TÉCNICO LISBOA

# Multimodal Human-Robot Interaction Using Gestures and Speech Recognition

**João Pedro Almas Garcia**

Thesis to obtain the Master of Science Degree in

## Electrical and Computer Engineering

Supervisors: Prof. Pedro Manuel Urbano de Almeida Lima
Dr. Tiago Santos Veiga

### Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Pedro Manuel Urbano de Almeida Lima
Members of the Committee: Prof. Alexandre José Malheiro Bernardino

**October 2016**

# Acknowledgments

First, I would like to thank my parents for their encouragement and support, which was vital in the achievement of this milestone.

To my better half, Patrícia, for the caring and the strength you give me. Thank you.

I would also like to acknowledge my dissertation supervisors, Prof. Pedro Lima and Tiago Veiga, for their insight and sharing of knowledge that has made this Thesis possible.

Moreover, thanks to the members of the IRSgroup, in particular André, Luís, Miraldo, Nuno and Tiago, for the leisure, the scientific discussions and their support to this work.

Finally, thanks to all my friends and colleagues that helped me grow as a person.

# Abstract

This work proposes a Decision-Theoretic (DT) approach to problems involving interaction between robot systems and human users, which takes into account the latent aspects of Human-Robot Interaction (HRI), e.g., the user's status.

The presented approach is based on the Partially Observable Markov Decision Process (POMDP) framework, which efficiently handles uncertainty in planning problems involving physical agents, extended with information rewards (POMDP-IR) to optimize the information-gathering capabilities of the system. The approach is formalized into a framework which considers: observable & latent variables; gesture & speech rooted observations; and action factors which are related to the agent's actuators or to the information gain goals (Information-Reward (IR) actions).

Under the proposed framework, the robot system is able to: actively gain information and react according to hidden features, inherent to HRI settings; effectively achieve the goals of the task in which the robot is employed; and follow a socially appealing behavior.

Finally, the framework was thoroughly tested in a socially assistive scenario, in a realistic apartment testbed and resorting to an autonomous mobile social robot. The experiments' results prove the validity of the proposed approach for problems involving robot systems in HRI scenarios.

# Keywords

Planning Under Uncertainty; Markov Decision Processes; Partial Observability; Information Gain; Human-Robot Interaction; Socially Assistive Robotics

# Resumo

Este trabalho propõe uma abordagem, baseada em métodos de Teoria da Decisão (TD), a problemas que envolvam a interação entre sistemas robóticos e utilizadores humanos, tendo em conta os apectos latentes da Interação Humano Robô (HRI), e.g., o estado do utilizador.

A abordagem proposta é fundamentada na estrutura dos Processos de Decisão de Markov Parcialmente Observáveis (POMDPs), que lidam eficientemente com a incerteza presente em problemas de planeamento que envolvam agentes físicos, aumentada de forma a incluir Recompensas de Informação (POMDP-IR), o que garante a otimização das capacidades de obtenção de informação do agente. A abordagem é formalizada numa estrutura para tomada de decisão que considera: variáveis observáveis e latentes; observações relacionadas com gestos e fala do utilizador humano; e ações referentes aos atuadores do agente ou aos objetivos de ganho de informação.

Seguindo a estrutura proposta, o sistema robótico tem a capacidade de: ganhar informação ativamente e reagir de acordo com variáveis latentes, inerentes a cenários de HRI; atingir eficazmente os objetivos da tarefa em que está empregue; e manter um comportamento socialmente apelativo.

Por fim, a estrutura foi sujeita a testes num cenário de terapia, num apartamento *testbed* e recorrendo a um robô social móvel. Os resultados das experiências demonstram a validade da abordagem proposta para problemas envolvendo sistemas robóticos em cenários de HRI.

# Palavras Chave

Planeamento Sob Incerteza; Processos de Decisão de Markov; Observabilidade Parcial; Ganho de Informação; Interação Humano Robô; Robótica de Assistência Social

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Acronyms

| | |
|---|---|
| **ADD** | Algebraic Decision Diagrams |
| **AR** | Assistive Robotics |
| **ASD** | Autism Spectrum Disorder |
| **ASR** | Automatic Speech Recognition |
| **BNF** | Backus-Naur Form |
| **CPD** | Conditional Probability Distribution |
| **CPT** | Conditional Probability Table |
| **DBN** | Dynamic Bayesian Network |
| **DP** | Dynamic Programming |
| **DT** | Decision-Theoretic |
| **HMM** | Hidden Markov Model |
| **HRI** | Human-Robot Interaction |
| **IR** | Information-Reward |
| **MDP** | Markov Decision Process |
| **NRS** | Network Robot System |
| **PBVI** | Point-based Value Iteration |
| **POMDP** | Partially Observable Markov Decision Process |
| **POMDP-IR** | Partially Observable Markov Decision Process with Information Reward |
| **PWLC** | Piecewise Linear Convex |

**RL** Reinforcement Learning

**SAR** Socially Assistive Robotics

**SDK** Software Development Kit

**SIR** Socially Interactive Robotics

**VGB** Visual Gesture Builder

# 1

# Introduction

## Contents

## 1.1 Motivation

Recent technological developments have extended robotics to social settings. From factories and laboratories, robots are shifting to the less structured, more uncertain human-inhabited environments. From this background emerged Human-Robot Interaction (HRI), an interdisciplinary research field which combines robotics, social science, cognitive science and artificial intelligence, to develop robots capable of sharing an environment and even goals with humans.

Beyond the basic capabilities of moving and acting autonomously, robots in HRI scenarios need to communicate and interact with human users in a social and engaging manner. In this context, socially intelligent robotics emerged with the purpose of creating robots capable of exhibiting natural social qualities.

Robotics researchers are earnestly developing more complex and intelligent robots to assist human users in hospitals, elder care centers and homes. As an illustration: pet-like robots are employed in the care of the elderly in nursing homes [1], and robot systems are used in the care of children with Autism Spectrum Disorder (ASD), in projects such as INSIDE[1]. Within the larger context of HRI, these robots, which provide assistance to human users through social interaction, are the focus of research of Socially Assistive Robotics (SAR) [2]. The relevance of this area is growing due, among other reasons, to:

- The rising number of dementia cases worldwide, along with the general aging of the population, which is already straining care giving services [3]: As depicted in Figure 1.1, by 2080, an estimated twenty percent of the world population will be over 65 years old in contrast to the eight percent in 2015. This rises questions such as who will care for the elderly population in the near future;
- The receptiveness of individuals with cognitive and social disabilities (e.g., persons with ASD) to therapy using social robots, due to the difficulties of these individuals with social cues.

Overall, there is a significant positive treatment effect in using robots in therapy [4]. Robots can be available twenty-four hours a day, help to deal with the shortage of trained support workers or ease the workload of human therapists, extend the reach of therapy to more people in need by providing more affordable health care, among other advantages.

Notwithstanding, robots need to become more sophisticated and versatile in order to be more effective and to expand its application range. The task of creating robots that can participate in the complex human social interaction is challenging and current robots present several limitations: most robots are still partially or fully teleoperated, and the others exhibit simple, rudimentary behavior.

In addition, using Network Robot Systems (NRSs) significantly helps enhancing HRI capabilities. This approach allows to handle difficult recognition tasks more efficiently, by using environmental sensor networks and ambient intelligence systems to augment the robot's recognition capabilities [5]. Sensors

[1] http://project-inside.pt/

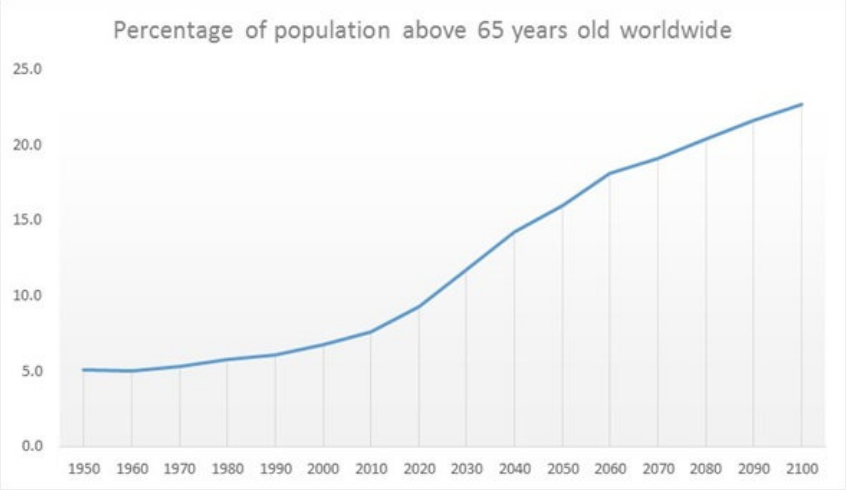



**Figure 1.1:** Estimate of the percentage of population aged 65 years old or older worldwide from 1950 to 2100[2]

on-board the robot might be ineffective in dealing with occlusions, decreasing the system's robustness (e.g., having an obstacle between the robot and the person would interfere with gesture and speech recognition). Furthermore, multiple processing units in the NRS decrease the latency of the system by distributing sensor information processing to different computers in the network.

A successful interaction implies, also, that the robot adapts its behavior to the user's personality, mood and other factors which are not directly observed. These are called latent variables and introduce a new problem: to plan in an uncertain environment while actively gaining information on the hidden variables.

An effective way to deal with this problem is through a Decision-Theoretic (DT) approach, using Partially Observable Markov Decision Processes (POMDPs), namely the extension to POMDPs: Partially Observable Markov Decision Processes with Information Rewards (POMDPs-IR). The POMDP is a model of the agent's decision process, able to deal with uncertainty in the environment and encode the goals of a task. POMDPs-IR extend POMDPs to allow the agent to actively seek to reduce the uncertainty regarding the state of the environment [6].

In this work, a decision-theoretic approach to social HRI is proposed. This approach effectively deals with the presence of latent variables in the environment and solves the problem of planning under uncertainty for a robot system acting, for instance, in a socially assistive setting.

[2]United Nations, Department of Economic and Social Affairs, Population Division (2015). World Population Prospects: The 2015 Revision, custom data acquired via website.

## 1.2 Socially Assistive Robotics

SAR, as previously mentioned, refers to robots that assist human users through social interaction. It is defined as the intersection of Assistive Robotics (AR) and Socially Interactive Robotics (SIR). On one hand, AR comprise all robots that give aid or support to a human user. This definition includes, for example: wheelchair robots [7], companion robots [8] and manipulator arms [9]. On the other hand, SIR characterize robots whose goal is to socially engage humans in HRI. This comprise robot toys [10], educational tools [11] or even therapeutic aids [12].

Both SAR and AR have the goal of assisting a human user but SAR restrains the assistance to social interaction. As an example, the MIT-Manus system [13], which helps stroke victims by physically guiding them through exercises, is an assistive robot but does not fit into the SAR category as it interacts physically but not socially with the user.

Socially intelligent or interactive robots and SAR share the purpose of socially engaging users, although socially assistive robots are specifically designed to help people. For instance, the iCub robot [14] is capable of exhibiting a wide range of facial and body expressions, respond to physical interaction and engage in social learning. Nevertheless, the iCub's purpose is not to aid people and, therefore, it is not considered a socially assistive robot.

The socially assistive robot inherits, from the area of SIR, the objective of having human social characteristics [15], namely:

- Express and perceive emotions;
- Communicate with high-level dialogue;
- Learn/recognize models of other agents;
- Establish/maintain social relationships;
- Use and perceive natural cues;
- Exhibit distinctive personality and character;
- Learn/develop social competencies.

SAR is a multidisciplinary field that encompasses robotics, medicine and social & cognitive sciences. This results in a multitude of applications to this field, ranging from guiding visitors through museums [16] to assist elderly patients in activities of daily living [17]. The fundamental feature of this field is the social factor of the interaction as a means of assisting a human user.

Robots in SAR adopt different platforms, from university unique robot systems to commercial robots, and different methods of assistance, such as coaching, education or motivation. Successful real-world

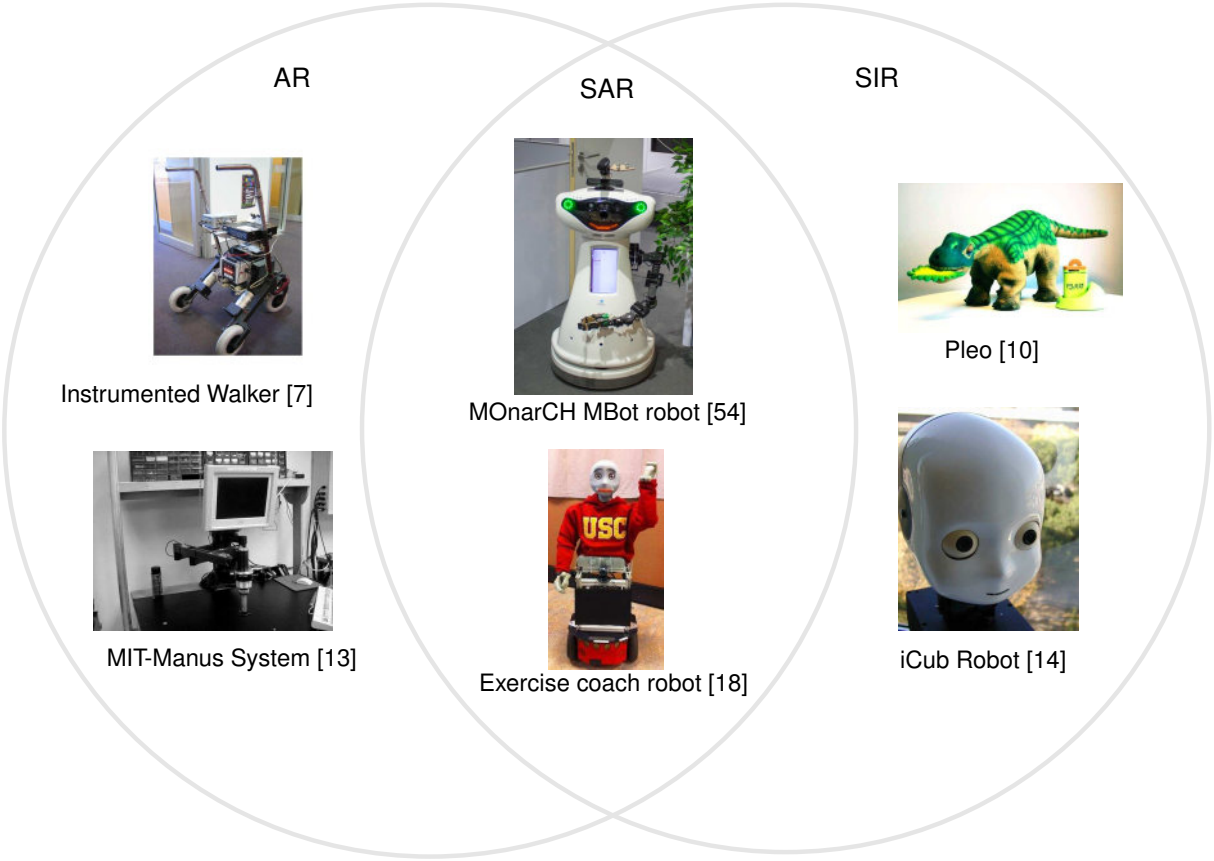



**Figure 1.2:** Illustration of robots employed in AR, SAR and SIR

applications of SAR include coaching elderly in physical exercise [18] and robot weight-loss assistance [19]. These applications have in common promising results, among which is highlighted the receptiveness of the human users in receiving help from and engaging with the socially assistive robot.

Figure 1.2 represents real robots employed in AR, SIR and in the intersection: SAR scenarios.

## 1.3 Network Robot Systems

A NRS is a distributed system which is capable of interacting with the environment [20]. This system includes robots and other technologies such as environmental sensors and actuators, connected via a communication network and cooperating in the same task.

NRS is a multidisciplinary field that combines robotics, ubiquitous computing and network communications. Its key topics include cooperative localization and navigation, cooperative environment perception, cooperative map building, cooperative task execution, among others.

A line of research in NRS started as an extension of the concept of sensor networks. The goal is to adapt the geographic distribution of the sensors or to have autonomous mobile robots, supporting a

static sensor network, in order to better perceive the environment.

Following recent technological developments in that area, research in NRS moved towards ubiquitous robotics, in which networked robots are integrated in environments that include humans and a pervasive network of sensors and actuators.

In a networked robot system operating in an ubiquitous environment, three types of robots are considered [21]:

**Visible robots:** which translate into the conventional robot, with physical embodiment (e.g., humanoid robot);

**Virtual robots:** which are software agents that interact with users through interfaces (e.g., smartphones);

**Unconscious robots:** which represent robots that the user is unaware of and are used to gather information on the environment (e.g., smart embedded sensors).

## 1.4 Objectives

The goal of this Thesis is to study planning under uncertainty in a HRI scenario with latent variables. In this context, this work presents a new approach to the problem of decision-making in such environments, based on the theory of POMDP-IR. This approach allows to handle uncertainty in the state of the environment, introduced by noisy sensors and the latent aspects of HRI.

To validate the proposed approach, the developed model is applied to a real NRS in a socially assistive scenario. The experiments consist of a robot therapist task, where the NRS assists the user in a physical recovery/rehabilitation exercise.

This project studies, also, the application of NRS to SAR. It explains the way to develop a distributed system to surpass hindrances inherent to deploying robot systems in the real-world, e.g., occlusive situations.

To summarize, the contributions of this Thesis are:

1. A framework for planning under uncertainty in HRI problems with latent variables, as in the case of SAR tasks;

2. A set of experiments that confirm the validity of the proposed framework in a real-world socially assistive scenario;

3. The development of a NRS to be deployed in a socially assistive setting.

## 1.5 Related Work

DT planning under uncertainty is a topic with growing importance in robotics, with applications in mobile robot localization and navigation [22], decision-making of teams of robot soccer [23], Cooperative Active Perception [6], among others. In the area of HRI, the POMDP framework has been used in some notable works, among which are highlighted:

1. The automated hand-washing assistant for persons with dementia [24];
2. The intelligent robot wheelchair [25];
3. The robot nursing aid [17].

The automated hand-washing assistant, proposed by Hoey et al., uses a real-time video-based system to assist (verbally or through visual prompts) a person with dementia in washing his/hers hands. The system is based on the POMDP framework, which considers:

- *Task* variables to keep track of the hand-washing sequence;
- *Attitude* variables to represent the person's health status;
- *Book-keeping* variables to track the progress of the task;
- Vision-based observations;
- And verbal or visual prompts as the set of actions.

The hand-washing system adapts to the *attitude* variables, with the goal of driving the user to complete the hand-washing task. The system was evaluated in a ten-week trial with seven persons with different dementia levels. The scenario studied in this project provides an opportunity to the POMDP-IR approach, which would allow to actively gain information on the latent status of the user.

The intelligent robot wheelchair, proposed by Taha et al., aims at predicting and driving the user to the intended destination, from the input provided by a standard wheelchair joystick. The intention recognition problem is transferred to a planning under uncertainty scenario, within the POMDP framework, with the wheelchair as the decision-making agent. The state space represents the wheelchair location and the destination, the observations correspond to the joystick input and the actions indicate the global direction of travel. This project does not consider an agent with social capabilities, namely verbal interaction, which would allow the agent to reduce uncertainty on the user's intention recognition task, and adapt its behavior to the user's preferences (e.g., adapt the wheelchair speed). An approach based on the POMDP-IR provides the means to complete the driving task while actively inferring the user's intention and preferences.

Finally, the robot nursing aid, proposed by Pineau et al., implements an automated reminder system, a people tracking and detection system and guidance capabilities. The agent is capable of reminding and driving a user to a planned activity and answer information requests. The controller, based on the POMDP framework, copes with the activity reminder system and user information request goals, but do not actively infer the person's status. The robot system was implemented in a retirement community, in twelve test scenarios, involving six elders. Once more, the POMDP-IR framework would allow to accomplish the tasks of the robotic nursing aid, actively gain information on the user's status, and adapt the behavior of the agent accordingly.

## 1.6 Outline

This Section describes the organization of this Dissertation, along with a brief summary of each Chapter:

- Chapter 2 details the background on Decision-Theoretic models relevant to the comprehension of this work. Namely, it addresses the Markov Decision Process (MDP), the extension of the MDP to partially observable scenarios (POMDP), the factorization of the DT models and the addition of Information Rewards to the POMDP framework (POMDP-IR);
- Chapter 3 introduces an approach for planning under uncertainty in social HRI scenarios, that is able to deal with latent/hidden variables; It reviews the problem under study and presents guidelines for the modeling of HRI problems;
- Chapter 4 illustrates the properties of the framework proposed in Chapter 3, in a case study inserted in a socially assistive setting;
- Chapter 5 concludes this Dissertation and presents potential directions for future research.

# 2

# Background

## Contents

As stated in Chapter 1, the focus of this work is on planning for autonomous robots in stochastic environments. The environment is considered stochastic or uncertain because the effects of the agent's actions are non-deterministic and the subsequent state of the world is determined probabilistically. Markov Decision Processes (MDPs) [26] provide the mathematical framework to formulate and solve decision-making problems in stochastic environments. In its original form, the MDP models the decision-making process of a single agent, assuming full knowledge of the environment and a discrete-time evolution of the system. To overcome these restraints, significant extensions to the MDP framework have been developed, namely addressing the ability to model: partial observability, multi-agent decision-making and time-driven dynamics.

This chapter reviews the MDP framework and the associated extension which is most relevant to this work: the Partially Observable Markov Decision Process (POMDP).

## 2.1 Markov Decision Processes

The MDP framework models decision-making problems where the outcome of an agent's actions is probabilistic and the state of the environment is known, without uncertainty, at each time step. Formally, a MDP is defined as a tuple $(S, A, T, R)$, in which:

- $S = \{s^1, \ldots, s^N\}$ is a finite set of mutually exclusive states, defining the model of the world;
- $A = \{a^1, \ldots, a^M\}$ is a finite set of actions at the disposal of the agent;
- $T$ is the transition function $T : S \times A \times S \to [0, 1]$. The Transition model $T(s, a, s') = P(s'|s, a)$ represents the probability of reaching state $s'$ from $s$ if action $a$ is performed;
- $R$ is the reward function $R : S \times A \to \mathbb{R}$. The reward model $R(s, a)$ defines the numeric reward for the agent to perform action $a$ while in state $s$. This model reflects the agent's goals or preferences;

At each discrete time step $t$, the agent performs an action $a \in A$, which causes the state of the system to evolve stochastically from $s \in S$ to $s' \in S$, in accordance with the transition model $T(s, a, s')$. This system complies with the Markov property, i.e., the probability distribution over the future states depends only on the present state and action:

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \ldots, s_0, a_0) = P(s_{t+1}|s_t, a_t). \tag{2.1}$$

Moreover, and in order to automate decision-making, the agent needs to have a measure of how its actions fit its purpose. This measure is given by the reward model and it consists of a numerical value $R(s, a)$ granted to the agent by performing action $a$, while the environment is in state $s$. Positive rewards are assigned to the goals of the agent (e.g., reaching a goal state or performing the desired action

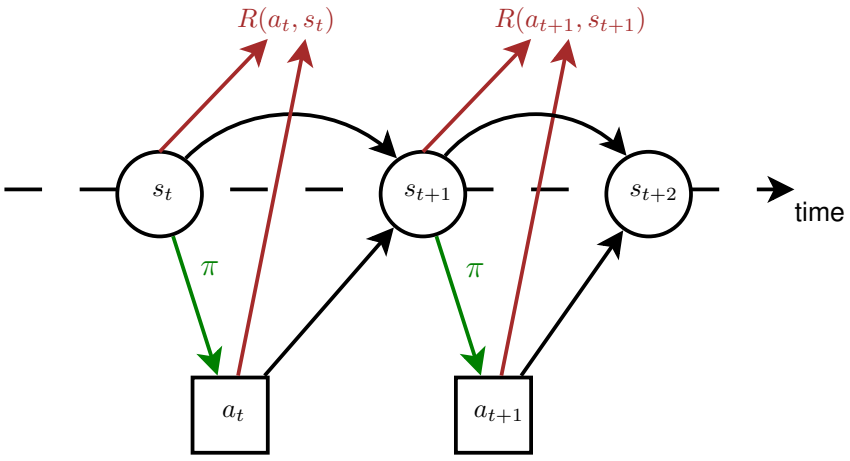



**Figure 2.1:** Dynamic Bayesian Network representation of the MDP model

in the current state) while negative rewards (penalties) are assigned to states to avoid or to forbidden actions in certain states. The reward model allows, likewise, to define priorities on the agent's objectives, considering an agent acting in a scenario with multiple goals.

### 2.1.1 Policies and Value Functions

Within the MDP framework, a solution to the decision-making problem is a set of decision rules, mapping states to actions. This set is denominated *policy* $\pi : S \rightarrow A$ and defines the action $a \in A$ the agent is to perform when the state of the environment is $s \in S$. Policies are stationary, i.e., time-independent, and only depend on the current state. Moreover, a policy fully defines the behavior of the agent. Figure 2.1 represents the dynamics of the MDP as a DBN, which demonstrates the probabilistic dependencies between variables at each time step.

The goal of the decision-making agent is to find the optimal policy $\pi^*$ (solution), which maximizes a performance measure.

Each policy is associated with a measure of its quality, which is designated *Value Function*. The value function $V^\pi : S \rightarrow \mathbb{R}$ is commonly defined as the expected discounted cumulative reward the agent receives by following policy $\pi$, when the initial state of the environment is $s$:

$$V^\pi(s) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t R\big(s_t, \pi(s_t)\big) \Big| \pi, s_0 = s\right], \tag{2.2}$$

where $\mathbf{E}[\cdot]$ is the expectation operator. The discount factor $\gamma \in [0, 1]$ assigns a greater influence to a reward obtained in a near-future, reducing the value of the rewards obtained in a long horizon, and ensures the performance measure converges.

The value function in Equation 2.2 can be decomposed into immediate reward plus discounted value

of successor states, leading to the Bellman recursion [26]:

$$V^{\pi}(s) = R\big(s, \pi(s)\big) + \gamma \sum_{s' \in S} P\big(s'|s, \pi(s)\big) V^{\pi}(s'). \tag{2.3}$$

The most pervasive solution methods for the MDP fit into the Dynamic Programming (DP) class, and are based on the *Bellman equation*:

$$V^{*}(s) = \max_{a \in A} \left[ R\big(s, a\big) + \gamma \sum_{s' \in S} P\big(s'|s, a\big) V^{*}(s') \right]. \tag{2.4}$$

Equation 2.4 defines the optimal value function. Optimality, in this context, implies there is no policy $\pi$ with value function $V^{\pi}$ greater than $V^{*}$. The optimal policy $\pi^{*}$ associated with the optimal value function $V^{*}$ can be computed by choosing, in each state, the action with the highest expected value:

$$\pi^{*}(s) = \arg\max_{a \in A} \left[ R\big(s, a\big) + \gamma \sum_{s' \in S} P\big(s'|s, a\big) V^{*}(s') \right]. \tag{2.5}$$

### 2.1.2 Value Iteration

Value Iteration is an iterative procedure, which computes the optimal value function for each state, by iterating Equation 2.4 in an update step:

$$V_{n+1}(s) = \max_{a \in A} \left[ R\big(s, a\big) + \gamma \sum_{s' \in S} P\big(s'|s, a\big) V_{n}(s') \right]. \tag{2.6}$$

This operation is designated the *Bellman backup*, denoted $H_{MDP}$:

$$V_{n+1} = H_{MDP} V_{n}, \tag{2.7}$$

and converges to the optimal value function $V^{*}$ when the iteration step $n \to \infty$ [27]. In practice, the value iteration algorithm iterates over the update step until convergence, i.e., until the difference between the previous and the updated value function is below a threshold $\epsilon$ :

$$\max_{s \in S} |V_{n+1}(s) - V_{n}(s)| < \epsilon. \tag{2.8}$$

The Value Iteration algorithm for the MDP framework is described in Algorithm 2.1.

**Algorithm 2.1:** MDP Value Iteration

**Data**:
$\epsilon$ Threshold
**Result**:
$\pi$ Approximate Optimal Policy
$V$ Approximate Optimal Value Function

**begin**
Initialize $V(s)$ arbitrarily
**repeat**
$V' \leftarrow V$
**foreach** $s \in S$ **do**
$V(s) \leftarrow \max_{a \in A} \left[ R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V'(s') \right]$
**end**
**until** $\max_{s \in S} |V'(s) - V(s)| < \epsilon$
**foreach** $s \in S$ **do**
$\pi(s) = \arg\max_{a \in A} \left[ R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V(s') \right]$
**end**
**return** $\pi$, $V$
**end**

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \cdots \xrightarrow{I} \pi^* \xrightarrow{E} V^*$$

**Figure 2.2:** Policy Iteration algorithm until convergence to the optimal policy. $\xrightarrow{E}$ symbolizes the *Policy Evaluation* step and $\xrightarrow{I}$ represents the *Policy Improvement* step

### 2.1.3 Policy Iteration

Policy Iteration is an alternative solution method which operates in the policy space, iteratively improving the starting policy. The policy iteration algorithm consists of two steps:

**Policy Evaluation:** Determine $V^\pi$ as defined in Equation 2.2. The set of $|S|$ linear equations can be solved by a linear equation solution method (e.g., Gaussian Elimination) or iteratively.

**Policy Improvement:** Choose the action, in each state, which maximizes the expected value, as defined in Equation 2.5.

The policy iteration stops if the *Policy Improvement* step does not change the policy, i.e., $\pi_{n+1} = \pi_n$. This algorithm guarantees convergence to the optimal policy $\pi^*$.

Figure 2.2 demonstrates the policy iteration algorithm process until convergence to the optimal policy. In the Figure, $\xrightarrow{E}$ symbolizes the *Policy Evaluation* step and $\xrightarrow{I}$ represents the *Policy Improvement* step.

The Policy Iteration algorithm for the MDP framework is detailed in Algorithm 2.2.

**Algorithm 2.2:** MDP Policy Iteration

**Result**: $\pi$ Optimal Policy

**begin**

  Initialize $\pi(s)$ and arbitrarily

  **repeat**

    $n \leftarrow n + 1$

    Solve $V(s) = R(s, \pi_{n-1}(s)) + \gamma \sum_{s' \in S} P(s'|s, \pi_{n-1}(s))V(s')$

    **foreach** $s \in S$ **do**

      $\pi_n(s) \leftarrow \arg\max_{a \in A} \left[ R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a)V(s') \right]$

    **end**

  **until** $\pi_{n+1} = \pi_n$

  **return** $\pi$

**end**

## 2.2 Partially Observable Markov Decision Processes

The assumption that the state of the environment is known to the agent is not realistic in many practical scenarios. A physical agent which is required to sense its environment might perceive noisy observations, possibly providing misleading information. Moreover, the agent may obtain incomplete or partial knowledge on the state of the environment, through its sensing capabilities. This leads to perceptual aliasing, i.e., the agent's observations are the same in different states. Partial observability refers to these scenarios, in contrast with full observability, where the state of the system is accessible to the agent at each time step.

The Partially Observable Markov Decision Process models the interaction between a decision-making agent and a stochastic environment, which is only partially observable to the agent. It provides a framework for planning in problems which combine partial observability, uncertain action effects, unknown environment dynamics and multiple objectives.

Formally, a POMDP is a tuple $(S, A, T, R, \Omega, O)$, in which:

- $S$ is the state space, $A$ is the action space, $T$ is the transition function and $R$ is the reward function, as defined in the MDP framework (Section 2.1).
- $\Omega = \{o^1, \ldots, o^W\}$ is a finite set of observations that correspond to features of the environment which can be directly perceived by the agent's sensors.
- $O$ is the observation function $O : S \times A \times \Omega \rightarrow [0, 1]$. The Observation model $O(o, a, s') = P(o|a, s')$ corresponds to the probability of observing $o$ after performing action $a$ and reaching state $s'$.

Figure 2.3 illustrates the dynamics of the POMDP model as a Bayesian Decision Network. In contrast with Figure 2.1, the state of the environment is now a hidden variable and the agent receives an

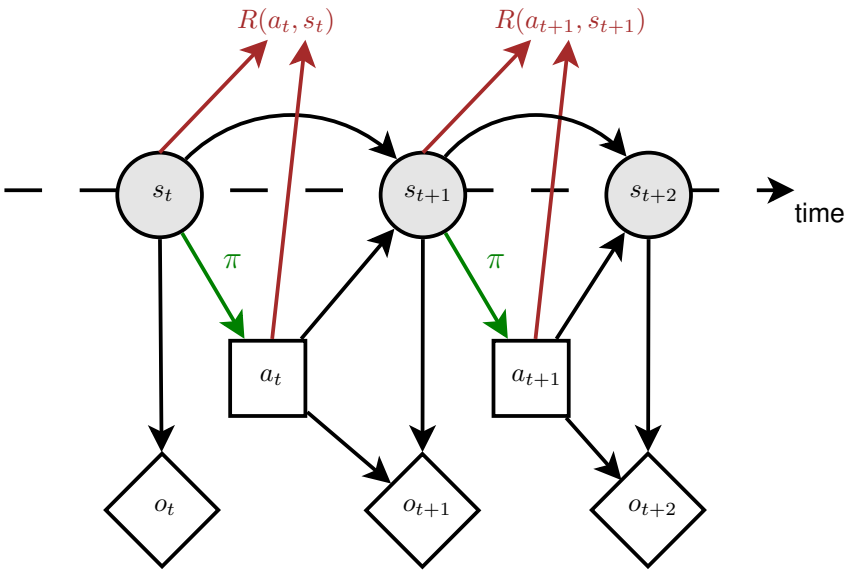



**Figure 2.3:** Dynamic Bayesian Network representation of the POMDP model

observation, which is probabilistically dependent on the state.

### 2.2.1 Belief State

In a partially observable environment, the system does not satisfy the Markov property, since observations do not uniquely identify the state of the environment and a direct mapping of observations to actions does not suffice to achieve optimal behavior. Therefore, an history of executed actions and perceived observations would be necessary to infer the current state. Storing all actions and observations would require increasing memory over time, rendering it impractical for long planning horizons.

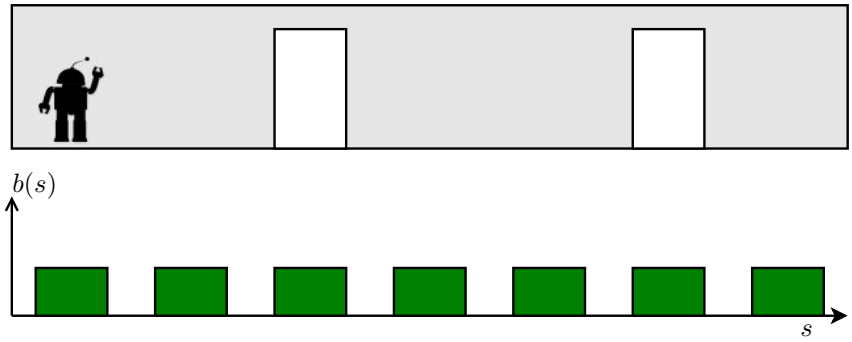
A solution is to encode the aforementioned history in a probability distribution over all states: the belief state. This belief $b(s)$ is a vector that denotes the probability that the state of the environment is $s \in S$. $b$ is dynamically updated by the Bayes' rule, every time the agent performs an action $a \in A$ and observes $o \in \Omega$:

$$b^{ao}(s') = \frac{P(o|s', a)}{P(o|b, a)} \sum_{s \in S} P(s'|s, a) b(s). \tag{2.9}$$

In Equation 2.9, $P(s'|s, a)$ and $P(o|s', a)$ are defined by the Transition and Observation model, respectively. $P(o|b, a)$ is a normalizing constant, defined by:

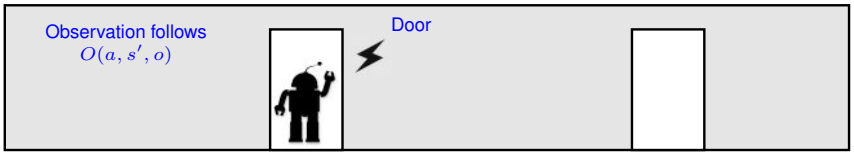
$$P(o|b, a) = \sum_{s' \in S} P(o|s', a) \sum_{s \in S} P(s'|s, a) b(s). \tag{2.10}$$

The belief respects the Markov property, i.e., at each transition, the next belief state only depends

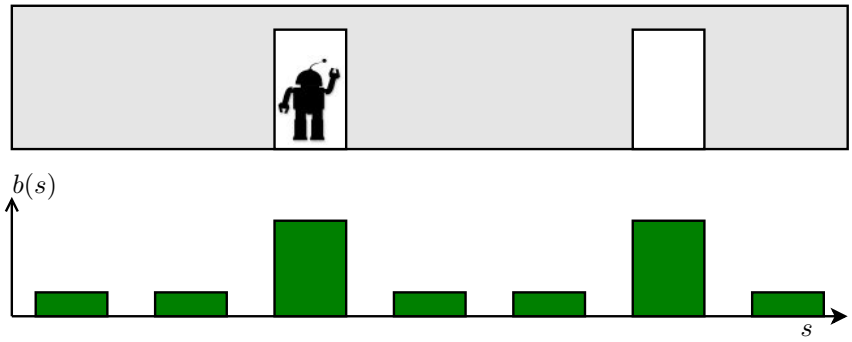



**(a)** Initial belief at time step $t = 0$


**(b)** The selected action is to move forward




**(c)** At time step $t = 1$ the robot observes $door$




**(d)** The belief is updated according to Equation 2.9

**Figure 2.4:** Belief update example for the mobile robot localization problem

on the current belief, action and observation. Also, it is a sufficient statistic of the history, meaning the agent's performance is not affected by not memorizing the complete sequence of actions and observations.

Figure 2.4 shows an example of the belief update process for a robot agent. The robot navigates through a corridor, which is discretized into areas. At each time step, the robot either observes $door$

or $corridor$. Figure 2.4(a) shows the initial belief state $b_0$, which is uniform, representing absence of knowledge on the location of the robot. The robot moves forward (Figure 2.4(b)) and at time step $t = 1$ it observes $door$ (Figure 2.4(c)). Finally, the belief is updated in accordance with the observation received. The updated belief state in Figure 2.4(d) shows that the most likely location of the robot is in one of the doors.

### 2.2.2 Policies and Value Functions

The goal of the agent is to choose actions to successfully complete its task following the best behavior possible, i.e., to compute the optimal policy. A policy $\pi(b)$ maps belief states to actions, indicating the action to perform for each belief. It is, therefore, a function over a continuous set of probability distributions over the state space $S$. The evaluation of a policy is done through the value function $V^{\pi}(b)$, defined as the expected future discounted reward given to the agent by following the policy $\pi$, starting from belief $b$:

$$V^{\pi}(b) = \mathbf{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(b_t, \pi(b_t)) \middle| b_0 = b\right], \tag{2.11}$$

where:

$$R(b_t, \pi(b_t)) = \sum_{s \in S} R\big(s, \pi(b_t)\big) b_t(s). \tag{2.12}$$

Equation 2.12 defines the reward given to the agent in time step $t$, according to the belief state, while following policy $\pi$.

The policy that maximizes the value function is the optimal policy $\pi^*$. It indicates the optimal action to perform in the current time step for each belief state $b$, assuming the agent will also act optimally onward. The value of the optimal function is called the optimal value function $V^*$.

The optimal value function satisfies the Bellman optimality equation $V^* = H_{POMDP} V^*$:

$$V^*(b) = \max_{a \in A}\left[R(b,a) + \gamma \sum_{o \in O} P(o|b,a) V^*(b^{ao})\right], \tag{2.13}$$

with $b^{ao}$ given by 2.9, $P(o|b,a)$ as defined in 2.10 and $R(b,a)$ given by 2.12.

The solution is optimal if 2.13 holds for every belief $b$. Value functions may have infinite values as they are defined over a continuous belief space, what would make their computation intractable. A value function has, however, a particular structure that can be explored: the state is known at the corners of the belief space and an agent can, generally, take better decisions when the uncertainty is lower. Specifically, for finite-horizons, the value function is Piecewise Linear Convex (PWLC) [28], and can be approximated closely by PWLC functions for infinite-horizons [29]. This means the value function, for each iteration $n$, can be parametrized by a finite set of vectors $\{\alpha_n^i\}$, with each vector associated to an action. To compute the value of a belief, it is necessary to find the vector that has the largest dot product

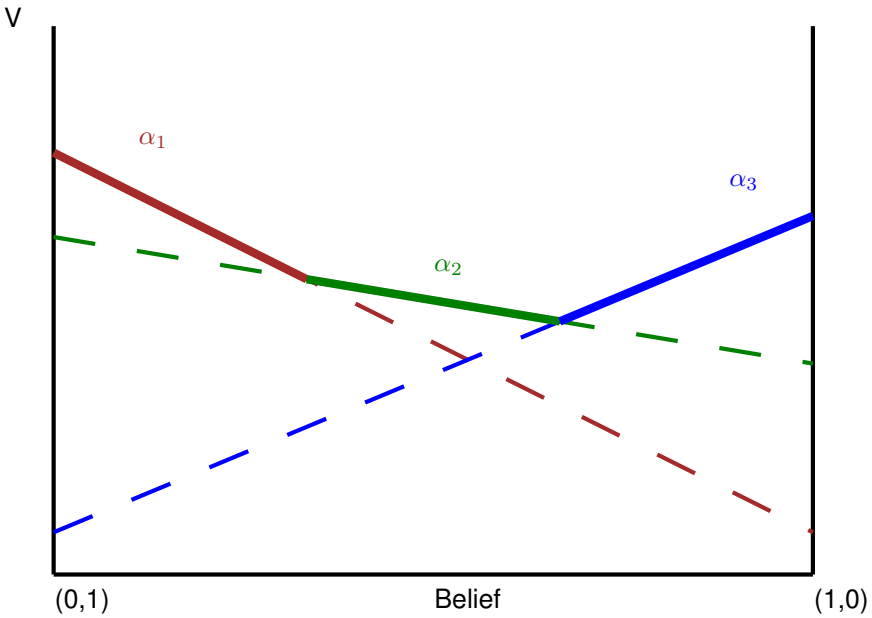



**Figure 2.5:** Example of a Value Function for a two-states POMDP

with the belief state:

$$V_n(b) = \max_{\{\alpha_n^i\}} b \cdot \alpha_n^i \tag{2.14}$$

The representation of the value functions by a finite set of vectors, divides the belief space into regions. Each region is related to a particular maximizing vector and, therefore, associated to a specific action. Figure 2.5 demonstrates a value function of a two-state POMDP in a given horizon. For a two state POMDP, the belief state can be represented in one-dimension, i.e., with a single number: if a state has probability $p$ the other must have probability $1 - p$. The value function corresponds to the PWLC function defined by the maximizing vectors, and is denoted as the continuous line in Figure 2.5.

### 2.2.3 Value Iteration

Value iteration is an iterative method for solving POMDPs that explores the structure of value functions. It builds a sequence of value functions estimates by looking one step further into the future in each iteration, considering all possible actions and observations. The iterating process continues until convergence to the optimal value function.

The main idea behind value-iteration algorithms is that the $\alpha$-vectors that represent the value function, in the next step $n + 1$ and for a given belief $b$, can be computed through:

$$backup(b) = \arg\max_{\{\alpha_{n+1}^k\}_k} b \cdot \alpha_{n+1}^k, \tag{2.15}$$

which is denoted the *Bellman backup operator*.

The policy is, therefore, the action associated with the Bellman backup operator for a given belief:

$$\pi(b) = a(backup(b)). \tag{2.16}$$

One approach to exact value iteration is the Monahan's enumeration algorithm [30]. It starts to enumerate all possible vectors in the next step and prune dominated vectors. The construction of a vector requires selecting an action and a vector in the current value function $V$, for each observation. A more computationally efficient algorithm for exact value iteration is Incremental Pruning [31]. Instead of trying all possible combinations of the $\alpha$-vectors, as in the Monahan's algorithm, the Incremental Pruning algorithm combines sets of vectors incrementally observation by observation, pruning dominated vectors in each combination.

Computing exact solutions, however, is intractable for large problems, which led to the development of methods for approximate solutions. These methods exchange optimality for scalability. The most common are Grid-based methods [32], policy search [33], heuristic search [34] and Point-based methods.

The concept behind point-based methods is that planning can be carried out on a finite subset of beliefs $B$ instead of the complete belief space. The quality of the resulting policy depends on whether $B$ includes the parts of the belief simplex that are reachable, i.e., the beliefs that can be reached during the execution of the problem.

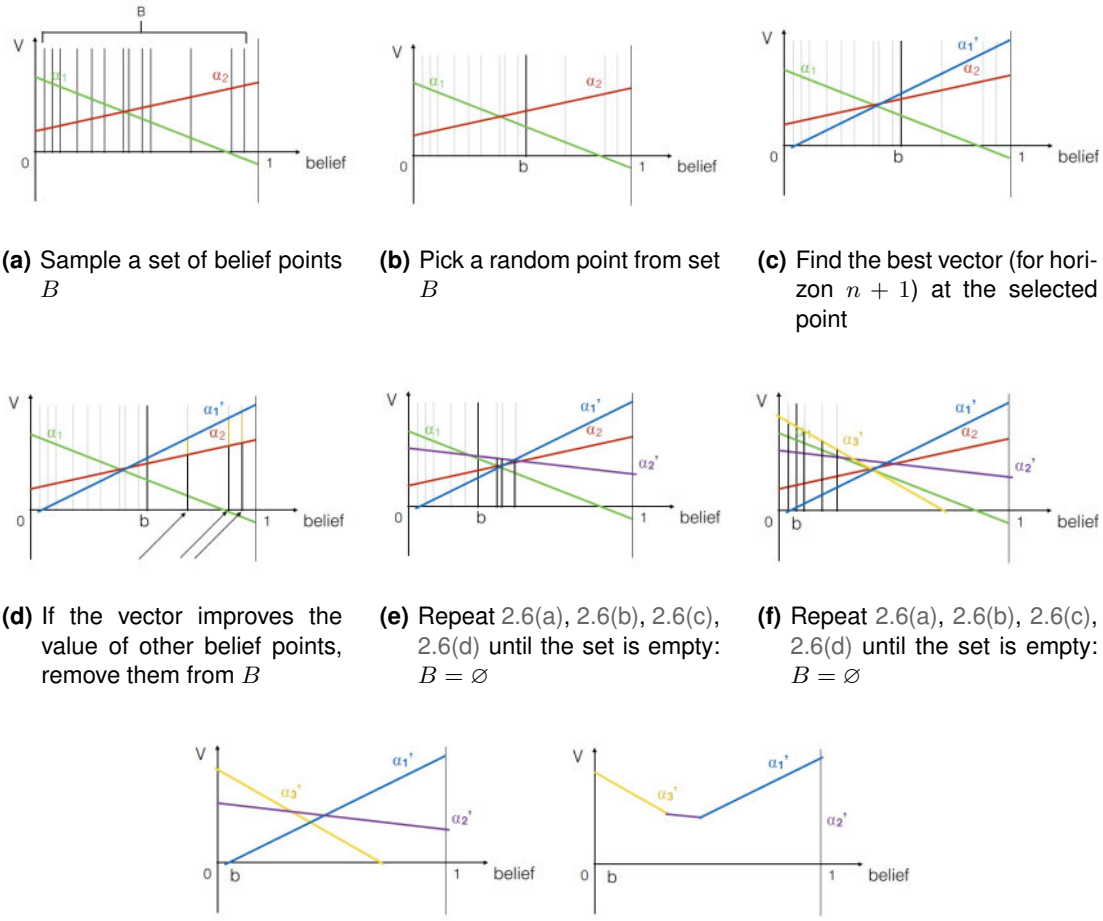
Point-based methods mainly differ in two parts of the algorithm: the collect and update stage. In the collect phase, the belief space is sampled to construct or augment the subset $B$; In the update phase, the value functions are calculated using the backup operator.

The original Point-based Value Iteration (PBVI) algorithm [35] initializes the subset $B$ with the initial belief state $b_0$, and expands it, in each iteration, by greedily choosing new reachable beliefs that improve the density of the belief set. The collect stage consists of:

1. For each $b \in B$:
   (a) Produce new beliefs $b_{a_i}$ by simulating each action;
   (b) Measure the L1-norm of the new beliefs to the points already in the set and keep only the new belief $b_{a_i}$ which is farthest away from the others in $B$.

The update step follows a full-backup strategy, meaning the algorithm executes the backup operation for every belief in $B$.

An alternative is provided by *PERSEUS* [37]: a randomized PBVI algorithm. In *PERSEUS*, the collection phase consists of selecting beliefs by randomly exploring the environment. Initially, this beliefs form the non-improved set $\tilde{B}$. Furthermore, the update stage randomly chooses the order of beliefs to




**(a)** Sample a set of belief points $B$

**(b)** Pick a random point from set $B$

**(c)** Find the best vector (for horizon $n+1$) at the selected point

**(d)** If the vector improves the value of other belief points, remove them from $B$

**(e)** Repeat 2.6(a), 2.6(b), 2.6(c), 2.6(d) until the set is empty: $B = \varnothing$

**(f)** Repeat 2.6(a), 2.6(b), 2.6(c), 2.6(d) until the set is empty: $B = \varnothing$

**(g)** Prune dominated vectors

**(h)** The approximated value function for horizon $n+1$

**Figure 2.6:** Example of the computation of the next horizon value function through the *PERSEUS* algorithm [36]

apply the backup operation. At each backup, the beliefs whose value is improved are removed from the set $\tilde{B}$.

The Perseus algorithm is described in Figure 2.6, with an example of a two-state POMDP. First, the agent randomly explores the environment and collects a set of reachable points $\tilde{B}$. Next, the algorithm picks a random belief point $b \in \tilde{B}$ and computes the maximizing vector for the iteration $n+1$: $\alpha = backup(b)$. If $\alpha$ improves the value of belief $b$, i.e., $b \cdot \alpha \geq V_n(b)$, then it is added to $V_{n+1}$. Otherwise, the maximizing vector of the time step $n$: $\alpha' = \arg\max\{\alpha_n^i\}$, is added to $V_{n+1}$. This step ensures the value function $V_{n+1} = \tilde{H}_{PERSEUS} V_n$, computed by *PERSEUS*, upper bounds $V_n$ over $B$, i.e.:

$$V_n(b) \leq V_{n+1}(b), \tag{2.17}$$

for all $b \in B$. Afterward, the belief points whose value is improved by $\alpha$ or $\alpha'$ are removed: $\tilde{B} = \{b \in B :$

$V_{n+1}(b) < V_n(b)\}$. This way, the algorithm only keeps track of the non-improved points. The previous steps are repeated until the set $\tilde{B}$ is empty.

The *PERSEUS* algorithm iterates until some convergence criterion is met. A typical criteria is to bound the difference between successive value functions: $\max_{b \in B}(V_{n+1}(b) - V_n(b)) < \beta$, where $\beta$ is the threshold below which *PERSEUS* stops performing backup stages.

*PERSEUS* is the algorithm of reference in solving POMDPs throughout this work.

## 2.3 Factored Models

The representation of the state, observation and actions spaces in Section 2.2, is denoted flat. It consists of enumerating all possible states, observations and actions, respectively. Alternatively, these spaces can be represented as a combination of variables/factors.

The environment, for a given problem, can be represented through certain features of interest (e.g., location of the human user and battery level of the robot). If each feature is associated with a variable $X_i$, with domain $D_i$, the state space becomes the cross product of the variables related to the features of the environment:

$$S = \{D_1 \times D_2 \times \cdots \times D_k\},$$

considering an environment with $k$ features.

Similarly, each actuator or actuation feature (e.g., the direction and speed of the agent) define the action space as a combination of variables $A_j$. Finally, each sensor or type of information transform the observation space in a likely manner.

Models with this representation are denoted factored models [38]. These models exploit the structure of the decision-making problem to solve it more efficiently: factored models reduce the number of variables involved in the Conditional Probability Distribution (CPD) of each state/action/observation factor, by considering the factors' conditional dependencies; simplify the description of each CPD through the use of decision diagrams or trees; enable the compact representation of the transition and observation models as local Conditional Probability Tables (CPTs) [39], or as Algebraic Decision Diagramss (ADDs) [40].

Factored models are typically represented as a Dynamic Bayesian Network (DBN), which clarifies the conditional dependencies between the model variables. Figure 2.7 depicts an example of a factored model represented by a DBN.

The factored representation reduces the transition model to the product of the CPDs of the state variables. In the factored model represented in Figure 2.7, the transition function is:

$$P(X_1', X_2'|X_1, X_2, A_1, A_2) = P(X_1'|X_1, A_1) \cdot P(X_2'|X_1, X_2, A_2). \tag{2.18}$$

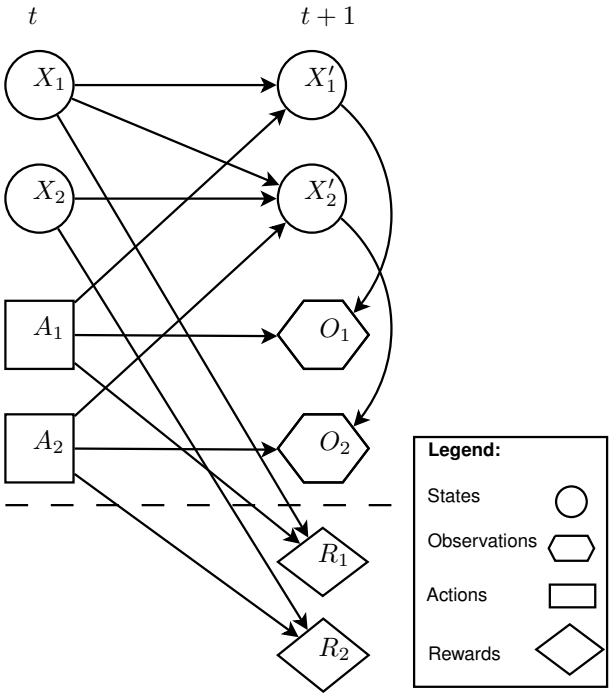



**Figure 2.7:** Example of a factored model represented by a DBN

Similarly, for the observation model:

$$P(O_1, O_2|X_1, X_2, A_1, A_2) = P(O_1|X'_1, A_1) \cdot P(O_2|X'_2, A_2). \tag{2.19}$$

The factored representation allows, through the additive separability property, to divide the reward model into multiple reward functions, each depending on a subset of the model's variables. In Figure 2.7, the reward model is:

$$R(X_1, X_2, A_1, A_2) = R_1(X_1, A_1) + R_2(X_2, A_2). \tag{2.20}$$

## 2.4 POMDP with Information Rewards

The traditional POMDP model defines a state-based reward function, which does not reward information gain. Consequently, if information gain is one of the objectives of a task, the POMDP framework needs to be extended to allow rewarding low-uncertainty beliefs. This extension is provided through the POMDP-IR framework [6].

While cooperating with human users in a given task, the objectives of an agent may not only consist of completing the task, but also to be aware of the human status (e.g., safety or mood). The latter can be formalized as keeping a low-uncertainty belief concerning the state of the human user.

The information gain goal is achieved via the inclusion of Information-Reward (IR) actions, which allow rewarding the agent for achieving a certain knowledge on particular features of the environment, namely specific state factors. Therefore, the standard POMDP action space, denoted as $A_d$, is extended with an IR action for each state factor of interest. That is, for a state factor of interest $X_i$ with domain $\{x_1, x_2, \cdots, x_j\}$, the corresponding IR action is:

$$A_i = \{commit_1, commit_2, \cdots, commit_j, null\},$$

and the action space of the POMDP-IR becomes:

$$A^{IR} = A_d \times A_1 \times A_2 \times \cdots \times A_l,$$

where $l$ is the number of IR actions.

At each time step, the agent performs a domain-level action $a \in A_d$, and chooses an extra information action for each state factor of interest. The latter do not influence the transition nor the observation model, but change the reward given to the agent:

$$R^{IR}(X, A) = R_d(X, A_d) + \sum_{i=1}^{l} R_i(X_i, A_i),$$

where $R_d$ is the reward function of the POMDP model and $R_i$ is the information reward.

Every time step, the agent either makes no assumption regarding the information objectives, by choosing the IR action $A_i = null$, or collects the reward for its belief over $X_i$, through $A_i = commit_k$, $1 < k < j$. The rewards given to the agent for a correct or an incorrect assumption are $r_i^{correct}$ and $-r_i^{incorrect}$, respectively. The presence of information rewards influence the policy towards lowering the uncertainty associated with the state factor of interest.

The threshold of belief regarding a particular state factor $\Big(b(X_i = x_k)\Big)$ above which the IR action is $commit_k$, is denoted $\beta$:

$$\beta = \frac{r_i^{incorrect}}{r_i^{correct} + r_i^{incorrect}}. \tag{2.21}$$

The exact values of $r_i^{correct}$ and $r_i^{incorrect}$ depend on the problem and need to take into account the rewards given for other tasks, such as $R_d$. Rewarding too much the IR action in regard to the other actions, might induce the agent to ignore the other tasks. Also, the value of $\beta$ should be such that $b(X_i = x_k) > \beta$ is reachable. Therefore, the value of $\beta$ is dependent on the sensory limitations of the agent, particularly on the ability of the agent to observe the state factor of interest $X_i$.

# 3

# Planning in social robots

## Contents

This work aims at providing a framework for planning in a Human-Robot Interaction (HRI) scenario with latent variables. This chapter, therefore, starts to explain the problem under analysis, then considers the methods to solve it and, finally, proposes a new approach based on Partially Observable Markov Decision Processes with Information Rewards (POMDPs-IR).

## 3.1 Problem Definition

As previously noticed in Chapter 1, social sciences play a determinant role in the development of robots with appealing social qualities. The concept that observable phenomena have underlying and unobserved causes is rooted in social sciences [41]. As an example, one can observe the answers of a patient to a questionnaire and from thence deduce the patient's personality. Despite this, the personality *per se* is not an observable variable, but inferred from the answers to the questionnaire (observed variables). As depicted in Figure 3.1, the answers form measurable variables dependent on latent variables such as personality and mood. The definition of latent variable to be used throughout this work is based on the sample realization definition [41], wherein latent variable is any variable for which there are no realizations in a given sample, i.e., it is not possible to sample values of the latent variable.

Building from theoretical research on human social relationships, Leite et al. [42] enumerated guidelines for social robots, which include the ability to empathize with users, i.e., the capacity to understand, adapt and respond adequately to the user's affective and motivational status. Empathy, as defined by Hoffman [43], is "an affective response that is more appropriate for another's situation than one's own". This concept is essential in the creation and development of social relationships [44]. Following the previous definition of latent variable, the goal of empathizing with the user clearly involves gaining information and reacting according to latent variables: the user's affective and motivational states.

The first step in modeling a Decision-Theoretic (DT) problem is to choose the appropriate framework. This choice is dependent on: the observability of the environment (full or partial); the number of agents considered (single or multiple); and the influence of continuous time in the decision-making process.

The agent acting in a HRI scenario must take into account the effects of its actions in the human user, which are uncertain, and the sensory information it receives, which is noisy. Also, in the scenarios considered in this work, the duration of the actions do not require to consider the time since the last decision was carried out. Planning under these conditions is attainable, for a single agent, through POMDPs. POMDPs, through the transition and observation models, deal with the aforementioned uncertainty, by statistically representing the possible outcomes of the agent's different actions and the accuracy of the sensory information. Furthermore, the problem of empathizing with the human user adds the goal of information gain on latent variables, which is addressed by the extensions to POMDPs introduced by POMDPs-IR.

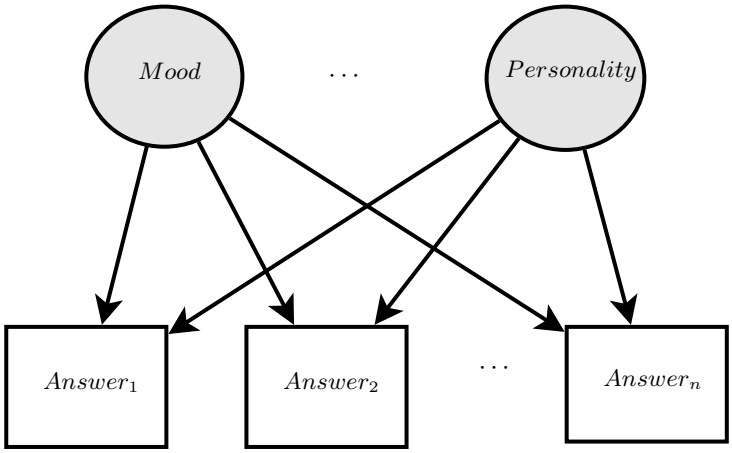



**Figure 3.1:** Example of a model of the observed and latent variables involved in answering a questionnaire (arrows represent conditional dependencies).

The problem under study in this work spans different areas, such as health care and therapy, education, work environments and public spaces. This results in a variety of possible applications, e.g., a robot physiotherapist, which assists a patient in a physical recovery/rehabilitation, a robot guiding visitors in a museum, a robot play partner, among others.

## 3.2 Framework for planning in social HRI scenarios

This Section proposes a general DT model for social HRI scenarios. It takes into account the particularities of the problem previously described and defines the framework, based on POMDPs-IR, for decision-making of the agent. The POMDP-IR model of Figure 3.2 represents the proposed framework. It is displayed as a two-stage DBN network to highlight variable dependencies. The states, observations and actions, along with the transition, observation and reward models that constitute the proposed framework are further detailed in this section.

### 3.2.1 States and Transition Model

In the described scenario, the agent considers two types of state factors: the *task* variables $T$ and the *person* variables $P$. The *task* variables model the environment features that provide information on the progress of the tasks. On the other hand, the *person* variables track the human status and are inherently latent. The latter are used to gain information on the human user's affective and motivational status and adapt the robot behavior accordingly.

The number of variables depend on the amount of features essential to represent the environment and is, therefore, dependent on the specific task. The criteria for the selection of states involve a trade-

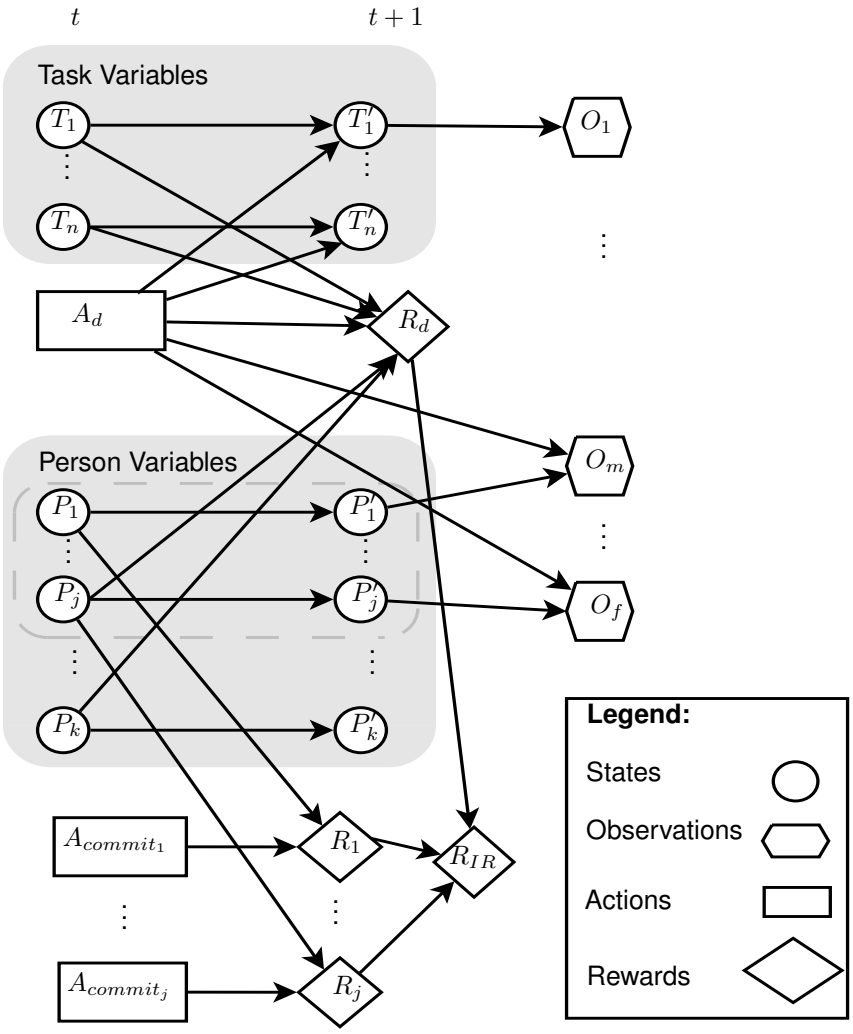



**Figure 3.2:** DT framework for modeling HRI problems represented as a DBN

off between operational complexity and predicted system performance, since operational complexity increases with the number of states.

Furthermore, depending on the objectives of the agent acting in a HRI setting, the *task* variables might not exist. This is the case when the single goal of the agent is to gain information on the human user, e.g., a robot psychologist.

Table 3.1 exploits some examples of robot systems employed in a social HRI setting and proposes *task* and *person* variables for each of them. The robot physiotherapist is employed in assisting a patient in a constraint-induced movement therapy, which consists of repeatedly moving an affected limb. The agent's goal is to track and evaluate the exercise, represented in the *task* variable $Exercise$, motivating the user whenever the movement is inadequately or not performed at all. The type of motivation (e.g., challenging or nurturing) is adjusted with regard to the user's affective and motivational status, represented in the *person* variables $Fatigue$ and $Personality$.

**Table 3.1:** Example of *person* and *task* variables for different social HRI scenarios

| Scenario | *Person Variables* | *Task Variables* |
|---|---|---|
| Robot Physiotherapist | – Fatigue<br>– Personality | – Exercise |
| Robot Guide in Museum | – Interest | – Tour |
| Robot Play Partner | – Enjoyment<br>– Preferences | – Game |

The robot guide in a museum keeps track of the stage of the tour and decides on what to present according to the *task* variable $Tour$. Furthermore, the agent might, for instance, extend the explanation or move on to the next stage of the tour, depending on a certain measure of the interest of the user represented in the *person* variable $Interest$.

Finally, a humanoid robotic play partner is capable of playing different games (e.g., football and chess), represented in the *task* variable $Game$. It might decide on which game to play based on the motivational status and the preferences of the user, represented in the *person* variables $Enjoyment$ and $Preferences$. The examples present in the table consider interactions with only one person. However, the application can be extended to more users if the DT model includes the *person* variables for each person involved in the interaction (e.g., $Enjoyment_i, i = 1, \ldots, k$ where $k$ is the number of users).

A *person* variable can be constant if its value does not change during the task. This is the case of personal traits (e.g., $Personality$ and $Preferences$ in Table 3.1), which are relevant for the robot behavior and do not alter for the duration of the interaction. In Figure 3.2, $P_k$ represents a constant *person* variable. The value of $P_k$ at each time step only depends on the value of the same variable in the previous time step. Otherwise, *person* variables are inferred from the user's behavior (factors $P_1$ to $P_j$ in Figure 3.2), which is represented in the model's observations. These may consist of state factors of interest, according to the POMDP-IR framework.

### 3.2.2 Observations and Observation Model

In a social HRI setting, observations reflect the user's behavior. This behavior is used to monitor the progress of the task and infer the user's affective and motivational status.

Observations are discrete, symbolic values, classified from sensory data, which correspond to features of the environment that are observable in a given state.

The observation factors are contingent on the sensory capabilities of the robot system. Nevertheless, the correct understanding of the user's status relies on the agent being capable of recognizing human communication methods. Humans exchange messages verbally and non-verbally, through speech and gestures [45], respectively. Consequently, the robot system ought to be able to recognize speech and

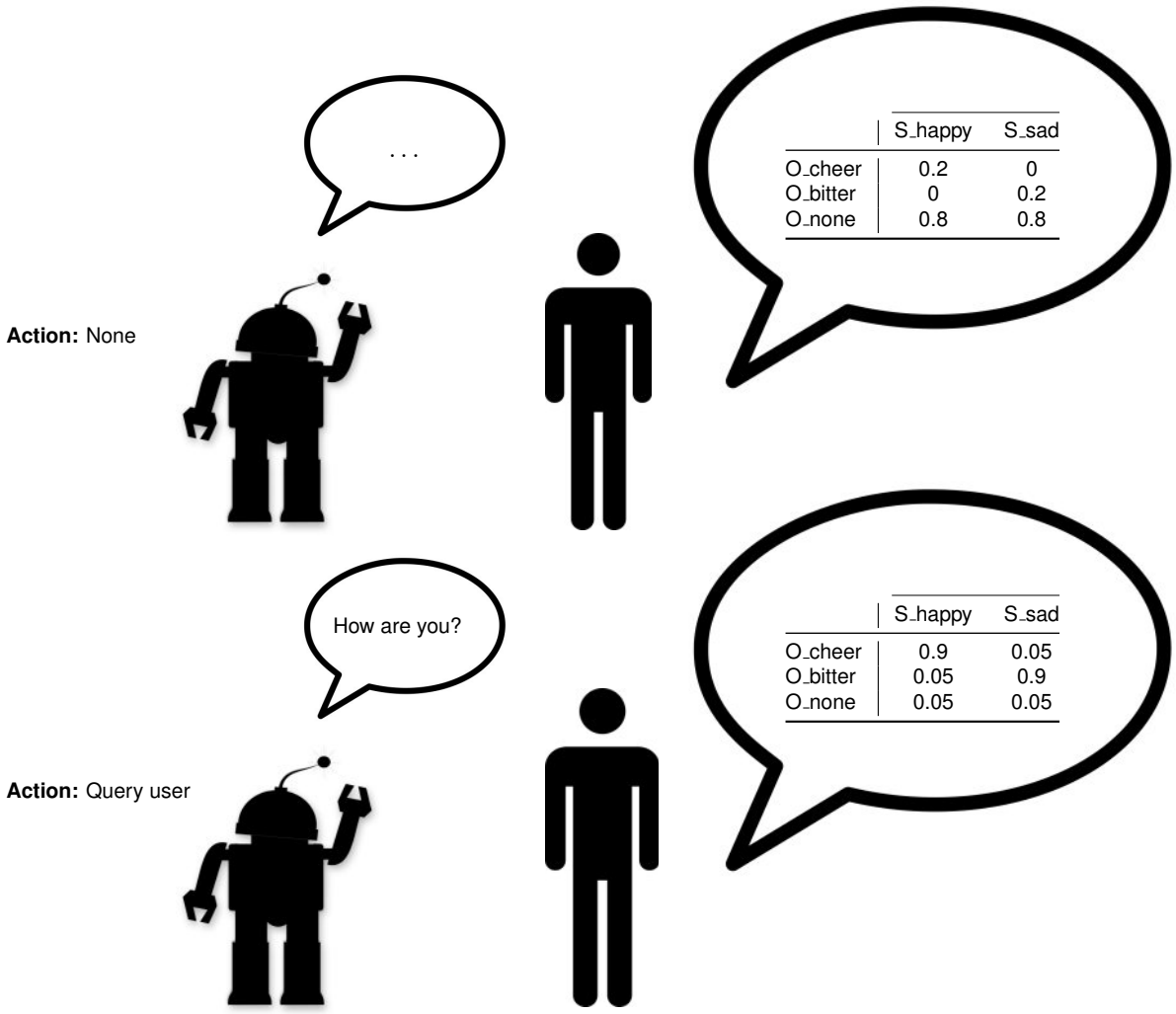



**Figure 3.3:** Example of the observation model of a social robot

gestures in order to understand the human user's affective and motivational status.

The observation model is of key importance in the achievement of the information gain goals of the agent. It reflects the probability of receiving a certain observation, given the state of the environment and the action performed. Certain actions, such as questioning or approaching the user, increase the probability of perceiving certain observations. This fact is of utter importance to actively gain information on the user's status. The dependency on the action is represented in observations $O_m$ to $O_f$ in Figure 3.2.

Figure 3.3 illustrates the observation model of a robot with an information gain goal. The user is considered to be either happy or sad, i.e., in the state $P = \{S_{happy}, S_{sad}\}$. At each time step, the agent perceives $\Omega = \{O_{none}, O_{cheer}, O_{bitter}\}$, which correspond to the user not speaking and saying he/she feels happy or sad, respectively. The robot might do nothing or query the user regarding his/hers

feelings: $A = \{None, Query\,User\}$. For a given state, doing nothing clearly results in a lower probability of perceiving a verbal reaction from the user. Otherwise, querying the user increases the probability of observing either $O_{cheer}$ or $O_{bitter}$ (e.g., $p(\Omega = O_{cheer}|P = S_{happy}, A = Query\,User) > p(\Omega = O_{cheer}|P = S_{happy}, A = None)$ ), resulting in a lower uncertainty regarding the affective status of the user. Naturally, for a given action, the probability of perceiving $\Omega = O_{cheer}$ is higher when the user is in state $S_{happy}$, just as $p(\Omega = O_{bitter}|P)$ is higher if $P = S_{sad}$.

### 3.2.3 Actions

The model of Figure 3.2 comprehends two types of actions: $A_d$ and $A_{commit}$. The first have an effect on the environment and is dependent on the actuators of the agent, while the latter are used for the information gain goals of the agent, as discussed in Section 2.4.

Typically, the action domain $A_d$ contains the minimum set of functionalities which allow the agent to complete its tasks. Social robots, however, need to communicate in a natural, easily understandable way with the human users. To achieve this objective, the robot must be able to express different moods and emotions, similar to what humans do [46]. Consequently, the action domain $A_d$ of a social robot ought to include speech and/or gestural capabilities and/or graphical emotion displays. As an example, Figure 3.4 shows the robot used in the INSIDE[1] project displaying different emotions.

Following the POMDP-IR framework, besides the domain-level action factor $A_d$, the model has additional action factors for each state factor of interest $A_{commit}$. The state factors of interest, in the problem under study, are included in the *person* variables, as these contain the aforementioned affective and motivational state of the human user. The actions $A_{commit}$ allow rewarding the agent for decreasing the uncertainty regarding particular features of the environment.

### 3.2.4 Reward Model

Generally, there is no definitive criteria to define the reward model, as rewards are defined over the abstract states and actions of the DT model. Therefore, a policy with satisfactory practical quality is usually obtained through a process of trial and error, where different reward models are used.

In the DT model of Figure 3.2, rewards are either associated with task objectives: $R_d$, or with the information gain goals: $R_i, i = 1, \ldots, j$. The sum of these rewards, $R_{IR}$, constitute the reward awarded to the agent at each time step.

The behavior of the robot consists of the sequence of domain actions $A_d$ the agent performs. In the social HRI scenario, and in order to adapt the robot's behavior to the user's affective and motivational status, the reward assigned to an action depends not only on the *task* variables but also on the *person*

[1] http://project-inside.pt/

**(a)** Shy and happy display

**(b)** Excitement display

**Figure 3.4:** Emotions displayed by the robot used in the INSIDE project

variables.

The information rewards $R_i$ influence the behavior of the agent, with the purpose of achieving a low uncertainty regarding certain *person* variables. The value of these rewards are dependent on the threshold of knowledge required.

### 3.2.5 On the Estimation of the Stochastic Models

Figure 3.2 represents a model-based solution for the problem defined in Section 3.1. The model, as discussed in Chapter 2, requires the definition of transition $T$ and observation $O$ functions. To obtain these functions, the problem designer needs to estimate the respective probability distributions.

One way to estimate the transition and observation models of a POMDP is by collecting experimental data. In this situation, the problem is similar to estimating the structure of a Hidden Markov Model (HMM), and the problem designer might use the Baum-Welch algorithm [47]. However, in the social HRI scenario, learning the model structure from data might prove a difficult task, due to the lack of well labeled data.

Another common method of modeling the stochastic environment is to simulate the physical system. This approach allows to collect a large number of transition/observation samples and to simplify the estimation problem, since the exact state is accessible. Nevertheless, robotic simulators are not capable of simulating humans and their stochastic behavior as of yet.

Finally, the probability distributions can reflect common knowledge on the problem under study. In

these cases, the models can be approximately estimated by means of the expertise of the problem designer, resulting, nevertheless, in policies with good practical quality.

# 4

# Case-Study in Socially Assistive Robotics: Robot Therapist

## Contents

This Chapter explores a particularly challenging task for a social robot in a Socially Assistive Robotics (SAR) scenario: rehabilitation therapy. First, it proposes a Decision-Theoretic (DT) model based on the framework defined in Section 3.2. Furthermore, the model is implemented in a networked robot platform in a series of experiments, with the purpose of validating the projected DT framework.

## 4.1 Scenario

The present case study considers a scenario of physical rehabilitation and training, namely post-stroke rehabilitation. Stroke is one of the major public health problems of the growing elderly population [48], and causes patients to suffer from limited extremity function, affecting their everyday functional movements and activities. This, however, can be mitigated through rehabilitation therapy, which involves repetitive exercises where the patient moves the stroke-affected limb as prescribed [49].

Rehabilitation therapy includes passive or active exercises. In the first, the therapist (human or robot) physically assists the patient to move the affected limb. On the other hand, in active exercises, the patient moves the affected limb by him/herself, while the therapist has the functions of coaching and motivating. SAR clearly has the potential to enhance physical recovery for individuals with rehabilitation needs [3], as it provides innovative ways to monitor, motivate and coach patients.

Up to date research in rehabilitation robotics covers mainly passive exercises: examples are the MIT-Manus [50] and the Lokomat [51] projects. Nevertheless, social robots, and SAR in particular, provide a way to approach active rehabilitation exercises, for instance: in project AHA[1], where the robot Vizzy's goal is to assist patients in physiotherapy; and in [52], where a hands-off robot therapist assists cardiac patients in a spirometry exercise.

Overall, the goals of the robot therapist in the considered rehabilitation scenario are:

- To help the user in the given setting, by monitoring the patient's movements (e.g., encourages the patient to continue if he/she stops performing the exercise);

- To adapt its behavior and, consequently, the therapy style (e.g., nurture or challenge the patient), in accordance with the patient's affective and motivational status.

Figure 4.1 illustrates the scenario of this case-study, in a situation where the robot actively seeks to understand the patient's motivational status and reacts accordingly, with the goal of driving the user to proceed with the exercise.

[1] http://aha.isr.tecnico.ulisboa.pt/

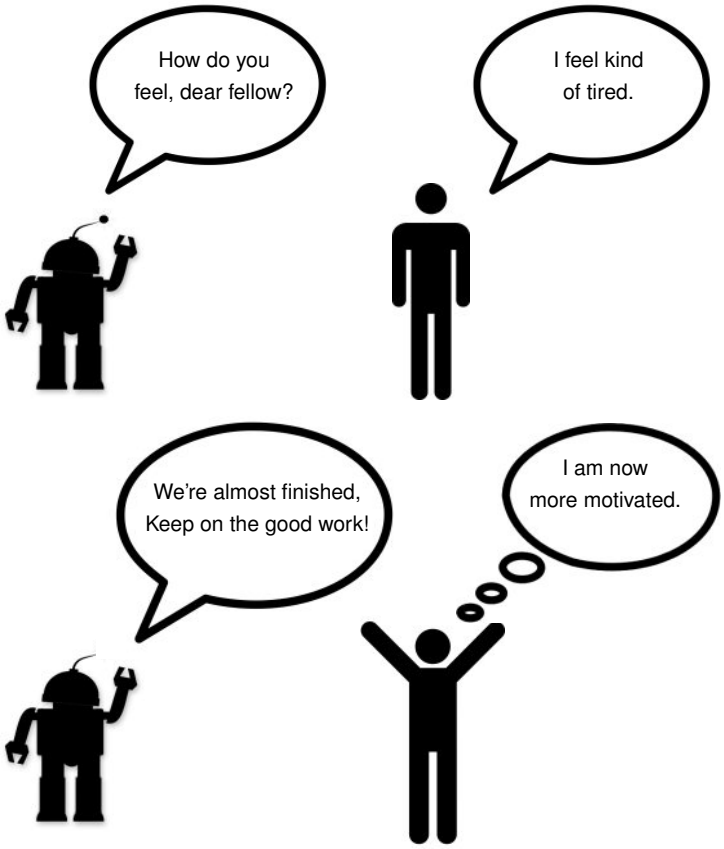



**Figure 4.1:** Illustration of an active exercise in robotic physical rehabilitation therapy

## 4.2 Decision-Theoretic Model for the Robot Therapist

This section describes the DT model, which is represented in Figure 4.2, for the robot operating in the aforementioned physical rehabilitation scenario.

### 4.2.1 States

The significant features of the environment in which the robot is to operate are related to the patient. The fulfillment of the task's objectives require that the agent keeps track of the patient's movements, possesses knowledge regarding relevant personal traits of the patient and infers his/hers affective status. Therefore, the proposed DT model considers the state space of Figure 4.3, represented in factored form.

The patient's movement is encoded in the *task* state factor *Exercise* ($Exer.$). When the exercise is performed as prescribed, the state factor assumes the value *correct*: $Exer. = Correct$. Otherwise, if the movement is inappropriately or not performed, $Exer. = Incorrect$. The state factor *Personality* ($Pers.$) is a constant *person* variable, known beforehand by the problem designer, which represents the patient's behavioral personality, as *Introverted* or *Extroverted*. Finally, the *Fatigue* state factor ($Fat.$) is a measure

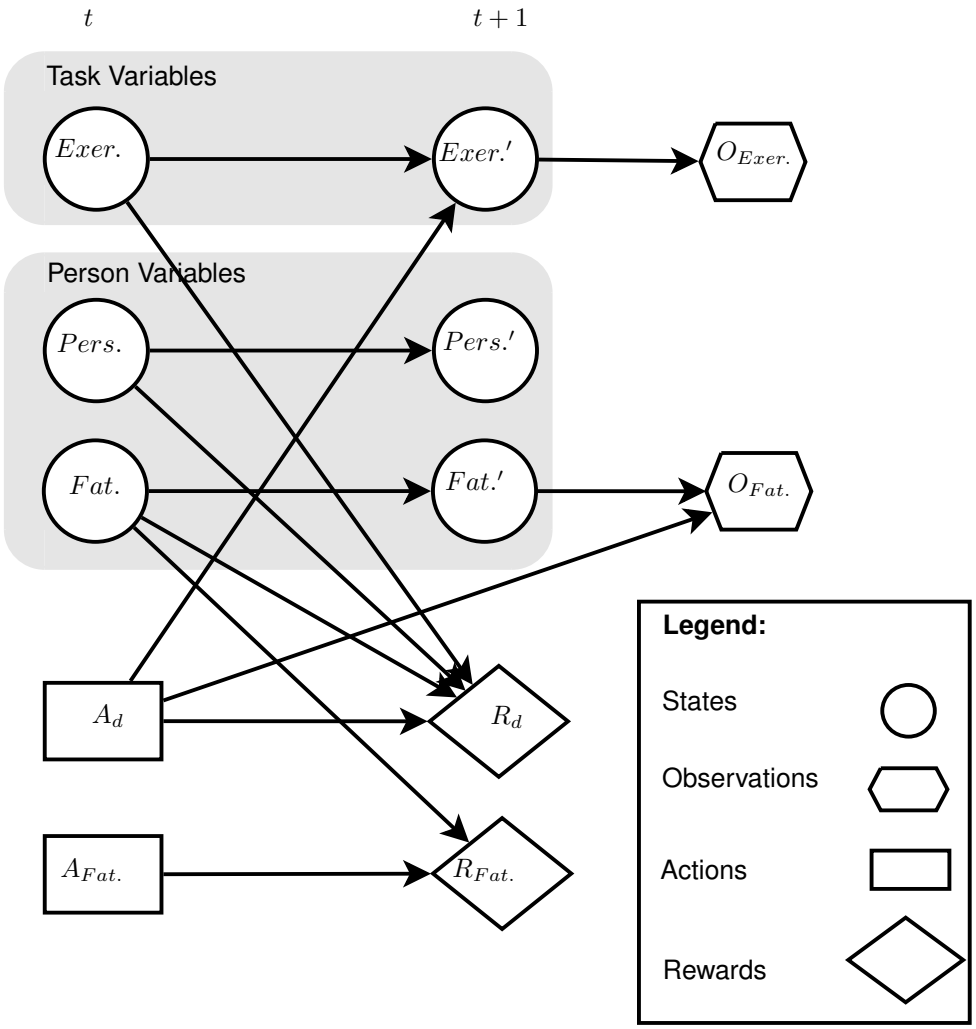



**Figure 4.2:** DBN representation of the DT model for the robot therapist

of the patient's weariness, caused by the physical exercise. It assumes the values *Tired* or *Energized* whether the patient shows signs of fatigue or liveliness, respectively.

### 4.2.2 Observations

The observation space is represented, in factored form, in Figure 4.4. Observations reflect the relevant behavior of the patient, in accordance with the task's goals. In the present case study, the agent ought to evaluate the movement performed by the patient and to infer his/hers affective status. Therefore, the observation factors considered in the DT model are:

- The gesture related $O_{Exer.}$, which is used to evaluate the exercise and assumes, as a result, the values *Proper* or *Wrong*. $O_{Exer.} = Proper$ whenever the agent perceives that the patient performed the movement as prescribed. Otherwise, $O_{Exer.} = Wrong$ if the agent perceives that

$$\text{Exercise: } Exer. = \begin{cases} Correct, \\ Incorrect. \end{cases}$$

$$\text{Personality: } Pers. = \begin{cases} Introverted, \\ Extroverted. \end{cases}$$

$$\text{Fatigue: } Fat. = \begin{cases} Tired, \\ Energized. \end{cases}$$

**Figure 4.3:** State Space of the POMDP model of the robot therapist case study

$$\text{Gesture: } O_{Exer.} = \begin{cases} Proper, \\ Wrong. \end{cases}$$

$$\text{Affective Status: } O_{Fat.} = \begin{cases} Weary, \\ Energetic, \\ None. \end{cases}$$

**Figure 4.4:** Observation Space of the POMDP model of the robot therapist case study

the patient did not perform the movement or performed it incorrectly;

- $O_{Fat.}$, which is related to the affective status of the patient represented in state factor *Fatigue*, and assumes the values *Weary*, *Energetic* or *None*. $O_{Fat.} = Weary$ or $O_{Fat.} = Energetic$ when the patient demonstrates feeling tired or lively, respectively. Otherwise, $O_{Fat.} = None$ if the agent does not perceive any relevant information regarding the affective status of the patient.

In this case study, $O_{Exer.}$ is obtained by visual classification of the patient's gestures and $O_{Fat.}$ through speech interaction, i.e., through classification of the user's verbal responses.

### 4.2.3 Actions

The proposed DT model considers two action factors: the *Action Domain* $A_d$ and the *IR Action* $A_{Fat.}$. At each time step, the agent chooses one value for each action factor. The possible values for the action factors are represented in Figure 4.5.

The IR action is defined according to the POMDP-IR framework, with a *commit* action for each value of the related state factor ($Fat.$), and a *null* action. $A_{Fat.}$ allows rewarding the agent for reducing the uncertainty regarding the state factor $Fat.$, related to the patient's fatigue.

The *Action Domain* $A_d$ contains the set of functionalities which allow the agent to achieve its goals, which are, in this case study, to monitor and motivate the patient in an active physical rehabilitation exercise. That is, whenever the patient stops performing the exercise or performs it incorrectly, the robot encourages him/her to proceed with the exercise.

$$\text{Action Domain: } A_d = \begin{cases} Nurture, \\ Challenge, \\ Query\ Patient, \\ End\,Therapy, \\ None. \end{cases}$$

$$\text{IR Action: } A_{Fat.} = \begin{cases} Commit\ Tired, \\ Commit\ Energized, \\ Null. \end{cases}$$

**Figure 4.5:** Action Space of the POMDP model of the robot therapist case study

The therapy style, i.e., the robot's approach to the patient changes as a function of his/hers *Fatigue* and *Personality*. Dependent on these factors, the encouragement is classified as $Nurture$ or $Challenge$ whether the agent opts for a softer (e.g. "You are doing great! Keep on the good work.") or a more defiant approach (e.g., "You can do better than that!").

Since the therapy style is dependent on the *person* variables, it is important to gain information and maintain a low uncertainty regarding the state factors $Pers.$ and $Fat.$. As $Pers.$ is constant, the agent only actively seeks to reduce uncertainty on the state factor $Fat.$, through the $Query\ Patient$ action. This action consists of verbally interacting with the patient to infer his/hers *Fatigue*.

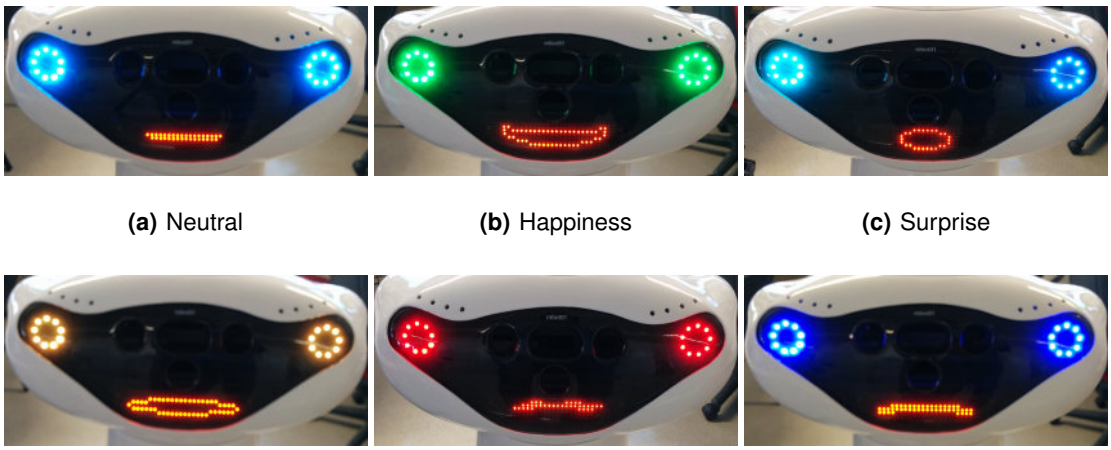
Moreover, the agent ought to end the exercise ($End\,Therapy$) when the patient persistently shows he/she is not able to proceed with it. Finally, at each time step, the agent might choose to do nothing ($None$).

Besides adapting speech in conformance with the behavior of the patient, the robot also modifies the emotion displayed to the more appropriate in the given situation. The set of emotions the robot therapist is able to display is represented in Figure 4.6.

### 4.2.4 Transition, Observation and Reward Functions

The proposed framework allows to take into account the effects of time in the states of the DT model. Namely, in the current case study, the transition function $T$ encodes that $b(Fat.) = Tired$ increases at each time step in the absence of opposing observations ($O_{Fat.} = Energetic$). That is, the agent realistically believes that the patient is feeling more tired over time. Also, the transition function of this case study dictates that the probability of the patient correctly performing the exercise ($Exer. = Correct$) increases with the motivation actions ($Nurture$ or $Challenge$).

The observation function $O$ encodes the error in sensory data classification. This means, for instance, that even if the patient's gesture is classified as incorrect ($O_{Exer.} = Wrong$), the agent's belief on $Exer. = Incorrect$ is not 100% and the robot might require more information before motivating



**(a)** Neutral **(b)** Happiness **(c)** Surprise

**(d)** Fear **(e)** Anger **(f)** Sadness

**Figure 4.6:** Set of emotions displayed by the robot therapist [53]

the patient. Furthermore, the probabilities in $O$ take into account that information-gathering actions (such as $Query\ Patient$) increase the probability of perceiving a verbal reaction from the user (e.g., $O_{Fat.} = Weary$).

The time step of the synchronous decision-making loop (i.e., the time that elapses between decision episodes), needs to consider the rate of classification of sensory data. The agent has multiple sensors and respective classification systems, operating at different frequencies. Consequently, the decision-making loop rate needs to be equal or lower than the lowest sensory operating frequency. In the present case-study, the time step of the decision-making loop is $5$ seconds.

The DT model of Figure 4.2 rewards IR actions ($R_{Fat.}$) and $A_d$ actions ($R_d$). The information rewards are defined, in accordance with the POMDP-IR framework, so that the agent actively seeks to have a belief on $Fat. = Tired$ or $Fat. = Energized$ greater than 75%: $b(Fat. = Tired) > 0.75$ or $b(Fat. = Energized) > 0.75$. The agent receives a reward of $0.19$ whenever the commit IR action matches the current state ($A_{Fat.} = Commit\ Tired$ and $Fat. = Tired$ or $A_{Fat.} = Commit\ Energized$ and $Fat. = Energized$) and a reward of $-0.57$ otherwise. Actions in $A_d$ are rewarded in accordance with the state of the environment:

- *Encouragement* actions ($Nurture$ and $Challenge$) are rewarded $0.2$ whenever the patient is incorrectly performing the exercise or $0.1$ when he/she shows signs of feeling tired, and penalized $-0.1$ otherwise. The reward given to each action also depends on the state factor $Pers.$: for an *Introverted* person, the $Nurture$ action is preferred while the $Challenge$ action is favored for an *Extroverted* person;

- The $Query\ Patient$ action is penalized with $-0.2$;

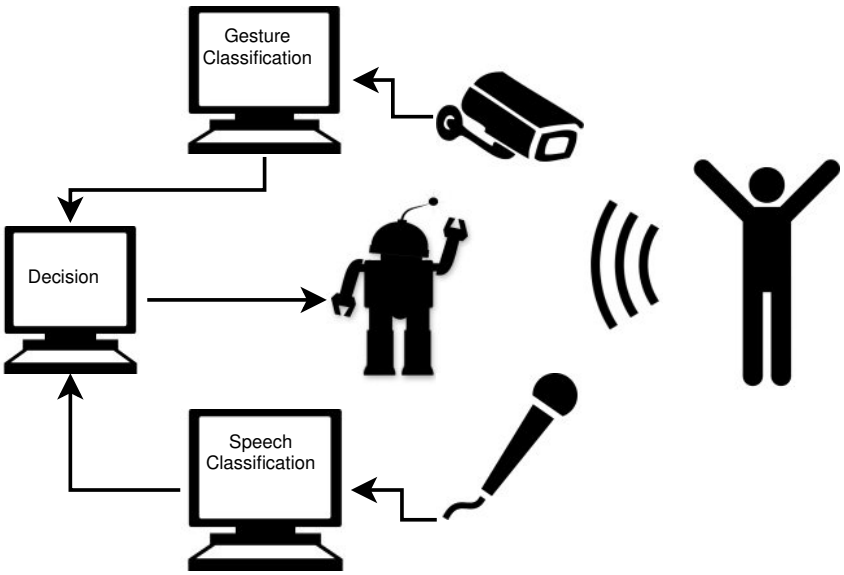



**Figure 4.7:** Components of the experimental setup

- $None$ is not rewarded nor penalized $\Big(R_d(A_d = None) = 0\Big)$;
- $EndTherapy$ receives high penalization when the patient feels energetic $\Big(R_d(A_d = EndTherapy, Fat. = Energetic) = -1\Big)$ and a reward of $0.1$ otherwise.

The discount factor in this case study is $0.9$.

## 4.3 Experimental Setup

The networked robot system used in the present case study consists of the MOnarCH robot platform [54] and an external Kinect camera. The robot platform provides the actuating capabilities required to implement the domain actions $A_d$ defined in Section 4.2.3 and the sensors necessary for the speech related observations $O_{Fat.}$. The Kinect camera is strategically located for a clear view of the patient's movements and is used, therefore, for the classification of the exercise $O_{Exer.}$, in accordance with the observations described in 4.2.2.

The sensory information is, after classification, used as input to the decision system that controls the actuators of the robot platform.

Figure 4.7 demonstrates the components of the experimental setup and their connection. Arrows represent directions of communication within the networked system. Communication between the elements of the NRS is based on the ROS middleware[2].

The experiments within this case study took place in the ISRobotNet@Home Testbed[3] [55]. This

[2] http://www.ros.org/about-ros/
[3] http://welcome.isr.tecnico.ulisboa.pt/isrobonet/

**(a)** Robotic platform used in the experiments

**(b)** Living room area of the ISRoboNet@Home testbed

**Figure 4.8:** Scenario of the Experiments

testbed provides the infrastructure to implement networked robot systems in a domestic environment. In particular, the experiments considered the living room area which is represented in Figure 4.8.

### 4.3.1 On the classification of sensory data

In accordance with the observation space of the DT model, the patient's movement is to be classified as *Correct* or *Incorrect*, at each time step. Likewise, in order to infer the patient's affective status, and as the preferred means of communication is through verbal interaction, the patient's speech is classified as: Demonstrative of the patient feeling $Weary$ or $Energetic$; $None$ if it does not add relevant information.

#### 4.3.1.1 Gesture Classification

Classification of the patient's movement is achieved resorting to a Kinect-based application, which makes use of a gesture database previously built through the Visual Gesture Builder (VGB)[4] tool, available in the *Kinect for Windows* Software Development Kit (SDK). First, within VGB, the system designer tags frames in recorded video clips, which are related to meaningful gestures. These tagged frames are, then, used as inputs to the detection algorithm during the training stage. On the application runtime, the detection technologies detect discrete and continuous gestures. Discrete gesture classification outputs a Boolean indicating if the user is performing a trained gesture and a confidence level on the Boolean classification. Continuous gesture classification results in a float indicating the progress of the user as

[4] https://msdn.microsoft.com/en-us/library/dn785529.aspx

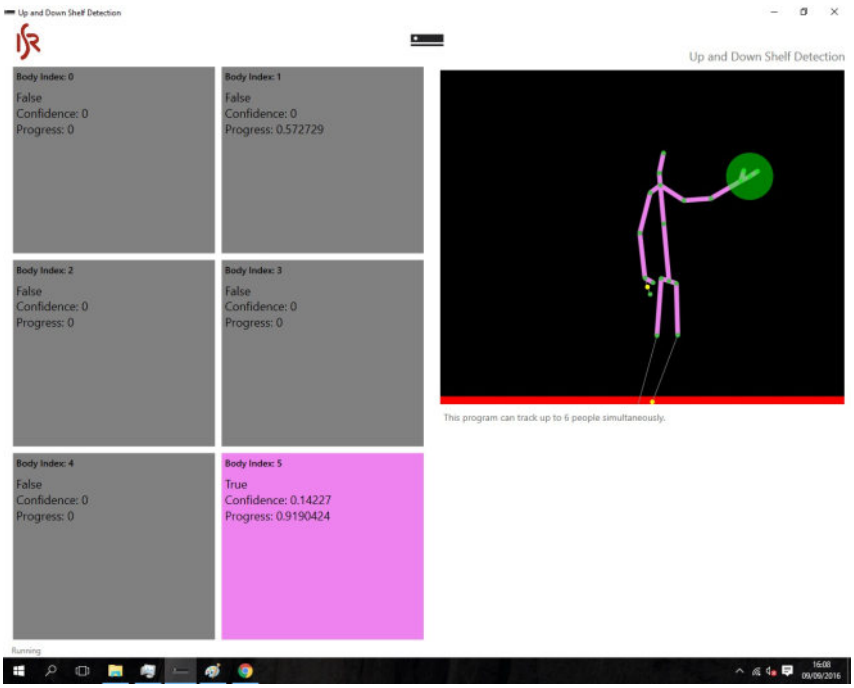



**Figure 4.9:** Interface of the application for gesture classification

he/she performs the gesture.

Figure 4.9 represents the interface of the application developed for recognition of the patient's movement, within the considered case study. The application is able to track up to six persons simultaneously, although only one person is considered in each of the experiments performed in this case study. Moreover, the application outputs the classification of the gestures perceived as *correct* or *incorrect* (True or False), whether the gesture corresponds to the training input or not. The interface also shows the confidence level on the Boolean classification and the percentage on the movement's progress. In accordance with the considered case study, the trained gesture consists of an up and down arm movement, similar to moving a book between upper and lower shelves (the actual therapy scenario).

#### 4.3.1.2 Speech Classification

Automatic Speech Recognition (ASR) is based on VoCon Hybrid[5], a state of the art commercial solution. This speech recognition engine is based on context-free grammars, written in Backus-Naur Form (BNF), which encode the utterances to be recognized. The grammars are created with prior knowledge of the scenarios that the robot needs to understand. Speech understanding follows the definition of a corpus over the context-free grammars, which spawns the possible sentences the ASR recognizes.

[5] http://www.nuance.com/for-business/speech-recognition-solutions/vocon-hybrid/index.htm

### 4.3.2 Decision System

The decision system used in the present case study is based on the Symbolic Perseus solver [39]. Symbolic Perseus uses the original Perseus algorithm and ADD as the underlying data structure.

The policy is computed offline, in order to save computational resources during the online execution of the task. The online computation, in the decision system, consists on the belief update and the selection of the action in accordance with the updated belief and the previously calculated policy.

## 4.4 Results

The experiments performed in this case study intend to prove that the proposed DT model is able to perform a given task with different persons, while taking into account the *person* (latent) variables to adapt the agent's behavior.

Each experiment considers a different user, which is classified according to his/hers personality, as defined in Section 4.2.1 (i.e., as introverted or extroverted), and with regard to his/hers ability to perform the exercise (athletic or unfit).

The experiments carried out within this work were recorded and are available at `https://www.youtube.com/playlist?list=PLp1xxEiDsjtBcYGJuoiwBytoK5ZdMDvIv`.

### 4.4.1 Experiment A: Extroverted Athletic User

Experiment A considers a user which is classified as extroverted ($Pers. = Extroverted$) and athletic. Figure 4.10 represents the data obtained in the experiment, namely the observations, actions and the belief on the two key state factors considered: $Fat.$ and $Exer.$. Figure 4.14(a) shows an episode of the interaction between the user and the robot, where the robot motivates the user. The video of the complete experiment is available at `https://youtu.be/25veR8NFIBU`.

The user feels energetic for the first fifty seconds (decision step 10), approximately, and tired afterwards.

At the beginning, the robot chooses not to act, since the exercise is well performed and the agent has a low uncertainty regarding the *fatigue* status of the user. This uncertainty on the state factor $Fat.$, however, increases over time, driving the robot to actively seek to reduce it, by querying the user (decision step 3). The answer ($O_{Fat.} = Energetic$), informs the robot that the user is still active and motivated, increasing the certainty on $Fat. = Energized$. This behavior is repeated until the user does not perform correctly the exercise ($O_{Exer.} = Incorrect$) in decision step 11. Then, the robot motivates the person through a challenging approach due to the considered *personality* of the user and the current *fatigue* status. Following these events, the agent's uncertainty on the state factor $Fat.$

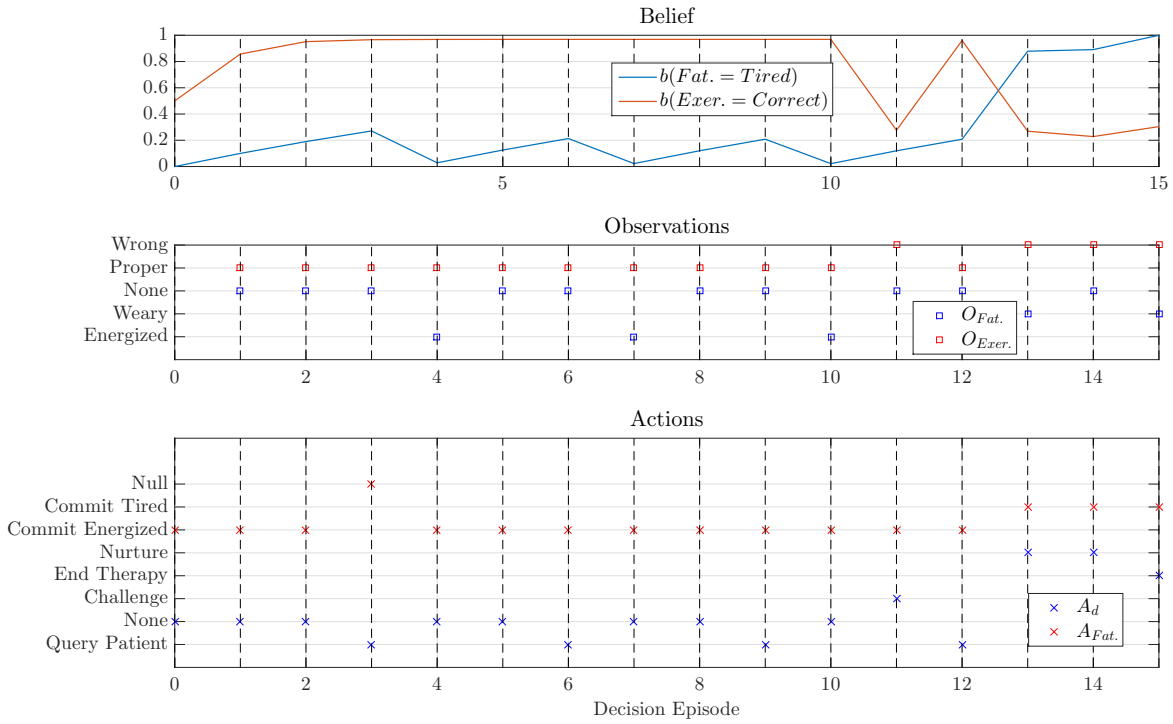



**Figure 4.10:** Experiment A: Evolution of the Belief on the states $Fat.$ and $Exer.$ w.r.t. the decision episode, the observations received and the actions performed

increased and the robot queries the user in decision step 12. After receiving information that the user now feels tired ($O_{Fat.} = Weary$), the robot changes therapy style and adopts a nurturing approach. As the user continuously shows not being able to carry out the exercise and the certainty on $Fat. = Tired$ increases, the robot finally chooses to end the therapy in decision step 15.

### 4.4.2 Experiment B: Extroverted Unfit User

Figure 4.11 represents the data obtained in experiment B, which considers a user classified as extroverted ($Pers. = Extroverted$) and unfit. Figure 4.14(b) shows an episode of the experiment where the robot queries the user. The video of the full experiment is available at `https://youtu.be/z4ZKKPCwBb4`.

The user feels energetic for the first forty seconds, approximately, and tired afterwards.

The behavior of the robot is similar to the previous experiment while the user shows feeling energetic and correctly performs the exercise. Nonetheless, the user incorrectly performs the exercise more often, at which occasions the robot acts in motivating with a challenging approach while the agent believes the user to feel motivated/energetic. Even though motivating the user, the robot keeps track of his/hers *fatigue* and reacts when the uncertainty on $Fat.$ is too high $\Big(b(Fat. = Tired) < 0.75$ and $b(Fat. = Energized) < 0.75\Big)$. Finally, the agent ends the therapy once it persistently observes the user is not performing the exercise and feels tired.

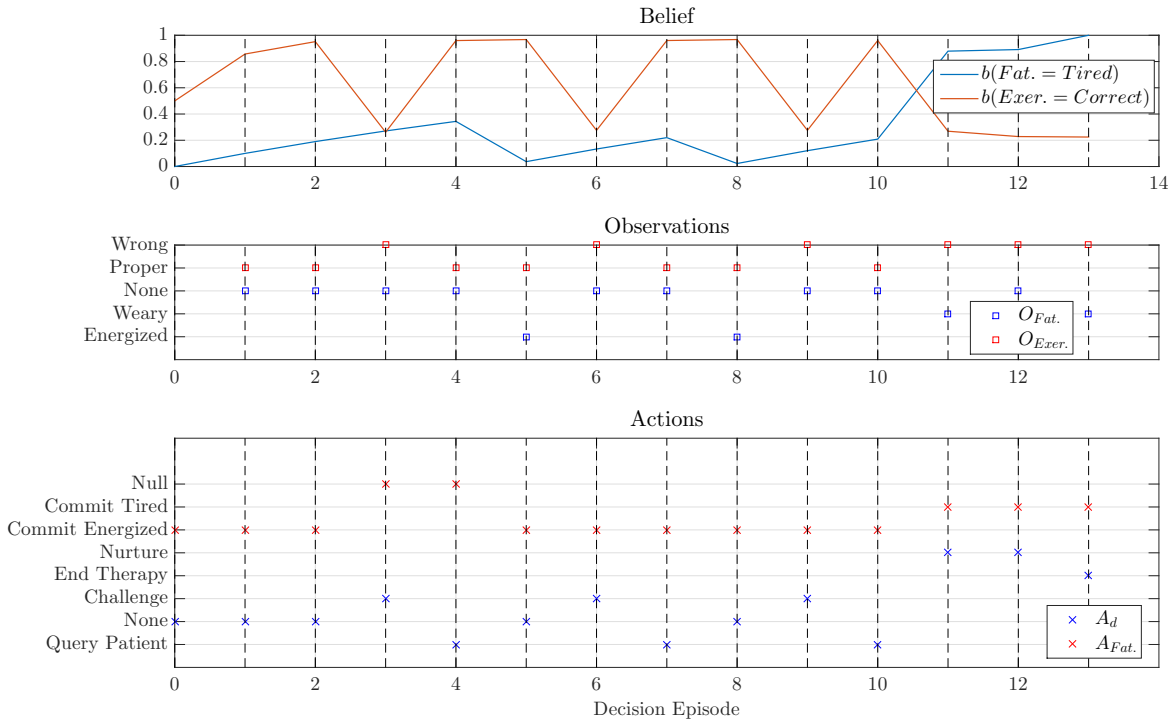



**Figure 4.11:** Experiment B: Evolution of the Belief on the states $Fat.$ and $Exer.$ w.r.t. the decision episode, the observations received and the actions performed

### 4.4.3 Experiment C: Introverted Athletic User

The data obtained in experiment C is represented in Figure 4.12. This experiment considers a user classified as introverted ($Pers. = Introverted$) and athletic. Figure 4.14(c) shows an episode of the experiment where the robot motivates the user through a nurturing approach. The video of the complete experience is available at https://youtu.be/4sJPwC3aGc0.

The patient feels energetic up to, approximately, 45 seconds (decision step 9), and tired afterwards.

The behavior of the robot is heavily dependent on its knowledge regarding the fatigue status of the user. While the uncertainty on the $Fat.$ state factor is high $\Big(b(Fat. = Tired) < 0.75$ and $b(Fat. = Energized) < 0.75\Big)$, the robot queries the user. Since the uncertainty on $Fat.$ increases over time, the agent performs the action $Query\ Patient$ until it perceives an answer $O_{Fat} = Energetic$ or $O_{Fat} = Weary$ (decision steps 3 and 4 / 7 and 8). Nevertheless, the robot performs the therapy task while actively gathering information on the environment, motivating the user once the belief on $b(Fat. = Tired)$ is high, and ending the therapy appropriately.

### 4.4.4 Experiment D: Introverted Unfit User

The final experiment is represented in Figure 4.13 and contemplates a user which is classified as introverted ($Pers. = Introverted$) and unfit. Figure 4.14(d) shows an episode of the experiment where the

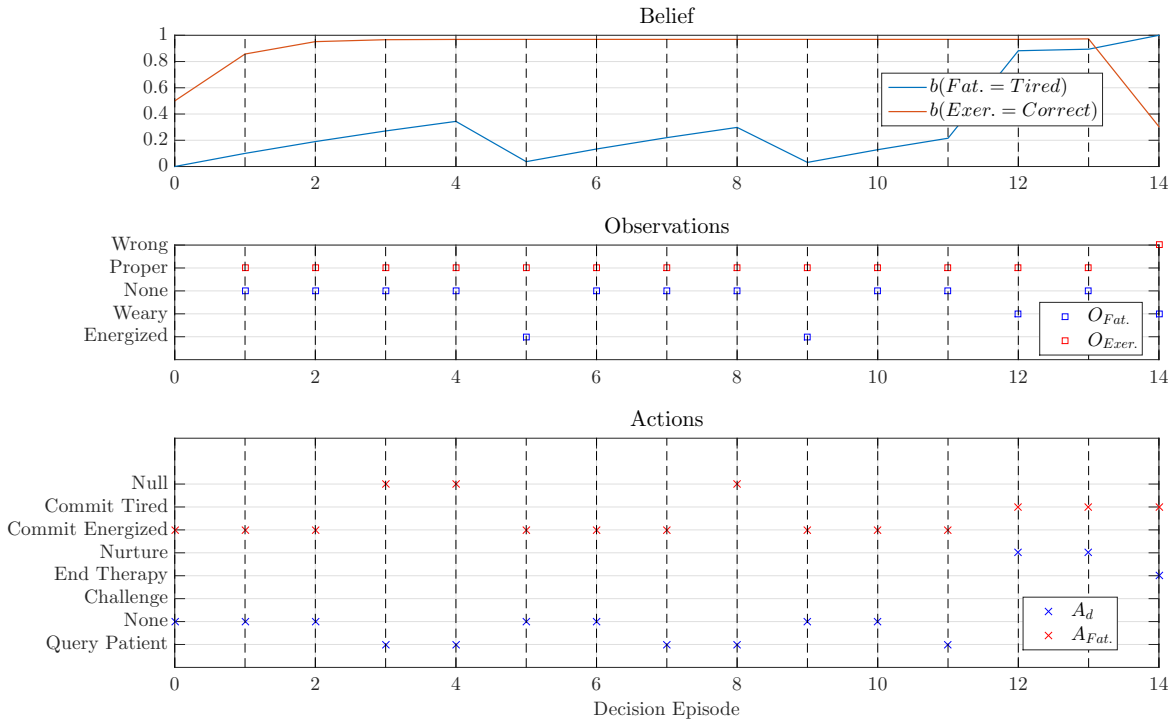



**Figure 4.12:** Experiment C: Evolution of the Belief on the states $Fat.$ and $Exer.$ w.r.t. the decision episode, the observations received and the actions performed

robot ends the therapy. The video of the full experiment is available at https://youtu.be/caT5NamuMVg.

The user feels energetic for the first 40 seconds (decision step 8), approximately, and tired onward.

The behavior of the robot changes in accordance with its belief on the states of the environment. In the present experiment, there is a "trade-off" between motivating or querying the user depending on the belief over the state factors $Fat.$ and $Exer.$. In decision step 3, the agent queries the agent due to the high uncertainty on $Fat.$. Afterwards, the agent perceives no answer but observes the user incorrectly performed the movement. This observation does not translate, however, into an absolute certainty on the exercise having been incorrectly performed $\Big(b_4(Exer. = Correct) \approx 0.3\Big)$, since the DT framework takes into account sensor related noise. The agent, then, queries the user once again (decision step 4), due to the increasing uncertainty on the *fatigue* of the user. Once again, the NRS receives no answer ($O_{Fat.} = None$), and observes the user incorrectly performed the movement. This time, the agent's belief on $Exer. = Incorrect$ is higher $\Big(b_5(Exer. = Incorrect) \approx 0.95\Big)$ and it motivated the user. Nevertheless, the uncertainty on $Fat.$ is still high on decision step 6 and the robot once again queries the user, perceiving this time an answer.

During the rest of the experiment, the robot once again queries the user when the uncertainty on $Fat.$ is high (decision steps 10 & 11) and motivates the user in accordance with the beliefs on the variables $Exer.$ and $Fat.$ (decision steps 9, 12 & 13). Finally, the agent ends the therapy in decision step 14.

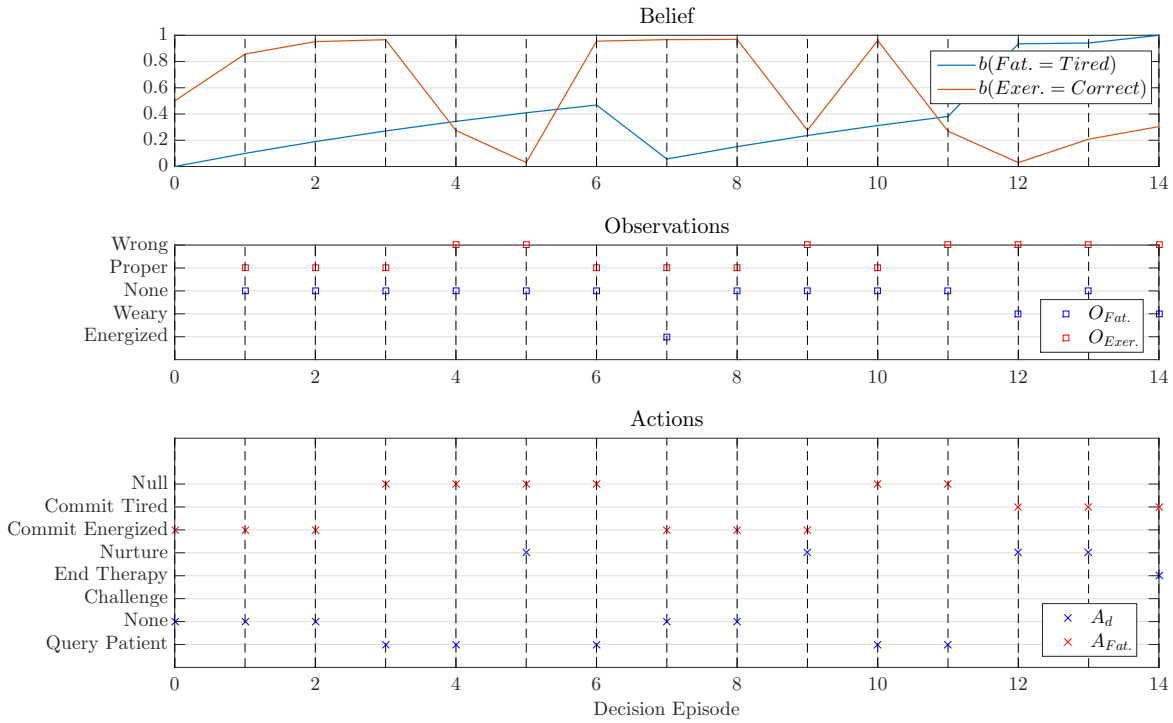



**Figure 4.13:** Experiment D: Evolution of the Belief on the states $Fat.$ and $Exer.$ w.r.t. the decision episode, the observations received and the actions performed

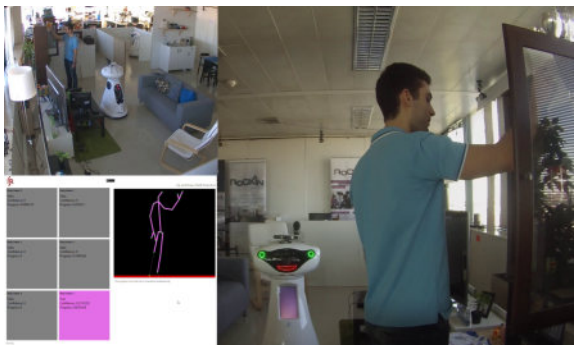
### 4.4.5 Discussion

The goal of the experiments is to demonstrate the effectiveness and the potential of the proposed DT approach to planning in social robotics. Therefore, the experiments consider the variations that can occur within the considered scenario, i.e., different sequences of observations, in order to evaluate the agent's behavior.

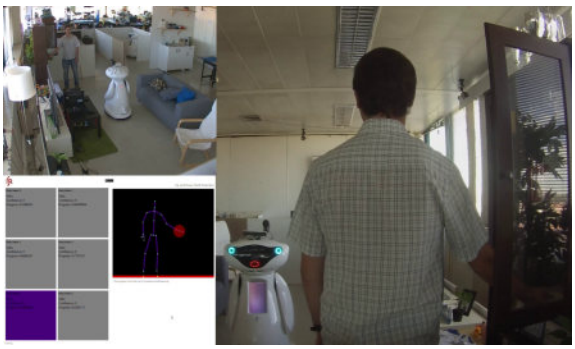
As to what concerns the validity of the experiments, they are replicable considering any system capable of faithfully perceiving the considered observations and performing the enumerated actions. The experiments involved different persons with different behaviors, personalities and athletic build, performing the same task. Therefore, they serve as a valid basis of evaluation for the proposed DT approach, taking into account the objectives of the projected framework (to allow the agent to perform and complete a given task(s) and adapt its behavior in accordance with the user's status).

The agent is expected to determine the real fatigue status of the user as rapidly as possible while maintaining a socially acceptable behavior, i.e., not querying nor motivating the user constantly. Also, the robot is expected to perform the therapy task, motivating the user to proceed the exercise and ending the therapy session when appropriate.

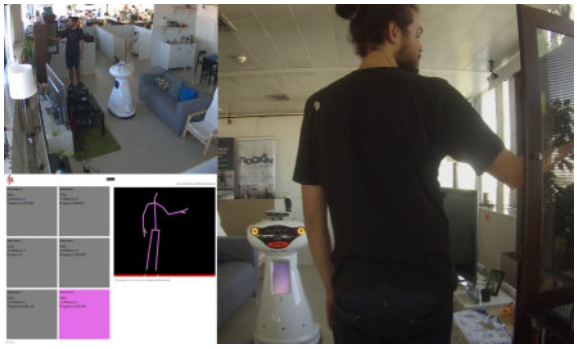
Table 4.1 details the behavior of the robot for each experiment. As expected: the number of motivation actions is higher for the users classified as unfit, which incorrectly perform the exercise more often



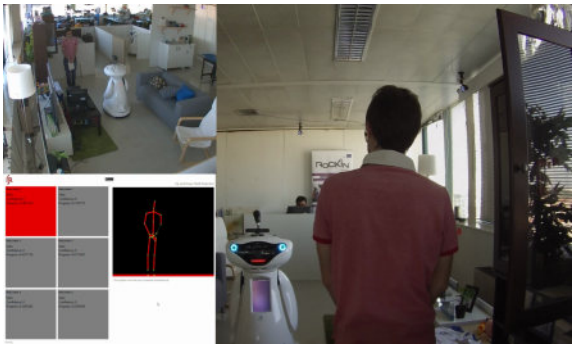
**(a)** Scene of experiment A: Robot challenges the user to proceed with the exercise.



**(b)** Scene of experiment B: Robot queries the user with regard to his status.



**(c)** Scene of experiment C: Robot motivates the user through a nurturing approach.



**(d)** Scene of experiment D: Robot ends the therapy.

**Figure 4.14:** Episodes of the experiments where the robot interacts with the user. In each figure: Right and top left images show different views of the ISRobotNet@Home Testbed; Bottom left image represents the interface of the gesture classification application.

**Table 4.1:** Behavior of the robot with regard to the experiment

| | Number of motivation actions | Number of query actions | Time elapsed until agent detected $Fat. = Tired$ | Time elapsed until $End\ Therapy$ since $b(Fat. = Tired) > 0.8$ | Duration of the experiment |
|---|---|---|---|---|---|
| **Experiment A** | 3 | 4 | 15 s | 10 s | 75 s |
| **Experiment B** | 5 | 3 | 15 s | 10 s | 65 s |
| **Experiment C** | 2 | 5 | 15 s | 10 s | 70 s |
| **Experiment D** | 4 | 5 | 20 s | 10 s | 70 s |

than the athletic users; and the number of query actions is higher for the users classified as introverted.

The robot detected the fatigue status change from $Energized$ to $Tired$ in all the experiments, taking between, approximately, 15 seconds (experiments 1, 2 & 3) to 20 seconds (experiment 4), to have a high belief on $Fat. = Tired$ $\Big(b(Fat. = Tired) > 0.8\Big)$ from the time the user started to feel tired. Moreover, the agent motivated the user upon detection of faulty movements, either immediately after observing $O_{Exer.} = Wrong$ (experiments 1, 2 and 3) or after two consecutive observations (experiment

4) depending on the belief over the state factors $Fat.$ and $Exer.$. Finally, the agent ended the therapy when consistently observing the user was not capable of proceeding with the exercise, after 10 seconds, approximately, of having a high certainty $\Big(b(Fat. = Tired) > 0.8\Big)$ on the user feeling tired.

Overall, the DT approach to planning in the robot therapist resulted in a behavior capable of achieving the task and information goals, adaptive to the user's status and socially appealing.

# 5

# Conclusion

## Contents

To conclude, this chapter reviews the main contributions of this Dissertation, and examines potential directions for future research.

## 5.1 Conclusions

This work studied the problem of decision making under uncertainty in social Human-Robot Interaction (HRI).

First, the problem under study was translated to the development of a Decision-Theoretic (DT) framework to:

- Complete a given task;
- Infer the latent status of the user(s);
- Gather information on the environment.

In the considered social HRI scenario, uncertainty stems from: the effects of the robot's actions in the environment, which include the human user, and are uncertain; and the noise inherent to real world sensory information. Task planning under uncertainty problems are naturally modeled as a Markov Decision Process (MDP), which provides the mathematical framework for decision making in stochastic environments. In particular, when taking into account partial observability of the environment, these problems fit into the Partially Observable Markov Decision Process (POMDP) framework. Classic POMDP formulation, however, is not optimized for gathering information simultaneously with other objectives. The Partially Observable Markov Decision Process with Information Reward (POMDP-IR) framework, on the other hand, overcomes the information gathering limitations of POMDPs and is, therefore, the most appropriate basis of the DT framework for the problem under study.

Building on the POMDP-IR framework, Chapter 3 introduces a DT approach to planning in social HRI, which includes:

- Task-related variables, which model environmental features that represent the progress on the task(s);
- Hidden person-related variables, which track the human users' affective and motivational status;
- Gesture and speech rooted observations;
- An action domain which contains the functionalities that allow the agent to complete its task(s) and gather information on the environment;
- IR actions which allow rewarding the agent for reducing the uncertainty on variables of interest.

Under the proposed framework, the agent is capable of achieving its task and information goals while following a socially appealing behavior. Moreover, the agent adapts its behavior in accordance with the affective and motivational status of the user.

The properties of the DT framework for social HRI are demonstrated in Chapter 4, through a case study inserted in Socially Assistive Robotics (SAR): the robot therapist. In this setting, the robot system ought to help the user in an active rehabilitation exercise, which consisted of repeatedly moving one arm up and down. The agent's task was to monitor and correct/motivate the user if the exercise was incorrectly performed. Moreover, the robot was expected to infer and adapt its behavior in accordance with the affective and motivational status of the user. The case study included experiments where the Network Robot System (NRS) interacted with different persons within the described scenario. Overall, the robot therapist achieved the task and information goals, besides adapting to the user's status. The experiments' results prove the validity of the proposed framework for problems involving robots systems in HRI scenarios.

## 5.2 Future Work

Solution methods for MDP-based models, such as the framework proposed in this work, present an important practical issue since they assume complete knowledge of the stochastic models (Transition and Observation models). Besides, any change to the parameters of these models imply a recalculation of the DT policy. On the other hand, Reinforcement Learning (RL) approaches [56] assume either absent or imperfect knowledge on the environment dynamics. The DT policies are learned, in this case, from the interaction of robotic agents with their environment. RL methods can support on the structure and properties of the proposed model to overcome the aforementioned implementation issues.

To further validate the framework developed within this work, further experiments ought to be performed, in particular considering cases of social assistance with real patients. This would involve the implementation of a DT-based NRS in a real therapy scenario, in order to evaluate the behavior of the robot system as to what concerns the accomplishment of its goals, its social qualities and adaptability.

Moreover, the proposed framework ought to be tested in a scenario which considers: more latent variables, in particular hidden variables of interest; more information-gathering actions; and more complex actions, e.g., manipulation.

# Bibliography


[1] K. Wada, T. Shibata, T. Saito, K. Sakamoto, and K. Tanie, "Psychological and social effects of one year robot assisted activity on elderly people at a health service facility for the aged," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 2785–2790.

[2] D. Feil-seifer and M. J. Matarić, "Defining Socially Assistive Robotics," in *International Conference on Rehabilitation Robotics (ICORR)*, Chicago, IL, June 2005, pp. 465–468.

[3] A. Tapus, M. J. Matarić, and B. Scassellati, "Socially assistive robotics [Grand Challenges of Robotics]." *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 35–42, 2007.

[4] C. A. Costescu, B. Vanderborght, and D. O. David, "The effects of robot-enhanced psychotherapy: A meta-analysis," *Review of General Psychology*, vol. 18, no. 2, pp. 127–136, June 2014.

[5] D. F. Glas, "The network robot system: Enabling social robots in the real world," Ph.D. dissertation, Osaka University, Toyonaka, Osaka, Japan, March 2013.

[6] T. Veiga, "Information Gain and Value Function Approximation in Task Planning Using POMDPs," Ph.D. dissertation, Instituto Superior Técnico, 2015.

[7] T. Taha, J. V. Miró, and G. Dissanayake, "A pomdp framework for modelling human interaction with assistive robots," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 544–549.

[8] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Margaritis, M. Montemerlo, J. Pineau, J. Schulte, and S. Thrun, "Towards personal service robots for the elderly," in *Workshop on Interactive Robots and Entertainment (WIRE 2000)*, 2000.

[9] A. Gimenez, C. Balaguer, A. M. Sabatini, and V. Genovese, "The MATS robotic system to assist disabled people in their home environments," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, Oct. 2003, pp. 2612–2617.

[10] Y. Fernaeus, M. Håkansson, M. Jacobsson, and S. Ljungblad, “How do you play with a robotic toy animal?: A long-term study of pleo,” in *Proceedings of the 9th International Conference on Interaction Design and Children*, 2010, pp. 39–48.

[11] E. ja Hyun, S. yeon Kim, S. Jang, and S. Park, “Comparative study of effects of language instruction program using intelligence robot and multimedia on linguistic ability of young children,” in *RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication*, Aug 2008, pp. 187–192.

[12] W. D. Stiehl, J. Lieberman, C. Breazeal, L. Basel, L. Lalla, and M. Wolf, “Design of a therapeutic robotic companion for relational, affective touch,” in *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, Aug 2005, pp. 408–415.

[13] S. E. Fasoli, H. I. Krebs, J. Stein, W. R. Frontera, and N. Hogan, “Effects of robotic therapy on motor impairment and recovery in chronic stroke,” *Archives of Physical Medicine and Rehabilitation*, vol. 84, no. 4, pp. 477 – 482, 2003.

[14] R. Beira, M. Lopes, M. Praca, J. Santos-Victor, A. Bernardino, G. Metta, F. Becchi, and R. Saltaren, “Design of the robot-cub (iCub) head,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 94–100.

[15] T. Fong, I. Nourbakhsh, and K. Dautenhahn, “A survey of socially interactive robots,” *Robotics and Autonomous Systems*, vol. 42, no. 3–4, pp. 143 – 166, 2003.

[16] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, “Minerva: a second-generation museum tour-guide robot,” in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 3, 1999, pp. 1999–2005.

[17] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, “Towards robotic assistants in nursing homes: Challenges and results,” *Special issue on Socially Interactive Robots, Robotics and Autonomous Systems*, vol. 42, no. 3 - 4, pp. 271 – 281, 2003.

[18] J. Fasola and M. J. Mataric, “Robot exercise instructor: A socially assistive robot system to monitor and encourage physical exercise for the elderly,” in *19th International Symposium in Robot and Human Interactive Communication*, Sept 2010, pp. 416–421.

[19] C. D. Kidd and C. Breazeal, “Robots at home: Understanding long-term human-robot interaction,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2008, pp. 3230–3235.

[20] H. L. Akin, A. Birk, A. Bonarini, G. Kraetzschmar, P. Lima, D. Nardi, E. Pagello, M. Reggiani, A. Saffiotti, A. Sanfeliu, and M. Spaan, “Two ”hot issues ” in cooperative robotics: Network robot systems, and formal models and methods for cooperation a white paper from the euron special interest group on cooperative robotics,” 2008.

[21] T. Akimoto and N. Hagita, “Introduction to a network robot system,” in *2006 International Symposium on Intelligent Signal Processing and Communications*, Dec 2006, pp. 91–94.

[22] A. Foka and P. Trahanias, “Real-time hierarchical POMDPs for autonomous robot navigation,” *Robotics and Autonomous Systems*, vol. 55, no. 7, pp. 561 – 571, 2007.

[23] J. Messias, “Decision-Making under Uncertainty for Real Robot Teams,” Ph.D. dissertation, Instituto Superior Técnico, 2014.

[24] J. Hoey, P. Poupart, A. v. Bertoldi, T. Craig, C. Boutilier, and A. Mihailidis, “Automated Handwashing Assistance for Persons with Dementia Using Video and a Partially Observable Markov Decision Process,” *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 503–519, May 2010.

[25] T. Taha, J. V. Miro, and G. Dissanayake, “Pomdp-based long-term user intention prediction for wheelchair navigation,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, May 2008, pp. 3920–3925.

[26] R. Bellman, “A Markovian Decision Process,” *Indiana Univ. Math. J.*, vol. 6, pp. 679–684, 1957.

[27] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[28] E. J. Sondik, “The optimal control of partially observable markov processes over the infinite horizon: Discounted costs,” *Oper. Res.*, vol. 26, no. 2, pp. 282–304, Apr. 1978.

[29] ——, “The optimal control of partially observable markov processes over the infinite horizon: Discounted costs,” *Operations Research*, vol. 26, no. 2, pp. 282–304, 1978.

[30] G. E. Monahan, “A survey of Partially Observable Markov Decision Processes: Theory, models, and algorithms,” *Management Science*, vol. 28, no. 1, pp. 1–16, January 1982.

[31] A. Cassandra, M. L. Littman, and N. L. Zhang, “Incremental pruning: A simple, fast, exact method for partially observable markov decision processes,” in *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI’97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 54–61.

[32] W. S. Lovejoy, “Computationally feasible bounds for partially observed markov decision processes,” *Operations Research*, vol. 39, no. 1, pp. 162–175, 1991.

[33] E. A. Hansen, “Solving pomdps by searching in policy space,” *CoRR*, vol. abs/1301.7380, 2013.

[34] M. Hauskrecht, “Value-function approximations for partially observable markov decision processes,” *CoRR*, vol. abs/1106.0234, 2011.

[35] J. Pineau, G. Gordon, and S. Thrun, “Point-based value iteration: An anytime algorithm for pomdps,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, August 2003, pp. 1025 – 1032.

[36] J. Messias, “Learning for Robots,” 2016, University of Lisbon. [Online]. Available: https://surfdrive.surf.nl/files/index.php/s/FwqIl8fnI5afXZt?path=%2Fslides

[37] M. T. J. Spaan and N. Vlassis, “Perseus: Randomized point-based value iteration for pomdps,” *J. Artif. Int. Res.*, vol. 24, no. 1, pp. 195–220, Aug. 2005.

[38] C. Boutilier and D. Poole, “Computing optimal policies for partially observable decision processes using compact representations,” in *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, ser. AAAI’96, 1996, pp. 1168–1175.

[39] P. Poupart, “Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes,” Ph.D. dissertation, University of Toronto, Toronto, Ont., Canada, 2005.

[40] J. Hoey, R. St-aubin, A. Hu, and C. Boutilier, “Spudd: Stochastic planning using decision diagrams,” in *In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1999, pp. 279–288.

[41] K. A. Bollen, “Latent variables in psychology and the social sciences,” *Annual Review of Psychology*, vol. 53, no. 1, pp. 605–634, 2002.

[42] I. Leite, C. Martinho, and A. Paiva, “Social robots for long-term interaction: A survey,” *International Journal of Social Robotics*, vol. 5, no. 2, pp. 291–308, 2013.

[43] M. Hoffman, *Empathy and Moral Development: Implications for Caring and Justice*. Cambridge University Press, 2001.

[44] I. Leite, A. Pereira, S. Mascarenhas, C. Martinho, R. Prada, and A. Paiva, “The influence of empathy in human-robot relations,” *Int. J. Hum.-Comput. Stud.*, vol. 71, no. 3, pp. 250–260, Mar. 2013.

[45] P. Wagner, Z. Malisz, and S. Kopp, “Gesture and Speech in Interaction: An Overview,” *Speech Communication*, vol. 57, no. Special Iss., pp. 209–232, 2014.

[46] R. Kirby, J. Forlizzi, and R. Simmons, “Affective social robots,” *Robot. Auton. Syst.*, vol. 58, no. 3, pp. 322–332, Mar. 2010.

[47] S. Koenig and R. Simmons, “Xavier: A robot navigation architecture based on partially observable markov decision process models,” in *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, R. B. D. Kortenkamp and R. Murphy, Eds. MIT Press, 1998, pp. 91 – 122.

[48] M. S. V. Elkind, “Stroke in the elderly,” *The Mount Sinai journal of medicine, New York*, vol. 70, no. 1, p. 27—37, January 2003.

[49] W. SL, W. CJ, M. J, and et al, “Effect of constraint-induced movement therapy on upper extremity function 3 to 9 months after stroke: The excite randomized clinical trial,” *JAMA*, vol. 296, no. 17, pp. 2095–2104, 2006.

[50] H. I. Krebs, M. Ferraro, S. P. Buerger, M. J. Newbery, A. Makiyama, M. Sandmann, D. Lynch, B. T. Volpe, and N. Hogan, “Rehabilitation robotics: pilot trial of a spatial extension for mit-manus,” *Journal of NeuroEngineering and Rehabilitation*, vol. 1, no. 1, pp. 1–15, 2004.

[51] R. Riener, L. Lunenburger, S. Jezernik, M. Anderschitz, G. Colombo, and V. Dietz, “Patient-cooperative strategies for robot-aided treadmill training: first experimental results,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 380–394, Sept 2005.

[52] K. I. Kang, S. Freedman, M. J. Mataric, M. J. Cunningham, and B. Lopez, “A hands-off physical therapy assistance robot for cardiac patients,” in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, June 2005, pp. 337–340.

[53] K. Baraka, A. Paiva, and M. Veloso, *Expressive Lights for Revealing Mobile Service Robot State*. Cham: Springer International Publishing, 2016, pp. 107–119.

[54] J. Messias, R. Ventura, P. Lima, J. Sequeira, P. Alvito, C. Marques, and P. Carriço, “A robotic platform for edutainment activities in a pediatric hospital,” in *Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on*, May 2014, pp. 193–198.

[55] M. Barbosa, A. Bernardino, D. Figueira, J. Gaspar, N. Gonçalves, P. U. Lima, P. Moreno, A. Pahliani, J. Santos-Victor, M. T. J. Spaan, and J. Sequeira, “ISRobotNet: A testbed for sensor and robot network systems,” in *Proc. of International Conference on Intelligent Robots and Systems*, 2009, pp. 2827–2833.

[56] T. Jaakkola, S. P. Singh, and M. I. Jordan, “Reinforcement learning algorithm for partially observable markov decision problems,” in *Advances in Neural Information Processing Systems 7*. MIT Press, 1995, pp. 345–352.

# A

# Support Information

This appendix presents suplementary material concerning the case study of this work. Namely, it defines the transition, observation and reward models, which are represented as ADDs.

## A.1 Transition Model

Figures A.1 and A.2 represent the Conditional Probability Distribution (CPD) of the state factors $Exer.$ and $Fat.$, which, according to the Decision-Theoretic (DT) model defined in Chapter 4, correspond to $P\big(Exer(t+1) \mid A_d(t), Exer(t)\big)$ and $P\big(Fat(t+1) \mid Fat(t)\big)$.

The transition function $T$ of the factored model is, therefore, represented as:

$$\begin{aligned} &P\Big(Exer(t+1), Fat(t+1) \;\Big|\; A_d(t), A_{Fat.}, Exer.(t), Fat.(t), Pers.(t)\Big) \\ &\quad = P\Big(Exer(t+1) \;\Big|\; A_d(t), Exer(t)) \cdot P(Fat(t+1)|Fat(t)\Big). \end{aligned}$$

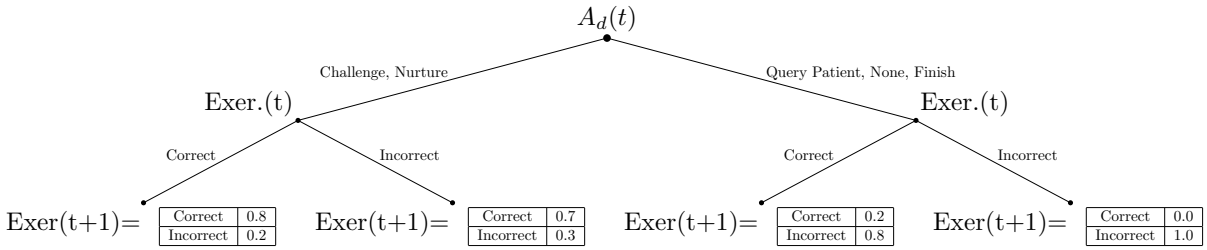



**Figure A.1:** Conditional Probability Distribution of state factor $Exer.$.

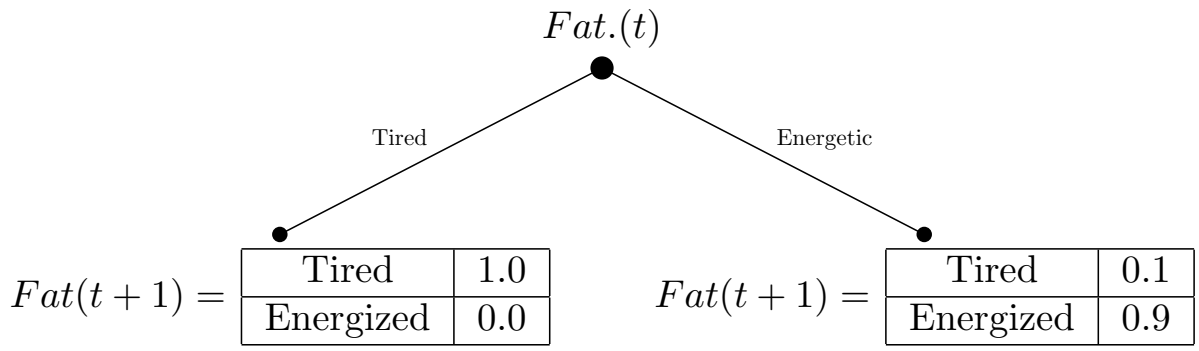



**Figure A.2:** Conditional Probability Distribution of state factor $Fat.$.

## A.2 Observation Model

The CPD of observation factors $O_{Exer.}$ and $O_{Fat.}$ is represented in Figures A.3 and A.4, respectively. In accordance with the variables' dependencies, these probability distributions correspond to $P\big(O_{Exer.}(t+1) \,\big|\, Exer.(t+1)\big)$ and $P(O_{Fat.}(t+1) \,\big|\, A_d(t), Fat.(t+1)\big)$.

The observation function $O$ of the factored model is, therefore, represented as:

$$\begin{aligned} &P\Big(O_{Exer}(t+1), O_{Fat}(t+1) \,\Big|\, A_d(t), A_{Fat.}, Exer.(t+1), Fat.(t+1), Pers.(t+1)\Big) \\ &\quad = P\Big(O_{Exer}(t+1) \,\Big|\, Exer.(t+1)) \cdot P(O_{Fat.}(t+1) | A_d(t), Fat.(t+1)\Big). \end{aligned}$$

## A.3 Reward Model

Figures A.5 and A.6 represent the reward functions $R_d$ and $R_{Fat.}$, respectively. The rewarded given to the agent at each time step is the sum of these functions: $R_{IR} = R_d + R_{Fat.}$.

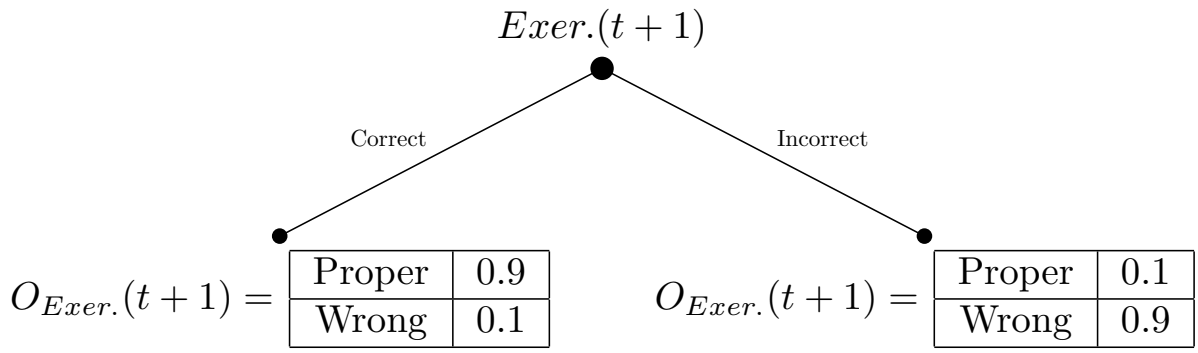



**Figure A.3:** Conditional Probability Distribution of observation factor $O_{Exer.}$.

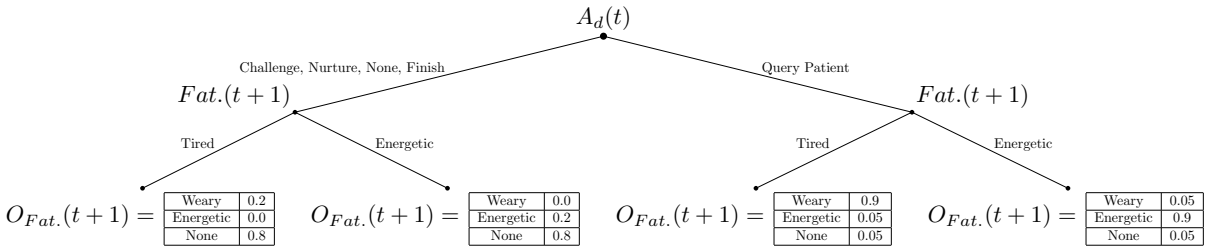



**Figure A.4:** Conditional Probability Distribution of observation factor $O_{Fat.}$.

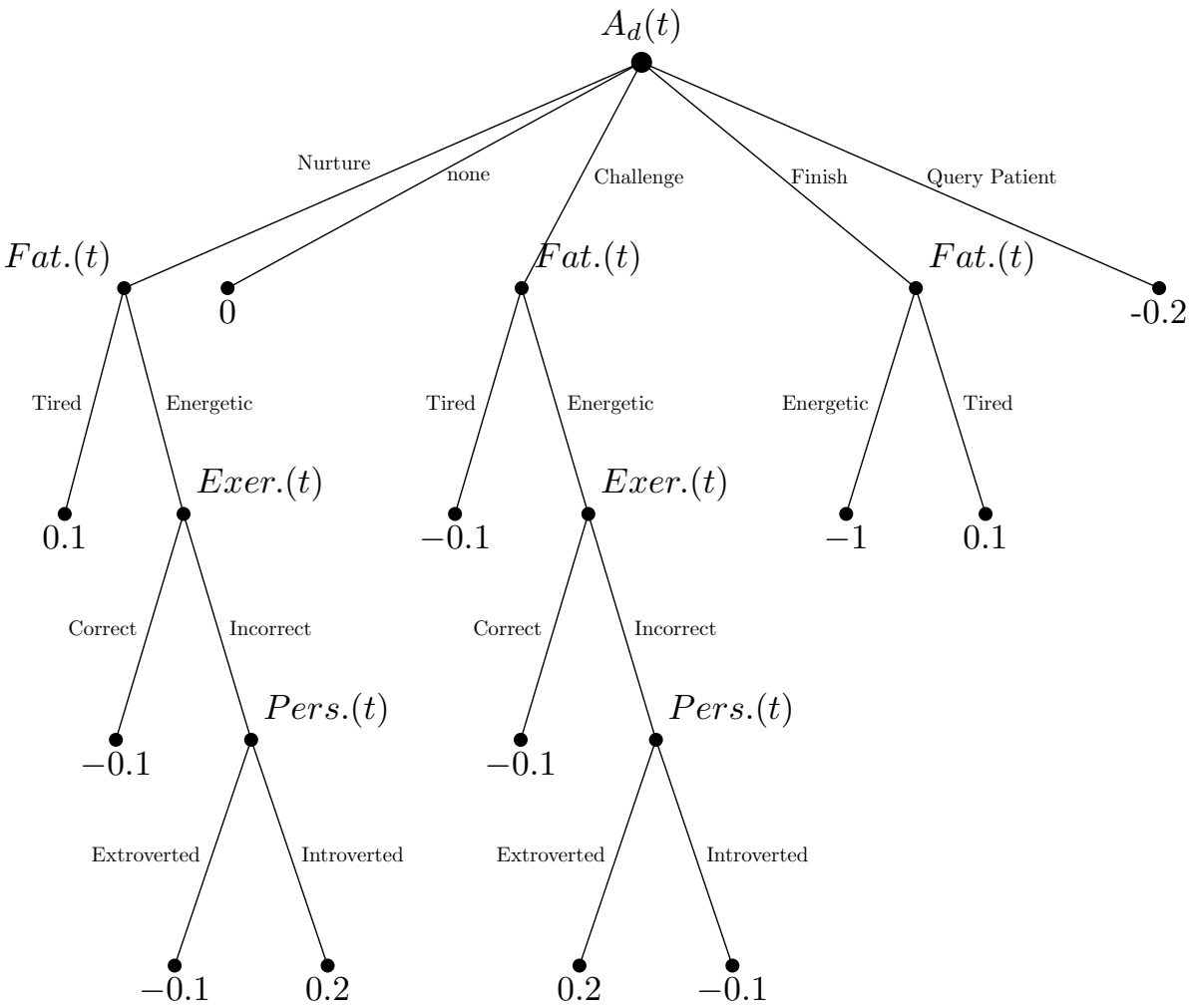



**Figure A.5:** Alebraic Decision Diagram representation of the reward function $R_d$.

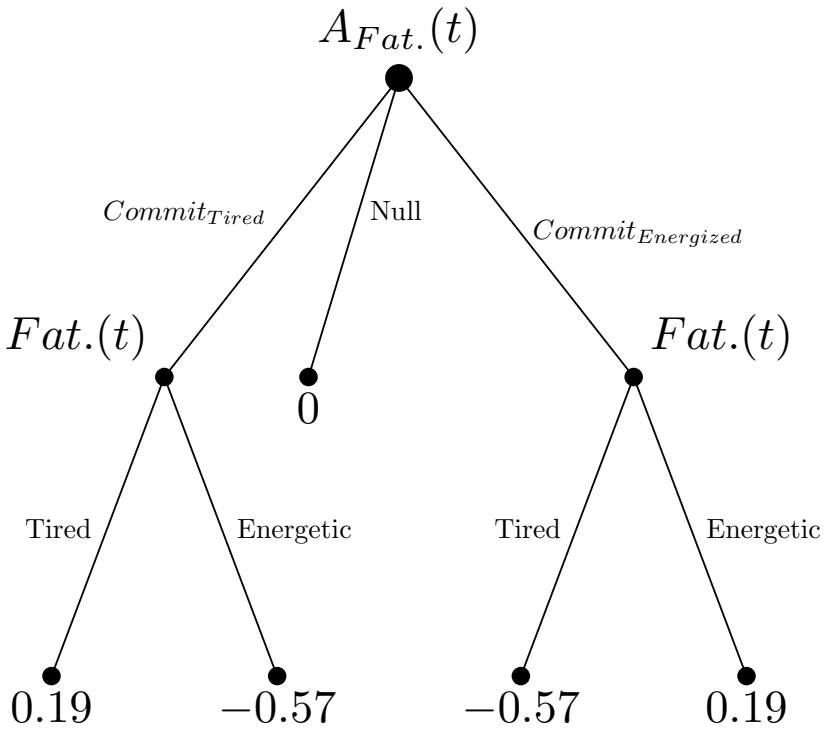



**Figure A.6:** Alebraic Decision Diagram representation of the reward function $R_{Fat.}$.