on Neural Networks* 22.9, pp. 1435–1445. DOI: 10.1109/tnn.2011.2162109 (cit. on p. 127).

Chatzis, S. P., D. Korkinof, Y. Demiris and S. Member (2012). 'A Quantum-Statistical Approach Toward Robot Learning by Demonstration'. In: *IEEE Transactions on Robotics* 28.6, pp. 1371–1381. DOI: 10.1109/tro.2012.2203055 (cit. on p. 31).

Chen, C., R. Jafari and N. Kehtarnavaz (2016). 'A real-time human action recognition system using depth and inertial sensor fusion'. In: *IEEE Sensors Journal* 16.3, pp. 773–781 (cit. on p. 37).

Cho, H., Y.-W. Seo, B. V. Kumar and R. R. Rajkumar (2014). 'A multi-sensor fusion system for moving object detection and tracking in urban driving environments'. In: *IEEE International Conference on Robotics and Automation*. IEEE, pp. 1836–1843 (cit. on p. 37).

Cho, K., A. Courville and Y. Bengio (2015). 'Describing multimedia content using attention-based encoder-decoder networks'. In: *IEEE Transactions on Multimedia* 17.11, pp. 1875–1886 (cit. on p. 37).

Choi, Y., S.-Y. Cheong and N. Schweighofer (2007). 'Local online support vector regression for learning control'. In: *International Sym-*

posium on Computational Intelligence in Robotics and Automation. IEEE, pp. 13–18. DOI: 10.1109/cira.2007.382883 (cit. on p. 35).

Choraś, M. et al. (2009). 'Ensemble Learning'. In: *Encyclopedia of Biometrics*. Springer, pp. 270–273. DOI: 10.1007/978-0-387-73003-5_293 (cit. on p. 43).

Copete, J. L., Y. Nagai and M. Asada (2016). 'Motor development facilitates the prediction of others' actions through sensorimotor predictive learning'. In: *Proceedings of the Joint IEEE International Conference on Development and Learning and Epigenetic Robotics*. IEEE, pp. 223–229. DOI: 10.1109/devlrn.2016.7846823 (cit. on pp. 33, 39, 68, 94, 95, 102).

Cruz, J. and R. M. Gordon (2003). 'Simulation Theory'. In: *Encyclopedia of Cognitive Sciences*, pp. 9–14. DOI: 10.1002/wcs.33 (cit. on pp. 27, 92, 102).

Cully, A., J. Clune, D. Tarapore and J.-B. Mouret (2015). 'Robots that can adapt like animals'. In: *Nature* 521.7553, pp. 1–26. DOI: 10.1038/nature14422 (cit. on p. 42).

D'Souza, A., S. Vijayakumar and S. Schaal (2001). 'Learning inverse kinematics'. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium.* Vol. 1. 1. IEEE, pp. 298–303. DOI: 10.1109/IROS.2001.973374 (cit. on p. 31).

Damas, B., L. Jamone and J. Santos-Victor (2013). 'Open and closed-loop task space trajectory control of redundant robots using learned models'. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. IEEE, pp. 163–169. DOI: 10.1109/IROS.2013.6696348 (cit. on p. 31).

Dearden, A. and Y. Demiris (2005). 'Learning forward models for robots'. In: *Proceedings of International Joint Conference on Artificial Intelligence*. Vol. 5, pp. 1440–1445 (cit. on pp. 30, 38).

Dearden, A. and Y. Demiris (2007). 'From exploration to imitation: using learnt internal models to imitate others'. In: *Proceedings of the International Symposium on Imitation in Animals and Artifacts* (cit. on pp. 34, 37).

Decety, J. and J. A. Sommerville (2003). 'Shared representations between self and other: a social cognitive neuroscience view'. In: *Trends in*

cognitive sciences 7.12, pp. 527–533. DOI: 10.1016/j.tics.2003.10.004 (cit. on pp. 27, 115).

Deisenroth, M. P., D. Fox and C. E. Rasmussen (2015). 'Gaussian processes for data-efficient learning in robotics and control'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.2, pp. 408–423. DOI: 10.1109/TPAMI.2013.218 (cit. on pp. 32, 68).

Deisenroth, M. and C. E. Rasmussen (2011). 'PILCO: A model-based and data-efficient approach to policy search'. In: *Proceedings of the International Conference on machine learning*, pp. 465–472 (cit. on p. 30).

Demiris, Y., L. Aziz-Zadeh and J. Bonaiuto (2014a). 'Information processing in the mirror neuron system in primates and machines'. In: *Neuroinformatics* 12.1, pp. 63–91. DOI: 10.1007/s12021-013-9200-7 (cit. on p. 27).

Demiris, Y., L. Aziz-Zadeh and J. Bonaiuto (2014b). 'Information processing in the mirror neuron system in primates and machines'. In: *Neuroinformatics* 12.1, pp. 63–91. DOI: 10.1007/s12021-013-9200-7 (cit. on p. 74).

Demiris, Y. and A. Dearden (2005). *From motor babbling to hierarchical learning by imitation: a robot developmental pathway* (cit. on pp. 34, 38, 42, 68).

Demiris, Y. and B. Khadhouri (2006). 'Hierarchical attentive multiple models for execution and recognition of actions'. In: *Robotics and Autonomous Systems* 54.5, pp. 361–369. DOI: 10.1016/j.robot.2006.02.003 (cit. on pp. 34, 42, 74).

Demiris, Y. and A. Meltzoff (2008). 'The robot in the crib: A developmental analysis of imitation skills in infants and robots'. In: *Infant and Child Development* 17.1, pp. 43–53. DOI: 10.1002/icd.543 (cit. on pp. 34, 42).

Dietrich, V., D. Chen, K. M. Wurm, G. v. Wichert and P. Ennen (2016). 'Probabilistic multi-sensor fusion based on signed distance functions'. In: *IEEE International Conference on Robotics and Automation*. IEEE, pp. 1873–1878 (cit. on p. 37).

Dietterich, T. G. (2000). 'Ensemble methods in machine learning'. In: *International workshop on multiple classifier systems*. Springer, pp. 1–15 (cit. on pp. 40, 43).

Doersch, C. (2016). 'Tutorial on variational autoencoders'. In: *arXiv preprint arXiv:1606.05908* (cit. on p. 95).

Droniou, A., S. Ivaldi and O. Sigaud (2015). 'Deep unsupervised network for multimodal perception, representation and classification'. In: *Robotics and Autonomous Systems* 71, pp. 83–98. DOI: 10.1016/j.robot.2014.11.005 (cit. on pp. 33, 94, 95).

Fabbri-Destro, M. and G. Rizzolatti (2008). 'Mirror neurons and mirror systems in monkeys and humans'. In: *Physiology* 23.3, pp. 171–179. DOI: 10.1152/physiol.00004.2008 (cit. on p. 27).

Fabisch, A. Kassahun, Y. Wöhrle, H. Kirchner, F. (2014). *OpenANN*. URL: http://openann.github.io/OpenANN-apidoc/ (cit. on p. 60).

Fadlil, M., K. Ikeda, K. Abe, T. Nakamura and T. Nagai (2013). 'Integrated concept of objects and human motions based on multilayered multimodal LDA'. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. IEEE, pp. 2256–2263. DOI: 10.1109/IROS.2013.6696672 (cit. on p. 38).

Fischer, T. and Y. Demiris (2016). 'Markerless Perspective Taking for Humanoid Robots in Unconstrained Environments Markerless Perspective Taking for Humanoid Robots in Unconstrained Environments'. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, pp. 3309–3316. DOI: 10.1109/icra.2016.7487504 (cit. on pp. 80, 102).

Freund, Y. and R. Schapire (1997). 'A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting'. In: *Journal of Computer and System Sciences* 55.1, pp. 119–139. DOI: 10.1006/jcss.1997.1504 (cit. on p. 44).

Fuke, S., M. Ogino and M. Asada (2007). 'Body Image Constructed From Motor and Tactile Images With Visual Information'. In: *International Journal of Humanoid Robotics* 04.02, pp. 347–364. DOI: 10.1142/S0219843607001096 (cit. on pp. 37, 68).

Gallese, V. and A. Goldman (1998). 'Mirror neurons and the simulation theory of mind-reading'. In: *Trends in cognitive sciences* 2.12, pp. 493–501 (cit. on pp. 27, 28, 92, 102).

Gan, C., N. Wang, Y. Yang, D.-Y. Yeung and A. G. Hauptmann (2015). 'Devnet: A deep event network for multimedia event detection

and evidence recounting'. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2568–2577 (cit. on p. 37).

Gislason, P. O., J. A. Benediktsson and J. R. Sveinsson (2006). 'Random forests for land cover classification'. In: *Pattern Recognition Letters* 27.4, pp. 294–300 (cit. on p. 43).

Gravina, R., P. Alinia, H. Ghasemzadeh and G. Fortino (2017). 'Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges'. In: *Information Fusion* 35, pp. 68–80 (cit. on p. 37).

Hadders-Algra, M (2000). 'The neuronal group selection theory: a framework to explain variation in normal motor development.' In: *Developmental medicine and child neurology* 42.8, pp. 566–572. DOI: 10.1111/j.1469-8749.2000.tb00714.x (cit. on pp. 42, 53).

Harding, S., M. Frank, F Alexander, S. Italiana, S. Italiana, J. Leitner, S. Harding, M. Frank, A. Förster and J. Schmidhuber (2012). 'Towards Spatial Perception: Learning to Locate Objects From Vision'. In: *Proceedings of the Post-Graduate Conference on Robotics and Development of Cognition*, pp. 20–23 (cit. on p. 36).

Haruno, M, D. M. Wolpert and M. Kawato (2001). 'Mosaic model for sensorimotor learning and control.' In: *Neural computation* 13.10, pp. 2201–20. DOI: 10.1162/089976601750541778 (cit. on pp. 28, 42, 74).

Haruno, M., D. M. Wolpert and M. Kawato (2003). 'Hierarchical MOSAIC for movement generation'. In: *International congress series*. Vol. 1250. Elsevier, pp. 575–590. DOI: 10.1016/s0531-5131(03)00190-0 (cit. on p. 34).

Hayes, G. M. and J. Demiris (1994). *A robot controller using learning by imitation* (cit. on p. 68).

Heinrich, S., P. Folleher, P. Springstübe, E. Strahl, J. Twiefel, C. Weber and S. Wermter (2013). 'Object Learning with Natural Language in a Distributed Intelligent System: A Case Study of Human-Robot Interaction'. In: *Advances in Intelligent Systems and Computing*. Springer Berlin Heidelberg, pp. 811–819. DOI: 10.1007/978-3-642-37835-5_70 (cit. on p. 43).

Hersch, M., E. Sauser and A. Billard (2008). 'Online Learning of the Body Schema'. In: *International Journal of Humanoid Robotics* 5.2, pp. 161–181. DOI: 10.1142/S0219843608001376 (cit. on pp. 34, 35).

Hesslow, G. (2012). 'The current status of the simulation theory of cognition'. In: *Brain research* 1428, pp. 71–79. DOI: 10.1016/j.brainres.2011.06.026 (cit. on pp. 27, 92, 102).

Hinton, G. E., S. Osindero and Y.-W. Teh (2006). 'A Fast Learning Algorithm for Deep Belief Nets'. In: *Neural Computation* 18.7, pp. 1527–1554. DOI: 10.1162/neco.2006.18.7.1527 (cit. on p. 32).

Hoffmann, H. (2007). 'Perception through visuomotor anticipation in a mobile robot'. In: *Neural Networks* 20.1, pp. 22–33. DOI: 10.1016/j.neunet.2006.07.003 (cit. on p. 35).

Hoffmann, M., H. G. Marques, A Hernandez Arieta, H. Sumioka, M. Lungarella and R. Pfeifer (2010). 'Body Schema in Robotics: A Review'. In: *IEEE Transactions on Autonomous Mental Development*, 2.4, pp. 304–324. DOI: 10.1109/tamd.2010.2086454 (cit. on p. 33).

Hofsten, C. von (2004). 'An action perspective on motor development'. In: *Trends in Cognitive Science* 8.6, pp. 266–272. DOI: 10.1016/j.tics.2004.04.002 (cit. on p. 34).

Hu, N., G. Englebienne, Z. Lou and B. Kröse (2015). 'Latent hierarchical model for activity recognition'. In: *IEEE Transactions on Robotics* 31.6, pp. 1472–1482. DOI: 10.1109/tro.2015.2495002 (cit. on p. 92).

Hunt, K. J., D Sbarbaro, R Żbikowski and P. J. Gawthrop (1992). 'Neural networks for control systems: a survey'. In: *Automatica* 28.6, pp. 1083–1112. DOI: 10.1016/0005-1098(92)90053-i (cit. on pp. 30, 39).

Ikonomovska, E., J. Gama and S. Džeroski (2015). 'Online tree-based ensembles and option trees for regression on evolving data streams'. In: *Neurocomputing* 150.PB, pp. 458–470. DOI: 10.1016/j.neucom.2014.04.076 (cit. on p. 43).

Ishikawa, T., S. Tomatsu, J. Izawa and S. Kakei (2016). 'The cerebro-cerebellum: Could it be loci of forward models?' In: *Neuroscience research* 104, pp. 72–79. DOI: 10.1016/j.neures.2015.12.003 (cit. on pp. 25, 26).

Izawa, J., S. E. Criscimagna-Hemminger and R. Shadmehr (2012). 'Cerebellar contributions to reach adaptation and learning sensory con-

sequences of action'. In: *Journal of Neuroscience* 32.12, pp. 4230–4239. DOI: 10.1523/jneurosci.6353-11.2012 (cit. on p. 26).

Jaeger, H. (2002). 'Adaptive nonlinear system identification with echo state networks'. In: *Advances in neural information processing systems*, pp. 593–600 (cit. on pp. 46, 47, 125, 126).

Jamone, L., L. Natale, K. Hashimoto, G. Sandini and A. Takanishi (2011). 'Learning task space control through goal directed exploration'. In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics*. IEEE, pp. 702–708. DOI: 10.1109/ROBIO.2011.6181368 (cit. on p. 34).

Jamone, L., L. Natale, F. Nori, G. Metta and G. Sandini (2012). 'Autonomous online learning of reaching behavior in a humanoid robot'. In: *International Journal of Humanoid Robotics* 9.03, p. 1250017. DOI: 10.1142/s021984361250017x (cit. on pp. 34, 35).

Jamone, L., B. Damas, J. Santos-Victor and A. Takanishi (2013). 'Online learning of humanoid robot kinematics under switching tools contexts'. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, pp. 4811–4817. DOI: 10.1109/ICRA.2013.6631263 (cit. on p. 34).

Johnsson, M. and C. Balkenius (2011). 'Sense of touch in robots with self-organizing maps'. In: *IEEE Transactions on Robotics* 27.3, pp. 498–507. DOI: 10.1109/TRO.2011.2130090 (cit. on p. 38).

Jordan, M. I. and D. E. Rumelhart (1992). 'Forward models: Supervised learning with a distal teacher'. In: *Cognitive Science* 16.3, pp. 307–354. DOI: 10.1016/0364-0213(92)90036-T (cit. on pp. 30, 45, 68).

Jung, H., J. Ju and J. Kim (2014). 'Rigid Motion Segmentation Using Randomized Voting'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1210–1217. DOI: 10.1109/CVPR.2014.158 (cit. on p. 134).

Kajic, I., G. Schillaci, S. Bodiroža and V. V. Hafner (2014). 'A Biologically Inspired Model for Coding Sensorimotor Experience Leading to the Development of Pointing Behaviour in a Humanoid Robot'. In: *Proceedings of the Workshop "HRI: a bridge between Robotics and Neuroscience", ACM/IEEE International Conference on Human-Robot Interaction* (cit. on p. 36).

Kang, C., S. Xiang, S. Liao, C. Xu and C. Pan (2015). 'Learning consistent feature representation for cross-modal multimedia retrieval'. In: *IEEE Transactions on Multimedia* 17.3, pp. 370–381 (cit. on p. 37).

Kawato, M. (1990). 'Feedback-error-learning neural network for supervised motor learning'. In: *Advanced neural computers* 6.3, pp. 365–372. DOI: 10.1016/b978-0-444-88400-8.50047-9 (cit. on p. 30).

Kawato, M. (1999). In: *Current opinion in neurobiology* 9.6, pp. 718–727. DOI: 10.1016/s0959-4388(99)00028-8 (cit. on pp. 28, 34, 42).

Kawato, M. and H. Gomi (1992). 'The cerebellum and VOR/OKR learning models'. In: *Trends in neurosciences* 15.11, pp. 445–453. DOI: 10.1016/0166-2236(92)90008-v (cit. on p. 28).

Kawato, M., Y. Uno, M. Isobe and R. Suzuki (1988). 'Hierarchical neural network model for voluntary movement with application to robotics'. In: *IEEE Control Systems Magazine* 8.2, pp. 8–15. DOI: 10.1109/37.1867 (cit. on p. 31).

Kilner, J. M., K. J. Friston and C. D. Frith (2007). 'Predictive coding: an account of the mirror neuron system'. In: *Cognitive Processing* 8.3, pp. 159–166. DOI: 10.1007/s10339-007-0170-2. URL: https://doi.org/10.1007%2Fs10339-007-0170-2 (cit. on pp. 27, 99).

Kingma, D. P. and M. Welling (2013). 'Auto-encoding variational bayes'. In: *arXiv preprint arXiv:1312.6114* (cit. on pp. 40, 95, 96, 98).

Klanke, S., S. Vijayakumar and S. Schaal (2008). 'A Library for Locally Weighted Projection Regression'. In: *Journal of Machine Learning Research* 9.1, pp. 623–626 (cit. on p. 129).

Korkinof, D. and Y. Demiris (2013). 'Online Quantum Mixture Regression for Trajectory Learning by Demonstration'. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3222–3229. DOI: 10.1109/iros.2013.6696814 (cit. on p. 31).

Kormushev, P., S. Calinon and D. Caldwell (2013). 'Reinforcement Learning in Robotics: Applications and Real-World Challenges'. In: *Robotics* 2.3, pp. 122–148. DOI: 10.3390/robotics2030122 (cit. on pp. 32, 68).

Kormushev, P., Y. Demiris and D. G. Caldwell (2015). 'Encoderless position control of a two-link robot manipulator'. In: *Proceedings of*

the IEEE International Conference on Robotics and Automation. IEEE, pp. 943–949. DOI: 10.1109/ICRA.2015.7139290 (cit. on p. 72).

Kuderer, M., H. Kretzschmar, C. Sprunk and W. Burgard (2012). 'Feature-Based Prediction of Trajectories for Socially Compliant Navigation.' In: *Robotics: science and systems*. Robotics: Science and Systems Foundation. DOI: 10.15607/rss.2012.viii.025 (cit. on p. 92).

Kuncheva, L. I. and J. J. Rodríguez (2014). 'A weighted voting framework for classifiers ensembles'. In: *Knowledge and Information Systems* 38.2, pp. 259–275 (cit. on p. 43).

Kuncheva, L. I. and C. J. Whitaker (2003). 'Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy'. In: *Machine learning* 51.2, pp. 181–207 (cit. on p. 46).

Kuperstein, M. (1988). 'Neural model of adaptive hand-eye coordination for single postures'. In: *Science* 239.4845, pp. 1308–1312. DOI: 10.1126/science.3344437 (cit. on p. 30).

Lazkano, E., B. Sierra, A. Astigarraga and J. M. Martinez-Otzeta (2007). 'On the use of Bayesian Networks to develop behaviours for mobile robots'. In: *Robotics and Autonomous Systems* 55.3, pp. 253–265. DOI: 10.1016/j.robot.2006.08.003 (cit. on p. 28).

LeCun, Y., Y. Bengio and G. Hinton (2015). 'Deep learning'. In: *Nature* 521.7553, pp. 436–444 (cit. on p. 32).

Lee, K., T. K. Kim and Y. Demiris (2012). 'Learning reusable task components using hierarchical activity grammars with uncertainties'. In: *Proceedings in the IEEE International Conference on Robotics and Automation*. IEEE, pp. 1994–1999. DOI: 10.1109/ICRA.2012.6224667 (cit. on p. 32).

Lee, K., Y. Su, T. K. Kim and Y. Demiris (2013). 'A syntactic approach to robot imitation learning using probabilistic activity grammars'. In: *Robotics and Autonomous Systems* 61.12, pp. 1323–1334. DOI: 10.1016/j.robot.2013.08.003 (cit. on p. 32).

Leitner, J., S. Harding, M. Frank, A. Förster and J. Schmidhuber (2012). 'Learning Spatial Object Localization from Vision on a Humanoid Robot'. In: *International Journal of Advanced Robotic Systems* 9.6, p. 243. DOI: 10.5772/54657 (cit. on p. 36).

Levine, S., N. Wagener and P. Abbeel (2015). 'Learning Contact-Rich Manipulation Skills with Guided Policy Search'. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, pp. 156–163. DOI: 10.1109/ICRA.2015.7138994 (cit. on pp. 32, 68).

Levine, S., C. Finn, T. Darrell and P. Abbeel (2016). 'End-to-end training of deep visuomotor policies'. In: *Journal of Machine Learning Research* 17.39, pp. 1–40 (cit. on p. 32).

Littlestone, N. and M. Warmuth (1994). 'The Weighted Majority Algorithm'. In: *Information and Computation* 108.2, pp. 212–261. DOI: 10.1006/inco.1994.1009 (cit. on p. 44).

Ljung, L. (1983). *Theory and Practice of Recursive Identification.pdf*. Vol. 21. 4. MIT Press Cambridge, MA, pp. 499–501 (cit. on pp. 46, 47, 128, 129).

Ljung, L. (1998). 'System identification'. In: *Signal analysis and prediction*. Springer, pp. 163–173 (cit. on pp. 46, 128).

Lopes, M. and J. Santos-Victor (2005). 'Visual learning by imitation with motor representations'. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 35.3, pp. 438–449. DOI: 10.1109/TSMCB.2005.846654 (cit. on p. 38).

Lungarella, M., G. Metta, R. Pfeifer and G. Sandini (2003). 'Developmental robotics: a survey'. In: *Connection Science* 15.4, pp. 151–190. DOI: 10.1080/09540090310001655110 (cit. on p. 33).

Luo, H. and R. E. Schapire (2015). 'Achieving all with no parameters: Adaptive NormalHedge'. In: *Proceedings of the 28th Conference on Learning Theory*, pp. 1–19 (cit. on pp. 44, 50, 65).

Lyon, C., C. L. Nehaniv and J. Saunders (2012). 'Interactive language learning by robots: The transition from babbling to word forms'. In: *PLoS One* 7.6, e38236. DOI: 10.1371/journal.pone.0038236 (cit. on p. 124).

Lyon, C., C. L. Nehaniv, J. Saunders, T. Belpaeme, A. Bisio, K. Fischer, F. Förster, H. Lehmann, G. Metta, V. Mohan et al. (2016). 'Embodied language learning and cognitive bootstrapping: methods and design principles'. In: *International Journal of Advanced Robotic Systems* 13.3, p. 105. DOI: 10.5772/63462 (cit. on p. 124).

Maestre, C., A. Cully, C. Gonzales and S. Doncieux (2015). 'Bootstrapping interactions with objects from raw sensorimotor data: a Nov-

elty Search based approach'. In: *Proceedings of the Joint IEEE International Conference on Development and Learning and Epigenetic Robotics*. IEEE, pp. 7–12. DOI: 10.1109/devlrn.2015.7346098 (cit. on p. 120).

Marshall, P. J. and A. N. Meltzoff (2015). 'Body maps in the infant brain'. In: *Trends in Cognitive Sciences* 19.9, pp. 499–505. DOI: 10.1016/j.tics.2015.06.012 (cit. on pp. 25, 26).

Martinez, H., M. Lungarella and R. Pfeifer (2008). 'Stochastic extension of the attention-selection system for the iCub technical report'. In: *Artificial Intelligence*, pp. 1–7 (cit. on pp. 37, 69).

Maye, A. and A. K. Engel (2011). 'A discrete computational model of sensorimotor contingencies for object perception and control of behavior'. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, pp. 3810–3815. DOI: 10.1109/ICRA.2011.5979919 (cit. on p. 45).

Meier, F. and S. Schaal (2016). 'Drifting Gaussian processes with varying neighborhood sizes for online model learning'. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 2016-June. IEEE, pp. 264–269. DOI: 10.1109/ICRA.2016.7487143 (cit. on pp. 44, 52, 56).

Mendes-Moreira, J., C. Soares, A. M. Jorge and J. F. D. Sousa (2012). 'Ensemble approaches for regression'. In: *ACM Computing Surveys* 45.1, pp. 1–40. DOI: 10.1145/2379776.2379786 (cit. on pp. 43, 45).

Metta, G., P. Fitzpatrick and L. Natale (2006). 'YARP: Yet Another Robot Platform'. In: *International Journal of Advanced Robotic Systems* 3.1, p. 8. DOI: 10.5772/5761 (cit. on p. 137).

Metta, G. et al. (2010). 'The iCub humanoid robot: An open-systems platform for research in cognitive development'. In: *Neural Networks* 23.8-9, pp. 1125–1134. DOI: 10.1016/j.neunet.2010.08.010 (cit. on p. 137).

Miall, R. C. and D. M. Wolpert (1996). 'Forward models for physiological motor control'. In: *Neural Networks* 9.8, pp. 1265–1279. DOI: 10.1016/S0893-6080(96)00035-4 (cit. on pp. 25, 26).

Miall, R. C., L. O. Christensen, O. Cain and J. Stanley (2007). 'Disruption of state estimation in the human lateral cerebellum'. In: *PLoS biology* 5.11, e316 (cit. on p. 26).

Miller, W. (1987). 'Sensor-based control of robotic manipulators using a general learning algorithm'. In: *IEEE Journal on Robotics and Automation* 3.2, pp. 157–165. DOI: 10.1109/jra.1987.1087081 (cit. on p. 30).

Mochizuki, K., S. Nishide, H. G. Okuno and T. Ogata (2013). 'Developmental human-robot imitation learning of drawing with a neuro dynamical system'. In: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, pp. 2336–2341. DOI: 10.1109/SMC.2013.399 (cit. on pp. 42, 68).

Mordatch, I., N. Mishra, C. Eppner and P. Abbeel (2016). 'Combining model-based policy search with online model learning for control of physical humanoids'. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 2016-June. IEEE, pp. 242–248. DOI: 10.1109/ICRA.2016.7487140 (cit. on pp. 32, 68).

Narendra, K. and J. Balakrishnan (1997). 'Adaptive control using multiple models'. In: *IEEE Transactions on Automatic Control* 42.2, pp. 171–187. DOI: 10.1109/9.554398 (cit. on p. 28).

Ngiam, J., A. Khosla, M. Kim, J. Nam, H. Lee and A. Y. Ng (2011). 'Multimodal Deep Learning'. In: *Proceedings of the International Conference on Machine Learning*, pp. 689–696 (cit. on pp. 33, 39, 93, 94, 96).

Nguyen-Tuong, D. and J. Peters (2011). 'Model learning for robot control: a survey'. In: *Cognitive processing* 12.4, pp. 319–340 (cit. on pp. 30, 34).

Nguyen-Tuong, D., M. Seeger and J. Peters (2009). 'Model learning with Local Gaussian Process Regression'. In: *Advanced Robotics* 23.15, pp. 2015–2034. DOI: 10.1163/016918609x12529286896877 (cit. on p. 35).

Nguyen, L, R. Patel and K Khorasani (1990). 'Neural network architectures for the forward kinematics problem in robotics'. In: *Proceedings of the International Joint Conference on Neural Networks*. IEEE, pp. 393–399. DOI: 10.1109/ijcnn.1990.137874 (cit. on p. 29).

Nowak, D. A., H. Topka, D. Timmann, H. Boecker and J. Hermsdörfer (2007). 'The role of the cerebellum for predictive control of grasping'. In: *The Cerebellum* 6.1, pp. 7–17. DOI: 10.1080/14734220600776379 (cit. on p. 26).

Nyga, D., F. Balint-Benczedi and M. Beetz (2014). 'PR2 looking at thingsâ€"Ensemble learning for unstructured information processing with Markov logic networks'. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, pp. 3916–3923. DOI: 10.1109/icra.2014.6907427 (cit. on p. 43).

Olfati-Saber, R. and J. S. Shamma (2005). 'Consensus filters for sensor networks and distributed sensor fusion'. In: *IEEE Conference on Decision and Control - European Control Conference*. IEEE, pp. 6698–6703 (cit. on p. 37).

Onan, A., S. Korukoğlu and H. Bulut (2016). 'Ensemble of keyword extraction methods and classifiers in text classification'. In: *Expert Systems with Applications* 57, pp. 232–247 (cit. on p. 43).

Pezzulo, G. (2007). *Anticipation and future-oriented capabilities in natural and artificial cognition*. Springer, pp. 257–270. DOI: 10.1007/978-3-540-77296-5_24 (cit. on p. 26).

Pickering, M. J. and A. Clark (2014). 'Getting ahead: Forward models and their place in cognitive architecture'. In: 9, pp. 451–456. DOI: 10.1016/j.tics.2014.05.006 (cit. on pp. 27, 92).

Popa, L. S., A. L. Hewitt and T. J. Ebner (2013). 'Purkinje cell simple spike discharge encodes error signals consistent with a forward internal model'. In: *The Cerebellum* 12.3, pp. 331–333. DOI: 10.1007/s12311-013-0452-4 (cit. on p. 26).

Poria, S., E. Cambria, N. Howard, G.-B. Huang and A. Hussain (2016). 'Fusing audio, visual and textual clues for sentiment analysis from multimodal content'. In: *Neurocomputing* 174, pp. 50–59 (cit. on p. 37).

Quigley, M., K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng (2009). 'ROS: an open-source Robot Operating System'. In: *Workshop on Open Source Software, International Conference on Robotics and Automation* (cit. on p. 137).

Ramırez-Contla, S., A. Cangelosi and D. Marocco (2012). 'Developing motor skills for reaching by progressively unlocking degrees of freedom on the iCub humanoid robot'. In: *Proceedings of the Post-Graduate Conference on Robotics and Development of Cognition*, pp. 10–12 (cit. on p. 34).

Ramisa, A., F. Yan, F. Moreno-Noguer and K. Mikolajczyk (2017). 'Breakingnews: Article annotation by image and text processing'. In: *IEEE transactions on pattern analysis and machine intelligence* (cit. on p. 37).

Ranawana, R. and V. Palade (2006). 'Multi-classifier systems: Review and a roadmap for developers'. In: *International Journal of Hybrid Intelligent Systems* 3.1, pp. 35–61 (cit. on p. 43).

Rasmussen, C. E. and C. K. Williams (2006). *Gaussian processes for machine learning*. Vol. 1. MIT press Cambridge (cit. on p. 127).

Ren, Y., L. Zhang and P. N. Suganthan (2016). 'Ensemble classification and regression-recent developments, applications and future directions'. In: *IEEE Computational intelligence magazine* 11.1, pp. 41–53 (cit. on p. 43).

Rethink Robotics. *Baxter - Redefining Robotics and Manufacturing - Rethink Robotics*. URL: \url{http://www.rethinkrobotics.com/baxter/} (cit. on p. 137).

Rizzolatti, G. (2005). 'The mirror neuron system and its function in humans'. In: *Anatomy and embryology* 210.5-6, pp. 419–421. DOI: 10.1007/s00429-005-0039-z (cit. on p. 27).

Rolf, M., J. J. Steil and M. Gienger (2010). 'Goal babbling permits direct learning of inverse kinematics'. In: *IEEE Transactions on Autonomous Mental Development* 2.3, pp. 216–229. DOI: 10.1109/tamd.2010.2062511 (cit. on p. 34).

Rolf, M., J. J. Steil and M. Gienger (2011). 'Online Goal Babbling for rapid bootstrapping of inverse models in high dimensions'. In: *Proceedings of the IEEE International Conference on Development and Learning*. Vol. 2. IEEE, pp. 1–8. DOI: 10.1109/DEVLRN.2011.6037368 (cit. on p. 34).

Roncone, A., M. Hoffmann, U. Pattacini and G. Metta (2014). 'Automatic kinematic chain calibration using artificial skin: Self-touch in the iCub humanoid robot'. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, pp. 2305–2312. DOI: 10.1109/ICRA.2014.6907178 (cit. on pp. 37, 68).

Rooney, N., D. Patterson, S. Anand and A. Tsymbal (2004). 'Dynamic integration of regression models'. In: *Proceedings of the International Workshop on Multiple Classifier Systems. Lecture Notes in Computer*

Science. Springer, pp. 164–173. DOI: 10.1007/978-3-540-25966-4_16 (cit. on p. 46).

Ross, D. A., D. Tarlow and R. S. Zemel (2010). 'Learning Articulated Structure and Motion.' In: *International Journal of Computer Vision* 88.2, pp. 214–237. DOI: 10.1007/s11263-010-0325-y (cit. on p. 36).

Sadjadian, H. and H. D. Taghirad (2005). 'Comparison of Different Methods for Computing the Forward Kinematics of a Redundant Parallel Manipulator'. In: *Journal of Intelligent and Robotic Systems* 44.3, pp. 225–246. DOI: 10.1007/s10846-005-9006-4 (cit. on p. 29).

Schaal, S., C. G. Atkeson and S. Vijayakumar (2002). 'Scalable techniques from nonparametric statistics for real time robot learning'. In: *Applied Intelligence* 17.1, pp. 49–60 (cit. on p. 35).

Schillaci, G. (2013). 'Sensorimotor Learning and Simulation of Experience as a Basis for the Development of Cognition in Robotics'. PhD thesis. Humboldt-Universität zu Berlin, Mathematisch - Naturwissenschaftliche Fakultät II (cit. on pp. 34, 37, 68).

Schillaci, G., V. V. Hafner and B. Lara (2014). 'Online learning of visuomotor coordination in a humanoid robot. A biologically inspired model'. In: *Proceedings of the Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*. IEEE, pp. 130–136. DOI: 10.1109/DEVLRN.2014.6982967 (cit. on p. 35).

Schmidts, A. M., D. Lee and A. Peer (2011). 'Imitation learning of human grasping skills from motion and force data'. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. IEEE, pp. 1002–1007. DOI: 10.1109/IROS.2011.6048638 (cit. on pp. 37, 38, 68, 80).

Schuller, B., S. Reiter, R. Muller, M. Al-Hames, M. Lang and G. Rigoll (2005). 'Speaker independent speech emotion recognition by ensemble classification'. In: *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, pp. 864–867 (cit. on p. 43).

Shotton, J., T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook and R. Moore (2013). 'Real-time human pose recognition in parts from single depth images'. In: *Communications of the ACM* 56.1, pp. 116–124. DOI: 10.1145/2398356.2398381 (cit. on p. 92).

Siciliano, B. and O. Khatib (2016). *Springer handbook of robotics*. Springer. DOI: 10.1007/978-3-319-32552-1_1 (cit. on pp. 29, 39).

Sigaud, O. and A. Droniou (2016). 'Towards Deep Developmental Learning'. In: *IEEE Transactions on Cognitive and Developmental Systems* 8.2, pp. 99–114. DOI: 10.1109/tamd.2015.2496248 (cit. on pp. 32, 33).

Soares, S. G. and R. Araújo (2015). 'A dynamic and on-line ensemble regression for changing environments'. In: *Expert Systems with Applications* 42.6, pp. 2935–2948. DOI: 10.1016/j.eswa.2014.11.053 (cit. on p. 44).

Soh, H. and Y. Demiris (2015). 'Spatio-temporal learning with the Online Finite and Infinite Echo-state Gaussian Processes'. In: *IEEE Transactions on Neural Networks and Learning Systems* 26.3, pp. 522–536. DOI: 10.1109/tnnls.2014.2316291 (cit. on pp. 31, 46, 47).

Soh, H. and Y. Demiris (2012). 'Iterative temporal learning and prediction with the sparse online echo state gaussian process'. In: *Proceedings of the International Joint Conference on Neural Networks*. IEEE, pp. 1–8. DOI: 10.1109/IJCNN.2012.6252504 (cit. on p. 127).

Soh, H. and Y. Demiris (2013). 'When and how to help: An iterative probabilistic model for learning assistance by demonstration'. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. IEEE, pp. 3230–3236. DOI: 10.1109/IROS.2013.6696815 (cit. on p. 31).

Spencer, J. and P. C. Quinn (1997). 'Explaining facial imitation: A theoretical model'. In: *Early Development and Parenting* 6, pp. 113–126. DOI: 10.1002/(SICI)1099-0917(199709/12)6 (cit. on pp. 28, 42).

Stepanova, K., M. Hoffmann, Z. Straka, F. B. Klein, A. Cangelosi and M. Vavrecka (2017). 'Where is my forearm? Clustering of body parts from simultaneous tactile and linguistic input using sequential mapping'. In: *arXiv preprint arXiv:1706.02490* (cit. on p. 37).

Stulp, F. and O. Sigaud (2013). 'Robot Skill Learning: From Reinforcement Learning to Evolution Strategies'. In: *Paladyn, Journal of Behavioral Robotics* 4.1, pp. 49–61. DOI: 10.2478/pjbr-2013-0003 (cit. on p. 68).

Sturm, J., C. Plagemann and W. Burgard (2009). 'Body schema learning for robotic manipulators from visual self-perception'. In: *Journal of Physiology Paris* 103.3-5, pp. 220–231. DOI: 10.1016/j.jphysparis.2009.08.005 (cit. on pp. 36, 37).

Sugimoto, N., J. Morimoto, S. H. Hyon and M. Kawato (2012). 'The eMOSAIC model for humanoid robot control'. In: *Neural Networks* 29-30, pp. 8–19. DOI: 10.1016/j.neunet.2012.01.002 (cit. on p. 34).

Sun, S., C. Zhang and D. Zhang (2007). 'An experimental evaluation of ensemble methods for EEG signal classification'. In: *Pattern Recognition Letters* 28.15, pp. 2157–2163 (cit. on p. 43).

Sun, Z., Q. Song, X. Zhu, H. Sun, B. Xu and Y. Zhou (2015). 'A novel ensemble method for classifying imbalanced data'. In: *Pattern Recognition* 48.5, pp. 1623–1637 (cit. on p. 43).

Sutton, R. S. and A. G. Barto (1998). *Reinforcement learning: An introduction*. Vol. 9. 5. IEEE, pp. 1054–1054. DOI: 10.1109/tnn.1998.712192 (cit. on p. 30).

Tang, S., Y.-T. Zheng, Y. Wang and T.-S. Chua (2012). 'Sparse ensemble learning for concept detection'. In: *IEEE Transactions on Multimedia* 14.1, pp. 43–54 (cit. on p. 43).

Tani, J. and S. Nolfi (1999). 'Learning to perceive the world as articulated: An approach for hierarchical learning in sensory-motor systems'. In: *Neural Networks* 12.7-8, pp. 1131–1141. DOI: 10.1016/S0893-6080(99)00060-X (cit. on p. 35).

Tidemann, A., P. Öztürk and Y. Demiris (2009). 'A groovy virtual drumming agent'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 5773 LNAI. Springer, pp. 104–117. DOI: 10.1007/978-3-642-04380-2_14 (cit. on pp. 37, 38, 68, 80).

Valentini, G. and F. Masulli (2002). 'Ensembles of Learning Machines'. In: *Neural Nets*. Springer Berlin Heidelberg, pp. 3–20. DOI: 10.1007/3-540-45808-5_1 (cit. on p. 43).

Vijayakumar, S. and S. Schaal (2000a). 'Fast and efficient incremental learning for high-dimensionaln movement systems'. In: *Proceedings of the IEEE International Conference on Robotics and Automation. Symposia Proceedings*. Vol. 2. IEEE, pp. 1–6. DOI: 10.1109/ROBOT.2000.844871 (cit. on pp. 46, 47, 129).

Vijayakumar, S., A. D 'souza and S. Schaal (2005). 'Incremental Online Learning in High Dimensions'. In: *Neural Computation* 17.12, pp. 2602–2634. DOI: 10.1162/089976605774320557 (cit. on p. 129).

Vijayakumar, S. and S. Schaal (2000b). 'Locally Weighted Projection Regression: An O(n) Algorithm for Incremental Real Time Learning in High Dimensional Space'. In: *Proceedings of the International Conference on Machine Learning*. Vol. 1086, pp. 1079–1086. DOI: 10.1.1.33.4151 (cit. on pp. 35, 129).

Whitehead, M. and L. Yaeger (2010). 'Sentiment mining using ensemble classification models'. In: *Innovations and advances in computer sciences and engineering*. Springer, pp. 509–514 (cit. on p. 43).

Williams, C., S. Klanke, S. Vijayakumar and K. M. Chai (2009). 'Multi-task Gaussian process learning of robot inverse dynamics'. In: *Advances in Neural Information Processing Systems*, pp. 265–272 (cit. on p. 30).

Wolpert, D. M., Z Ghahrmani and M. I. Jordan (1995). 'An internal model for sensorimotor integration'. In: *Science* 269.5232, pp. 1880–1882. DOI: 10.1126/science.7569931 (cit. on p. 27).

Wolpert, D. M., J. Diedrichsen and J. R. Flanagan (2011). 'Principles of sensorimotor learning.' In: *Nature reviews. Neuroscience* 12.12, pp. 739–51. DOI: 10.1038/nrn3112 (cit. on pp. 25, 26).

Wolpert, D. M., K. Doya and M. Kawato (2003). 'A unifying computational framework for motor control and social interaction.' In: *Philosophical transactions of the Royal Society of London* 358, pp. 593–602. DOI: 10.1098/rstb.2002.1238 (cit. on p. 27).

Wolpert, D. M. and J. R. Flanagan (2001). 'Motor prediction.' In: *Current biology* 11.18, R729–R732. DOI: 10.1016/s0960-9822(01)00432-8 (cit. on pp. 25–28, 92, 115).

Wolpert, D. M. and M. Kawato (1998). 'Multiple paired forward and inverse models for motor control'. In: *Neural networks* 11.7, pp. 1317–1329. DOI: 10.1016/s0893-6080(98)00066-5 (cit. on pp. 26, 27, 30, 34, 45).

Wolpert, D. M., R. Miall and M. Kawato (1998). 'Internal models in the cerebellum'. In: *Trends in Cognitive Sciences* 2.9, pp. 338–347. DOI: 10.1016/s1364-6613(98)01221-2 (cit. on pp. 28, 33).

Wu, Y. and Y. Demiris (2009). 'Efficient Template-based Path Imitation by Invariant Feature Mapping'. In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics*. IEEE, pp. 913–918. DOI: 10.1109/robio.2009.5420496 (cit. on p. 32).

Wu, Y. and Y. Demiris (2010a). 'Hierarchical learning approach for one-shot action imitation in humanoid robots'. In: *Proceedings of the International Conference on Control, Automation, Robotics and Vision*. IEEE, pp. 453–458. DOI: 10.1109/ICARCV.2010.5707349 (cit. on pp. 32, 38).

Wu, Y. and Y. Demiris (2010b). 'Towards One Shot Learning by Imitation for Humanoid Robots'. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, pp. 2889–2894. DOI: 10.1109/robot.2010.5509429 (cit. on p. 32).

Wu, Y., Y. Su and Y. Demiris (2014). 'A Morphable Template Framework for Robot Learning by Demonstration: Integrating One-shot and Incremental Learning Approaches'. In: *Robotics and Autonomous Systems* 62.10, pp. 1517–1530. DOI: 10.1016/j.robot.2014.05.010 (cit. on p. 32).

Yan, F., J. Kittler, D. Windridge, W. Christmas, K. Mikolajczyk, S. Cox and Q. Huang (2014). 'Automatic annotation of tennis games: An integration of audio, vision, and learning'. In: *Image and Vision Computing* 32.11, pp. 896–903 (cit. on p. 37).

Yan, Y., Q. Zhu, M.-L. Shyu and S.-C. Chen (2016). 'A classifier ensemble framework for multimedia big data classification'. In: *Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on*. IEEE, pp. 615–622 (cit. on p. 43).

Yang, X., T. Zhang and C. Xu (2015). 'Cross-domain feature learning in multimedia'. In: *IEEE Transactions on Multimedia* 17.1, pp. 64–78 (cit. on p. 37).

Yijing, L., G. Haixiang, L. Xiao, L. Yanan and L. Jinling (2016). 'Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data'. In: *Knowledge-Based Systems* 94, pp. 88–104 (cit. on p. 43).

Yoshikawa, Y., H. Kawanishi, M. Asada and K. Hosoda (2002). 'Body Scheme Acquisition by Cross Modal Map Learning among Tactile, Visual, and Proprioceptive Spaces'. In: *Proceedings of the International Workshop on Epigenetic Robotics*, pp. 181–184 (cit. on pp. 37, 68).

Zhong, J., A. Cangelosi and T. Ogata (2017). 'Toward abstraction from multi-modal data: Empirical studies on multiple time-scale recurrent models'. In: *Proceedings of the International Joint Conference on Neural Networks*. IEEE. DOI: 10.1109/ijcnn.2017.7966312 (cit. on pp. 37, 38, 124).

Zhong, J., A. Cangelosi and S. Wermter (2014). 'Toward a self-organizing pre-symbolic neural model representing sensorimotor primitives'. In: *Frontiers in behavioral neuroscience* 8. DOI: 10.3389/fnbeh.2014.00022 (cit. on p. 34).

Zhong, J., A Cangelosi, T Ogata and C. Yang (2017). 'Understanding Natural Language Sentences with Word Embedding and Multi-modal Interaction'. In: *Proceedings of the International Conferene on Development and Learning and Epigenetic Robotics* (cit. on pp. 38, 124).