



HAL
open science

A Curious Robot Learner for Interactive Goal-Babbling: Strategically Choosing What, How, When and from Whom to Learn

Sao Mai Nguyen

► **To cite this version:**

Sao Mai Nguyen. A Curious Robot Learner for Interactive Goal-Babbling: Strategically Choosing What, How, When and from Whom to Learn. Other [cs.OH]. Université Sciences et Technologies - Bordeaux I, 2013. English. NNT : 2013BOR15223 . tel-00977385

HAL Id: tel-00977385

<https://theses.hal.science/tel-00977385>

Submitted on 11 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Curious Robot Learner for Interactive Goal-Babbling

*Strategically Choosing
What, How, When and from Whom to Learn*

A Dissertation Presented

by

Nguyen Sao Mai

Flowers Team, Inria

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of Cognitive Developmental Robotics

with the department of Computer Science, Universite de Bordeaux 1

Talence, France

27 November 2013

Thesis Committee:

| | | | |
|-------------------------|-----------------------------|----------------------------------|-----|
| Thesis Advisor | Dr. OUDEYER PIERRE-YVES | Inria - ENSTA | HdR |
| Thesis Rapporteur | Prof. Dr. DEMIRIS YIANNIS | Imperial College London | HdR |
| Thesis Rapporteur | Prof. Dr. GAUSSIER PHILIPPE | Université de Cergy-Pontoise | HdR |
| Committee President | Prof. Dr.-Ing WREDE BRITTA | Universität Bielefeld | HdR |
| Presentation Rapporteur | Dr. STASSE OLIVIER | LAAS - CNRS | HdR |
| Thesis Reader | Prof. SIGAUD OLIVIER | Université Pierre et Marie Curie | HdR |

Un Robot curieux pour l'apprentissage actif par babillage d'objectifs : choisir de manière stratégique quoi, comment, quand et de qui apprendre

Les défis pour voir des robots opérant dans l'environnement de tous les jours des humains et sur une longue durée soulignent l'importance de leur adaptation aux changements qui peuvent être imprévisibles au moment de leur construction. Ils doivent être capable de savoir quelles parties échantillonner, et quels types de compétences il a intérêt à acquérir. Une manière de collecter des données est de décider par soi-même où explorer. Une autre manière est de se référer à un mentor. Nous appelons ces deux manières de collecter des données des *modes d'échantillonnage*. Le premier mode d'échantillonnage correspond à des algorithmes développés dans la littérature pour automatiquement pousser l'agent vers des parties intéressantes de l'environnement ou vers des types de compétences utiles. De tels algorithmes sont appelés des algorithmes de *curiosité artificielle* ou *motivation intrinsèque*. Le deuxième mode correspond au *guidage social* ou *l'imitation*, où un partenaire humain indique où explorer et où ne pas explorer.

Nous avons construit une architecture algorithmique intrinsèquement motivée pour apprendre comment produire par ses actions des effets et conséquences variées. Il apprend de manière active et en ligne en collectant des données qu'il choisit en utilisant plusieurs modes d'échantillonnage. Au niveau du meta apprentissage, il apprend de manière active quelle stratégie d'échantillonnage est plus efficace pour améliorer sa compétence et généraliser à partir de son expérience à un grand éventail d'effets. Par apprentissage par interaction, il acquiert de multiples compétences de manière structurée, en découvrant par lui-même les séquences développementale.

Mots clés : apprentissage actif, apprentissage interactif, apprentissage par imitation, exploration orientée par objectifs, collecte de données, apprentissage par démonstration

A Curious Robot Learner for Interactive Goal-Babbling

The challenges posed by robots operating in human environments on a daily basis and in the long-term point out the importance of adaptivity to changes which can be unforeseen at design time. The robot must learn continuously in an open-ended, non-stationary and high dimensional space. It must be able to know which parts to sample and what kind of skills are interesting to learn. One way is to decide what to explore by oneself. Another way is to refer to a mentor. We name these two ways of collecting data *sampling modes*. The first sampling mode correspond to algorithms developed in the literature in order to autonomously drive the robot in interesting parts of the environment or useful kinds of skills. Such algorithms are called *artificial curiosity* or *intrinsic motivation* algorithms. The second sampling mode correspond to *social guidance* or *imitation* where the teacher indicates where to explore as well as where not to explore. Starting from the study of the relationships between these two concurrent methods, we ended up building an algorithmic architecture with a hierarchical learning structure, called **Socially Guided Intrinsic Motivation** (SGIM).

We have built an intrinsically motivated active learner which learns how its actions can produce varied consequences or outcomes. It actively learns online by sampling data which it chooses by using several sampling modes. On the meta-level, it actively learns which data collection strategy is most efficient for improving its competence and generalising from its experience to a wide variety of outcomes. The interactive learner thus learns multiple tasks in a structured manner, discovering by itself developmental sequences.

Keywords : active learning, interactive learning, imitation learning, goal-oriented exploration, data-collection, exploration, programming by demonstration

*Xin chân thành cảm ơn bố mẹ
Nguyễn Bá Hà
và
Đặng Thị Thu Oanh*

*Un Robot curieux pour l'apprentissage actif par babillage
d'objectifs : choisir de manière stratégique quoi, comment,
quand et de qui apprendre*

ABSTRACT

Les défis pour voir des robots opérant dans l'environnement de tous les jours des humains et sur une longue durée soulignent l'importance de leur adaptation aux changements qui peuvent être imprévisibles au moment de leur construction. C'est pourquoi, les robots doivent être capables d'apprendre continuellement dans des espaces infinis, non-stationnaires et de grande dimension. Il leur est impossible d'explorer tout son environnement pour apprendre pendant la durée limitée de sa vie. Pour être utile et acquérir des compétences, le robot doit au contraire être capable de savoir quelles parties échantillonner, et quels types de compétences il a intérêt à acquérir. Une manière de collecter des données est de décider par soi-même où explorer. Une autre manière est de se référer à un mentor. Nous appelons ces deux manières de collecter des données des *modes d'échantillonnage*. Le premier mode d'échantillonnage correspond à des algorithmes développés dans la littérature pour automatiquement pousser l'agent vers des parties intéressantes de l'environnement ou vers des types de compétences utiles. De tels algorithmes sont appelés des algorithmes de *curiosité artificielle* ou *motivation intrinsèque*. Le deuxième mode d'échantillonnage correspond au *guidage social* ou *l'imitation*, où un partenaire humain indique où explorer et où ne pas explorer. D'une étude des liens entre ces deux méthodes concurrentes, nous avons finalement construit une architecture algorithmique où les deux modes s'entremêlent en une structure hiérarchique, appelée **Socially Guided Intrinsic Motivation** (SGIM).

Nous avons conçu une méthode avancée pour combiner apprentissage par guidage social et motivation intrinsèque, pour l'apprentissage tout au long de la vie de multiples compétences. Cette combinaison a été construite dans un contexte plus général d'apprentissage stratégique, où l'agent choisit comment apprendre le mieux parmi différents modes d'apprentissage. Notre approche consiste à permettre à l'agent de décider en ligne des aspects de sa interaction avec son environnement physique et social: quoi et comment apprendre; quoi, quand, comment et qui imiter. Nous présenterons plusieurs implémentations de SGIM qui utilisent plusieurs représentations et algorithmes pour ses différentes sous-parties.

En effet, nous avons construit une architecture algorithmique intrinsèquement motivée pour apprendre comment produire par ses actions des effets et conséquences variées. Par exemple, le robot apprend à jeter une balle à diverses distances, en associant une distance (conséquence) avec un mouvement spécifique (action). Il apprend de manière active et en ligne en collectant des données qu'il choisit en utilisant plusieurs modes d'échantillonnage. Au niveau du meta apprentissage, il apprend de manière active quelle stratégie d'échantillonnage est plus efficace pour améliorer sa compétence et généraliser à partir de son expérience à un grand éventail de conséquences. Par apprentissage par interaction, il acquiert de multiples compétences de manière structurée, en

découvrant par lui-même les séquences développementale. En étudiant SGIM, nous contribuons à différents domaines de l'apprentissage automatique:

- **apprentissage par imitation:** Nous explorons les questions *quoi, comment, quand et qui imiter*. Nous proposons une structure unifiée pour aborder ces questions fondamentales de l'apprentissage par imitation. Par apprentissage stratégique, le choix en ligne des options permet un plus grand progrès en compétence. En particulier, pour l'**apprentissage interactif**, nous analysons et identifions les avantages à combiner exploration autonome et guidage social. Nous construisons un agent qui décide par lui-même quand interagir avec les enseignants.
- **apprentissage multi-tache et babillage d'objectifs:** SGIM peut découvrir la structure de son environnement par exploration orientée par objectifs. Nous proposons une architecture unifiée pour aborder à la fois l'imitation de conséquence et l'exploration autonome de conséquences.
- **apprentissage actif:** nous étudions différents niveaux d'apprentissage actif: l'agent décide quelle action exécuter, quel objectif se fixer, et quel mode utiliser. Ses décisions sont en ligne, poussées par la curiosité artificielle, en mesurant empiriquement son progrès en apprentissage.
- **apprentissage hiérarchique:** nous proposons une architecture hiérarchique pour apprendre sur plusieurs niveaux: les politiques, les conséquences et les modes d'échantillonnage. L'agent décide activement et de manière hiérarchique quoi et comment apprendre, en se basant sur ses mesures de son progrès en apprentissage.

La thèse est structurée de la manière suivante. Nous formalisons d'abord dans le chapitre 1 le problème dans le contexte de la robotique cognitive développementale. C'est un domaine de recherche qui a à la fois pour but des systèmes d'apprentissage robotique efficaces en s'inspirant des sciences cognitives et de la psychologie développementale, et la modélisation et la validation de théories de sciences cognitives et de psychologie développementale à l'aide de robots. Dans le chapitre 2, nous illustrons le problème auquel nous voulons répondre par une expérience où le robot iCub apprend à reconnaître des objets en 3D par manipulation. Ensuite, nous présentons successivement trois architectures algorithmiques, chacune permettant à l'agent de prendre plus de décisions actives concernant sa stratégie d'apprentissage. Notre but dans leur conception est un apprentissage multi-tâches rapide et précis. Leur conception est fondé sur des études développementale sur la motivation intrinsèque et l'apprentissage par imitation.

Tout d'abord, nous avons construit une architecture algorithmique qui apprend *quoi et comment apprendre*, appelée Socially Guided Intrinsic Motivation by Demonstration (SGIM-D, Motivation Intrinsèque Guidée Socialement par Démonstrations). Elle explore de manière active les espaces des politiques et des effets, en utilisant à la fois apprentissage par imitation et exploration autonome orientée par objectif. La conception de SGIM-D permet d'analyser la complémentarité entre ces deux modes d'échantillonnage. Puis, nous étudions la question de *quand imiter* au travers

Résumé

| Apprentissage actif | Expériences | Résultats |
|---------------------|--|--|
| <p>SGIM-D</p> | <p>cf. chapitre 3</p> | <ul style="list-style-type: none"> - apprend avec une plus grande précision et plus rapidement - utilise les démonstrations pour biaiser son exploration dans les espaces des politiques et des conséquences - utilise l'exploration autonome pour palier aux problèmes de correspondance - utilise l'exploration autonome pour compenser l'insuffisance des démonstrations |
| <p>SGIM-IM</p> | <p>cf. section 4.1</p> | <ul style="list-style-type: none"> - apprentissage interactif - auto-ajuste le rythme de ses demandes d'aide (en fonction du coût d'une démonstration) - testé sur des environnements déterministes et stochastiques. |
| <p>SGIM-ACTS</p> | <p>cf. section 4.3</p> <p>cf. chapitre 5</p> <p>cf. chapitre 1</p> | <ul style="list-style-type: none"> - apprentissage interactif avec plusieurs enseignants - apprentissage de plusieurs types de tâches - testé sur des espaces continus et discrets - testé en simulation et sur des robots - modélisation du développement des enfants |

Figure 0.0.1: Trois architectures algorithmiques sont présentées avec diverses expériences.

de la décision supplémentaire de quel mode d'échantillonnage choisir de manière active. Notre nouveau système, appelé SGIM with Interactive learning at the Meta level (SGIM-IM, SGIM avec meta-apprentissage Interactif), permet à l'agent d'explorer de manière interactive, active et hiérarchique, les espaces des politiques, des effets et des modes d'échantillonnages. Finalement, nous améliorons SGIM-D et SGIM-IM pour obtenir un système qui apprend *comment, quand, quoi et qui imiter*. SGIM with Active Choice of Teacher and Strategy (SGIM-ACTS, SGIM avec Choix Actif d'Enseignant et de Stratégie) est la version complète de l'apprentissage actif. L'agent décide de tous les aspects de sa stratégie d'apprentissage. SGIM-ACTS étudie le cas où plusieurs enseignants sont disponibles et que plusieurs modes d'échantillonnage peuvent être choisis. SGIM-IM et SGIM-ACTS sont décrits dans le chapitre 4. Finalement, alors que dans les chapitres précédents nous avons pour but la construction d'une intelligence artificielle pour apprendre de manière autonome le long de la vie, le chapitre 5 montre que l'architecture conçue peut servir pour modéliser et mieux comprendre le développement des enfants. Nous utilisons les algorithmes développés pour étudier un phénomène décrit par la psychologie infantile : le développement de la vocalisation par les bébés. Nous illustrons comment un agent incarné et doté de l'algorithme SGIM-ACTS peut apprendre à vocaliser et comment une séquence développementale peut naître. Cette étude montre aussi que l'architecture algorithmique SGIM-ACTS peut être implémentée avec une autre représentation d'un modèle et un autre algorithme d'optimisation. En conclusion, les limites et développements possibles de nos contributions sont discutés en chapitre 6.

Mots clés : apprentissage actif, apprentissage interactif, apprentissage par imitation, exploration orientée par objectifs, collecte de données, exploration, apprentissage par démonstration

A Curious Robot Learner for Interactive Goal-Babbling

ABSTRACT

The challenges posed by robots operating in human environments on a daily basis and in the long-term point out the importance of adaptivity to changes which can be unforeseen at design time. Therefore, the robot must learn continuously in an open-ended, non-stationary and high dimensional space. It can not possibly explore all its environment to learn about everything within a life-time. To be useful and acquire skills, the robot must on the contrary be able to know which parts to sample and what kind of skills are interesting to learn. One way is to decide what to explore by oneself. Another way is to refer to a mentor. We name these two ways of collecting data *sampling modes*. The first sampling mode correspond to algorithms developed in the literature in order to autonomously drive the robot in interesting parts of the environment or useful kinds of skills. Such algorithms are called *artificial curiosity* or *intrinsic motivation* algorithms. The second sampling mode correspond to *social guidance* or *imitation* where the teacher indicates where to explore as well as where not to explore. Starting from the study of the relationships between these two concurrent methods, we ended up building an algorithmic architecture where relationships between the two modes intertwine into a hierarchical learning structure, called **Socially Guided Intrinsic Motivation** (SGIM).

We developed an advanced technique to combine learning by social guidance and intrinsic motivation, for life-long learning of multiple skills. This combination has been designed in a more general context of strategic learning, where the learning agent chooses how to learn best between different learning modes. Our approach enabled the learner to decide online about its interaction with its physical and social environment: what and how to learn; what, when, how and whom to imitate. We will present several implementations of SGIM using different representations and algorithms for its different functions.

Indeed, we have built an intrinsically motivated active learner which learns how its actions can produce varied consequences or outcomes. For instance, the robot learns to throw a ball at different distances, by associating a distance (outcome) to a specific movement (action). It actively learns online by sampling data which it chooses by using several sampling modes. On the meta-level, it actively learns which data collection strategy is most efficient for improving its competence and generalising from its experience to a wide variety of outcomes. The interactive learner thus learns multiple tasks in a structured manner, discovering by itself developmental sequences. By studying SGIM, we contribute to different fields of machine learning:

- **imitation learning:** we explore the questions of *what, how, when and who to imitate*. We propose a unified structure to address simultaneously these fundamental questions of imitation learning. The strategic learner chooses online the options that enable most competence progress. In particular, in

interactive learning : we analyse and identify advantages of combining autonomous and socially guided exploration, and build an agent which decides by itself when to interact with teachers.

- **multi-task learning and goal-directed learning**: SGIM can discover the structure of its environment by a goal-oriented exploration. We propose a unified architecture to approach goal-oriented imitation learning and goal-directed autonomous exploration.
- **active learning**: we investigate different levels of active learning: the learner decides which action to take, which goal to aim, and which mode to perform. Its decisions are made online, driven by artificial curiosity based on its monitoring of the learning progress.
- **hierarchical learning**: we propose a hierarchical learning architecture to learn on several levels: policy, outcome, and sampling mode. The learner relies on hierarchical active decisions of what and how to learn driven by empirical evaluation of learning progress.

This thesis is structured as follows. We first formalise in Chapter 1 a computational framework in the context of cognitive developmental robotics. This is a field which both builds efficient robot learners using inspiration from cognitive science and developmental psychology, and models and tests cognitive science and developmental psychology theories using robotic learners. In Chapter 2, we illustrate the question we address in our computational framework with an experiment where the iCub robot learns to recognise 3D objects by manipulation. Then we present successively three algorithmic architectures, each one allowing the agent to take more active control of its learning strategy. We have built these algorithms for quick and accurate multi-task learning, grounding their designs on developmental studies of intrinsic motivation and imitation learning. Firstly, we design an algorithmic architecture which learns *what and how to learn*, called Socially Guided Intrinsic Motivation by Demonstration (SGIM-D). It actively explores policy and outcome spaces using both imitation learning and goal-oriented autonomous exploration. The design of SGIM-D allows an analysis of the complementarity between these two sampling modes. Secondly, we explore the question of *when to imitate* by additionally actively learning which of the two sampling modes to use. Our new system, called SGIM with Interactive learning at the Meta level (SGIM-IM) allows the interactive learner to actively and hierarchically explore the policy, outcome, and also sampling mode space. Finally, we extend SGIM-IM into a system which learns *how, when, what and who to imitate*. SGIM with Active Choice of Teacher and Strategy (SGIM-ACTS) is a complete active learner, deciding on all aspects of its strategic learning. It extends SGIM-D and SGIM-IM. SGIM-ACTS considers the case when more teachers are available and more modes can be chosen. SGIM-IM and SGIM-ACTS are described in Chapter 4. Finally, while in the previous chapters we aimed at building artificial systems for autonomous life-long learning, chapter 5 shows that the developed architecture can be useful to model and understand better infant development. We use the developed algorithms to study an observation in child psychology: the development of vocalisation in babies. We illustrate how an embodied agent using SGIM-ACTS can learn to vocalise and the emergence of a developmental sequence. This work also shows that the algorithmic architecture SGIM-ACTS can be instantiated with another representation of a model and another optimisation algorithm. In conclusion, the limits and possible extensions of these contributions are discussed in Chapter 6.

Keywords : **active learning, interactive learning, imitation learning, goal-oriented exploration, data-collection, exploration, programming by demonstration**

Contents

| | | |
|----------|--|-----------|
| 1 | A STRATEGIC LEARNER FOR LIFE-LONG LEARNING | 1 |
| 1.1 | Principles of Life-Long Learning | 2 |
| 1.1.1 | The Challenges of Life-Long Learning | 2 |
| 1.1.2 | Cognitive Developmental Robotics | 4 |
| 1.2 | Methods for Life-Long Learning | 6 |
| 1.2.1 | Active Learning | 7 |
| 1.2.2 | Intrinsic Motivation | 8 |
| 1.2.3 | Teleological Learning | 10 |
| 1.2.4 | Social Guidance | 10 |
| 1.2.4.1 | What? | 11 |
| 1.2.4.2 | How? | 13 |
| 1.2.4.3 | When? | 14 |
| 1.2.4.4 | Who? | 15 |
| 1.3 | Limitations of Each Method | 16 |
| 1.3.1 | Limitations of Social Guidance | 16 |
| 1.3.2 | Limitations of Intrinsic Motivation | 16 |
| 1.3.3 | Combining Social Guidance and Intrinsic Motivation | 17 |
| 1.4 | Strategic Learning | 19 |
| 1.4.1 | Definition of the Strategic Learning | 19 |
| 1.4.2 | Formalisation | 20 |
| 1.4.2.1 | Examples of Learning Problems | 20 |
| 1.4.2.2 | Formalisation of the Learning Problem | 21 |
| 1.4.2.3 | Formalisation of the Strategic Learner | 22 |
| 1.4.2.4 | Examples of Strategic Learners | 23 |
| 1.4.2.5 | Algorithmic Architecture for a Strategic Learner | 23 |
| 2 | LEARNING TO RECOGNISE 3D OBJECTS BY CURIOSITY-DRIVEN MANIPULATION | 27 |
| 2.1 | Problem Description | 28 |
| 2.1.1 | Experimental Protocol | 28 |
| 2.1.2 | Mathematical Formalisation | 30 |
| 2.2 | Methods | 30 |

| | | |
|----------|---|-----------|
| 2.2.1 | Scene Perception and Learning Algorithm | 30 |
| 2.2.2 | Action | 33 |
| 2.2.3 | Decision Making | 33 |
| 2.2.4 | SGIM-ACTS Architecture | 34 |
| 2.3 | Experimental Results | 35 |
| 2.3.1 | Experimental Platform | 35 |
| 2.3.2 | Evaluation of the Learning Process | 35 |
| 2.3.3 | Results | 35 |
| 2.4 | Conclusion | 39 |
| 3 | LEARNING IN COMPLEX CONTINUOUS ENVIRONMENTS | 40 |
| 3.1 | A Passive Learner Benefitting from the Combination of Imitation Learning and Autonomous Exploration | 41 |
| 3.1.1 | Chosen Sampling Modes for a Passive Learner | 42 |
| 3.1.1.1 | Social Guidance | 42 |
| 3.1.1.2 | Autonomous Exploration with Intrinsic Motivation | 43 |
| 3.1.2 | Problem Statement and Assumptions for Motor Learning | 43 |
| 3.1.3 | SGIM-D Overview | 45 |
| 3.1.4 | SGIM-D Architecture | 47 |
| 3.1.4.1 | Lower Level : Policy Space Exploration | 47 |
| 3.1.4.2 | Higher Level : Active Goal Babbling for Outcome Space Exploration | 48 |
| 3.2 | The Fishing Rod Experiment | 50 |
| 3.2.1 | Motor Primitives and Correspondence Mapping | 50 |
| 3.2.2 | Mimic a Policy | 52 |
| 3.2.3 | Performance Measure | 53 |
| 3.2.4 | Goal-Directed Policy Optimisation | 53 |
| 3.2.4.1 | Global Exploration Regime | 54 |
| 3.2.4.2 | Local Optimisation Regime | 54 |
| 3.2.5 | Stochastic Environment | 56 |
| 3.3 | Experimental Protocol | 57 |
| 3.3.1 | Comparison of Learning Algorithms | 59 |
| 3.3.2 | Evaluation | 59 |
| 3.3.3 | Demonstrations | 59 |
| 3.4 | Experimental Results | 61 |
| 3.4.1 | Better Precision | 61 |
| 3.4.2 | A Wide Range of Outcomes | 63 |
| 3.4.3 | Dependence on the Size of the Outcome Space | 64 |
| 3.4.4 | Identification of the Interesting Subspaces | 64 |
| 3.5 | Analysis of the Bootstrapping Effect | 66 |
| 3.5.1 | Outcome Space Exploration | 66 |
| 3.5.1.1 | Dependence of the Performance on the Teacher | 66 |
| 3.5.1.2 | Difference in the Explored Outcome Spaces | 66 |

| | | |
|----------|--|------------|
| 3.5.2 | Policy Space Exploration | 68 |
| 3.5.2.1 | Dependence of SGIM-D Performance on the Quality of Demonstrations | 68 |
| 3.5.2.2 | Analysis of the Demonstrated Movements | 70 |
| 3.6 | Preliminary Results on a Physical Robot | 70 |
| 3.7 | Benefits of the Combination: Conclusion | 73 |
| 4 | INTERACTIVE STRATEGIC LEARNER | 74 |
| 4.1 | What is Interactive Learning? | 75 |
| 4.2 | Interactive Learning at the Meta Level : SGIM-IM | 76 |
| 4.2.1 | Algorithm Description | 76 |
| 4.2.1.1 | SGIM-IM Overview | 76 |
| 4.2.1.2 | Select Behaviour | 78 |
| 4.2.2 | Air Hockey Experiment | 79 |
| 4.2.2.1 | Air Hockey Experimental Setup | 79 |
| 4.2.2.2 | Experimental Protocol | 80 |
| 4.2.2.3 | Results | 81 |
| 4.2.2.4 | Active Choice of Behaviour | 81 |
| 4.2.3 | Fishing Experiment | 83 |
| 4.2.3.1 | Experimental Setup | 83 |
| 4.2.3.2 | Results | 83 |
| 4.2.4 | Discussion and Conclusion | 86 |
| 4.3 | SGIM-ACTS | 87 |
| 4.3.1 | Actively Learning When, Who and What to Imitate | 87 |
| 4.3.1.1 | Choice for Social Guidance | 87 |
| 4.3.1.2 | Interactive Learning Based on Intrinsic Motivation | 88 |
| 4.3.2 | Algorithm Description | 89 |
| 4.3.2.1 | Architecture Outline | 89 |
| 4.3.2.2 | Hierarchical Structure | 91 |
| 4.3.2.3 | Policy Space Exploration | 92 |
| 4.3.2.4 | Sampling Mode and Outcome Space Exploration | 92 |
| 4.3.3 | Throwing and Placing a Ball | 95 |
| 4.3.3.1 | Experimental Setup | 95 |
| 4.3.3.2 | Several Teachers and Sampling Modes | 97 |
| 4.3.3.3 | Comparison of Learning Algorithms | 97 |
| 4.3.3.4 | Results | 98 |
| 4.3.3.5 | Conclusion and Discussion | 103 |
| 5 | ILLUSTRATION ON A DEVELOPMENTAL SEQUENCE FOR VOCALISATION | 104 |
| 5.1 | Development of Vocalisation with Intrinsic Motivation and Social Interaction | 105 |
| 5.1.1 | Vocalisation, Intrinsic Motivation and Social Interaction | 106 |
| 5.1.2 | Development of Vocalisation | 109 |
| 5.1.3 | A Computational Model of the Development of Vocalisation | 109 |

| | | |
|----------|--|------------|
| 5.2 | Model | 111 |
| 5.2.1 | Sensorimotor System | 111 |
| 5.2.1.1 | Vocal Tract and Auditory System | 111 |
| 5.2.1.2 | Dynamical Properties | 114 |
| 5.2.1.3 | Vocalisation Classification | 116 |
| 5.2.2 | Internal Sensorimotor Model | 117 |
| 5.2.3 | Intrinsically Motivated Active Exploration | 118 |
| 5.2.4 | Imitation System | 122 |
| 5.3 | Results | 123 |
| 5.3.1 | Emergence of developmental sequences in autonomous vocal exploration | 124 |
| 5.3.2 | Influence of the Auditory Environment | 127 |
| 5.4 | Conclusion | 131 |
| 5.5 | Discussion | 134 |
| 6 | CONCLUSION | 135 |
| 6.1 | Keynote | 135 |
| 6.1.1 | Synopsis | 135 |
| 6.1.2 | Summary | 136 |
| 6.1.3 | Result | 138 |
| 6.1.4 | Take-away Message | 138 |
| 6.2 | Our approach, its Limitations and Extensions | 138 |
| 6.2.1 | Main Contribution | 138 |
| 6.2.2 | Originality of our Approach | 139 |
| 6.2.3 | Complementary Studies | 140 |
| 6.2.4 | Limitations and Extensions | 140 |
| 6.2.5 | Impact | 142 |
| 6.3 | Journal Papers | 155 |
| 6.4 | Conference Papers with Proceedings | 155 |
| 6.5 | Other International Public Presentations | 156 |

List of Algorithms

| | |
|--|-----|
| 1.4.1 SGIM architecture | 24 |
| 2.2.1 SGIM-ACTS for Discrete Outcomes Spaces | 36 |
| 2.2.2 $[\mathcal{R}] = \text{Goal Interest Mapping}(\mathcal{R}, \mathcal{H}, b, \gamma)$ | 36 |
| 2.2.3 $[\chi, b_g] = \text{Select Label And Sampling Mode}(\mathcal{R})$ | 36 |
| 3.1.1 SGIM-D | 46 |
| 3.1.2 $[\mathcal{R}] = \text{UpdateRegions}(\mathcal{R}, (c, a), \gamma)$] | 49 |
| 3.2.1 $[\mathcal{D}] = \text{Mimic Policy}(b_d, c)$ | 52 |
| 3.2.2 $[(a, b, c)] = \text{Execute}(c, b)$ Set context c and perform policy parameters b | 53 |
| 3.2.4 $[\mathcal{D}] = \text{Goal-Directed Policy Optimization}(c, a_g, p(b a), \mathcal{H})$. Search for policies to reach a_g in context c while building model | 54 |
| 3.2.3 LocalOptimization algorithm using Nelder-Mead simplex algorithm | 55 |
| 3.2.5 $[K_{best}] = \text{LocalData}(a_g, \mathcal{H})$. Retrieve from the memory \mathcal{H} experiences in the locality of a_g | 56 |
| 4.2.1 SGIM-IM | 78 |
| 4.2.2 $[\chi] = \text{SelectSamplingMode}(\Delta_S, \Delta_A)$ | 79 |
| 4.3.1 SGIM-ACTS | 90 |
| 4.3.2 $[\mathcal{R}] = \text{Update Outcome and Sampling Mode Interest Mapping}(\mathcal{R}, \mathcal{H}, a_g, progress_g, \chi)$ | 94 |
| 5.2.1 Self-exploration with active goal babbling (stochastic SAGG-RIAC architecture). | 121 |
| 5.2.2 Strategic active exploration (active goal babbling and imitation with stochastic SGIM-ACTS architecture). | 123 |

Listing of figures

| | | |
|-------|--|----|
| 0.0.1 | Trois architectures algorithmiques sont présentées avec diverses expériences. | v |
| 1.2.1 | Learning forward model for motor control consists in predicting the outcomes of the execution of a given policies in a given context. Learning inverse model consists in choosing a good policy to produce a given outcome from a given context. Models can be stochastic : repeating π_3 in context \mathcal{C}_2 can lead to different outcomes \mathcal{A}_3 and \mathcal{A}_4 . Models can be redundant : both policies π_1 and π_2 produce outcome \mathcal{A}_2 from context \mathcal{C}_2 . The environment can be inhomogeneous with reachable and unreachable parts in the outcome space. | 6 |
| 1.3.1 | Combining social guidance and intrinsic motivation to overcome the limitations of each sampling mode taken separately. | 18 |
| 1.4.1 | The strategic learner samples data by actively choosing various aspects of its exploration, such as its sampling mode, the outcome to focus on, the policy to try, the teacher to learn from. | 20 |
| 1.4.2 | The strategic learner samples data by choosing points (\mathcal{D}_e) with a sampling mode, and then selecting among the points ($\mathcal{D}_{e,As,Bs}$) that belong to a goal subspace $As \times Bs$ | 21 |
| 1.4.3 | Three algorithmic architectures are presented with various illustrative experiments. The details of each algorithm and experiment can be read in the corresponding section or chapter. Each algorithmic architecture allows the agent to take active control of various aspects of its learning strategy, and to test hypotheses about active learning. | 25 |
| 2.1.1 | The strategic learner samples data by actively choosing two aspects of its exploration: its sampling mode and the object to focus on. | 28 |
| 2.1.2 | The objects used during the experiments: some coloured cubes, a yellow car, a grey dog, a violet/blue ball, a red bear. Left and right images respectively show the front/rear sides of the objects. | 29 |
| 2.2.1 | The visual information is processed through a hierarchy of layers, which elaborate the camera images to extract the entities in the scene. The proprioceptive information from the robot is used for categorising the entities. | 32 |
| 2.2.2 | Time flow chart of SGIM-ACTS, which combines Intrinsic Motivation and Social Guidance exploration modes into 2 layers: the sampling mode and object space B exploration, and the image space A exploration. | 34 |
| 2.3.1 | A portion of the database of objects views used for evaluating the recognition performance: precisely, the images related to the cubes. | 37 |

2.3.2 SGIM-ACTS vs Random: recognition performance, i.e. the number of images of the evaluation database correctly recognised by the two exploration modes with two different responses of the teacher (see text). 37

2.3.3 f-measure on the evaluation database, with respect to time. The bottom part of the plot shows the manipulated object at each timestep. 38

3.0.1 The strategic learner samples data by choosing a goal outcome and policy to try. 41

3.1.1 Representation of the problem. The environment can evolve from context state \mathcal{C} to an outcome state \mathcal{A} by means of the learner’s actions with policy π or the teacher’s ζ . The learner and the teacher have a priori different policy spaces. The learner estimates $p(b|a, c)$. By imitation, the learner can take advantage of the demonstrations (c, ζ, a_d) of the teacher to improve its estimation $p(b|a, c)$ 44

3.1.2 Time flow chart of SGIM-D into 3 layers that pertain to the human-machine interface (mode level), the outcome space exploration and the policy space exploration respectively. The architecture combines sub-modules for intrinsically motivated learning and socially guided learning in both the policy and outcomes spaces. 46

3.2.1 Experimental setup with a robot arm holding a fishing can with a flexible wire (simulated by 30 free revolute joints). The robot can produce a movement of its 6 DOF arm by setting the real number values b of its 25 dimensional motor primitive. Then, it can observe the effect/outcome of such a movement, by observing where the float has arrived in the goal/outcome space, i.e. on the surface of the water which is a 2D space. Using SGIM-D, which combines intrinsically motivated active learning and human demonstration, the robot has to learn the complex inverse model mapping all goals/outcomes to the adequate 25 dimensional parameters of motor movement. 51

3.2.2 Mapping of the demonstrations given by the human teacher by the robot. Horizontal axis: time, vertical axis: joint angle (best seen in colors). Are plotted for 2 different joint trajectories of a human demonstrator, the demonstrated trajectory, and the corresponding movement parameters and trajectory mapped by the robot. For a demonstrated trajectory u_{Hd} , parameters u_1, u_2, u_3, u_4 minimise eq. 3.7. Then the parameters u_1, u_2, u_3, u_4 generate the trajectory executed by the robot $u(t)$ according to **Equation 4.1**. For joint trajectory 1, the mapping has a high error value, while for joint trajectory 2, the mapping has a low error value. 52

3.2.3 Outcomes for 3 different policy parameters over 20 repetitions of the same movement, represented in the 2-D space A . Standard deviations are for each policy parameters, respectively (0.005, 0.033) for b_1 , (0.0716, 0.041) for b_2 , and (0.016, 0 .016) for b_3 (best seen in colors). 57

3.3.1 (best seen in colors) (a): The experiment compares the performance of several exploration algorithms: Random exploration of the policy space B , autonomous exploration SAGG-RIAC, Learning from Observation, Imitation learning and SGIM-D. The comparison is made through the same experimental duration (5000 policies performed by the robot), through the same teaching frequency (every 30 policies) and through regular evaluation (every 1000 policies). (b): Map in the 2D outcome space A of the benchmark points used to assess the performance of the robot: by measuring how close they can reach each of these points. (c): Maps in the 2D outcome space A of the teaching sets used in SGIM-D, by three demonstrators. Demonstrator 1 is a SAGG-RIAC learner, while demonstrator 2 is an optimised SAGG-RIAC learner, and demonstrator 3 is a human teacher. 58

3.3.2 Demonstration by kinesthetics. A human demonstrator manipulates a physical robot which is connected to a physical simulator. 60

3.4.1 (best seen in colors) Evaluation of the performance of the robot under the learning algorithms: random exploration, SAGG-RIAC, imitation and SGIM-D (for the human demonstrator 3. We plotted the mean distance to the benchmark points over several runs of the experiment with its variance errorbar. 62

3.4.2 Histograms of the positions explored by the fishing rod inside the 2D outcome space (a^1, a^2) . Each column represents a different learning algorithm: random input parameters, SAGG-RIAC and SGIM-D. We plotted the histogram for one example run of the experiment of each algorithm. In the case of SGIM-D (3rd column), we also graphed the demonstrated outcomes with black crosses. 63

3.4.3 Evaluation of the performance of the robot in the case of a large outcome space ($A = [-100, 100]^2$ is 10^4 times larger than the reachable space, but we only plotted here the distribution on the subspace $[-1, 1]^2$), under the learning algorithms: random exploration, SAGG-RIAC and SGIM-D. 65

3.4.4 Distribution of all the goals set by the higher level during learning in a large space. Each column shows the distribution of an experimental run of the SAGG-RIAC algorithm (col 1) or SGIM-D (col 2). 65

3.5.1 SGIM-D's performance depends on the demonstrator 67

3.5.2 Histogram of the outcomes explored by the fishing rod inside the 2D outcome space. Each algorithm is illustrated by 2 example experiments. 68

3.5.3 Evaluation of the performance of the robot learning with 3 different demonstrators, under the learning algorithms: SGIM-D, Observation and Imitation. 69

3.5.4 Plot for the demonstrations of the trajectories for joint 1 (vertical axis: joint angles, horizontal axis: time). 70

3.6.1 A 6 DOF robot arm used in our fishing experiment. 71

3.6.2 The robot observes the final position of the ball after its movement, and learns which movement can reach different positions on the floor. The camera is placed above the wide surface, and can only see the white surface which materialises the outcome space A 71

3.6.3 Outcomes reached by the red ball if it stabilises on the floor. The axis are the two dimensions of the floor A . The region visible by the camera is normalised so that $A = [0, 1]^2$ 72

3.6.4 Mean error on an experiment for each algorithm, with respect to time. 72

4.2.1 The strategic learner samples data by actively choosing on the three levels on its exploration space: its sampling mode, the object to focus on and the policy to try. 76

4.2.2 Time flow chart of SGIM-IM, which combines Intrinsic Motivation and Social Learning into 3 layers that pertain to the human-machine interface, the outcome space exploration and the action space exploration respectively. 77

4.2.3 Air Hockey Table: the task space is defined as the top border of the square. The puck moves in straight line without friction until it hits either the mallet, the table borders or the obstacle placed on the right side. 80

4.2.4 Comparison of several learning algorithms. Each box represents the chronology of the adopted modes (the figures correspond to the number of actions experimented in the episode). The figures here are given for the Fishing experiment). 81

4.2.5 Evaluation of the performance of the robot with respect to the number of actions performed, under different learning algorithms. We plotted the mean distance to the benchmark set with its standard deviation errorbar. 82

4.2.6 1/ Behaviours chosen through time by SGIM-IM: percentage of times each mode is chosen with respect to the number of actions performed (summed over 100 bins and averaged over several runs of SGIM-IM) 2/ The average progress made by socially guided and intrinsically motivated modes Δ_S and Δ_A 83

4.2.7 Fishing experimental setup. 84

4.2.8 Evaluation of the performance of the robot with respect to the number of actions performed, under the learning algorithms: random exploration, SAGG-RIAC, SGIM-IM, SGIM-D with a demonstration every $M = 30$ movements, and SGIM-D with a demonstration every $M = 80$ movements (to equal the total number of demonstrations of SGIM-IM). We plotted the mean distance with its standard deviation errorbar. 85

4.2.9 Analysis of the fishing experiment. 86

4.3.1 The strategic learner samples data by actively choosing whether to explore by autonomous exploration or social guidance. If it explores by autonomous exploration, it decides a goal outcome and a policy to try. If it explores with social guidance, it chooses whether to imitate the demonstrated policy or the demonstrated outcome. 87

4.3.2 Emulation and Mimicry for motor learning. In emulation, the learner tries to reproduce the outcome demonstrated by the teacher without trying to reproduce the teacher’s policy, but uses its own movement. In emulation, the learner reproduces the demonstrated policy or movement, without trying to reproduce the demonstrated outcome. 88

4.3.3 Time flow chart of SGIM-ACTS, which combines Intrinsic Motivation and Mimicking and Emulation into 3 layers that pertain to the sampling mode, the outcome space and the policy space exploration respectively. 91

4.3.4 The selection of outcome and sampling mode is based on a partition of the outcome space with respect to different competence progress levels. We illustrate with the case of an outcome space of 3 different types of outcomes. $A = A1 \cup A2 \cup A3$ where $A1 \subset \mathbb{R}^2$, $T2 \subset \mathbb{R}$ and $T3 \subset \mathbb{R}^3$. A is partitioned in regions R_i to which are associated measures of competences γ for each mode. The "Select Goal Outcome and Sampling Mode" function chooses the (region, mode) pair that makes the most competence progress. 94

4.3.5 An arm, described by its angle ϕ , is controlled by a motor primitive with 14 continuous parameters (taking bounded values) that determine the evolution of its acceleration $\ddot{\phi}$. A ball is held by the arm and then released at the end of the motion. The objective of the robot is to learn the mapping between the parameters of the motor primitive and two types of outcomes he can produce: a ball thrown at distance x and height h , or a ball placed at the arm tip at angle ϕ with velocity smaller than $|v_{max}|$ 95

4.3.6 Comparison of several learning algorithms 98

4.3.7 Mean error for the different learning algorithms averaged over the two sub outcome spaces (final variance value Δ is indicated in the legend) 99

4.3.8 Mean error for the different learning algorithms for each of the throwing outcomes and placing outcomes separately. The legend is the same as in **Figure 4.3.7**. 100

4.3.9 Sampling Mode chosen by SGIM-ACTS through time: percentage of times each mode is chosen for several runs of the experiment. 101

4.3.10 Types of outcome chosen by SGIM-ACTS through time: percentage of times each kind of outcome is chosen for several runs of the experiment. 101

4.3.11 Consistency in the choice of outcome, teacher and mode: percentage of times each sampling mode, teacher and outcome are chosen over all the history of the robot. 102

5.1.1 The strategic learner samples data by choosing on the 3 levels of its exploration space: which sampling mode to use, the sound to produce, and the motor command to use. 105

5.1.2 The first year of infant vocal development. 109

5.2.1 Speech production general principles. The vocal fold vibration by the lung air flow provides a source signal: a complex sound wave with fundamental frequency F_0 . According to the vocal tract shape, acting as a resonator, the harmonics of the source fundamental frequency are selectively amplified or faded. The local maxima of the resulting spectrum are called the formants, ordered from the lower to the higher frequencies. They belong to the major features of speech perception. 112

5.2.2 Articulatory dimensions controlling vocal tract shape (10 dimensions, from left to right and top to bottom), adapted from the documentation of the DIVA source code. Each subplot shows a sagittal contour of the vocal tract, where we can identify the nose and the lips on the right side. Bold contours correspond to a positive value of the articulatory parameter, the two thin contours are for a null (neutral position) and negative values. These dimensions globally correspond to the dimensions of movements of the human vocal tract articulators. For example, Art_1 mainly controls the jaw height, whereas Art_3 rather controls the tongue front-back position. 113

- 5.2.3 An illustrative vocalization example. A) Articularory trajectories of 5 articulators during the 800ms of the vocalization (4 articulators, from *art4* to *art7* are not plotted for the sake of readability but display the same trajectory as *art2*). Circles at 250 and 800ms represents the values of the first and second commands, respectively, for each trajectory. The first commands are active from 0 to 250ms and second ones from 250 to 800ms, as represented by dotted black boxes. The trajectories are computed by the second order dynamical equation (5.1), starting in a neutral position (all articulators set to 0). B) Resulting vocal tract shapes at the end of each command, i.e. at 250 and 800ms. Each subplot displays a sagittal view with the nose and the lips on the left side. The tongue is therefore to the right of the lower lip. C) Sound wave resulting from the vocalization. D) Trajectories of the 3 auditory parameters, the intensity I and the two first formants $F1$ and $F2$. Dotted black boxes represent the two perception time windows. The agent perceives the mean value of the auditory parameters in each time window, represented by the circles at 250 and 650ms. 115
- 5.2.4 Illustration of incremental learning and inference in the sensorimotor model in a toy 2-dimensional sensorimotor space. The figure has three columns, corresponding to the state of a learning agent after 500, 1000 and 1500 sensorimotor experiments ($t = 500, 1000, 1500$). Each column is divided in three panels A, B and C, as indicated in the middle column (boxed letters in gray panels). X-axis (M space) and y-axis (S space) of A are shared by B and C, respectively. A) The unknown function $s = f(m)$ is represented by the blue curve. The red points are the sensorimotor experiments made at this stage (i.e. until the corresponding time index t): when m is produced, $s = f(m) + \epsilon$ is perceived, where ϵ is here a Gaussian noise with a standard deviation of 0.5. The ellipses represent the state of G_{SM} learned from the sensorimotor experiments, which is here a GMM with 6 components (each ellipse represents a 2D Gaussian). B) The three vertically-aligned plots show the motor distributions $G_{SM}(M | s_g)$ for 3 different goals, $s_1 = -9.0$ (top), $s_2 = 0.0$ (middle), and $s_3 = 8.0$ (bottom), in each of three columns (i.e. at the three time indexes). They are inferred from G_{SM} in A using Bayesian inference. C) The probability distributions on S (rotated 90 degrees anti-clockwise) resulting from sampling motor configurations according to $G_{SM}(M | s_g)$, to reach the three goals s_1, s_2 , and s_3 , the shade of grey of each one corresponding to that used in B: this means for example that, at a given time index t , producing motor commands according to the distribution $G_{SM}(M | s_3)$ (panel B, bottom) will result in sensory consequences following the darker distribution in panel C. The three considered goals s_1, s_2 and s_3 are represented by the three horizontal red lines, which are the same in the three columns. The distributions in C thus reflect how the learner is able to reach one of the three considered goals using the current state of its sensorimotor model: we observe that at $t = 500$, it can only reach $s_2 = 0$; at $t = 1000$, it can also reach $s_1 = -9$ and at $t = 1500$ it can reach those three goals. 119

5.2.5 Illustration of interest distribution computation. Top-left: the recent history of competences of the agent, corresponding to blue points in the space $T \times S \times C$, where T is the space of recent time indexes (in \mathbb{R}^+), S the space of recently chosen goals s_g (mono-dimensional in this toy example) and C the space of recent competences of reaching those goals (in \mathbb{R}^+). For the sake of the illustration, the competence variations over time are here hand-defined (surf surface) and proportional to the values in S (increases for positive values, decreases for negative values). We train a GMM of 6 components, G_{IM} , to learn the joint distribution over $T \times S \times C$, represented by the six 3D ellipses. Projections of these ellipses are shown in 2D spaces $S \times C$ and $T \times C$ in the top-right and bottom-left plots. To reflect the competence progress in this dataset, we then bias the weight of each Gaussian to favor those which display a higher competence progress, that we measure as the covariance between time and competence for each Gaussian (in the example the purple ellipse shows the higher covariance in the bottom-left plot). We weight the Gaussians with a negative covariance between time T and competence C (blue, black and red ellipses) with a negligible factor, such that they do not contribute to the mixture. Using Bayesian inference in this biased GMM, we finally compute the distribution over the goal space S , $G_{IM}(S)$, thus favoring regions of S displaying the highest competence progress (bottom-right). 121

5.3.1 Self-organization of vocal developmental stages. At each time step t (x-axis), the percentage of each vocalization class between t and $t+30.000$ is plotted (y-axis), in a cumulative manner (sum to 100%). Vocalization classes are defined in section 5.2.1.3. Roman numerals shows three distinct developmental stages. I: mainly no phonation or unarticulated vocalizations. II: mainly unarticulated. III: mainly articulated. The boundaries between these stages are not preprogrammed and are here manually set by the authors, looking at sharp transitions between relatively homogeneous phases. 125

5.3.2 Evolution of the distribution of auditory goals, motor commands and sounds actually produced over the life time of a vocal agent (the same agent as in **Figure 5.3.1**). The variables are in three groups (horizontal red lines): the goals chosen by the agent in line 3 of **Algorithm 5.2.1** (top group), the motor commands it inferred to reach the goals using its inverse model in line 4 (middle group), and the actual perceptions resulting from the motor commands through the synthesizer in line 5 (bottom group). There are two columns (1st and 2nd), because of the sequential nature of vocalizations (two motor commands per vocalization). Each subplot shows the density of the values taken by each parameter (y-axis) over the life time of the agent (x-axis, in number of vocalizations since the start). It is computed using an histogram on the data (with 100 bins per axis), on which we apply a 3-bins wide Gaussian filter. The darker the color, the denser the data: e.g. the auditory parameter I actually reached by the second command ($I(2)$, last row in ‘Reached’, 2nd column), especially takes values around 0 (y-axis) until approximately 150.000th vocalization (x-axis), then it takes rather values around 1. The three developmental stages of **Figure 5.3.1** are reported at the top. 126

5.3.3 The two vocalizations of the adult Teacher 1 used in **Figure 5.3.4**, with the same convention as in **Figure 5.2.3**. 128

5.3.4 Vocalizations of the learning agent in the early and mature stages of vocal development. A) All auditory outcomes s produced by the agent in its early stage of vocalization are represented by blue dots in the 6-dimensional space of the auditory outcomes. The adult sounds are represented in red circles. The actually produced auditory outcomes only cover a small area of physically possible auditory outcomes, and correspond mostly to $I(2) = 0$, which represent vowel-consonant or consonant-consonant types of syllables. B) The auditory outcomes produced by the infant in its mature stage of vocalization cover a much larger area of auditory outcomes and extend in particular over areas in which vocalizations of the social peer are located. 128

5.3.5 Progress made by each strategy with respect to the number of updates of the sensorimotor model G_{SM} . These values have been smoothed over a window of 100 updates. For $t < 450$, the agent makes no progress using emulation strategy. After $t = 450$, both strategies enable the agent to make progress. 130

5.3.6 Percentage of times each strategy is chosen with respect to the number of updates of the sensorimotor model G_{SM} . These values have been smoothed over a window of 100 updates. For $t < 450$, the agent mainly uses self-exploration strategy. When its knowledge enables it to make progress in emulation, it chooses emulation strategy until it can emulate the ambient sounds well (and its competence progress decreases). 131

5.3.7 Vocalizations of the learning agent in the early and mature stage of vocalization in two different speech environments (Teacher 2 and Teacher 3). A and C) All auditory outcomes produced by the vocal learner in its early stage of vocal development are represented by blue dots in the 6-dimensional space of the auditory outcomes. The sounds of the environment are represented in red circles. The auditory outcomes only cover a small area, and do not depend on the speech environment. B and D) The auditory outcomes produced by the infant in its mature stage of vocal development cover a larger area of auditory outcome, which depend on the speech environment. 132

6.1.1 Three algorithmic architectures are presented with various illustrative experiments. The details of each algorithm and experiment can be read in the corresponding section or chapter. Each algorithmic architecture allows the agent to take active control of various aspects of its learning strategy. Each of the results presented allow us to present aspects of the advantages for a fully active system, that can decide on all aspects of its learning strategy. 137

Lexicon

Summary of the notations used throughout this manuscript. They are introduced in 1.4.2 (page 20), and their use may vary depending on the example at hand.

| | |
|------------------------|---|
| a, b | stochastic variables |
| $p(b a)$ | conditional probability distribution of b given a |
| \mathcal{L} | learning algorithms |
| $\chi \in \mathcal{X}$ | sampling modes |
| $\psi \in \Psi$ | focus functions |
| J | cost function |
| I_e | mean cost at episode e , function to minimise |
| \mathcal{D}_e | data set collected at episode e |
| \mathcal{H} | history of the learning agent (episodic memory) |

Algorithm acronyms frequently used

| | |
|-----------|--|
| SGIM | Socially Guided Intrinsic Motivation |
| SGIM-D | Socially Guided Intrinsic Motivation by Demonstration |
| SGIM-IM | Socially Guided Intrinsic Motivation with Interactive learning at the Meta-level |
| SGIM-ACTS | Socially Guided Intrinsic Motivation with Active Choice of Teacher and Strategy |

Acknowledgments

There are many people to thank for contributing to this dissertation and to my years in the phd program.

Foremost is Oudeyer Pierre-Yves, who has been my principal guide throughout the past three years. Pierre-Yves has been an amazing mentor from whom I have learned a lot. Just to cite a few points that stand out: the way he chooses and tackles research problems, how he interacts with members of the team, and especially how he communicates about research through scientific papers, presentations and mass media.

I am very grateful to all with collaborators with whom I had the chance to team up. Moulin-Frier Clément patiently shared his insight in infant development in vocalisation and his outlook on intrinsic motivation. Baranes Adrien mentored me during my first months with his endless beaming enthusiasm. Most of my work would not have been possible without the help of Fudal Paul, Fogg Haylee, Thomas Huet and especially Béchu Jérôme. They offered unfailing support while conducting experiments with me, especially for the fishing robot. I would also like to thank Ivaldi Serena, Droniou Alain, Lyobova Natalia, Gérardaux-Viret Damien, Filliat David, Padois Vincent and Sigaud Olivier for the two months I spent with the iCub. It had been a dream for me to work with the iCub, and these few weeks we spent together were very intense and most certainly memorable. I am very grateful for their team spirit and supporting my crazy work schedule.

A special thanks to Sanchez Marie, Robin Natalie and Jahier Nicolas for their diligence in helping me hop around the world. I have also earnestly appreciated the support Inria gave us. Inria is definitely a very comfortable workplace, especially with their very fancy new building. I was lucky to be part of the Flowers team, of whom I remember most the scientific discussions, the passion of all for our research topics. I would like to single out Cederborg Thomas who has always provided me with good ideas in the orientations of my work. Whenever I needed new inspiration, I could discuss with him and rely on his advice. I would like to thank Degris Thomas for all the very passionate conversations about reinforcement learning and artificial intelligence. His earnest enthusiasm and his readiness to discuss for long hours over a tea are representative of the team's spirit. I would like to mention Rouanet Pierre, the first person of the team whom I have met, and has always shared his experience to advise me. I would like to thank all those who make this team a stimulating and relaxed environment: Laviolle Jeremy, Lopes Manuel, Mangin Olivier, Grizou Jonathan, Lapeyre Matthieu, Ly Olivier, Louis ten Bosch, Bénureau Fabien, Forestier Sébastien, Sprauel Jonathan, Danieau Fabien, Cavallé Damian, Devaux Timothée, Monge Alexandre and Solé-Blanco Blaise.

During these three years, I was able to publish a few papers owing to many readers and their comments. To cite a few names: Stulp Freek, Lopes Manuel, Degris Thomas, Bénureau Fabien, Fogg Haylee...

Finally, the writing of this dissertation benefitted from its numerous readers and their helpful feedback. I wish to thank the reviewers, Demeris Yiannis and Gaussier Philippe, and the members of the jury, Wrede Britta, Stasse Olivier and Sigaud Olivier, for their perusal and detailed comments. Degris Thomas, Cederborg Thomas, Moulin-Frier Clément, Baranes Adrien and Hinaut Xavier helped me considerably raise the standard of this dissertation.

1

A Strategic Learner for Life-Long Learning

Contents

| | | |
|-------|--|----|
| 1.1 | Principles of Life-Long Learning | 2 |
| 1.1.1 | The Challenges of Life-Long Learning | 2 |
| 1.1.2 | Cognitive Developmental Robotics | 4 |
| 1.2 | Methods for Life-Long Learning | 6 |
| 1.2.1 | Active Learning | 7 |
| 1.2.2 | Intrinsic Motivation | 8 |
| 1.2.3 | Teleological Learning | 10 |
| 1.2.4 | Social Guidance | 10 |
| | 1.2.4.1 What? | 11 |
| | 1.2.4.2 How? | 13 |
| | 1.2.4.3 When? | 14 |
| | 1.2.4.4 Who? | 15 |
| 1.3 | Limitations of Each Method | 16 |
| 1.3.1 | Limitations of Social Guidance | 16 |
| 1.3.2 | Limitations of Intrinsic Motivation | 16 |
| 1.3.3 | Combining Social Guidance and Intrinsic Motivation | 17 |
| 1.4 | Strategic Learning | 19 |
| 1.4.1 | Definition of the Strategic Learning | 19 |
| 1.4.2 | Formalisation | 20 |
| | 1.4.2.1 Examples of Learning Problems | 20 |
| | 1.4.2.2 Formalisation of the Learning Problem | 21 |
| | 1.4.2.3 Formalisation of the Strategic Learner | 22 |
| | 1.4.2.4 Examples of Strategic Learners | 23 |
| | 1.4.2.5 Algorithmic Architecture for a Strategic Learner | 23 |

Robots are expected to deal with a wide variety of tasks like manipulating objects or interacting with humans in a changing environment. With this outlook, not all relevant information is known at design time. Without being reprogrammed by the designer, robots should be able by interacting with the physical and social environment, to learn cumulatively novel skills that were not initially programmed, in a way that is analogous to human development. This challenge raises the issue, among others, of exploration. **Self-experimentation and learning by social interaction are essential to explore the environment.**

In section 1, we discuss the principles and mechanisms for machines to learn throughout their lives. The challenges of life-long learning put into relief the importance of data collection strategy for learning. In section 2, we present two types of data collection modes, and motivate our study of combining these two modes in section 3. Finally, in section 4, we present the framework of strategic learning which is our aim.

1.1 PRINCIPLES OF LIFE-LONG LEARNING

The promise of personal robots operating in human environments to interact with people on a daily basis points out the importance of adaptivity. The robot can no longer simply be all-programmed in advance by engineers, and reproduce actions predesigned in factories. It needs to adapt to its changing and open-ended environment, match its behaviour and learn new skills as the environment and users' needs evolve, in a way which can seldom be predictable at design time. Therefore **adaptation and learning need to take place all along its life time**. This is generally referred to as *life-long learning*. The learner should be able to accomplish necessary skills, unspecified at design, in different environment states, without requiring a specialised engineer to reprogram or retune all the learning parameters by hand. Life-long learning involves multi-task learning but also active choice of tasks to be learned. From our studies of the capacity of biological agents for life-long learning, we can identify several challenges which artificial agents have to face. **Mechanisms based on theories of natural intelligence have thus given rise to the new field of *cognitive developmental robotics***, within which scope we fall.

1.1.1 THE CHALLENGES OF LIFE-LONG LEARNING

Efforts in building personal robots imply first building an efficient artificial intelligence which can learn and adapt to its environment.

Artificial intelligence is the branch of computer science which studies and designs intelligent agents. The field was founded on the hypothesis that a central property of natural intelligence can be described precisely enough that it can be simulated by a machine. Thus the human mind is seen as the reference and the model for intelligent systems. To build an intelligent artificial system we could thus get inspiration from humans.

Actually, humans are not born with all their skills. They **learn throughout childhood to adulthood and their brains change constantly**. This is referred to as *neuroplasticity*. Neuronal studies have classified these changes into three categories: 1) changes in the functional organisation of the brain when the strength of existing synaptic connections varies, called *functional plasticity*; 2) changes in synaptic connections called *structural plasticity*; 3) birth of brain cells, called *neurogenesis*. For instance, [Globus and Scheibel \(1967\)](#) showed that depriving rabbits of visual stimuli entails changes in spine morphology in the visual cortex. More generally, learning and memory consolidation is continuously occurring ([Dudai, 2012](#)). Unfortunately, sometimes maladaptive plasticity occurs and results in phenomena in which a person continues to feel pain or sensation within a part of their body which has been amputated. This phenomena has been named the phantom limb and has been extensively described in ([Ramachandran, 1999](#)). This capacity of the brain to adapt has led to works in treatment of brain damage such as sensory substitution. If one sense is damaged, another sense can sometimes take over. For instance, [Bach-y Rita \(1967\)](#) studied how to substitute a retina with skin and touch receptors.

From these observations in our behaviour and in our brain, it is reasonable to analyse and propose mechanisms to emulate such intelligence in machines and robots. These mechanisms have to face the challenges of :

- **stochasticity**: the same action repeated by an agent several times can cause different outcomes. These outcomes can also occur with varying delay. The time at which the agent receives or perceives the resulting state can be stochastic. The mapping between the action policies and the outcomes is

generally not a simple function, a one to one mapping. It is generally better described by a probability density between policies and outcomes.

- **high-dimensionality:** New borns find themselves in a "blooming and buzzing confusion" as described by (James, 1890). They have to figure out the meaning of all the sounds, touches, smells, tastes, colours, images... and learn how to use their some 650 muscles, in order to control their sensorimotor space. Policies that the learning agent can perform and outcomes that can be resulted may lie in very high-dimensional spaces. The volumes of these spaces increase as their dimensionalities grow. The learner faces what has been named the curse of dimensionality (Bishop, 2007). Sampling and learning decreases in efficiency as the dimensionality of the environment increases. For example, random motor exploration is bound to fail for building forward or inverse models through regression in high-dimension, as showed Baranes and Oudeyer (2013). Thus, discovering structure in their sensorimotor space and learning new skills in such an environment can seem daunting.
- **unlearnability:** There are very large regions of the sensorimotor spaces for which predictive or control models cannot be learnt at a given moment in time or even at any moment in time. Some other regions of the sensorimotor space are unlearnable at a given moment of time/development, but may become learnable later on. For instance, learning to play tennis is impossible for a child who does not even know how to grasp a spoon. It only becomes possible later when the child has acquired necessary skills that he can reuse to play tennis. Some of these regions of the sensorimotor space are definitively unlearnable. For example, trying to control the movement of the sun is impossible for babies as well as adults. An individual is not told at birth the adequate causal groupings of variables he may observe nor what he can control. Rather, he discovers by himself which are the sensible correlations and causalities.
- **unboundedness:** Even if the learning agent were told what is learnable and what is not at a moment in time, the set of learnable associations between motor commands and sensory feedback is still infinite and can not be all tested within a lifetime. Let us take the example of a baby trying to explore his environment, both 1/ mapping motor commands to sensory feedback, which we call *knowledge* or *predictive model*, and 2/ mapping sensory feedback to motor commands which we call *skills* or *inverse models*. Were the baby be given a ball, there is a very large amount of both knowledge and skills to be learnt: learning to throw the ball in various boxes in the room, at various distances, with a various number of bounces, using various parts of the body (hands, shoulders, head, legs, ...). Now imagine what the same child may learn with all the other toys and objects in the room, then with all the objects in the house and on Earth. Even with no increase of complexity, the child could basically always find something to learn. Actually, this would even apply if there would be no objects or no house around the child: the set of skills he could learn to do with his sole own body, conceptualised as an "object/tool" to be discovered and learnt, is already unbounded. He thus discovers the sensorimotor space of his own body, which is called his *proprioception*. Learning in everyday human environment thus poses the problem of open-ended learning: the learner must decide what skill it should improve and which new skill he should explore. It balances between specialisation and generalisation, between good mastery of skills and a mastery of a wide variety of skills. In open-ended learning, exploring all localities in a lifetime with a constant density is impossible. The identification of interesting subspaces becomes crucial.

1.1.2 COGNITIVE DEVELOPMENTAL ROBOTICS

To address the problem of robotic learning in natural environments, approaches inspired by natural intelligent systems have developed. (Lungarella et al., 2003; Asada et al., 2009; Pfeifer and Scheier, 1999; Oudeyer, 2011b) have rephrased the problem as: Can a robot learn like a child? Can it learn a variety of new skills and new knowledge unspecified at design time and in a partially unknown and changing environment? How can it discover its body and its relationships with the physical and social environment? How can its cognitive capacities continuously develop without the intervention of an engineer? What can it learn through natural social interactions with humans? **Taking inspiration from developmental psychology, neurosciences, biology and linguistics**, these approaches use principles of :

- **development** : it indicates the progressive evolution of abilities and skills. Indeed, (human or of other species) babies' abilities are far from the adults'. Intelligence is acquired through a prolonged period of maturation and growth during which a single fertilised egg first turns into an embryo, then grows into a newborn baby, and eventually becomes an adult individual which, typically before growing old and dying, reproduces. Even in adulthood, the human brain changes and adapts to the changes of his body and environment. The processes underlying developmental changes are inherently robust and flexible as demonstrated by the amazing ability of biological organisms to devise adaptive solutions to cope with environmental changes in time and space and guarantee their survival. Because evolution has selected development as the process through which to realise some of the highest known forms of intelligence, it is reasonable to assume that development is mechanistically crucial to emulate such intelligence in machines and other human-made artefacts. More precisely, it is impossible to pre-program all skills to prepare agents to all situations. On the contrary, they need adaptation mechanisms to evolve and develop in their personal environment, through developmental stages as described by Piaget (Piater, 1952).
- **action perception loop**: movements are modulated by perceptual information to insure a functionally organised and adapted response. Conversely, as we move and influence the environment, the perceptual information varies. Perception and action form a continuous loop. According to Gibson (1986): "we must perceive in order to move, but we must also move in order to perceive". The cycle of perceptual changes and motor responses is dynamic and continuous. Movement is essential for perceptual development. Held and Hein (1963) showed that presenting visual stimuli resulting from passive motion and not from its own motion to a cat deprived the cat from its ability to move by itself. Self-produced movement with its corresponding visual feedback is necessary for the development of visually-guided behaviour. This experiment thus outlines the importance of the perception-action loop.
- **enactivism** : introduced by Varela et al. (1991), enactivism hypothesises that cognition is based on situated, embodied agents. It is a theoretical approach to understanding the mind which emphasises the way that organisms and human mind organise themselves by interacting with the environment. Enactivism thus uses the notion of *embodiment* (Brooks, 1991; Pfeifer and Scheier, 1999) and action for cognition, which hypothesises that the mind is largely determined by the form of the organism's body and the actions it can do. The embodied cognition is grounded on self-experience. Enactivism is also related to the notion of *situated cognition*, which argues that knowing

is inseparable from doing as all knowledge is situated in activity bound to social, cultural and physical contexts. The way we conceptualise and reason depends on "the kinds of bodies we have, the kinds of environments we inhabit, and the symbolic systems we inherit, which are themselves grounded in our embodiment" (Johnson, 1987). The mind builds from the personal history of each agent of his sensorimotor perception, and is not mere manipulation or operations of abstractions. Researchers from neuroscience, anthropology, linguistics, philosophy, psychology, computer science, artificial intelligence and robotics converge to the conclusion that brains, environments and bodies are coupled. Cognitive processes emerge from the evolution of a body with specific initial sensorimotor capacities interacting with its environment.

- **trial-and-error:** this idea comes from the enactivism concept. Trial-and-error is characterised by repeated attempts until success or abandonment. Thorndike (1898) observed the behaviour of cats trying to escape from home-made puzzle boxes and established that their behaviour is improved by experience. As the agent learns from its personal experience, it needs to act on the environment to measure the outcome of its actions to improve its knowledge about the environment. Trial and error can be seen as one of the two basic approaches to problem solving or knowledge acquisition, in contrast to an approach using insight and theory.

Robotic approaches using these principles, and more generally inspiration from natural intelligent systems are called *cognitive developmental robotics*, also known as epigenetic robotics. It is a highly interdisciplinary subfield of robotics in which ideas from artificial intelligence, developmental psychology, neuroscience, and dynamical systems theory play a pivotal role in motivating the research. The main goal of developmental robotics is to model the development of increasingly complex cognitive processes and to understand how such processes emerge through physical and social interaction in natural and artificial systems. It thus links artificial to natural systems and conversely. On the one hand, it targets task-independent architectures and learning mechanisms for artificial systems that enable them to learn new tasks unknown to the designer/programmer, allowing the complexity of acquired skills to increase progressively in life-long learning. To build efficient machines, it takes inspiration from the living world. cognitive developmental robotics approaches have recently grown popular because of their promising results to build efficient artificial intelligence systems to evolve in daily environments. On the other hand, cognitive developmental robotics uses machines as tools for understanding the living. Robots are typically employed as testing platforms for theoretical models of the emergence and development of action and cognition. If a model is instantiated in a system embedded in the real world, a lot can be learned about its strengths and potential flaws. Unlike evolutionary robotics which operates on phylogenetic time scales and populations of many individuals, developmental robotics capitalises on "short" ontogenetic time scales and single individuals or small groups of individuals. Developmental robotics develops computational systems to model and experiment theories from developmental sciences which not only are a means to build more versatile and adaptive machines, but are also as means to evaluate the coherence of these algorithms and a means to explore alternative explanations for understanding biological development.

Our work is grounded on the principles of developmental learning, embodiment and trial and error. More precisely, our approach belongs to cognitive developmental robotics, linking natural and artificial systems in both directions. In chapters 2 to 4, we aim at building an efficient data collection strategy for artificial systems learning by using theories of cognitive sciences on natural systems. In chapter 5, while testing that

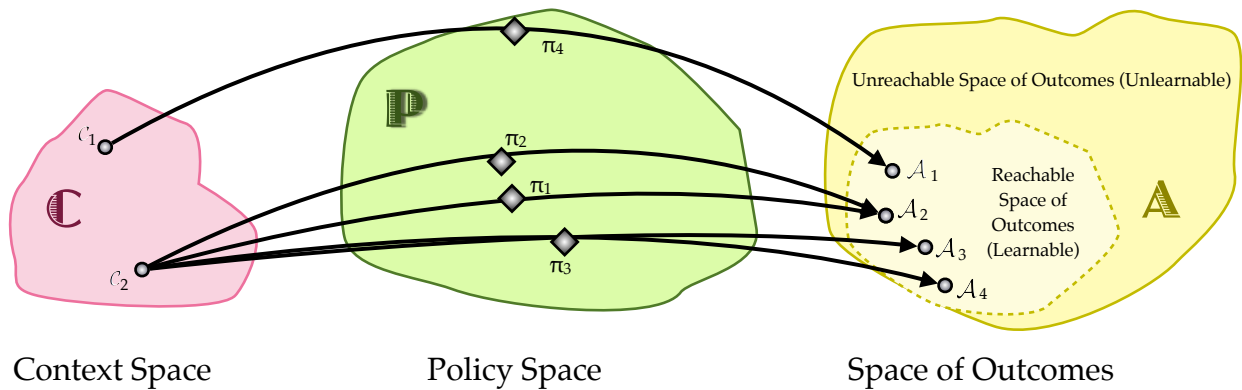


Figure 1.2.1: Learning forward model for motor control consists in predicting the outcomes of the execution of a given policies in a given context. Learning inverse model consists in choosing a good policy to produce a given outcome from a given context. Models can be stochastic : repeating π_3 in context C_2 can lead to different outcomes A_3 and A_4 . Models can be redundant : both policies π_1 and π_2 produce outcome A_2 from context C_2 . The environment can be inhomogeneous with reachable and unreachable parts in the outcome space.

our data collection system is still efficient with another instantiation, we show that our system can model the development of vocalisation observed in child psychology and thus use an algorithmic architecture to study child development.

Life-long learning by robots to acquire multiple skills in unstructured environments poses challenges of not only predicting the consequences or outcomes of their actions on the environment, but also learning which actions cause desired outcomes. The set of possible outcomes can be in large and high-dimensional sensorimotor spaces, while the physical embedding of robots allows only limited time for collecting training data. To address these challenges of life-long learning, we get inspired more specifically by the principles of active learning, intrinsic motivation, teleological learning, and socially guided learning. In the next section, we describe in detail how these inspirations have influenced machine learning, and how it influences our work.

1.2 METHODS FOR LIFE-LONG LEARNING

Robot are ideally able to perform several tasks, and to learn tasks cumulatively. Yet, learning of new action skills is a difficult problem because their sensorimotor spaces are large and high-dimensional, and at the same time their physical embedding allows only limited time for collecting training data. Thus, learning must be associated to mechanisms for guided exploration. **Exploration methods developed in the recent years can be classified into two interacting types of guidance: 1) socially guided exploration (Nehaniv and Dautenhahn, 2007; Billard et al., 2007; Argall et al., 2009); 2) internally guided exploration and in particular intrinsically motivated exploration (Schmidhuber, 1991a; Barto et al., 2004b; Oudeyer et al., 2007).** We call these two methods for data collection *sampling modes*.

In this section we detail the general methods for learning, with a particular focus on motor learning. The

robot needs to know how to act on its environment to produce different effects or outcomes, it has to adapt its actions and movements to the state of the environment and to the task to complete. In the following, we refer to :

- the state of the environment prior to the action as the context \mathcal{C} . We note \mathbb{C} the set of all possible states that can describe the environment.
- the actions of the robot as a policy π . Policies are generally speaking probability distributions of performing certain motions for the robot. In our work, we mainly use a parameterised encoding of movement. We specify a movement by a vector of real numbers which are parameters of a constrained lower-level motor controller, also called motor primitive. Motor primitives consist in this study in innate or acquired neurally embedded motor and muscle synergies used by humans for control (d'Avella et al., 2006; Weiss and Flanders, 2004). We note \mathbb{P} the set of policies available to the robot.
- the effect of the robot's action, or environment change as an outcome \mathcal{A} . An outcome thus describes the state or state change of the environment after an action. We note \mathbb{A} the set of all possible outcomes. We suppose that \mathbb{A} is perceivable by the robot.

Motor learning comprises both 1/ learning to predict the outcomes of one's policies given a context (this is called a *forward model*, see **Figure 1.2.1**) and 2/ learning control policies to produce desired outcomes depending on the context (this is called an *inverse model* or a *control model*). These models can be stochastic: as illustrated in **Figure 1.2.1** the execution of policy π_3 in the same context \mathcal{C}_2 can lead to different outcomes \mathcal{A}_3 and \mathcal{A}_4 . The model can also be redundant: from context \mathcal{C}_2 , different policies π_1 and π_2 lead to the same outcome \mathcal{A}_2 . The environment can have unlearnable parts, and the learning agent has to detect this.

In the following subsections, we introduce the notions of active learning and teleological learning, then describe existing works in intrinsic motivation and social guidance. The description and analysis of these methods have been partially published in (Nguyen and Oudeyer, 2013a).

1.2.1 ACTIVE LEARNING

The challenges of high-dimensionality, unlearnability and unboundedness outline the importance of data collection for learning. From this observation comes the idea that a learner, which not only processes the data that are given to him to update its knowledge but is also allowed to choose the data from which to learn, can achieve greater accuracy with fewer training data. Such a system has been formalised under the name of *active learner*.

Active learning was initially developed in the field of statistical learning for classification and regression learning problems where the cost of querying a given data point for its label or output value is high. Therefore **finding strategies to minimise the number of queries and therefore maximising the usefulness of each experiment** becomes essential. A large diversity of criteria can be used to measure the usefulness of a query, such as the uniformity of the sampling density (Whitehead, 1991), the maximisation of the prediction errors (Thrun, 1995) the maximisation of the model variance (Cohn et al., 1996), the maximisation of the expected improvement (Jones et al., 1998), value of demonstration (Shon et al., 2007) ... It has been proved that an active learner outperforms a passive learner (Shon et al., 2007).

For embedded robots learning various skills, acquiring new data has a cost in time and energy. The learning agent therefore has to decide for instance in which order he should focus on learning how to

achieve the different outcomes, how much time he can spend to learn to achieve an outcome or which data collection modes to use for learning to achieve a given outcome. The field of cognitive developmental robotics has developed Intrinsic motivation algorithms which can be conceptualised as active learning mechanisms. Intrinsic motivation algorithms are a way to guide explorations of the environment in a self-organised manner, as we explain in the next subsection.

1.2.2 INTRINSIC MOTIVATION

Approaches to robot skill learning based on optimisation and reinforcement learning techniques have been widely studied recently. In reinforcement learning, one has assumed that an engineer provides manually a reward function that is associated to a pre-defined specific task (Kober et al., 2010; Peters and Schaal, 2008; Schaal et al., 2003; Stulp and Schaal, 2011). Once the reward function is defined, techniques allowing efficient and fast use of training data have been elaborated, such as natural actor-critic architectures (Peters and Schaal, 2008), path integral approaches (Theodorou et al., 2010) or advanced Black Box optimisation techniques (Stulp and Sigaud, 2012). In optimisation, stochastic methods have been developed mainly to reach a given goal. For instance, evolutionary algorithms have been developed since Holland (1975) has learned a binary classification task with a generic population-based metaheuristic optimization algorithm. More recently Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen and Ostermeier, 2001) have been proven very efficient in reaching maximising a given fitness function, or in other words in reaching a predetermined goal. While these techniques may seem to rely less on the human expert, they still require an engineer to provide a specific reward or fitness function associated to each new particular task to learn.

In order to allow robots to learn more autonomously a wider diversity of tasks, defined here as goals in a parameterised outcome space, methods have been devised for learning forward and inverse models. Once learnt, these forward and inverse models can then be used in conjunction with for example planning methods in order to reach goals. Yet, exploration is a fundamental challenge to achieve the autonomous learning of such forward and inverse models in high-dimensional robots. This is why methods of active exploration and learning have recently been developed in the fields of developmental robotics and robot learning (Lopes and Oudeyer, 2010), reusing some of the concepts elaborated in the statistical active learning framework (Fedorov, 1972; Cohn et al., 1996; Roy and McCallum, 2001). These methods are inspired by *intrinsic motivation* in psychology.

Intrinsic motivation was described in (White, 1959) : “While the purpose is not known to animal or child, an intrinsic need to deal with the environment seems to exist and satisfaction (the feeling of efficacy) is derived from it.” Intrinsic motivations are not homeostatic: the general tendency to explore is not a consummatory response to a stressful perturbation of the organism’s body. It triggers spontaneous exploration and curiosity in humans. For (Ryan and Deci, 2000a), “Intrinsic motivation is defined as the doing of an activity for its inherent satisfaction rather than for some separable consequence. When intrinsically motivated, a person is moved to act for the fun or challenge entailed rather than because of external products, pressures or reward”. Intrinsic motivation is to be contrasted with extrinsic motivation, which is “a construct that pertains whenever an activity is done in order to attain some separable outcome. Extrinsic motivation thus contrasts with intrinsic motivation, which refers to doing an activity simply for the enjoyment of the activity itself, rather than its instrumental value” (Ryan and Deci, 2000a). In (Berlyne, 1965), Berlyne proposes an integration of these motivations: “The probability and direction of specific exploratory responses can

apparently be influenced by many properties of external stimulation, as well as by many intraorganism variables. They can, no doubt, be influenced by stimulus intensity, color, pitch, and association with biological gratification and punishment, ... [but] the paramount determinants of specific exploration are, however, a group of stimulus properties to which we commonly refer by such words as *novelty, change, surprisingness, incongruity, complexity, ambiguity and indistinctiveness*. ”

Intrinsic motivation is clearly visible in young infants, that consistently try to grasp, throw or lick new objects they encounter. Even human adults are still often intrinsically motivated while they play chess or music.

Such motivations are obviously useful, since they are incentives to learn many skills that will potentially be readily available later on for challenges and tasks which are not yet foreseeable. In order **to develop in an open-ended manner so as to learn a wide variety of tasks and to learn more and more complex tasks autonomously, without being told step by step how to learn their elementary components and combine them into complex tasks, robots should certainly be equipped with intrinsic motivation systems, forming the core of an architecture for task-independent learning.** Computational architectures based on intrinsic motivation have been developed since the 1990s, and can be categorised based on the measures that are used by the learning agent to evaluate the intrinsic interestingness of an activity or a situation. Three broad types of measures of interestingness can characterise intrinsic motivation and its measure of interest (Oudeyer and Kaplan, 2007; Baldassarre and Mirolli, 2013a) :

- Knowledge-based models, in which interestingness is related to the difference between the outcome observed and the expectation of the robot. Within this approach, knowledge and expectations are represented in an information theoretic framework and a prediction framework. For instance, (Klyubin et al., 2008) defined a measure for the maximum amount of information that an agent could send from its actuators to its sensors via the environment, called empowerment. Measures such as minimisation of the prediction error, local density of already sampled points, decrease of the global variance, minimisation of the model uncertainty ... have also been used (Barto et al., 2004b; Oudeyer, 2011a). In (Oudeyer et al., 2007; Schmidhuber, 2010, 1991a), parameters of motor policies are chosen for experimentation so that the observed consequences in the outcome space provide maximal improvement of the quality of the learned forward model, which is then inverted for control when needed.
- Competence-based models, in which interestingness is related to the degree of performance/competence of an agent for self-determined tasks. They are directly inspired by theories of affectance (White, 1959) or competence and self-determination (Ryan and Deci, 2000a). For instance, (Rolf et al., 2010) has developed learning algorithms based on measures of competence for tasks pre-determined by the designer. (Baranes and Oudeyer, 2013) went further and developed goal-oriented exploration algorithms where the agent self-determines goals where they make more competence progress.
- Morphological models, where interestingness is related to the structural relationship among multiple sensorimotor channels, to compare information characterising several pieces of stimuli perceived at the same time in several parts of the sensory input. For instance Sporns and Lungarella (2006) studied how various information-theoretic cost functions to be optimised by a sensorimotor system allowed various coordinated behaviour to self-organise.

Such methods inspired by intrinsic motivation recently led to novel robotic and machine active learning methods which outperform traditional active learning methods.

1.2.3 TELEOLOGICAL LEARNING

Knowledge-based intrinsic motivation has developed in the recent years. Yet, these methods were shown to become inefficient when dimension increases in (Baranes and Oudeyer, 2013), and these limitations were addressed by competence-based approaches where instead of performing active motor babbling, parameterised tasks were actively sampled through active goal babbling, then generating lower-level goal directed exploration. These approaches are inspired by psychological studies highlighting *teleological approaches* (Csibra, 2003) which consider **actions as goal-oriented**. Indeed, a series of experiments finds that infants connect actions to both their antecedents (context) and their consequents (outcome) (Csibra, 2003; Csibra and Gergely, 2007). Thus, every learning episode can be described as [context][policy][outcome]. Goal babbling has been shown recently to considerably fasten learning by exploiting the sensorimotor redundancies and the lower dimensionality of outcome spaces (Baranes and Oudeyer, 2013; Rolf et al., 2010; Baranes and Oudeyer, 2010a). For example, with the SAGG-RIAC architecture which we re-use in this thesis, it was shown how robots could learn omnidirectional quadruped walking (thus learning to find the parameters of motor policies to achieve the whole variety of possible displacement tasks) or learn inverse arm kinematics with several dozen dimensions (thus learning the parameters of motor policies to reach all spatial goals possible in the visual outcome space) (Baranes and Oudeyer, 2013).

1.2.4 SOCIAL GUIDANCE

In order to build a robot that can learn and adapt to human environment, the most straightforward way might be the knowledge transfer from a human into a machine. Humans and many animals do not just learn a task by trial and error. Rather they extract knowledge about how to approach a problem from watching other people performing a similar task. Behavioural psychology studies (Whiten, 2000; Tomasello and Carpenter, 2007) highlight **the processes through which the behaviour of an individual β may come to be like α 's, such as mimicry, stimulus enhancement, imitation or emulation**. Imitation is a mechanism that witnesses emerging representational capabilities (Piaget and Cook, 1952).

Learning a policy from demonstrations provided by a teacher is commonly referred to as Programming by Demonstration (PbD) or imitation learning (Nehaniv and Dautenhahn, 2007; Billard et al., 2007; Argall et al., 2009). PbD targets an implicit means of training a machine, such that explicit and tedious programming of a task by a human user can be minimised. It is an intuitive medium of communication for humans, who already use demonstrations to teach other humans. It can in principle offer a natural means of teaching machines that would be accessible to non experts. For instance trajectory and keyframe demonstrations have been shown to be efficient and easy to use for non-experts (Akgun et al., 2012). That is why several works incorporate human input to guide the robot learning process.

Imitation learning has been developed around three main challenges. The first challenge takes the viewpoint of computational motor control, learning from demonstration is a highly complex problem that requires to map a perceived action that is given in an external (world) coordinate frame into a totally different internal frame of reference to activate motoneurons and subsequently muscles. This challenge

involves the remapping of the demonstration in the learner’s reference, the detection of the right reference, the correspondence problems... Such a challenge has proposed trajectory-based PbD where statistical regression techniques are used to model the invariances of demonstrated movements (Billard et al., 2007; Grollman and Jenkins, 2010; Chernova and Veloso, 2009; Lopes et al., 2009c; Cederborg and Oudeyer, 2012; Calinon, 2009a; Calinon et al., 2007; Peters and Schaal, 2008), or inverse reinforcement learning approaches (Abbeel and Ng, 2004; Verma and Rao, 2006; Mangin and Oudeyer, 2012) where one attempts to achieve goal imitation by inferring the hidden cost function maximised by the demonstrated movement (Lopes et al., 2010). The second challenge is about the exploitation of a demonstration to generalise and adapt it to apply to different situations or different tasks. For instance, prior works have given a human trainer control of the reinforcement learning reward (Blumberg et al., 2002; Kaplan et al., 2002), provide advice (Clouse and Utgoff, 1992), or teleoperate the agent during training (Smart and Kaelbling, 2002). A third challenge is to study imitation from a social interaction and a communication point of view. Gaussier et al. (2007) for instance have looked into the emotional communication rather than the explicit content of the message. Nadel et al. (2004) have studied the co-development of imitation and communication, especially how imitation can enhance communication. In this dissertation, we will concentrate on the second challenge.

More precisely in the context of motor learning, we can formalise the guidance of a human teacher to boost the learning of the relationship between the outcomes $\mathcal{A} \in \mathbb{A}$ and the policies $\pi \in \mathbb{P}$ in contexts $\mathcal{C} \in \mathbb{C}$.

As in many approaches and for the sake of clarity, we assume in this section that the correspondence problem is solved, and do not differentiate the state, outcome and policy spaces between the robot and teacher. This correspondence problem will be partially studied in the experiments of next chapters.

Nevertheless, both the human and the robot have acquired different knowledge, which changes throughout their interaction. We can describe this interaction as the way information flows between the human and the robot, intentionally or unintentionally:

- the human teacher’s behaviour or information flow from the human to the robot, si_H .
- the robot learner’s behaviour or information flow from the robot to the teacher, si_R .

In order to define the social interaction that we wish to consider, let us characterise the different possibilities of information flow as reviewed in (Argall et al., 2009; Billard et al., 2007; Schaal et al., 2003; Lopes et al., 2009b) with respect to: what, how, when and who to imitate. This categorisation have been introduced in (Dautenhahn and Nehaniv, 2002; Breazeal and Scassellati, 2002). In this study, we only examine the possibilities of the information flow from the human to the robot si_H . Intentional communication from the robot to the human is a fundamental aspect of social learning (Chernova and Veloso, 2009; Thomaz, 2006), and will be studied in chapter 4. Please also note that the current review and the work presented in this thesis does not aim at friendly human-robot interaction, the social rules for a comfortable and natural interaction or goal understanding and intentionality. Here, we focus more on the modalities for efficiency to convey content by human-robot interaction, which is summarised in **Table 1.2.1**.

1.2.4.1 WHAT?

Let us examine the target of the information given by the teacher, or mathematically speaking, the space on which he operates. This can be either the policy, context or outcome spaces, or combinations of them.

| | | |
|------|--|--|
| What | Policy space Context space Outcome space | |
| How | Demonstration at a low-level Demonstration at a high-level Advice Reward Labelling (ch. 2) | |
| When | Batch learning | Fixed frequency (ch. 3) Beginning of the learning process |
| | Interactive Learning Learning | At the teacher’s initiative At the learner’s initiative (ch. 4, 5) |
| Who | Decide who to imitate (ch. 4) | |

Table 1.2.1: Different types of social interaction. In bold are the types of social interaction examined in this manuscript (with an indication of which chapter to refer to if it is specific to a chapter)

(I) **POLICY SPACE:** Many social learning studies target the policy space \mathbb{P} . For instance, in programming by demonstration (PbD), si_H shows the right policy to perform in order to reach a given goal. As an illustration, when teaching how to play tennis, your coach could show you how to hit a backhand by a demonstration, or by taking your hand and directing your movement. This approach relates to two levels of social learning: *mimicry*, in which the learner copies the policies of others without an appreciation of their purpose, and *imitation*, in which the learner reproduces the policies and the outcomes, as formalised in (Lopes et al., 2009b; Call and Carpenter, 2002; Whiten, 2000). The policies demonstrated can be mimicked faithfully (Cakmak et al., 2009), be saved as corrections for the current situation (Chernova and Veloso, 2009), form an initial dataset on which to build upon more complicated behaviour (Argall et al., 2008, 2011), or indicate a locality to start an optimum search (Peters and Schaal, 2008). The information can be a trajectory or policy (Peters and Schaal, 2008), high-level instructions (Thomaz, 2006) or high-level advice (Argall et al., 2008, 2011). It can pertain to the entire policy, or only a part of it (Argall et al., 2008, 2011; Nicolescu and Mataric, 2003; Thomaz, 2006). The literature often considers that targeting the policy space is the most directive and efficient method. However, it relies on the human teacher’s expertise, which bears limitations such as ambiguity, imprecision, under-optimality or the correspondence problem. Furthermore, the interaction is more effective at correcting visited situations, than exploring undemonstrated areas of \mathbb{C} and \mathbb{A} .

(II) **CONTEXT SPACE:** The teacher can show interesting contexts $\mathcal{C} \in \mathbb{C}$ in which the learner will have to work out. To illustrate, your tennis coach could train you specifically for situations where you are near the baseline while the ball falls near the net. Your coach would create this situation for you to handle, without saying which policy to perform. During infant-parent joint play with toys, parents are able to play a role in the selection of the attended objects in the highly cluttered environment. These processes of visual selection are realised by implicit or explicit “social cues” like pointing or gaze-following (Slater and Lewis, 2006; Tomasello and Carpenter, 2007). Such social learning are classified as *stimulus enhancement* or *observational conditioning* (Whiten, 2000). The teacher can select objects to be attended to (Cakmak

et al., 2009), structure the environment by defining landmark states (Thomaz, 2006), indicate desirability of contexts through reinforcement signals (Thomaz and Breazeal, 2008), or give advice (Argall et al., 2008, 2011). Acting on the context space enables the learner to explore new situations.

(III) OUTCOME SPACE: The third kind of information is about possible outcomes $\mathcal{A} \in \mathbb{A}$, and is related to goal-directed exploration, where the learner focuses on discovering different outcomes instead of different means of completing the same goal. This pertains to the *emulation* level of social learning, where the observer witnesses someone produce a result on an object, but then employs his own policy repertoire to reproduce the result, as formalised in (Lopes et al., 2009b; Call and Carpenter, 2002; Whiten, 2000; Nehaniv and Dautenhahn, 2007). Your tennis coach could ask you to hit with the ball the right corner of the court, wherever you received the ball, whichever shot you use. Goal-directed approaches allow the teacher to reset goals (Argall et al., 2008), to request the execution of goals (Thomaz, 2006) or to label goal states (Thomaz, 2006; Thomaz and Breazeal, 2008). The learner can infer from the demonstrations the goal by positional and force profiles to iron and open doors (Kormushev et al., 2011), or by using inverse reinforcement learning (Lopes et al., 2011). This approach is essential to learn multiple tasks/goals, and all the more interesting as it is inspired by psychological behaviours (Whiten, 2000; Tomasello and Carpenter, 2007; Csibra, 2003). The drawback is that the learning needs a policy repertoire large enough to be used to reach various goals, before it improves.

1.2.4.2 HOW?

Whichever the target, the information can be communicated from the teacher to the learner in several ways:

(I) DEMONSTRATION AT A LOW LEVEL: The teacher acts on the environment of the experiment and performs the action/movement himself or shows the task or context (Cakmak et al., 2009; Chernova and Veloso, 2009; Peters and Schaal, 2008) : the information flow $si_H \in \mathbb{C} \cup \mathbb{P} \cup \mathbb{A}$. This approach is the most natural for non-expert teachers, and requires little training for the teacher. However, demonstrations are generally assumed of high quality, whereas in reality, they can be ambiguous, unsuccessful or suboptimal in certain areas. Methods to eliminate unnecessary or inefficient parts of the teacher’s execution, to address the ambiguities, are required.

(II) DEMONSTRATION AT A HIGH LEVEL: The teacher shows the context/policy/goal at a symbolic level. A language protocol often enables instructions of policies (Nicolescu and Mataric, 2003; Thomaz, 2006; Thomaz and Breazeal, 2008; Argall et al., 2008, 2011), or suggestions of goals (Thomaz, 2006; Thomaz and Breazeal, 2008). In this case, $si_H \in \tilde{\mathbb{C}}$ or $\tilde{\mathbb{P}}$ or $\tilde{\mathbb{A}}$, which bear a direct transformation to \mathbb{C}, \mathbb{P} and \mathbb{A} . A high-level approach seems more natural by the use of a language, but it is dependant on the predefined communication channel and often lacks flexibility for new situations or changing environments. It also forces the teacher to follow this language scheme, and entails a training of the teacher before he can efficiently communicate with the robot.

(III) ADVICE: The teacher shows the desired context/policy/goal indirectly. He does not show the right desired state but indicates how to approach that state (Argall et al., 2008, 2011). si_H is a function of

the context/policy/goal experienced by the robot and the desired value. Advice is an efficient way of providing instructions at a high-level even for continuous environments, while avoiding the limitations of the demonstrator’s performance, as well as the re-creation of difficult or dangerous states. Nevertheless advice is an indirect way of giving instructions, which may be imprecise and limited by the language definition, which again lacks flexibility and requires the teacher adapting to it.

(IV) REWARD: Reward-like signals ($si_H \in \mathbf{R}$) or "good or bad" indications ($si_H \in \{-1; 1\}$) are common in reinforcement-based approaches, which benefit considerably from the formalism of reinforcement learning (Nicolescu and Mataric, 2003; Thomaz, 2006; Thomaz and Breazeal, 2008). They easily couple social learning with techniques of learning from experience. However, defining the reward function is known to be non-trivial. Especially, human teachers tend to give anticipatory and asymmetrically positive rewards (Thomaz, 2006). Taking into account the non-Markovian behaviour of human beings would induce high complexity in the reinforcement learning framework. Furthermore, reinforcement learning research has so far focused on reaching a single goal $\mathcal{A} \in \mathbb{A}$, and not a set of goals.

(V) LABELLING: A few works have labelled previously reached goals to help structure the environment and facilitate communication between the teacher and the learner (Thomaz, 2006; Thomaz and Breazeal, 2008). In this case, si_H takes discrete values that symbolise the different classes.

1.2.4.3 WHEN?

The timing of the interaction varies with respect to its timing within an episode [context][policy][outcome], and with respect to its general activity during the whole learning process.

(I) TIMING WITHIN AN ACTIVITY EPISODE: If we consider that each activity episode involves a reading of the context state of the environment, before performing a policy, and finishes by observing the outcome in the environment, we can classify the various types of timing of the interaction into two types:

- Feedback: A past-directed message informs the learner about its past behaviour. The chronology would be $[\mathcal{C}][\pi][si_H][\mathfrak{T}]$ or $[\mathcal{C}][\pi][\mathcal{C}][si_H]$. These messages can be good/bad assessments on its past behaviour (Thomaz, 2006; Thomaz and Breazeal, 2008; Nicolescu and Mataric, 2003; Lopes et al., 2011), a scalar reward given by the human teacher (Thomaz, 2006), a correction demonstration (Chernova and Veloso, 2009), an advice to modulate the wrong behaviour (Argall et al., 2008, 2011), or a label of previously reached goals (Thomaz, 2006; Thomaz and Breazeal, 2008). According to his partial knowledge of the internal state of learning of the robot, the human adapts his teaching. However, the robot trial policy can be time consuming when it is very far from any good solution.
- Feedforward: A future-directed message informs the learner before deciding its future behaviour. The chronology would be $[\mathcal{C}][si_H][\pi][\mathfrak{T}]$. These messages are commonly instructive demonstrations of good example behaviours (Cakmak et al., 2009; Chernova and Veloso, 2009). Not only have behavioural studies shown that human teachers tend to give future-directed messages (Thomaz, 2006), feedforward messages also seem more instructive with respect to the immediate future behaviour of the robot. However they do not take into account any information flow from the robot to the teacher.

(II) GENERAL TIMING DURING THE WHOLE LEARNING PROCESS: The rhythm of social interaction varies considerably among studies of social learning:

- At a fixed frequency: In classical imitation learning, the learner uses a demonstration to improve its learning at every policy it performs (Argall et al., 2008, 2011; Cakmak et al., 2009). This solution is ill-adapted to the teacher’s availability or the needs of the learner, who requires more support in difficult situations. Though, this continuous interaction allows steady bootstrapping of the learning and adaptation to changing environments.
- Beginning of learning: A limited number of examples is given to initialise the learning, as a basic behaviours repertoire (Argall et al., 2008, 2011), or a sample behaviour to be optimised (Peters and Schaal, 2008; Kormushev et al., 2010). The learner is endowed with some basic competence before self-exploration. Nevertheless, if the interactions are restricted to the beginning, the learner could face difficulties adapting to changes in the environment.
- At the teacher’s initiative: The teacher alone decides when he interacts with the robot (Thomaz, 2006). In most examples, the teacher gives corrections when seeing errors (Koenig et al., 2010; Cakmak et al., 2010), to restrict human interventions to when it is needed. Nevertheless, it still is time consuming as he needs to monitor the robot’s errors to give adequate information to the learner.
- At the learner’s initiative: The learner can request for the teacher’s help in an ambiguous (Chernova and Veloso, 2009; Cakmak et al., 2010) or unknown (Thomaz, 2006) situation, or when he estimates that the demonstration bring in more information than self-exploration, as in Shon et al. (2007). Some studies also make the learner use the teacher’s messages only when the messages apply to his situation. In goal-based imitation or goal-based mimicking, the learner only follows the teacher when he observes that the teacher completes the goal that the learner aims at (Cakmak et al., 2009). This approach is the most beneficial to the learner, for the information arrives as it needs them, and the teacher needs not monitor the process.

These 4 types can be classified into 2 larger groups:

- batch learning, where the data provided to the learner is decided before the learning phase, and is given independently of the learning progress, generally in the beginning of the learning phase.
- interactive learning, where the user interacts with the incrementally learning robot, either at the teacher’s or the learner’s initiative. This case will be studied extensively in chapter 4, when we design learning algorithms where the agent requests demonstrations to teachers.

1.2.4.4 WHO?

While most social guidance studies only consider a single teacher, in natural environments, a household robot in reality interacts with several users. Moreover, being able to request help to different experts is also an efficient way to address the problem of the reliability of the teacher. Imitation learning studies often rely on the quality of the demonstrations, whereas in reality a teacher can be performant for some outcomes but not for others. Demonstrations can be ambiguous, unsuccessful or suboptimal in certain areas. Like students who learn from different teachers who are experts in the different topics of a curriculum, a robot learner should be able to determine its best teacher for the different outcomes it wants to achieve. To our

knowledge, only [Shon et al. \(2007\)](#) has proposed a framework to enable the learning agent to decide who to imitate. They proposed an active imitation learning algorithm to decide when to ask for demonstrations from unhelpful demonstrators. This could theoretically be used to decide from whom to imitate. Nevertheless, their experiments have only used a single teacher.

Two classical families of approaches to life-long learning, intrinsic motivation and imitation learning, have developed with various implementations. Both families show very promising results. Nevertheless, as illustrated in **Figure 1.3.1**, each of them bears important limitations.

1.3 LIMITATIONS OF EACH METHOD

In this section, we describe the limitations of each of the sampling modes mentioned above, then evaluate the advantages of combining them together.

1.3.1 LIMITATIONS OF SOCIAL GUIDANCE

In socially guided systems, learning has been strongly relying on the involvement of the human user. However, the more dependent on the human the system, the more challenging learning from interactions with a human is, due to limitations such as human patience, attention, memory, or the sparsity of teaching datasets, the absence of teaching for some subspaces, ambiguous and suboptimal human input, correspondence problems, etc, as highlighted in ([Nehaniv and Dautenhahn, 2007](#)). This is one of the reasons why in most approaches to robot learning of motor skills, either in trajectory based approaches or inverse reinforcement learning, only a few movements or motor policies were learnt in any single studies. **Increasing the learner’s autonomy from human guidance could address these limitations.**

1.3.2 LIMITATIONS OF INTRINSIC MOTIVATION

Likewise, intrinsically motivated active exploration methods for learning forward and inverse models still have limitations. In particular, they address only partially the challenges of unlearnability and unboundedness, which rises with the use of real high-dimensional bodies with continuous sensorimotor channels and an open-ended environment. Especially, whatever the measure used, computing meaningful measures of interest is only based on the evaluation of performances of predictive models or of skills. It thus requires a sampling density which decreases its efficiency as dimensionality grows. Goal-oriented intrinsic motivation has improved the performance of learning agents in high-dimensional policy spaces, guiding the policy space exploration by the outcome space exploration. Nevertheless it remains inefficient in high-dimensional outcomes space and in unbounded explorable space. Without additional mechanisms, the identification of learnable zones with knowledge or competence progress becomes less and less efficient as dimensionality grows.

Therefore, complementary developmental mechanisms need to constrain the growth of the size and complexity of practically explorable spaces and structure the environment and the learning sequence, by guiding them rapidly toward learnable subspaces and away from unlearnable subspaces. We argue that social guidance, leveraging knowledge and skills of others, can be key for bootstrapping the intrinsically motivated learning of such models. For example, adequate human demonstration of skills, as we will show in

this thesis, can help the learner to identify which part of the outcome space are reachable and learnable, as well as to provide examples of motor trajectories useful to reach particular goals, and which can be further explored by the robot to reach self-determined nearby goals.

1.3.3 COMBINING SOCIAL GUIDANCE AND INTRINSIC MOTIVATION

Thus, while intrinsic motivation and socially guided learning have so far often been studied separately in developmental robotics and robot learning literature, we believe their integration has high potential. Their combination could push the respective limits of each family of exploration mechanisms we stated above.

Social guidance can drive a learner into new intrinsically motivating spaces or activities which it may continue to explore alone and for their own sake, but might have discovered only due to social guidance. For example, random uniform exploration of the space of movements by a tennis player has low probability to making the ball bounce on the opposite court. The player has higher probability of putting the ball out of the court or even of not hitting the ball at all. Yet, a human may demonstrate early on to the robot specific movements that allow to touch the ball with the racket and put it in the opposite court, and then the robot may later on explore variations of these movements through curiosity, allowing the reaching of goals close to the demonstrated bouncing positions.

Conversely, intrinsically motivated learning can build on information provided by human demonstrators/teachers, such as examples of movements or goals to reach, to then spontaneously explore novel movements allowing to reach similar goals in a refined manner or to reach other self-defined goals with the help of these bootstrapping structure provided by humans. In principle, as human demonstrations are only used as a bias for further autonomous exploration, intrinsically motivated learning can even use information from human teachers with limited skills, and improve over these demonstrated skills by learning to achieve a higher diversity of goals with more efficient movements.

Thus, while self-exploration alone tends to result in a broader repertoire of skills (i.e. capacity to reach many goals in an outcome space), and while exploration guided by a human teacher tends to be more specialised and resulting in fewer tasks that are learnt faster, combining both can bring out a system that acquires a diversity of skills with fast bootstrapping thanks to human guidance, and the possibility on the long-term to bias the system towards learning more precisely skills in the preferred areas of the user. Indeed, [Nicolescu and Mataric \(2003\)](#) showed that the combination of experienced demonstration and autonomous learning, and more generally the incorporation of multiple means for robot instruction and learning are much faster and precise methods for learning and refining previously learned tasks.

The combination of autonomous learning and imitation learning of continuous high-dimensional motor skills was previously studied in ([Kober et al., 2010](#); [Peters and Schaal, 2008](#); [Schaal et al., 2003](#); [Stulp and Schaal, 2011](#)), but this was done only in the context of reinforcement learning for one skill, defined as one goal in the outcome space, and did not rely on active intrinsically motivated learning of forward or inverse models. For example, ([Kober et al., 2010](#)) presented algorithms that allow a robot to learn how to throw a ball at a pre-specified location, by finding adequate parameters of a motor primitive using a human demonstration as bootstrapping and then further optimisation through episodic reinforcement learning. Recently, extensions of these approaches have been presented to allow a robot to generalise motor primitives to novel goals that are close to a set of goals previously learnt with these methods, and leveraging regression techniques ([Kober et al., 2012](#); [da Silva et al., 2012](#)). For example, in ([Kober et al., 2012](#)), a robot can generalise to throw a ball

Combining social guidance and intrinsic motivation

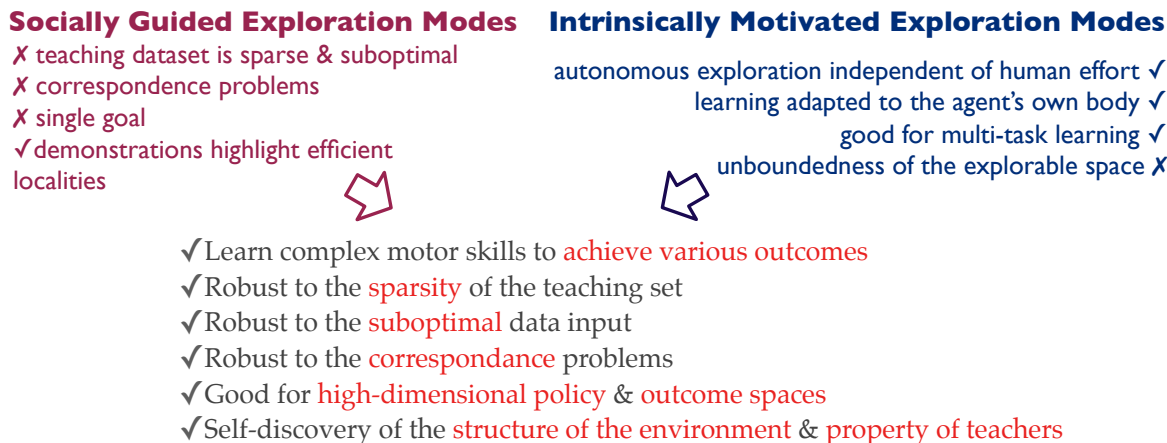


Figure 1.3.1: Combining social guidance and intrinsic motivation to overcome the limitations of each sampling mode taken separately.

close to a few goals it has already learnt. Yet, in (Kober et al., 2010; Peters and Schaal, 2008; Schaal et al., 2003; Stulp and Schaal, 2011), a human engineer has to provide manually a repertoire of goals carefully chosen, and the robot is not able to learn parameters of motor primitives to reach goals that are far away from these pre-specified goals. In (Nicolescu and Mataric, 2003), the robot was learning tasks at a high level by combining low-dimensional parametrized policies. Besides, no method for active learning were used in (Kober et al., 2012; da Silva et al., 2012; Nicolescu and Mataric, 2003).

An active imitation learning algorithm has been developed in (Shon et al., 2007), which combines self-exploration and self-exploration in the reinforcement learning framework. Measures of value of demonstrations based on information theory allows the learning agent to decide whether it is more profitable to explore autonomously or request help from a a-priori non helpful mentor. Unfortunately the algorithm has been designed in discrete, maze-like environments, to learn how to produce a single outcome. The challenge of complex environments and of learning various skills have not been addressed. A combination of social learning with intrinsic motivational drives was proposed and studied by Thomaz et al. (Thomaz and Breazeal, 2008; Thomaz, 2006), with a system called Socially Guided Exploration. In this work, a robot was capable to learn several skills defined as sequences of discrete actions, and as a result of both social dialogue with a human and self-exploration using a hierarchical reinforcement learning algorithm. The focus of this study was on the qualitative dynamics of learning and teaching in the flow of human-robot interaction, and on the design of a full integrated cognitive architecture. While a physical robot was used, the state of the environment as well as robot actions were discrete and few in number. Also, since it was not the focus of these studies, the mechanisms for active learning, for e.g. measuring novelty and mastery, were kept rudimentary and tailored for small discrete state-action spaces.

We would like to address **the learning of skills to achieve various outcomes in the case of an unbounded, non-preset and continuous environment**. As illustrated in **Figure 1.3.1**, by merging socially guided exploration and intrinsic motivation, we aim at a system robust to the sparsity of teaching datasets, to suboptimal data input and correspondence problems, while being good in high-dimensional spaces and automatically discovering the reachable space.

1.4 STRATEGIC LEARNING

We introduced in the previous sections two families of exploration modes which have been developed recently with promising results, but also with their own limitations. Nevertheless, combining them could allow to overcome these limitations. We thus devise a learner which has to learn how to strategically combine these exploration modes into an efficient algorithm. We call this learner a strategic learner, which we define and formalise in this section.

1.4.1 DEFINITION OF THE STRATEGIC LEARNING

We consider the problem of an embedded agent learning to produce a wide range of outcomes with several sampling modes. In the case of our everyday environments, the sensorimotor spaces are large and cannot be all sampled within a lifetime. The learner should discover the properties of the environment and choose adaptively its sampling mode. **To be able to learn for all outcomes and generalise, the agent has to decide both what and how to learn at the same time, and in which order.**

The learning agent has to decide for instance in which order he should focus on learning how to achieve the different outcomes, how much time he can spend to learn to achieve an outcome or which data collection mode to use for learning to achieve a given outcome. These questions can be formalised under the notion of strategic learning (Lopes and Oudeyer, 2012).

One perspective is learning to achieve varied outcomes. It aims at selecting which outcome to spend time on. A typical classification was proposed in (Reichart et al., 2008; Qi et al., 2008) where active learning methods improved the overall quality of the learning. In sequential problems as in robotics, producing an outcome has been modelled as a local predictive forward model (Oudeyer et al., 2007), an option (Barto et al., 2004b), or a region in a parameterised goal/option space (Baranes and Oudeyer, 2013). In these works each sampling and learning of an outcome entails a cost. The learning agent has to decide which outcome to explore/observe next. However most studies using this perspective do not consider several sampling modes.

Another perspective is learning how to learn, by making explicit the choice and dependence of the learning performance on the method. For instance, Baram et al. (2004) selected among different learning behaviours depending on the results for different outcomes. Rebguns et al. (2011) implemented a control based on information gain to classify categories of objects in a room. Shon et al. (2007) proposed an **active imitation learning** algorithm which allows the learner to actively choose whether to self-explore or request a demonstration, based on computations of value of demonstration. However most studies using this perspective consider a single outcome. Shon et al. (2007) focuses only on the search of an action to perform, and not on the object/outcome the action is directed to.

Indeed, these works have not addressed the learning of both how to learn and what to learn, to select at the same time which outcome to spend time on, and which learning method to use. Only (Lopes and Oudeyer, 2012) studies the framework of these questions. They experimented on a toy example with a discrete and finite number of states, outcomes and sampling modes. We would like to investigate in this direction, and devise a system which learns simultaneously both how and what to learn for continuous, high-dimensional and complex environments.

We present a system that allows a robot to learn a diverse repertoire of policies to complete a diversity of outcomes. To be able to generalise from sampled data to the whole space, it has to collect data in an

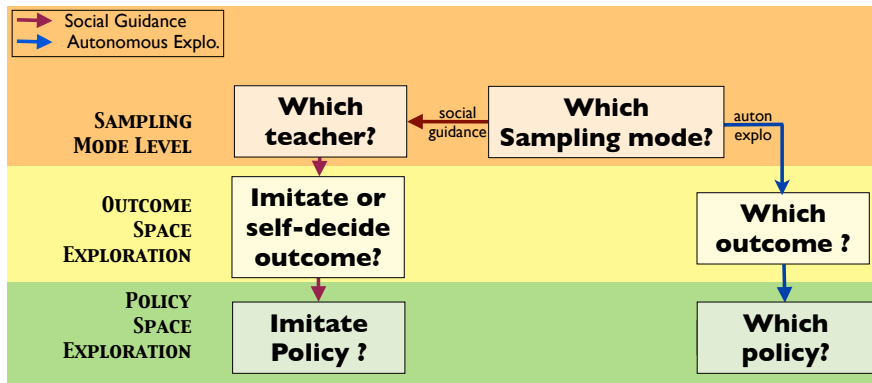


Figure 1.4.1: The strategic learner samples data by actively choosing various aspects of its exploration, such as its sampling mode, the outcome to focus on, the policy to try, the teacher to learn from.

efficient manner. It also has to discover the structure of the environment and the properties of teachers, in order to automatically select the most adapted strategy and the best teachers for a given outcome, and in order to automatically discover the easy, reachable and difficult outcomes.

We will present in this thesis an algorithmic architecture called SGIM (Socially Guided Intrinsic Motivation) for a learning agent to **strategically sample its environment, by actively and empirically choosing various aspects of its exploration, such as its sampling mode, the outcome to focus on, the policy to try, the teacher to learn from.** All these active choices can be summarised in **Figure 1.4.1.**

1.4.2 FORMALISATION

Before proposing algorithms for strategic learning, we formalise the learning problem and use the same formalisation to rewrite the definition of a strategic learner. First let us describe and model the problem to be learned by the agent.

1.4.2.1 EXAMPLES OF LEARNING PROBLEMS

Typically for robotic control, we learn a relationship between a task space A and a joint space B . This relationship can be represented as a probability distribution $\tilde{p}(b|a)$. It represents the inverse kinematic mapping, which can be very redundant for robots with high degree of freedom. For every end-effector position a , the robot needs to know at least one joint position b that enables it to reach a . It thus has to minimise the mean distance J between the goal end-effector positions a and the positions reached by joint configuration b as computed using its current internal model $p_e(b|a)$.

In a tennis game, the task space A is the tennis court where a is the position where the ball bounces, while B is the policy space where b is any possible displacement and racket handling. When you learn to play tennis, you typically train to be able to put the ball everywhere on the tennis court: far or near the nest, right or left... A good tennis player is able to run and hit the ball with policy b so as to make the ball bounce at any position a , and minimise the distance between the goal and the effective positions. Here we note J

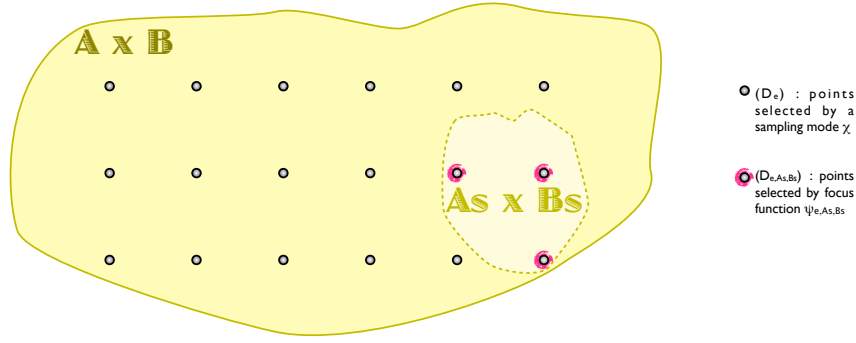


Figure 1.4.2: The strategic learner samples data by choosing points (\mathcal{D}_e) with a sampling mode, and then selecting among the points ($\mathcal{D}_{e, A_s, B_s}$) that belong to a goal subspace $A_s \times B_s$.

the euclidian distance between two bouncing positions. The physics of the ball hitting the racket and its trajectory can be described by a probability distribution $\tilde{p}(a, b)$. It is usually different from the probability distribution $p_e(b|a)$ of the control model in the mind of the learner. The learner represents with $p_e(b|a)$ which policy b is the most likely to make the ball bounce at position a . Training consists in acquiring data to learn $p_e(b|a)$.

To train for tennis, you can spend hours trying to hit the ball using forehand. In this case, you concentrate on a specific subspace of the policy space B . Then you can spend hours trying to make the ball bounce right behind the nest. In this case, you concentrate on a specific subspace of A . This decision to focus on subregions of A or B is hereafter noted ψ . To make progress in tennis, you can also train in casual games autonomously with a friend, or you can also attend lessons with a coach. This second choice constitutes a sampling mode and will be noted χ hereafter. Both choices represent a data collection strategy for learning.

In the next section, we will formalise this description.

1.4.2.2 FORMALISATION OF THE LEARNING PROBLEM

Let us consider two random variables a and b , which take values in possibly continuous, multi-dimensional spaces A and B . The learner has to estimate the conditional probability distribution $\tilde{p}(b|a)$ of the two variables $a \wedge b$, by collecting a set of observations (a, b) in learning episodes. An episode is a step of data collection which can provide a set of observations. Its estimate at episode e is $p_e(b|a)$. We assume the agent knows a supervised learning algorithm \mathcal{L} that improves with a data set $\mathcal{D}_e = \{(a_{e,1}, b_{e,1}), (a_{e,2}, b_{e,2}), (a_{e,3}, b_{e,3}) \dots\}$ its estimate $p_{e+1} = \mathcal{L}(p_e, \mathcal{D}_e)$.

We consider that the learning is efficient if $I_e = \int_A p(a) J(a, \tilde{p}(a | \arg \max_b (p_e(b|a)))) da$ is minimal, where J is a cost function. Formally, J is a function decreasing with respect to p , and defined as:

$$\begin{aligned}
 J : A \times [0, 1] &\rightarrow \mathbb{R} \\
 (a, p) &\mapsto J(a, p)
 \end{aligned} \tag{1.1}$$

1.4.2.3 FORMALISATION OF THE STRATEGIC LEARNER

In this section, we formalise our approach for a strategic learner, which we schematise in **Figure 1.4.2**. For the learning algorithm \mathcal{L} to improve the model estimation p_e , the agent needs to collect data. We suppose the agent has several data collection modes that generate data sets given the current estimation p_e . If we note $C = [0, 1]^{A \times B}$ the space of functions $A \times B \rightarrow [0, 1]$, and $(A \times B)^{\mathbb{N}}$ the space of all sets of pairs $(a, b) \in A \times B$, a sampling mode can be formalised in terms of data collection as a function :

$$\begin{aligned} \chi : C &\rightarrow (A \times B)^{\mathbb{N}} \\ p_e &\mapsto \mathcal{D}_e = \{(a_{e,1}, b_{e,1}), (a_{e,2}, b_{e,2}), (a_{e,3}, b_{e,3}) \dots (a_{e,n}, b_{e,n})\} \end{aligned}$$

To learn better, the system can decide to focus on subregions $As \subset A$ and $Bs \subset B$ to improve its estimation of $p_e(a, b)$ on these subregions at an episode e :

$$\begin{aligned} p_{e,As \times Bs} : A \times B &\rightarrow [0, 1] \\ (a, b) &\mapsto \begin{cases} p_e(a, b) & \text{if } (a, b) \in (As \times Bs) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

We can define a focus function on subregion $As \times Bs$ as a function sampling from a dataset \mathcal{D}_e with a probability distribution $p_{e,As,Bs}$:

$$\begin{aligned} \psi_{e,As \times Bs} : (A \times B)^{\mathbb{N}} &\rightarrow (A \times B)^{\mathbb{N}} \\ \mathcal{D}_e = \{(a_{e,1}, b_{e,1}) \dots (a_{e,n}, b_{e,n})\} &\mapsto \mathcal{D}_{e,As,Bs} = \{ \forall (a, b) \in \mathcal{D}_e, \\ &\text{keep } (a, b) \text{ with proba. } p_{e,As \times Bs}(a, b) \} \end{aligned}$$

Thus data collection modes and focus functions generate data so that \mathcal{L} learns $p_{e+1} = \mathcal{L}(p_e, \psi_{e,As \times Bs}(\chi(p_e)))$. Let us note Ψ and \mathcal{X} the set of all focus functions and modes. A priori $\Psi \subset ((A \times B)^{\mathbb{N}})^2$ and $\mathcal{X} \subset ((A \times B)^{\mathbb{N}})^C$. The goal of our algorithmic architecture is to learn $p_{e+1} = \mathcal{L}(p_e, \chi(\psi(p_e)))$ and minimise $I_{e+1} = \int_A p(a) J(a, \tilde{p}(a | \arg \max_b (p_{e+1}(b|a)))) da$, by learning on the meta-level which ψ and χ enhance best the learning. More precisely, we will consider two ways of determining data collection strategies: for χ , our agent chooses which sampling mode χ to perform from a set of predefined modes; while for ψ , our agent builds incrementally its focus function by building partitions of A and B .

To sum up, we determine a meta-learning algorithmic architecture :

$$\begin{aligned} \mathcal{M} : C^{\mathbb{N}} \times (A \times B)^{\mathbb{N}} \times \Psi^{\mathbb{N}} \times \mathcal{X}^{\mathbb{N}} &\rightarrow \Psi \times \mathcal{X} \\ ((p_e, p_{e-1}, \dots, p_1), & \\ (\mathcal{D}_{e-1}, \mathcal{D}_{e-2}, \dots, \mathcal{D}_1), & \mapsto (\psi_e, \chi_e) = \arg \min_{(\psi_e, \chi_e)} (I_{e+1}) \\ (\psi_{e-1}, \psi_{e-2}, \dots, \psi_1), & \\ (\chi_{e-1}, \chi_{e-2}, \dots, \chi_1)) & \end{aligned}$$

to minimise $I_{e+1} = \int_A p(a) J(a, \tilde{p}(a | \arg \max_b [\mathcal{L}(p_e, \psi_e \circ \chi_e \circ p_e(b|a))])) da$.

1.4.2.4 EXAMPLES OF STRATEGIC LEARNERS

A summary of these notations can be found on page xviii. We illustrate the notations introduced with three learning examples:

- The first one resumes the example of learning tennis previously mentioned.
- The second example considers how a robot can learn to recognise 3D objects with all its different points of view. The robot is allowed to manipulate objects or ask humans to manipulate them to acquire more data. What data collection strategy enable it to correctly label images of different objects under different points of view and at different positions? This example is studied in detail in chapter 2.
- The third example illustrates how a robot can learn how to manipulate a fishing rod, or in other words, how to move its robotic arm to make the float land at any desired position on the surface of the water. This example is studied in detail in chapter 3.

For these examples, the functions are defined as summarised in the table below.

| Problem | Tennis | Object recognition by active manipulation | Fishing |
|-------------------------------------|--|--|---|
| a | position where the ball bounces | image | arm movement |
| b | player displacement and racket handling | object | position of the float on the surface of the water |
| \mathcal{L} Learning algorithm | a supervised learning algorithm | classification algorithm | locally-weighted learning based on the k-nearest neighbours |
| \mathcal{X} Sampling modes | play casually, hire a coach | push the object, drop the object, ask a human to manipulate the object | imitation, intrinsically motivated exploration |
| Ψ Focus functions | train on backhand, train on forehand, put the ball far the nest, put the ball near the nest, put the ball in the right half, put the ball in the left half,... | choose which object to manipulate | a position of the surface of the water to reach with the hook |

1.4.2.5 ALGORITHMIC ARCHITECTURE FOR A STRATEGIC LEARNER

In our approach, a learner can thus decide on these sampling modes with an algorithmic structure described in **Algorithm 1.4.1**. It learns by episodes where it generates a set of observations (line 6). These new data allow the update of its model p with the given learning algorithm \mathcal{L} (line 7). This new estimation will orient its data collection of the next episode by focusing on a region of the environment and choosing a sampling mode (line 9). Three algorithmic architectures following these guidelines will be presented, each

Algorithm 1.4.1 SGIM architecture

```

1: Initialization: initialise the focus and modal functions  $\psi_1 \in \Psi$  and  $\chi_1 \in \mathcal{X}$ 
2: Initialization: initialise the estimated probability distribution  $(a, b) \mapsto p_1(a, b)$ 
3: Initialization: initialise the history  $\mathcal{H} \leftarrow (p_1, \emptyset, \emptyset, \emptyset)$ 
4:  $e \leftarrow 1$ 
5: while true do
6:   generate a set of observations  $(a, b) : \mathcal{D}_e \leftarrow \psi_e(\chi_e(p_e))$ 
7:   Learning:  $p_{e+1} \leftarrow \mathcal{L}(p_e, \mathcal{D}_e)$ 
8:   append  $p_{e+1}, \mathcal{D}_e, \psi_e, \chi_e$  to  $\mathcal{H}$ 
9:   Meta-Learning:  $(\psi_{e+1}, \chi_{e+1}) \leftarrow \mathcal{M}(\mathcal{H})$ 
10:   $e \leftarrow e + 1$ 
11: end while

```

one allowing the agent to take more active control of its learning strategy, as illustrated in **Figure 1.4.3**. Whereas the algorithmic architecture SGIM-ACTS (Socially Guided Intrinsic Motivation with Active choice of Teacher and Strategy) is a fully active system, that can decide on all aspects of its learning strategy, we also presented 2 simpler architectures that make fewer active choices. SGIM-D (Socially Guided Intrinsic Motivation by Demonstration) makes the fewest active choices. **SGIM-D only makes choices when it explores autonomously, of which outcome to focus on, and which policy to try.** This simple architecture allowed us to test whether the combination of autonomous intrinsically motivated exploration and socially guided exploration could bootstrap the learning process. The simple architecture allowed us to analyse the reasons of this bootstrapping effect. Then, we designed SGIM-IM (Socially Guided Intrinsic Motivation with Interactive learning at the Meta-level). **SGIM-IM can decide on an additional aspect of its learning strategy: when to imitate and when to explore autonomously.** We tested (1) the hypothesis active learning based on empirical measure of competence progress could be used at the meta-level, to choose the best sampling mode; and (2) that competence progress can be an empirical measure for simultaneous active choices on the 3 levels of its exploration of policy, outcome and sampling modes. Finally, we built **SGIM-ACTS to fully decide on what and how to learn; what, when and whom to imitate.** We tested how a strategic learner could adapt in a complex high-dimensional continuous environment with several teachers to learn an open-ended number of tasks and tasks of different types.

Hypothesis

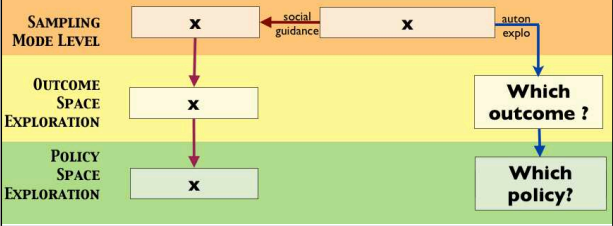
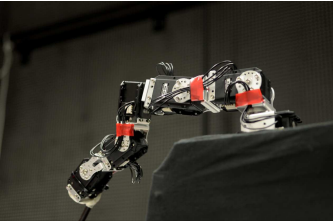
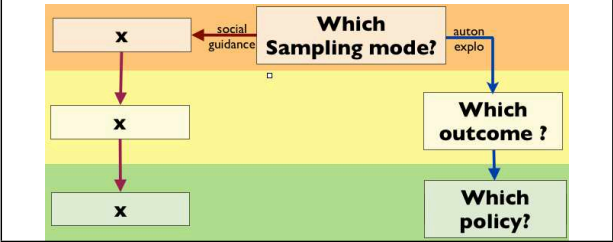
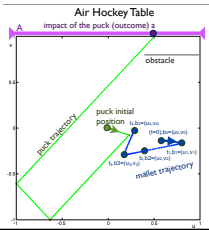
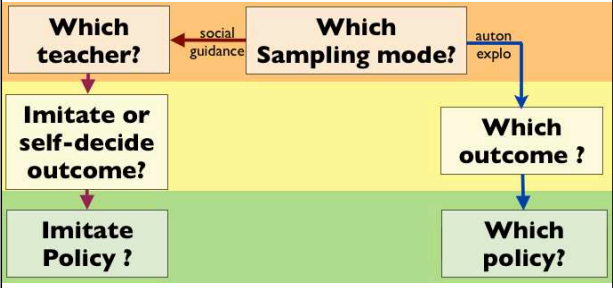
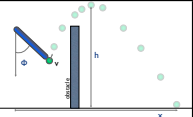
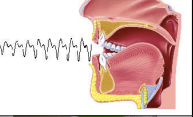

| Active learning | Experimental setup | Hypotheses to test |
|---|---|---|
| <p>SGIM-D</p>  | <p>cf. chapter 3</p>  | <ul style="list-style-type: none"> - Can a combination of autonomous exploration and social guidance bootstrap learning? - What are the properties of this combination that bootstrap the learning process? |
| <p>SGIM-IM</p>  | <p>cf. section 4.1</p>  | <ul style="list-style-type: none"> - Can competence progress be a measure for an active choice of when to imitate? - Can an learner make empirical active choice on the 3 levels of its exploration to learn effectively? |
| <p>SGIM-ACTS</p>  | <p>cf. section 4.3</p>  <p>cf. chapter 5</p>  <p>cf. chapter 1</p>  | <ul style="list-style-type: none"> - Can competence progress be a measure for an active choice of what, when, whom to imitate? - Can the learner adapt to in a high-dimensional continuous environment with several teachers to learn an open-ended number of tasks and tasks of different types. |

Figure 1.4.3: Three algorithmic architectures are presented with various illustrative experiments. The details of each algorithm and experiment can be read in the corresponding section or chapter. Each algorithmic architecture allows the agent to take active control of various aspects of its learning strategy, and to test hypotheses about active learning.

This thesis is structured as follows. We present successively three algorithmic architectures, each one allowing the agent to take more active control of its learning strategy. We have built these algorithms for quick and accurate multi-task learning, grounding their designs on developmental studies of intrinsic motivation and imitation learning. Finally, we extend SGIM-IM to a system which learns **how, when, what and who to imitate**. SGIM with Active Choice of Teacher and Strategy (SGIM-ACTS) considers the case when more teachers are available and more modes can be chosen. The first one presented in Chapter 3, called for Socially Guided Intrinsic Motivation by Demonstration (SGIM-D), actively explores policy and outcome spaces using both imitation learning and goal-oriented autonomous exploration. The design of SGIM-D allows an analysis of the complementarity between these two sampling modes. SGIM-D thus answers the questions of **what and how to learn**. The second one, called SGIM with Interactive learning at the Meta level (SGIM-IM), allows the interactive learner to actively explore the policy, outcome, and also sampling mode space. SGIM-IM learns **when to imitate** by additionally choosing actively which of the two sampling modes to use. The third one, called SGIM with Active Choice of Teacher and Strategy (SGIM-ACTS), extends SGIM-IM and considers this active choice when more teachers are available and more modes can be chosen. SGIM-ACTS explores the questions of **how, when, what and who to imitate**. SGIM-IM and SGIM-ACTS are described in Chapter 4. Finally, while in the previous chapters we aimed at algorithmic efficiency for building artificial systems, chapter 5 explores how the SGIM architecture can be useful to model and understand better infant development. We use the developed algorithms to study an observation in child psychology: the development of vocalisation in babies. We illustrate how an embodied agent using SGIM-ACTS can learn to vocalise and the emergence of a developmental sequence. This work also studies whether the algorithmic architecture SGIM-ACTS can be instantiated with different model representations and optimisation algorithms. Finally, the limits and possible extensions of these contributions are discussed in Chapter 6.

In this introduction chapter, we explained the background of our study: how can agents learn to perform various skills in life-long learning? Such learning faces the problem of an open-ended learning in a large and complex environment. The search space is very large whereas the agent only has a limited life-time. Our idea is to make the agent learn online and incrementally to choose what to learn and how to learn it. We name our agent a *strategic learner*. It learns a data collection strategy inspired by learning processes by humans who explore their environments guided by both social guidance and intrinsic motivation. Our approach thus belongs to active learning and developmental cognitive robotics. In the next chapter, we illustrate this approach by a simple case study in discrete spaces A and B , where the humanoid robot iCub learns to recognise 3D objects by active manipulation.

[Q]uoique personne ne puisse parvenir à tout savoir, il faut néanmoins qu'il soit possible de tout apprendre.

Talleyrand, Rapport sur l'instruction publique,
septembre 1791

2

Learning to Recognise 3D Objects by Curiosity-Driven Manipulation

Contents

| | | |
|-------|---|----|
| 2.1 | Problem Description | 28 |
| 2.1.1 | Experimental Protocol | 28 |
| 2.1.2 | Mathematical Formalisation | 30 |
| 2.2 | Methods | 30 |
| 2.2.1 | Scene Perception and Learning Algorithm | 30 |
| 2.2.2 | Action | 33 |
| 2.2.3 | Decision Making | 33 |
| 2.2.4 | SGIM-ACTS Architecture | 34 |
| 2.3 | Experimental Results | 35 |
| 2.3.1 | Experimental Platform | 35 |
| 2.3.2 | Evaluation of the Learning Process | 35 |
| 2.3.3 | Results | 35 |
| 2.4 | Conclusion | 39 |

We illustrate the formal description of a strategic learner from chapter 1, by a simple case study where the agent estimates the probability distribution over discrete random variables a and b . The robot learns to **associate a camera view with an object by episodes where it collects new data of images and labels**. At each episode, it has to decide which object it wants to learn more, which manipulation to use as an exploration mode to collect new visual data about the object. Once the object has been manipulated, the robot acquires a new image of the object, for which it computes its competence at recognising the right object. This new data is used to improve the recognition algorithm and learn to better distinguish between objects, while it is also used to update the meta-learning system which decides actively how to manipulate objects. In this section, we focus, not on the classification algorithm which is described in section 2.2.1, but on the exploration method: how does the robot generate new images by deciding a manipulative mode?

After a presentation of the learning problem in section 2.1, we describe the algorithms used in section 2.2. Finally in section 2.3, we present the results of the learning experiment. The results presented in this chapter have been partially published in (Nguyen et al., 2013) and Ivaldi et al. (2013). They are part of a collaboration with the Macsi Project, where we had a central role in designing and carrying out the experiments, and in evaluating the curiosity and the interaction system.

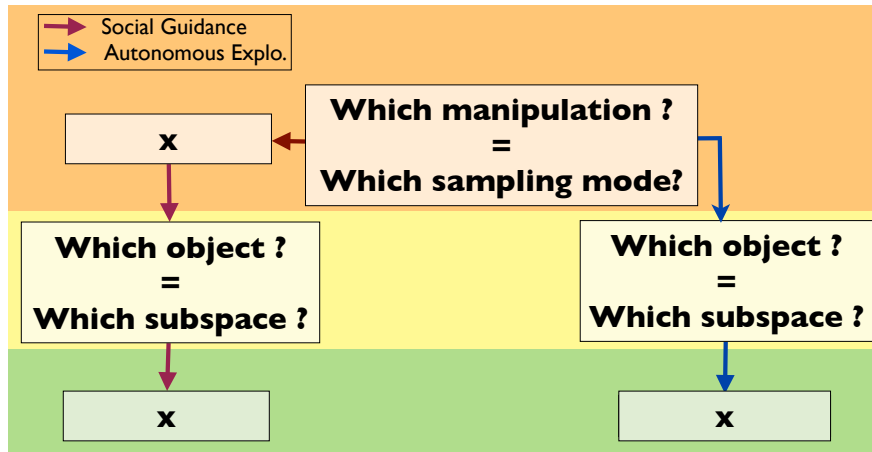


Figure 2.1.1: The strategic learner samples data by actively choosing two aspects of its exploration: its sampling mode and the object to focus on.

2.1 PROBLEM DESCRIPTION

This project aims at learning to recognise objects through manipulation. Its long-term goal is to learn to recognise objects with its visual features but also by their response to actions on them, or in other terms their affordance. In a first step, we leave out the affordance characteristics of the objects, and we design an experiment for learning to recognise objects by revealing its visual features through manipulation. In this section, we describe shortly our experiment, then re-formulate the general formalisation of section 1.4.2 for the particular case of classification with several data sampling strategies which encapsulates our recognition learning problem. In this study, as summarised in **Figure 2.1.1**, our strategic student makes active choices at each episode about: which sampling mode to use, and which object to focus on.

2.1.1 EXPERIMENTAL PROTOCOL

In this experiment, a robot learns to recognise objects $b \in B$ and their 3d views $a \in A$ by interacting with the objects or a caregiver. At each episode, the robot can decide to perform the actions autonomously or to ask the caregiver to manipulate an object in order to collect new data and improve its classification algorithm. Precisely, the system chooses an object to manipulate owing to function ψ , and a sampling mode owing to function χ among the following:

- χ_1 : push the object chosen by ψ
- χ_2 : take the object ψ , lift it, and let it fall on the table
- χ_3 : ask the human to manipulate a object specified by ψ

In the experiment, the human first presents and labels each of the objects one by one and lets the iCub manipulate them. At any time, the robot can ask the caregiver to switch to a specific object. It thus knows which object b it is manipulating. During the execution of the action, the vision processing system is inactive. When the action is completed, the object is generally immobile on the table (notably, in a different pose), and the vision system is triggered. After each manipulation, the robot tests which object it associates with

the new object image, computes a confidence measure on its capability to recognise the object, and sends the evaluation results to the curiosity system, before gathering new knowledge about the object and updating its recognition model L with the known label b . Depending on the progress, the strategic learning system based on curiosity decides the next action to trigger.



Figure 2.1.2: The objects used during the experiments: some coloured cubes, a yellow car, a grey dog, a violet/blue ball, a red bear. Left and right images respectively show the front/rear sides of the objects.

The objects b used in the experiments are shown in **figure 2.1.2**: notably, some objects are more “challenging” to recognise because their appearance is different depending on their side (generally their colour, but also their size - in the case of cubes and bear). The space of objects B is the following set :

- a grey dog-shaped stuffed toy. Its colour and shape are quite different from the others, and it is therefore easy to recognise it.
- a purple and blue coloured ball. The colours and shape are quite different from the other objects, so it is quite easy to distinguish. However, because the two sides of the ball are of different colours, more samples are required to associate the different views to the ball.
- a red teddy bear. Its colour and shape are quite easy to recognise, but it can be confused with the cubes which also have red parts.
- a yellow car. This toy offers numerous views depending on its orientation and position on the table. We expect such a toy to arouse the interest of an agent because of its rich “perceptive affordance”. Moreover, the toy has the same colour as parts of the cubes, and almost the same shape as some views of the cubes (when a lateral view shows only the yellow cubes). Thus its classification may be difficult.
- a patchwork of yellow-red-green cubes. This toy also offers numerous views depending on its orientation and position. This object is the most tricky to recognise as it can be confused with both the car and the teddy bear.

2.1.2 MATHEMATICAL FORMALISATION

Our agent learns to associate a camera view $a \in A$ with an object $b \in B$. It thus has to estimate the probability distribution $\tilde{p}(b|a)$, which represents the real id of the image. At each episode e , it gets a new object view and it updates its estimate $p_e(b|a)$.

Thus B is the set of objects to be recognised. A is the space of all possible rgb-d images. In our experimental setting, as each image is of resolution 480×640 , A is of dimension $4 \times 480 \times 640$. The space A of images generated with the various objects is infinite, as an infinity of images correspond to the same object seen through different angles, positions and distances.

Let $\gamma_e(a)$ be a measure of competence at recognising the right object in image a with the estimation p_e . It corresponds to the inverse of the cost function J introduced in section 1.4. More precisely $\gamma_e(a)$ is the inverse of $J(a, \tilde{p}(\alpha | \operatorname{argmax}_b(p_e(b|a))))$. Our goal is to recognise all objects, i.e. to maximise:

$$I_e = \sum_a p(a) \gamma_e(a) \quad (2.1)$$

where $p(a)$ is a probability over A that a appears to the robot. The agent is endowed with a learning algorithm \mathcal{L} for recognising different views of objects. The learner must sample images of each object to give data to this recognition algorithm \mathcal{L} . While classical active learning methods choose images $a \in A$ and then ask for their labels $b \in B$, our method mainly explores the object space B by choosing first an object (function ψ of section 1.4.2), and generating images with 3 different sampling modes χ : it can push the object, lift and drop the object, or it can ask a human to manipulate an object. These different modes have different costs $\kappa(\chi)$ that take into account the time cost, energy cost, caregiver effort of each mode. In this study $\forall \chi, \kappa(\chi)$ are set to the same value 1.

To summarise, $\Psi = \bigcup_{b \in B} ((A \times \{b\})^{\mathbb{Z}})^2$ and \mathcal{X} is a set of 3 predefined sampling modes. At each episode the robot has to decide on a data collection method: which object it wants to learn more, which manipulation to use as a sampling mode to generate new sample data, in order to learn to distinguish between objects. In the next section, we describe the learning algorithm \mathcal{L} and the meta-learning method \mathcal{A} for exploration.

2.2 METHODS

In this section, we focus, after a short description of the classification algorithm \mathcal{L} in section 2.2.1 and the control system in section 2.2.2, on the exploration method \mathcal{A} : how the robot generates new images by deciding a mode of manipulation.

2.2.1 SCENE PERCEPTION AND LEARNING ALGORITHM

The perceptual system of the robot is a RGB-D sensor placed over the area where the interaction with objects and caregivers takes place. From these camera images are extracted features. A learning algorithm \mathcal{L} then makes statistical inferences to associate these features with objects.

The object learning and recognition module has been designed with the constraints of developmental robotics in mind. It uses minimal prior knowledge of the environment. A short overview is given here. Details can be found in (Lyubova and Filliat, 2012).

All information about the visual scene is incrementally acquired as illustrated in Fig. 2.2.1. The main processing steps include the detection of physical entities in the visual space as proto-objects, learning their appearance, and categorising them.

At the first stage of our system the visual scene is segmented into *proto-objects* (Pylyshyn, 2001) that correspond to units of visual attention defined from coherent motion and appearance. Assuming that the visual attention of the robot is mostly attracted by motion, proto-object detection starts from optical flow estimation, while ignoring the regions of the scene that are far away according to the constraints of the robot’s workspace. Then, a tracker is used to extract features inside moving regions and to group them based on their relative motion and distance. Each cluster of coherently moving points is associated with one proto-object and its contour is defined according to the variation of depth. Each proto-object is therefore tracked across frames and finally identified as an already known or a new entity.

Each proto-object appearance is incrementally analysed by extracting low-level visual features and grouping them into a hierarchical representation. As a basis of the feature hierarchy we use SURF points (Bay et al., 2006) and colour of superpixels (Micusik and Kosecka, 2009) obtained by segmenting the scene into regions of similar adjacent pixels. These low-level features are grouped into pairs and triples incorporating local geometry and called mid-features. Both low- and mid-level features are quantised into dictionaries of visual words. The Bag of visual Words approach with incremental dictionaries (Filliat, 2007) is used to characterise the appearance of entities from different viewpoints that we call *views*. Views are encoded by the occurrence frequency of extracted mid-features. An overall entity appearance is characterised by a multi-view model constructed by tracking an entity across frames and collecting its views occurrence frequency.

Besides tracking, the association of the current view to an entity can also be based on appearance recognition when an object appears in the field of view. In this case, appearance-based *view* recognition is performed first, using all extracted mid-features to participate in a voting procedure that uses the TF-IDF (Term-Frequency - Inverse Document Frequency) (Sivic and Zisserman, 2003) and a maximum likelihood approach. If the recognition likelihood is high, the view is identified as the most probable among already known views; otherwise, a new view is created. Then, appearance-based *entity* recognition is performed using an occurrence statistics of views among known entities.

Using the ability to categorise entities, the models of objects previously constructed during their observation can be improved during robot interactive actions. Since the manipulated object does not change during the robot action, its corresponding model can be updated with recognised views. The updates with recognised views reduce noise in object models, while the updates with new views allow the robot to accumulate views corresponding to unseen perspectives of the objects.

At each image $a \in A$ seen, \mathcal{L} computes the likelihood for each already known view, and returns the two highest likelihood measures p_{m1}, p_{m2} , as well as the objects b_{m1}, b_{m2} of the objects associated with the views, and the number n_{m1}, n_{m2} of known views for each of the objects. As through social interaction, the caregiver teaches to the iCub the object b_g of the object he is manipulating, the robot can estimate its competence at distinguishing b_g from other objects, with the dissimilarity of likelihood measures between the 1st object associated and the 2nd object associated, and by estimating its gain of information about the object by collecting new views. We consider that the competence is high when the learner’s top two answers are correct. In this case, its competence should be proportional to its confidence, or probability of giving these correct answers. If among its top two answers, it only gets one right, the competence is lower and should

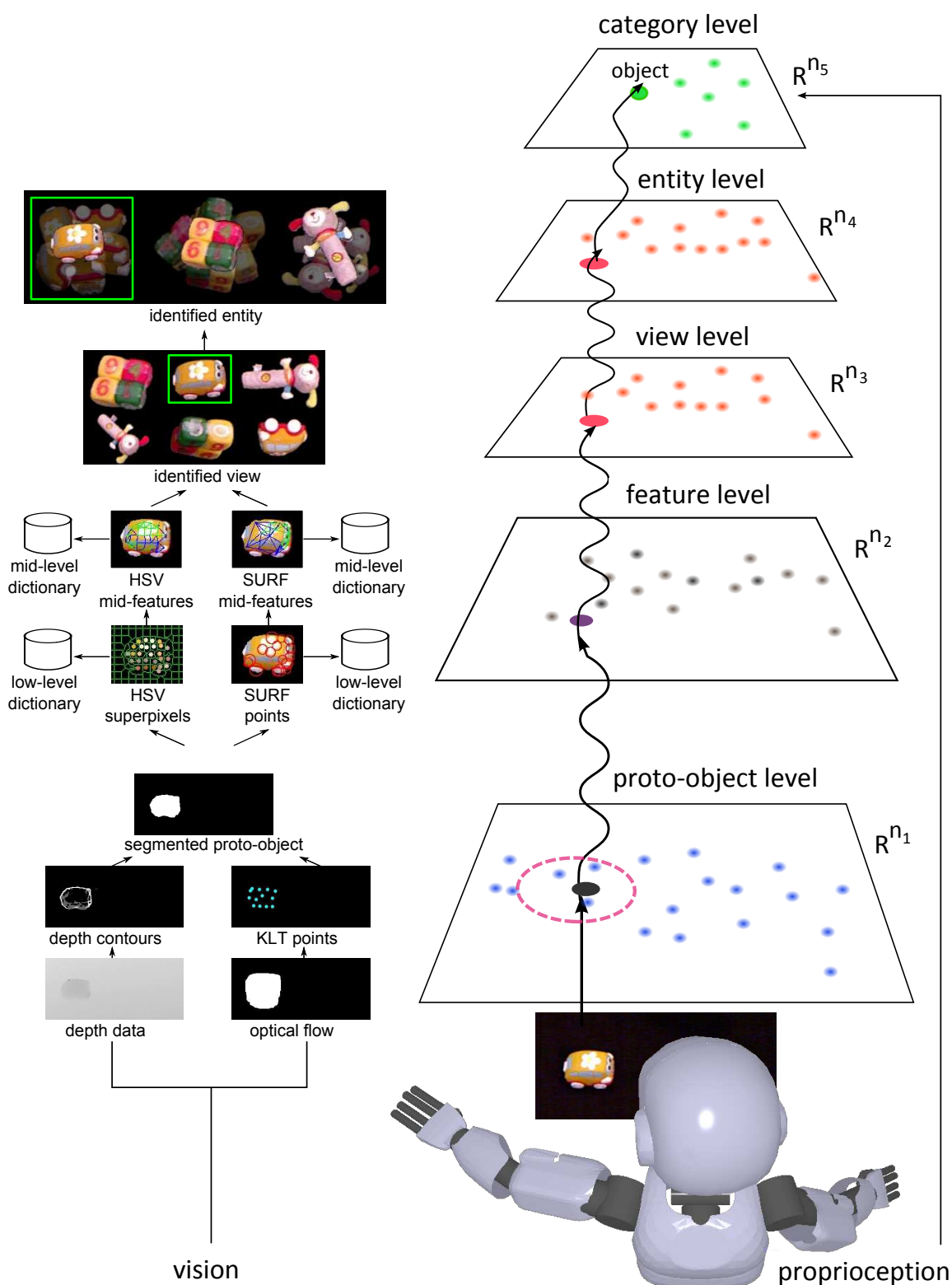


Figure 2.2.1: The visual information is processed through a hierarchy of layers, which elaborate the camera images to extract the entities in the scene. The proprioceptive information from the robot is used for categorising the entities.

be related to the ratio between the probability of the correct answer and the incorrect answer. Finally, if its top two answers are incorrect, it gets a low score. The competence at recognising object b_g in image a is thus defined as

$$\gamma(b, a) = \begin{cases} n_{m1} \times p_{m1} + c_1 & \text{if } b_g = b_{m1} = b_{m2} \\ n_{m1} \times p_{m1} / (1 + p_{m2}) + c_1 & \text{if } b_g = b_{m1}, b_g \neq b_{m2} \\ n_{m2} \times p_{m2} / (1 + p_{m1}) + c_1 & \text{if } b_g \neq b_{m1}, b_g = b_{m2} \\ c_1 & \text{if } b_g \neq b_1, b_g \neq b_{m2} \end{cases}$$

where c_1 is a constant, set to -1 in our experiment. γ corresponds to the inverse of the cost function J introduced in section 1.4.

2.2.2 ACTION

An action module controlling the robot exposes a set of high level commands to the perceptive and cognitive modules. It acts as intermediate controller for speech and motor joints. Modules can send commands to the robot, specifying the type of action and a variable list of parameters (the object properties, e.g. location on the table, orientation; *etc.*). Actions can be simple (for example the primitives *push*, *speak*) but also more complex (as taking an object, lifting it and dropping it on the table). More specifically in our experiment, we use 2 motor primitives parameterised by the object position for pushing and dropping an object, and a speech primitive parameterised by the object label.

2.2.3 DECISION MAKING

Differently from (Ivaldi et al., 2012), where social guidance was restricted to the mere execution of commands received from the caregiver, in this study the robot takes its decisions autonomously based on intrinsic motivation and curiosity. We hereinafter describe in detail the *Socially Guided Intrinsic Motivation with Active Choice of Teacher and Strategy* (SGIM-ACTS) algorithm.

Our learner improves its estimation p of \tilde{p} to maximise $I = \sum_a P(a)\gamma(a)$ both by self-exploring A and B spaces. It generates new perception samples by manipulating the objects and by asking for help to a caregiver, who hands the objects to the robot. When an object is placed on the table, a rgb-d image $a \in A$ is retrieved at each step. SGIM-ACTS learns by episodes during which it actively chooses both an object $b \in B$ to learn to recognise and a sampling mode χ between: pushing the object, taking and dropping the object or asking the caregiver to manipulate the object. For each object b it has decided to explore, it also chooses the mode χ which maximises its *competence progress* or *interest*, defined as *the local competence progress, over a sliding time window of δ for an object b with mode χ at cost $\kappa(\chi)$* . If the competence measured for object b with mode χ constitute the list $R(b, \chi) = \{\gamma_1, \dots, \gamma_N\}$:

$$interest(b, \chi) = \frac{1}{\kappa(\chi)} \frac{\left| \left(\sum_{j=N-\delta}^{N-\frac{\delta}{2}} \gamma_j \right) - \left(\sum_{j=N-\frac{\delta}{2}}^N \gamma_j \right) \right|}{\delta} \quad (2.2)$$

This sampling mode enables the learner to generate new samples a in subspaces of A . The SGIM-ACTS learner explores preferentially objects where it makes progress the fastest. It samples views from the object to improve its vision system, re-using and optimising the recognition algorithm built through its different exploration modes.

This behavioural description of SGIM-ACTS is completed in the next section by the description of its architecture.

2.2.4 SGIM-ACTS ARCHITECTURE

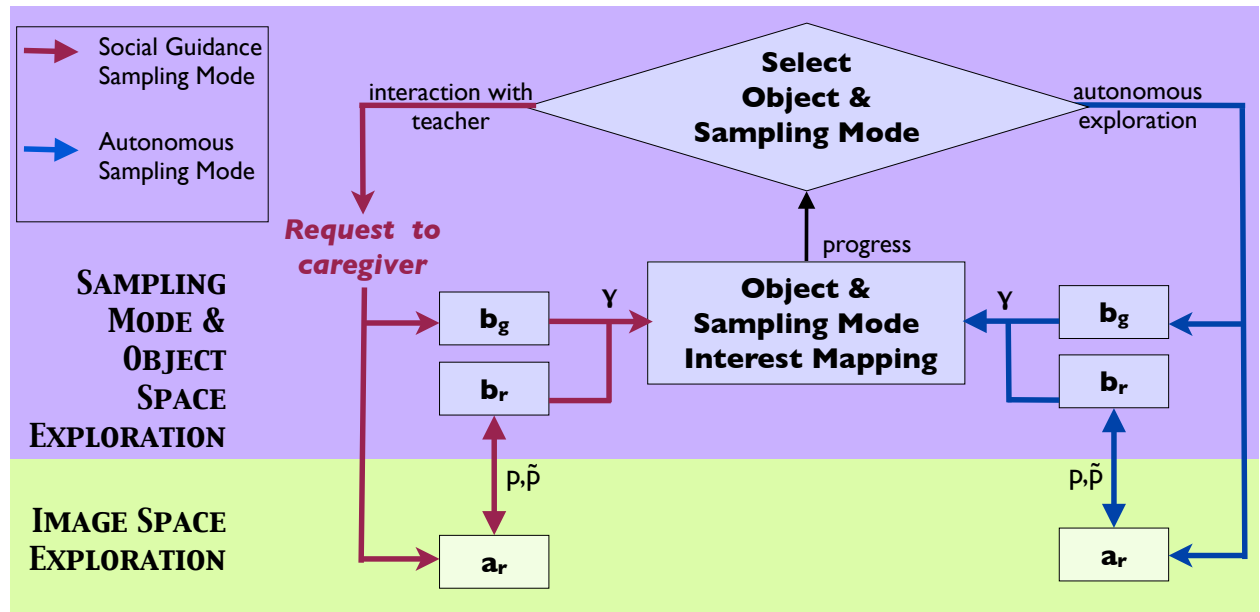


Figure 2.2.2: Time flow chart of SGIM-ACTS, which combines Intrinsic Motivation and Social Guidance exploration modes into 2 layers: the sampling mode and object space B exploration, and the image space A exploration.

SGIM-ACTS is an algorithm based on interactive learning and intrinsic motivation. It learns to recognise different objects by actively choosing which object $b \in B$ to focus on (ψ), and which sampling mode χ to adopt to learn local inverse and forward models. Its architecture is separated into two levels as described in **Algorithm 2.2.1**:

- A *Mode and Label Space Exploration* level which decides actively which object b_g to set as a goal, which sampling mode χ to adopt, and which object to manipulate. To motivate its choice, it associates to each $b \in B$ an interest level for each mode (*Goal Interest Mapping*). As detailed in **Algorithm 2.2.2**, the interest is computed as the competence progress at recognising the object b with the strategy χ over a time window of δ . As described in **Algorithm 2.2.3**, the function (*Select Label and Mode*) selects objects and strategies stochastically mainly according to their competence progress (line 8). To ensure a minimum of exploration, it sometimes selects randomly any objects and modes (line 5).

- A *Image Space Exploration* level that explores A , according to the object b_g and sampling mode χ chosen by the Mode and Label Space Exploration level. With each chosen mode/manipulation, different samples (a_r, b_r) are generated to minimise γ , while improving its estimation of p_e and discriminating b_g from other objects. It finally returns the competence measure $\gamma(b_g)$ to the Mode and Label Space Exploration level.

2.3 EXPERIMENTAL RESULTS

2.3.1 EXPERIMENTAL PLATFORM

Experiments are carried out with a 53 DOF full-body humanoid robot, called *iCub* (Natale et al., 2012). The whole upper-body has been used in our experiments: head, torso, arms and hands, for a total of 41 DOF. Thanks to proximal force sensing, the main joints (arms, torso) are compliant (Ivaldi et al., 2012).

All software modules used in the experiments of this chapter belong to the MACSi software architecture (Ivaldi et al., 2012).

2.3.2 EVALUATION OF THE LEARNING PROCESS

To evaluate the efficiency of our algorithm, we compare our SGIM-ACTS exploration architecture with the random algorithm where the agent would choose at each episode a random object and a random mode. To evaluate the efficiency of each algorithm, we freeze the learning process after each episode and evaluate the classification accuracy on an image database, made up of 64 images of each object in different positions and orientations built independently from the learning process (see **Figure 2.3.1** for a sample).

2.3.3 RESULTS

We conducted the experiments with each of the algorithms (SGIM-ACTS and random) under two conditions: with an unbiased teacher who shows objects to the learner under different angles; and with a biased teacher who always shows the same view of each object. We plot results for each case of exploration, sampling mode and teacher, detailing the learning performance separated by object. We plot the f-measure (i.e. the harmonic mean of precision and recall (van Rijsbergen, 1979)) and the number of images correctly recognised in the evaluation database.

As shown in **Figure 2.3.2** the progress in recognition is better with SGIM-ACTS than with random exploration, for both teachers. At the end of the experiments, the SGIM-ACTS learner is able to correctly recognise the objects in 57 over 64 images, against 50 in the case of the random learner.

Figure 2.3.3 plots how well the system can distinguish objects, and which objects it manipulates for example experiments under different conditions. We can see in **Figure 2.3.3a**, **2.3.3b**, **2.3.3c** and **2.3.3d** that the random learner often switches objects, and explores equally all objects, while the SGIM-ACTS

Algorithm 2.2.1 SGIM-ACTS for Discrete Outcomes Spaces

1: **Input:** s_1, s_2, \dots : available sampling modes with cost κ_i .
 2: **Initialization:** $\mathcal{R} \leftarrow$ singleton $\{B\}$.
 3: **Initialization:** $\mathcal{H} \leftarrow$ empty episodic memory of sets of associations (a,b).
 4: **loop**
 5: $\chi_i, b_g \leftarrow$ **Select Label and Sampling Mode**(\mathcal{R})
 6: **repeat**
 7: **if** χ_i is a Social Guidance learning mode **then**
 8: $(a_r, b_r, b_g) \leftarrow$ Interact with caregiver with mode χ_i .
 9: **else if** χ_i is an Auton. Exploration learning mode **then**
 10: $(a_r, b_r, b_g) \leftarrow$ Perform action with mode χ_i .
 11: **end if**
 12: **Learning:** Update p with (a_r, b_r) .
 13: $\gamma \leftarrow$ Competence for b_g
 14: **until** end of trials for the same object
 15: **Meta-Learning:** $\mathcal{R} \leftarrow$ Update **Goal Interest Mapping**($\mathcal{R}, \mathcal{H}, b_g, \gamma$)
 16: **end loop**

Algorithm 2.2.2 $[\mathcal{R}] =$ Goal Interest Mapping($\mathcal{R}, \mathcal{H}, b, \gamma$)

1: **input:** b : set of objects and corresponding $interest(b, \chi)$ for each strategy χ .
 2: **input:** δ : a time window used to compute the interest.
 3: Add γ to $R(b, \chi)$, the list of competence measures for object $b \in B$ with strategy χ .
 4: Compute the new value of competence progress of b :

$$interest(b, \chi) = \frac{1}{\kappa(\chi)} \frac{\left| \left(\sum_{j=N-\delta}^{N-\frac{\delta}{2}} \gamma_j \right) - \left(\sum_{j=N-\frac{\delta}{2}}^N \gamma_j \right) \right|}{\delta}$$

6: **return** \mathcal{R} the set of all $R(b, \chi)$

Algorithm 2.2.3 $[\chi, b_g] =$ Select Label And Sampling Mode(\mathcal{R})

1: **input:** \mathcal{R} : set of regions R_n and corresponding $interest_{R_n}(\chi)$ for each strategy χ .
 2: **parameters:** $0 \leq p_1 \leq 1$: probability for random mode.
 3: $p \leftarrow$ random value between 0 and 1.
 4: **if** $p < p_1$ **then**
 5: Ensure a minimum of exploration, i.e. :
 6: Choose χ and $b_g \in B$ randomly
 7: **else**
 8: Focus on areas of highest competence progress, i.e. :
 9: $\forall (\chi, n), P_n(\chi) \leftarrow \frac{interest_{R_n}(\chi) - \min(interest_{R_i})}{\sum_{i=1}^{|R_n|} interest_{R_i}(\chi) - \min(interest_{R_i})}$
 10: $(b, \chi) \leftarrow argmax_{n, \chi} P_n(\chi)$
 11: **end if**
 12: **return** (b, χ)

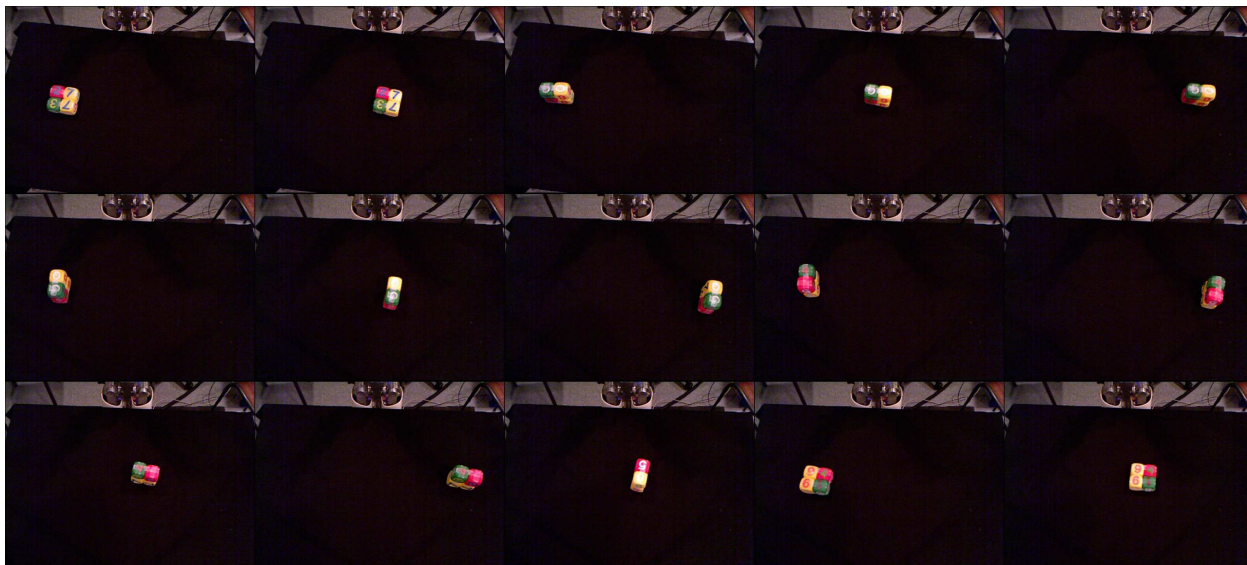


Figure 2.3.1: A portion of the database of objects views used for evaluating the recognition performance: precisely, the images related to the cubes.

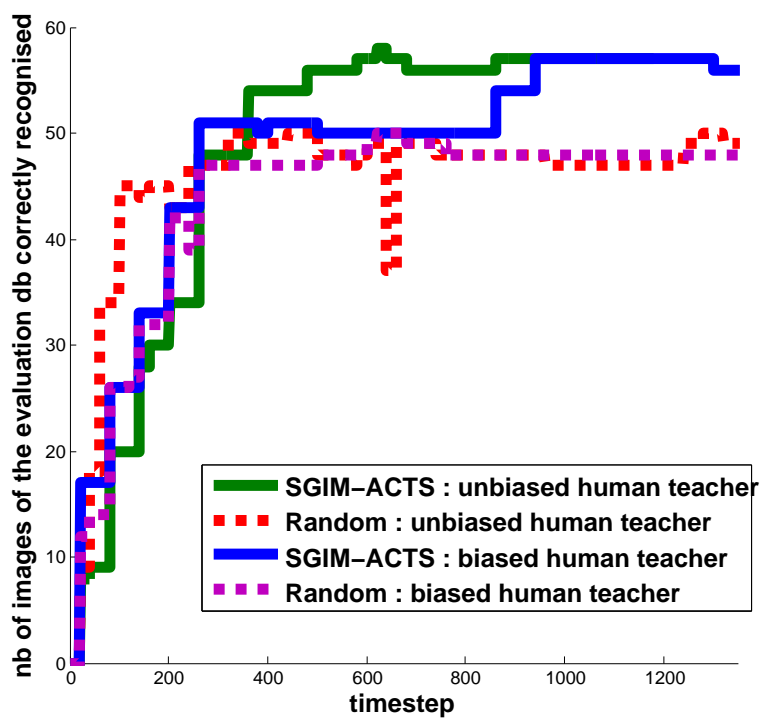


Figure 2.3.2: SGIM-ACTS vs Random: recognition performance, i.e. the number of images of the evaluation database correctly recognised by the two exploration modes with two different responses of the teacher (see text).

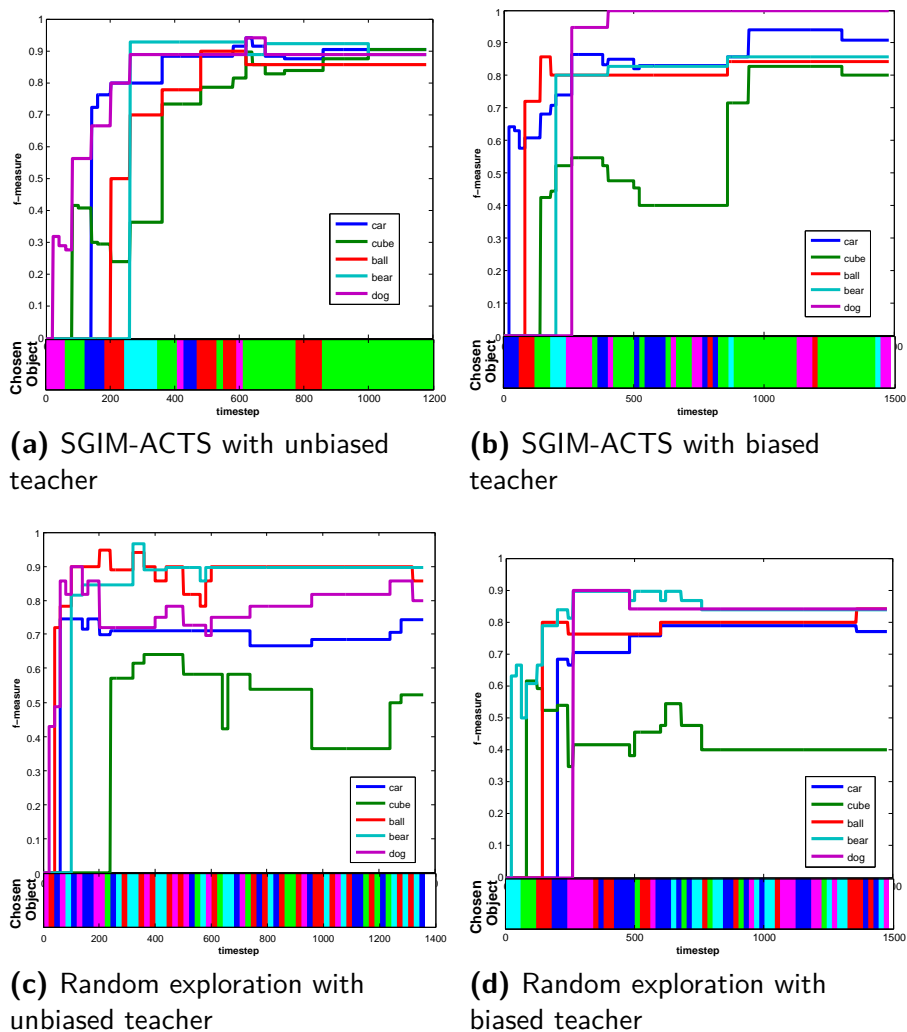


Figure 2.3.3: f-measure on the evaluation database, with respect to time. The bottom part of the plot shows the manipulated object at each timestep.

learner focuses on objects for longer periods of time. We note that SGIM-ACTS manipulates more the cubes, especially when its competence progress increases. Indeed, as stated above, the cube is the most complex of objects because it offers very different views due to its various colours, but also because it can easily be confused with other objects that bear the same colours. Manipulating it brings every time more information about the object since their appearance changes substantially depending on the action (a frontal view consists of four cubes, while a lateral view consists of two cubes only, and depending on the side it could be yellow or red/green), and improves its discrimination from other objects. The iCub has spent 54% and 51% of its time learning about cubes with SGIM-ACTS for both teachers. The system thus allocates more time for the difficulties.

Overall, the iCub focuses its attention on complex objects, asking human intervention or manipulating autonomously to improve its recognition capability. **Figure 2.3.3a** clearly illustrates this mechanism: the red bear (cyan line) is easily recognised, hence the robot does not ask again to interact with the object once

it is learnt; conversely, the cubes (green line) are difficult to recognise, hence the robot focuses more on them.

Conversely, as shown in **Figure 2.3.3c**, in the “random” case the robot does not focus on any particular object. Hence, the recognition performance at the end of the experiment is worse, because the “difficult” objects (such as the cubes - green line) are not sufficiently explored.

Furthermore, the SGIM-ACTS algorithm is robust to the quality of the teaching, for the recognition performance is high in both cases. Whether the teacher helps by showing new views of objects and bringing new information, the learner improves its discrimination of objects. This is to contrast with the case of the random algorithm who is dependent on the teacher. We can see that the f-measures of **Figure 2.3.3d** are lower than in **Figure 2.3.3c**. Again, SGIM-ACTS is able to recognise how profitable a teacher can be, and choose to take advantage of him or not.

2.4 CONCLUSION

In this chapter, we described a method to choose actively a data collection strategy in order to learn fast how to recognise objects, which exploits curiosity to guide exploration and manipulation, such that the robot can improve its knowledge of objects and take advantage of teachers in an autonomous and efficient way. The autonomous mode driven by intrinsic motivation has been fruitfully integrated in the Cognitive Architecture developed in the MACSi project. Experimental results show the effectiveness of our approach: the humanoid iCub is now capable of deciding autonomously which actions must be performed on objects in order to improve its knowledge, requiring a minimal assistance from its caregiver.

In this experiment, the robot learns object categories based on views of the objects and visual characteristics extracted from the images. Manipulation here is a means of revealing these different views and visual characteristics. A follow up study should use a more developmental approach and study how agents can recognise objects with respect to their visual characteristics but also their affordance, or their physical response to various manipulations.

In conclusion, in the long-term **SGIM-ACTS algorithm yields better performances, because it facilitates learning all objects dedicating more time and efforts to the complicated objects.** Guiding data collection with SGIM-ACTS yields to better results than Random exploration in the case of discrete variables. In the next chapter, we extend this case with the study of the combination of intrinsic motivation and social guidance for learning the probability density for continuous and high-dimensional spaces.

[I] faut bien que l'on interpole; l'expérience ne nous donne qu'un certain nombre de points isolés, il faut les réunir par un trait continu; c'est une véritable généralisation. Mais on fait plus, la courbe que l'on tracera passera entre les points observés et près de ces points; elle ne passera pas par ces points eux-mêmes. Ainsi on ne se borne pas à généraliser l'expérience, on la corrige;

Henri Poincaré, La science et l'hypothèse, 1902, réédition Flammarion Paris
1968, p. 159

3

Learning in Complex Continuous Environments

Contents

| | | |
|---------|---|----|
| 3.1 | A Passive Learner Benefitting from the Combination of Imitation Learning and Autonomous Exploration . . . | 41 |
| 3.1.1 | Chosen Sampling Modes for a Passive Learner | 42 |
| 3.1.1.1 | Social Guidance | 42 |
| 3.1.1.2 | Autonomous Exploration with Intrinsic Motivation | 43 |
| 3.1.2 | Problem Statement and Assumptions for Motor Learning | 43 |
| 3.1.3 | SGIM-D Overview | 45 |
| 3.1.4 | SGIM-D Architecture | 47 |
| 3.1.4.1 | Lower Level : Policy Space Exploration | 47 |
| 3.1.4.2 | Higher Level : Active Goal Babbling for Outcome Space Exploration | 48 |
| 3.2 | The Fishing Rod Experiment | 50 |
| 3.2.1 | Motor Primitives and Correspondence Mapping | 50 |
| 3.2.2 | Mimic a Policy | 52 |
| 3.2.3 | Performance Measure | 53 |
| 3.2.4 | Goal-Directed Policy Optimisation | 53 |
| 3.2.4.1 | Global Exploration Regime | 54 |
| 3.2.4.2 | Local Optimisation Regime | 54 |
| 3.2.5 | Stochastic Environment | 56 |
| 3.3 | Experimental Protocol | 57 |
| 3.3.1 | Comparison of Learning Algorithms | 59 |
| 3.3.2 | Evaluation | 59 |
| 3.3.3 | Demonstrations | 59 |
| 3.4 | Experimental Results | 61 |
| 3.4.1 | Better Precision | 61 |
| 3.4.2 | A Wide Range of Outcomes | 63 |
| 3.4.3 | Dependence on the Size of the Outcome Space | 64 |
| 3.4.4 | Identification of the Interesting Subspaces | 64 |
| 3.5 | Analysis of the Bootstrapping Effect | 66 |
| 3.5.1 | Outcome Space Exploration | 66 |
| 3.5.1.1 | Dependence of the Performance on the Teacher | 66 |
| 3.5.1.2 | Difference in the Explored Outcome Spaces | 66 |
| 3.5.2 | Policy Space Exploration | 68 |
| 3.5.2.1 | Dependence of SGIM-D Performance on the Quality of Demonstrations | 68 |
| 3.5.2.2 | Analysis of the Demonstrated Movements | 70 |
| 3.6 | Preliminary Results on a Physical Robot | 70 |
| 3.7 | Benefits of the Combination: Conclusion | 73 |

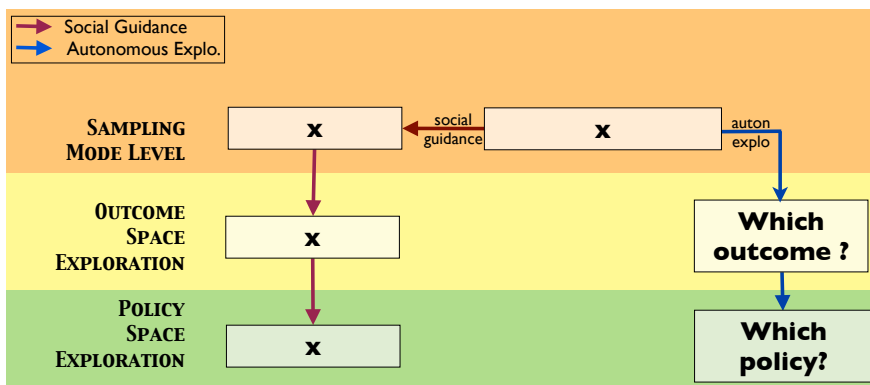


Figure 3.0.1: The strategic learner samples data by choosing a goal outcome and policy to try.

After describing the questions we address in chapter 1, we illustrated our approach in chapter 2 with an experiment where the humanoid iCub manipulates objects in order to gather more information to recognise objects from their different views. We showed that an active strategic learner can indeed enhance learning by choosing the appropriate data collection strategy. Nevertheless, the spaces considered by the classification algorithm were discrete. We would like to extend this work to continuous spaces. We first analyse whether combining autonomous intrinsically motivated exploration and socially guided exploration is advantageous in **continuous high-dimensional spaces**, as conjectured in section 1.3.3 of chapter 1.

In this chapter, we design a passive learner in the sense that it does not actively choose its sampling mode between autonomous exploration or socially guided exploration, although it makes active choices where to sample in the self-exploration mode. It only actively chooses which subregions of the environment to focus on. These active choices are summarised in **Figure 3.0.1**. We describe in section 3.1 such a goal-oriented algorithmic architecture, called Socially Guided Intrinsic Motivation by Demonstration (SGIM-D). We then implement it for a fishing robot which we describe in section 3.2. The results reported in section 3.3 and 3.4 show that the learning algorithm for motor control is enhanced by the combination of the two sampling modes. Thus in section 3.5, we investigate the reasons of this bootstrapping effect, and analyse the influence of demonstrations on the learning process. The results presented in this chapter have been partially presented in (Nguyen et al., 2011a; Nguyen and Oudeyer, 2012a, 2013a).

3.1 A PASSIVE LEARNER BENEFITTING FROM THE COMBINATION OF IMITATION LEARNING AND AUTONOMOUS EXPLORATION

As argued in section 1.3.3, combining intrinsically motivated exploration and socially guided exploration can be beneficial to learn density distributions in continuous high-dimensional spaces, such as highly redundant motor controls.

In this chapter, we present an algorithmic architecture, called **Socially Guided Intrinsic Motivation by Demonstration (SGIM-D)**, that allows a robot to learn a diverse repertoire of **parameterised motor primitives, in high-dimensional continuous spaces** similar to those used in (Kober et al., 2010; Peters and Schaal, 2008; Schaal et al., 2003; Stulp and Schaal, 2011; Kober et al., 2012; da Silva et al., 2012), but **allowing to reach a diversity of goals which spans the whole reachable**

outcome space. This system will re-use regression techniques allowing to generalise motor primitives to goals close to previously learnt goals, like in (Kober et al., 2012; da Silva et al., 2012), but will allow to self-generate and learn actively goals that are also far from those given by humans. This system will also leverage efficient techniques for active learning of inverse models using goal babbling (Baranes and Oudeyer, 2013; Rolf et al., 2010; Baranes and Oudeyer, 2010a), but extend them with a technical integration with robot learning by demonstration techniques (Billard et al., 2007). Thus, while the combination of social guidance and intrinsic motivation is similar in spirit to the one explored in (Thomaz and Breazeal, 2008), it will be technically very different and applied to learning sensorimotor skills in continuous high-dimensional spaces more alike the work in (Kober et al., 2012; da Silva et al., 2012; Stulp and Schaal, 2011).

To better integrate programming by demonstration and intrinsic motivation, in section 3.1.1, we first motivate our choice of social interaction, and the intrinsic motivation algorithm that we use. In 3.1.2, we re-formulate our formalisation of section 1.4.2 for the learning of an inverse control. We then present an overview of our SGIM-D algorithm, before detailing its architecture.

3.1.1 CHOSEN SAMPLING MODES FOR A PASSIVE LEARNER

In this chapter, we tackle a motor control learning problem to learn the inverse model which associates an outcome $\mathcal{A} \in \mathbb{A}$ in a context $\mathcal{C} \in \mathbb{C}$ to a policy $\pi \in \mathbb{P}$.

3.1.1.1 SOCIAL GUIDANCE

Among all the possibilities of social guidance detailed in section 1.2.4 of chapter 1, in the model and experiments presented in this chapter, we choose to use a feedforward signal, as it is more natural for human teachers. Among the possibilities of social guidance, our preset imitation mode is:

- **What:** We opted for an information flow targeting both policy and outcome spaces, to enable the biggest progress for the learner. As we want the learner to accomplish not only a single goal but to be efficient on a large variety of goals, we choose to bootstrap its learning with information targeting the task space. Furthermore, we also want the learning process to benefit from the social interaction early. So that the learner builds its policy repertoire quickly, we choose to target the policy space \mathbb{P} too. Here, **the learner does not choose to take into account specifically between policy or outcome space, but uses both at the same time.** This active choice will be considered in section 4.3.
- **How:** As we wish in the model and experiments presented below, to address the learning for large, complex and continuous environments, so that the robot learns a wide variety of goals/tasks, we opt for low-level demonstrations. So as to minimise the correspondence issues, we teleoperate our robot using **kinaesthetic demonstrations**, while recording from its own sensors. This choice avoids any symbolic thus discrete representations of policies or the environment, or a preset language to communicate at the high-level.
- **When:** Although interactive learning at either the learner’s or the teacher’s initiative seems interesting theoretically, it introduces combinatorially many variants. For simplicity reasons, we set the interaction **at regular frequency**, allowing easier assessment of our SGIM-D algorithm and

comparison with other learning algorithms. An interactive learner deciding autonomously when to imitate will be studied in section 4.2.

- Who: In this proof-of-concept study, we deliberately ignored the who question, which examines cases of multiple teachers. This very stimulating question yet requires a separate examination to avoid too much complexity in a single study. This question will be tackled in section 4.3. For now, we only consider **a single teacher**.

3.1.1.2 AUTONOMOUS EXPLORATION WITH INTRINSIC MOTIVATION

For the intrinsic motivation mode, as reviewed in section 1.2, a wide variety of intrinsic motivation algorithms have been developed based on knowledge, competence or morphology. One of the state-of-the-art algorithms, the Self-Adaptive Goal Generation-Robust Intelligent Adaptive Curiosity (SAGG-RIAC), is an implementation of intrinsic motivations based on measures of competence progress (Baranes and Oudeyer, 2013). It efficiently learns forward and inverse models to reach a wide range of goals in continuous high-dimensional spaces including both easy and unlearnable subparts (see (Rolf et al., 2010) for another related goal exploration algorithm). Moreover, its hierarchical structure proposes 2 levels of learning targeting the outcome and policy spaces respectively. Its goal directedness allows bidirectional mapping to our social interaction representation as [context][policy][outcome], for combining social guidance and intrinsic motivation.

3.1.2 PROBLEM STATEMENT AND ASSUMPTIONS FOR MOTOR LEARNING

We place ourselves in an episodic motor learning framework. As explained in section 1.2, we describe a timestep as a data triplet [context][policy][outcome]. Like in (Kober et al., 2012; da Silva et al., 2012; Stulp and Schaal, 2011), a robot is provided with a parameterised encoding of an outcome space (i.e. it perceives the effect of its movement as a vector or real numbers, e.g. where the tennis ball bounces) as well as a parameterised encoding of movement. **The robot has to learn the inverse model mapping all goals in the outcome space to corresponding adequate parameters of movements.** High-dimensionality in this setting concerns the dimensionality of the vector of parameters for producing movements, which can be different from the actual number of degrees of freedom of the robot since motor primitives control the time evolution of values in each degree of freedom, and this time evolution can be encoded with multiple parameters. For example, in the fishing experiment below, a robot produces a movement of its 6 DOF arm by setting the real number values of its 25 dimensional motor primitives, which controls the evolution of each DOF values by settings targets at different times (global duration being also one of these parameters). Then, it can observe the outcome of such a movement by observing where the float has arrived in the outcome space, i.e. on the surface of the water which is a 2D value. Using SGIM-D, and thus combining intrinsically motivated learning and human demonstration, the robot has to learn the complex inverse model mapping all goals/outcomes (i.e. 2D targets on the water) to adequate parameters of motor movement.

More formally we reuse the notations of section 1.4 and specify them for problems of motor learning. Let us consider an agent learning motor skills, i.e. how to induce any possible goal/task/outcome $\mathcal{A} \in \mathbb{A}$ from given contexts states $\mathcal{C} \in \mathbb{C}$ with motor programs $\pi \in \mathbb{P}$ (**Figure 3.1.1**). We parameterise the context

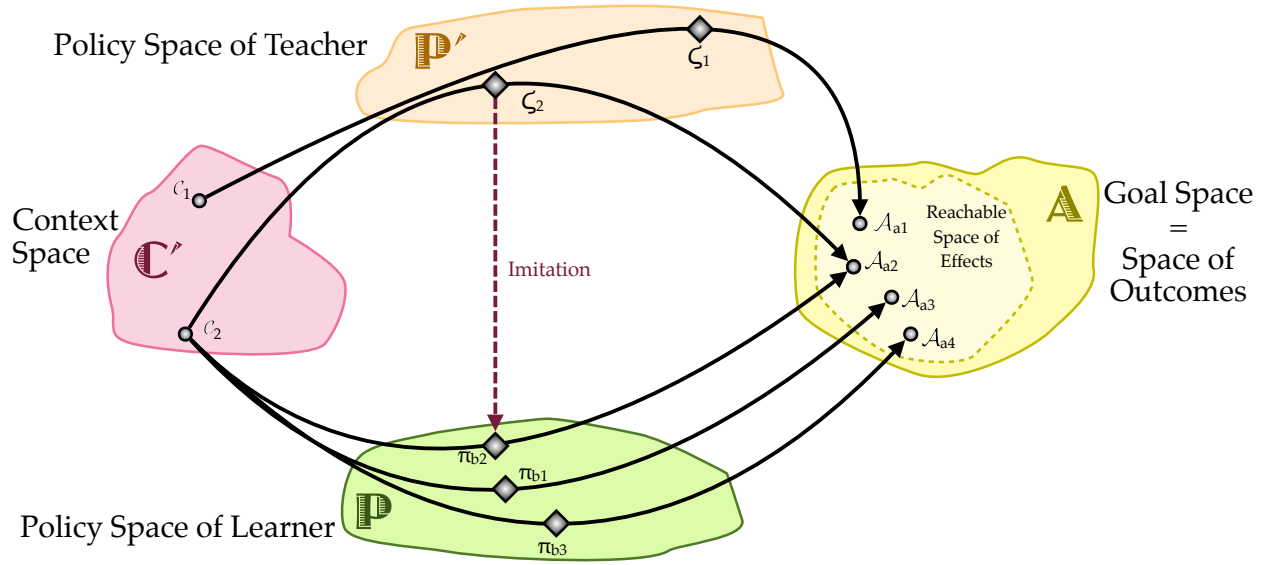


Figure 3.1.1: Representation of the problem. The environment can evolve from context state \mathcal{C} to an outcome state \mathcal{A} by means of the learner's actions with policy π or the teacher's ζ . The learner and the teacher have a priori different policy spaces. The learner estimates $p(b|a, c)$. By imitation, the learner can take advantage of the demonstrations (c, ζ, a_d) of the teacher to improve its estimation $p(b|a, c)$.

space with parameters $c \in \mathcal{C}$, and the outcome space with parameters $a \in \mathcal{A}$. A policy π_b is described by motor primitives parameterised by $b \in \mathcal{B}$. For a context parameter $c \in \mathcal{C}$, the probability of that the policy parameter b produces the outcome of parameter a is $\tilde{p}(a|b, c)$, where the probability density \tilde{p} represents the physics of the environment which the agent estimates. The association (c, b, a) corresponds to a learning exemplar that will be memorised.

The agent focuses on learning the inverse model and builds its current estimate $p(b_e|a, c)$. It tries to find at least one adequate policy parameter b to complete every goal/outcome $a \in \mathcal{A}$ from contexts c . We note that the inverse of the model, might not be a function, for the forward model can be highly redundant. Let us define D a distance measure on $\mathcal{A} \times \mathcal{A}$. The performance of a policy π_b at producing the outcome a from context c is measured by the cost function $J(a, b, c)$ defined as a mean distance between a and the outcome π_b :

$$\begin{aligned} J: \mathcal{A} \times \mathcal{B} \times \mathcal{C} &\rightarrow \mathbb{R} \\ (a, b, c) &\mapsto J(a, b, c) = \int_{\alpha} D(a, \alpha) p(\alpha|b, c) d\alpha \end{aligned} \quad (3.1)$$

Learning an inverse model means that the agent endeavours to minimise :

$$I = \int_{a \in \mathcal{A}, c \in \mathcal{C}} P(a, c) \min_b (J(a, b, c)) da dc \quad (3.2)$$

where $P(a, c)$ is a probability density distribution over $\mathcal{A} \times \mathcal{C}$. A priori unknown to the learner, $P(a, c)$ can describe the probability of c occurring or the reachable space for a or a region of interest.

Note that in this section, we have described our method only assuming a motor learning problem, and without specifying a particular choice of policy representation, learning algorithm \mathcal{L} or outcome space

properties. These designs can indeed be decided according to the application at hand.

Globally, the learner tries to learn to reach all reachable goals/outcomes a , and to generalise on the whole outcome space. The learner does not initially know which outcomes can be produced or not, so it at the same time learns its own limits of reachability.

For the strategic learner in this chapter, \mathcal{X} is a set of 2 predefined modes described in the previous section (section 3.1.1):

- imitation of demonstrations (χ_{dem}) of both policy and outcome of the single teacher.
- autonomous exploration with intrinsic motivation (χ_{auto}) based on SAGG-RIAC algorithm.

Nevertheless, in this chapter the learner does choose when to interact with the teacher. It does not decide on its mode, and the changes of between χ_{dem} and χ_{auto} are predefined as explained in section 3.1.1.1. The automatic decision on the sampling mode is studied in chapter 4

This problem statement enables a description of an active learning algorithm merging intrinsic motivation with social guidance with teacher’s demonstrations. We thus design the Socially Guided Intrinsic Motivation by Demonstration (**SGIM-D**) algorithm which alternates between two modes.

The active choice of strategy of SGIM-D resides in its focus function χ which it builds based on its experience, as we explain hereafter.

3.1.3 SGIM-D OVERVIEW

SGIM-D improves its estimation $p(b|a, c)$ to minimise I both by self-exploring the policy and outcome space and by imitating demonstrations (c, ζ, a_d) .

With the self-exploration mode, it actively self-generates goals $a_g \in A$ by stochastically choosing the goals for which its empirical evaluation of learning progress is maximal. For each a_g , the robot explores through goal-directed optimisation which policy π_b can induce the given goal a_g in context c . The exploration of the policy parameter space provides data to improve its estimation of the inverse model $p(b|a, c)$, that it can use later on to reach other goals. This autonomous exploration mode is only interrupted when the teacher gives a demonstration $[c_d, \zeta_d, a_d]$, when it switches to the social guidance mode.

With the social guidance mode, our SGIM-D learner mimics the demonstrated policy for a short while, and memorises the demonstrated outcome/goal as interesting, before resuming its autonomous exploration. It then generates a new goal, taking into account all its history, autonomous and social exploration phases alike. It chooses a goal with the highest interest level, which is defined as the competence progress.

The SGIM-D learner would thus try to explore goals where it makes progress the fastest. For each goal that it deems interesting, it would try different policies to reach it, using the policy repertoire of its past autonomous exploration or the policies suggested by the teacher’s demonstrations. Once its competence for these easy goals is high, it no longer makes progress, and as its interest level for them drops, it progressively aims at more difficult goals and expands its search in the outcome space. The human teacher boosts its learning by indicating policies to perform, so that its competence level increases, but also by indicating interesting goals/outcomes to emulate, to orient its search in the outcome space.

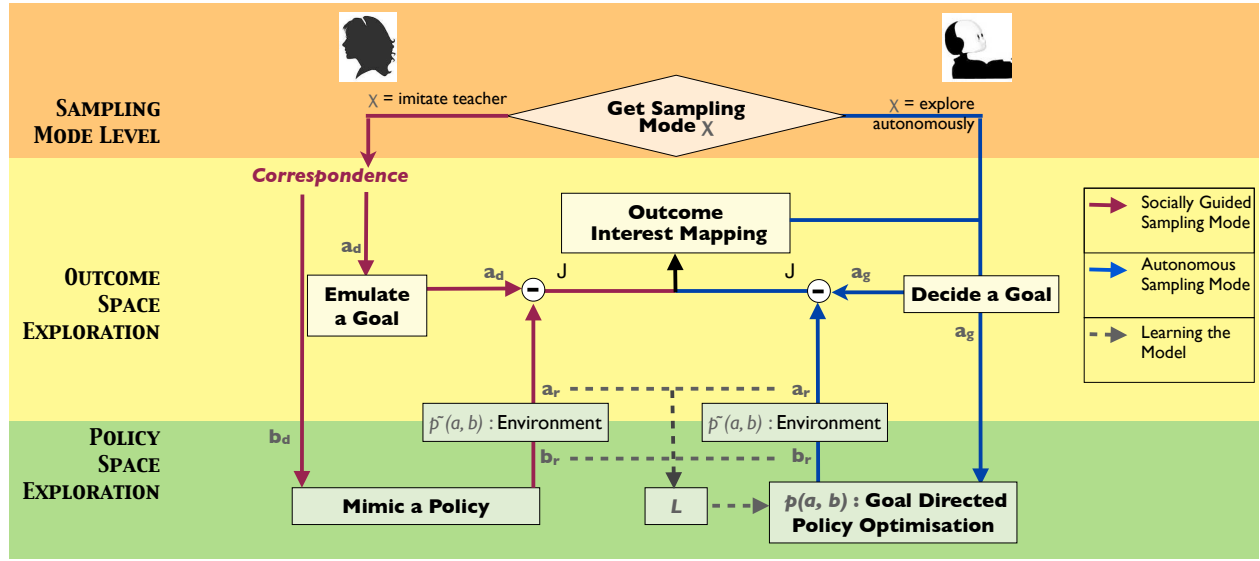


Figure 3.1.2: Time flow chart of SGIM-D into 3 layers that pertain to the human-machine interface (mode level), the outcome space exploration and the policy space exploration respectively. The architecture combines sub-modules for intrinsically motivated learning and socially guided learning in both the policy and outcomes spaces.

Algorithm 3.1.1 SGIM-D

- 1: **Initialization:** $\mathcal{R} \leftarrow$ singleton $C \times A$
 - 2: **Initialization:** Initialise p_1 to a homogeneous probability density
 - 3: **Initialization:** $\mathcal{H} \leftarrow$ empty episodic memory (collection of episodes of reached outcome a with policy parameter b in context c , (c, b, a))
 - 4: **loop**
 - 5: Retrieve sampling mode χ
 - 6: **if** $\chi = \textit{Imitation}$ **then**
 - 7: **Social Learning Mode**
 - 8: **repeat**
 - 9: $(a_d, b_d, c_d) \leftarrow$ Correspondence of the teacher's demonstration
 - 10: Emulate Goal: $a_g \leftarrow a_d$
 - 11: $\mathcal{D} \leftarrow$ Mimic Policy (b_d, c)
 - 12: $p \leftarrow \mathcal{L}(p, \mathcal{D})$
 - 13: **until** End of social interaction
 - 14: **else**
 - 15: **Intrinsic Motivation Mode**
 - 16: Measure current context c
 - 17: $a_g \leftarrow$ Decide a goal with a probability proportional to its associated expected competence progress
 - 18: **repeat**
 - 19: $\mathcal{D} \leftarrow$ Goal-Directed Policy Optimisation(c, a_g, p, \mathcal{H})
 - 20: $p \leftarrow \mathcal{L}(p, \mathcal{D})$
 - 21: **until** Terminate reaching of a_g
 - 22: **end if**
 - 23: Append \mathcal{D} to \mathcal{H}
 - 24: $\mathcal{R} \leftarrow$ Update Goal Interest Mapping($\mathcal{R}, \mathcal{H}, c, a_g$)
 - 25: **end loop**
-

3.1.4 SGIM-D ARCHITECTURE

Socially Guided Intrinsic Motivation by Demonstration (**SGIM-D**) is an algorithm that merges programming by demonstration as social interaction mode with the SAGG-RIAC algorithm as intrinsic motivation mode, for the learning of local inverse and forward models in complex, high-dimensional and continuous spaces. **Its architecture is separated into three layers** where sub-modules of each sampling mode interact (**Figure 4.3.3** and **Algorithm 3.1.1**) :

- An interface with the teacher, which manages the "physical" interaction with the teacher and determines the mode of the agent. It detects that the teacher performs a demonstration and translates it into parameters for the robot. The implementation interface is specific to each robot and experimental setting, and will be detailed specifically for the experimental setup in section 3.2.1.
- A higher level of active learning, the *Outcome Space Exploration level* which drives the exploration of the outcome space. It sets goals a_g depending on their interest levels that is based on the competence of previous goals, retrieves from the teacher information about goals, and maps A in terms of interest level. It learns at a longer time scale. Its structure is detailed in subsection 3.1.4.2.
- A lower level of active learning, the *Policy Space Exploration level* that explores the policy parameter space B to improve its estimation of $p(b|a_g, c)$ and the more general inverse model $p(b|a, c)$. While interacting with the teacher, it would mimic his policies ζ_d , whereas during self-exploration, it would attempt to reach the goals a_g set by the Outcome Space Exploration level. It learns at a shorter time scale. Its structure is shortly described in subsection 3.1.4.1 and detailed for our implementation in section 4.2.3.1.

3.1.4.1 LOWER LEVEL : POLICY SPACE EXPLORATION

The *Policy Space Exploration* searches the policy parameters space B how to reach the goal a_g set by the higher level or mimics the demonstrated policy ζ_d , and returns to the Outcome Space Exploration level the measure of competence at reaching a_g .

The implementation details will depend on the experimental setup, but mainly, the Policy Space Exploration level contains 2 functions:

- The *Mimic Policy* function takes as input a policy parameter b_d demonstrated by the teacher and tries to repeat it. This function can be changed to match other social interaction modes. An implementation is described for our experimental setting in subsection 3.2.2.
- The *Goal-Directed Policy Optimisation* function searches for policy parameters b that guide the system toward the goal a_g in the given context c by 1) building local inverse $p(b|a, c)$ model during exploration that can be re-used for later goals and 2) selecting new policies depending on interestingness measures of policies with respect to the current goal to get a better estimate of $b \mapsto J(a_g, b, c)$. Mainly, it can be implemented by classical autonomous learning methods mentioned earlier which learn for a single goal only such as the Cost-regularized Kernel Regression (Kober et al., 2010), Episodic Natural Policy Gradient algorithm (Peters and Schaal, 2008), or PI2 (Stulp and Schaal, 2011). An example is presented for our experimental setting in subsection 3.2.4. This function minimises $b \mapsto J(a_g, b, c)$.

3.1.4.2 HIGHER LEVEL : ACTIVE GOAL BABBLING FOR OUTCOME SPACE EXPLORATION

The *Outcome Space Exploration* relies on feedback from the *Policy Space Exploration* level to decide which goal $a_g \in A$ is interesting to focus on. It explores A using teacher’s demonstrations of outcomes (*Emulate a Goal*) and self-determines a goal (*Decide a Goal*) using competence measures, more precisely competence improvement which it maps on $C \times A$ (*Goal Interest Mapping*).

(I) GOAL INTEREST MAPPING FUNCTION To determine which goals it should attempt in order to better generalise for the whole outcome space, the agent **self-structures the outcome space in to different regions depending on its level of interest**. It assigns a competence γ_{c,a_g} to each goal a_g explored in context c , as a measure of how close the learner can reach a_g :

$$\gamma_{c,a_g} = \min_{(c,b,a_g) \in \mathcal{H}} J(a_g, b, c) \tag{3.3}$$

where \mathcal{H} is the list of all the collected data (c, b, a) .

Along with the estimated inverse model $p(b|a, c)$, SGIM-D estimates at the same time the interest mapping function over $C \times A$ (**Algorithm 3.1.1**, l. 24)) to build its focus function ψ . In our approach, while $p(b|a, c)$ is estimated as a complex function, we model the interest mapping as a piecewise constant function.

Let us consider a partition $\biguplus_i R_i = C \times A$. Each R_i contains attempted goals given a context $\{(c_{t_1}, a_{t_1}), (c_{t_2}, a_{t_2}), \dots (c_{t_k}, a_{t_k})\}_{R_i}$ of competences $\{\gamma_{t_1}, \gamma_{t_2}, \dots, \gamma_{t_k}\}_{R_i}$, indexed by their relative time order of experimentation $t_1 < t_2 < \dots < t_k$ inside subspace R_i .

An estimation of interest is computed for each region R_i as *the local competence progress, over a sliding time window of the ζ more recent goals attempted inside R_i* :

$$interest_i = \frac{\left| \left(\sum_{j=|R_i|-\zeta}^{|R_i|-\frac{\zeta}{2}} \gamma_j \right) - \left(\sum_{j=|R_i|-\frac{\zeta}{2}}^{|R_i|} \gamma_j \right) \right|}{\zeta} \tag{3.4}$$

By using a derivative, the interest considers the *variation of competences*, and by using an absolute value, it considers cases of *increasing and decreasing competences*. In SGIM-D, we will use the term *competence progress* with its general meaning to denote this increase and decrease of competences. An increasing competence signifies that the expected competence gain in R_i is important. Therefore, selecting new goals in regions of high competence progress could bring both a high information gain for the learned model, and also drive the reaching of previously unachieved goals. Depending on the starting position and potential evolution of the environment, a decrease of competences inside already well-reached regions can arise. In this case, the system should be able to focus again in these regions to attempt to re-establish a high level of competence inside. This explains the usefulness of considering the absolute value of the competence progress as shown in equation 4.3.

Based on this definition of interest, the module builds an interest level mapping, at each new outcome a_g chosen as goal or produced by autonomous exploration or at each goal a_d observed in social guidance. It

Algorithm 3.1.2 $[\mathcal{R}] = \text{UpdateRegions}(\mathcal{R}, (c, a), \gamma)$]

- 1: **input:** \mathcal{R} : set of regions and corresponding *interest*.
- 2: **input:** (c, a) : context and effect of the learning exemplar.
- 3: **input:** γ : competence at reaching a in context c .
- 4: **parameter:** g_{Max} : the maximal number of elements inside a region.
- 5: **parameter:** ζ : a time window used to compute the interest.
- 6: Find the region $R_n \in \mathcal{R}$ such that $(c, a) \in R_n$.
- 7: Add γ to R_n .
- 8: Compute the new value of $interest_n$ of R_n according to each $(c_i, a_i) \in R_n$ of competence γ_i such that:

$$interest_n = \frac{\left| \left(\sum_{i=|R_n|-\zeta}^{|R_n|-\frac{\zeta}{2}} \gamma_i \right) - \left(\sum_{i=|R_n|-\frac{\zeta}{2}}^{|R_n|} \gamma_i \right) \right|}{\zeta}$$

- 10: **if** $|R_n| > g_{max}$ **then**
 - 11: Split R_n .
 - 12: **end if**
 - 13: **return** \mathcal{R}
-

partitions $C \times A$ into subspaces, so as to maximally discriminate areas according to their levels of interest, as described in (Baranes and Oudeyer, 2013). We use a recursive split of the space, each split occurring once a maximal number of goals have been attempted inside. Each split maximises the difference of the *interest* measure in the two resulting subspaces, and easily separates areas of different interest, and thus, of different reaching difficulty (cf. **Algorithm 3.1.2**).

The partition of $C \times A$ is done recursively and so as to maximally discriminate areas according to their levels of interest. A split is triggered once a number of outcomes g_{max} has been attempted inside a region R_n with the same strategy σ . The split separates areas of different interest levels and different reaching difficulties. The split of a region R_n into R_{n+1} and R_{n+2} is done by selecting among m randomly generated splits, a split dimension j and then a position v_j such that:

- All the $\tau \in R_{n+1}$ have a j th component smaller than v_j ;
- All the $\tau \in R_{n+2}$ have a j th component higher than v_j ;
- It maximises the quantity $Qual(j, v_j) = |R_{n+1}| \cdot |R_{n+2}| |interest_{R_{n+1}(\sigma)} - interest_{R_{n+2}(\sigma)}|$, where $|R_i|$ is the size of the region R_i ;

(II) **DECIDE A GOAL FUNCTION** The *Decide a Goal* function uses the interest level mapping to select the next goal to perform (**Algorithm 3.1.1**, l. 17)). **Goals are chosen stochastically according to their interest level**, with either of the following modes:

- **Mode(1)**: A chosen random goal inside a region which is selected with a probability proportional to its interest value. The probability of selecting the region R_n that contains the current context c is:

$$P_n = \frac{\text{interest}_n - \mathbf{min}(\text{interest}_i)}{\sum_{i=1}^{|R_n|} \text{interest}_i - \mathbf{min}(\text{interest}_i)} \quad (3.5)$$

- **Mode(2)**: A selected random goal inside the whole space A .
- **Mode(3)**: A first selected region according to the interest value (like in $mode(1)$) and then a generated new goal close to the already experimented one which received the lowest competence estimation $\min_{R_n}(\gamma_{t_i})$.

(III) **EMULATE A GOAL FUNCTION** At each demonstration, the learner observes not only the policy performed, but also its outcome a_d . It henceforward considers this outcome as a potential goal, and assigns an interest level according to its own policy repertoire and model it has built (**Algorithm 3.1.1**, l. 10)).

The above description is detailed for SGIM-D’s choice of imitating teachers’ low-level demonstrations of outcomes and policies. Such a structure would remain suitable for other choices of social interaction modes, and we only have to change the content of the *Emulate a Goal* function, and change the *Mimic a Policy* function to match the chosen mode.

In the following section, we illustrate the principle of SGIM-D through a proof-of-concept experiment, where our robot learns how to fish.

3.2 THE FISHING ROD EXPERIMENT

In this section we describe our experimental setup with the environment’s description, and then detail how SGIM-D functions adapt for this specific setup.

In this illustration experiment, we consider a simulated 6 degrees-of-freedom robotic arm holding a fishing rod (**Figure 3.2.1**). The aim is that **it learns how to reach any point on the surface of the water with the float at the tip of the fishing line**. This is an inverse model in a continuous and unbounded environment of a complex system that can hardly be described by physical equations.

In our experiment, the context space C describes the initial actuator/joint positions and state of the fishing rod. $A = [-1, 1]^2$ is a 2-D space that describes the position of the float when it reaches the water. For each position $a \in A$, it has to learn a new goal : with which movement π_b he can place the float closest to a . The robot base is positioned at (0,0) and it always starts with the same configuration c_{org} .

3.2.1 MOTOR PRIMITIVES AND CORRESPONDENCE MAPPING

Variable b describes the parameters of the motor primitives of the joints, defining for each joint 4 scalar parameters that represent the joint positions at $t = 0, t = \frac{\delta}{3}, t = \frac{2\delta}{3}$ and $t = \delta$. These 4 parameters u_1, u_2, u_3, u_4 generate a trajectory for the joint by Gaussian distance weighting:

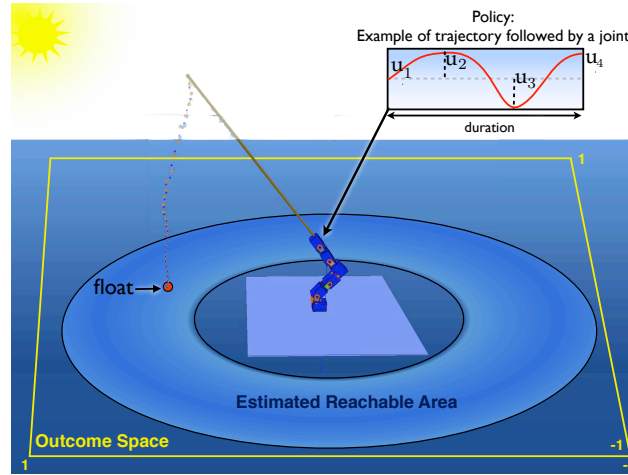


Figure 3.2.1: Experimental setup with a robot arm holding a fishing can with a flexible wire (simulated by 30 free revolute joints). The robot can produce a movement of its 6 DOF arm by setting the real number values b of its 25 dimensional motor primitive. Then, it can observe the effect/outcome of such a movement, by observing where the float has arrived in the goal/outcome space, i.e. on the surface of the water which is a 2D space. Using SGIM-D, which combines intrinsically motivated active learning and human demonstration, the robot has to learn the complex inverse model mapping all goals/outcomes to the adequate 25 dimensional parameters of motor movement.

$$u(\mathbf{t}) = \sum_{i=0}^4 \frac{w_i(\mathbf{t})u_i}{\sum_{j=0}^4 w_j(\mathbf{t})} \text{ with } w_i(\mathbf{t}) = e^{\sigma * |\mathbf{t} - \frac{i\delta}{5}|^2}, \sigma > 0 \quad (3.6)$$

Each of the 6 joints' trajectories is determined by 4 parameters. Another parameter sets δ . Therefore a policy is represented by $6 \times 4 + 1 = 25$ parameters: $b = (b^1, b^2, \dots, b^{25})$. $B = [0, 1]^{25}$. This choice of taking only 4 samples of the movement trajectory is arbitrarily, and other parametrisations have been also used in other studies (Nguyen and Oudeyer, 2012e,c).

Because our experiment uses for each trial the same context c_{org} , our system memorises after executing every policy parameter b , simply the context-free association (b, a) .

Upon observation of a demonstration (ζ_{Hd}, a_{Hd}) , the *Correspondence* function first computes the parameters b_d that enable him to reproduce the teacher's policy ζ_d closest (**Algorithm 3.1.1**, l. 9).

From ζ_{Hd} , it can extract for each joint a trajectory $u_{Hd}(t)$ and the duration of the trajectory δ . To map a given joint trajectory $u_{Hd}(t)$ into our robot's parameterised dynamic motor primitive, we need to determine the 4 parameters u_1, u_2, u_3, u_4 so as to minimise the error (Figure 3.2.2):

$$d = \left\| \left\| u_{Hd}(\mathbf{t}) - \sum_{i=0}^4 \frac{w_i(\mathbf{t})u_i}{\sum_{j=0}^4 w_j(\mathbf{t})} \right\| \right\|_2 \quad (3.7)$$

b_d is thus the set of parameters u_1, u_2, u_3, u_4 for each of the 6 joints, and δ , which minimise d by the

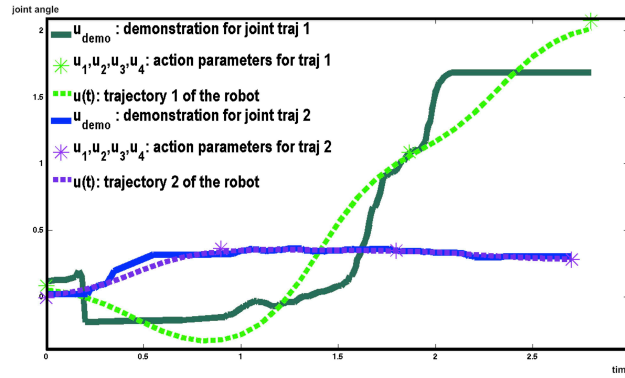


Figure 3.2.2: Mapping of the demonstrations given by the human teacher by the robot. Horizontal axis: time, vertical axis: joint angle (best seen in colors). Are plotted for 2 different joint trajectories of a human demonstrator, the demonstrated trajectory, and the corresponding movement parameters and trajectory mapped by the robot. For a demonstrated trajectory u_{Hd} , parameters u_1, u_2, u_3, u_4 minimise eq. 3.7. Then the parameters u_1, u_2, u_3, u_4 generate the trajectory executed by the robot $u(t)$ according to **Equation 4.1**. For joint trajectory 1, the mapping has a high error value, while for joint trajectory 2, the mapping has a low error value.

Algorithm 3.2.1 [\mathcal{D}] = Mimic Policy(b_d, c)

- 1: **Input:** nIm : duration of the imitation phase;
 - 2: **Input:** thresholds: ϵ_{max} ;
 - 3: **Initialise:** $\mathcal{D} \leftarrow$ empty list of (a, b, c)
 - 4: **for** $nbMimic$ times **do**
 - 5: $b_{rand} \leftarrow$ random vector such that $|b_{rand}| < \epsilon_{max}$
 - 6: $b \leftarrow b_d + b_{rand}$
 - 7: $(a, b, c) \leftarrow$ Execute (c, b)
 - 8: Add (a, b, c) to \mathcal{D}
 - 9: **end for**
 - 10: **return** \mathcal{D}
-

trust-region-reflective algorithm described in (Coleman and Li, 1994, 1996).

3.2.2 MIMIC A POLICY

The *Mimic a Policy* function (cf. **Algorithm 3.2.1** and Figure 4.3.3) tries to mimic a demonstration (ζ_d, a_d) by **repeating the observed policy** with policy parameters $b_{Mimic} = b_d + b_{rand}$ with a random movement parameter variation $|b_{rand}| < \epsilon$ and π_{b_d} is the closest policy to reproduce ζ_d (**Algorithm 3.2.1**, l. 5).

After a fixed number of executions, SGIM-D computes its competence at reaching the goal indicated by the teacher a_d (cf. Equation 3.3). This function thus makes an estimate of $J(a_d, \tilde{p}(b|a_d))$. Then, it shifts back to the autonomous exploration mode. The measure of competence returned is defined hereafter.

Algorithm 3.2.2 $[(a, b, c)] = \text{Execute}(c, b)$

Set context c and perform policy parameters b

- 1: Initialise robot at c
- 2: Perform policy of parameters b .
- 3: Measure outcome a in the outcome space
- 4: $\text{flagInteraction} \leftarrow$ check for a teacher’s demonstration
- 5: **return** (a, b, c)

3.2.3 PERFORMANCE MEASURE

We define D as the euclidian distance $D(a_g, a)$ between two positions on the surface of the water, and normalised by the distance between the original position a_{org} and the goal: $D(a_{org}, a_g)$. This allows, for instance, to give the same competence level when considering a goal at 1km from the origin position that the robot approaches at 0.1km, and a goal at 100m that the robot approaches at 10m:

$$D(a_g, a) = \begin{cases} -1 & \text{if } \frac{D(a, a_g)}{D(a_g, a_{org})} > 1 \\ -\frac{D(a, a_g)}{D(a_g, a_{org})} & \text{otherwise} \end{cases} \quad (3.8)$$

Here, our direct model $p(a|b)$ only considers the 25 parameters $b = (b^1, b^2, \dots, b^{25})$ as inputs of the system, and a position in $a = (a^1, a^2)$ as output. We wish to build the estimate inverse model $p(b|a)$ by using the following optimisation mechanism for goal-directed learning (\mathcal{L}) and exploration, which can be divided into two different regimes.

3.2.4 GOAL-DIRECTED POLICY OPTIMISATION

The *Goal-Directed Policy Optimisation* function (cf. **Algorithm 3.2.4** and Figure 4.3.3) **learns to reach the goal a_g generated by the Outcome Space Exploration level**. This function can be implemented by any single outcome learning algorithm. Classical reinforcement methods such as natural actor-critic architectures (Peters and Schaal, 2008), path integral approaches (Theodorou et al., 2010) or advanced Black Box optimisation techniques (Stulp and Sigaud, 2012), or evolutionary algorithms such as CMA-ES (Hansen and Ostermeier, 2001) could be used. For the sake of proving that the efficiency of our SGIM-D algorithm relies on its general structure, and not so much on its per-goal learning algorithm, we choose a learning algorithm based on nearest neighbours, and a simulated annealing exploration method that builds memory-based local direct and inverse models. Simulated annealing is a generic probabilistic metaheuristic for the global optimisation problem of locating a good approximation to the global optimum by slowly decreasing the probability of accepting worse solutions as it explores the solution space. More precisely, SGIM-D would tend to do global exploration of the policy space if the current goal has never been approached, but if it has reached outcomes near the current goal, SGIM-D would do more local optimisation. SGIM-D uses locally weighted regression in order to infer the motor policy parameters corresponding to a given novel parametrized task, and based on the previously learnt correspondences between policy and task parameters. Local optimisation is here implemented with the Nelder-Mead simplex algorithm that is

Algorithm 3.2.4 $[\mathcal{D}] = \text{Goal-Directed Policy Optimization}(c, a_g, p(b|a), \mathcal{H})$.

Search for policies to reach a_g in context c while building model

```

1: Initialise:  $\mathcal{D} \leftarrow$  empty list of  $(c, b, a)$ 
2:  $(a_{close}, b_{close}, c_{close}) \leftarrow$  Search in  $\mathcal{H}$  for the  $a_{close}$  closest to  $a_g$ 
3:  $mLow \leftarrow$  mode global-exploration or local-optimization with probability  $\propto J(a_g, b_{close}, c_{close})$ 
4: if  $mLow = \textit{global-exploration}$  then
5:   Action parameter  $a \leftarrow$  random movement parameters
6:    $(c, b, a) \leftarrow \text{Execute}(c, b)$ 
7:   Add  $(a, b, c)$  to  $\mathcal{D}$ 
8: else if  $mLow = \textit{local-optimization}$  then
9:    $\mathcal{D} \leftarrow \text{LocalOptimization}(c, a_g, p(b|a), \mathcal{H})$ 
10: end if
11: return  $\mathcal{D}$ 

```

summarised in **Algorithm 3.2.3** and detailed in (Lagarias et al., 1998) It also sometimes does global random exploration to avoid local minima. It builds memory-based local direct and inverse models, using locally weighted learning with a gaussian kernel such that presented in (Atkeson et al., 1997).

To decide which mode is triggered given a goal a_g , we examine the memory of the system, and consider that the closest one has been able to reach a_g , the more the system should focus on local optimisation. On the contrary, if during the system’s history, it has never reached a point close enough to the goal a_g , it should prefer global exploration.

The system continuously estimates the distance between the goal a_g and the closest already reached position a_c : $D(a_c, a_g)$. The system has a probability proportional to $D(a_c, a_g)$ of being in the Global Exploration regime, and the complementary probability of being in the Local Optimisation regime.

3.2.4.1 GLOBAL EXPLORATION REGIME

In the global exploration regime, the system just picks **random policy parameters** $b \in T$ to explore the **policy space** (**Algorithm 3.2.4**, l. 5).

3.2.4.2 LOCAL OPTIMISATION REGIME

The local optimisation regime (**Algorithm 3.2.4**, l. 9) represents the learning algorithm \mathcal{L} . It uses the memory data to infer locally inverse models $p(b|a)$. Given the high redundancy of the problem, we choose a local approach and **explore around the potentially more reliable data determined as roughly the k-nearest neighbours** (**Algorithm 3.2.5**). More precisely, (**Algorithm 3.2.5**, l. 3), we first compute the set H of the h_{max} nearest neighbours of a_g and their corresponding movement parameters using an ANN method (Muja and Lowe, 2009), which is based on a tree split using the k-means process :

$$H = \{(a, b)_1, (a, b)_2, \dots, (a, b)_{h_{max}}\} \subset (A \times B)^{h_{max}} \quad (3.9)$$

Algorithm 3.2.3 LocalOptimization algorithm using Nelder-Mead simplex algorithm

```

1: Let  $x(i)$  denote the list of points in the current simplex,  $i = 1, \dots, n+1$ .
2: Initialise the points of the the simplex with the data given by LocalData
3: repeat
4:   Order the points in the simplex from lowest function value  $f(x(1))$  to highest  $f(x(n+1))$ . At each step
   in the iteration, the algorithm discards the current worst point  $x(n+1)$ , and accepts another point
   into the simplex. [Or, in the case of step 39 below, it changes all  $n$  points with values above  $f(x(1))$ ].
5:   Generate the reflected point  $r = 2m - x(n+1)$ , where  $m = \sum x(i)/n$ ,  $i = 1 \dots n$ , and calculate  $f(r)$ .
6:   if  $f(x(1)) \leq f(r) < f(x(n))$  then
7:     accept  $r$  and terminate this iteration. Reflect
8:   end if
9:   if  $f(r) < f(x(1))$  then
10:    calculate the expansion point  $s: s = m + 2(m - x(n+1))$ 
11:    calculate  $f(s)$ 
12:    if  $f(s) < f(r)$  then
13:      accept  $s$  and terminate the iteration. Expand
14:    else
15:      accept  $r$  and terminate the iteration. Reflect
16:    end if
17:  end if
18:  if  $f(r) \geq f(x(n))$  then
19:    perform a contraction between  $m$  and the better of  $x(n+1)$  and  $r$ :
20:    if  $f(r) < f(x(n+1))$  (i.e.,  $r$  is better than  $x(n+1)$ ) then
21:      calculate  $c = m + (r - m)/2$  and calculate  $f(c)$ 
22:    end if
23:    if  $f(c) < f(r)$  then
24:      accept  $c$  and terminate the iteration. Contract outside
25:    else
26:      continue with Step 39 (Shrink).
27:    end if
28:    if  $f(r) \geq f(x(n+1))$  then
29:      calculate  $cc = m + \frac{(x(n+1) - m)}{2}$ 
30:      calculate  $f(cc)$ 
31:      if  $f(cc) < f(x(n+1))$  then
32:        accept  $cc$ 
33:        terminate the iteration. Contract inside
34:      else
35:        continue with Step 39 (Shrink).
36:      end if
37:    end if
38:  end if
39:  Calculate the  $n$  points  $v(i) = x(1) + (x(i) - x(1))/2$ 
40:  calculate  $f(v(i))$ ,  $i = 2, \dots, n+1$ .
41:  The simplex at the next iteration is  $x(1), v(2), \dots, v(n+1)$ . Shrink
42: until minimisation

```

Algorithm 3.2.5 $[K_{l_{best}}] = \text{LocalData}(a_g, \mathcal{H})$.

Retrieve from the memory \mathcal{H} experiences in the locality of a_g

- 1: **input:** thresholds distM , distN
- 2: Get from the memory the nearest neighbours of a_g :
- 3: $H \leftarrow \{(a_h, b_h, c_h) \in \mathcal{H} \mid \left. \begin{array}{l} J(a_g, b_h, c_h) < \text{distM} \\ J(a_g, b_h, c_h) < J(a_h, b, c), \forall (a, b, c) \in \mathcal{H} - H \end{array} \right\}$
- 4: $\forall (a_h, b_h, c_h) \in L, K_l \leftarrow \{(a, b, c) \in \mathcal{H} \mid |b - b_h| < \text{distN}\}$
- 5: Select the best locality : $l_{best} \leftarrow \text{argmin}(\text{LocalQuality}(K_l, a_g))$
- 6: **return** $K_{l_{best}}$

Then, for each element $(a, b)_h \in H$, we compute its reliability. Let us consider the set K_h which contains the nearest neighbours of b_h within distN of b_h in the memory set \mathcal{H} with respect to norm $\|\cdot\|_2$ (**Algorithm 3.2.5**, l. 4) :

$$K_h = \{(a, b)_1, (a, b)_2, \dots, (a, b)_{k_{max}}\} \quad (3.10)$$

As the reliability of the local model depends both on the knowledge of the locality and the reproducibility of the movement due to non-linear noise that produces small variations in a of magnitude depending on b (Figure 3.2.3), we define for each element $(a, b)_h \in H$, its reliability as $\text{dist}(a_h, a_g) + \alpha \times \text{var}_h$, where var_l is the variance of the set K_h , and α is a constant set to 0.5 in our experiment. We choose the smallest value, as the most reliable set $(a, b)_{best}$ (**Algorithm 3.2.5**, l. 5).

In the locality of the set $(a, b)_{best}$, we interpolate using the k_{max} elements of K_{best} to compute the policy corresponding to a_g : $b_g = \sum_{k=1}^{k_{max}} \beta_k b_k$ where $\beta_k \propto \text{Gaussian}(\text{dist}(a_k, a_g))$ is a normalised Gaussian of the euclidian distance between a_k and the goal a_g .

We execute policy of parameter a_g (**Algorithm 3.2.5**, l. 6) and continue with the Nelder-Mead simplex algorithm (Lagarias et al., 1998), to minimise the distance of the outcome a_2 to the goal a_g . This algorithm uses a simplex of 26 points for 25-dimensional vectors b . It first makes a simplex around the initial guess b_g with the $b_k, k = 1, \dots, k_{max}$. It then updates the simplex with points around the locality until the distance to minimise falls below a threshold.

3.2.5 STOCHASTIC ENVIRONMENT

All the experimental setup has been designed for a 6 DOF robot arm in the real world. Nevertheless, to be able to collect statistics through numerous experiments, we built a model of our 6 DOF arm on V-REP physical simulator (Freese), which uses a ODE physics engine that updates every 50 ms.

Due to stochasticity of the simulated experimental setup, repetitions of the same movement do not lead to the same exact outcome. Moreover, **the stochasticity does not follow a uniform distribution rule** and can not be modelled by a simple Gaussian. The standard deviation varies along the different dimensions and depends on the dynamic properties of the movement performed (**Figure 3.2.3**). The mean variance

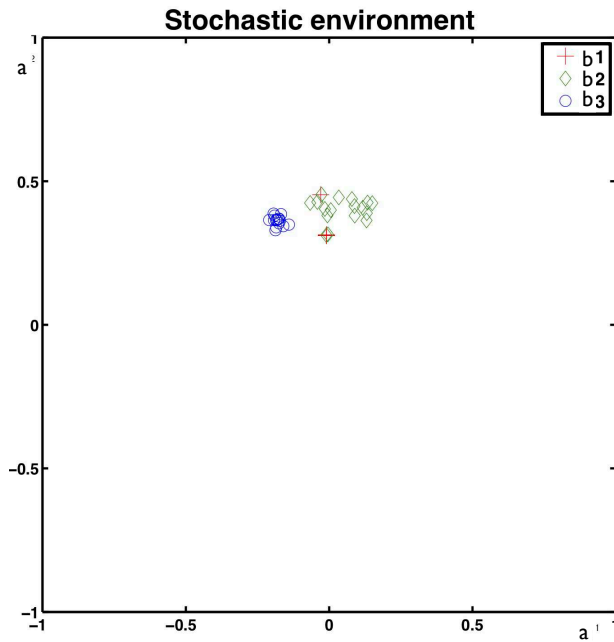


Figure 3.2.3: Outcomes for 3 different policy parameters over 20 repetitions of the same movement, represented in the 2-D space A . Standard deviations are for each policy parameters, respectively (0.005, 0.033) for b_1 , (0.0716, 0.041) for b_2 , and (0.016, 0.016) for b_3 (best seen in colors).

of the control system of the robot is estimated to 0.073 for measures of 10 attempts of 20 random policy parameters, while the reachable area spans between -1 and 1 for each dimension of A .

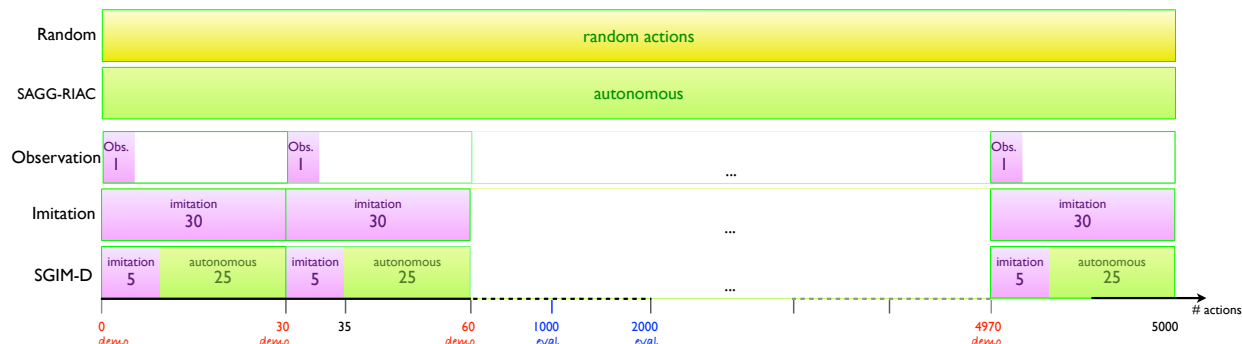
This fishing experiment focuses on the learning of inverse models in a continuous space, and deals with high-dimensional and highly redundant models. Our setup is all the more interesting as a real-world fishing rod’s and flexible line’s dynamics would be difficult to model. The model of a fishing rod in the simulator might be mathematically computed. However, To represent the complexity of the fishing line manipulated by the robot arm, we modelled it as a set of 30 segments and 30 revolute joints, which leads to complex movements hard to predict. Even though the direct mapping has been modelled by the simulator, **the inverse model, which is even more complicated due to redundancy and stochasticity, is yet to learn**. Besides, our fishing environment’s stochasticity distribution is hard to model. Thus learning directly the outcome of one’s policies is all the more advantageous.

The next section describes how we evaluate the SGIM-D algorithm using the fishing experimental setup.

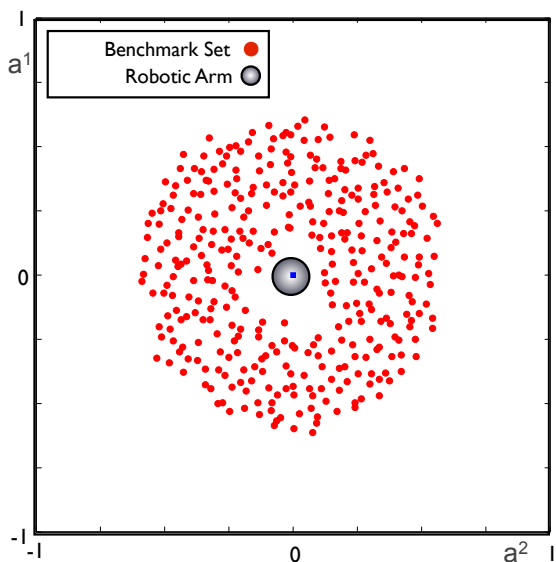
3.3 EXPERIMENTAL PROTOCOL

In this section, we detail the experiments we carry with our fishing robot setup to evaluate SGIM-D and how we provide our learner with demonstrations.

(a) Comparison of several learning algorithms



(b) Benchmark set



(c) Demonstration sets

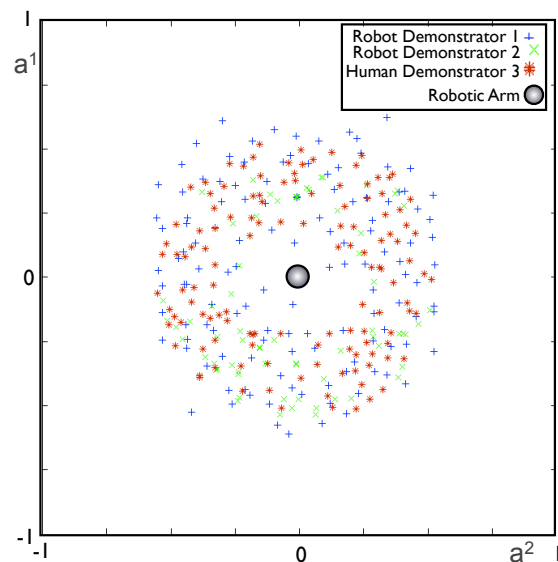


Figure 3.3.1: (best seen in colors) (a): The experiment compares the performance of several exploration algorithms: Random exploration of the policy space B , autonomous exploration SAGG-RIAC, Learning from Observation, Imitation learning and SGIM-D. The comparison is made through the same experimental duration (5000 policies performed by the robot), through the same teaching frequency (every 30 policies) and through regular evaluation (every 1000 policies). (b): Map in the 2D outcome space A of the benchmark points used to assess the performance of the robot: by measuring how close they can reach each of these points. (c): Maps in the 2D outcome space A of the teaching sets used in SGIM-D, by three demonstrators. Demonstrator 1 is a SAGG-RIAC learner, while demonstrator 2 is an optimised SAGG-RIAC learner, and demonstrator 3 is a human teacher.

3.3.1 COMPARISON OF LEARNING ALGORITHMS

To assess the efficiency of SGIM-D, we decide to compare the performance of several exploration algorithms (**Figure 3.3.1a**):

- Random exploration : throughout the experiment, the robot picks policies randomly in the policy parameter space B .
- SAGG-RIAC: throughout the experiment, the robot explores autonomously, without taking into account any demonstration by the teacher, and is driven by intrinsic motivation .
- Imitation learning: every time the robot sees a new demonstration b_d of the teacher, it repeats the policy while making small variations: $b_{Mimic} = b_d + b_{rand}$ with b_{rand} a random movement parameter variation, so that $|b_{rand}| < \epsilon$. It keeps on repeating this demonstration until it sees a new demonstration every N policies, and then starts imitating the new demonstration.
- Observation learning: the robot does not make any policy, but only watches the teacher’s demonstrations.
- SGIM-D: the robot’s mode is a mixture between Imitation learning and SAGG-RIAC. When the robot sees a new demonstration, it mimics the policy, but only for a short while. Then, it resumes its autonomous exploration, until it sees a new demonstration by the teacher. Its autonomous exploration phases take into account all its history from both the autonomous and imitation phases.

For each experiment, we let the robot perform 5000 policies in total, and evaluate its performance every 1000 policies, using the method described below.

3.3.2 EVALUATION

After several runs of Random explorations, SAGG-RIAC and SGIM-D, we determined the apparent reachable space basing on the set of all the reached points in the goal/outcome space, which makes up some 300.000 points. We then tiled the reachable space into small rectangles, and generated a point randomly in each tile. We thus obtained a set of 358 goal points in the outcome space, representative of the reachable space (**Figure 3.3.1b**). We will use these points to measure how close the system can get to each of these points with:

$$mean_{a_g \in BenchmarkSet}(D(a_g, a_r)) \tag{3.11}$$

where a_r is the outcome observed by the robot when attempting to produce outcome a_g .

3.3.3 DEMONSTRATIONS

For demonstrations, we used kinesthetics. **The human teacher physically moves the robot**, using both the physical robot and its model in the simulator. The model in the simulator is tele-operated by the teacher through the physical robot, as is shown in **Figure 3.3.2** and in http://youtu.be/L1_S-u00kD0. The human



Figure 3.3.2: Demonstration by kinesthetics. A human demonstrator manipulates a physical robot which is connected to a physical simulator.

subject is presented with a grid of points to reach on the surface of the water, and he has to manipulate the physical robot to place the simulator’s fishing rod nearest one of those point. After a habituation phase, we record the trajectories of each of the joints, and the position of the float when touching the surface of the water. We obtained a teaching set (**Figure 3.3.1c**) from an expert teacher of 127 samples.

In order to analyse the specific properties of human demonstrations compared to random demonstrations in the SGIM-D algorithm, we also prepared two other sets of demonstrations, evenly distributed in the reachable space, and taken from a pool of data from several runs of SAGG-RIAC, using the previous SAGG-RIAC learners as teachers.

Thus we have 5 demonstration sets (Figure 3.3.1):

- demonstrator 1: SAGG-RIAC learners who now teach in return our SGIM-D robot. They choose demonstrations randomly among their memory exemplars (b, a) . It would illustrate the case of a naive teacher in a context of robot to robot teaching.
- demonstrator 2: SAGG-RIAC learners who now teach in return our SGIM-D, but carefully choose among their memory exemplars (b, a) that are most reliable. The evenly distributed demonstrations minimise the variance of a over several re-executions of the same policy π_b . It would illustrate the case of a more evolute teacher in a scenario of robot to robot teaching. We built it taking inspiration from our observations of the demonstrator 3, to obtain a case halfway between the two other demonstrators in order to analyse the specific properties of human demonstrations.
- demonstrator 3: a human teacher who tries to give demonstrations (ζ_d, a_d) evenly distributed in the reachable space of A . These demonstrations are then processed by the learner as explained in section

3.2.1. The demonstrator was one of the authors, who however has no experience in fishing. The demonstrations used were captured only after a few attempt trials, therefore it does not give enough time to the demonstrator to get proficient at this fishing task. The teaching set is composed of 127 samples.

- demo 4: in this set, the demonstrator 3 only selects demonstrations where $a_d^1 < 0$ (in the bottom part)
- demo 5: the demonstrator 3 only selects demonstrations where $a_d^1 > 0$ (in the upper part).

As with the evaluation set, we define a tile of the reachable space. The teacher observes the exploration of the learner, and gives to the learner a demonstration belonging to a subspace randomly chosen among those it has explored the least. This teaching mode is a simple algorithm for active teaching, and can grow more elaborate taking inspiration from the field of Algorithmic Teaching (Cakmak and Lopes, 2012; Cakmak and Thomaz, 2010).

The simulation data and analysis of the results are presented in the following section.

3.4 EXPERIMENTAL RESULTS

In this section, for every simulation on the fishing experiment setup, 5000 movements are performed, and demonstrations taken from either of the demonstrator 3 are given at fixed frequency every 30 movements. The performance was assessed on the same benchmark set every 1000 movements (Figure 3.3.1a).

3.4.1 BETTER PRECISION

Figure 3.4.1 represents how close the learner can get to any goal/outcome of the reachable space in A , at the same timestep of learning and, in the case of social guidance, with the same amount of information given by the teacher. It plots the mean distance error of the attempts to reach the points in the benchmark set, with respect to the learning time (number of movements performed by the robot). The errors are averaged on all points in the benchmark, and also on different runs of the experiment. The 5 algorithms are ranked :

- Learning from Observation performs the worst: this is on the one hand due to the small number of samples, as the learner does not acquire experience on its own but only through observation of others. It is on the other hand due to the correspondence problems. Since the learner and teacher do not have the same policy primitives, the robot can not reproduce exactly the teacher’s movements.
- RANDOM performs better because the learner acquires more data through its own experience, although the exploration is totally random.
- SAGG-RIAC decreases significantly the error value compared to Random Exploration. Not only has the asymptotic performance improved, but SAGG-RIAC also learns faster from the beginning.
- Imitation Learning also decreases significantly the error value compared to Random Exploration. Its error level is comparable to SAGG-RIAC. Therefore, autonomous exploration, and learning that heavily depends on the teacher’s demonstrations are comparable in terms of performance. We can

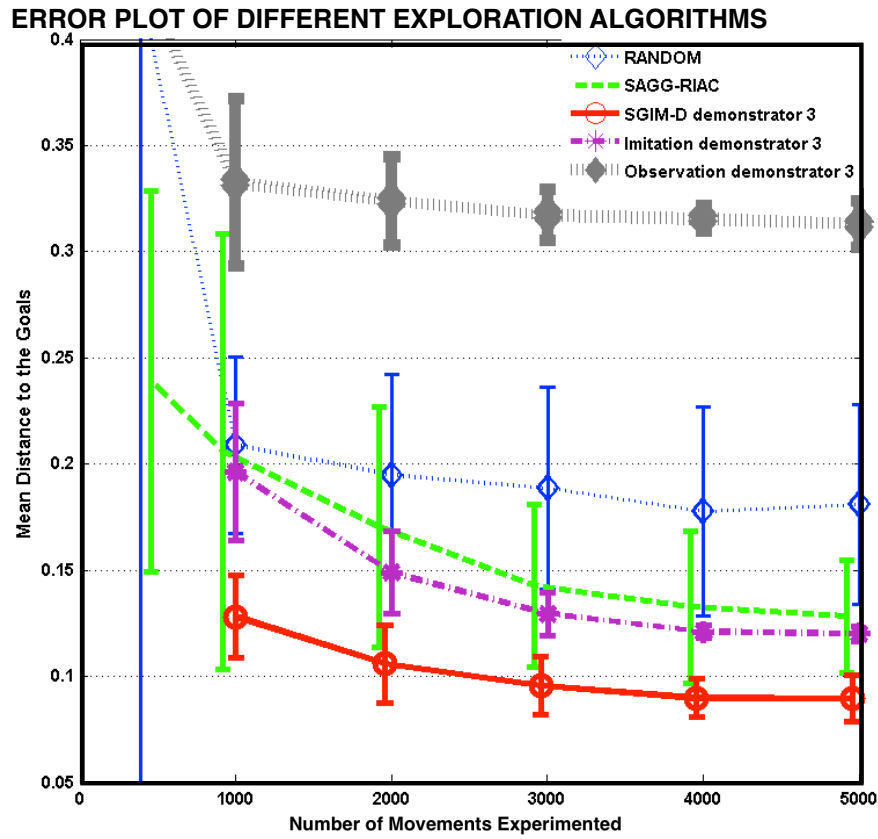


Figure 3.4.1: (best seen in colors) Evaluation of the performance of the robot under the learning algorithms: random exploration, SAGG-RIAC, imitation and SGIM-D (for the human demonstrator 3). We plotted the mean distance to the benchmark points over several runs of the experiment with its variance errorbar.

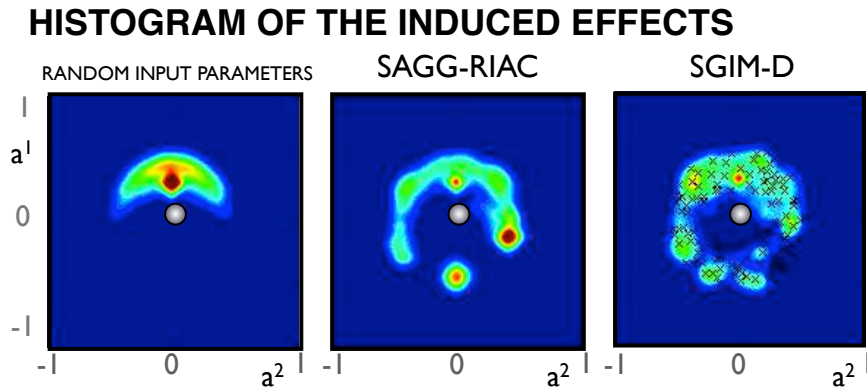


Figure 3.4.2: Histograms of the positions explored by the fishing rod inside the 2D outcome space (a^1, a^2) . Each column represents a different learning algorithm: random input parameters, SAGG RIAC and SGIM-D. We plotted the histogram for one example run of the experiment of each algorithm. In the case of SGIM-D (3rd column), we also graphed the demonstrated outcomes with black crosses.

note that the error variance of Imitation Learning is considerably smaller than that of SAGG-RIAC, because we use the same demonstrator with the same demonstration set, although the order of demonstrations changes. The error variance is likely to increase if we carry out our experiments with various demonstrators.

- SGIM-D performs best and halves the error value compared to Random Exploration. Its asymptotic error approaches the noise level of the stochastic environment. Not only is the error level lower asymptotically, but it drops from the beginning of the learning process. SGIM-D performs better than pure autonomous exploration and pure socially guided exploration.

The combination of autonomous exploration and socially guided exploration has thus bootstrapped the learning to decrease the performance error but also to improve the learning speed.

3.4.2 A WIDE RANGE OF OUTCOMES

To visualise the subspaces explored by each learning algorithm, we plot the histogram of the positions of the float a in the outcome space A when it reaches the water (**Figure 3.4.2**). Each column represents a different algorithm, and we represented for each 2 example experiment runs. The 1st column shows that a natural position lies around $a_c = (0, 0.5)$ in the case of an exploration with random movement parameters. Most movement parameters map to a position of the float around that central position. The second column shows the histogram in the outcome space of the explored points under SAGG-RIAC algorithm. Compared to random exploration, SAGG-RIAC has increased the explored space, and most of all, covers more uniformly the explorable space. Besides, the exploration changes through time as the system finds new interesting subspaces to focus on and explore. Intrinsically motivated exploration has resulted in a wider repertoire for the robot. SGIM-D even emphasises this outcome: the explored space even increases further, with

a broader range of outcomes covered: the minimum and maximum distances to the centre have respectively decreased and increased. Furthermore, the explored space is more uniformly explored, around multiple centres.

The examination of the explored parts of A show that random exploration only reaches a restricted subspace of A , while SAGG-RIAC and SGIM-D increase this explored space. This difference is mainly explained by the fact that most policies map to a restricted subspace of A , and on the contrary, the other parts of the reachable space can only be reached by a very small subset of policy parameters in B . In other words, with random movements, the float has high chances of landing near that natural position. To make it reach other areas of the surface of the water, the arm needs to perform quite specific movements. SGIM-D highlights these areas owing to its outcome space exploration and to demonstrations. The teacher gives a demonstration that triggers the robot’s interest and it is going to focus its attention on that area provided that local exploration improves its competence in this subspace. We also note that the demonstrations occurred only once every 30 movements. Even an occasional presence of the teacher, who does not need to monitor continuously the robot, can significantly improve the performance of the autonomous exploration.

3.4.3 DEPENDENCE ON THE SIZE OF THE OUTCOME SPACE

To test whether our algorithms are scalable to large spaces, we plotted the mean distance error to the benchmark set, for a different outcome space (**Figure 3.4.3**). This time, the boundaries of each dimension have been multiplied by 100, which means that the size of A has been multiplied by 10^4 . We can observe the effects on the performance of the SAGG-RIAC learner. Even though its mean error is lower than the random learner, it has increased compared to the case of the smaller outcome space. On the other hand, SGIM-D still learns to reach any point with good precision. Its mean error is significantly lower than the one of the SAGG-RIAC or the random learners. Consequently, **the social guidance part of SGIM-D has helped it scale to larger spaces by allowing the robot to infer more quickly which parts of the outcome space are actually reachable and learnable.**

3.4.4 IDENTIFICATION OF THE INTERESTING SUBSPACES

To investigate the reasons of the difference in performance between SAGG-RIAC and SGIM-D, especially their different dependence on the outcome space size, we can examine the system’s exploration of the outcome space. **Figure 3.4.4** plots the distribution of all the goals $a_g \in A$ chosen during the outcome space exploration of SAGG-RIAC and SGIM-D. The goals chosen by the SAGG-RIAC learner look disorganised, and cover all the outcome space A , because it needs to sample at a minimum density before computing meaningful measures of interest, and find subspaces where it can actually learn. On the contrary, the SGIM-D learner only chooses its goals around the reachable space. Thus the teacher has helped the SGIM-D learner to identify and target the reachable space.

In conclusion, **SGIM-D improves the precision of the system even with little intervention from the teacher, and helps point out key subregions to be explored.** The teacher successfully transfers his knowledge to the learner and bootstraps autonomous exploration robustly, even in large outcome spaces. This bootstrapping is all the more efficient than the demonstrations chosen by the teacher enhance

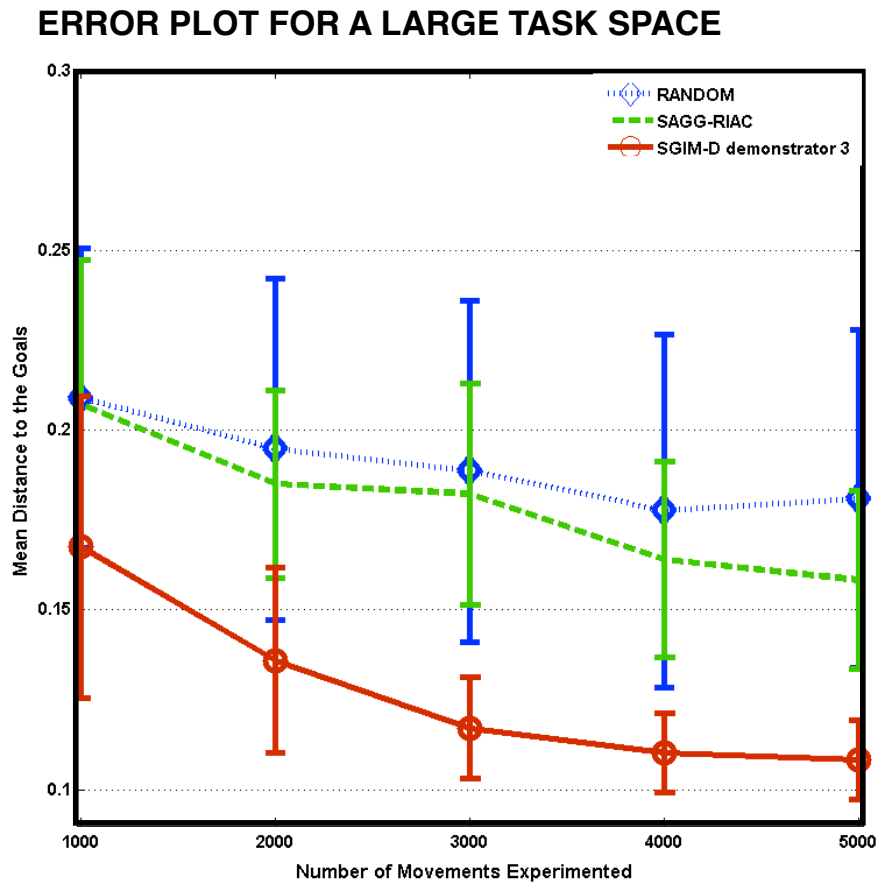


Figure 3.4.3: Evaluation of the performance of the robot in the case of a large outcome space ($A = [-100, 100]^2$ is 10^4 times larger than the reachable space, but we only plotted here the distribution on the subspace $[-1, 1]^2$), under the learning algorithms: random exploration, SAGG-RIAC and SGIM-D.

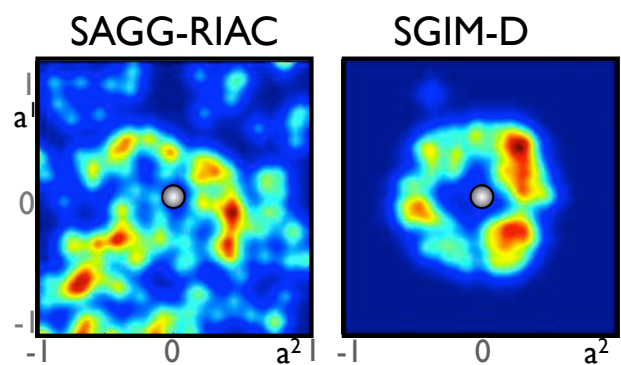


Figure 3.4.4: Distribution of all the goals set by the higher level during learning in a large space. Each column shows the distribution of an experimental run of the SAGG-RIAC algorithm (col 1) or SGIM-D (col 2).

generalisation, for instance through similarity of the policies demonstrated. Although this example has shown that SGIM-D can complete one type of goals only, studies in the next chapter show that it can learn in different kinds of outcome space.

The illustration experiment conducted showed good performance of SGIM-D in learning all the infinity of goals defined by the outcome space A , compared to pure autonomous exploration and socially guided methods, in terms of precision and explored area. Moreover, analysis showed that on the one hand, it benefits from human teacher’s demonstrations which orient its exploration towards small subspaces of policies and goals, and enable a faster identification of interesting subspaces. On the other hand, self experimentation helps it be more robust to demonstrations quality.

3.5 ANALYSIS OF THE BOOTSTRAPPING EFFECT

In the previous section, we showed that SGIM-D can benefit from a bootstrapping effect that comes from the combination of social guidance and intrinsic motivation. However, social guidance heavily depends on the quality of the demonstration. We propose to investigate in this section the reasons and limitations of this bootstrapping effect, by studying the performance of SGIM-D with the different teachers defined in 3.3.3.

The different teachers influence the exploration of both the outcome space and the policy space.

3.5.1 OUTCOME SPACE EXPLORATION

3.5.1.1 DEPENDENCE OF THE PERFORMANCE ON THE TEACHER

Let us examine how the learning of the same SGIM-D algorithm differs in the case of various teachers. **Figure 3.5.1** shows that error rates depend on the teachers. The difference between teachers 1, 2 and 3 will be examined in the following section. We here examine the more interesting contrast between demonstrators 3, 4 and 5. All three demonstration sets come from human teacher teleoperation, with demonstrations 4 and 5 being the subsets of demonstrations 3 for $a_d^1 < 0$ and $a_d^1 > 0$ respectively. Nevertheless, the error plot for demonstrator 4 is similar to that of demonstrator 3, whereas the error rate for demonstrator 5 is in between the error plot of a random or a SAGG-RIAC learner. Therefore, the subspace of A covered by demonstrations is a main factor to the learner’s performance.

3.5.1.2 DIFFERENCE IN THE EXPLORED OUTCOME SPACES

To visualise how the teachers influence the subspaces explored by each learning algorithm, we plot the histogram of the positions a in the outcome space A of the float when it reaches the water (**Figure 3.5.2**). Each column represents a different algorithm or teacher. We represent for each 2 example experiment runs. The 1st column shows that a natural position lies around $a_c = (0, 0.5)$ in the case of an exploration with random movement parameters. Most movement parameters map to a position of the float around that central position. This is due to the configuration of the fishing rod which initial state is close to the water surface. Therefore, most random movements would easily drop the float into the water. On the contrary, to reach positions far from a_c , the robot has to make quite specific movements to lift the rod and make the float reach

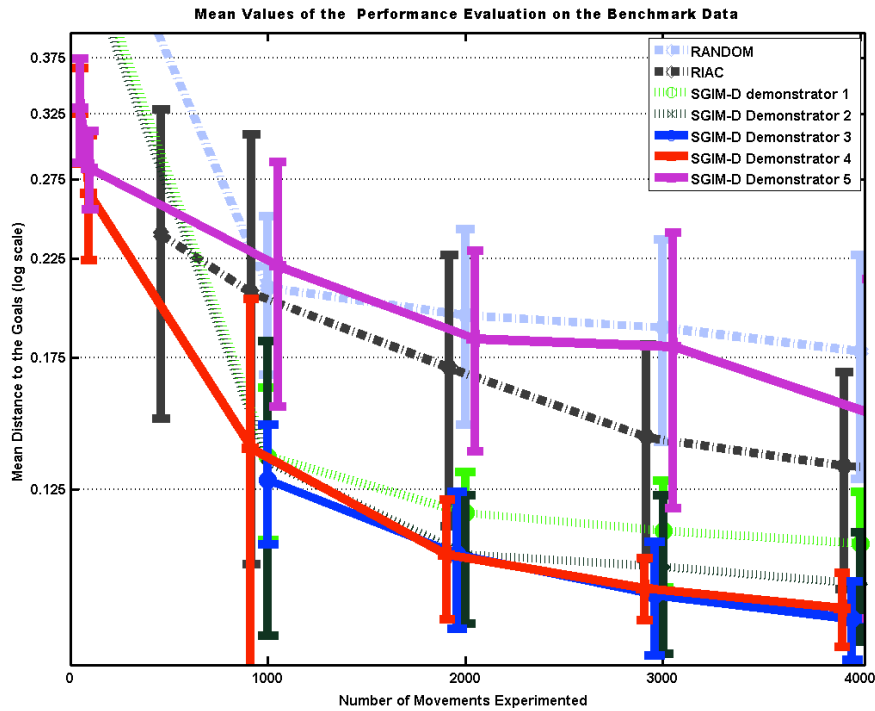


Figure 3.5.1: SGIM-D's performance depends on the demonstrator

farther areas. The second column shows the histogram in the outcome space of the explored points under SAGG-RIAC algorithm. Compared to random exploration, SAGG-RIAC has increased the explored space, and most of all, covers more uniformly the explorable space. Besides, the exploration changes through time as the system finds new interesting subspaces to focus on and explore. Intrinsically motivated exploration has resulted in a wider repertoire for the robot. SGIM-D (demonstrator 3 and 4) even emphasises this effect: the explored space even increases further, with a broader range of radius covered: the minimum and maximum distances to the centre have respectively decreased and increased. Furthermore, the explored space is more uniformly explored, around multiple centres. The examination of the explored parts of A show that random exploration only reaches a restricted subspace of A , while SGIM-D increases this explored space owing to its outcome space exploration and to demonstrations. However, the case of demonstrator 5 (SGIM-D), demonstrations are given only in subspaces $a_d^1 > 0$ of A that are often reached by random or SAGG-RIAC exploration. **Figure 3.5.2** shows a outcome space exploration which is broader than the random learner, but still more restricted than the SAGG-RIAC learner. Indeed, this SGIM-D learner only explores around the demonstrated area and neglects other parts of the outcome space. Demonstrations for easy goals entail poor performance for the learner, whereas demonstrations for difficult goals enhance better progress.

Therefore, one of the main bootstrapping factors of SGIM-D is the outcome space exploration. **The teacher influences the exploration of difficult outcomes, either by encouraging it with demonstrations of difficult goals, or by hindering it from focusing attention too much on the easy goals.**

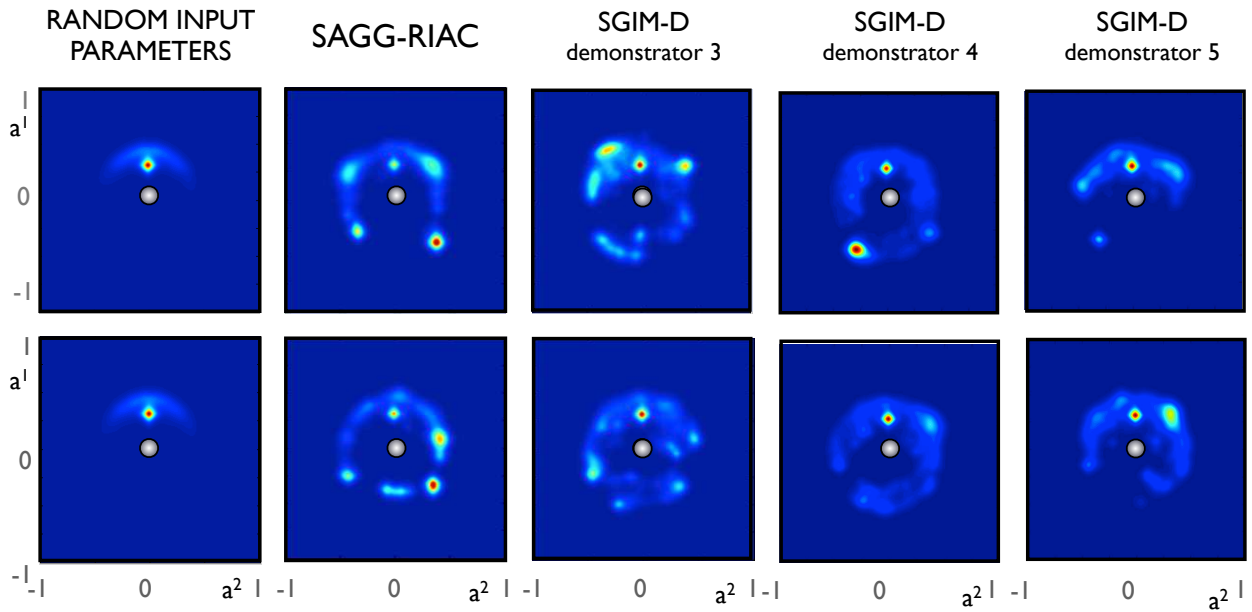


Figure 3.5.2: Histogram of the outcomes explored by the fishing rod inside the 2D outcome space. Each algorithm is illustrated by 2 example experiments.

3.5.2 POLICY SPACE EXPLORATION

Figure 3.5.3 also shows that there are differences in the error plots for the case of teachers 1, 2 and 3, even though their demonstrations cover the same subspace in A . Let us examine the difference between the teachers 1, 2 and 3.

3.5.2.1 DEPENDENCE OF SGIM-D PERFORMANCE ON THE QUALITY OF DEMONSTRATIONS

We plot the mean error of the socially guided algorithms for our 3 different demonstrators (**Figure 3.5.3**). First of all, we notice that for all 3 teachers, SGIM-D performs better than the other algorithms (t-test with $p < 0.05$ for the error (mean distance to the goals) at $t=5000$). **SGIM-D is therefore robust with respect to the quality of the demonstration** as the teacher only guides the learner towards interesting action or effect subspaces, and the learner lessens its dependence on the teacher owing to self-exploration. Still, among the 3 demonstration sets we used, some perform in average better than others. As expected, the demonstrations 1 that are chosen randomly bootstrap less than the demonstrations 2 that have smaller variance (t-test with $p < 0.05$). We also note that the human demonstrations (3), also bootstrap better than demonstrations 1 (t-test with $p < 0.05$). This result seems at first sight surprising, as the results of learning by observation seem to indicate the contrary: demonstrator 1 or 2 are more beneficial to the observation learner (t-test with $p < 0.05$), since demonstrator 3’s actions can be not easily reproduced due to correspondence problems.

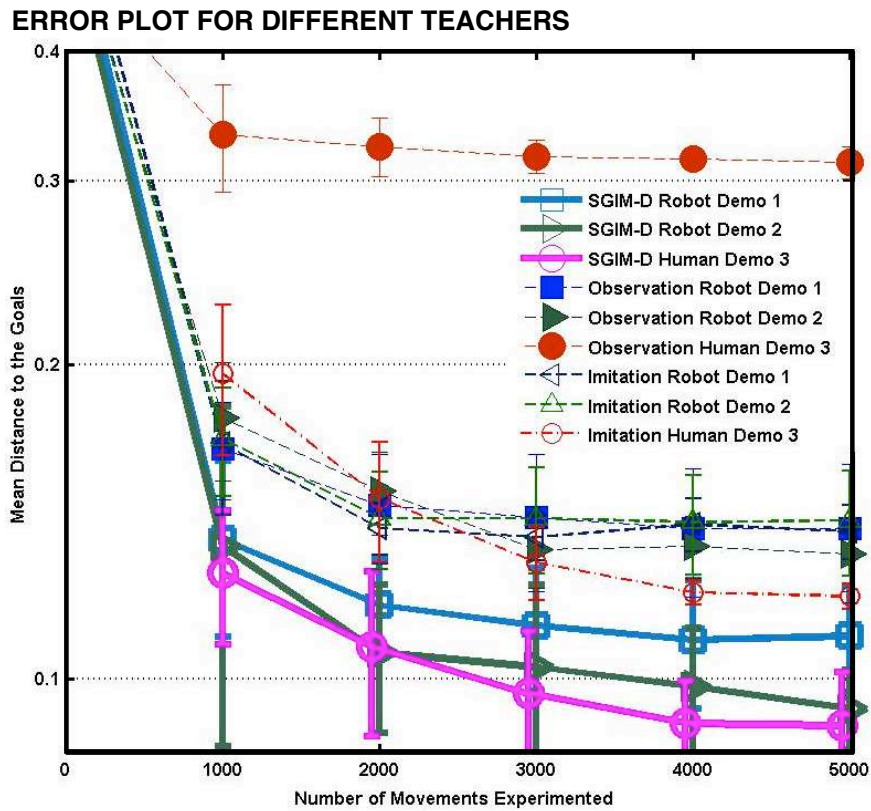


Figure 3.5.3: Evaluation of the performance of the robot learning with 3 different demonstrators, under the learning algorithms: SGIM-D, Observation and Imitation.

PROPERTIES OF THE DEMONSTRATIONS:

Trajectories for joint 1

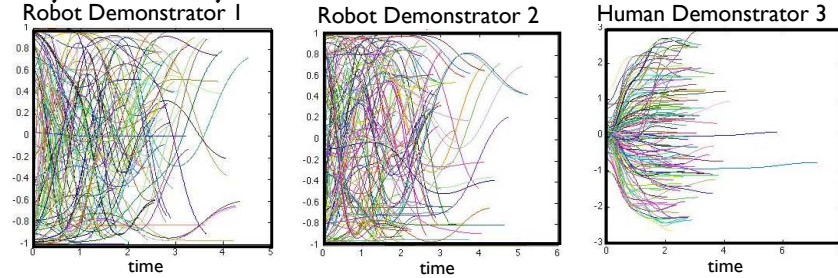


Figure 3.5.4: Plot for the demonstrations of the trajectories for joint 1 (vertical axis: joint angles, horizontal axis: time).

3.5.2.2 ANALYSIS OF THE DEMONSTRATED MOVEMENTS

To understand the reasons of this result, let us examine the different demonstrations. **Figure 3.5.4** plots the trajectories of the demonstrations. We can see that demonstrations show different distribution characteristics in the trajectory profile. The most noticeable difference is the case of demonstrator 3. Whereas the trajectories of demonstrators 1 and 2 seem disorganised, the joint value trajectories of demonstrator 3 are all monotonous, and seem to have the same shape, only scaled to match different final values. Indeed, the comparison of the demonstrations set 3 to random movements with ANOVA (Krzanowski, 1988) indicates that we can reject the hypothesis that demonstration set 3 comes from a random distribution ($p = 4.10^{-40}$). The demonstrations set 3 is not randomly generated but are well structured and regular. Therefore, the human demonstrator shows a bias through his demonstrations to the robot, and orients the exploration towards different subspaces of the policy space. Indeed, the ANOVA analysis of the movements parameters a performed during the learning reveals that they have different distributions with separate means. Because his demonstrations have the same shape, they belong to a smaller, denser and more structured subset of trajectories from which is easier for the learner to generalise, and build upon further knowledge. Moreover, this comparative study highlights another advantage of SGIM-D: its **robustness to the quality of demonstrated policies**. The performance varies depending on the teacher, but still is significantly better than the SAGG-RIAC or imitation learner.

3.6 PRELIMINARY RESULTS ON A PHYSICAL ROBOT

In addition to these results in simulation, we also started to make the same experiments with a physical 6 DOF robot (**Figure 3.6.1**). To this robot we attached a real fishing rod and a wire. At the end of the wire, is placed a red ball, which position is tracked by a camera from above the floor. The robot can thus learn to throw the red ball anywhere on the field of view of the camera, which is represented by the white surface on **Figure 3.6.2**. This region visible by the camera is normalised so that $A = [0, 1]^2$. The robot considers

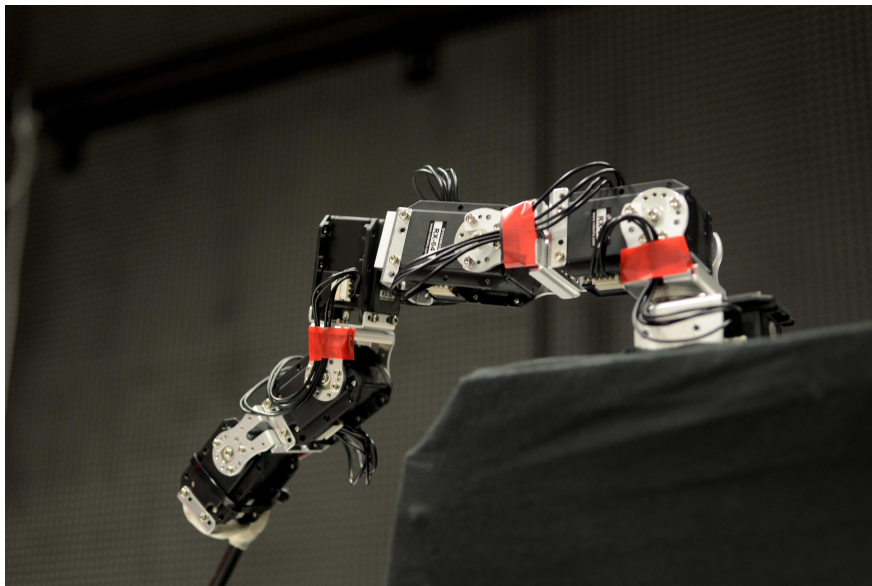


Figure 3.6.1: A 6 DOF robot arm used in our fishing experiment.

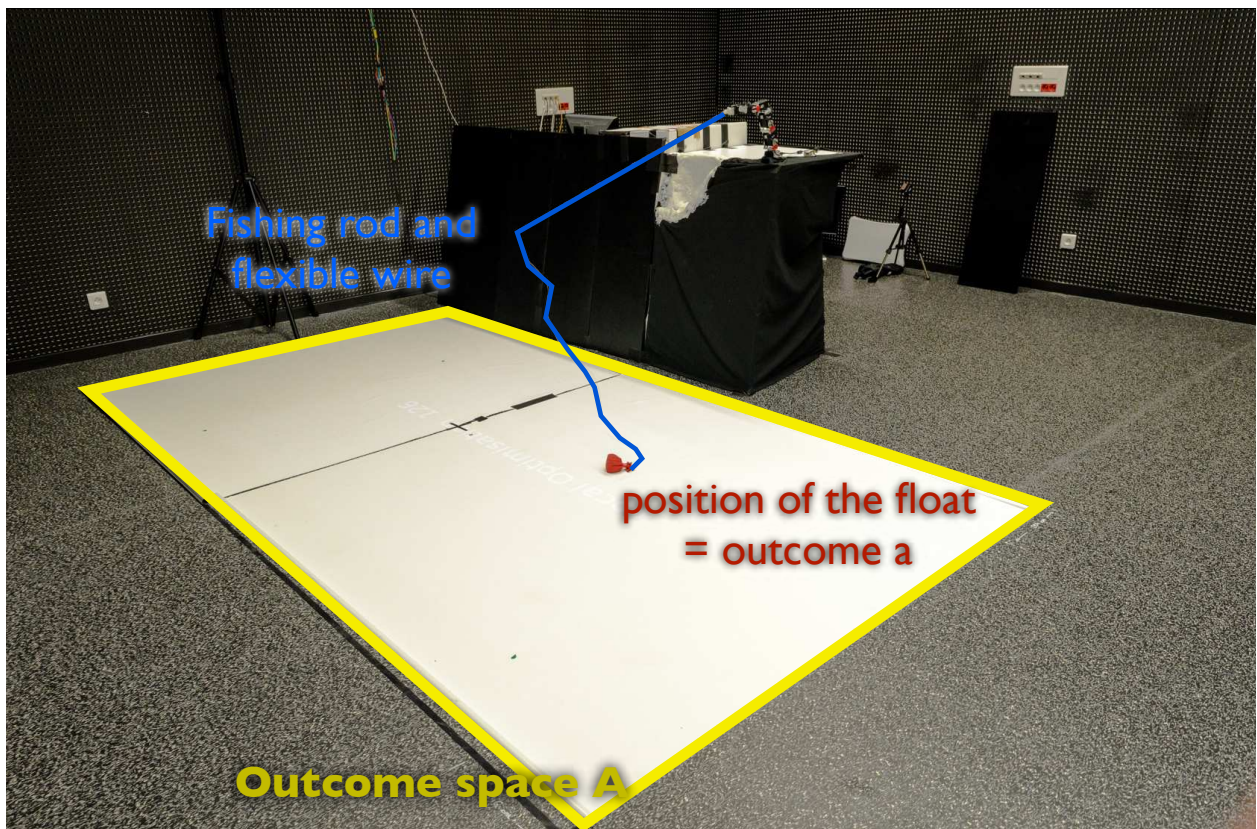


Figure 3.6.2: The robot observes the final position of the ball after its movement, and learns which movement can reach different positions on the floor. The camera is placed above the wide surface, and can only see the white surface which materialises the outcome space A .

that it reached the outcome space A only when the ball reaches the floor and stabilises in the field of view of the camera. If the ball is still in the air, or is outside the field of view of the camera, it considers it has reached an outcome which is distant from A from a constant, set to 100.

In this experiment, we controlled the robot by parameterised dynamic trajectories too, but this time the number of parameters is 54. The policy space is $B = [0, 1]^{52}$.

Preliminary results show that while random exploration and SAGG-RIAC seldom touch the floor, SGIM-D reaches the floor much more often. Are represented in **Figure 3.6.3** the points that were reached by the red ball on the floor. They cover a greater surface in the case of SGIM-D than in the case of random exploration or autonomous exploration. We also plotted in **Figure 3.6.4** the mean error of each algorithm. It shows that random exploration and SAGG-RIAC make the same level of errors. It can be explained by the fact that most movements do not make the float touch the floor due to the large dimensionality of the policy space. On the contrary, owing to demonstrations, SGIM-D learns quickly how to reach the floor, and its error level is 5 to 10 times smaller. These are only preliminary results. More experiments should confirm these results, and enable statistical analysis.

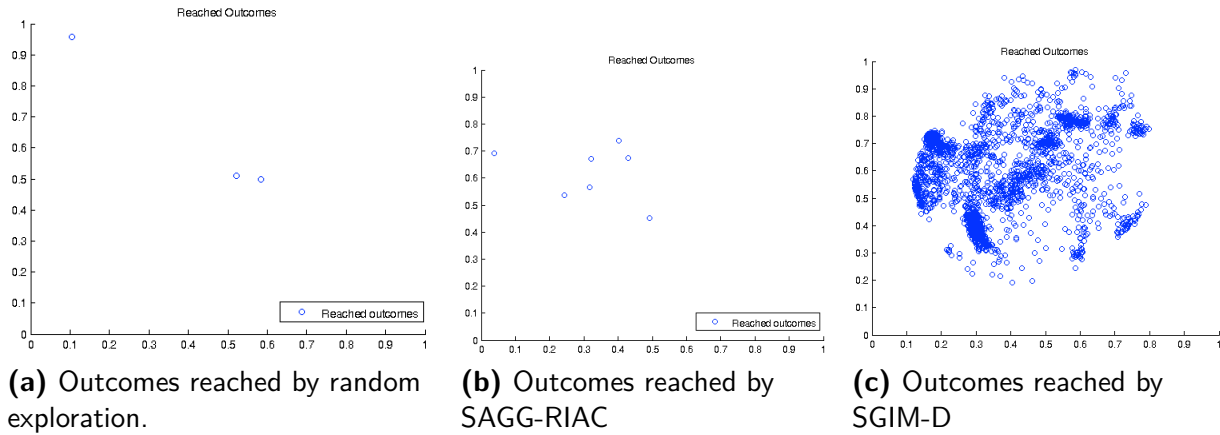


Figure 3.6.3: Outcomes reached by the red ball if it stabilises on the floor. The axis are the two dimensions of the floor A . The region visible by the camera is normalised so that $A = [0, 1]^2$.

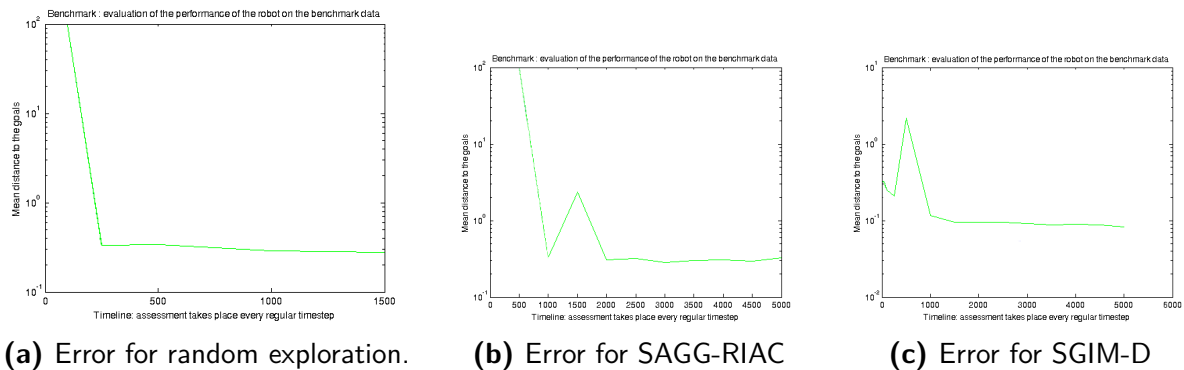


Figure 3.6.4: Mean error on an experiment for each algorithm, with respect to time.

3.7 BENEFITS OF THE COMBINATION: CONCLUSION

This chapter introduced Socially Guided Intrinsic Motivation by Demonstration, **SGIM-D**, an architecture for online active learning of inverse models in continuous high-dimensional robotic sensorimotor spaces, and allowing a robot to learn multiple goals and generalise over a continuous ensemble of goals. SGIM-D efficiently combines social guidance and intrinsic motivation modes on both the policy and goal exploration levels. It actively samples goals while adapting to the difficulty of different subspaces. The analysis of the properties of this combination shows that the demonstrations structure and orient the exploration towards a subspace of the policy space, independently of whether the demonstrations can be exactly reproduced by the learner or not. SGIM-D also takes advantage of the intrinsically motivated autonomous exploration to improve its performance and gain precision in the absence of the teacher for a wide range of outcomes/goals. It is an original algorithm in that it is at the same time an active learning system of inverse models benefiting from human demonstrations, and also a PbD system which can learn and generalise to new goals. Our simulation indicates that SGIM-D successfully learns motor control even in an experimental setup as complex as having a continuous 25-dimensional policy parameter space.

In this first step, for the sake of comparison of SGIM-D to other algorithms, we do not study further the effects of different parameters of social interaction on the performance of the robot, for instance the impact of the frequency of the demonstrations given by the teacher. The parameters of the teaching, such as the rationales for selecting timing of the social interaction and demonstrations have not been chosen in this chapter to optimise SGIM-D. In other words, using the notations of the formalisation of section 1.4.2, χ_e was not determined by the learning agent, but was hand-coded. A more precise study of the teaching parameters, and an optimisation with respect to χ_e are undertaken in the next chapter. Moreover, we could explore in depth the dependency of SGIM-D on the teacher. Cases of sparse teachers, where the demonstrations belong to a small subspace only, or are in smaller number have been studied in this chapter. Such studies illustrate the most general case when the human teacher can not perform everything, but is only proficient in a small subset of goals. We can also extend the work with a learner who self-determines whether to take into account a demonstration or not, taking inspiration from child psychology studies that show limitations of the role of parents (Xu et al., 2011). Such work on the automatic determination of the field of competence of different demonstrators, and the decision whether to take into account their demonstrations will be studied further in section 4.3.

Most of all, we only considered in this chapter a very simple interaction scenario between the learner and the teacher, and we did not take into account interactive learning (Chernova and Veloso, 2009; Thomaz, 2006; Nicolescu and Mataric, 2003), where the learner asks for information when needed. More generally, exploring and evaluating systematically the other scenarios in which a human teacher can be involved, as mentioned in section 1.2.4, should be instructive. An interesting angle to study would also be the switching between mimicking, imitation and emulation modes. In this chapter, the robot mimics the teacher for a fixed amount of time, and afterwards, SGIM-D takes into account these new data only from the goal point of view, as in emulation. A more natural and autonomous algorithm for switching between or combining these different modes is shown to improve the efficiency of the system in the next chapter.

An expert is a person who has found out by his own painful experience all the mistakes that one can make in a very narrow field.

Niels Bohr

4

Interactive Strategic Learner

Contents

| | | |
|---------|--|-----|
| 4.1 | What is Interactive Learning? | 75 |
| 4.2 | Interactive Learning at the Meta Level : SGIM-IM | 76 |
| 4.2.1 | Algorithm Description | 76 |
| 4.2.1.1 | SGIM-IM Overview | 76 |
| 4.2.1.2 | Select Behaviour | 78 |
| 4.2.2 | Air Hockey Experiment | 79 |
| 4.2.2.1 | Air Hockey Experimental Setup | 79 |
| 4.2.2.2 | Experimental Protocol | 80 |
| 4.2.2.3 | Results | 81 |
| 4.2.2.4 | Active Choice of Behaviour | 81 |
| 4.2.3 | Fishing Experiment | 83 |
| 4.2.3.1 | Experimental Setup | 83 |
| 4.2.3.2 | Results | 83 |
| 4.2.4 | Discussion and Conclusion | 86 |
| 4.3 | SGIM-ACTS | 87 |
| 4.3.1 | Actively Learning When, Who and What to Imitate | 87 |
| 4.3.1.1 | Choice for Social Guidance | 87 |
| 4.3.1.2 | Interactive Learning Based on Intrinsic Motivation | 88 |
| 4.3.2 | Algorithm Description | 89 |
| 4.3.2.1 | Architecture Outline | 89 |
| 4.3.2.2 | Hierarchical Structure | 91 |
| 4.3.2.3 | Policy Space Exploration | 92 |
| 4.3.2.4 | Sampling Mode and Outcome Space Exploration | 92 |
| 4.3.3 | Throwing and Placing a Ball | 95 |
| 4.3.3.1 | Experimental Setup | 95 |
| 4.3.3.2 | Several Teachers and Sampling Modes | 97 |
| 4.3.3.3 | Comparison of Learning Algorithms | 97 |
| 4.3.3.4 | Results | 98 |
| 4.3.3.5 | Conclusion and Discussion | 103 |

In the previous chapter, we showed that the combination of two exploration modes, social guidance and intrinsic motivation, bootstrap the learning of redundant inverse models in continuous high-dimensional spaces. Nevertheless, the algorithmic architecture used, SGIM-D (Socially Guided Intrinsic Motivation by Demonstration) is a passive system with respect to its teacher. It only imitates when the teacher decides to give a demonstration and does not try to optimise the different parameters of social interaction, such as the frequency of the demonstrations by requesting the teacher’s help. SGIM-D does not learn which method enables it to perform best. In other words, SGIM-D addresses the problem of *what and how to learn*, but does not answer the questions of *what, when, who to imitate*. Using the notations of the formalisation of section 1.4.2, for each learning episode e , SGIM-D only determines its point of focus ψ_e , and does not determine its sampling mode χ_e .

In this chapter, we design algorithmic architectures to address these limitations by allowing the agent to **make an increasing number of decisions about its learning method and interaction with teachers**. We thus design an *interactive learner* as we explain in section 4.1. In section 4.2, we first propose an architecture called Socially Guided Intrinsic Motivation with Interactive Learning at the Meta level (SGIM-IM), which determines its sampling mode χ_e by deciding *when* to imitate. Then in section 4.3, we design Socially Guided Intrinsic Motivation with Active Choice of Teacher and Strategy (SGIM-ACTS), which determines its behaviour χ_e by actively deciding *when, what and who to imitate*.

4.1 WHAT IS INTERACTIVE LEARNING?

An *interactive learner* not only listens to the teacher, but **actively requests for the information it needs and when it needs help**. Such interaction upon the learner’s initiative has been shown to be a fundamental aspect of social guidance. Under the interactive learning approach, the robot can combine programming by demonstration, learning by exploration and tutor guidance. Several works in interactive learning have considered extra reinforcement signals (Thomaz and Breazeal, 2008), action requests (Grollman and Jenkins, 2010; Lopes et al., 2009a) or disambiguation among actions (Chernova and Veloso, 2009). In (Cakmak et al., 2010) the comparison of a robot that has the option to ask the user for feedback, to the passive robot, shows a better accuracy and fewer demonstrations. Chernova and Veloso (2009) shows that a learning agent which actively requires demonstrations from human teachers by identifying uncertain states, requires fewer demonstrations than an agent which learns from demonstrations given at the teacher’s initiative. Therefore, requesting demonstrations when it is needed can lessen the dependence on the teacher and reduce the quantity of the demonstrations required. This approach is the most beneficial to the learner for the information arrives as it needs it. It is also beneficial to the teacher who no longer needs to monitor the learning process to optimise his teaching.

Nevertheless, most of these works address mainly only the question of when to imitate. This is why we design an interactive learning algorithms with intrinsically motivated robot learners, which decide themselves firstly *when* it is most beneficial to imitate the teacher with algorithm **SGIM-IM**, but also in the section 4.3 *what and who* to imitate with algorithm **SGIM-ACTS**.

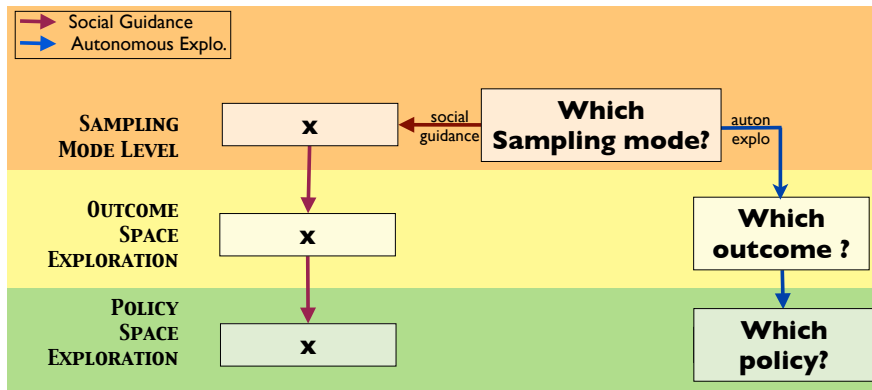


Figure 4.2.1: The strategic learner samples data by actively choosing on the three levels on its exploration space: its sampling mode, the object to focus on and the policy to try.

4.2 INTERACTIVE LEARNING AT THE META LEVEL : SGIM-IM

SGIM-IM (Socially Guided Intrinsic Motivation with Interactive learning at the Meta level) is an algorithm that merges interactive learning as social interaction, with the SAGG-RIAC algorithm of intrinsic motivation (Baranes and Oudeyer, 2013), to learn local inverse and forward models in complex, redundant, high-dimensional and continuous spaces. We first describe in section 4.2.1 the design of our SGIM-IM algorithm, which **actively chooses the best exploration mode between intrinsically motivated exploration and imitation learning**. Then we show that SGIM-IM efficiently requests for the teacher’s demonstrations to complete a wide range of tasks, while being specialised in specific subspaces through 2 experimental setups: an air hockey game and a fishing skill learning in sections 4.2.2 and 4.2.3 respectively. The results presented in this section have been published in (Nguyen and Oudeyer, 2012c).

4.2.1 ALGORITHM DESCRIPTION

In this section based on the formalisation of section 3.1, we describe the SGIM-IM algorithmic architecture, which combines both intrinsic motivation and imitation learning, like SGIM-D. The difference being that SGIM-IM builds up an interactive learner deciding actively which mode to use at each data sampling episode to learn motor skills. As summarised in **Figure ??**, the active learner decides on its sampling mode, its goal outcome and the policy to try.

4.2.1.1 SGIM-IM OVERVIEW

SGIM-IM learns by episodes during which it selects which of the two preset sampling modes to use, between intrinsically motivated (χ_{auto}) or socially guided (χ_{dem}) exploration. Thus \mathcal{X} is a set of 2 predefined sampling modes as described in section 3.1.1. The algorithmic architecture is summarised in **Figure 4.2.2**.

In an episode with the intrinsic motivation mode, it explores autonomously following the SAGG-RIAC algorithm (Baranes and Oudeyer, 2013). It actively self-generates a goal a_g where its competence improvement is maximal, then explores which policy π_b can achieve a_g best. The SGIM-IM learner explores preferentially goal tasks which are easy to reach and where it makes progress the fastest. It tries different

policies to approach the self-determined task a_g , re-using and optimising the estimation of J built through its past autonomous and socially guided explorations. The episode ends after a fixed duration

In an episode with the socially guided exploration mode, our SGIM-IM learner observes the demonstration $[\zeta_d, a_d]$, memorises this task a_d as a possible goal, and mimics the teacher by performing policies π_b to reproduce ζ_d , for a fixed duration. This mode highlights useful tasks, and teaches the learner at least one way to complete a new task, whereas self-exploration has low chance of discovering useful tasks.

The difference with SGIM-D is that the SGIM-IM learner actively decides on a meta level which mode to choose according to the recent learning progress enabled by each mode. If it has recently made the most progress in the intrinsic motivation mode, it prefers exploring autonomously. Conversely, if the demonstrations do not enable him to make higher progresses than by autonomous learning (limited teacher, or inappropriate teacher) it would prefer autonomous exploration.

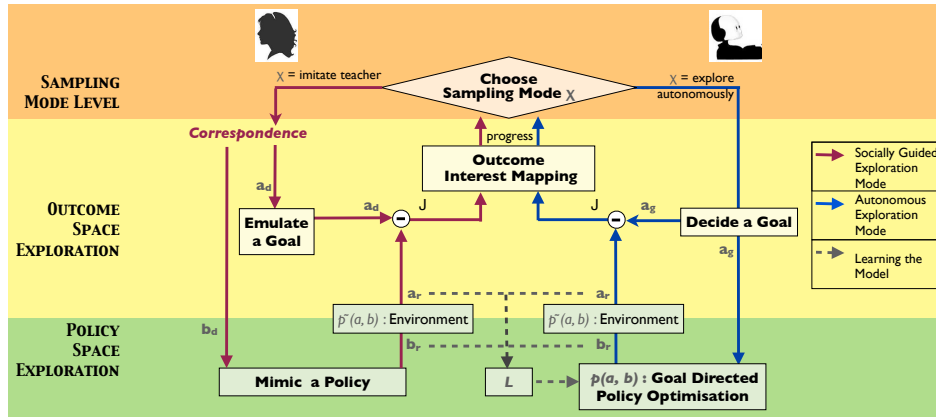


Figure 4.2.2: Time flow chart of SGIM-IM, which combines Intrinsic Motivation and Social Learning into 3 layers that pertain to the human-machine interface, the outcome space exploration and the action space exploration respectively.

Its architecture is separated into three layers (**Figure 4.2.2**), two of which are based on SGIM-D :

- Outcome Space Exploration : This level of active learning drives the exploration of the outcome space. With the autonomous exploration mode, it sets goals a_g depending on the interest level of previous goals (*Decide a Goal*). With the socially guided exploration mode, it retrieves from the teacher information about demonstrated effects a_d (*Emulate a Goal*). Then, it maps A in terms of interest level (*Goal Interest Mapping*). It learns at a longer time scale.
- Action Space Exploration : This lower level of learning explores the policy parameters space B to build an action repertoire and local models. With the socially guided exploration mode, it imitates the demonstrated actions ζ_d (*Imitate an Action*), while during self-exploration, the *Goal-Directed Policy Optimisation* function attempts to reach the goals a_g set by the *Task Space Exploration* level, then, it returns the measure of competence at reaching a_d or a_g .

For details on these two layers, please refer to 3.1.4. In the following paragraphs, we describe the third layer: the sampling mode selection.

Algorithm 4.2.1 SGIM-IM

```

1: Initialization:  $\mathcal{R} \leftarrow$  singleton  $C \times A$ 
2: Initialization:  $\mathcal{H} \leftarrow$  empty episodic memory (collection of episodes (c,b,a))
3: Initialization:  $\Delta_S$  : progress values made by social guidance mode
4: Initialization:  $\Delta_A$  : progress values made by intrinsic motivation mode
5: Initialization:  $e \leftarrow 1$ 
6: loop
7:    $\chi_e \leftarrow$  Select Sampling Mode( $\Delta_S, \Delta_A$ )
8:   if  $\chi_e == \chi_{demo}$  then
9:     Social Learning Mode
10:    demo  $\leftarrow$  ask and perceive demonstration
11:     $(c_d, b_d, a_d) \leftarrow$  Correspondence (demo)
12:    Emulate Goal:  $a_g \leftarrow a_d$ 
13:     $\gamma_i \leftarrow$  Competence for  $a_g$ 
14:     $\mathcal{D}_e \leftarrow$  Mimic Policu( $b_d$ )
15:     $\gamma \leftarrow$  Competence for  $a_g$ 
16:     $p_{e+1} \leftarrow \mathcal{L}(p_e, \mathcal{D}_e)$ 
17:    Add  $\gamma - \gamma_i$  to stack  $\Delta_S$ 
18:   else
19:     Intrinsic Motivation Mode
20:     $a_g \leftarrow$  Decide a goal( $\mathcal{R}$ )
21:     $\gamma_i \leftarrow$  Competence for  $a_g$ 
22:    repeat
23:       $\mathcal{D}_e \leftarrow$  Goal-Directed Policy Optimisation( $a_g$ )
24:       $p_{e+1} \leftarrow \mathcal{L}(p_e, \mathcal{D}_e)$ 
25:    until Terminate reaching of  $a_g$ 
26:     $\gamma \leftarrow$  Competence for  $a_g$ 
27:    Add  $\gamma - \gamma_i$  to stack  $\Delta_A$ 
28:   end if
29:   Append  $\mathcal{D}_e$  to  $\mathcal{H}$ 
30:    $\mathcal{R} \leftarrow$  Update Goal Interest Mapping( $\mathcal{R}, \mathcal{H}, c, a_g$ )
31:    $e \leftarrow e + 1$ 
32: end loop

```

4.2.1.2 SELECT BEHAVIOUR

A meta level actively chooses the best sampling mode based on the recent progress made by each of them. As in SGIM-D, \mathcal{X} is a set of two predefined sampling modes as detailed in section 3.1.1:

- intrinsically motivated exploration (χ_{auto})
- socially guided exploration (χ_{dem})

For each episode e , the learner measures its competence progress for goal a (**Algorithm 4.2.1**, lines 17 and 27) as the difference between the competence before any attempt to reach a_g (lines 13 and 21) and the competence after imitation or goal-directed policy optimisation (lines 15 and 26). The learner adds this progress value to stacks Δ_A or Δ_S . The preference for each mode is computed as the average on a window frame of the last ns progress values of Δ_A and Δ_S . Besides, in order to limit the reliance on the teacher, we penalise the preference for social guidance with a *cost* factor (**Algorithm 4.2.2**, lines 12 and 13). The

Algorithm 4.2.2 $[\chi] = \text{SelectSamplingMode}(\Delta_S, \Delta_A)$

input: Δ_S : progress values made by social learning Mode
input: Δ_A : progress values made by intrinsic motivation learning Mode
output: χ : chosen Mode
parameter: $nbMin$: duration of the initiation phase
parameter: ns : window frame for monitoring progress
parameter: $cost$: cost of requesting a demonstration
Initiation phase
if Social Learning and Intrinsic Motivation Modes have not been chosen each $nbMin$ times yet **then**
 $ps \leftarrow 0.5$
else
Permanent phase
 $wa \leftarrow \text{average}(\text{last } ns \text{ elements of } \Delta_A)$
 $ws \leftarrow \text{average}(\text{last } ns \text{ elements of } \Delta_S)$
 $ps \leftarrow \min(0.9, \max(0.1, \frac{cost \cdot ws}{ws + wa}))$
end if
 $\chi \leftarrow \chi_{dem}$ with probability ps
return χ

modes are selected stochastically with a probability proportional to their preference (line 7). Therefore, autonomous exploration is preferred if it provided highest competence progress in the recent past, while social guidance is preferred only if its progress were $cost$ times higher.

We applied our hierarchical **SGIM-IM** algorithm with 2 layers of active learning to 2 illustration experiments in sections 4.2.2 and 4.2.3. In the next section, we report our first experiment with an air hockey game.

4.2.2 AIR HOCKEY EXPERIMENT

We first apply SGIM-IM to our air hockey game, which setup is described in section 4.2.2.1. Then we report the experimental results in section 4.2.2.3.

4.2.2.1 AIR HOCKEY EXPERIMENTAL SETUP

(I) **DESCRIPTION OF THE ENVIRONMENT** Let us illustrate the SGIM-IM algorithmic structure with an example of a simulated square air hockey table that contains an obstacle (**Figure 4.2.3**). Always starting with the same position and velocity, the puck moves in straight line without friction. The outcome the agent can observe is the position of the impact when the puck collides with the top border of the table. A is thus the top border of the table, mapped into the $[-1, 1]$ segment. We note that the subregion hidden by the obstacle is difficult to reach.

We control the mallet with a parameterised trajectory determined by 5 key positions $b_0, b_1, b_2, b_3, b_4 \in [-1, 1]^2$ at times $t_0 = 0 < t_1 < t_2 < t_3 < t_4$. The executed trajectory is generated by Gaussian distance weighting:

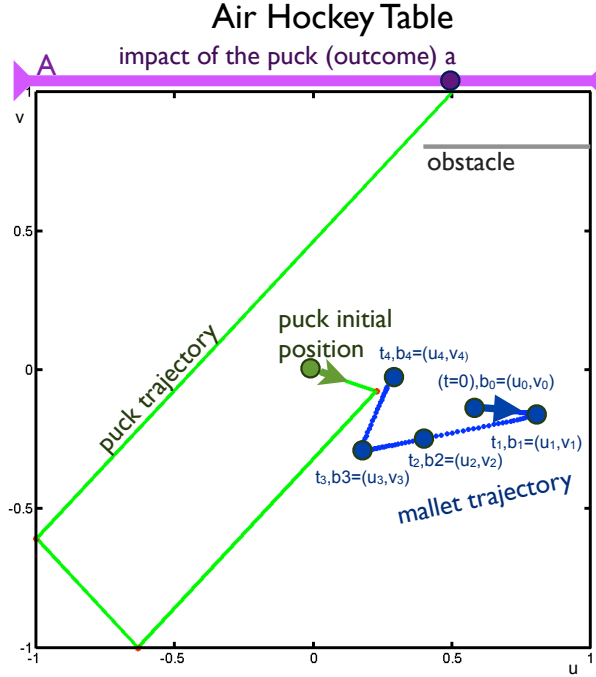


Figure 4.2.3: Air Hockey Table: the task space is defined as the top border of the square. The puck moves in straight line without friction until it hits either the mallet, the table borders or the obstacle placed on the right side.

$$\zeta(\mathbf{t}) = \sum_{i=0}^5 \frac{w_i(\mathbf{t})b_i}{\sum_{j=0}^5 w_j(\mathbf{t})} \text{ with } w_i(\mathbf{t}) = e^{\chi*|\mathbf{t}-\mathbf{t}_i|^2}, \chi > 0 \quad (4.1)$$

Therefore, the policy parameter space $B = \mathbf{R}^n$ and $A = [-1, 1]$. B is of dimension $n = 14$ and A of dimension 1. The learner maps which trajectory of the mallet with parameter $b = (b_1, \dots, b_{14})$ induces a collision with the top border at position a . This is an inverse model of a highly redundant mapping, which is all the more interesting than the obstacle introduces discontinuities in the model.

(II) DEMONSTRATIONS AND EVALUATION We simulate a teacher by using the learning data (ζ_d, a_d) taken from a random explorer and intrinsically motivated explorer based on SAGG-RIAC algorithm (Baranes and Oudeyer, 2013) as detailed later in section 3.1.1.2. We choose 500 demonstrations so that a_d is evenly distributed in $[0.5, 1]$. The teacher is thus specialised in a restricted domain of A . The demonstrations of that batch are given to the learner in a random order.

We assess our agent by measuring how close it can reach a benchmark set that defines the user's region of interest. In this case, the benchmark set is distributed over $T = [-1, 1]$ and placed every 0.05, to get the mean error at reaching these benchmark points.

4.2.2.2 EXPERIMENTAL PROTOCOL

In the same principle as in Chapter 3 for SGIM-D, here, to assess the efficiency of SGIM-IM, we compare the performance of several learning algorithms (Figure 4.3.6):

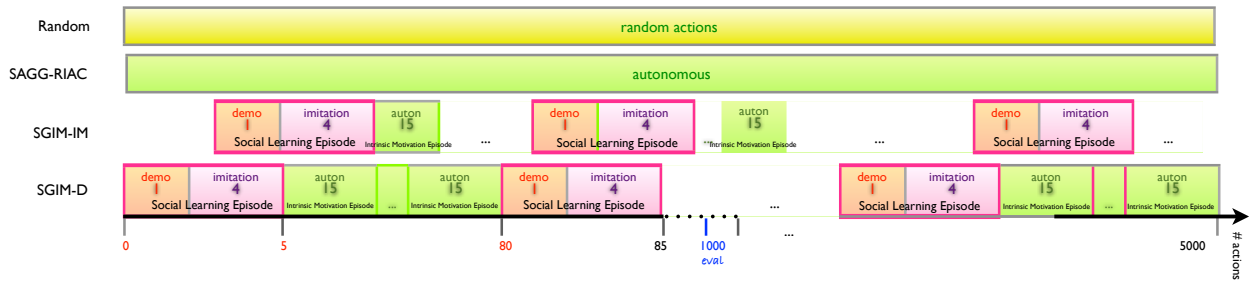


Figure 4.2.4: Comparison of several learning algorithms. Each box represents the chronology of the adopted modes (the figures correspond to the number of actions experimented in the episode). The figures here are given for the Fishing experiment).

- Random exploration: throughout the experiment, the robot picks policy parameters randomly in B .
- SAGG-RIAC: throughout the experiment, the robot explores autonomously, without taking into account any demonstration, and is driven by intrinsic motivation.
- SGIM-IM: interactive learning where the robot learns by actively choosing between socially guided exploration mode or intrinsic motivation mode.
- SGIM-D: the robot’s mode is a mixture between Imitation learning and SAGG-RIAC, as detailed in section 3.1. When the robot sees a new demonstration, it imitates the trajectory for a short while. Then, it resumes its autonomous exploration, until it sees a new demonstration by the teacher, which occurs every M actions experimented by the robot.

For each experiment in our air hockey setup, we let the robot perform 8000 actions in total. We evaluate its performance every 1000 actions. For the air hockey experiment, we set the parameters of SGIM-IM to: $cost = 100$ and $ns = 20$. The parameters of SGIM-D are set to $M= 10$ and $M= 100$ which are the best and worst parameters of SGIM-D according to **Figure 4.2.5**.

4.2.2.3 RESULTS

Figure 4.2.5 plots the mean distance error of the attempts to hit the border at the benchmark points, with respect to the number of actions performed by the mallet. It shows that SGIM-IM performs significantly better, and faster than Random exploration or SAGG-RIAC (t-test on the final distance error with $p < 0.05$). It divides by a factor of 10 the final error value compared to SAGG-RIAC. Moreover, its error rate is smaller since the very beginning. SGIM-IM has taken advantage of the demonstrations very fast to be able to hit the puck and place it on the top border, instead of making random movements which have little probability of hitting the puck, let alone placing it at a desired position. Its performance is close to SGIM-D with the best parameters. SGIM-IM manages to tune its percentage of social interaction so as to take most advantage of the demonstrations.

4.2.2.4 ACTIVE CHOICE OF BEHAVIOUR

As for the mode adopted, **Figure 4.2.6** shows 3 phases. In the first part, demonstration requests are useful in the beginning, as each indicate to the learner which kind of actions can make the mallet hit the puck to

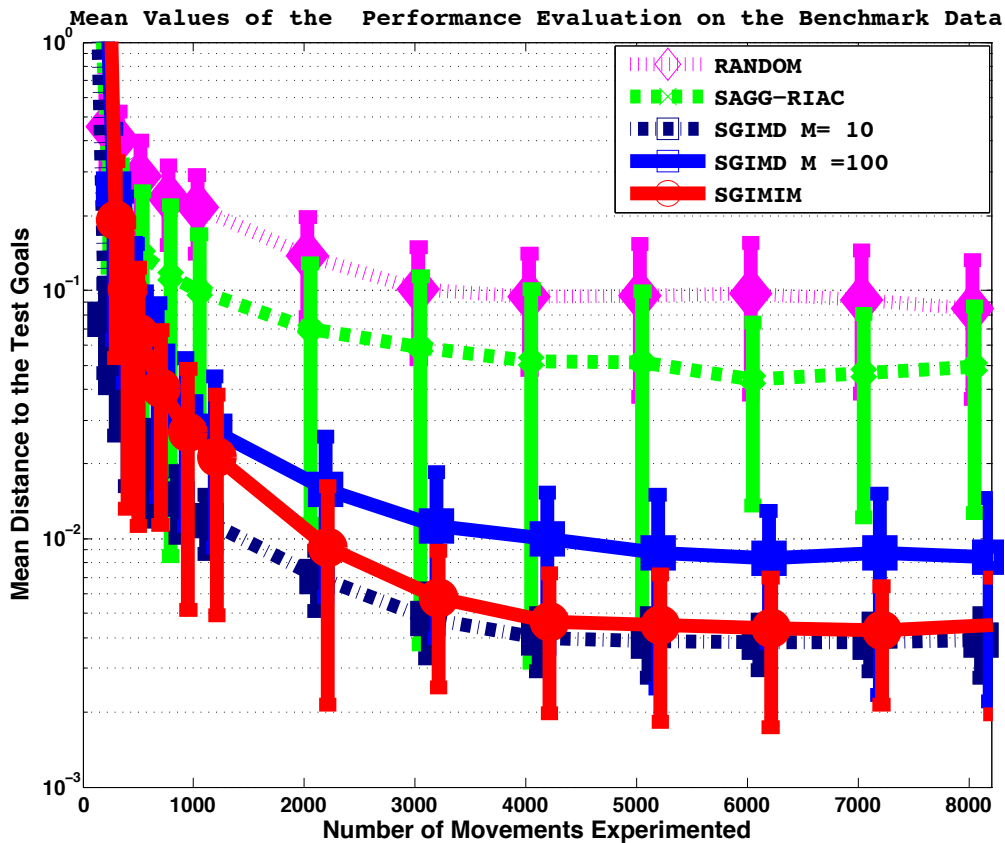


Figure 4.2.5: Evaluation of the performance of the robot with respect to the number of actions performed, under different learning algorithms. We plotted the mean distance to the benchmark set with its standard deviation errorbar.

place it in A and induce a high competence progress value. $\Delta_S \gg \Delta_A$, but autonomous learning still makes good progress. As the progress of autonomous learning decreases, the number of requests for demonstrations increase for $1500 < t < 4000$. In the second part, the progress by the socially guided exploration mode decreases and varies like the progress of autonomous learning, $\Delta_S \approx \Delta_A$. The bootstrapping effect enabled by demonstrations has decreased. Therefore, preference for autonomous exploration increases.

In this experimental setting, the learner can quickly improve its performance by a combination of demonstrations and autonomous exploration. The demonstrations first bootstrap autonomous learning, thus demonstrations are preferred to self-exploration. In the end, as requests for demonstrations no longer help improve the robot’s skill, the learner prefers to improve its learning by intrinsic motivation. The SGIM-IM learner shows an **improvement in both the decrease of the final error value, and the speed of learning**, in this bounded and deterministic environment. Let us illustrate SGIM-IM in a stochastic environment.

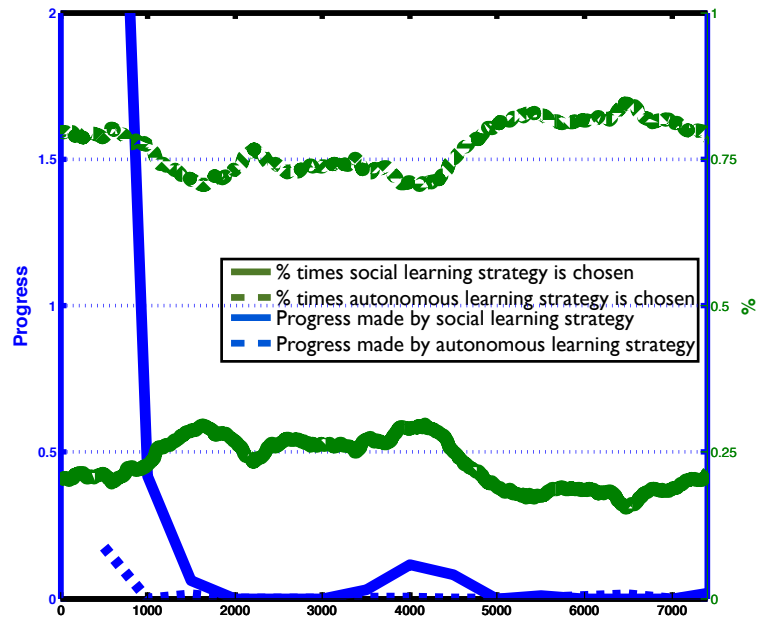


Figure 4.2.6: 1/ Behaviours chosen through time by SGIM-IM: percentage of times each mode is chosen with respect to the number of actions performed (summed over 100 bins and averaged over several runs of SGIM-IM) 2/ The average progress made by socially guided and intrinsically motivated modes Δ_S and Δ_A

4.2.3 FISHING EXPERIMENT

4.2.3.1 EXPERIMENTAL SETUP

In this second experiment, we consider a simulated 6 degrees-of-freedom robotic arm holding a fishing rod (**Figure 4.2.7**) which we used in section 4.2 of chapter 4. We sum up briefly the main points of this experimental setup.

The aim is that it learns how to reach any point on the surface of the water with the float at the tip of the fishing line. $A = [-1, 1]^2$ is a 2-D space that describes the position of the float when it reaches the water. The robot base is fixated at $(0,0)$. The actions are parametrized motor primitives defined for each joint by 4 scalar parameters that represent the joint positions at $t = 0$, $t = \frac{\eta}{3}$, $t = \frac{2\eta}{3}$ and $t = \eta$. These 4 parameters b_1, b_2, b_3, b_4 generate a trajectory for the joint by Gaussian distance weighting. Therefore a set of 24 parameters determine the movement of the 6 joints of the robot. A detailed analysis of this simulation environment can be found in section 4.2.

4.2.3.2 RESULTS

(I) **PRECISION IN THE EXPLORATION OF THE REACHABLE SPACE** We run the simulation of the simulation environment. For every simulation on the fishing experiment setup, 5000 movements are performed. Human demonstrations are taken from demonstrator 3 of section 3.3.3 and assessing how close the robot can reach the benchmark set defined in section 3.3.2 by measuring error every 1000 movements. Our SGIM-IM learner parameters are set to: $cost = 2$ and $ns = 15$.

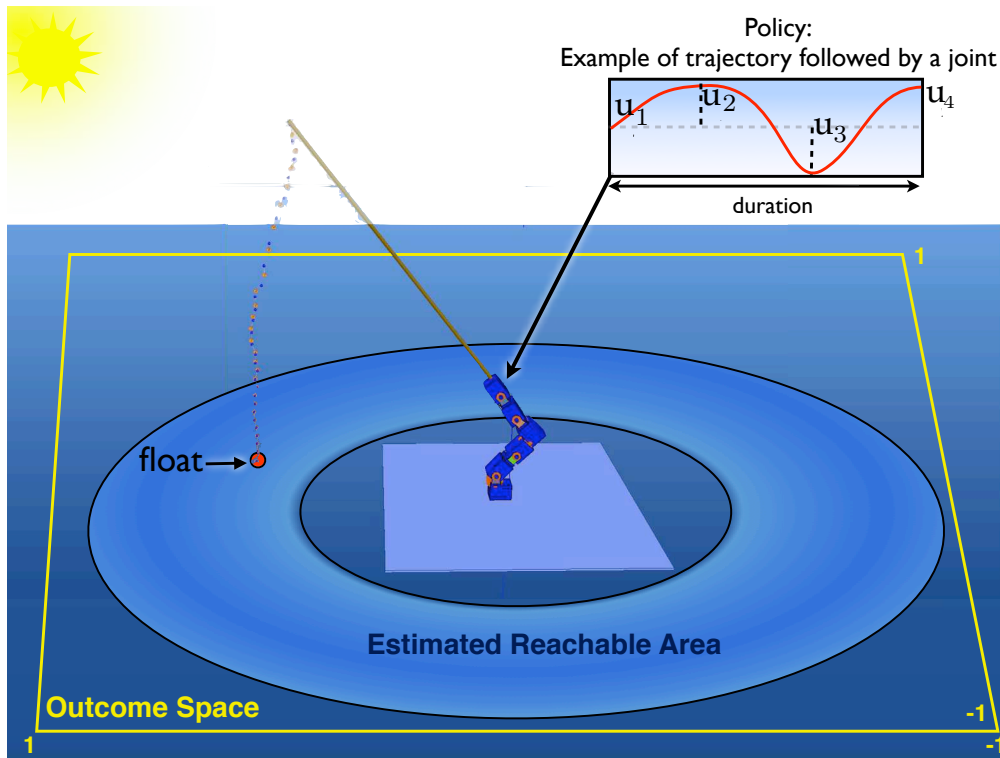


Figure 4.2.7: Fishing experimental setup.

We examine how close the learner can get to any point of the reachable space in A , with respect to the number of actions performed by the robot (**Figure 4.2.8**), and with respect to the number of demonstrations given by the teacher (**Figure 4.2.9b**).

RANDOM performs the worst, while SAGG-RIAC decreases significantly the error value compared to RANDOM (t-test with $p < 0.05$). Not only has the asymptotic performance improved, but SAGG-RIAC also learns faster from the beginning. Requesting demonstrations every 80 actions performed (SGIM-D $M=80$) bootstraps slightly the learning error. In this case, the socially guided exploration mode only makes up 7% of the total time, with 61 demonstrations requests. SGIM-IM performs better than SAGG-RIAC (t-test with $p < 0.05$). The main difference lies in the beginning of the learning process, where it could take advantage of the teacher to guide him and discover the reachable space. With 52 demonstrations requested in average, SGIM-IM yet performs better than SGIM-D($M=80$) with $p < 0.5$, owing to its active choice of mode, that fits better its needs. If we increase the number of demonstrations to 162 (SGIM-D $M=30$), and let the robot adopt the socially guided exploration mode 20% of the time, they indeed efficiently bootstrap the autonomous learning. SGIM-IM manages to request a fair amount of demonstrations and still obtain a performance in between the 2 SGIM-D parameters.

Not only has the error decreased, but the explored space has also increased. **Figure 4.2.9a** plots the histogram of the positions of the float $a \in A$ when it reaches the water. The first column shows that a natural position lies around $a_c = (-0.5, 0)$ in the case of random exploration : most actions map to a region around a_c for the action space does not map linearly to the task space. As the initial position of the float is close to the surface of the water, the robot needs to lift it with quite specific movements to throw it far away, whereas most movements would make the float touch the water immediately, around the region of a_c .

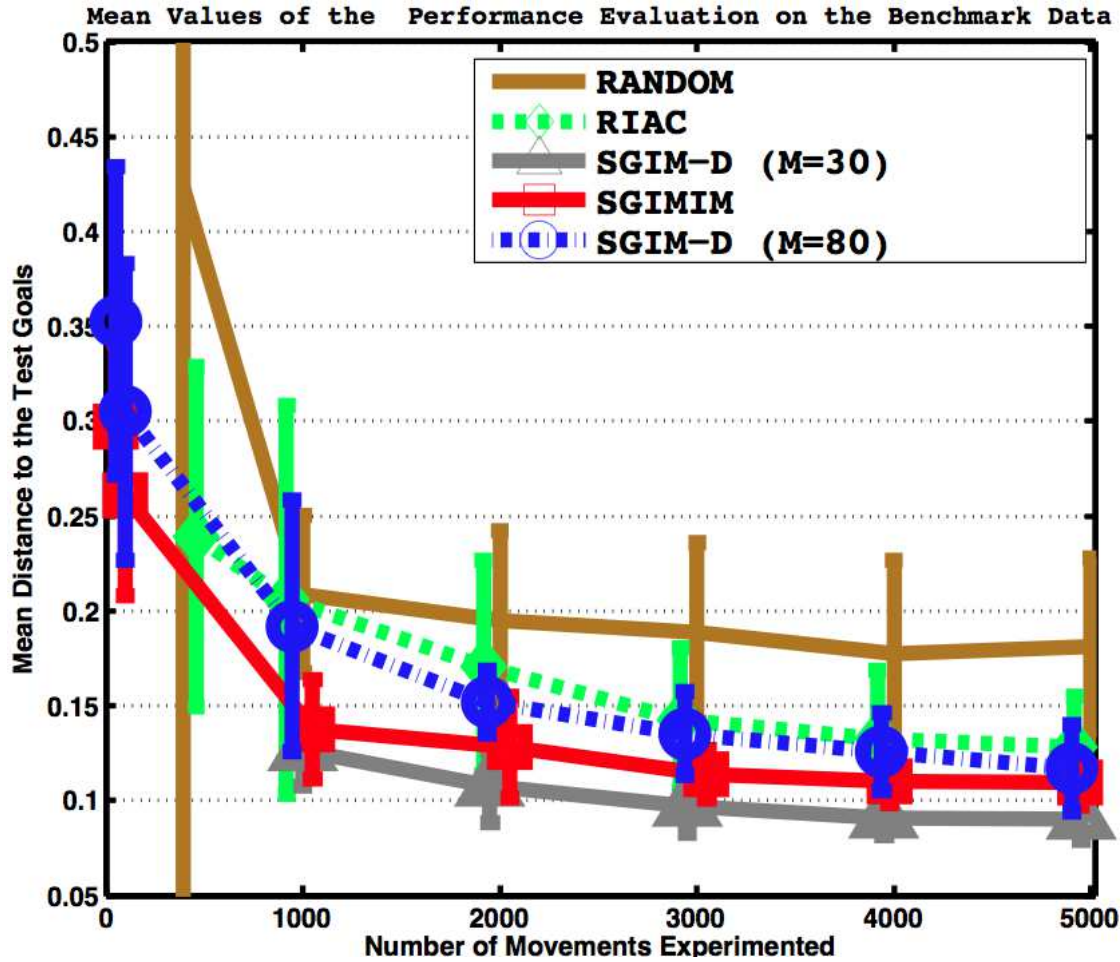


Figure 4.2.8: Evaluation of the performance of the robot with respect to the number of actions performed, under the learning algorithms: random exploration, SAGG-RIAC, SGIM-IM, SGIM-D with a demonstration every $M = 30$ movements, and SGIM-D with a demonstration every $M = 80$ movements (to equal the total number of demonstrations of SGIM-IM). We plotted the mean distance with its standard deviation errorbar.

The second column show that SAGG-RIAC has increased the explored space, and most of all, covers more uniformly the explorable space. SGIM-D and SGIM-IM emphasise the increase even further as a broader range of radius covered in the explored space.

(II) PERFORMANCE OF THE INTERACTION The simple consideration of performance with respect to time spent by the robot must be completed by considerations about the load of work for the teacher. A robot that constantly requests for help would quickly exceed the time and effort a user is ready to devote to teach. Therefore, we must examine the performance of the learner with respect to the number of the demonstrations given. **Figure 4.2.9b** shows that while for the first demonstrations SGIM-IM and SGIM-D($M=80$) perform the same progress, a difference quickly as SGIM-IM requests fewer demonstrations. Each demonstration has a better impact on the performance of the robot, as its error plot in **Figure 4.2.9b** is below the one of SGIM-D.

Indeed, **Figure 4.2.9c** shows that the demonstrations are actively requested in the beginning of the

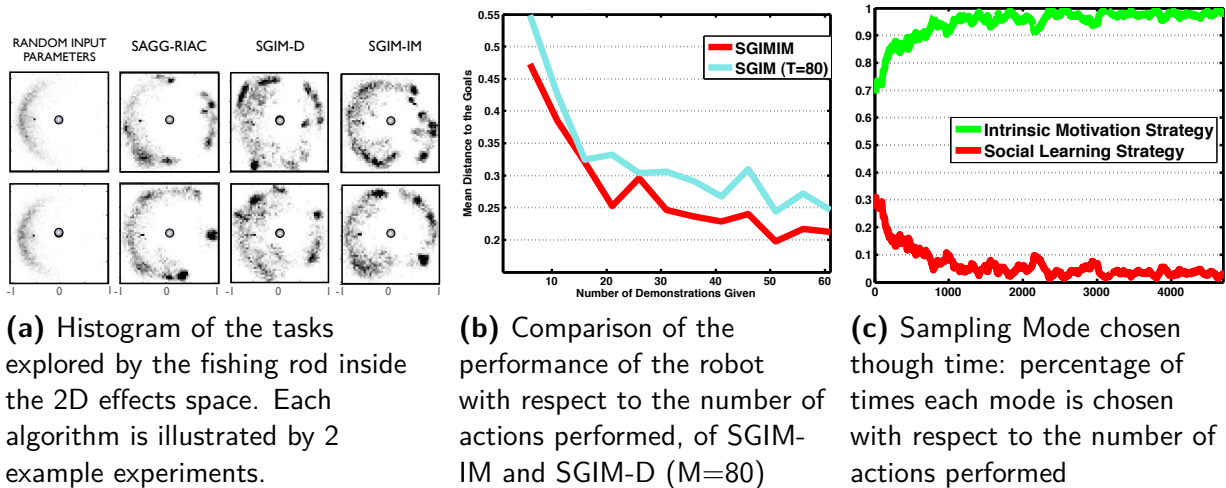


Figure 4.2.9: Analysis of the fishing experiment.

learning process, when the demonstrations enhance the most progress by showing how to avoid the central region around a_c . The requests then decrease as the robot acquires a good knowledge of the explorable space, and can autonomously search around the already explored localities.

In this fishing experiment, the SGIM-IM learner’s active choice of exploration mode enabled it to take advantage of the teacher to request demonstrations, while carefully choosing when the teacher’s demonstrations enhance the most learning progress, in order to lessen its dependence on the teacher.

4.2.4 DISCUSSION AND CONCLUSION

We showed through two illustration experiments that the Socially Guided Intrinsic Motivation with Interactive learning at the Meta level algorithm could learn to complete multiple tasks in both deterministic and stochastic environments. **It can also manage the interaction with both a human teacher whose demonstrations can not be exactly reproduced by him, and a specialised teacher who only gives demonstrations in a restricted subspace of the task space.** In both experiments, our robot learns efficiently and faster all possible tasks, in continuous task and action spaces. The robot could learn high-dimensional models for highly redundant problems, which constitutes a typical issue for humanoid robots who evolve in continuous and non-preset environments and who have to control their numerous degrees of freedom with high redundancy. The **SGIM-IM** learner can handle its interaction with human users owing to interactive learning. **It automatically balances learning by imitation and autonomous learning, by taking in account both its need and the cost of an interaction, so as to minimise the teacher’s effort and maximise the impact of each demonstration.** It thus offers a flexible interaction between a robot and the human users.

The Socially Guided Intrinsic Motivation with Interactive learning at the Meta level algorithm has a 3-layered hierarchical structure which includes two levels of active learning. Based on its exploration in the action space, it actively chooses in the task space which goals could be interesting to target, and selects on a meta level between autonomous intrinsically motivated or socially guided exploration modes. It can actively interact with the teacher instead of being a passive system. This structure could easily be extended to take

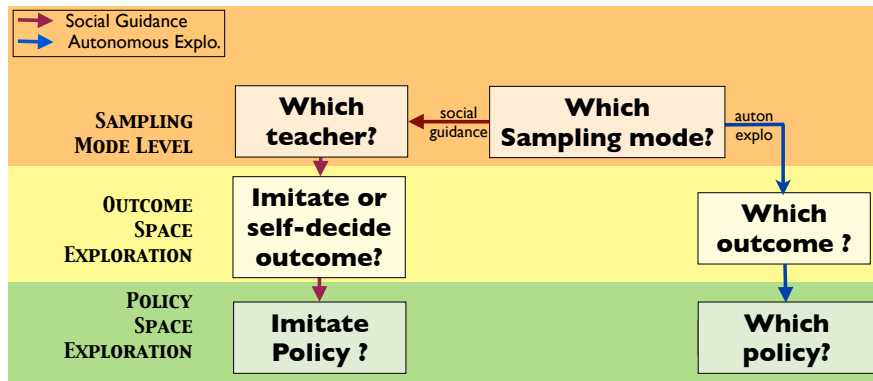


Figure 4.3.1: The strategic learner samples data by actively choosing whether to explore by autonomous exploration or social guidance. If it explores by autonomous exploration, it decides a goal outcome and a policy to try. If it explores with social guidance, it chooses whether to imitate the demonstrated policy or the demonstrated outcome.

into account more complex social interaction scenarios, such as an interaction with several teachers, where the learner can choose who it should imitate. This is the aim of the next section.

4.3 SGIM-ACTS

Previously, we proposed to investigate the relationship between imitation and intrinsically-motivated exploration with a “passive” learner SGIM-D, then with an the interactive learner SGIM-IM which decides when to imitate. We now would like to allow the learner to decide on more questions about its interaction to actively choose at the same time *how, when, what and who* to imitate. These active choices are summarised in **Figure 4.3.1**. The results shown in this section have been presented in (Nguyen and Oudeyer, 2012d).

In this section, we first detail our new problem, then describe the proposed algorithmic architecture, called Socially Guided Intrinsic Motivation with Active Choice of Teacher and Strategy (SGIM-ACTS). Finally, we analyse its mode in an experimental setup where the learner can achieve different types of outcomes.

4.3.1 ACTIVELY LEARNING WHEN, WHO AND WHAT TO IMITATE

4.3.1.1 CHOICE FOR SOCIAL GUIDANCE

To address the four fundamental questions of imitation learning as formulated in (Dautenhahn and Nehaniv, 2002; Breazeal and Scassellati, 2002), we choose among all the possibilities of social guidance detailed in section 1.2.4 of chapter 1:

- **What:** We opted for an information flow targeting both policy and outcome spaces, to enable the biggest progress for the learner. It can imitate to reproduce either a demonstrated policy or outcome. Contrarily to SGIM-D and SGIM-IM which imitate both at the same time without choosing, SGIM-ACTS can decide whether to *mimic* and *emulate* by learning what is the most interesting information. Emulation and Mimicry are illustrated in Figure 4.3.2.

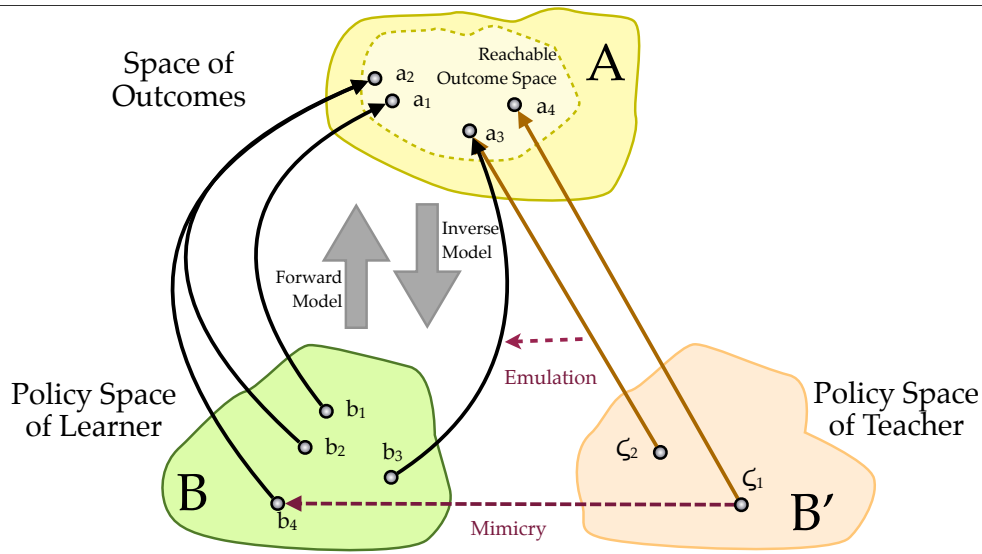


Figure 4.3.2: Emulation and Mimicry for motor learning. In emulation, the learner tries to reproduce the outcome demonstrated by the teacher without trying to reproduce the teacher’s policy, but uses its own movement. In emulation, the learner reproduces the demonstrated policy or movement, without trying to reproduce the demonstrated outcome.

- When: Interactive learning at the *learner’s initiative* seems the most natural interaction approach, the most efficient for learning and less costly for the teacher than if he would have to monitor the learner’s progress to adapt his demonstrations. As with SGIM-IM, the robot has to learn when it is useful to imitate.
- Who: Interactive learning where the learner can *choose who* to interact with and to whom to ask for help, is an important mode choice in learning, and has not been tackled by SGIM-D or SGIM-IM. This aims at addressing problems of suboptimal human inputs which occur because human teachers are generally expert only in specific domains and not on all kinds of skills.

We propose an approach with active learning for varied outcomes with multiple modes, multiple teachers, with a structured continuous outcome space (embedding sub-spaces with different properties). The modes we consider are autonomous self-exploration, emulation and mimicking, by interactive learning with several teachers. Hereafter we describe the design of our **SGIM-ACTS** (Socially Guided Intrinsic Motivation with Active Choice of Teacher and Strategy) algorithm. Then we show through an illustration experiment that SGIM-ACTS efficiently learns to realise different types of outcomes in continuous outcome spaces, and it coherently selects the right teacher to learn from.

4.3.1.2 INTERACTIVE LEARNING BASED ON INTRINSIC MOTIVATION

To solve the problem formalised above, we propose a system, called Socially Guided Intrinsic Motivation with Active Choice of Teacher and Strategy (**SGIM-ACTS**) that allows an online interactive learning of inverse models in continuous high-dimensional robotic sensorimotor spaces with multiple teachers, and exploration

modes. SGIM-ACTS learns various outcomes with different types of outcomes, and generalises from sampled data to continuous sets of outcomes.

Technically, we adopt a method of generalisation of policies for new outcomes similar to (Kober et al., 2012; da Silva et al., 2012). Whereas in their approaches the algorithms use a pool of examples given by the teacher preset from the beginning of the experiment to learn outcomes specified by the engineer of the robot, in a batch learning method; in our case, the SGIM-ACTS algorithm decides by itself which outcomes it needs to learn more to better generalise for the whole outcome space, like in (Oudeyer et al., 2007; Barto et al., 2004b; Baranes and Oudeyer, 2013). Moreover, SGIM-ACTS actively requests the teacher’s demonstrations online, by choosing online the best exploration mode, similarly to (Baram et al., 2004), except that we do not learn with a discrete outcome space for a classification problem, but with a continuous outcome space. SGIM-ACTS also interacts with several teachers and uses several social guidance methods, in an interactive learning approach.

4.3.2 ALGORITHM DESCRIPTION

The proposed algorithm, SGIM-ACTS has already been used in section 2.1 of chapter 2 in the case of discrete variables. In this section, we describe the SGIM-ACTS architecture in the case of motor control in continuous environments by giving a behavioural outline in section 4.3.2.1, before describing its general structure in section 4.3.2.2. We then detail the different functions in sections 4.3.2.3 and 4.3.2.4. The overall architecture is summarised in **Algorithm 4.3.1** and is illustrated in **Figure 4.3.3**.

4.3.2.1 ARCHITECTURE OUTLINE

In this section, based on the formalisation of section 3.1.2 using a single context, we describe the architecture of SGIM-ACTS which merges intrinsically motivated self-exploration with interactive learning for socially guided exploration. In the case of social guidance, a teacher performs an observed trajectory ζ which achieves an outcome a_d . Note that the observed trajectory might be impossible for the learner to re-execute, and he can only approach it best with a policy π_{b_d} .

The agent learns to achieve different types of outcomes by actively choosing which outcomes to focus on and set as goals (ψ), which sampling mode to adopt and to which teacher to ask for help (χ). It learns local inverse and forward models in complex, redundant and continuous spaces.

SGIM-ACTS learns by episodes during which it actively chooses simultaneously an outcome $a_g \in A$ to produce and a exploration mode with a specific teacher (III). Its mode space \mathcal{X} is defined by these preset sampling modes : intrinsically motivated exploration, mimicry from teacher 1, emulation of teacher 1, mimicry from teacher 2, emulation of teacher 2

In an episode with a mimicking mode (red arrows in **Figure 4.3.3** and **Algorithm 4.3.1**, line 7), our SGIM-ACTS learner actively self-generates a goal a_g where its competence improvement is maximal (cf. (III)). The SGIM-ACTS learner explores preferentially goal outcomes easy to reach and where it makes progress the fastest. The selected teacher answers its request with a demonstration $[\zeta_d, a_d]$ to produce an outcome a_d that is closest to a_g (cf. section 3.2.2). The robot mimics the teacher to reproduce ζ_d , for a fixed duration, by performing policies π_b which are small variations of an approximation of ζ_d .

Algorithm 4.3.1 SGIM-ACTS

Input: the different modes $\chi_\alpha, \dots, \chi_\kappa$.
Initialization: partition of outcome space $\mathcal{R} \leftarrow$ singleton A
Initialization: episodic memory (collection of produced outcomes) $\mathcal{H} \leftarrow$ empty memory
Initialization: $e \leftarrow 1$
loop
 $a_i, \chi \leftarrow$ Select Goal Outcome and Sampling Mode(\mathcal{R})
if $\chi =$ Mimic teacher i mode **then**
 $(\zeta_d, a_d) \leftarrow$ ask and observe demonstration to teacher i .
 $\gamma_1 \leftarrow$ Competence for a_g
 $\mathcal{D}_e \leftarrow$ Mimic Action(ζ_d)
 $p_{e+1} \leftarrow \mathcal{L}(p_e, \mathcal{D}_e)$
 $\gamma_2 \leftarrow$ Competence for a_g
else if $\chi =$ Emulate teacher i mode **then**
 $(\zeta_d, a_d) \leftarrow$ ask and observe demonstration to teacher i .
 Emulation: $a_g \leftarrow a_d$
 $\gamma_1 \leftarrow$ Competence for a_g
 $\mathcal{D}_e \leftarrow$ Goal-Directed Policy Optimisation(a_g)
 $p_{e+1} \leftarrow \mathcal{L}(p_e, \mathcal{D}_e)$
 $\gamma_2 \leftarrow$ Competence for a_g
else
 $\chi =$ **Intrinsic Motivation mode**
 $a_g \leftarrow a_i$
 $\gamma_1 \leftarrow$ Competence for a_g
 $\mathcal{D}_e \leftarrow$ Goal-Directed Policy Optimisation(a_g)
 $p_{e+1} \leftarrow \mathcal{L}(p_e, \mathcal{D}_e)$
 $\gamma_2 \leftarrow$ Competence for a_g
end if
 $nbA \leftarrow$ number of episodes in \mathcal{D}_e
 $prog \leftarrow 2(\text{sig}(\alpha_p * \frac{\gamma_2 - \gamma_1}{|T_i| \cdot nbA}) - 1)$
Append \mathcal{D}_e to \mathcal{H}
 $\mathcal{R} \leftarrow$ Update Outcome and Sampling Mode Interest Mapping($\mathcal{R}, \mathcal{H}, a_g, prog, \chi$)
 $e \leftarrow e + 1$
end loop

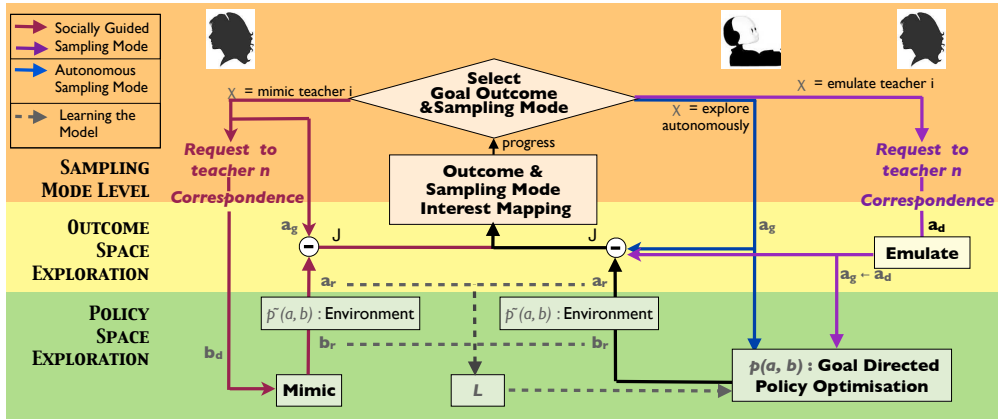


Figure 4.3.3: Time flow chart of SGIM-ACTS, which combines Intrinsic Motivation and Mimicking and Emulation into 3 layers that pertain to the sampling mode, the outcome space and the policy space exploration respectively.

In an episode with an emulation mode (purple arrows in **Figure 4.3.3** and **Algorithm 4.3.1**, line 13), our SGIM-ACTS learner observes from the selected teacher a demonstration $[\zeta_d, a_d]$. It tries different policies using goal-directed optimisation algorithms to approach the observed outcome a_d , without taking into account the demonstrated policy ζ_d . It re-uses and optimises its policy repertoire built through its past autonomous and socially guided explorations (cf. section 3.2.4). The episode ends after a fixed duration.

In an episode with the intrinsic motivation mode (blue arrows in **Figure 4.3.3** and **algorithm 4.3.1**, line 21), it explores autonomously following the SAGG-RIAC algorithm (**Baranes and Oudeyer, 2013**). It actively self-generates a goal a_g where its competence improvement is maximal (cf. section (III)), as in the mimicking mode. Then, it explores which policy π_b can achieve a_g best. It tries different policies to approach the self-determined outcome a_g , as in the emulation mode (cf. section 3.2.4). The episode ends after a fixed duration. The intrinsic motivation and emulation modes differ mainly by the way the goal outcome is chosen.

An extensive study of the role of these different exploration modes can be found in section 3.5 of chapter 3. Thus the mimicry exploration increases the learner’s policy repertoire on which to build up emulation and self-exploration, while biasing the policy space exploration. Demonstrations with structured policy sets, similar policy shapes, bias the policy space exploration to interesting subspaces, that allow the robot to overcome high-dimensionality and redundancy issues and interpolate to generalise in continuous outcome spaces. With emulation learning, the teacher influences the exploration of the outcome space. He can hinder the exploration of subspaces attracting the learner’s attention to other subspaces. On the contrary, he can encourage their exploration by making demonstrations in those subspaces. Self-exploration is essential to build up on these demonstrations to overcome correspondence problems and collect more data to acquire better precision according to the embodiment of the robot.

This modal description of SGIM-ACTS is followed in the next section by the description of its architecture.

4.3.2.2 HIERARCHICAL STRUCTURE

SGIM-ACTS improves its estimation of the inverse model in robot control $p(b|a)$ to minimise $I = \int_A p(a)J(a, \tilde{p}(a|\arg\max_b(p(b|a))))da$ by exploring with the different sampling modes the outcome and policy

spaces. Like SGIM-IM, its architecture is separated into three layers:

- A *Sampling Mode Exploration* level. Like in SGIM-IM, this meta-level layer decides on the data collection strategy of the agent based on the feedback from the other layers. In SGIM-ACTS, it decides on more aspects of its social interaction. It actively chooses which exploration sampling mode χ to use between intrinsic motivation, emulation and mimicry, and which teacher to ask for demonstrations (*Select Goal Outcome and Sampling Mode*). In a goal-oriented approach, it maps A in terms of interest level for each sampling mode (*Outcome and Sampling Mode Interest Mapping*) to keep track which sampling mode and which subspace of A leads to the best learning progress.
- An *Outcome Space Exploration* level which minimises I by exploring A . It decides actively where to focus in the outcome space. More precisely it chooses for which outcome a_g to to minimise $J(a_g, \operatorname{argmax}_b(p(b|a_g)))$ according to the adopted sampling mode χ . In the case of an emulation sampling mode, it sets the observed outcome of the demonstration a_d as a goal. In the case of mimicry and intrinsic motivation sampling modes, it self-determines a goal a_g selected by the *Select Goal Outcome and Sampling Mode* function.
- A *Policy Space Exploration* level which explores the policy parameters space B to achieve a_g or a_d set by the *Outcome Space Exploration* layer, and gets a better estimate of the forward model to build its control model. More formally, it improves its estimation of $J(a_g, \tilde{p}(a_g|b))$ and thus build its control policy $p(b|a_g)$. With the mimicry exploration sampling mode, it mimics the demonstrated trajectory ζ_d by the chosen teacher to estimate $\tilde{p}(a, b)$ around the locality of ζ_d (*Mimicry*). With the emulation and autonomous exploration sampling mode, the *Goal-Directed Policy Optimisation* function minimises $J(a_g, \tilde{p}(a_g|b))$ with respect to b . It finally returns to the Sampling Mode and Outcome Space Exploration level the measure of competence progress for reaching a_g or a_d .

The exploration in the three levels is the key to the robustness of SGIM-ACTS in high dimensional policy spaces.

4.3.2.3 POLICY SPACE EXPLORATION

The policy space exploration is carried out by two functions, the *Mimic a Policy* and *Goal-Directed Policy Optimisation* functions. They have been detailed in sections 3.2.2 (page 52) and 3.2.4 (page 53).

4.3.2.4 SAMPLING MODE AND OUTCOME SPACE EXPLORATION

The Sampling Mode and Outcome Space Exploration of SGIM-ACTS bears high resemblance with SGIM-IM. However, the choice of sampling mode and of goal outcome has been unified under a same decision process by mapping the Outcome and Sampling Mode Spaces in terms of interest. It also has a new function, *Emulation*, which implements a new sampling mode.

(I) **EMULATION** In the emulation sampling mode, the learner explores outcomes a_d that he observed from the demonstrations: $a_g \leftarrow a_d$. The learner tries to achieve a_d by goal-oriented policy optimisation, which allows data collection and updating of the control model $p(b|a_d)$.

(II) **OUTCOME AND SAMPLING MODE INTEREST MAPPING** A is partitioned according to interest levels. We note $\mathcal{R} = \{R_i, A = \cup_i R_i\}$ a partition of A . For each outcome a explored with sampling mode χ , the learner evaluates its competence progress, where competence measure assesses how close it can reach a : $\gamma_{a,\chi} = \min_{(A,b) \in \mathcal{H}} J(a,b)$. A high value of $\gamma_{a,\chi}$ means a good competence at reaching the goal a by sampling mode χ .

For each episode, it can compute its competence for the goal outcome at the beginning of the episode $\gamma_{a,\chi}^1$ and the end of the episode $\gamma_{a,\chi}^2$ after trying nbA movements and measure its competence progress:

$$prog = 2(sig(\alpha_p * \frac{\gamma_{a,\chi}^1 - \gamma_{a,\chi}^2}{|A_i| \cdot nbA}) - 1) \text{ with } sig(x) = \frac{e^x + e^{-x}}{2} \quad (4.2)$$

where α_p is a constant and $|A_i|$ is the size of the subspace A_i .

A is partitioned so as to maximally discriminate areas according to their competence progress, as described in **Algorithm 4.3.2** and (Baranes and Oudeyer, 2013). For each sampling mode χ , we define a cost $\kappa(\chi)$, which are weights for the computation of the interest of each region of the outcome space. $\kappa(\chi)$ represents the preference of the teachers to help the robot or not, or the cost in time and energy ... of each sampling mode, and in this study $\kappa(\chi)$ are set to arbitrary constant values.

We compute the interest as *the local competence progress, over a sliding time window of the δ most recent goals attempted inside R_i with sampling mode χ* which builds the list of competence progress measures $R_i(\chi) = \{progress_1, \dots, progress_{|R_i(\chi)|}\}$:

$$interest_{R_i}(\chi) = \frac{mean_{j=|R_i(\chi)|-\delta}^{|R_i(\chi)|} progress_j}{\kappa(\chi)} \quad (4.3)$$

The partition of A is done recursively and so as to maximally discriminate areas according to their levels of interest. A split is triggered once a number of outcomes g_{max} has been attempted inside R_n with the same mode χ (**Algorithm 4.3.2**, line 12). The split separates areas of different interest levels and different reaching difficulties (line 13). The split of a region R_n into R_{n+1} and R_{n+2} is done by selecting among m randomly generated splits, a split dimension $j \in |A|$ and then a position v_j (we suppose that $R_n \subset A_i \subset A$ with A_i a n -dimensional space) such that:

- All the $a \in R_{n+1}$ have a j th component smaller than v_j ;
- All the $a \in R_{n+2}$ have a j th component higher than v_j ;
- It maximises the quantity $Qual(j, v_j) = |R_{n+1}| \cdot |R_{n+2}| |interest_{R_{n+1}(\chi)} - interest_{R_{n+2}(\chi)}|$, where $|R_i|$ is the size of the region R_i ;

(III) **SELECT GOAL OUTCOME AND SAMPLING MODE** In order to balance exploitation and exploration, the next goal outcome and mode are selected according to one of the 3 modes, chosen stochastically with respectively probabilities p_1 , p_2 and p_3 :

Algorithm 4.3.2 $[\mathcal{R}] = \text{Update Outcome and Sampling Mode Interest Mapping}(\mathcal{R}, \mathcal{H}, a_g, progress_g, \chi)$

input: \mathcal{R} : set of regions R_n and corresponding $interest_{R_n}(\chi)$ for each sampling mode χ .
input: $a_g, progress_g$: goal outcome of the episode and its progress measure.
input: \mathcal{H} : the set of all observed outcomes during the episode and their progress measures $(a_r, progress_r)$.

input: χ : sampling mode and teacher used during the episode.
parameter: g_{Max} : the maximal number of elements inside a region.
parameter: δ : a time window used to compute the interest.
for all $(a, progress) \in \{\mathcal{H}, (a_g, progress_g)\}$ **do**
 Find the region $R_n \in \mathcal{R}$ such that $a \in R_n$.
 Add $progress$ in $R_n(\chi)$, the list of competence progress measures of experiments $a \in R_n$ with sampling mode χ .
 Compute the new value of competence progress of $R_n(\chi)$:

$$interest_{R_n}(\chi) = \frac{mean_{i=|R_n|-\delta}^{|R_n|} progress_i}{\kappa(\chi)}$$

if $|R_n(\chi)| > g_{max}$ **then**
 $\mathcal{R} \leftarrow \text{Split } R_n$.
end if
end for
return \mathcal{R}

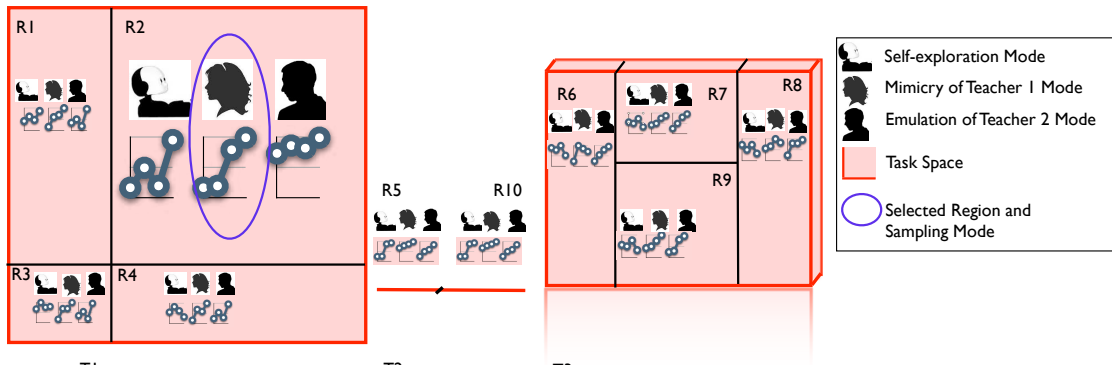


Figure 4.3.4: The selection of outcome and sampling mode is based on a partition of the outcome space with respect to different competence progress levels. We illustrate with the case of an outcome space of 3 different types of outcomes. $A = A1 \cup A2 \cup A3$ where $A1 \subset \mathbb{R}^2$, $T2 \subset \mathbb{R}$ and $T3 \subset \mathbb{R}^3$. A is partitioned in regions R_i to which are associated measures of competences γ for each mode. The "Select Goal Outcome and Sampling Mode" function chooses the (region, mode) pair that makes the most competence progress.

- mode 1: choose χ and $a \in A$ randomly. It ensures a minimum of exploration of the full mode and outcome spaces.
- mode 2: choose the region $R_n(\chi)$ and thus the mode χ with a probability proportional to its interest value $interest_{R_n}(\chi)$:

$$P_n(\chi) = \frac{interest_{R_n}(\chi) - \mathbf{min}(interest_{R_i})}{\sum_{i=1}^{|R_n|} interest_{R_i}(\chi) - \mathbf{min}(interest_{R_i})} \quad (4.4)$$

An outcome a is then generated randomly inside R_n . This mode uses exploitation to choose the region with highest interest measure.

- mode 3: the mode and regions are selected like in mode 2, but the outcome $a \in R_n$ is generated close to the already experimented one which received the lowest competence estimation. This mode also uses exploitation to choose the best outcome and mode with respect to interest measures.

We illustrate in the following section this hierarchical algorithm through an illustration example where a robot learns to throw a ball or to place it at different angles with 7 sampling modes: intrinsically motivated exploration, mimicry from 3 teachers and emulation from 3 teachers.

4.3.3 THROWING AND PLACING A BALL

4.3.3.1 EXPERIMENTAL SETUP

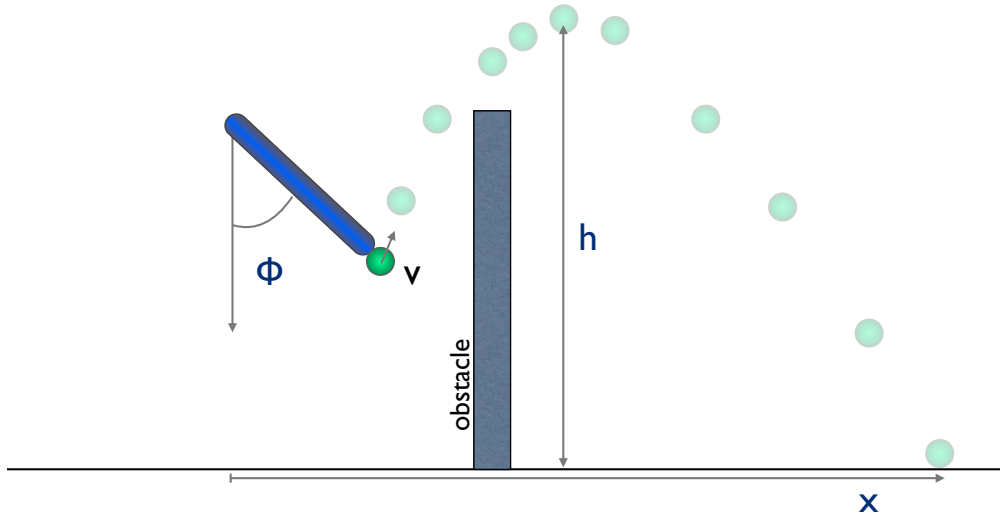


Figure 4.3.5: An arm, described by its angle ϕ , is controlled by a motor primitive with 14 continuous parameters (taking bounded values) that determine the evolution of its acceleration $\ddot{\phi}$. A ball is held by the arm and then released at the end of the motion. The objective of the robot is to learn the mapping between the parameters of the motor primitive and two types of outcomes he can produce: a ball thrown at distance x and height h , or a ball placed at the arm tip at angle ϕ with velocity smaller than $|v_{max}|$.

In our simulated experimental setup, we have a 1 degree-of-freedom arm place a ball at different angles or throw the ball by controlling its angular acceleration $\ddot{\phi}$ (**Figure 4.3.5**). The time evolution of its angular acceleration is described with motor primitives determined by 14 parameters. $B \subset \mathbb{R}^{14}$ as described in paragraph (I). The outcome space is composed of 2 types of outcomes $A = A1 \cup A2$, that we detail in paragraph (II) and paragraph (III).

(I) **POLICY PARAMETER SPACE** Starting from angle $\phi = 0$, the robot can control its angular acceleration $\ddot{\phi}$. Its movement is parameterised by $b = (\ddot{\phi}_1, t_1, \dots, \ddot{\phi}_7, t_7)$ which defines the acceleration of the arm for the 7 durations t_i . It thus defines $\ddot{\phi}(t)$ as a piecewise constant function. The policy parameter space B is arbitrarily set to a 14 dimensional space.

(II) **THROWING OUTCOMES** The first type of outcomes is the different distance x and height h at which the ball M can be thrown. $A1 = \{(x, h)\}$ is a continuous space of dimension 2. The ball, initially in the robot's hand is first accelerated by the robot arm, and then automatically released:

- at position $\vec{O}\vec{M}_{t=0}$ which is the position of the tip of the arm,
- with velocity $\frac{d\vec{O}\vec{M}}{dt}_{t=0}$ which magnitude is the velocity of the arm, and which direction is the tangent of the arm movement.

Then, the ball falls under gravity force, described by the equation:

$$\vec{O}\vec{M}_t = \frac{\vec{g}}{2} \cdot t^2 + \frac{d\vec{O}\vec{M}}{dt}_{t=0} \cdot t + \vec{O}\vec{M}_{t=0}, \quad (4.5)$$

where \vec{g} is the gravity force. x is therefore computed for t_{impact} , the time when the ball touches the ground, or in other words the solution to the 2nd polynomial equation:

$$\frac{-g}{2} \cdot t^2 + \frac{dz}{dt}_{t=0} \cdot t + z_{t=0} = 0 \quad (4.6)$$

The maximum height is also directly computed by equation:

$$h = z_{t=0} + \frac{(\frac{dz}{dt}_{t=0})^2}{2g}; \quad (4.7)$$

To make the throwing less trivial, we also added a wall as an obstacle at $x= 10$. The ball can bounce on the wall using an immobile wall model and elastic collision.

(III) **PLACING OUTCOMES** The second type of outcomes is placing a ball at different angles ϕ . Therefore $T2$ is of dimension 1. To achieve an outcome in $T2$, the robot has to stop its arm in a direction ϕ before releasing the ball, i.e. it learns to reach ϕ at a small velocity $|v| < |v_{max}|$.

Any policy would move the arm to a final angle ϕ , but to "place" the ball at an angle, it also needs to reach a velocity smaller than $|v_{max}|$. Therefore placing a ball is difficult.

The robot learns which arm movement it needs to perform to either place at a given angle ϕ or to throw a ball at a given height and distance. Mathematically speaking, it learns highly redundant mappings between a 14-dimensional policy space and a union of a 1D and a 2D continuous outcome spaces.

In our experimental setup, the outcome space is thus the union of two continuous spaces of different dimensionalities, related to throwing and placing skills, which makes it complex because of the continuous and composite nature of the space. The complexity of the placing of the ball depends on the physics of the body and on the structure of motor commands. We choose to control the robot by angular acceleration to emphasise the difference in the ease of control between the "throwing outcomes" which require rather a velocity control, and the "placing outcomes" which require rather a position control. Given the motor control by acceleration and the encoding of motor primitives, the placing outcomes are thus more difficult to achieve than the throwing outcomes.

4.3.3.2 SEVERAL TEACHERS AND SAMPLING MODES

We create simulated teachers by building 3 demonstration sets from which to pick a random demonstration when asked by the learner :

- teacher 1 has learned how to throw a ball with SAGG-RIAC. The teacher 1 has the same motor primitives encoding as the learner, and the robot observes from the demonstrated trajectories directly the demonstrated $b_d = (\ddot{\phi}_1, t_1, \dots, \ddot{\phi}_7, t_7)$.
- teacher 2 is an expert in placing, programmed by an explicit equation to place at any angle with a null velocity. The teacher 2 too has the same motor primitives encoding as the learner, and the robot observes from the demonstrated trajectories directly the demonstrated $b_d = (\ddot{\phi}_1, t_1, \dots, \ddot{\phi}_7, t_7)$.
- teacher 3 is an expert in placing, except that in this case the learner faces correspondence problems and misinterprets the two parameters $\ddot{\phi}_6$ and $\ddot{\phi}_7$ as the opposite values. In this experiment, we do not attempt to solve this correspondence problem. We also note that while the learner has issues mimicking teacher 3, he has no issues emulating teacher 3, as the outcome space parametrisation is the same.

Therefore in our experiment, the interactive learner can choose between the 7 sampling modes of \mathcal{X} : SAGG-RIAC autonomous exploration, emulation of each of the 3 teachers or mimicry of each of the 3 teachers.

4.3.3.3 COMPARISON OF LEARNING ALGORITHMS

To assess the efficiency of SGIM-ACTS, we decide to compare the performance of several learning algorithms (**Figure 4.3.6**):

- Random exploration : throughout the experiment, the robot learns by picking policy parameters randomly. It explores randomly the policy parameter space B .
- SAGG-RIAC : throughout the experiment, the robot uses active goal-babbling to explore autonomously, without taking into account any demonstration by the teacher, and is driven by intrinsic motivation.
- mimicry : at a regular frequency, the learner determines a goal a_g where learning progress is maximal, and requests to the chosen teacher a demonstration. The teacher selects among his data set a

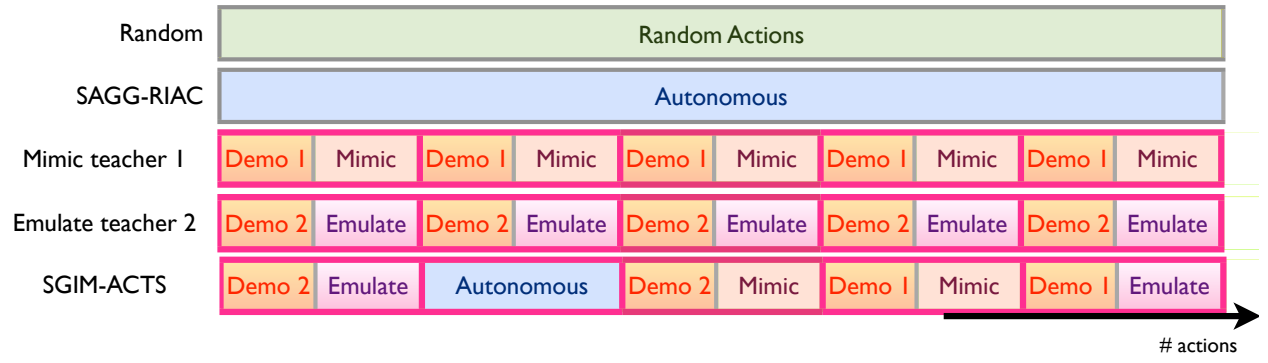


Figure 4.3.6: Comparison of several learning algorithms

demonstration $[\zeta_d, a_d]$ so that $a_d = \operatorname{argmin}_{a \in \{DemoSet\}} \|a_g - a\|$. The learner mimics the demonstrated policy ζ_d by repeating the movement with small variations.

- emulation : at a regular frequency, the learner determines a goal a_g where learning progress is maximal, and requests to the chosen teacher a demonstration. The teacher selects among his data set a demonstration $[\zeta_d, a_d]$ so that $a_d = \operatorname{argmin}_{a \in \{DemoSet\}} \|a_g - a\|$. The learner tries to reproduce the outcome a_d .
- SGIM-ACTS : interactive learning where the robot learns by actively choosing between intrinsic motivation mode or one of the socially guided exploration modes with the chosen teacher: mimicking or emulation.

We run simulations with the following parameters. The costs of all socially guided modes $\kappa(\chi)$ are set to 2, and the cost of intrinsic motivation is set to 1. The probabilities for the different modes of selecting a region of the outcome space and a mode are: $p_1 = 0.05$, $p_2 = 0.7$ and $p_3 = 0.25$. Other parameters are $\epsilon = 0.05$, $g_{max} = 10$, $\alpha_p = 1000$ and $v_{max} = 0.01$.

For each experiment, we let the robot perform 8000 actions in total, and evaluate its performance every 1000 actions, by requiring the system to produce outcomes from a benchmark set that is evenly distributed in the outcome space and independent from the learning data.

4.3.3.4 RESULTS

The comparison of these four learning algorithms in **Figure 4.3.7** shows that SGIM-ACTS decreases its cumulative error for both placing and throwing. It performs better than autonomous exploration by random search or intrinsic motivation, and better than any socially guided exploration with any teacher. **Figure 4.3.8** details that SGIM-ACTS error rate for both placing and throwing is low. For throwing, SGIM-ACTS performs the best in terms of error rate and speed because it could find the right sampling mode. We also note that random exploration and SAGG-RIAC also perform well for solving the 2nd degree polynomial **Equation (4.5)** to achieve throwing outcomes. While mimicking and emulating teacher 1 decreases the

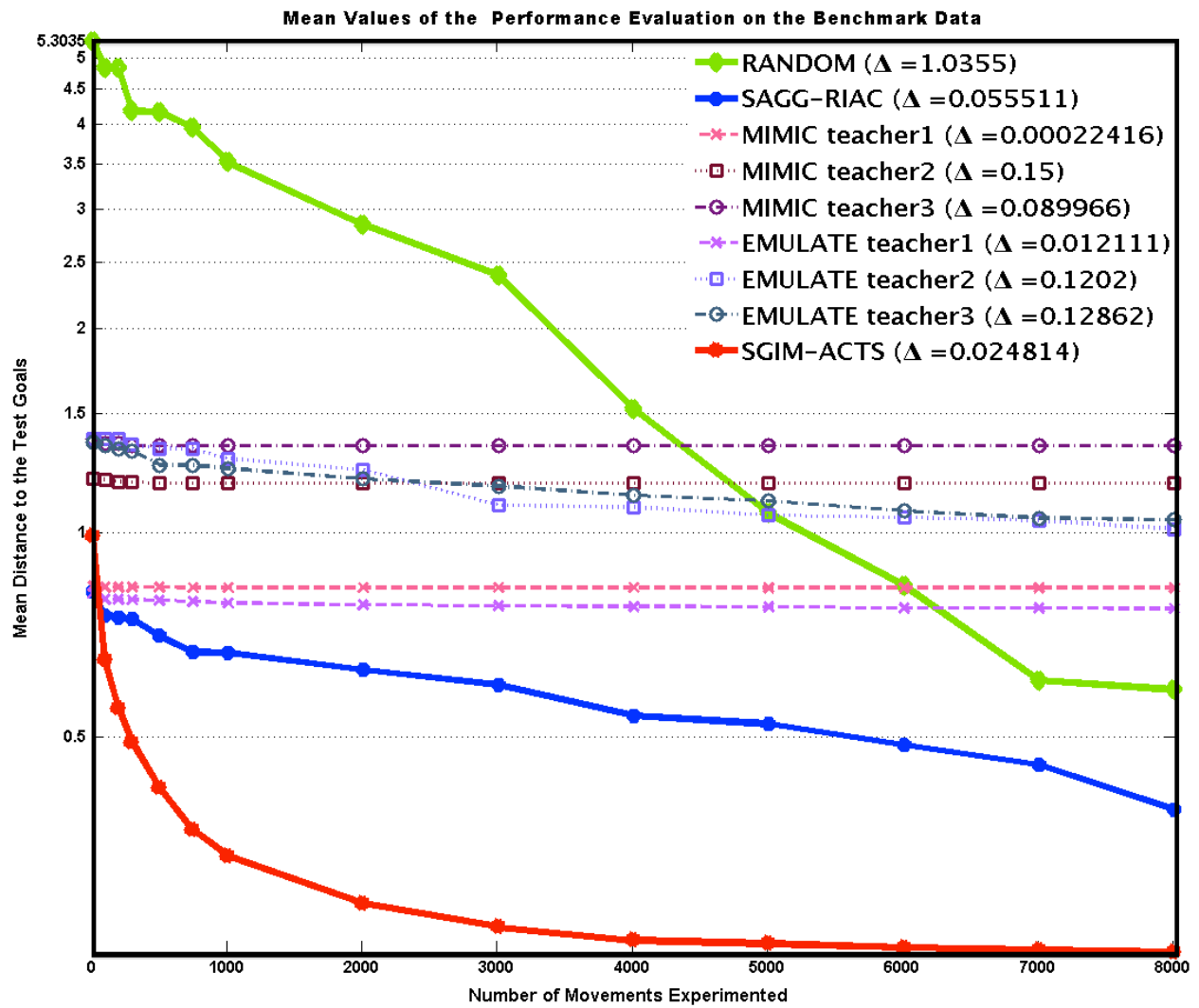


Figure 4.3.7: Mean error for the different learning algorithms averaged over the two sub outcome spaces (final variance value Δ is indicated in the legend) .

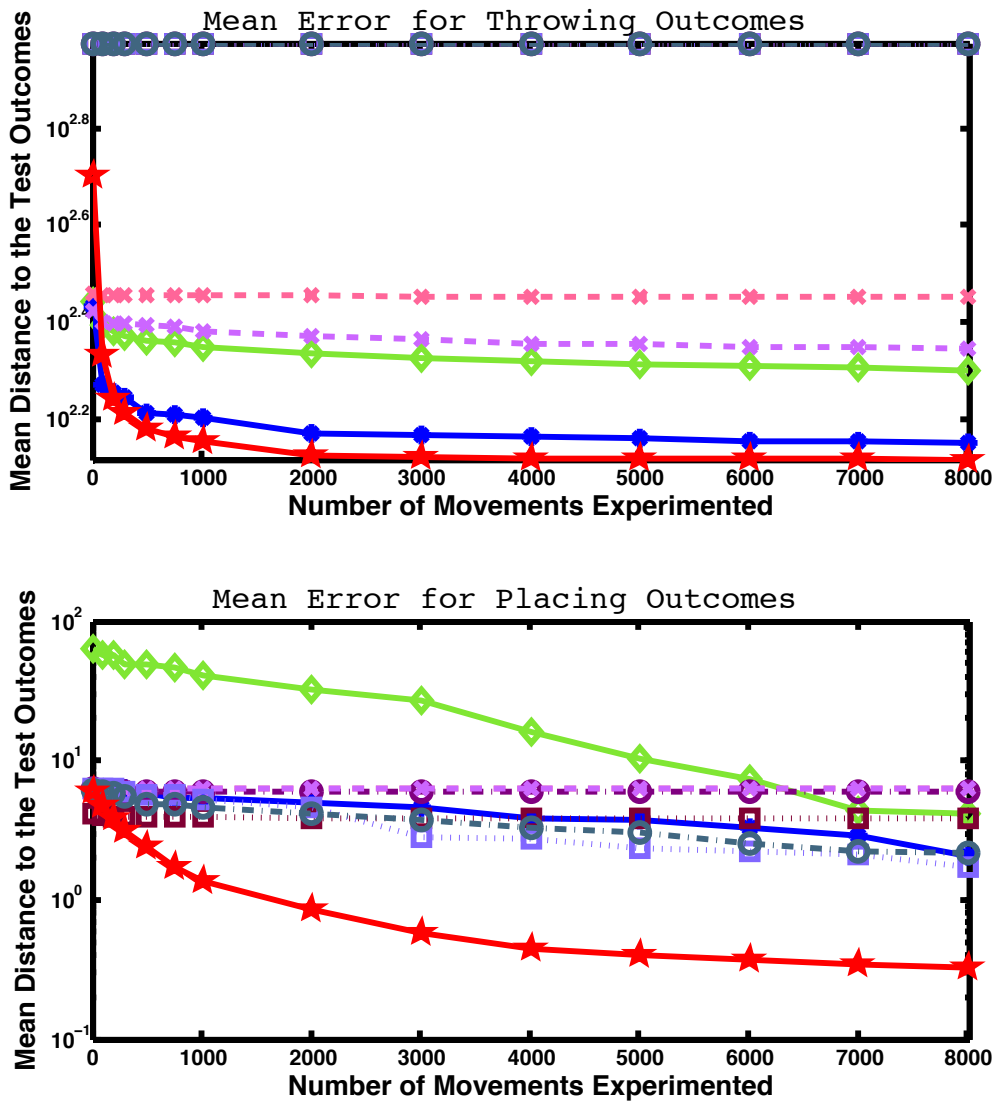


Figure 4.3.8: Mean error for the different learning algorithms for each of the throwing outcomes and placing outcomes separately. The legend is the same as in **Figure 4.3.7**.

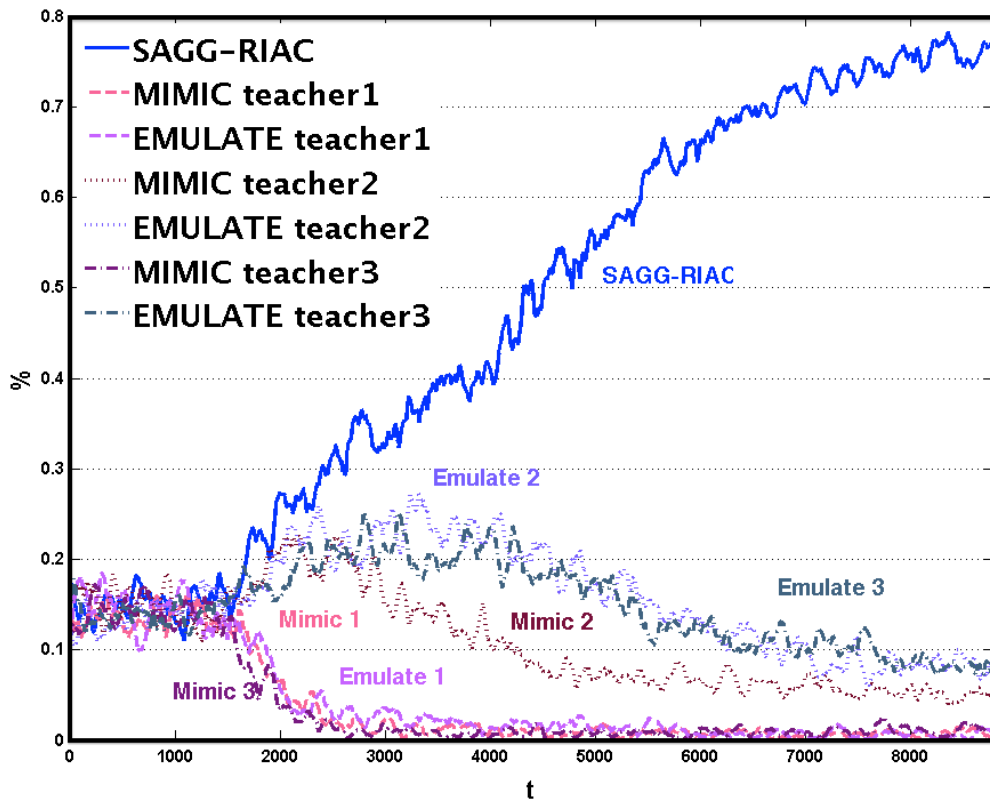


Figure 4.3.9: Sampling Mode chosen by SGIM-ACTS through time: percentage of times each mode is chosen for several runs of the experiment.

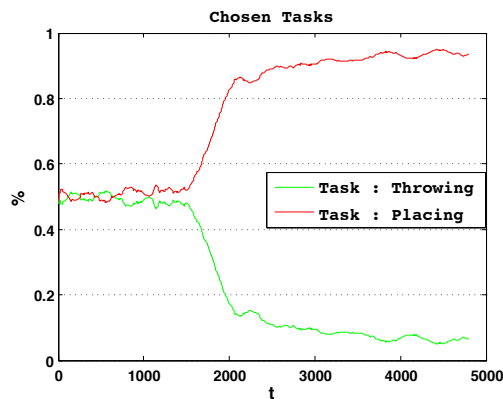


Figure 4.3.10: Types of outcome chosen by SGIM-ACTS through time: percentage of times each kind of outcome is chosen for several runs of the experiment.

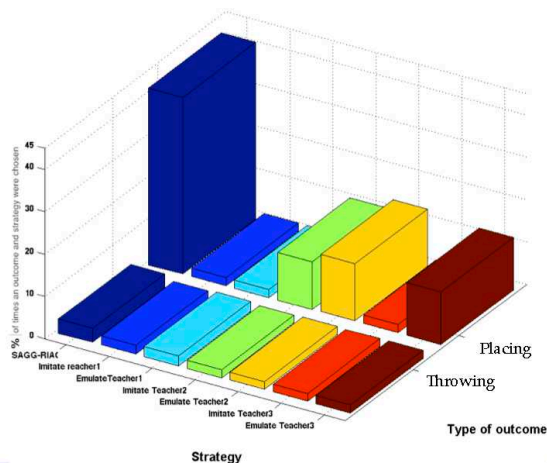


Figure 4.3.11: Consistency in the choice of outcome, teacher and mode: percentage of times each sampling mode, teacher and outcome are chosen over all the history of the robot.

error as expected, mimicking and emulating a teacher who is expert in another kind of outcomes and is bad in that outcome leaves a high error rate. For placing, SGIM-ACTS makes less error than all other algorithms. Indeed, as we expected, mimicking the teacher 2, and emulating teachers 2 and 3 enhances low error rates, while mimicking a teacher with correspondence problem (teacher 3) or an expert on another outcome (teacher 1) gives poor result. We also note that for both outcomes, mimicry does not lead to important learning progress, and the error curve is almost flat. This is due to the lack of exploration which leads the learner to ask demonstrations for outcomes only in a small subspace.

Indeed, we see in **Figure 4.3.9** which illustrates the percentage times each mode is chosen by SGIM-ACTS with respect to time, that mimicry of teacher 3, which lacks efficiency because of the correspondence problem, is seldom chosen by SGIM-ACTS. Mimicry and emulation of teacher 1 is also little used because autonomous learning learns quickly throwing outcomes. Teachers 2 and 3 are exactly the same with respect to the outcomes they demonstrate, and are emulated in the same proportion. This figure also shows that the more the learner cumulates knowledge, the more autonomous he grows : his percentage of autonomous learning increases steadily.

Not only does he choose the right sampling modes, but also the right outcome to concentrate on. **Figure 4.3.10** shows that he concentrates in the end more on placing, which are more difficult.

Finally, **Figure 4.3.11** shows the percentage of times over all the experiments where he chooses at the same time each outcome type, a sampling mode and a teacher. We can see that for the placing outcomes, he seldom requests help from the teacher 1, as he learns that teacher 1 does not know how to place the ball. Likewise, because of the correspondence problems, he does not mimic teacher 3. But he learns that mimicking teacher 2 and emulating teachers 2 and 3 are useful for placing outcomes. For the throwing outcomes, he uses slightly more the autonomous exploration mode, as he can learn efficiently by himself. The high percentage for the other modes is due to the fact that the throwing outcomes are easy to learn, therefore are learned in the beginning when a lot of sampling of all possible modes is carried out. SGIM-ACTS is therefore consistent in its choice of outcomes , sampling modes and teachers.

4.3.3.5 CONCLUSION AND DISCUSSION

We presented the **SGIM-ACTS** (Socially Guided Intrinsic Motivation with Active Choice of Teacher and Strategy) algorithm that efficiently and actively combines autonomous self-exploration and interactive learning, to address the learning of multiple outcomes, with outcomes of different types, and with different sampling modes. In particular, it learns actively to decide on the fundamental questions of programming by demonstration: *what and how* to learn; but also *what, how, when and who* to imitate. This interactive learner decides efficiently and coherently whether to use social guidance. **It learns when to ask for demonstration, what kind of demonstrations (action to mimic or outcome to emulate) and who to ask for demonstrations, among the available teachers.** Its hierarchical architecture bears three levels. The lower level explores the policy parameters space to build skills for determined goal outcomes. The upper level explores the outcome space to evaluate for which outcomes he makes the best progress. A meta-level actively chooses the outcome and sampling mode that leads to the best competence progress. We showed through our illustration example that **SGIM-ACTS can focus on the outcome where it learns the most, while choosing the most appropriate associated sampling mode.** The active learner can explore efficiently a composite and continuous outcome space to be able to generalise for new outcomes of the outcome spaces.

SGIM-ACTS has been shown an efficient method for learning with multiple teachers and multiple outcome types. The number of outcomes used in the experiment is infinite, with a continuous outcome space that is made of 2 types of outcomes, but all the formalism and framework is in principle scalable to a higher number of types of outcomes. Likewise, the method should apply to domestic or industrial robots who usually interact with a finite number of teachers. Even in the case of correspondence problems, the system still takes advantage of the demonstrations to bias its exploration of the outcome space. When the discrepancies between the teacher and the learner are small, demonstrations advantageously bias the exploration of the outcome space, as argued in section 3.5 (page 66). Future work should test SGIM-ACTS on more complex environments, and with real physical robots and everyday human users.

It would also be interesting to compare the outcomes selected by our system to developmental behavioural studies, and highlight developmental trajectories. A first case study is described in the next section, where we show that using SGIM-ACTS for an agent learning to produce sounds entails developmental sequences that are qualitatively supported by developmental psychology studies.

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education, one would obtain the adult brain [...] Our hope is that there is so little mechanism in the child brain that something like it can be easily programmed. The amount of work in the education we can assume, as a first approximation, to be much the same as for the human child.

Turing, A.M. (1950). Computing machinery and intelligence. *Mind*, LIX(236):433–46

5

Illustration on a Developmental Sequence for Vocalisation

Contents

| | | |
|---------|--|-----|
| 5.1 | Development of Vocalisation with Intrinsic Motivation and Social Interaction | 105 |
| 5.1.1 | Vocalisation, Intrinsic Motivation and Social Interaction | 106 |
| 5.1.2 | Development of Vocalisation | 109 |
| 5.1.3 | A Computational Model of the Development of Vocalisation | 109 |
| 5.2 | Model | 111 |
| 5.2.1 | Sensorimotor System | 111 |
| 5.2.1.1 | Vocal Tract and Auditory System | 111 |
| 5.2.1.2 | Dynamical Properties | 114 |
| 5.2.1.3 | Vocalisation Classification | 116 |
| 5.2.2 | Internal Sensorimotor Model | 117 |
| 5.2.3 | Intrinsically Motivated Active Exploration | 118 |
| 5.2.4 | Imitation System | 122 |
| 5.3 | Results | 123 |
| 5.3.1 | Emergence of developmental sequences in autonomous vocal exploration | 124 |
| 5.3.2 | Influence of the Auditory Environment | 127 |
| 5.4 | Conclusion | 131 |
| 5.5 | Discussion | 134 |

In this chapter, we apply the algorithmic architecture SGIM-ACTS described in the previous chapter, to another learning problem: the development of vocalisation in babies.

The first contribution of our work is to show that the algorithmic architecture SGIM-ACTS is not specific to a particular instantiation of the different functions described in the previous chapter. **We implement SGIM-ACTS with a different model representation and a different learning algorithm, and show its efficiency.**

The second contribution is epistemological. While the previous chapters aimed at building artificial systems for life-long learning, this study shows that **the developed architecture can be useful to model and understand better infant development.** We use the developed algorithms to give an insight in an issue in child psychology: how do children learn to vocalise, developing from undistinguishable sounds to language-dependent words? We illustrate that SGIM-ACTS reproduces a developmental trajectory as a side effect. It could thus model child development. The work presented in this chapter has been submitted in (Moulin-Frier et al., accepted) for which we played a central role in the design of the learning paradigm and the assessment of the influence of the mother tongue on the development of vocalisation.

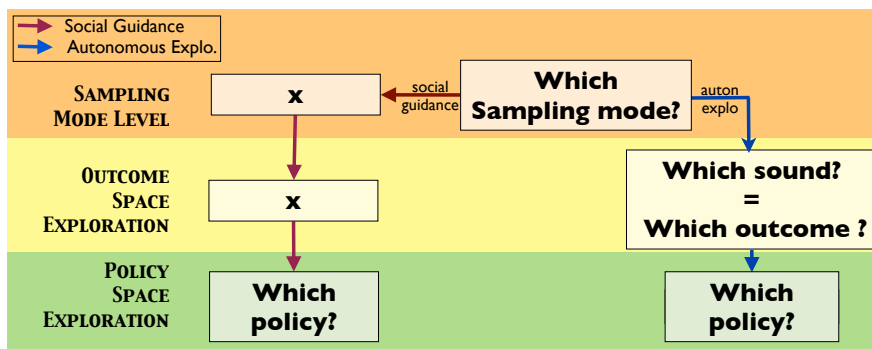


Figure 5.1.1: The strategic learner samples data by choosing on the 3 levels of its exploration space: which sampling mode to use, the sound to produce, and the motor command to use.

5.1 DEVELOPMENT OF VOCALISATION WITH INTRINSIC MOTIVATION AND SOCIAL INTERACTION

Within the goals of cognitive developmental robotics, this chapter aims at **understanding the living using artificial systems**. We developed SGIM-ACTS inspired by developmental science which is not only a means to build more versatile and adaptive machines. In this chapter, SGIM-ACTS is also a means to evaluate the coherence of theories trying to understand biological development on the one hand, and these studies on biological development will validate the coherence of SGIM-ACTS on the other hand.

In these experiments, we bridge the gap between two issues in infant development: vocal development and intrinsic motivation. We propose and experimentally **test the hypothesis that general mechanisms of intrinsically motivated spontaneous exploration, also called curiosity-driven learning, can self-organize developmental stages during early vocal learning**. We introduce a computational model of intrinsically motivated vocal exploration, which allows the learner to autonomously structure its own vocal experiments, and thus its own learning schedule, through a drive to maximize competence progress. This model relies on a physical model of the vocal tract, the auditory system and the agent’s motor control as well as vocalizations of social peers. We present computational experiments that show how such a mechanism can explain the adaptive transition from vocal self-exploration with little influence from the speech environment, to a later stage where vocal exploration becomes influenced by vocalizations of peers. Within the initial self-exploration phase, we show that **a sequence of vocal production stages self-organizes, and shares properties with data from infant developmental psychology**: the vocal learner first discovers how to control phonation, then focuses on vocal variations of unarticulated sounds, and finally automatically discovers and focuses on babbling with articulated proto-syllables. As the vocal learner becomes more proficient at producing complex sounds, imitating vocalizations of peers starts to provide high learning progress explaining **an automatic shift from self-exploration to vocal imitation**. For this aim, (Moulin-Frier et al., accepted) extended an existing implementation of a probabilistic framework for autonomous exploration developed by (Moulin-Frier and Oudeyer, 2013a,b) and used it for the set of experiments (2). The author of this thesis then included a social guidance mechanism to extend to SGIM-ACTS algorithm and study the influence of the mother tongue on the development of vocalisation for the set of experiments (1). In this study, the choices made by the active learner are summarised in **Figure 5.1.1**.

5.1.1 VOCALISATION, INTRINSIC MOTIVATION AND SOCIAL INTERACTION

Early on, babies seem to explore vocalizations as if it was a game in itself, as reported by Oller (Oller, 2000) who cites two studies from the 19th century:

“[At] three months were heard, for the first time, the loud and high crowing sounds, uttered by the child spontaneously, [...] the child seemed to take pleasure in making sounds.” (Sigismund, 1971)

“[He] first made the sound *mm* spontaneously by blowing noisily with closed lips. This amused [him] and was a discovery for [him].”¹ (Taine, 1971)

Such play with his vocal tract, where the baby discovers the sounds he can make, echoes other forms of body play, such as exploration of arm movements or how he can touch, grasp, mouth or throw objects. The concept of *intrinsic motivation* has been proposed in psychology to account for such spontaneous exploration (Berlyne, 1954; Deci and Ryan, 1985; Ryan and Deci, 2000a; Csikszentmihalyi, 1997; Gottlieb et al., 2013):

“Intrinsic motivation is defined as the doing of an activity for its inherent satisfaction rather than for some separable consequence. When intrinsically motivated, a person is moved to act for the fun or challenge entailed rather than because of external products, pressures or reward.” (Ryan and Deci, 2000a)

Intrinsic motivation refers to a mechanism pushing individuals to select and engage in activities for their own sake because they are inherently interesting (in opposition to *extrinsic motivation*, which refers to doing something because it leads to a separable outcome). A key idea of recent approaches to intrinsic motivation is that *learning progress* in sensorimotor activities can generate intrinsic rewards in and for itself, and drive such spontaneous exploration (Gottlieb et al., 2013). Learning progress refers to the infant’s improvement of his predictions or control over activity they practice, which can also be described as reduction of uncertainty (Friston et al., 2012).

Although spontaneous vocal exploration is an identified phenomenon, occurring in the early stages of infant development, the specific mechanisms of such exploration and the role of intrinsic motivation for the *structuration* of early vocal development has not received much attention so far to our knowledge. We propose that mechanisms of intrinsically motivated spontaneous exploration, which we also refer to as curiosity-driven learning, play an important role in speech acquisition, by driving the infant to follow a self-organized developmental sequence which will allow him to progressively learn to control his vocal tract. This is to our knowledge a largely unexplored hypothesis. The goal of this article is to formalize in detail this hypothesis and study general properties of such mechanisms in computer experiments.

Several computational models of speech development, where speech acquisition is organized along a developmental pathway, have been elaborated so far. They have shown how such stage-like organization can ease the acquisition of complex realistic speech skills.

The DIVA model (Guenther et al., 1998; Guenther, 2006), as well as Kröger’s model (Kröger et al., 2009), propose architectures partly inspired by neurolinguistics. They involve two learning phases. The first one

¹We have changed the gender of the subject to a male in this quotation, in order to follow the convention of the present article. Throughout this paper, we will use “he” for an infant, “she” for a caregiver (e.g. the mother) and “it” for a learning agent (the model).

is analogous to infant babbling and corresponds to semi-random articulator movements producing auditory and somatosensory feedbacks. This is used to tune the correspondences between representation maps within a neural network. In the second phase, the vocal learner is presented with external speech sounds analogous to an ambient language and learns how to produce them adequately. The Eliza model (Howard and Messum, 2011) also distinguishes several learning phases. In the first phase of exploration, the agent is driven by a reward function, including intrinsic rewards such as sound salience and diversity, as well as articulatory effort. Various parameterizations of this reward function allows the model to produce vocalizations in line with Oller’s vocal developmental stages of infants. In a subsequent phase, the sounds produced by the model attract the attention of a caregiver, providing an external reinforcement signal. Other models also use a reinforcement signal, either from human listeners (social reinforcement (Warlaumont, 2013, 2012)) or based on sound saliency (intrinsic reinforcement (Warlaumont, 2012)), and show how this can influence a spiking neural network to produce canonical syllables. Such computational models of speech acquisition pre-determine the global ordering and timing of learning experiences, which amounts to preprogramming the developmental sequence. Understanding how a vocal developmental sequence can be formed is still a major mystery to solve, and this article attempts a first step in this direction.

We build on recent models of skill learning in other modalities (e.g. locomotion or object manipulation), where it was shown that mechanisms of intrinsically motivated learning can self-organize developmental pathways, adaptively guiding exploration and learning in high-dimensional sensorimotor spaces, involving highly redundant and non-linear mappings (Oudeyer et al., 2007; Baranes and Oudeyer, 2013; Oudeyer et al., 2013; Gottlieb et al., 2013). Such models concretely formalize concepts of intrinsic motivation described in the psychology literature into algorithmic architectures that can be experimented in computers and robots (Schmidhuber, 1991b; Barto et al., 2004b; Oudeyer and Kaplan, 2007; Baldassarre, 2011). Detailed discussions of the engineering aspects of such intrinsic motivation mechanisms, casted in the statistical framework of active learning, have been recently published and showed their algorithmic efficiency to learn sensorimotor coordination skills in redundant non-linear high-dimensional mappings (Baldassarre and Mirolli, 2013a; Baranes and Oudeyer, 2013; Srivastava et al., 2013).

Indeed, transposed in curiosity-driven learning machines (Schmidhuber, 1991b; Barto et al., 2004b; Schmidhuber, 2010; Schembri et al., 2007; Merrick and Maher, 2009; Hart, 2009; Stout and Barto, 2010) and robots (Oudeyer et al., 2007; Baranes and Oudeyer, 2013), these developmental mechanisms have been shown to yield highly efficient learning of inverse models in high-dimensional redundant sensorimotor spaces (Baranes and Oudeyer, 2013, 2010a). These spaces share many mathematical properties with vocal spaces. Efficient versions of such mechanisms are based on the active choice of learning experiments that maximize learning *progress*, e.g. improvement of predictions or of competences to reach goals (Schmidhuber, 1991b; Oudeyer et al., 2007; Oudeyer and Kaplan, 2007; Baranes and Oudeyer, 2013; Srivastava et al., 2013). Such learning experiments are called “progress niches” (Oudeyer et al., 2007).

Yet, beyond pure considerations of learning efficiency, exploration driven by intrinsic rewards measuring learning progress was also shown to self-organize structured developmental pathways, both behaviorally and cognitively. Indeed, such mechanisms automatically drive the system to explore and learn first easy skills, and then progressively explore skills of increasing complexity (Oudeyer et al., 2007). They have been shown to generate automatically behavioural and cognitive developmental structures and have been analyzed in relation to their similarities with infant development (Oudeyer et al., 2007; Kaplan and Oudeyer, 2005; Oudeyer and Kaplan, 2006; Moulin-Frier and Oudeyer, 2012). For example, in the Playground

Experiment, a curiosity-driven learning robot was shown to self-organize its own learning experiences into a sequence of behavioural and cognitive stages where it spontaneously acquired various affordances and skills of increasing complexity (Oudeyer et al., 2007). It was also shown how it could discover and focus on elementary vocal interaction with a peer as a spontaneous consequence of its general drive to explore situations where it can improve its predictions (Oudeyer and Kaplan, 2006). Focusing on vocal interactions was thus explained as a special case of focusing on an activity that provides learning progress (i.e. a particular progress niche). This therefore allowed to generate some novel hypotheses to explain infant development, from the behavioural (Oudeyer and Kaplan, 2006), cognitive (Kaplan and Oudeyer, 2005) or brain circuitry (Kaplan and Oudeyer, 2007) perspectives (see (Gottlieb et al., 2013) for a review on these novel perspectives). Intrinsically motivated spontaneous learning has also been combined with mechanisms of imitation learning within the SGIM-ACTS architecture, as detailed in (Nguyen and Oudeyer, 2012d). In this model, formulated within the framework of strategic learning (Lopes and Oudeyer, 2012), a hierarchical active learning architecture allows an interactive learning agent to choose by itself when to explore autonomously, and when, what and who to imitate, based on measures of competence progress.

Although intrinsic motivation and socially guided learning have already been considered in computational models specifically studying speech acquisition, to our knowledge, they have so far been considered as two distinct learning phases with a hard-coded switch between them (e.g. (Guenther et al., 1998; Guenther, 2006; Kröger et al., 2009; Howard and Messum, 2011)). In other words, the existence of distinct developmental stages was presupposed in these models. In contrast, these distinct learning phases emerge from the Playground Experiment, even though only a simplistic vocal system was considered (only pitch and duration were controlled, and no physical model of the vocal tract was used; modeling of speech acquisition per se was not the focus of this study).

Our main contribution in this paper is to show how mechanisms of intrinsically motivated exploration applied on a realistic articulatory-auditory system self-organizes autonomously into coherent *vocal* developmental sequences. This follows the approach of our previous works (Moulin-Frier and Oudeyer, 2012, 2013a,b), which were preliminary studies limited to vowel production and focusing only on autonomous learning, i.e. without considering a surrounding ambient language.

In such a conceptual framework, developmental structures are neither learnt from “tabula rasa” nor a pre-determined result of an innate “program”: they self-organize out of the dynamic interaction between constrained cognitive mechanisms (including curiosity, learning, and abstraction), the morphological properties of the body, and the physical and social environment which itself is constrained and ordered by the developmental level of the organism (Oudeyer et al., 2007; Thelen and Smith, 1996). Thus, the approach we take can be viewed as an instantiation of the concept of epigenesis, in the sense proposed by (Gottlieb, 1991).

The study of such a dynamical systems approach, where curiosity-driven learning is an important force, can take ample advantage of computer modeling as a research tool. Here in particular, it can help to understand better the dynamics underlying early vocal development, and in particular understand what are the mechanisms which generate the developmental sequence(s) in vocal productions and capabilities observed in infants. In particular, it can help to understand what is the precise role of intrinsic motivation.

In the next sections of this introduction, we summarize properties of vocal development during the first year and describe the general principles of the computational model we study in this article.

5.1.2 DEVELOPMENT OF VOCALISATION

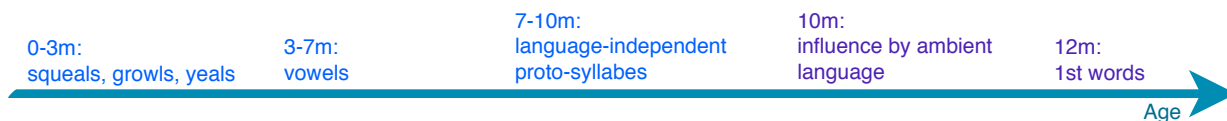


Figure 5.1.2: The first year of infant vocal development.

Despite inter-individual variations in infant vocal development (e.g. (Vihman et al., 1986)), strong regularities in the global structuration of vocal development are identified (Oller, 2000; Kuhl, 2004). In this article, we adopt the view from Oller (Oller, 2000) as well as Kuhl (Kuhl, 2004). **Figure 5.1.2** schematizes this vocal development during the first year of infant. It can be summarized as follows. First, until the age of approximately 3 months, an infant produces non-speech sounds like squeals, growls and yeals. During this period, he seems to learn to control infrastructural speech properties, e.g. phonation and primitive articulation (Oller, 2000). Then, from 3 to 7 months, he begins to produce vowel-like sounds (or quasi-vowels) while he probably learns to control his vocal tract resonances. At 7 months, canonical babbling emerges where well-timed sequences of proto-syllables are mastered. But it is only around the age of 10 months that infant vocal productions become more influenced by the ambient language, leading to first word productions around 1 year of age.

Two features of this developmental sketch are particularly salient.

- Infants seem to first play with their vocal tracts in a relatively language-independent way, and then are progressively influenced by the ambient speech sounds.
- In the initial phase, when sounds produced by their peers influence little their vocalizations, infants seem to learn skills of increasing complexity: normal phonation, then quasi-vowels and finally proto-syllables. According to Oller (Oller, 2000), such a sequence displays a so-called natural, or logical hierarchy. For example, it is impossible to master quasi-vowel production without previously mastering normal phonation.

5.1.3 A COMPUTATIONAL MODEL OF THE DEVELOPMENT OF VOCALISATION

To articulate hypotheses about the possible roles of intrinsic motivation in the first year of vocal development, we build here a computational model of an intrinsically motivated vocalizing agent, in contact with vocalizations of peers. In the model, an individual speech learner has the following characteristics, described in detail in next sections:

- It embeds a realistic model of a human vocal tract: the articulatory synthesizer used in the DIVA model (Guenther et al., 2006). This model provides the way to produce sequences of vocal commands and to compute corresponding sequences of acoustic features, both in multi-dimensional continuous domains.
- It embeds a dynamical model for producing motions of the vocal tract, based on a an over-damped spring-mass model. This model describes dynamical aspects such as co-articulation in sequences of vocal targets.

- It is able to iteratively learn a probabilistic sensorimotor model of the articulatory-auditory relationships according to its own experience with the vocal tract model. Because the sensorimotor learning is iterative during the life time of the agent, it will first be inefficient at using this model for control, and then progresses by learning from its own experience.
- It is equipped with an intrinsically motivated exploration mechanism, which allows it to generate and select its own auditory goal sequences. Such mechanism includes a capability to empirically measure its own competence progress to reach sequences of goals. Then, an action selection system stochastically self-selects target goals that maximize competence progress.
- It is able to hear sounds of a simulated ambient language, and its intrinsic motivation system is also used to decide whether to self-explore self-generated auditory goals, or to try to emulate adult sounds. This choice is also based on a measure of competence progress for each strategy.

Then, we present experiments allowing us to study how the developmental structuration of early vocal exploration could be self-organized in an intrinsically motivated speech learner, under the influence of sounds in the environment and constrained by the physical properties of the sensorimotor system.

In a first series of experiments, we consider a speech learner who is not exposed to external speech sounds. This allows the study of the role of intrinsic motivation independently of any social influence. We show how a cognitive architecture for intrinsically motivated autonomous exploration (SAGG-RIAC, (Baranes and Oudeyer, 2013; Moulin-Frier and Oudeyer, 2013b)), applied to learning to control an articulatory synthesizer (i.e. a vocal tract model able to produce speech sounds from articulatory configurations), can self-organize coherent vocal developmental sequences. This work extends preliminary studies (Moulin-Frier and Oudeyer, 2012, 2013b,a) through the use of a different vocal tract model and a more complex model of motion control dynamics with an overdamped spring-mass dynamical system, providing the agent with a more realistic and powerful mechanism to produce (un)articulated sounds.

In a second series of experiments, the speech learner is exposed to speech sounds from its environment. The cognitive architecture is extended to strategic interactive intrinsically motivated learning (SGIM-ACTS, (Nguyen and Oudeyer, 2012d)), where intrinsic motivation is also used by the learner to decide when to self-explore and when to try to imitate sounds in the environment. In the present study, we suppose that the sounds of the adult are directly imitable (we do not account for the pitch and formant differences between infants and adults for instance). We show that the system first focuses on self-exploration of vocalization. It later on shifts to vocal imitation, which then influences its vocal learning in ways that are specific to the speech environment. Yet, in this paper, we do not study the social interaction aspect of the teacher and, in particular, we do not model the behavior of the adult in response to the learner behavior.

Our aim is to study how important aspects of infant vocal development in the first year of life, described in the previous section, could be explained by the interaction between these building blocks: an intrinsic motivation system, a dynamic motor system associated to morphological and physiological constraints, an imitation system and a system for learning a sensorimotor model out of physical experiments. We will show that competence progress based autonomous exploration is able to provide a unified explanation for both the tendency to produce vocalizations of increasing complexity and the progressive influence of the ambient adult sounds. Imitating adult sounds becomes interesting for the speech learner only when basic speech production principles have been previously mastered. Contrarily to existing models of speech acquisition we described so far, our aim is not to reproduce infant vocalizations in a phonetically detailed manner, but

rather to suggest an hypothesis about how a succession of distinct developmental stages can self-organize autonomously. Howard & Messum’s model (Howard and Messum, 2011) for example, shows how distinct parameterizations of an intrinsic reward function can enable a vocal agent to discover several type of sounds coherent with observed developmental stages in infants. These parameterizations however, are hard-coded. In contrast, our model is not designed to reproduce precisely infant vocalizations within distinct vocalization stages, but rather to understand how the *transition* from one stage to another can be explained by a drive to maximize the competence progress to reach self-generated or ambient auditory goals. In consequence, the switch from self-generated auditory goals to the imitation of adult sounds is not hard-coded in our model, but emerges as a by-product of the drive to focus on progress niches.

5.2 MODEL

In this section, we describe the models that we use for the vocal tract and auditory signals. We describe the learning of the internal model of the sensorimotor mapping, and the intrinsic motivation mechanism which allows the learner to decide adaptively which vocalization to experiment at given moments during its development, and whether to do so through self-exploration or through imitation of external sounds.

5.2.1 SENSORIMOTOR SYSTEM

5.2.1.1 VOCAL TRACT AND AUDITORY SYSTEM

Our computational model involves the articulatory synthesizer of the DIVA model described in (Guenther et al., 2006)² based on Maeda’s model (Maeda, 1989). Without going into technical details, the model corresponds to a computational approximation of the general speech production principles illustrated in **Figure 5.2.1**. The model receives 13 articulatory parameters as input. The first 10 are from a principal component analysis (PCA) performed on sagittal contours of images of the vocal tract of a human speaker, allowing to reconstruct the sagittal contour of the vocal tract from a 10-dimensional vector. The effect of the 10 articulatory parameters from the PCA on the vocal tract shape is displayed **Figure 5.2.2**. In this study, we will only use the 7 first parameters (the effect of the others on the vocal tract shape is negligible), fixing the 3 last in the neutral position (value 0 in the software). Through an area function, associating sections of the vocal tract with their respective area, the model can compute the 3 first formants of the resulted signal if phonation occurs. Phonation is controlled through the 3 last parameters: glottal pressure controlling the intensity of the signal (from quiet to loud), voicing controlling the voice (from voiceless to voiced) and pitch controlling the tone (from low-pitched to high-pitched). It is then able to compute the formants of the signal (among other auditory and somato-sensory features) through the area function. In this study, we only use the glottal pressure and voicing parameters. In addition to the 7 articulatory parameters from the PCA, a vocal command is therefore defined by a 9-dimensional vector. From the vocal command, the synthesizer computes the auditory and somatosensory consequences of the motor command, thus approximating the speech production principles of **Figure 5.2.1**.

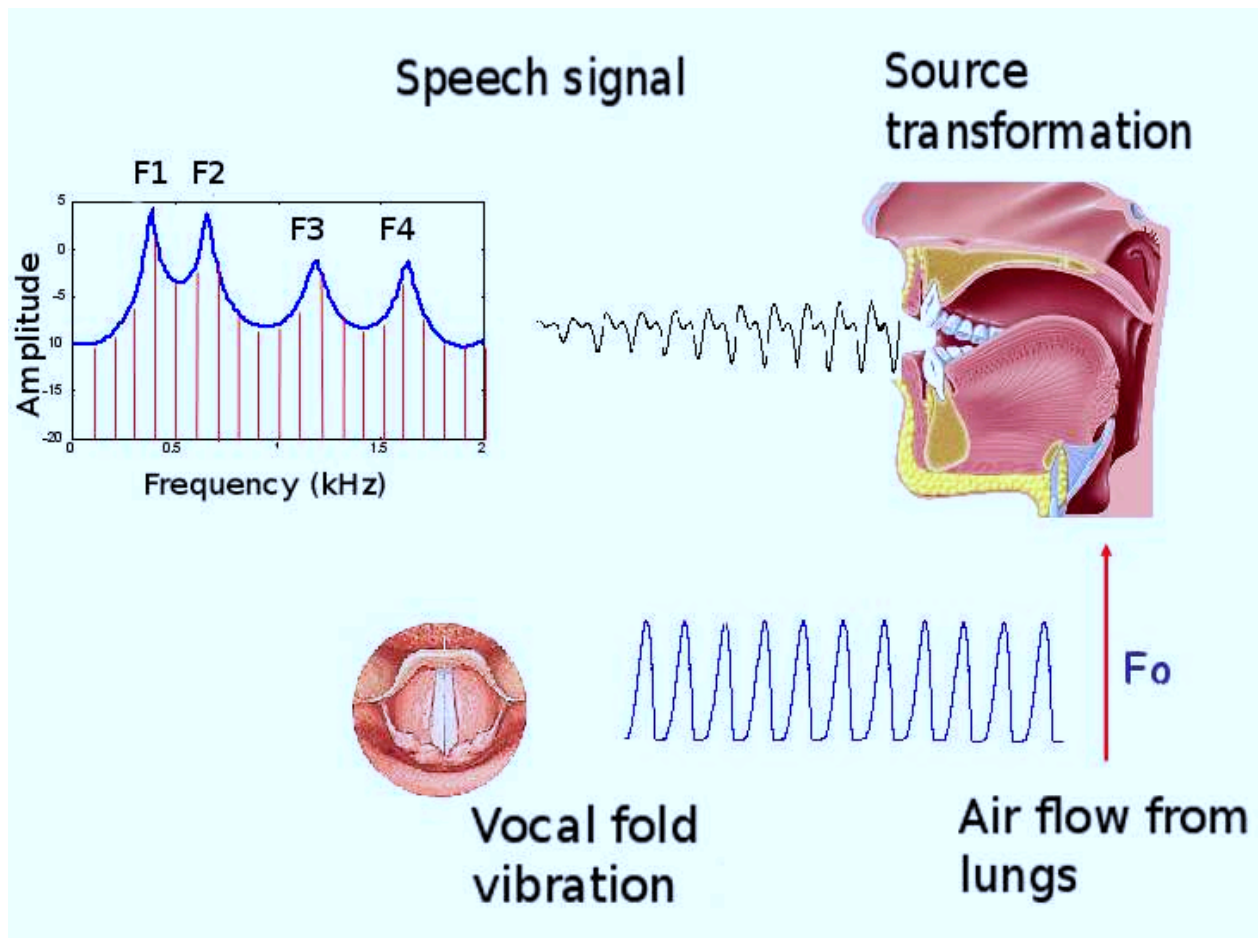


Figure 5.2.1: Speech production general principles. The vocal fold vibration by the lung air flow provides a source signal: a complex sound wave with fundamental frequency F_0 . According to the vocal tract shape, acting as a resonator, the harmonics of the source fundamental frequency are selectively amplified or faded. The local maxima of the resulting spectrum are called the formants, ordered from the lower to the higher frequencies. They belong to the major features of speech perception.

On the perception side of our model, we use the first two formants of the signal, $F1$ and $F2$, approximately scaled between -1 and 1. We also define a third parameter I which measures the intensity (or phonation level) of the auditory outcome. I is supposed to be 0 when the agent perceives no sound, and 1 when it perceives a sound. Technically, $I = 1$ if and only if two conditions are checked: (1) both pressure and voicing parameters are above a fixed threshold (null value) and (2) the vocal tract is not closed (i.e. the area function is positive everywhere). In human speech indeed, the formants are not measurable when phonation is under a certain threshold. We model this by setting that when $I = 0$, the formants do not exist anymore and are set to 0. This drastic simplification is yet arguable in term of realism, but what we want to model here is

² available online at <http://www.bu.edu/speechlab/software/diva-source-code>. DIVA is a complete neurocomputational model of speech acquisition, in which we only use the synthesizer computing the articulatory-to-auditory function.

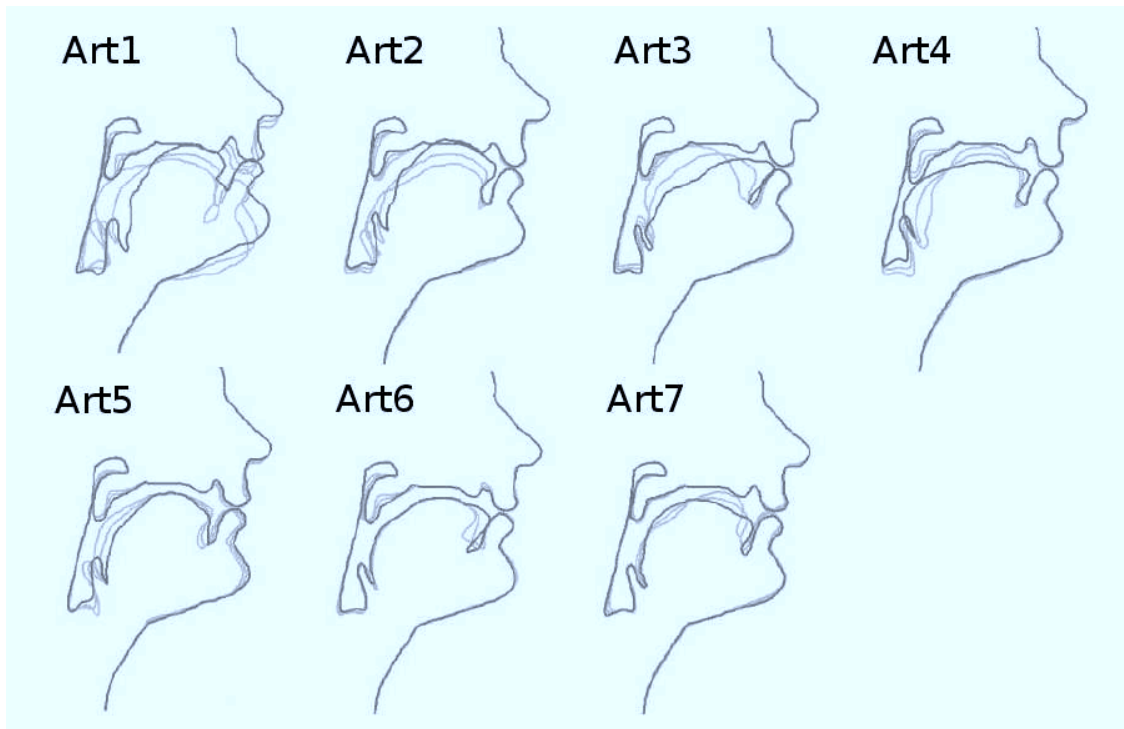


Figure 5.2.2: Articulatory dimensions controlling vocal tract shape (10 dimensions, from left to right and top to bottom), adapted from the documentation of the DIVA source code. Each subplot shows a sagittal contour of the vocal tract, where we can identify the nose and the lips on the right side. Bold contours correspond to a positive value of the articulatory parameter, the two thin contours are for a null (neutral position) and negative values. These dimensions globally correspond to the dimensions of movements of the human vocal tract articulators. For example, Art_1 mainly controls the jaw height, whereas Art_3 rather controls the tongue front-back position.

the fact that no control of the formant values can be learnt when no phonation occurs.

5.2.1.2 DYNAMICAL PROPERTIES

Speech production and perception are dynamical processes and the principles of **Figure 5.2.1** have to be extended with this respect. Humans control their vocal tract by variations in muscle activations during a vocalization, modulating the produced sound in a complex way. Closure or opening movements during a particular vocalization, coupled with variations in phonation level, are able to generate a wide variety of modulated sounds. We thus define a vocalization as a trajectory of the 9 motor parameters over time, lasting 800 milliseconds, from which the articulatory synthesizer is able to compute the corresponding trajectories in the auditory space (i.e. trajectories in the 3-dimensional space of $F1$, $F2$ and I). The agent is able to control this trajectory by setting 2 commands for each articulator: one from 0 to 250ms, the other one from 250 to 800ms. Then, the motor system is modeled as an overdamped spring-mass system driven by the following second-order dynamical equation:

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2(x - m) = 0, \quad (5.1)$$

where x is a motor parameter, and m is the command for that motor parameter. ζ is set to 1.01, ensuring that the system is overdamped (no oscillation), and ω_0 to $\frac{2\pi}{0.8}$ (0.8 being the duration of the vocalization in seconds). Thus, the agent’s policy for a vocalization is defined by two vectors m_1 and m_2 (one for each command) of 9 real values each (one for each motor parameter). The policy space is 18-dimensional. The first command is applied for the beginning of the vocalization to 250ms, the second one from 250ms to 800ms.

Figure 5.2.3A illustrates the process by showing a typical syllabic vocalization. In this illustrative example, the controlled articulators are the first and third articulators of **Figure 5.2.2** (roughly controlling the jaw height and the tongue front/back dimensions), as well as pressure and voicing. The two last ones are set to 0.5 and 0.7 respectively, for both commands, to allow phonation to occur. The “jaw parameter” (*art1* on the figure) is set to 2.0 (jaw closed) for the first command and to -3.0 for the second one (jaw open). We observe that these commands, quite far from the neutral position, are not completely reached by the motor system. This is due to the particular dynamics of the system, defined with ζ and ω_0 in the dynamical system. For the third articulator (*art3*), the commands are both at 2.0. We observe that, whereas the value 2.0 cannot be achieved completely at 250ms, it can however be reached before the end of the vocalization.

This motor system implies interaction between the two commands, i.e. a form of co-articulation. Indeed, a given motor configuration may sometimes be harder to reach if it is set as the first command, because time allocated to reach the first command is less than for the second command. Reversely, some movements may be harder to control in the second command because the final articulator positions will depend both on the first and the second commands (e.g., it is harder to reach the value -3.0 for the second command if the first command is set to 2.0, than if the first command is set to -3.0 , as seen in the example of **Figure 5.2.3**).

These characteristics are the results of modeling speech production as a damped spring-mass system (Eq. 5.1), which is a common practice in the literature (Markey, 1994; Boersma, 1998; Howard and Messum, 2011).

Figure 5.2.3B shows the resulting vocal tract shape at the end of the 2 commands (i.e. at 250ms and at

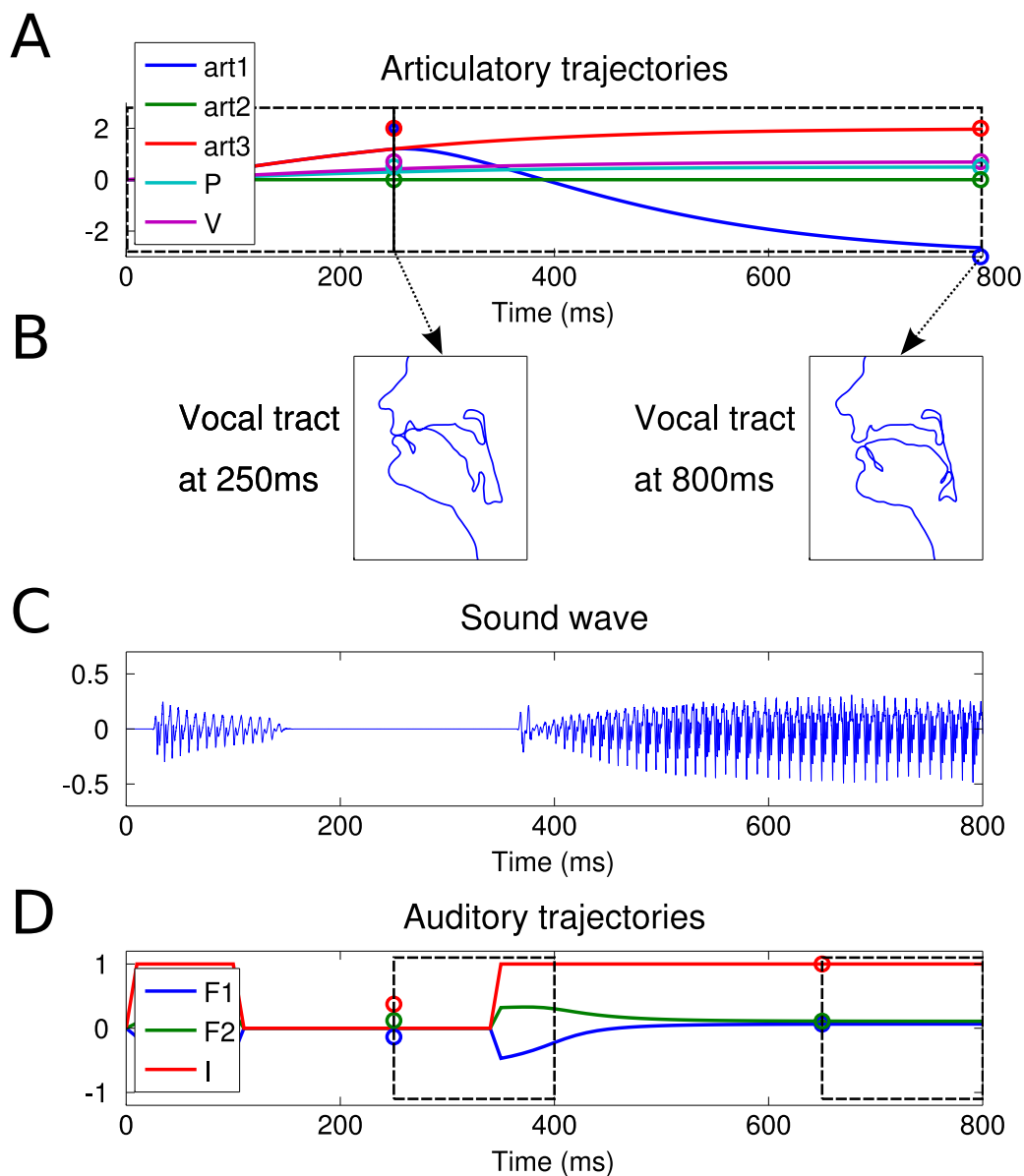


Figure 5.2.3: An illustrative vocalization example. A) Articulatory trajectories of 5 articulators during the 800ms of the vocalization (4 articulators, from *art4* to *art7* are not plotted for the sake of readability but display the same trajectory as *art2*). Circles at 250 and 800ms represents the values of the first and second commands, respectively, for each trajectory. The first commands are active from 0 to 250ms and second ones from 250 to 800ms, as represented by dotted black boxes. The trajectories are computed by the second order dynamical equation (5.1), starting in a neutral position (all articulators set to 0). B) Resulting vocal tract shapes at the end of each command, i.e. at 250 and 800ms. Each subplot displays a sagittal view with the nose and the lips on the left side. The tongue is therefore to the right of the lower lip. C) Sound wave resulting from the vocalization. D) Trajectories of the 3 auditory parameters, the intensity I and the two first formants $F1$ and $F2$. Dotted black boxes represent the two perception time windows. The agent perceives the mean value of the auditory parameters in each time window, represented by the circles at 250 and 650ms.

800ms). We observe that the vocal tract is closed at the end of the first command, open at the end of the second one.

Figure 5.2.3C shows the resulting sound. We observe that there is no sound during vocal tract closure.

Figure 5.2.3D shows the resulting trajectories of auditory parameters. In our experiments, we model the auditory perception of the agent of its own vocalization as the mean value of each parameter I , $F1$ and $F2$ in two different time windows lasting 150ms: the first one from 250 to 400ms, the second one from 650 to 800ms. The auditory representation of a vocalization is therefore a 6-dimensional vector $(I(1), I(2), F1(1), F1(2), F2(1), F2(2))$. Perceived auditory values are represented by circles on **Figure 5.2.3D**. Note that the agent does not have any perception of what happens before 250ms, and that $I(1)$ and $I(2)$ can take continuous values in $[0, 1]$ due to the averaging in a given perception time window. We will refer to the perceived “phone” of a given command for the perception occurring around the end of that command, although such an association will not be assumed in the internal sensorimotor model of the agent. Indeed, this sensorimotor system has the interesting property that the perceptions in both time windows depend on both motor commands. In the example of **Figure 5.2.3**, the perception for the first command, i.e. the mean auditory values between 250 and 400ms, would not be the same if the second motor command did not cause the vocal tract opening.

5.2.1.3 VOCALISATION CLASSIFICATION

We define three types of phones, according to the value of I for a given command. In this description, we use common concepts like vowels or consonants to make an analogy with the human types of phones, although this analogy is limited.

- Those where $I > 0.9$: , i.e. phonation occurs during almost all the 150ms of perception around the end of the command. We call them *Vowels* (V).
- Those where $I < 0.1$, i.e. there is almost no phonation during the 150ms of perception around the end of the command. We call them *None* (N).
- Those where $0.1 < I < 0.9$, i.e. phonation occurs partially during the 0.15s of perception around the end of the command. This means that the phonation level I has switched during that period. This can be due either to a closure or opening of the vocal tract, or to variations in the pressure and voicing parameters. We call them *Consonants* (C), although they are sometimes more comparable to a sort of prosody (when due to a variation in the phonation level).

This classification will be used as a tool for the analysis of the results in section 5.3, but is never known by the agent (which only has access to the values of I , $F1$ and $F2$).

Thus, each vocalization produced by the agent, belongs to the combination of 2 of these 3 types (because a vocalization corresponds to 2 commands), i.e. there are $3^2 = 9$ types of vocalizations: VV, VN, VC, NV, NN, NC, CV, CN, CC.

Then, we suggest to group these 9 types into 3 classes.

- The class *No Phonation* contains only NN: the agent has not produced an audible sound. This is due either to the fact the pressure and voicing motor variables have never been sufficiently high (not both positive, as explained in the description of the motor system) during the two 150ms perception periods, or that the vocal tract was totally closed.

- The class *Unarticulated* contains VN, NV, CN, NC: the vocalization is not well-formed. Either the first or the second command produces a phone of type *None* ($I < 0.1$, see above).
- The class *Articulated* contains CV, VC, VV and CC: the vocalization is well-formed, in the sense that there is no *None* phone. Phonation is modulated in most cases (i.e. except in the rare case where the two commands of a VV are very similar). Note that according to the definition of *consonants*, phonation necessarily occurs in both the perception time windows.

It is important to note that the auditory values of these vocalization classes span subspaces of increasing complexity. Indeed, whereas various articulatory configurations belong to the *No Phonation* class, their associated auditory values are always null, inducing a 0-dimensional auditory subspace (i.e. a point). Regarding the *Unarticulated* class, the associated auditory values span a 3-dimensional subspace because at least one command produces a phone of type *None* (i.e. the corresponding auditory values are null). Finally, in the *Articulated* classes, the auditory values span the entire 6-dimensional auditory space. These properties will have important consequences for the learning of a sensorimotor model by the agent, as we will see.

5.2.2 INTERNAL SENSORIMOTOR MODEL

The sensorimotor internal model and the intrinsic motivation system which follow were firstly described in conference papers (Moulin-Frier and Oudeyer, 2013b,a) in a more general context where the goal was to compare various exploration strategies. In this paper, we use the active goal exploration strategy – analog to the SAGG-RIAC algorithm in (Baranes and Oudeyer, 2013, 2010a).

During its life time, the agent iteratively updates an internal sensorimotor model by observing the auditory results of its vocal experiments. We denote motor commands M and sensory perceptions S . We call $f : M \rightarrow S$ the unknown function defining the physical properties of the environment (including the agent’s body). When the agent produces a motor command $m \in M$, it then perceives $s = f(m) \in S$, modulo an environmental noise and sensorimotor constraints. In the sensorimotor system defined in the previous section, M is 18-dimensional and S is 6-dimensional. f corresponds to the transformation defined section 5.2.1 and illustrated **Figure 5.2.3**, and has a Gaussian noise with a standard deviation of 0.01. By collecting (m, s) pairs through vocal experiments, the agent learns the joint probability distribution defined over the entire sensorimotor space SM (therefore 24-dimensional). This distribution is encoded in a Gaussian Mixture Model (GMM) of 28 components, i.e. a weighted sum of 28 multivariate normal distributions³. Let us note G_{SM} this GMM. It is learnt using an online version of the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) proposed by (Calinon, 2009b) where incoming data are considered incrementally. Each update is executed once each sm_step (=400) vocalizations are collected. G_{SM} is thus refined incrementally during the agent life, updating each time a number sm_step of new (m, s) pairs are collected. Moreover, we adapted this online version of EM to introduce a *learning rate* parameter α which decreases logarithmically from 0.1 to 0.01 over time. α allows to set the relative weight of the new learning data with respect to the old ones.

This GMM internal model is used to solve the inverse problem of inferring motor commands $m \in M$ that allow the learner to reach a given auditory goal $s_g \in S$. From this sensorimotor model G_{SM} , the

³We empirically chose a number of components which is a suitable trade-off between learning capacity and computational complexity.

agent can compute the distribution of the motor variables knowing a given auditory goal to reach s_g , noted $G_{SM}(M | s_g)$. This is done by Bayesian inference on the joint distribution, and results in a new GMM over the motor variables M (see e.g. (Calinon, 2009b)), from which the agent can sample configurations in M .

The whole process is illustrated **Figure 5.2.4**, on a toy example with mono-dimensional M and S . Given the current state of the sensorimotor model, the agent tries to achieve three goals, $s_1 = -9$, $s_2 = 0$, and $s_3 = 8$, i.e. three points in S (how the agent is going to self-generate such goals with intrinsic motivation will be explained below). At the beginning of the life time, the model is very poor at finding a good solution because the GMM is trained with only a few data, not necessarily concentrated in the regions useful to achieve the goals. For example, at $t = 500$, the agent is only able to correctly reach $s_2 = 0$ but is inefficient at reaching $s_1 = -9$ and $s_3 = 8$, as shown by the distributions over S in the top left corner (rotated 90 degrees anti-clockwise). Then it becomes better and better while the agent produces new vocalizations, covering a larger part of the sensorimotor space: at $t = 1500$, the agent is able to reach the three goals.

The sensorimotor system we specified in the previous section, however, involves a 24-dimensional sensorimotor space (18 articulatory dimensions and 6 auditory ones). Moreover, as we have already noted, the three vocalization classes we defined (*No Phonation*, *Unarticulated* and *Articulated*) span subspaces of the 6-dimensional auditory space with increasing dimensionality. Learning an inverse model using GMMs with a fixed number of Gaussians is harder, i.e. requires more sensorimotor experiments, as the spanned auditory subspace is of higher dimensionality. Although we do not provide mathematical arguments to this claim in this paper, it seems clear that learning an inverse model to produce *No Phonation* requires fewer learning data than learning an inverse model to produce various *Articulated* vocalizations, because the range of sensory effect is much larger in the second case.

5.2.3 INTRINSICALLY MOTIVATED ACTIVE EXPLORATION

In order to provide training data to the sensorimotor model we just described, the agent autonomously and adaptively decides which vocal experiments to make. The key idea is to self-generate and choose goals for which the learner predicts that experiments to reach these goals will lead to maximal competence progress.

The specific model we use in the first series of experiments (section 3.1) is a probabilistic version of the SAGG-RIAC architecture (Baranes and Oudeyer, 2013, 2010a). This architecture was itself derived as a functional model (Oudeyer and Kaplan, 2007; Gottlieb et al., 2013) of theories in psychology (Berlyne, 1954; Deci and Ryan, 1985; Ryan and Deci, 2000a; Csikszentmihalyi, 1997) which describe spontaneous exploration and curiosity in humans. It combines two principles: 1) goal babbling, also called goal exploration; 2) active learning driven by the maximization of empirically measured learning progress (which corresponds to the active goal strategy in (Moulin-Frier and Oudeyer, 2013b,a)). In practice, the learner self-generates its own auditory goals in the sensory space S . One goal is here a sequence of two auditory targets encoded in a 6-dimensional vector $s_g = (I(1), I(2), F1(1), F1(2), F2(1), F2(2))$ (see section 5.2.1). For each goal, it uses the current sensorimotor estimation to infer a motor program $m \in M$ in order to reach that goal. Through the sensorimotor system, this produces a vocalization and the agent perceives the auditory outcome $s \in S$, hence a new (m, s) training data. Goals are selected stochastically so as to maximize the expected competence progress (i.e. the learner is interested in goals where it predicts it can improve maximally its competence to reach them at a particular moment of its development). This allows the learner to avoid spending too

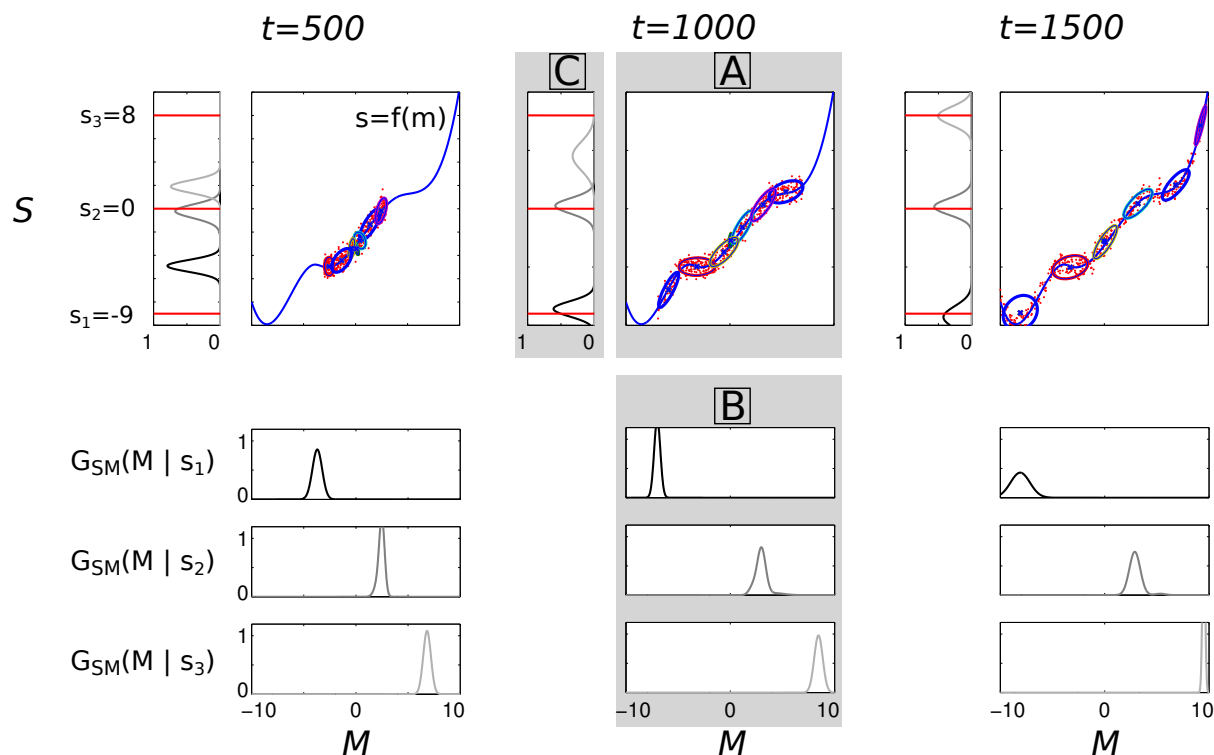


Figure 5.2.4: Illustration of incremental learning and inference in the sensorimotor model in a toy 2-dimensional sensorimotor space. The figure has three columns, corresponding to the state of a learning agent after 500, 1000 and 1500 sensorimotor experiments ($t = 500, 1000, 1500$). Each column is divided in three panels A, B and C, as indicated in the middle column (boxed letters in gray panels). X-axis (M space) and y-axis (S space) of A are shared by B and C, respectively. A) The unknown function $s = f(m)$ is represented by the blue curve. The red points are the sensorimotor experiments made at this stage (i.e. until the corresponding time index t): when m is produced, $s = f(m) + \epsilon$ is perceived, where ϵ is here a Gaussian noise with a standard deviation of 0.5. The ellipses represent the state of G_{SM} learned from the sensorimotor experiments, which is here a GMM with 6 components (each ellipse represents a 2D Gaussian). B) The three vertically-aligned plots show the motor distributions $G_{SM}(M | s_g)$ for 3 different goals, $s_1 = -9.0$ (top), $s_2 = 0.0$ (middle), and $s_3 = 8.0$ (bottom), in each of three columns (i.e. at the three time indexes). They are inferred from G_{SM} in A using Bayesian inference. C) The probability distributions on S (rotated 90 degrees anti-clockwise) resulting from sampling motor configurations according to $G_{SM}(M | s_g)$, to reach the three goals s_1, s_2 , and s_3 , the shade of grey of each one corresponding to that used in B: this means for example that, at a given time index t , producing motor commands according to the distribution $G_{SM}(M | s_3)$ (panel B, bottom) will result in sensory consequences following the darker distribution in panel C. The three considered goals s_1, s_2 and s_3 are represented by the three horizontal red lines, which are the same in the three columns. The distributions in C thus reflect how the learner is able to reach one of the three considered goals using the current state of its sensorimotor model: we observe that at $t = 500$, it can only reach $s_2 = 0$; at $t = 1000$, it can also reach $s_1 = -9$ and at $t = 1500$ it can reach those three goals.

much time on unreachable or trivial goals, and progressively explore self-generated goals/tasks of increasing complexity. As a consequence, the learner self-explores and learns only sub-parts of the sensorimotor space that are sufficient for reachable goals: this allows to leverage the redundancy of these spaces by building dense tubes of learning data only where it is necessary for control.

We define the competence c associated to a particular experiment (m, s) to reach the goal s_g as $c = \text{comp}(s_g, s) = e^{-\|s_g - s\|}$. This measure is in $[0, 1]$ and exponentially increases towards 1 when the Euclidean distance between the goal and the actual realization $s = f(m) + \epsilon$ tends to 0.

The measure of competence progress uses another GMM, G_{IM} , learnt using the classical version of EM on the recent goals and their associated competences. This GMM provides an interest distribution $G_{IM}(S)$ used to sample goals in the auditory space S maximizing the competence progress in the recent sensorimotor experiments of the agent. This was firstly formalized in (Moulin-Frier and Oudeyer, 2013b,a). In this paper, we provide a graphical explanation of the process in **Figure 5.2.5**.

Following all the previous definitions, we now consider that the agent possesses the following abilities:

- Producing a complex vocalization, sequencing two motor commands interpolated in a dynamical system. It is encoded by a 18-dimensional motor configuration $m \in M$.
- Perceiving the 6-dimensional auditory consequence $s = f(m) + \epsilon \in S$, computed by an articulatory synthesizer. f is unknown to the agent.
- Iteratively learning a sensorimotor model from lots of (m, s) pairs it collects by vocalizing through time. It is encoded in a GMM G_{SM} over the 24-dimensional sensorimotor space $M \times S$.
- Controlling its vocal tract to achieve a particular goal s_g . This is done by computing $G_{SM}(M | s_g)$, the distribution over the motor space M knowing a goal to achieve s_g .
- Actively choosing goals to reach in the sensory space S by learning an interest model G_{IM} in the recent history of experiences. By sampling in the interest distribution $G_{IM}(S)$, the agent favors goals in regions of S which maximizes the competence progress.

This agent is thus able to act at two different levels. At a high level, it chooses auditory goals to reach according to its interest model G_{IM} maximizing the competence progress. At a lower level, it attempts to reach those goals using Bayesian inference over its sensorimotor model G_{SM} , and incrementally refines this latter with its new experiences. The combination of both levels results in a self-exploration algorithm (**Algorithm 5.2.1**).

The agent starts in line 1 with no experience in vocalizing. Both GMMs have to be initialized in order to be used. To do this, the agent acquires a first set of (m, s) pairs, by sampling in M around the neutral values of the articulators (see **Figure 5.2.2**). Regarding the pressure and voicing motor parameters, we consider that the neutral value is at -0.25 , which leads to *no phonation* (recall that both these parameters have to be positive for phonation to occur, section 5.2.1). This models the fact that the agent does not phonate in its neutral configuration, and has at least to raise the pressure and voicing parameters to be able to do it. The agent then executes this first set of motor configurations (mostly not phonatory), observes the sensory consequences, and initialises G_{SM} with the corresponding (m, s) pairs using incremental EM. G_{IM} is initialised by setting the interest distribution $G_{IM}(S)$ to the distributions of the sounds it just

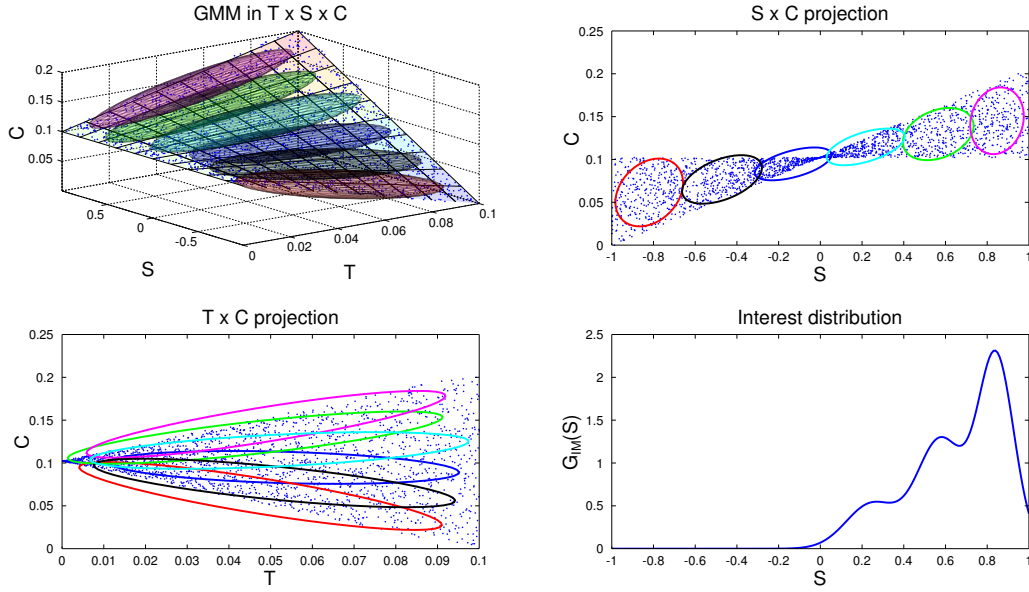


Figure 5.2.5: Illustration of interest distribution computation. Top-left: the recent history of competences of the agent, corresponding to blue points in the space $T \times S \times C$, where T is the space of recent time indexes (in \mathbb{R}^+), S the space of recently chosen goals s_g (mono-dimensional in this toy example) and C the space of recent competences of reaching those goals (in \mathbb{R}^+). For the sake of the illustration, the competence variations over time are here hand-defined (surf surface) and proportional to the values in S (increases for positive values, decreases for negative values). We train a GMM of 6 components, G_{IM} , to learn the joint distribution over $T \times S \times C$, represented by the six 3D ellipses. Projections of these ellipses are shown in 2D spaces $S \times C$ and $T \times C$ in the top-right and bottom-left plots. To reflect the competence progress in this dataset, we then bias the weight of each Gaussian to favor those which display a higher competence progress, that we measure as the covariance between time and competence for each Gaussian (in the example the purple ellipse shows the higher covariance in the bottom-left plot). We weight the Gaussians with a negative covariance between time T and competence C (blue, black and red ellipses) with a negligible factor, such that they do not contribute to the mixture. Using Bayesian inference in this biased GMM, we finally compute the distribution over the goal space S , $G_{IM}(S)$, thus favoring regions of S displaying the highest competence progress (bottom-right).

Algorithm 5.2.1 Self-exploration with active goal babbling (stochastic SAGG-RIAC architecture).

- 1: initialise G_{SM} and G_{IM}
 - 2: **while** true **do**
 - 3: $s_g \sim G_{IM}(S)$
 - 4: $m \sim G_{SM}(M | s_g)$
 - 5: $s = f(m) + \epsilon$
 - 6: $c = \text{comp}(s_g, s)$
 - 7: $\text{update}(G_{SM}, (m, s))$
 - 8: $\text{update}(G_{IM}, (s_g, c))$
 - 9: **end while**
-

produced with this first set of experiences. Thus, at the first iteration of the algorithm, the agent tries to achieve auditory goals corresponding to the sounds it produced during the initialisation phase. Then, in the subsequent iterations, the interest distribution $G_{IM}(S)$ reflects the competence progress measure, and is computed as explained above.

Line 3, the agent thus selects stochastically $s_g \in S$ with high interest values. Then it uses $G_{SM}(M | s_g)$ to sample a vocalization $m \in M$ to reach s_g (line 4). The execution of m will actually produce an auditory outcome s (line 5), and a competence measure to reach the goal, $c = comp(s_g, s)$, is computed (line 6). This allows it to update the sensorimotor model G_{SM} with the new (m, s) pairs (line 7). Finally, it updates the interest model G_{IM} (line 8) with the competence c to reach s_g

Algorithm 5.2.1 will be run and the results analyzed in section 5.3.1.

5.2.4 IMITATION SYSTEM

In language acquisition and vocalization, the social environment plays naturally an important role. Thus we consider an active speech learner that not only can self-explore its sensorimotor space, but can also learn by imitation. In a second series of experiments (section 3.2), we extend the previous model by integrating the previous learning algorithm in the SGIM-ACTS architecture, which has been proposed in (Nguyen and Oudeyer, 2012d).

We consider here that the learning agent can use one of two learning strategies, which it chooses adaptively:

- explore autonomously with intrinsically motivated goal babbling, as described previously,
- or explore with imitation learning. We distinguish mimicry, in which the learner copies the policies of others without an appreciation of their purpose, from emulation, where the observer witnesses someone producing an outcome, but then employs its own policy repertoire to reproduce the outcome, as formalized in (Lopes et al., 2009b; Call and Carpenter, 2002; Whiten, 2000; Nehaniv and Dautenhahn, 2007). As the learner a priori can not observe the vocal tract of the demonstrator, it can only emulate the demonstrator by trying to reproduce the auditory outcome observed, by using its own means, finding its own policy to reproduce the outcome. We consider that the demonstrator (the social peer) has a finite set of auditory outcomes, and every time the learner chooses to learn by social guidance, it chooses at random an auditory outcome among the set to emulate.

The learner can monitor the competence progress resulting from using each of the strategies. This measure is used to decide which strategy is the best progress niche at a given moment: a strategy is chosen with a probability directly depending on its associated expected competence progress. Thus, competence progress is used at two hierarchical levels of active learning, forming what is called strategic learning (Lopes and Oudeyer, 2012): at the higher-level, it is used to decide when to explore autonomously, and when to imitate; at the lower-level, if self-exploration is selected, it is used to decide which goal to self-explore (as in the previous model). Since competence progress is a non-stationary measure and is continuously re-evaluated, the individual *learns* to choose both the strategy $\chi \in \{autonomous_exploration, social_guidance\}$ and the auditory goals $s_g \in S$ to target, by choosing which combination enables highest competence progress.

For the particular implementation of SGIM-ACTS of this paper, we use the same formalism and implementation as in **Algorithm 5.2.1** and consider that the strategy is another choice made by the agent. This leads to **Algorithm 5.2.2**, where the interest model G_{IM} now learns an interest distribution

as in section 5.2.3. The difference is that the space of interest is now the union of the strategy space $\{autonomous_exploration, social_guidance\}$ and the auditory space S . We call $StrS$ this new space $StrS = \{autonomous_exploration, social_guidance\} \times S$. Hence G_{IM} is a distribution over $StrS$ (**Algorithm 5.2.2**, line 3). If the self-exploration strategy is chosen ($\chi = autonomous_exploration$), the agent acts as in Algorithm 5.2.1. If the social guidance strategy is chosen ($\chi = social_guidance$, line 4), the learner then emulates an auditory demonstration $s_g \in S$ chosen randomly among the demonstration set of adult sounds (line 5), overwriting s_g of line 3. It then uses its sensorimotor model G_{SM} to choose a vocalization $m \in M$ to reach s_g , by drawing according to the distribution $G_{SM}(M | s_g)$ (line 7), as in the self-exploration strategy. The execution of m will produce an auditory outcome s (line 8), from which it updates its models G_{IM} and G_{SM} (lines 10 and 11).

Algorithm 5.2.2 Strategic active exploration (active goal babbling and imitation with stochastic SGIM-ACTS architecture).

```

1: Initialise  $G_{SM}$  and  $G_{IM}$ 
2: while true do
3:    $(\chi, s_g) \sim G_{IM}(StrS)$ 
4:   if ( $\chi = social\_guidance$ ) then
5:      $s_g \leftarrow$  random auditory demonstration from the ambient language
6:   end if
7:    $m \sim G_{SM}(M | s_g)$ 
8:    $s = f(m) + \epsilon$ 
9:    $c = comp(s_g, s)$ 
10:   $update(G_{SM}, (m, s))$ 
11:   $update(G_{IM}, (\chi, s_g, c))$ 
12: end while

```

Thus, this new exploration algorithm is augmented with yet another level of learning, allowing to choose between different exploration strategies. This strategy choice moreover uses the same mechanism as the choice of auditory goals, by means of the interest model G_{IM} .

Algorithm 5.2.2 will be run and the results analyzed in section 5.3.2.

5.3 RESULTS

The results of our experiments are presented in this section. We first run experiments where our agent learns in a pure self-exploration mode (**Algorithm 5.2.1**), without any social environment or sounds to imitate. In a second time, we introduce an auditory environment to study the influence of ambient language (**Algorithm 5.2.2**).

5.3.1 EMERGENCE OF DEVELOPMENTAL SEQUENCES IN AUTONOMOUS VOCAL EXPLORATION

We ran 9 independent simulations of **Algorithm 5.2.1** with the same parameters but different random seeds, of 240.000 vocalizations each⁴. Most of these 9 simulations display the formation of a developmental sequence, as we will see. Before describing the regularities and variations observed in this set of simulations, let us first analyse a particular one where the developmental sequence is clearly observable. **Figure 5.3.1** exhibits such a simulation. We observe three clear developmental stages, i.e three relatively homogeneous phases with rather sharp transitions. These stages are not pre-programmed, but emerge from the interaction of the vocal productions of the sensorimotor system, learning within the sensorimotor model, and the active choice of goals by intrinsically motivated active exploration. First (until $\simeq 30.000$ vocalizations), the agent produces mainly motor commands which results in *no phonation* or in *unarticulated* vocalizations (in the sense of the classes defined section 5.2.1.3). Second (until $\simeq 150.000$ vocalizations), phonation almost always occurs, but the vocalizations are mostly *unarticulated*. Third, it produces mainly *articulated* vocalizations.

The agent explores its sensorimotor space by producing vocalizations of increasing complexity. The class *no phonation* is indeed the easiest to learn to produce for two reasons: the rest positions of the pressure and voicing motor parameters do not allow phonation (both around -0.25 at the initialisation of the agent, line 1 of **Algorithm 5.2.1**) ; and there is no variations on the formant values, which makes the control task trivial as soon as the agent has a bit of experience. There is more to learn with *unarticulated* vocalizations, where formant values are varying in at least one part of the vocalization, and still more with *articulated* ones where they are varying in both parts (for the first and second command).

Figure 5.3.2 shows what happens in the particular simulation of **Figure 5.3.1** in more details.

This developmental sequence is divided into 3 stages, I, II and III, stages being separated by vertical dark lines on **Figure 5.3.2**, identical on each subplot (stage boundaries are the same than in **Figure 5.3.1**).

In stage I, until approximately 30.000 vocalizations, the agent produces mainly *no phonation* and *unarticulated* vocalizations. We observe that the agent set goals for $I(1)$ either around 0, either around 1, whereas the goals for $I(2)$ stay around 0 (last row in “Goals”). By trying to achieve these goals, the agent progressively refines its sensorimotor model and progresses by raising the values of the pressure and voicing motor parameter in the first command (two last rows of the section “Motor commands”, 1st column). Other articulators remain around the neutral position (value 0). The agent is learning to phonate. The percentages of vocalization belonging to each vocalization class is provided **Table 5.3.2**.

Then, in stage II, from 30.000 to approximately 150.000 vocalizations, the agent is mainly interested in producing vocalizations which begin with a *Vowels* ($I(1) > 0.9$, see the definition of phone types in section 5.2.1.3) and finish with a *None* ($I(2) < 0.1$). During this stage, it learns to produce relatively high $F1(1)$ values, in particular by decreasing the $Art_1(1)$ parameter (approximately controlling the jaw height, see **Figure 5.2.2**). Regarding the second command, although the agent self-generates various goals for $F1(2)$ and $F2(2)$, and produces various motor commands to try to reach them, the sound produced mostly

⁴Each simulation involves several hours of computing on a desktop computer, due to the complexity of **Algorithm 5.2.1**, in particular in the Bayesian inference and update procedures.

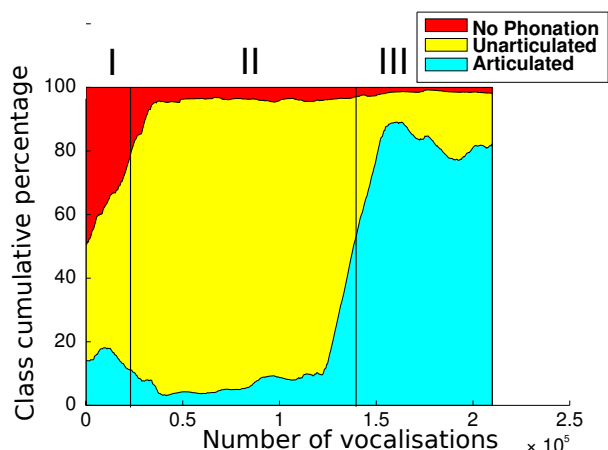


Figure 5.3.1: Self-organization of vocal developmental stages. At each time step t (x-axis), the percentage of each vocalization class between t and $t + 30.000$ is plotted (y-axis), in a cumulative manner (sum to 100%). Vocalization classes are defined in section 5.2.1.3. Roman numerals shows three distinct developmental stages. I: mainly no phonation or unarticulated vocalizations. II: mainly unarticulated. III: mainly articulated. The boundaries between these stages are not preprogrammed and are here manually set by the authors, looking at sharp transitions between relatively homogeneous phases.

| Types of sounds produced | Stage I | Stage II | Stage III | Stage IV |
|----------------------------|---------|----------|-----------|----------|
| No phonation-Unarticulated | 7 | 0 | 2 | 0 |
| Unarticulated | 0 | 7 | 0 | 3 |
| Articulated | 0 | 2 | 4 | 0 |
| Other | 2 | 0 | 1 | 0 |

Table 5.3.1: Count of vocalization stages in 9 simulations. The “types of sounds produced” (first column of the table) correspond to the most prominent class in a given stage, where stages are manually set, looking at sharp transitions between relatively homogeneous phases. These developmental stages are therefore subjective to a certain extent, in the sense that another observer could have set different ones (but hopefully also would observe major structural changes). “No phonation-Unarticulated” means a mix between *No phonation* and *Unarticulated* classes (as defined in section 5.2.1.3 in that stage. A number x in a cell means *this type of vocalizations (row) appears x times at the n^{th} stage of development (column) in the set of 9 simulations*. Two to four developmental stages were identified in each simulation, explaining why the “Stage I” and “Stage II” columns sum up to 9 (the total number of simulations), but not the “Stage III” and “Stage IV” columns.

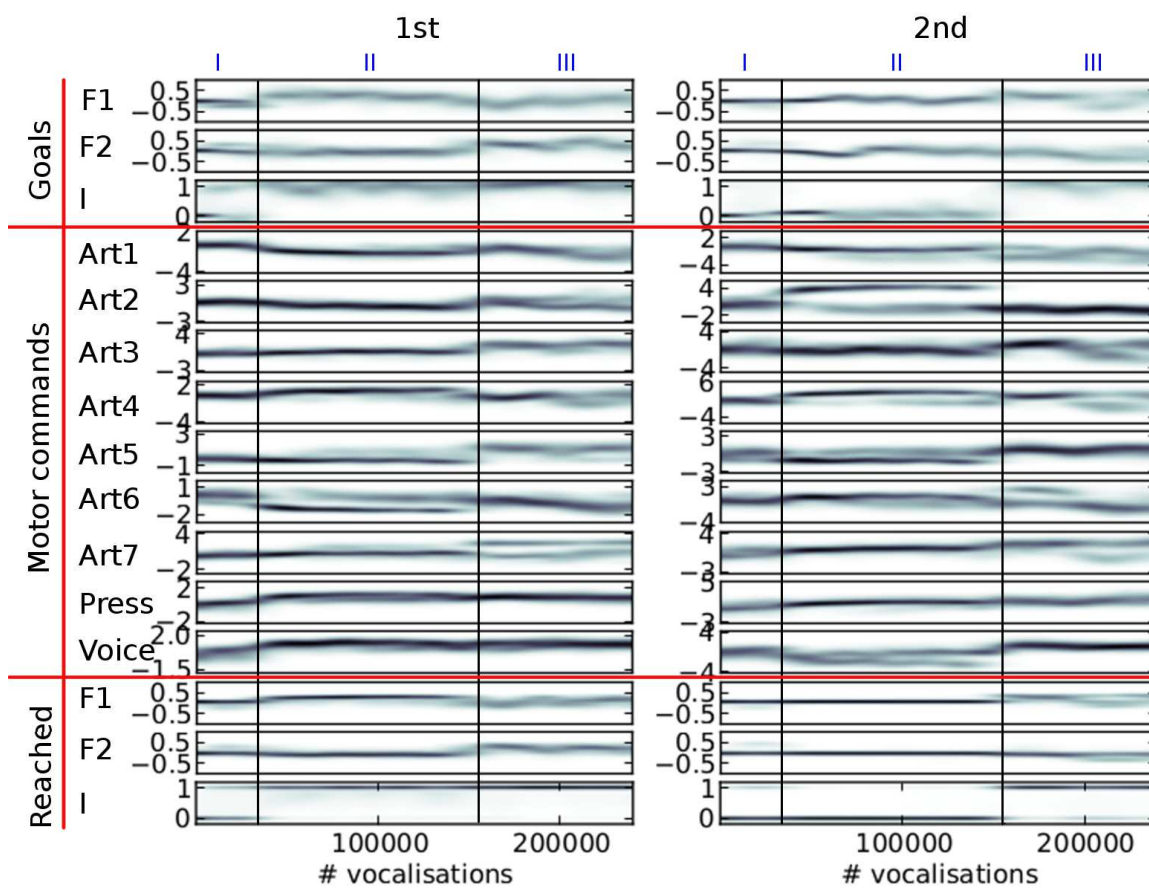


Figure 5.3.2: Evolution of the distribution of auditory goals, motor commands and sounds actually produced over the life time of a vocal agent (the same agent as in **Figure 5.3.1**). The variables are in three groups (horizontal red lines): the goals chosen by the agent in line 3 of **Algorithm 5.2.1** (top group), the motor commands it inferred to reach the goals using its inverse model in line 4 (middle group), and the actual perceptions resulting from the motor commands through the synthesizer in line 5 (bottom group). There are two columns (1st and 2nd), because of the sequential nature of vocalizations (two motor commands per vocalization). Each subplot shows the density of the values taken by each parameter (y-axis) over the life time of the agent (x-axis, in number of vocalizations since the start). It is computed using an histogram on the data (with 100 bins per axis), on which we apply a 3-bins wide Gaussian filter. The darker the color, the denser the data: e.g. the auditory parameter I actually reached by the second command ($I(2)$, last row in ‘Reached’, 2nd column), especially takes values around 0 (y-axis) until approximately 150.000th vocalization (x-axis), then it takes rather values around 1. The three developmental stages of **Figure 5.3.1** are reported at the top.

| NN | CN | NC | VN | NV | VV | CV | VC | CC |
|--------|--------|-------|--------|-------|-------|-------|-------|-------|
| 45.3 % | 13.4 % | 0.6 % | 18.9 % | 4.5 % | 9.9 % | 6.6 % | 0.7 % | 0.2 % |

Table 5.3.2: Percentage of vocalization classes produced in stage I of the studied developmental sequence.

corresponds to a *None* ($I(2) = 0$, and therefore $F1(2) = F2(2) = 0$). This is due both to the negative value of the voicing parameter (last row in “Motor commands”, second column), and to the fact that the vocal tract often ends in a closed configuration due to the poor quality of the sensorimotor model in this region (because phonation occurs very rarely for the second command, leaving the agent without an adequate learning set). During this stage, the agent explores a limited part of the sensorimotor space both in time (sound only for the first command) and space (around the neutral position), until it finally manages to phonate more globally at the end of this stage. This could be correlated to the acquisition of articulated vocalizations. The percentages of vocalization belonging to each vocalization class is provided in **Table 5.3.3**.

| NN | CN | NC | VN | NV | VV | CV | VC | CC |
|-------|--------|-------|--------|-------|-------|-------|-------|-------|
| 4.0 % | 26.9 % | 0.1 % | 62.2 % | 0.1 % | 3.4 % | 0.5 % | 2.5 % | 0.2 % |

Table 5.3.3: Percentage of vocalization classes produced in stage II of the studied developmental sequence.

Finally, in stage III (until 150.000 to the end), phonation almost always occurs during both the perception time windows (see I densities, both for goals and reached values). This is much harder to achieve for two reasons: firstly because there is a need to control a sequence of 2 articulators movement in order to reach two formant values in sequence (i.e. $F1(1)$, $F1(2)$, $F2(1)$, $F2(2)$) instead of one in the previous stage (the second command leading to no sound), and secondly because the position of the articulators reached for the second command also depends on the position of the articulators reached for the first one (a kind of coarticulation due to the dynamical properties of the motor system). We observe that the range of values explored in the sensorimotor space is larger than for the previous stage (both in motor and auditory spaces). The percentages of vocalizations belonging to each vocalization class is provided in **Table 5.3.4**.

| NN | CN | NC | VN | NV | VV | CV | VC | CC |
|-------|-------|-------|--------|-------|--------|-------|-------|-------|
| 1.6 % | 3.7 % | 0.1 % | 12.1 % | 0.8 % | 67.5 % | 6.5 % | 6.8 % | 0.8 % |

Table 5.3.4: Percentage of vocalization classes produced in stage III of the studied developmental sequence.

5.3.2 INFLUENCE OF THE AUDITORY ENVIRONMENT

In a second set of experiments, we integrated a social environment providing a set of adult vocalizations. As explained in section 5.2.4, the learner has an additional choice: it can explore autonomously, or emulate the adult vocalizations. An “ambient language” is here modeled as a set of two speech sounds. To make it coherent with human language and the learning process observed in development, we chose speech-like sounds, typically vowel or consonant-vowel sounds. In terms of our sensorimotor descriptions, the adult sounds correspond to $I1$ with low values and $I2$ with high values. **Figure 5.3.3** shows such vocalizations corresponding to those used by Teacher 1 in **Figure 5.3.4** .

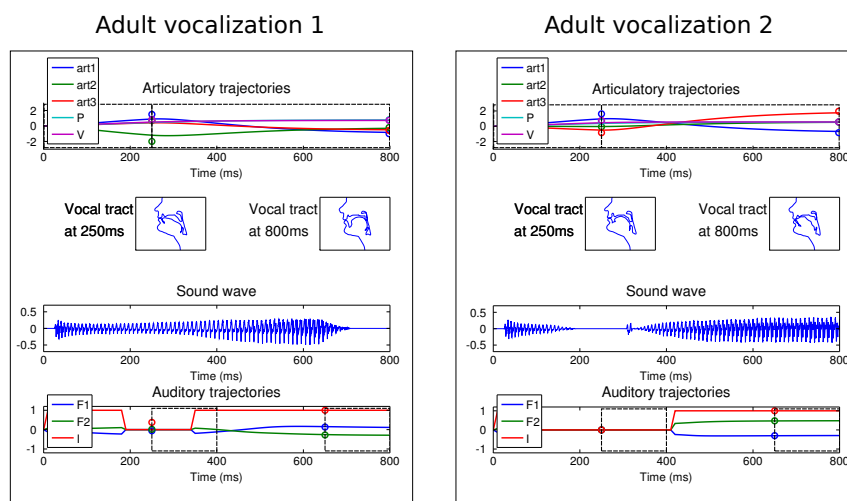


Figure 5.3.3: The two vocalizations of the adult Teacher 1 used in **Figure 5.3.4** , with the same convention as in **Figure 5.2.3** .

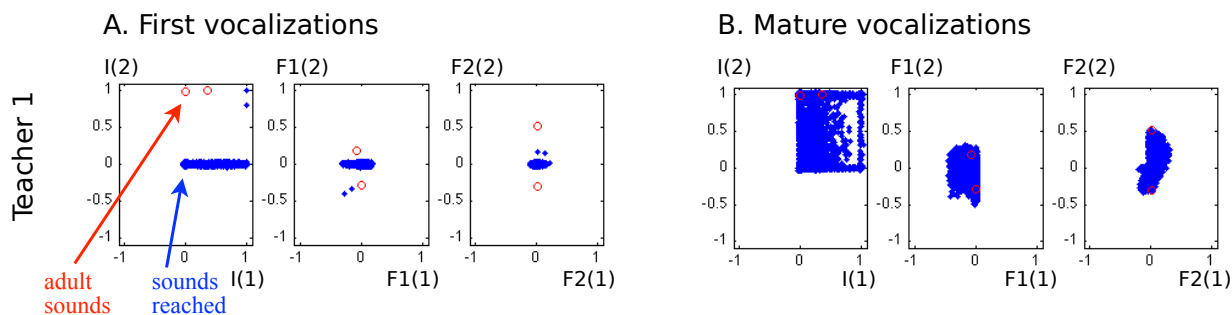


Figure 5.3.4: Vocalizations of the learning agent in the early and mature stages of vocal development. A) All auditory outcomes s produced by the agent in its early stage of vocalization are represented by blue dots in the 6-dimensional space of the auditory outcomes. The adult sounds are represented in red circles. The actually produced auditory outcomes only cover a small area of physically possible auditory outcomes, and correspond mostly to $I(2) = 0$, which represent vowel-consonant or consonant-consonant types of syllables. B) The auditory outcomes produced by the infant in its mature stage of vocalization cover a much larger area of auditory outcomes and extend in particular over areas in which vocalizations of the social peer are located.

Figure 5.3.4 shows a significant evolution in the agent’s vocalizations. In the early stage of its development, it can only make a few sounds. Most sounds correspond to small values of $I1(2)$, $F1(1)$, $F1(2)$, $F2(1)$ and $F2(2)$, as in the first developmental stage of the previous experiment (see **Table 5.3.2** and **Figure 5.3.2**). Therefore the agent is not able to reproduce the ambient sounds of its environment. In contrast, in later periods of its development, its vocalizations cover a wider range of sounds, with notably $I(1)$ and $I(2)$ both positive, which means it now produces more articulated sounds. The development of vocalizations for a self-exploring agent in the last section showed that it progressively was able to produce articulated vocalizations, which we observed at times at the end of its development. This effect has been reinforced by the environment: with articulated vocalizations to emulate, it produces this class more regularly.

Another important result is that mature vocalizations can now reproduce the ambient sounds of the environment: the regions of the sounds produced by the learner (blue dots) overlap the teacher’s demonstrations (red circles). It seems that, during the first vocalizations, the agent cannot emulate the ambient sounds because they are too far away from its possible productions, and thus it can hardly make any progress and approach these demonstrations. **Figure 5.3.5** confirms this interpretation. In the beginning, the agent makes no progress with emulation, and it is only around $t = 450$ that it makes progress with the emulation strategy. At that point, as we can see in **Figure 5.3.6**, it uses equally both strategies. This enables the agent to make considerable progress from $t = 450$ to $t = 800$. Indeed, once its mastery improves and the set of sounds it can produce increases, it then increasingly emulates ambient sounds. Once it manages to emulate the ambient sounds well, and thus its competence progress decreases, it uses less the emulation strategy and more the self-exploration strategy.

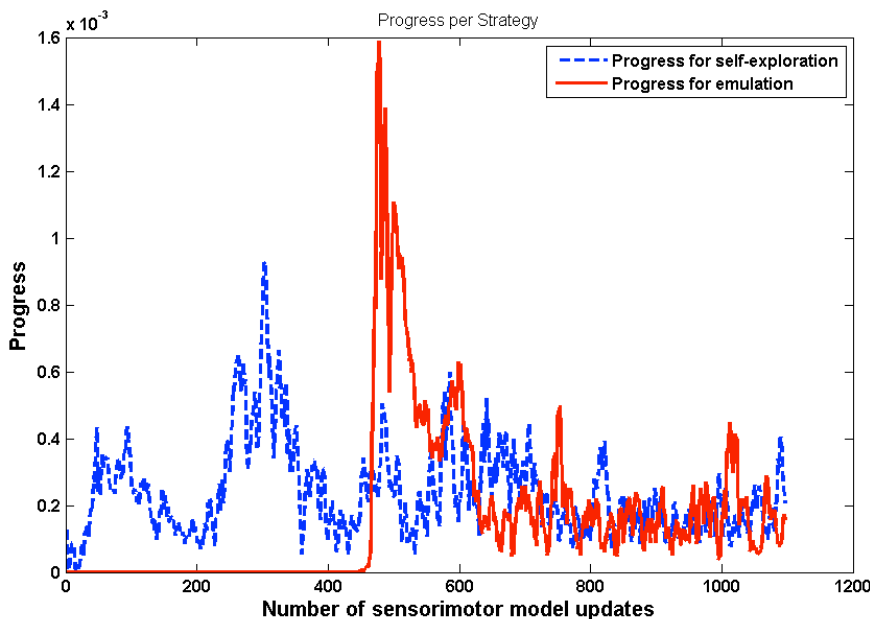


Figure 5.3.5: Progress made by each strategy with respect to the number of updates of the sensorimotor model G_{SM} . These values have been smoothed over a window of 100 updates. For $t < 450$, the agent makes no progress using emulation strategy. After $t = 450$, both strategies enable the agent to make progress.

To analyse better this emulation phenomenon and assess the influence of the ambient language, we run the same experiment with different acoustic environments. We used two other sets of speech sound demonstrations from simulated peers, and analysed the auditory productions of the agent in **Figure 5.3.7**. The first property that can be noted is that in the early phase of the vocal exploration (**Figure 5.3.7. A** and **C**), the auditory productions of the two agents are alike, and do not depend on the speech environment. On the contrary, the mature vocalizations vary with respect to the speech environment. With Teacher 1, the productions have their values $F2(1)$ and $F2(2)$ along the axis formed by the demonstration (**Figure 5.3.4. A**, last column). Comparatively, Teacher 2's speech sounds have different $F1(1)$, $F1(2)$, $F2(1)$ and $F2(2)$. As represented in **Figure 5.3.7. B**, the two speech sounds now differ mainly by their $F1(1)$ (instead of $F1(2)$) and in their subspace ($F2(1)$, $F2(2)$) the speech sounds have approximately rotated from those of Teacher 1. The produced auditory outcomes of the learner look like they have changed in the same way. Whereas the reached space (blue area) seemed to be along axis $F1(2)$ and $F2(2)$ and little on $F1(1)$ or $F2(1)$ for Teacher 1, it has extended its exploration along $F1(2)$ and $F2(2)$ for Teacher 2. With Teacher 3, the demonstrations are more localised in the auditory space, with $F1(1) < 0$ and $F2(2) > 0$. The effect we observe in **Figure 5.3.7. D** is that the exploration is more localised too: the explored space is more oriented toward areas where $F1(1) < 0$ and $F2(2) > 0$. Thus, these three examples strongly suggest a progressive influence of the auditory environment, in the sense that the first vocalizations in **Figure 5.3.4** and **5.3.7** are very similar, whereas we observe a clear influence of the speech environment on the produced vocalizations in later stages.

Altogether, the results of these experiments provide a computational support to the hypothesis that the

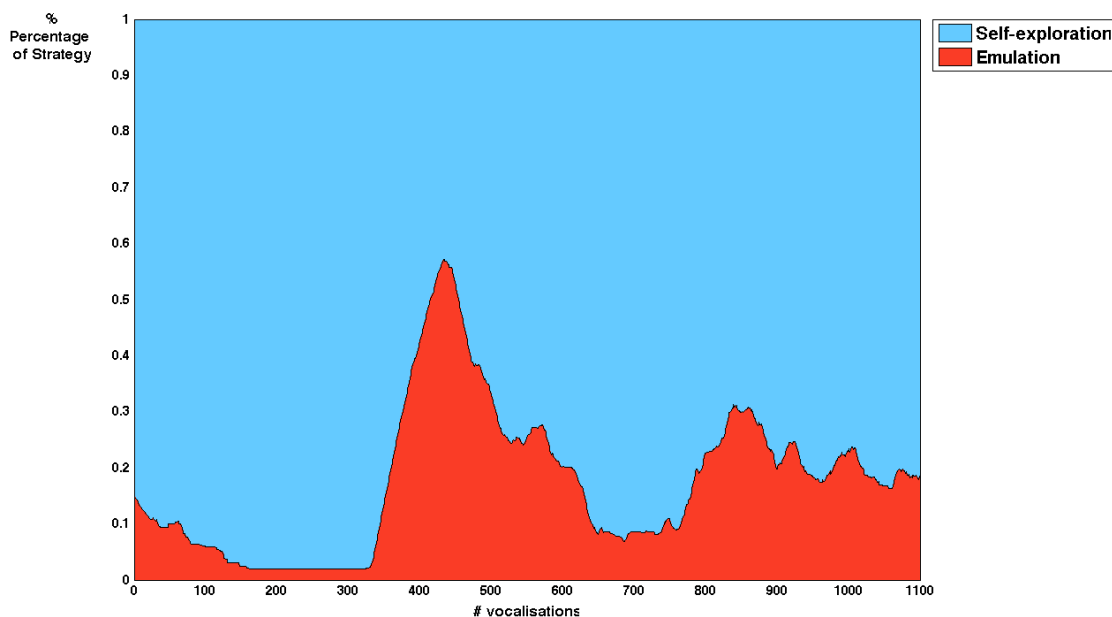


Figure 5.3.6: Percentage of times each strategy is chosen with respect to the number of updates of the sensorimotor model G_{SM} . These values have been smoothened over a window of 100 updates. For $t < 450$, the agent mainly uses self-exploration strategy. When its knowledge enables it to make progress in emulation, it chooses emulation strategy until it can emulate the ambient sounds well (and its competence progress decreases).

progressive influence of the ambient language observed in infant vocalizations can be driven by an intrinsic motivation to maximize competence progress. At early developmental stages, attempts to imitate adult vocalizations are certainly largely unsuccessful because basic speech principles, such as phonation, are not yet mastered. In this case, focusing on simpler goals probably yields better progress niches than an imitative behavior. While they are progressively mastered, the interest in these goals decreases whereas the ability to imitate adult vocalizations increases. Imitation thus becomes a new progress niche to explore

5.4 CONCLUSION

Our main contribution with respect to previous computational models of speech acquisition is that **we do not presuppose the existence of successive developmental stages, but rather they can emerge from an intrinsic drive to maximize the competence progress**. We showed that **vocal developmental stages can self-organize autonomously, from simple sensorimotor activities to more complex ones**. The agent starts producing *no phonation* and *unarticulated* vocalizations, which are easy to produce because limited in the range of their auditory effects. This can be related to the first stage in infant vocal development (**Figure 5.1.2**), where the agent produces non speech-sounds (e.g. growls, squeals...) before learning phonation and then produces not well-articulated quasi-vowels. Later on, once the agent does not progress much in producing *unarticulated* vocalizations, it focuses on more complex vocalizations of the *articulated* class. The reason is that, due to the properties of the sensorimotor system and internal model,

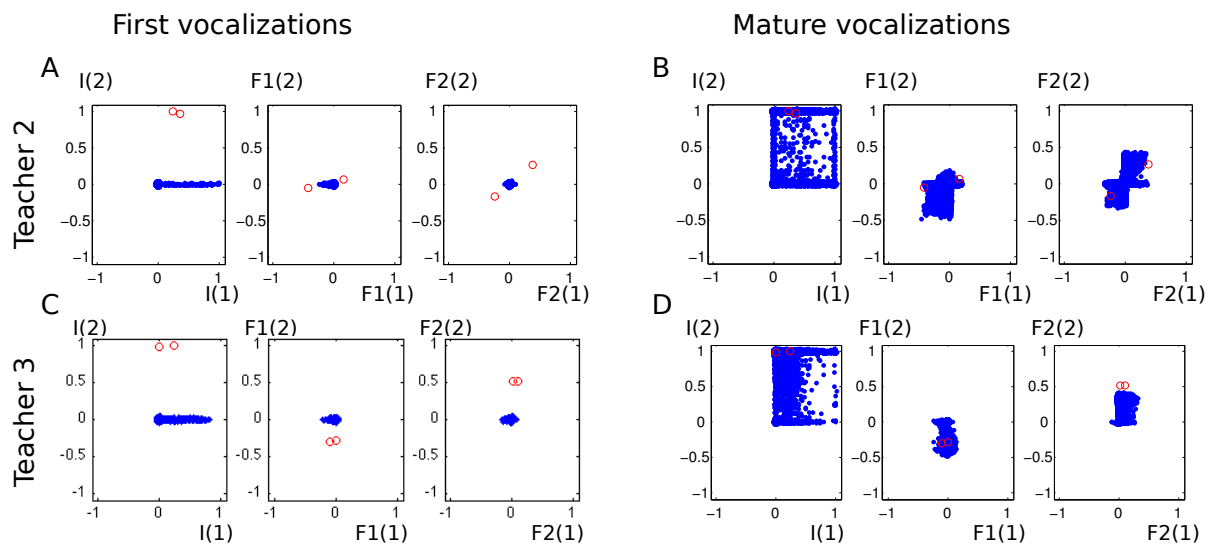


Figure 5.3.7: Vocalizations of the learning agent in the early and mature stage of vocalization in two different speech environments (Teacher 2 and Teacher 3). A and C) All auditory outcomes produced by the vocal learner in its early stage of vocal development are represented by blue dots in the 6-dimensional space of the auditory outcomes. The sounds of the environment are represented in red circles. The auditory outcomes only cover a small area, and do not depend on the speech environment. B and D) The auditory outcomes produced by the infant in its mature stage of vocal development cover a larger area of auditory outcome, which depend on the speech environment.

the mastering of complex tasks require first the mastering of simpler tasks in order to yield significant competence progress, so that these complex tasks are selected as interesting goals.

We also showed that intrinsically motivated exploration can lead to a progressive interest towards the sounds of the ambient language. Whereas the first vocalizations are mainly the result of self-exploration, they progressively lead to mastering necessary speech principles (e.g. phonation). This progressive mastering allows in turn to make significant progress in adult-speech imitation, which explains why the vocal learner starts to choose more often as targets the sound of its environment. **Competence-progress based curiosity-driven exploration could thus explain a progressive influence of the ambient language on infant vocalizations.**

We therefore showed that intrinsically motivated active exploration can self-organize a coherent developmental sequence, without any external clock or preset specification of this sequence. This possible role of intrinsic motivation, providing a mechanism to discover autonomously necessary developmental stages to structure the learning process, is here validated computationally. We believe that it could be of major interest for understanding the structuration of early vocal development in infants. Speech acquisition is such a complex task that intrinsic motivation could be a crucial component to make it possible in the infant's first year of life.

Our model, however, has a number of limitations. Firstly, our modeling choices of the articulatory and auditory representations, as well as the implementation of the transformation from the former to the latter, is somewhat less realistic than in some previous models: articulatory trajectories are specified using two commands per articulator with fixed durations and the auditory representation uses only three acoustic parameters (the intensity and the two first formants) averaged in fixed and relatively arbitrary perception time windows. Moreover, the fact that formant values are set to 0 whenever the intensity of the signal is null can appear quite unrealistic. Although previous models often provide more meticulous implementations of the sensorimotor system, including e.g. pitch or tactile information, what is important to us is a sensorimotor system where all vocalizations are not equally easy to learn in terms of control. Such a requirement is certainly necessary for a clear developmental sequence to emerge. Secondly, we did not treat a major issue in speech acquisition research, the so-called correspondence problem: how the child is able to relate its own vocalizations to adult vocalizations, whereas the vocal tract of the child is very different in size and geometry than the one of an adult, and therefore the spectral characteristics of the produced sounds are different. Solutions to overcome this problem have been proposed, generally based on adult feedback or reformulations associated with infant productions (Ishihara et al., 2009; Miura et al., 2012; Howard and Messum, 2011). This is outside the scope of this paper where our focus is on the self-organization of the developmental sequence in successive stages of increasing complexity. Extending our model to the interaction with real humans would definitely require to consider this issue.

Further works will consider higher-dimensional sensorimotor spaces for more realism. For example, the free software Praat (Boersma, 2012) is a powerful tool allowing to synthesize a speech signal from a trajectory in a 29-dimensional space of respiratory and oro-facial muscles. Numerous acoustic features can in turn be extracted from the synthesized sound, among which the Mel-frequency cepstral coefficients (MFCC, (Davis and Mermelstein, 1980)). It would also be interesting to study the effect of a vocal tract growing during the learning process, to study if our intrinsically motivated agent could re-explore only parts of the sensorimotor space which were the most affected by the vocal tract shape change. Generally, we believe that a developmental robotics approach applied to a realistic articulatory model can appropriately manage

the learning process of a complex and changing mapping in high-dimensional spaces, and that observed developmental sequences can lead to interesting comparisons with infant data and predictions. Regarding the present study, such a prediction could be that a human infant should be influenced by adult sounds earlier if they were easier to produce than well-formed syllables. For example, one could imagine an experiment in which a very young infant is put in an environment where he hears external sounds that are simpler than vowels/consonants/syllables (e.g. growls) and test whether his vocalizations become influenced by external environment earlier and/or if we can measure a greater interest than in a normal speech environment.

5.5 DISCUSSION

In this chapter, we showed that the SGIM-ACTS remains efficient with another implementation of the different modules using a probabilistic framework and more precisely with Gaussian Mixture Models. We also showed that **SGIM-ACTS can be useful to model and understand better infant development**. Moreover, we showed that **intrinsically motivated active exploration can self-organise a logical developmental sequence, coherent with developmental descriptions**. This partly validates the coherence of SGIM-ACTS. Most of all, **the developmental sequence emerged without any external clock or preset specification of this sequence**. This possible role of intrinsic motivations, providing a mechanism to discover autonomously necessary developmental stages to structure the learning process, is here validated computationally. We believe that it could be of major interest for understanding the structuring of early development in infants for vocalisation or other skills.

[T]here are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until - in a visible future - the range of problems they can handle will be coextensive with the range to which the human mind has been applied.

Newell and Simon, 1958

6

Conclusion

Contents

| | | |
|-------|--|-----|
| 6.1 | Keynote | 135 |
| 6.1.1 | Synopsis | 135 |
| 6.1.2 | Summary | 136 |
| 6.1.3 | Result | 138 |
| 6.1.4 | Take-away Message | 138 |
| 6.2 | Our approach, its Limitations and Extensions | 138 |
| 6.2.1 | Main Contribution | 138 |
| 6.2.2 | Originality of our Approach | 139 |
| 6.2.3 | Complementary Studies | 140 |
| 6.2.4 | Limitations and Extensions | 140 |
| 6.2.5 | Impact | 142 |
| 6.3 | Journal Papers | 155 |
| 6.4 | Conference Papers with Proceedings | 155 |
| 6.5 | Other International Public Presentations | 156 |

6.1 KEYNOTE

6.1.1 SYNOPSIS

Our long-term purpose is to enable learning agents to perform various tasks and adapt to their changing environment and users, which is named *life-long learning*. A main challenge of life-long learning is to explore within a limited life-time to learn open-ended skills in a too large continuous high-dimensional space describing humans' everyday environment. We proposed the idea of a data collection strategy inspired by human learning by combining social guidance and exploration based on artificial curiosity, also called *intrinsic motivation*. We thus designed an algorithmic architecture which, by combining intrinsic motivation and imitation learning, allows life-long adaptive learning because it learns faster and with better precision. It performs a wide range of tasks by structuring the social and physical environment. The take-away message is that a **robot learning with artificial curiosity and social guidance benefits from the human bias and at the same time is able to detect interesting subspaces to refine its knowledge.**

6.1.2 SUMMARY

Indeed, to have learning agents adapt and evolve in environments as complex and changing as ours, we address the problem of life-long learning, and adopt methods of cognitive developmental robotics. We have argued in chapter 1 that **strategic learning is crucial for collecting data** in complex, continuous, redundant and stochastic environments. We identified two families of learning methods in robotics, intrinsic motivation and social guidance, which we call *data collection modes*. Starting from the analysis of these two modes, we conjectured that combining both data collection modes into a strategic robot learner can push off the limits of each of these methods taken separately. Based on principles of learning by trial-and-error, enactivism, and development, we used theories of imitation learning, intrinsic motivation and teleological learning. We designed a strategic curious robot learner for interactive goal-babbling to explore areas of the space where it learns the most, i.e. where it makes most competence progress. Our agent strategically learns by goal-babbling, i.e. by goal-oriented exploration of the outcomes produced as opposed to the exploration of the actuator space, and by interactive learning, i.e. by actively requesting for social guidance when it needs. We proposed an algorithmic architecture, called Socially Guided Intrinsic Motivation (SGIM). Its different implementations are studied in chapters 2 to 5. As summarised in **Figure 6.1.1**, we present three algorithmic architectures, where SGIM-ACTS is the fully active learner, choosing all aspects of its sampling strategy. SGIM-D and SGIM-IM are simpler versions where the active learner makes fewer choices about its sampling strategy.

We first illustrated in chapter 2, SGIM for a simple case with discrete spaces. We implemented an active learning algorithm for the humanoid robot iCub to actively choose data collection modes. We showed that **the active strategic learner chooses a good sampling strategy and efficiently gathers information** to recognise 3D objects by manipulation.

In chapter 3, we investigated more precisely about the relationship between autonomous intrinsically motivated data collection mode and socially guided data collection mode. We built an agent which has to learn how to produce a wide variety of outcomes, that can use both modes passively. This first architecture was called Socially Guided Intrinsic Motivation by Demonstration (SGIM-D). We showed that **the combination of the two modes improves the performance for motor control learning, in terms of accuracy but mostly in terms of speed**. The learner ends up with better results but also learns faster. This difference is all the greater as the environment is large. Our analysis shows that the learner thus **benefits from demonstrations which guide it to interesting subspaces of the outcome and the policy spaces**, and in the meanwhile the learner also benefits from **self-exploration to gain in accuracy in the absence of demonstrations and compensate for correspondence problems or the sparsity of demonstration sets**.

In chapter 4, we investigated how this learner can decide strategically to interact with teachers, to be a fully active learner choosing a sampling mode. We first designed an active learning algorithmic architecture for two preset sampling modes, called Socially Guided Intrinsic Motivation with Interactive learning at the Meta level (SGIM-IM). SGIM-IM chooses the most useful data collection mode based on competence progress measures. We showed that based on measures of competence progress, it can **choose when to request for demonstrations**, in two deterministic and stochastic environments.

Then, we extended its prerogatives by increasing the number of possible sampling modes, and thus allowing it to **choose who to imitate and what to imitate**, when several teachers are available, and when it

Summary

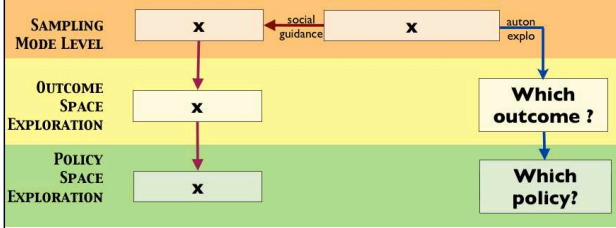
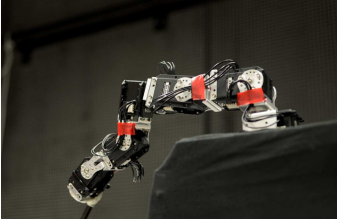
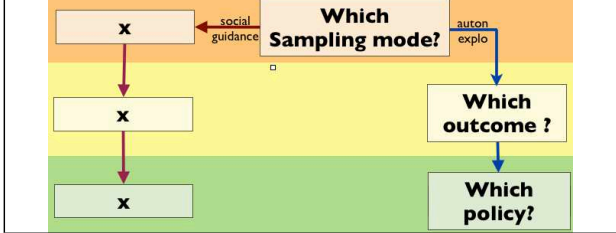
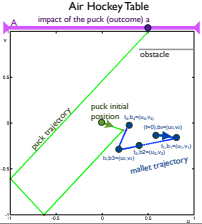
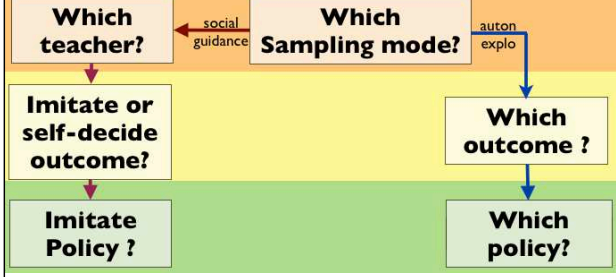
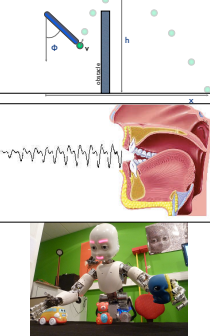
| Active learning | Experimental setup | Results |
|---|---|---|
| <p>SGIM-D</p>  | <p>cf. chapter 3</p>  | <ul style="list-style-type: none"> - learns as well or better precision, more reliably, faster - uses demonstrations to bias its search in the policy and outcome spaces - uses self-exploration to overcome correspondence problems - uses self-exploration to compensate sparsity of demonstration set |
| <p>SGIM-IM</p>  | <p>cf. section 4.1</p>  | <ul style="list-style-type: none"> - interactive learning - self-adjust the timing of requests for help (to the cost of a request for demonstration) - tested on deterministic and stochastic environments |
| <p>SGIM-ACTS</p>  | <p>cf. section 4.3</p>  <p>cf. chapter 5</p> <p>cf. chapter 1</p> | <ul style="list-style-type: none"> - interactive learning with several teachers - learn several types of tasks - tested in continuous and discrete spaces - tested in simulation and physical robot - model for child development |

Figure 6.1.1: Three algorithmic architectures are presented with various illustrative experiments. The details of each algorithm and experiment can be read in the corresponding section or chapter. Each algorithmic architecture allows the agent to take active control of various aspects of its learning strategy. Each of the results presented allow us to present aspects of the advantages for a fully active system, that can decide on all aspects of its learning strategy.

can choose to reproduce the demonstrated action or the demonstrated outcome. This third version is called Socially Guided Intrinsic Motivation with Active Choice of Teacher and Strategy (SGIM-ACTS). We showed through an experiment where a robotic arm can choose between several sampling modes, several teachers and several kinds of outcomes, that it can make these choices coherently to learn more accurately and faster. Thus, the same principle of competence-based intrinsic motivation used in a hierarchical algorithmic architecture for goal-oriented exploration by social guidance or autonomous exploration led to a robotic learner which can decide on many important questions of its interaction with the physical and social environment: what and how to learn; what, how, when and whom to imitate.

Finally, on a vocalisation experiment, we illustrated the algorithmic architecture SGIM with a different implementation, using a probabilistic framework. Most of all, we show that the learning agent can **self-organise its exploration**. A coherent **developmental sequence** could thus emerge, without any external clock or preset specification.

6.1.3 RESULT

We have thus designed a robot learner for interactive goal-babbling, which algorithmic architecture **connects the intrinsically motivated and socially guided data collection modes at each level of its hierarchical structure on 3 levels: for the exploration of the policy space, the outcome space, and the different modes**. It chooses a data collection strategy for online learning, based on trial-and-error and competence progress measures. Its exploration allows an improvement of its competence, but also the generalisation over its experience to achieve new outcomes. We showed better precision and faster learning of SGIM compared to agents learning with random exploration or with a single data collection mode. Our learner is characterised by its motivation for learning, its drive to improve, its ability to seek out the expertise of others and its robustness to bad demonstrators. We devised an algorithmic architecture for efficient learning and used our system to model infant development.

6.1.4 TAKE-AWAY MESSAGE

In conclusion, **combining intrinsic motivation and social guidance offers a good algorithmic architecture for life-long adaptive learning because it learns faster and with better precision to produce a wide range of outcomes**. While such a system benefits from the human knowledge to bias its exploration, at the same time, it learns the structure of its physical environment and the specialities of its human teachers. This self-structuring of the learning process sees the emergence of developmental sequences, similar to those observed in child development.

6.2 OUR APPROACH, ITS LIMITATIONS AND EXTENSIONS

6.2.1 MAIN CONTRIBUTION

We devised an algorithmic architecture for efficient learning and tested our algorithms to model infant development.

The algorithmic architecture we designed has been tested in several experimental setups, both in simulation and with physical robots, both for discrete and continuous spaces, both in deterministic and stochastic environments. It has also been used with different knowledge representations and learning algorithms. It also proposes an explanation to the bootstrapping effect by analysing the properties and effects of demonstrations on the exploratory behaviour. We highlighted the bias introduced by social guidance for the exploration of both the outcome and the policy spaces, and showed the robustness of the system to the teacher’s quality.

We developed an advanced **technique to combine learning by social guidance and intrinsic motivation**, for life-long learning of multiple skills. This combination has been designed in a more general context of **strategic learning**, where the learning agent chooses how to learn best between different learning modes. Our approach enabled the learner to decide online about its interaction with its physical and social environment: **what and how to learn; what, when, how and whom to imitate**.

6.2.2 ORIGINALITY OF OUR APPROACH

We addressed life-long learning by combining imitation learning and autonomous exploration because:

- as detailed in chapter 1, we take inspiration from biological systems. It has been observed that they learn both by social guidance and by autonomous exploration. Psychological studies have theorised the playful and exploratory behaviour of children as intrinsic motivation.
- as the search space is too large to be exhaustively explored within a life-time, a way of constraining the search is needed. [Baranes and Oudeyer \(2013\)](#) have proposed maturational constraints as a physical factor for constraining the search space through time, where a clock increases gradually the search space. Other approaches explore first the policy space to identify the interesting dimensionalities of the outcome space, to be able to restraint their exploration of the outcome space to these dimensions. In our case, we used human knowledge in the form of demonstrations to bias our exploration of the outcome and policy spaces.

To our knowledge, we devised the **first interactive learning system which combines intrinsic motivation and social guidance**. With that respect, our work resembles the approaches of ([Thomaz, 2006](#); [Thomaz and Breazeal, 2008](#)). A main difference lies in the fact that [Thomaz \(2006\)](#) and ([Thomaz and Breazeal, 2008](#)) used a symbolic representation of the environment, whereas we used continuous spaces to describe our non-preset environment. Their robot uses a discrete repertoire of preset actions to perform a discrete number of tasks, whereas our system can evolve in high-dimensional continuous policy and outcome spaces, and has to deal with the curse of dimensionality.

Our system learns to perform various outcomes in non-preset environments and in high-dimensional and continuous spaces. With that respect, our environments resemble those of ([Kober et al., 2012](#); [da Silva et al., 2012](#)) who use demonstrations to make robots learn complex tasks then leverage regression techniques to produce novel outcomes close to those previously learnt. Nevertheless, in these studies, the demonstrations are carefully chosen by an engineer beforehand without any active learning method, and the robot can not learn to reach novel outcomes that are far away from these demonstrations.

This system is also the first system tested in experiments to actively decide what and how to learn; and what, when, how and whom to imitate. With that respect, our strategic learning system resembles the active imitation learning algorithm developed in ([Shon et al., 2007](#)), which also decides when and theoretically whom

to imitate. Nevertheless, [Shon et al. \(2007\)](#) did not test their system in the case there are several teachers. Furthermore, their system only learns how to produce a single outcome in discrete environments.

6.2.3 COMPLEMENTARY STUDIES

In our work, we have build a meta algorithmic architecture. We have instantiated the architectures with two implementations, and both showed similar behaviours. Nevertheless, no study has yet been conducted to **explicit the influence on the algorithm of different choices of learning algorithms, , model representation, policy encoding, or of the various parameters.**

For instance, our implementations used for goal-oriented policy optimisation two instances: (1) a simulated annealing method based on the Nelder-Mead local optimisation algorithm, and (2) a probabilistic method using Gaussian Mixture Models. Nevertheless, it would be of interest to measure the effect of other single goal learning algorithms such as natural actor-critic architectures ([Peters and Schaal, 2008](#)), path integral approaches ([Theodorou et al., 2010](#)), advanced Black Box optimisation techniques ([Stulp and Sigaud, 2012](#)), or CMA-ES ([Hansen and Ostermeier, 2001](#))

Second, the actions used in our experiments were coded as parametrized motor primitives, where parameters determine the amplitude and (in some experiments) the timing of motor commands. They constitute constraints that reduce the state spaces in a reasonable way to make learning tractable in high dimensional action space. They were general enough and allowed a wide range of actions to be used in all our experiments. These constraints we introduced also reflect the difference in dynamics and range of movement that can occur between human teachers and robotic learners. They allow us to study the effect of this correspondence problem in section 3.5. Nevertheless, the number of parameters and thus the maximum complexity of the movement is predetermined, and good solutions could potentially be eliminated. It would be of interest to study the impact of different movement encodings. Another possibility could be dynamical movement primitives (DMP), a line of research for modelling attractor behaviours of autonomous nonlinear dynamical systems with the help of statistical learning techniques. They have first been presented in rhythmic movement task ([Schaal et al., 1996](#)), but they can also encode discrete movements about every DOF. In that sense, our movement encoding can be considered as a simplified version of the DMPs. A study of interest would be to use DMPs and measure the effect on the learning process.

6.2.4 LIMITATIONS AND EXTENSIONS

Nevertheless, our work bears some limitations. For instance, we have only considered in our experiments a small number of teachers and a small number of kinds of outcome, albeit an infinite number of outcomes that lie in continuous spaces. Therefore, complementary experiments should study **whether SGIM scales to higher numbers of teachers and kinds of outcome.** Theoretically, SGIM’s formalisation and framework can handle such cases. The method should apply to domestic or industrial robots who usually interact with a finite number of teachers. Nevertheless, experiments need be carried out to test this scalability property.

Likewise, the algorithm is meant to tackle learning problems in the physical, real world by addressing problems of stochasticity, redundancy and high-dimensionality. Nevertheless, in the experiments described in this thesis, we only used physical robots for kinaesthetic demonstration to robots in a simulator like in the fishing experiment (chapter 3), or we only used a physical robot to learn a simple discrete mapping, as

was the case of the robot iCub learning learning to recognise objects. The other experiments, despite the complexity and high-dimensionality of their environments, were only carried out in simulators. On-going experiments are testing SGIM on a physical robot learning to use a fishing rod, as shortly described in section 3.6.

In our work, the agent chooses between preset sampling modes, which **does not allow the learning of new sampling modes**. Nevertheless, an extension can improve the way we consider sampling modes. These could evolve, and the agent could adapt the different modes according to its experience. In this case, his sampling mode space could be parameterised by continuous variables. In this case, the mode parameter space would be a union of continuous spaces, like its policy and outcome spaces. Thus its choice of mode parameters would bear strong resemblance to its choice of focus which we studied in this thesis.

This point is related to the strong supposition from our work that imitation modes, self-exploration modes, and learning algorithms pre-exist/are preprogrammed in our agent. Our framework do not allow any of these to evolve. Studies like (Andry et al., 2004) propose a model linking the development of sensori-motor and imitation capabilities. Our work also takes the strong point of view that imitation is a means of learning. Yet, imitation also has the function of communication. Nadel et al. (2004) suggests that imitation is not only linked with learning, but can also be linked to the fact of being imitated.

Even though our framework allows theoretically to address the problem of **changing environments**, we have not tackled the problem specifically. Indeed, our formalism computes density probabilities with respect to the policies and outcomes, but also context. This measure of the context allows to take into account changing environments, providing a means of comparison between different contexts. Nevertheless, in our experiments, our robots have always started from the same state, and we have not investigated the effect of changes in context. A system with SGIM using a learning algorithm which favours more recent data than older data can be tested to study its ability to adapt to changing environments.

Another point for tackling contexts is the determination within the sensory inputs of objects of attention. We have been specifying an outcome space to indicate to the learner what kind of skills we wish it to acquire: fishing skills, placing skills, throwing skills. **It would be interesting to let the learner discover which dimensionalities of its sensory inputs constitute an interesting outcome space**. I believe this question is one of the most interesting extensions of SGIM to make the robot flexible and adaptive. This would require a longer learning time, just as a baby learns its own proprioception before exploring more visible tasks that we consider interesting or useful. But on the other hand, it should give the learner the opportunity to discover richer and more complete repertoires of skills, that could be far from the programmer's mind at first. The main limitation of such an attempt is the high-dimensionality of the sensory space of the blooming and buzzing world. Nevertheless I believe such a learning is possible even in high-dimensionality, as we have shown that SGIM can be robust in high-dimensional spaces and in outcome spaces much larger than its reachable space.

The last immediate extension, and most interesting to our mind is the **building of macro actions to accomplish more and more complex tasks**. In our work so far, we have only considered policies encoded by single motor primitives, and have not considered combinations of actions to produce more complex outcomes. For instance, learning to control one's body is a first step to the learning of how to touch objects, and afterwards to learning how to grasp objects, and finally to learning how to lift objects. Each step involves the reuse of policies to perform easy tasks, then its extension to reach a more difficult outcome. Such work could be done by using for instance the options framework suggested in (Konidaris and Barto,

2009).

6.2.5 IMPACT

This thesis proposes a new mechanism for life-long learning for building artificial leaning agents and a mechanism to understand biological agents. The meta-level mechanism relies on social guidance to restrain the exploration to some subspaces; and on intrinsic motivation to expand the exploration around the subspaces already indicated by teachers, and choose a sampling strategy that improves the competence progress of the system. The principles of the strategic data collection can be linked to various fields.

The learning of skills of growing complexity, coupled with the structuring properties into logical developmental sequences of our learning architecture (chapter 5), gives promises of learning of complex chains of skills without having to specify each step of the learning. Albeit a few indications from human users, the robot should be able to discover by itself how to fragment the complex skill into first easy ones and then tackle increasingly difficult ones. Such work should be highly interesting to be coupled with planning. Indeed, where classical planning approaches fail, especially in unforeseen configurations, exploration and learning should help the agent to recover from dead ends. Reversely, planning can provide a necessary framework for building complex skills for robots. A combination of exploratory learning and planning could therefore enable the robot to realise more and more complex skills.

The learning of various skills as well as the structuring of the learning into developmental sequences could also be investigated more, in terms of cognitive science, or more specifically of in terms of neuroscience. If the learning of new skills reuses policies or structures that have been learnt, such reuse could be observable in the brain or plausible for neuronal models. Can we observe recruitments of neuronal circuits for the learning of new skills? How do these neuronal circuits evolve, how are they copied or synthesised with the agent's experience? We could model neuronal replication patterns of agents learning by multi-modal exploration, and build a neurally plausible cognitive model for agents exploring their environments both autonomously and with social guidance. Such a combination between neuroscience and robotic learning would lead to the design of a cognitive architecture based on neurobiology to model how learning agents explore their environments. It would be an inter-disciplinary investigation of the principles of learning in natural and artificial systems, towards a biologically plausible basis for human problem solving and at the same time a practical mechanism for learning by artificial systems.

Lastly, we have been designing a data collection method for robot learners. Nevertheless, this data collection method is applicable in more general cases than robotics. The advantage of our method is all the more acute when the cost of data acquisition is high. In the case a data acquisition comes at a high cost in time, material, energy, the learning agent should decide first what kind of data would give it more information, improve its control or improve its model. For instance, this is also the case of recommender systems. In order to give good recommendations, the system should acquire two kinds of data. First, it needs to get enough data both on the user to be able to build a user profile and adjust its recommendations accordingly. Second, it also needs to get a better model about the objects of recommendations (books, movies, jobs ...) to be able know which adequate recommendations to give to specific user profiles. Nevertheless, in order to be user-friendly, the system can only ask a few questions to users, and should instead attempt to get more of these two kinds of data in a seamless way, such as by measuring the effectiveness of its recommendations. Such a recommendation entails a cost in terms of time, but also in terms of trustworthiness, for a bad

recommendation would undermine the user's confidence in the system. That is why, determining the right question to ask to the user or the right kind of data to acquire becomes crucial. We thus believe that our methods of data collection could inspire other fields than robotics.

Therefore, further studies can improve SGIM to be adaptive to changing environment, and to be able to learn more open-ended skills, with less specification from the programmer, and more complex skill-chaining. It also has potential to impact other fields than robotic learning, and hopefully can contribute to practical applications as well as more fundamentally change our perception of how artificial and biological agents can learn and evolve in our world.

References

- Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, pages 1–8, 2004.
- B. Akgun, M. Cakmak, J.W. Yoo, and Andrea L. Thomaz. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *International Conference on Human-Robot Interaction*, 2012.
- P. Andry, Ph. Gaussier, J. Nadel, and B. Hirsbrunner. Learning invariant sensorimotor behaviors: A developmental approach to imitation mechanisms. *Adaptive behavior*, 12(2):117–138, october 2004.
- Brenna D. Argall, B. Browning, and Manuela Veloso. Learning robot motion control with demonstration and advice-operators. In *In Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 399–404. IEEE, September 2008.
- Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469 – 483, 2009. ISSN 0921-8890. doi: 10.1016/j.robot.2008.10.024.
- Brenna D. Argall, B. Browning, and Manuela Veloso. Teacher feedback to scaffold and refine demonstrated motion primitives on a mobile robot. *Robotics and Autonomous Systems*, 59(3-4):243–255, 2011.
- Minoru Asada, Koh Hosoda, Yasuo Kuniyoshi, Hiroshi Ishiguro, Toshio Inui, Yuichiro Yoshikawa, Masaki Ogino, and Chisato Yoshida. Cognitive developmental robotics: a survey. *IEEE Transactions on Autonomous Mental Development*, 1(1):12–34, 2009.
- C.G. Atkeson, Moore Andrew, and Schaal Stefan. Locally weighted learning. *AI Review*, 11:11–73, April 1997.
- Paul Bach-y Rita. Sensory plasticity. *Acta Neurologica Scandinavica*, 43(4):417–426, 1967. ISSN 1600-0404. doi: 10.1111/j.1600-0404.1967.tb05747.x.
- G. Baldassarre. What are intrinsic motivations? a biological perspective. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2, pages 1–8. IEEE, 2011.
- Gianluca Baldassarre and Marco Mirolli, editors. *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer, 2013a.
- Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. *The Journal of Machine Learning Research*,, 5:255–291, 2004.
- A. Baranes and P-Y. Oudeyer. Intrinsically motivated goal exploration for active motor learning in robots: a case study. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010), Taipei, Taiwan*, 2010a.

- Adrien Baranes and Pierre-Yves Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013.
- Andrew G. Barto, S. Singh, and N Chentanez. Intrinsically motivated learning of hierarchical collections of skills. In *ICDL International Conference on Developmental Learning*, pages 112–119, 2004b.
- Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006.
- D. Berlyne. *Structure and Direction in Thinking*. New York: John Wiley and Sons, Inc., 1965.
- D. E. Berlyne. A theory of human curiosity. *British Journal of Psychology*, 45:180–191, 1954.
- Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. *Handbook of Robotics*, chapter Robot Programming by Demonstration. Number 59. MIT Press, 2007.
- C.M. Bishop. Pattern recognition and machine learning. In *Information Science and Statistics*. Springer, 2007.
- Bruce Blumberg, Marc Downie, Yuri Ivanov, Matt Berlin, Michael Patrick Johnson, and Bill Tomlinson. Integrated learning for interactive synthetic characters. *ACM Trans. Graph.*, 21:417–426, July 2002. ISSN 0730-0301. doi: <http://doi.acm.org/10.1145/566654.566597>.
- David (2012). Boersma, Paul & Weenink. Praat: doing phonetics by computer [computer program]. <http://www.praat.org/>, 2012.
- Paul Boersma. *Functional phonology: Formalizing the interactions between articulatory and perceptual drives*. Holland Academic Graphics, 1998.
- Cynthia Breazeal and B. Scassellati. Robots that imitate humans. *Trends in Cognitive Sciences*, 6(11): 481–487, 2002.
- R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–160, 1991.
- Maya Cakmak and Andrea L. Thomaz. Optimality of human teachers for robot learners. In *IEEE International Conference on Development and Learning*, volume 4, 2010.
- Maya Cakmak, Nick DePalma, Andrea L. Thomaz, and Rosa Arriaga. Effects of social exploration mechanisms on robot learning. In *The 18th IEEE International Symposium on Robot and Human Interactive Communication, 2009. RO-MAN 2009.*, pages 128–134. IEEE, 2009.
- Maya Cakmak, C. Chao, and Andrea L. Thomaz. Designing interactions for robot active learners. *Autonomous Mental Development, IEEE Transactions on*, 2(2):108–118, 2010.
- Maya Cakmak and Manuel Lopes. Algorithmic and Human Teaching of Sequential Decision Tasks. *AAAI Conference on Artificial Intelligence*, 2012.
- Sylvain Calinon. *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009a. EPFL Press ISBN 978-2-940222-31-5, CRC Press ISBN 978-1-4398-0867-2.
- Sylvain Calinon. *Robot Programming by Demonstration*. CRC, 2009b.
- Sylvain Calinon, F. Guenter, and Aude Billard. On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics*, 37(2):286–298, 2007.

- J. Call and M. Carpenter. *Imitation in animals and artifacts*, chapter Three sources of information in social learning, pages 211–228. Cambridge, MA: MIT Press., 2002.
- Thomas Cederborg and Pierre-Yves Oudeyer. Imitating operations on internal cognitive structures for language acquisition. In *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012. ISBN 978-1-61284-866-2.
- Sonia Chernova and Manuela Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34(1):1, 2009.
- J.A. Clouse and P.E. Utgoff. *A teaching method for reinforcement learning*. University of Massachusetts at Amherst, Dept. of Computer and Information Science, 1992.
- David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.
- T.F. Coleman and Y. Li. On the convergence of reflective newton methods for large-scale nonlinear minimization subject to bounds. *Mathematical Programming*, 67(2):189–224, 1994.
- T.F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.
- Gergely Csibra. Teleological and referential understanding of action in infancy. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):447, 2003.
- Gergely Csibra and Gyorgy Gergely. Obsessed with goals: Functions and mechanisms of teleological interpretation of actions in humans. *Acta Psychologica*, 124(1):60 – 78, 2007. ISSN 0001-6918. doi: 10.1016/j.actpsy.2006.09.007. Becoming an Intentional Agent: Early Development of Action Interpretation and Action Control.
- M. Csikszentmihalyi. *Creativity: Flow and the Psychology of Discovery and Invention*. HarperCollins, 1997. ISBN 9780060928209.
- B.C. da Silva, G. Konidaris, and Andrew G. Barto. Learning parameterized skills. In *29th International Conference on Machine Learning (ICML 2012)*, 2012.
- Kerstin Dautenhahn and Chrystopher L. Nehaniv. *Imitation in Animals and Artifacts*. MIT Press, 2002.
- A. d’Avella, A. Portone, L. Fernandez, and F. Lacquaniti. Control of fast-reaching movement by muscle synergies combinations. *The Journal of Neuroscience*, 26(30):7791–7810, 2006.
- S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4): 357–366, August 1980. ISSN 0096-3518. doi: 10.1109/TASSP.1980.1163420.
- E.L. Deci and Richard M. Ryan. *Intrinsic Motivation and self-determination in human behavior*. Plenum Press, New York, 1985.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246.
- Yadin Dudai. The Restless Engram: Consolidations Never End. In Hyman, SE, editor, *ANNUAL REVIEW OF NEUROSCIENCE, VOL 35*, volume 35 of *Annual Review of Neuroscience*, pages 227–247. ANNUAL REVIEWS, 4139 EL CAMINO WAY, PO BOX 10139, PALO ALTO, CA 94303-0897 USA, 2012. ISBN 978-0-8243-2435-3. doi: {10.1146/annurev-neuro-062111-150500}.

- V. Fedorov. *Theory of Optimal Experiment*. Academic Press, Inc., New York, NY, 1972.
- David Filliat. A visual bag of words method for interactive qualitative localization and mapping. In *IEEE Int. Conf. on Robotics and Automation*, pages 3921–3926, 2007.
- Marc Andreas Freese. V-rep. URL <http://www.coppeliarobotics.com/>.
- Karl Friston, Rick A Adams, Laurent Perrinet, and Michael Breakspear. Perceptions as hypotheses: saccades as experiments. *Frontiers in psychology*, 3, 2012.
- Ph. Gaussier, S. Boucenna, and J. Nadel. Emotional interactions as a way to structure learning. In *Epigenetic Robotics*, pages 193–194. Lucs, 2007.
- James Jerome Gibson. *The ecological approach to visual perception*. Psychology Press, 1986.
- Albert Globus and Arnold B. Scheibel. The effect of visual deprivation on cortical neurons: A golgi study. *Experimental Neurology*, 19(3):331 – 345, 1967. ISSN 0014-4886. doi: [http://dx.doi.org/10.1016/0014-4886\(67\)90029-5](http://dx.doi.org/10.1016/0014-4886(67)90029-5).
- Gilbert Gottlieb. Experiential canalization of behavioral development: Theory. *Developmental Psychology*, 27(1):4, 1991.
- Jacqueline Gottlieb, Pierre-Yves Oudeyer, Manuel Lopes, and Adrien Baranes. Information-seeking, curiosity, and attention: computational and neural mechanisms. *Trends in cognitive sciences*, 2013.
- Daniel H Grollman and Odest Chadwicke” Jenkins. Incremental learning of subtasks from unsegmented demonstration. In *Intelligent Robots and Systems IROS 2010 IEEE/RSJ International Conference on*, pages 261–266, 2010. doi: 10.1.1.169.4242.
- F H Guenther, M Hampson, and D Johnson. A theoretical investigation of reference frames for the planning of speech movements. *Psychological Review*, 105(4):611–633, October 1998. ISSN 0033-295X. PMID: 9830375.
- Frank H. Guenther. Cortical interactions underlying the production of speech sounds. *Journal of Communication Disorders*, 39(5):350–365, September 2006. ISSN 0021-9924. doi: 10.1016/j.jcomdis.2006.06.013.
- Frank H Guenther, Satrajit S Ghosh, and Jason A Tourville. Neural modeling and imaging of the cortical interactions underlying syllable production. *Brain and language*, 96(3):280–301, 2006.
- N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- S. Hart. An intrinsic reward for affordance exploration. In *ICDL International Conference on Developmental Learning*, 06 2009.
- Richard Held and Alan Hein. Movement-produced stimulation in the development of visually guided behaviour. *Journal of comparative and physiological psychology*, 56(5):872–876, 1963.
- John H Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- I.S. Howard and P. Messum. Modeling the development of pronunciation in infant speech acquisition. *Motor Control*, 15(1):85–117, 2011.

- Hisashi Ishihara, Yuichiro Yoshikawa, Katsushi Miura, and Minoru Asada. How caregiver's anticipation shapes infant's vowel through mutual imitation. *Autonomous Mental Development, IEEE Transactions on*, 1(4):217–225, 2009.
- S. Ivaldi, N. Lyubova, D. G erardeaux-Viret, A. Droniou, S. M. Anzalone, M. Chetouani, D. Filliat, and O. Sigaud. Perception and human interaction for developmental learning of objects and affordances. In *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, Osaka, Japan, 2012.
- Serena Ivaldi, Sao Mai Nguyen, Natalia Lyubova, Alain Droniou, Vincent Padois, David Filliat, Pierre-Yves Oudeyer, and Olivier Sigaud. Object learning through active exploration. *Transactions on Autonomous Mental Development*, PP(99):1–1, 2013. ISSN 1943-0604. doi: 10.1109/TAMD.2013.2280614.
- W. James. *The principles of psychology*. Cambridge, MA: Harvard University Press, 1890.
- Mark Johnson. *The body in the mind: The bodily basis of meaning, imagination, and reason*. University of Chicago Press, 1987.
- D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box function. *J. Global Optim.*, 13(4):455–492, 1998.
- F. Kaplan and P-Y. Oudeyer. The progress-drive hypothesis: an interpretation of early imitation. In K. Dautenhahn and C. Nehaniv, editors, *Models and mechanisms of imitation and social learning: Behavioural, social and communication dimensions*. Cambridge University Press, 2005.
- Frederic Kaplan and Pierre-Yves Oudeyer. In search of the neural circuits of intrinsic motivation. *Frontiers in neuroscience*, 1(1):225, 2007.
- Frederic Kaplan, Pierre-Yves Oudeyer, Eniko Kubinyi, and A. Miklosi. Robotic clicker training. *Robotics and Autonomous Systems*, 38((3-4):197–206, 2002.
- Alexander S. Klyubin, Daniel Polani, and Chrystopher L. Nehaniv. Keep your options open: An information-based driving principle for sensorimotor systems. *PLoS ONE*, 3(12), 12 2008. doi: 10.1371/journal.pone.0004018.
- Jens Kober, Erhan Oztop, and Jan Peters. Reinforcement learning to adjust robot movements to new situations. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.
- Jens Kober, Andreas Wilhelm, Erhan Oztop, and Jan Peters. Reinforcement learning to adjust parametrized motor primitives to new situations. *Autonomous Robots*, pages 1–19, 2012. ISSN 0929-5593.
- N Koenig, L Takayama, and M Matari c. Communication and knowledge sharing in human-robot interaction and learning from demonstration. *Neural Netw*, 23(8-9):1104–1112, Oct-Nov 2010. doi: 10.1016/j.neunet.2010.06.005.
- G.D. Konidaris and Andrew G. Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. *Advances in Neural Information Processing Systems (NIPS)*, pages 1015–1023, 2009.
- Petar Kormushev, Sylvain Calinon, and Darwin G. Caldwell. Robot motor skill coordination with EM-based reinforcement learning. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 3232–3237, Taipei, Taiwan, October 2010.
- Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Advanced Robotics*, 25(5):581–603, 2011.

- Bernd J Kröger, Jim Kannampuzha, and Christiane Neuschaefer-Rube. Towards a neurocomputational model of speech production and perception. *Speech Communication*, 51(9):793–809, 2009.
- W. J. Krzanowski. *Principles of Multivariate Analysis: A User’s Perspective*. Oxford University Press, 1988.
- P. K. Kuhl. Early language acquisition: cracking the speech code. *Nature Reviews Neuroscience*, 2004.
- J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.
- M. Lopes and Pierre-Yves Oudeyer. Active learning and intrinsically motivated exploration in robots: Advances and challenges (guest editorial). *IEEE Trans. Aut. Mental Development*, 2(2):65–69, 2010.
- Manuel Lopes and Pierre-Yves Oudeyer. The Strategic Student Approach for Life-Long Exploration and Learning. In *IEEE Conference on Development and Learning / EpiRob*, San Diego, États-Unis, November 2012.
- Manuel Lopes, Francisco Melo, and Luis Montesano. Active learning for reward estimation in inverse reinforcement learning. *Machine Learning and Knowledge Discovery in Databases*, pages 31–46, 2009a.
- Manuel Lopes, Francisco Melo, Luis Montesano, and Jose Santos-Victor. *From Motor to Interaction Learning in Robots*, chapter Abstraction Levels for Robotic Imitation: Overview and Computational Approaches. Springer, 2009b.
- Manuel Lopes, Francisco S.Melo, Ben Kenward, and Jose Santos-Victor. A computational model of social-learning mechanisms. *Adaptive Behaviour*, 467(17), 2009c.
- Manuel Lopes, Francisco Melo, Luis Montesano, and Jose Santos-Victor. Abstraction levels for robotic imitation: Overview and computational approaches. In Olivier Sigaud and Jan Peters, editors, *From Motor to Interaction Learning in Robots*, volume 264 of *Studies in Computational Intelligence*, pages 313–355. Springer, 2010.
- Manuel Lopes, Thomas Cederborg, and Pierre-Yves Oudeyer. Simultaneous acquisition of task and feedback models. *Development and Learning (ICDL), 2011 IEEE International Conference on*, pages 1 – 7, 2011.
- M. Lungarella, Giorgio Metta, R. Pfeifer, and Giulio Sandini. Developmental robotics: a survey. *Connection Science*, 15(4):151–190, 2003.
- Natalia Lyubova and David Filliat. Developmental approach for interactive object discovery. In *Int. Joint Conf. on Neural Networks*, 2012.
- Ian Maddieson and Kristin Precoda. Updating UPSID. *The Journal of the Acoustical Society of America*, 86(S1):S19, November 1989.
- S. Maeda. Compensatory articulation during speech: Evidence from the analysis and synthesis of vocal tract shapes using an articulatory model. *Speech production and speech modelling*, pages 131–149, 1989.
- Olivier Mangin and Pierre-Yves Oudeyer. Learning the combinatorial structure of demonstrated behaviors with inverse feedback control. In Albert Ali Salah, Javier Ruiz-del Solar, Çetin Meriçli, and Pierre-Yves Oudeyer, editors, *HBU 2012. LNCS, vol. 7559*, pages 135–148. Springer, Heidelberg, 2012.
- Kevin Lee Markey. *The sensorimotor foundations of phonology: a computational model of early childhood articulatory and phonetic development*. PhD thesis, University of Colorado at Boulder, 1994.
- Kathryn Merrick and Mary Lou Maher. Motivated learning from interesting events: adaptive, multitask learning agents for complex environments. *Adaptive Behavior*, 17(1):7–27, 2009.

- B. Micusik and J. Kosecka. Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In *IEEE Int. Conf. on Computer Vision*, pages 625–632, 2009.
- Katsushi Miura, Yuichiro Yoshikawa, and Minoru Asada. Vowel acquisition based on an auto-mirroring bias with a less imitative caregiver. *Advanced Robotics*, 26(1-2):23–44, 2012.
- Clément Moulin-Frier and Pierre-Yves Oudeyer. Curiosity-driven phonetic learning. In *International Conference on Development and Learning, Epirob, San Diego, USA*, 2012.
- Clément Moulin-Frier and Pierre-Yves Oudeyer. The role of intrinsic motivations in learning sensorimotor vocal mappings: a developmental robotics study. In *Proceedings of Interspeech*, page In press, Lyon, France, 2013a.
- Clément Moulin-Frier and Pierre-Yves Oudeyer. Exploration strategies in developmental robotics: a unified probabilistic framework. In *International Conference on Development and Learning, Epirob, Osaka, Japan*, 2013b.
- Clément Moulin-Frier, Jean-Luc Schwartz, Julien Diard, and Pierre Bessière. *Primate communication and human language: Vocalisations, gestures, imitation and deixis in humans and non-humans*, chapter Emergence of articulatory-acoustic systems from deictic interaction games in a "Vocalize to Localize" framework. Advances in Interaction Studies' series by John Benjamins Pub. Co., 2011.
- Clement Moulin-Frier, Sao Mai Nguyen, and Pierre-Yves Oudeyer. Self-organization of early vocal development in infants and machines: The role of intrinsic motivation. *Frontiers in Cognitive Science*, accepted.
- M. Muja and D.G. Lowe. Fast approximate nearest neighbors with automatic algorithm. In *International Conference on Computer Vision Theory and Applications*, 2009.
- J. Nadel, A. Revel, P. Andry, and Ph. Gaussier. Toward communication: First imitations in infants, low-functioning children with autism and robots. *Interaction Studies*, 5(1):45–74, 2004.
- L. Natale, F. Nori, G. Metta, M. Fumagalli, S. Ivaldi, U. Pattacini, M. Randazzo, A. Schmitz, and G. G. Sandini. *Intrinsically motivated learning in natural and artificial systems*, chapter The iCub platform: a tool for studying intrinsically motivated learning. Springer-Verlag, 2012.
- Chrystopher L Nehaniv and Kerstin Dautenhahn. *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*. Cambridge Univ. Press, Cambridge, March 2007.
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Properties for efficient demonstrations to a socially guided intrinsically motivated learner. In *21st IEEE International Symposium on Robot and Human Interactive Communication*, 2012a.
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Socially guided intrinsically motivated learner. In *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*, pages 1–2, nov. 2012b. doi: 10.1109/DevLrn.2012.6400809.
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Interactive learning gives the tempo to an intrinsically motivated robot learner. In *IEEE-RAS International Conference on Humanoid Robots*, 2012c.
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner. *Paladyn Journal of Behavioural Robotics*, 3(3):136–146, 2012d. ISSN 2080-9778. doi: 10.2478/s13230-013-0110-z.

- Sao Mai Nguyen and Pierre-Yves Oudeyer. Whom will an intrinsically motivated robot learner choose to imitate from? In Joanna Szufnarowska, editor, *Proceedings of the Post-Graduate Conference on Robotics and Development of Cognition*, pages 32–35, September 2012e. doi: 10.2390/biecoll-robotdoc2012-12.
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Socially guided intrinsic motivation. In *Spring School on Developmental Robotics and Cognitive Bootstrapping*, 2012f.
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Socially guided intrinsic motivation for robot learning of motor skills. *Autonomous Robots*, pages 1–22, 2013a. doi: 10.1007/s10514-013-9339-y.
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Strategic robot learner for interactive goal-babbling. In *Reinforcement Learning and Decision Making*, Princeton, New Jersey, U.S.A, 2013b.
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Strategic robot learner for interactive goal-babbling. In *Workshop in Active Learning in Robotics at Robotic Science and Systems*, Berlin, Germany, 2013c.
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Strategic robot learner for interactive goal-babbling. In *Workshop in Hierarchical Learning at Robotic Science and Systems*, Berlin, Germany, 2013d.
- Sao Mai Nguyen, Masaki Ogino, and Minoru Asada. Real-time face swapping as a tool for understanding infant self-recognition. In *Proceedings of the 10th International Conference on Epigenetic Robotics*, pages pp.171–172, Glumslöv, Sweden, 2010.
- Sao Mai Nguyen, Adrien Baranes, and Pierre-Yves Oudeyer. Bootstrapping intrinsically motivated learning with human demonstrations. In *IEEE International Conference on Development and Learning*, Frankfurt, Germany, 2011a.
- Sao Mai Nguyen, Adrien Baranes, and Pierre-Yves Oudeyer. Constraining the size growth of the task space with socially guided intrinsic motivation using demonstrations. In *IJCAI Workshop on Agents Learning Interactively from Human Teachers*, 2011b.
- Sao Mai Nguyen, Serena Ivaldi, Natalia Lyubova, Alain Droniou, Damien Gerardeaux-Viret, David Filliat, Vincent Padois, Olivier Sigaud, and Pierre-Yves Oudeyer. Learning to recognize objects through curiosity-driven manipulation with the icub humanoid robot. In *IEEE International Conference on Development and Learning - Epirob*, 2013.
- M.N. Nicolescu and M.J. Mataric. Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 241–248. ACM, 2003.
- D. Kimbrough Oller. *The Emergence of the Speech Capacity*. Mahwah, NJ: Lawrence Erlbaum Associates, 2000. ISBN 9780805826296.
- Pierre-Yves Oudeyer. The self-organization of speech sounds. *Journal of Theoretical Biology*, 233(3):435–449, April 2005. ISSN 0022-5193. doi: 10.1016/j.jtbi.2004.10.025.
- Pierre-Yves Oudeyer. Developmental constraints on the evolution and acquisition of sensorimotor skills. *Habilitation a Diriger des Recherches*, 2011a.
- Pierre-Yves Oudeyer. *Encyclopedia of the Sciences of Learning*, chapter Developmental Robotics. Springer Reference Series, Springer, 2011b.
- Pierre-Yves Oudeyer and Frederic Kaplan. Discovering communication. *Connection Science*, 18(2):189–206, 06 2006.

- Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in Neurorobotics*, 2007.
- Pierre-Yves Oudeyer, Frederic Kaplan, and Verena Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286, 2007. doi: 10.1109/TEVC.2006.890271.
- Pierre-Yves Oudeyer, Adrien Baranes, and Frederic Kaplan. *Intrinsically Motivated Cumulative Learning in Natural and Artificial Systems*, chapter Developmental constraints on intrinsically motivated skill learning: towards addressing high-dimensions and unboundedness in the real world. Springer, 2013.
- Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008.
- Rolf Pfeifer and C. Scheier. *Understanding Intelligence*. MIT Press, Cambridge, MA, 1999.
- Jean Piaget and Margaret Trans Cook. *The origins of intelligence in children*. WW Norton & Co, 1952.
- Justus H. Piater. *The Origins of Intelligence in Childhood*. International University Press, 1952.
- Z W Pylyshyn. Visual indexes, preconceptual objects, and situated vision. *Cognition*, 80:127–158, 2001.
- G. Qi, X. Hua, Y. Rui, J. Tang, and H. Zhang. Two-dimensional active learning for image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- Vilayanur S Ramachandran. *Phantoms in the brain: Probing the mysteries of the human mind*. Harper Perennial, 1999.
- Antons Rebguns, Daniel Ford, and Ian Fasel. Infomax control for acoustic exploration of objects by a mobile robot. In *AAAI Conference on Artificial Intelligence*, pages 22–28, 2011.
- Roi Reichart, Katrin Tomanek, Udo Hahn, and Ari Rappoport. Multi-task active learning for linguistic annotations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. Citeseer, 2008.
- M. Rolf, J. Steil, and M. Gienger. Gobar babbling permits direct learning of inverse kinematics. *IEEE Trans. Autonomous Mental Development*, 2(3):216–229, 09/2010 2010.
- N. Roy and A. McCallum. Towards optimal active learning through sampling estimation of error reduction. In *Proc. 18th Int. Conf. Mach. Learn.*, volume 1, pages 143–160, 2001.
- Richard M. Ryan and Edward L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25(1):54 – 67, 2000a.
- S. Schaal, D. Sternad, and C. G. Atkeson. One-handed juggling: A dynamical approach to a rhythmic movement task. (2):165–183, 1996.
- Stefan Schaal, A Ijspeert, and Aude Billard. Computational approaches to motor learning by imitation. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 358(1431), 03 2003.
- M Schembri, M Mirolli, and G Baldassarre. Evolving childhood’s length and learning parameters in an intrinsically motivated reinforcement learning robot. In *Proceedings of the seventh international conference on epigenetic robotics*, volume 134, pages 141–148. Lund: Lund University, 2007.

- J. Schmidhuber. Curious model-building control systems. In *Proc. Int. Joint Conf. Neural Netw.*, volume 2, pages 1458–1463, 1991a.
- J. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In J. A. Meyer and S. W. Wilson, editors, *Proc. SAB'91*, pages 222–227, 1991b.
- J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.
- J.-L. Schwartz, L.-J. Boë, N. Vallée, and C. Abry. Major trends in vowel system inventories. *Journal of Phonetics*, 25(3):233–253, 1997.
- A.P. Shon, D. Verma, and Rajesh PN Rao. Active imitation learning. In *American Association for Artificial Intelligence*, volume 22, page 756. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.
- B. Sigismund. *Child language: a book of readings*, chapter Kind und Welt, pages 17–18. Englewood Cliffs, NJ: Prentice-Hall. (Original work published in 1856), 1971.
- J. Sivic and A. Zisserman. Video google: Text retrieval approach to object matching in videos. In *Int. Conf. on Computer Vision*, volume 2, pages 1470–1477, 2003.
- Alan Slater and Michael Lewis, editors. *Introduction to infant development*. Oxford University Press, 2006.
- W.D. Smart and L.P. Kaelbling. Effective reinforcement learning for mobile robots,. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3404–3410., 2002.
- Linda B. Smith and Esther Thelen. Development as a dynamic system. *Trends in Cognitive Sciences*, 7(8): 343 – 348, 2003. ISSN 1364-6613. doi: [http://dx.doi.org/10.1016/S1364-6613\(03\)00156-6](http://dx.doi.org/10.1016/S1364-6613(03)00156-6).
- O. Sporns and M Lungarella. Evolving coordinated behavior by maximizing information structure. In L. Rocha and al., editors, *Proceedings of the 10th International Conference on Artificial Life (Alife)*, 2006.
- Rupesh Kumar Srivastava, Bas R Steunebrink, and Jürgen Schmidhuber. First experiments with powerplay. *Neural Networks*, 2013.
- Andrew Stout and Andrew G Barto. Competence progress intrinsic motivation. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, pages 257–262. IEEE, 2010.
- Freek Stulp and Stefan Schaal. Hierarchical reinforcement learning with movement primitives. In Andrej Gams, editor, *11th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2011), Bled, Slovenia, October 26-28, 2011*, pages 231–238. IEEE, IEEE, 2011. ISBN 978-1-61284-866-2.
- Freek Stulp and Olivier Sigaud. Policy Improvement Methods: Between Black-Box Optimization and Episodic Reinforcement Learning. 34 pages, October 2012.
- H. Taine. *Child language: a book of readings*, chapter Acquisition of language by children, pages 20–26. Englewood Cliffs, NJ: Prentice-Hall. (Original work published in 1856), 1971.
- E. Thelen and L.B. Smith. *A Dynamic Systems Approach to the Development of Cognition and Action*. A Bradford book. MIT Press, 1996. ISBN 9780262700597.
- E. Theodorou, J. Buchli, and S. Schaal. reinforcement learning of motor skills in high dimensions: a path integral approach. In *robotics and automation (icra), 2010 ieee international conference on*, pages 2397–2403, 2010.

- Andrea L. Thomaz. *Socially Guided Machine Learning*. PhD thesis, MIT, 5 2006.
- Andrea L. Thomaz and Cynthia Breazeal. Experiments in socially guided exploration: Lessons learned in building robots that learn with and without human teachers. *Connection Science*, 20 Special Issue on Social Learning in Embodied Agents(2-3):91–110, 2008.
- Edward L Thorndike. Animal intelligence: An experimental study of the associative processes in animals. *The Psychological Review: Monograph Supplements*, 2(4):i, 1898.
- S. Thrun. *Handbook of Brain Science and Neural Networks*, chapter Exploration in active learning. Cambridge, MA: MIT Press,, 1995.
- M. Tomasello and M. Carpenter. Shared intentionality. *Developmental Science*, 10(1):121–125, 2007.
- Paul Van Geert. A dynamic systems model of cognitive and language growth. *Psychological review*, 98(1): 3, 1991.
- C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.
- F. Varela, E. Thompson, and E. Rosch. *The embodied mind : cognitive science and human experience*. MIT Press, 1991.
- Deepak Verma and Rajesh Rao. Goal-based imitation as probabilistic inference over graphical models. In *Advances in NIPS 18*, 2006.
- Marilyn M Vihman, Charles A Ferguson, and Mary Elbert. Phonological development from babbling to speech: Common tendencies and individual differences. *Applied Psycholinguistics*, 7(1):3–40, 1986.
- A. S Warlaumont. Prespeech motor learning in a neural network using reinforcement. *Neural Networks*, 38: 64–95, 2013.
- A.S. Warlaumont. A spiking neural network model of canonical babbling development. In *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*, pages 1–6, 2012. doi: 10.1109/DevLrn.2012.6400842.
- E. Weiss and M. Flanders. Muscular and postural synergies of the human hand. *J. Neurophysiol.*, 92: 523–535, 2004.
- R. White. Motivation reconsidered: The concept of competence. *Psychological review*, (66):297–333, 1959.
- S. Whitehead. A study of cooperative mechanisms for faster re- inforcement learning. Tech. rep. tr-365, Univ. Rochester, Rochester, NY, 1991.
- Andrew Whiten. Primate culture and social learning. *Cognitive Science*, 24(3):477–508, 2000.
- T Xu, C Yu, and Linda Smith. It’s the child’s body: The role of toddler and parent in selecting toddler’s visual experience. 2011.

List of Publications

6.3 JOURNAL PAPERS

- Serena Ivaldi, Sao Mai Nguyen, Natalia Lyubova, Alain Droniou, Vincent Padois, David Filliat, Pierre-Yves Oudeyer, and Olivier Sigaud. Object learning through active exploration. *Transactions on Autonomous Mental Development*, PP(99):1–1, 2013. ISSN 1943-0604. doi: 10.1109/TAMD.2013.2280614
- Clement Moulin-Frier, Sao Mai Nguyen, and Pierre-Yves Oudeyer. Self-organization of early vocal development in infants and machines: The role of intrinsic motivation. *Frontiers in Cognitive Science*, accepted
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner. *Paladyn Journal of Behavioural Robotics*, 3(3):136–146, 2012d. ISSN 2080-9778. doi: 10.2478/s13230-013-0110-z
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Socially guided intrinsic motivation for robot learning of motor skills. *Autonomous Robots*, pages 1–22, 2013a. doi: 10.1007/s10514-013-9339-y

6.4 CONFERENCE PAPERS WITH PROCEEDINGS

- Sao Mai Nguyen and Pierre-Yves Oudeyer. Strategic robot learner for interactive goal-babbling. In *Reinforcement Learning and Decision Making*, Princeton, New Jersey, U.S.A, 2013b
- Sao Mai Nguyen, Serena Ivaldi, Natalia Lyubova, Alain Droniou, Damien Gerardeaux-Viret, David Filliat, Vincent Padois, Olivier Sigaud, and Pierre-Yves Oudeyer. Learning to recognize objects through curiosity-driven manipulation with the icub humanoid robot. In *IEEE International Conference on Development and Learning - Epirob*, 2013
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Properties for efficient demonstrations to a socially guided intrinsically motivated learner. In *21st IEEE International Symposium on Robot and Human Interactive Communication*, 2012a
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Whom will an intrinsically motivated robot learner choose to imitate from? In Joanna Szufnarowska, editor, *Proceedings of the Post-Graduate Conference on Robotics and Development of Cognition*, pages 32–35, September 2012e. doi: 10.2390/biecoll-robotdoc2012-12
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Interactive learning gives the tempo to an intrinsically motivated robot learner. In *IEEE-RAS International Conference on Humanoid Robots*, 2012c
- Sao Mai Nguyen, Adrien Baranes, and Pierre-Yves Oudeyer. Bootstrapping intrinsically motivated learning with human demonstrations. In *IEEE International Conference on Development and Learning*, Frankfurt, Germany, 2011a

6.5 OTHER INTERNATIONAL PUBLIC PRESENTATIONS

- Sao Mai Nguyen and Pierre-Yves Oudeyer. Strategic robot learner for interactive goal-babbling. In *Workshop in Hierarchical Learning at Robotic Science and Systems*, Berlin, Germany, 2013d
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Strategic robot learner for interactive goal-babbling. In *Workshop in Active Learning in Robotics at Robotic Science and Systems*, Berlin, Germany, 2013c
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Socially guided intrinsically motivated learner. In *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*, pages 1 –2, nov. 2012b. doi: 10.1109/DevLrn.2012.6400809
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Socially guided intrinsic motivation. In *Spring School on Developmental Robotics and Cognitive Bootstrapping*, 2012f
- Sao Mai Nguyen, Adrien Baranes, and Pierre-Yves Oudeyer. Constraining the size growth of the task space with socially guided intrinsic motivation using demonstrations. In *IJCAI Workshop on Agents Learning Interactively from Human Teachers*, 2011b
- Sao Mai Nguyen, Masaki Ogino, and Minoru Asada. Real-time face swapping as a tool for understanding infant self-recognition. In *Proceedings of the 10th International Conference on Epigenetic Robotics*, pages pp.171–172, Glumslöv, Sweden, 2010

