Heriot-Watt University
University of Edinburgh

LITERATURE REVIEW

Real-Time Vision-Based Learning
for Human-Robot Interaction
in Social Humanoid Robotics

Author: Boris MOCIALOV
Matriculation Number: H00180016, s1581976

Supervisor: Dr. Patricia A. VARGAS
Co-Supervisor: Dr. Micael S. COUCEIRO

Literature review submitted in fulfillment of the requirements for the
robotics research proposal course

April 2016
Abstract

This report proposes a research topic in the field of human-robot interaction. The research will focus on real-time learning paradigm and its application in social robotics. Gestures, as part of more complex actions, will provide information necessary for a social robot to infer emotional state of a human demonstrator. The research may lead to further investigation of acceptance of humanoid robots in social settings. The report gives an overview over existing literature, presenting the evolution of the real-time learning and focusing on a particular subset that deals with vision-based data acquisition in social robotics tailored to research in human-robot interaction. In addition, the report enumerates the details of the project, such as aims, objectives and analyses the requirements, proposing the research plan that will lead to development of an integrated self-organising system.
## Contents

1 Introduction ................................................. 1
   1.1 Outline ............................................. 1

2 Literature Review ........................................... 2
   2.1 Intelligent Robotics .................................... 2
       2.1.1 Evolutionary Robotics .......................... 3
   2.2 Social Robotics ...................................... 3
   2.3 Real-Time Learning Evolution .......................... 4
   2.4 Approaches to Real-Time Learning ...................... 6
   2.5 Data Acquisition in Real-Time Learning .............. 7
   2.6 Real-Time Learning in Social Robotics ............... 8

3 Hypothesis .................................................. 9
   3.1 Justification ......................................... 9

4 Approach .................................................... 10
   4.1 Hardware ............................................. 11
   4.2 Methodologies ....................................... 12
       4.2.1 System Design .................................. 12
       4.2.2 Development .................................... 12
       4.2.3 Communication .................................. 12
       4.2.4 Evaluation ..................................... 13

5 Past Work .................................................. 14
   5.1 Vision-Based Feature Extraction ....................... 15

6 Organisation ............................................... 16
   6.1 Objectives ........................................... 16
   6.2 Requirements Analysis ................................ 17
       6.2.1 Functional Requirements ....................... 18
       6.2.2 Non-Functional Requirements .................... 18
   6.3 Research Plan ....................................... 19
   6.4 Performance Assessment ............................... 22
   6.5 Risk Analysis ....................................... 23

7 Future Applications ....................................... 24

8 Conclusion ............................................... 25

References .................................................. 26
List of Figures

1. NAO H25 Hardware ................................................. 11
2. Overall view of existing gesture recognition system .................... 14
3. ROI Detection and Feature Extraction ............................. 15
4. Project Plan ......................................................... 20

List of Tables

1. Objectives ......................................................... 16
2. Requirements ..................................................... 17
3. Prototypes Assessments .......................................... 22
4. Enumerated Risks ................................................. 23
1 Introduction

Interest in real-time learning capabilities in software arisen after it had been noticed that systems can become more efficient by adapting dynamic behaviour. Efficiency, in the context of real-time learning systems, is improved with adaptation and self-organisation. Both approaches use existing capabilities of a system that are often implemented as simpler behaviours used to adapt to changing environments by developing complex behaviours.

Ability of a system to learn in real time can benefit the user. The user, who is the most knowledgeable about the operation domain, otherwise known as task expert, can instruct a robot to perform certain desired actions or give a high-level description of the task that needs to be accomplished. Similarly, robot experts, developing the hardware and software for a robot, benefit from spending less to no time at all on collection of training data.

For the user to be able to instruct a robot to perform a new task, some form of communication must be established between the two parties. The aim in human–robot interaction is to empower the user and give as much control and freedom as possible when operating the robot. Therefore, the communication mechanism should be simple yet expressive.

Apart from the ability of explicit training of some new task, the robot should be able to track the behaviour of the user and become more stable and efficient in day-to-day operations on the known tasks.

Beyond strengthening the trained or user–taught behaviours, advanced sensing and processing capabilities on board of a robot should support lifelong learning from observations.

1.1 Outline

The report starts off by introducing intelligent robotics and how it fits into social robotics paradigm. It presents an overview over available literature in the field of real-time learning together with the evolution of the topic. Later, particularly interesting and relevant research is selected. After the literature review, the hypothesis is proposed, justifying every factor in the hypothesis
and describing the bigger picture. The hypothesis is supported with a scenario for experiments that may help the reader to understand the intentions of this project. Later, robotic platform is described that will be used for this project and the methodologies that will be employed, depending on the phase of the project (e.g. system design, development, etc.).

Before diving into organisation of the project, past work gives an idea of the current state of the project and what will be the first step. Project organisation presents the aim of the project, gives high-level objectives, analyses requirements, proposes the research plan, proposes evaluation criteria and lists potential risks of the project with possible resolution plans.

At the end a short discussion gives a glimpse into the possible future work and how it can be applied to human-robot interaction.

2 Literature Review

2.1 Intelligent Robotics

Jarvis tells that adaptable intelligent robots should process sensory data when accomplishing useful tasks as opposed to robots that plan and actuate without using sensors as a feedback mechanism. Even though intelligent robotics adds environment into a loop and solves the adaptation problem to dynamic environments by feeling and responding to forces in the environment, it leads to additional challenges posed by applications of intelligent robotics, such as (i) optimised kinematic/inverse kinematic derivations, (ii) advanced localisation methods, (iii) determining efficient navigation paths, (iv) advanced locomotion, (v) HRI (Human-Robot Interaction), and (vi) Emergence of collective intelligence [35].

Two paradigms of intelligent robotics that facilitate emergence of complex behaviour are (i) Behaviour-Based Robotics (BBR) and (ii) Evolutionary Robotics (ER). Both paradigms are bottom-up. BBR combines simpler innate abilities to achieve an emergent complex behaviour [15], while ER achieves emergence with the artificial process of evolution and survival of the fittest [44], [63].
2.1.1 Evolutionary Robotics

ER techniques are extensively employed in designing controllers and/or body morphologies that are used by intelligent robots. ER key component is evolutionary computation that evaluates a range of solutions for every generation and propagates some selected solutions into next generations. The evaluation of a potential solution is done with the fitness function (cost function), which tells how good the current solution is independent of other solutions in the population. In ER, evolved complex behaviour (which consists of simpler behaviours) is evaluated with fitness function, which factors are tailored to evaluate simpler underlying behaviours.

2.2 Social Robotics

Available to date robots are used widely as research platforms. Social robotics is a very broad subject that encapsulates research on robotic platforms among others on learning, adaptation, cognition, developmental psychology, embodiment design, system engineering, societal organisation. On one hand, the aim of the various research on these platforms is to advance the algorithms on planning, navigation, and manipulation. The bigger picture, on the other hand, is obtained by studying the effects of these algorithms, implemented on the embodied agents in real environments, where humans or other robots are present. During the course of research, the robots are given certain roles, while operating in social settings. Robots may assume a role of a partner or an assistant within a bigger picture in social robotics. No matter what the role is, Terrence Fong et al. point out that a social robot has to be adaptable and flexible to spur humans to interaction and engage them to cooperate [22].

Following authors focus on research in learning in social robotics. Bandera Rubio and Juan Pedro define social robot in [49] as a robot that is able to learn from others in a society of other robots, humans, or both. More specifically, Cynthia Breazeal reduces social robotics to robots interacting with humans through the process of social exchange in [11]. One such approach can be studied through imitation games [12] during which a robot imitates behaviour
of a human by which it explores its own motor abilities. Such bottom-up way of learning can be used to study how humans interact with robots and how robots explore their abilities through imitation and at the same time use new abilities to communicate with humans.

### 2.3 Real-Time Learning Evolution

There is no particular point in time when the real-time learning had become popular tool for social robots’ development. The idea evolved from the desire to generalise programs, written for robotic arms for the tasks involving object grasping, trajectory or motion planning. Initially, Bruno Dufay presents sequential or inductive learning approach for building programs that decide in real time which motions to execute for a certain task and generate a program to solve that particular task [20].

Next, programming by demonstration (PbD) approach is applied to real-time learning problems. Holger Friedrich and Rüdiger Dillmann claim that the problem with PbD at that time was that, although real time learning could be achieved by providing demonstration, many demonstrations were required to cover classes of tasks. Therefore, authors propose a method for generalisation of a single demonstration with the help of user intentions to cover one task class using STRIPS [21] programming language [25].

Nearly at the same time, Christopher Lee and Yangsheng Xu make a system that interacts with users using gestures and updates its knowledge about gestures [39]. Aude Billard develops a real-time learning system with loose sensory requirements and applies it on real robots to demonstrate communication grounding [7–9]. On PbD side, Paul Rybski and Richard Voyles give a mobile robot a set of basic capabilities and let it determine which of these capabilities are needed to replicate a demonstrated task [51]. Soshi Iba et al. adapts WYSIWYG (what you see is what you get) pattern when performing PbD, which empowers the user to coach the robot, by continuously being able to view the training results and, thus, being engaged in the process. John Demiris develops a system that allows a robot to match perceived movements with equivalent movements of its own, claiming that imitation can be used as a
mechanism for bootstrapping further learning [19] with which Cynthia Breazeal later agrees, introducing imitative games as a mechanism for bootstrapping social behaviour in robots [13]. Stefan Schaal concludes this stream of research on real-time learning, achieved with imitation learning, on a pessimistic note, arguing that none of the existing to date approaches satisfy challenges raised with the imitation learning paradigm, such as appropriate feature extraction for perceptual data encoding, actuation primitives representation, and learning mapping between sensory input and actuation [52].

Later, Luc Steels develops action/language games used for self-organised communication grounding [57–60]. The idea of action/language/imitation games is not novel in respect to communication grounding and had been investigated and defined formally by Chrystopher Nehaniv in [43] for evolution of sensory-motor loop, which is similar to the communication grounding as both agents establish the common meaning of needs or intentions.

More recent studies of real-time learning haven’t gone far from the initial research in PbD. In the application domain of the PbD, the tasks had become more complex, whereas the techniques remain the same [17], [49], [30], [66], [18]. Hidden Markov Models (HMM) together with Baum-Welch algorithm for parameter estimation [64] or raw sampling (e.g. position, speed, orientation) together with Dynamic Time Warping techniques (DTW) [6] for sampling comparison are used to represent and compare dynamic systems (e.g. gestures and actions). Recently Radial Basis Function (RBF) [16] have gained attention modelling such systems. Many cognitive architectures are being built on top of artificial neural networks (ANN) that use either recurrent model (for short-term memory) or feedforward model for sensorimotor processing.

From the reviewed evolution of research on real-time learning, the trend tilts towards establishment of frameworks depending on whether learning is achieved sequentially, via imitation, or demonstration. Distinct frameworks tend to be biologically inspired as in the case of the neural networks that resemble the human brain and represent a cognitive model of a robot or imitation learning that relates to mimicking during developmental stages in developmental psychology.
2.4 Approaches to Real-Time Learning

Many approaches to real-time learning can be distinguished throughout the identified literature. Because the field is not well-defined, the approaches are often ambiguous and contradictory in various publications. Following approaches were selected for real-time learning after careful literature walkthrough:

- Reinforcement Learning (RL)
- Artificial Neural Networks with Backpropagation (ANN BP)
- Genetic Algorithms (GA)
- Programming by Demonstration (PbD) through:
  - Imitation
  - Walk-Through/Lead-Through
  - Learning from Observation
- Sequential Learning
- Iterative Learning

RL [61] seeks for optimal policy based on experience. ANN BP trains neural network to adjust weights to obtain desired output given some input [50]. GA applies genetic operators (mutation and crossover) to a population of data structures over a number of generations, where every data structure encodes a possible solution. The algorithm compares the desired output to the actual output (fitness evaluation) to guide the evolution [27]. Both RL and GA operate in hypothetical landscape of possible solutions, trying to converge to a global optimal solution. The landscape is infinite and, therefore, there are infinite number of possible solutions. Algorithms use sophisticated approaches to ensure fast traversal of the search space. Unfortunately, finding the global optimal solution can take very long time and, in fact, not always achievable, which may be caused by premature convergence to local optima. To cope with this challenge, a set of heuristics and tricks exist that simplifies the traversal of the search spaces [62].

PbD paradigm reduces the search space of all possible solutions by guiding the traversal to the right directions on the landscape. In addition, PbD allows users to train robots to perform desired tasks, which makes this approach far more robust than RL and GA [29]. PbD can be accomplished by imitation,
where a robot roughly repeats a motion/action/gesture/etc. demonstrated to it [52]. Another way is to either walk the robot through a particular task, physically moving the hardware or lead the robot through the task by instructing it a set of commands and intermediate points in space that it has to go through to accomplish that task. The advantage of walking the robot through the task is the chance of transferring the lifelong skill of the demonstrator onto a robotic platform.

Although both sequential learning and iterative learning may sound similar, they target real-time learning differently. Sequential learning (consecutive learning, incremental learning) evolves current knowledge at each step by consecutively seeing new training examples. Iterative learning exploits repetitiveness of batch processing training examples and uses knowledge obtained from previous batches to revise knowledge of newly arriving batches. Therefore, iterative learning operates on the training data itself, while sequential learning modifies learned model [2], [53], [68].

Originally, selected approaches were mutually exclusive, although there is research that combines RL with ANN BP [3], RL with GA [42], PbD with GA [36].

2.5 Data Acquisition in Real-Time Learning

Soshi Iba et al. distinguish four data acquisition methods in [32] used in robot programming by demonstration, which are (i) vision–based, (ii) range sensing, (iii) external wearable devices, and (iv) tactile sensing. Vision–based sensing, compared to other, has the advantage of obtaining a lot of data from the environment. On the downside, to be able to build general frameworks, a standard has to be in place that would tell what data is important and what can be ignored. Unfortunately, the standard is unrealistic because one set of data may be crucial in one application and insignificant in another application.

In listed available approaches to real-time learning, the data/demonstrations have to be recorded using some sensor. Different approaches employed mean different sensory preferences. The following methods are available for obtaining the information:
• Teleoperation
• External devices-based motion tracking
• Vision-based motion tracking
• Kinaesthetic
• Shadowing

In teleoperation, the robot is recording operator’s input, while the operator is being relayed the haptic feedback [31]. Motion tracking external devices (e.g. data gloves, motion capture suit, etc.) provide less noisy data, but are obtrusive to the user. Kinaesthetic tracking had been mentioned previously as the walk-through procedure. During kinaesthetic user’s input, the robot records changing parameters of own structure (e.g. orientation, position, joint angles, etc.). During shadowing, the robot records changing parameters, while mimicking the demonstrator. Many authors, who research real-time learning by imitation, observation, or demonstration with humanoid robots, like in [5], [9], [13], [18], [19], [51], [57–60], [66], focus on vision-based data acquisition method with an exception in [30], where Kinect is used additionally. The vision-based data can convey more information than other listed methods. In addition, vision-based sensing does not impede user’s operation and requires relatively cheap hardware.

Independent of what method is used for obtaining the information from the environment, the information must be condensed and encoded in a convenient for the task form that would strip the data from all the present noise. Real time domain implies that the acquisition and encoding of the incoming data must be done in real time since delays are obviously not desirable in real-time systems as they lead to confusion and frustration.

2.6 Real-Time Learning in Social Robotics

This literature review focuses on previous research in social robotics that uses vision-based sensing and looks at how it can be extended to improve human-robot interaction with the help of real-time learning.

Early research had been driven by the desire to make a system that would be able to learn in real time. Resulted frameworks with different designs, reported
in [9], [19], and [51], showed that such systems are feasible both in simulations and on real robots. Ambitions for the future work ranged from extension of developed systems to accommodate object-related manipulations to application in robot-robot interactions.

Later, research had been more directed towards social robotics with the aim to build robots that would show social behaviour that would manifest in such phenomenon as shared attention and turn taking that are very common among humans. Most of the backbone algorithms are reused from previous research on cognitive architectures, while additional modules are added to facilitate social behaviour in robots [13], [57].

The latest research attempts an implementation of recently discovered spiking neuron system using available processing algorithms [18]. In addition, application becomes more complex, which requires development of tailored system [66].

Selected publications indicate that the research in real-time learning is headed towards complex novel applications of the technique. Unfortunately, there are cases, where available cognitive systems are either not suitable or not desirable for new application areas, in which case new architectures are being developed.

3 Hypothesis

Combining evolutionary robotics techniques with existing approaches to real-time learning is expected to improve HRI experience in social humanoid robotics as opposed to commonly used bare PbD or iterative learning approaches.

3.1 Justification

ER techniques are good for optimisation tasks. Finding an optimal sensori-motor mapping can be viewed as an optimisation problem. This technique will be used in this project due to the personal interest in using genetic algorithms in various applications. Apart from the mere interest, it is necessary to explore
ER generalisation capabilities in complex real-time systems. Additionally, the technique will be tested on ability to evolve a controller in real time.

Ability to learn in real time as well as the advantages of having a robust and adaptable controllers had been discussed earlier. Being able to perform real-time learning requires excessive resources. Robots will be given the ability to learn in real time to discover whether it is feasible, despite presented earlier cases from literature review, to have a system on board that would be able to update itself in real-time sufficiently.

Based on identified previous work, it is desirable among human users to have robots that exhibit social behaviour. Such study may, in turn, give a better understanding of the uncanny valley phenomenon in robotics. More obvious justification for adapting social behaviour in social robots is to make user more engaged when interacting with robot.

Social robot would have to analyse user’s behaviour and adapt its programmed responses to boost its likeability in social setting.

4 Approach

Programming by demonstration approach to real-time learning will be chosen for the following scenario using two NAO robots (demonstrator and learner).

Demonstrator will communicate its emotional state to the learner. Two emotional states are considered for the scenario: anger and happiness. The demonstrator will be given a set of primitives (actions) to execute based on the current emotional state. The primitives for the angry state will consist of the demonstrator raising hand up and approaching the learner (trying to hit) followed by shining red eyes. The primitives for the happy state will consist of the demonstrator spreading arms and approaching the learner (trying to hug) followed by green eyes colour. Learner is expected to ground understanding of the two emotional states, namely, anger and happiness and react accordingly before it sees the light. Therefore, the learner is expected to associate gestures with lights, where lights would have an innate meaning for the learner. Learner’s reaction primitives will include evasion by moving to the side (avoiding being hit
by angry demonstrator) and spreading arms (hugging back the demonstrator).

Neural network will be used as a cognitive architecture of the learner that will be evolved in real time using genetic algorithm.

4.1 Hardware

![NAO H25 Hardware Diagram](image)

**Figure 1:** NAO H25 Hardware [1]

NAO humanoid robots will be used for this project, built by Aldebaran-Robotics. Range of sensors is present on board of a NAO robot: 1. eight force-
sensing resistors on soles of robot’s feet, 2. two axis gyrometer in the centre of the body, 3. one three axis accelerometer, 4. two sonar emitters and two receivers, 5. tactile sensor on the head, three on each hand and one on tip of each foot, and 6. two video cameras in the head. The robot features twenty-five degrees of freedom allowing it to successfully perform a wide range of complex movements and object manipulations [28].

NAO robots are perfectly suitable for the social robotics experiments, because, as it is pointed out by Fong et al. in [23], a robot must have an embodiment that would allow it to interact with the environment in the same way living creatures do (humans in the case of social robotics) and see things humans find salient in the environment.

4.2 Methodologies

A set of methods will be used throughout the project. Choice of methodologies is split into system design, development, communication, and evaluation categories.

4.2.1 System Design

System design will fall back on biologically inspired approaches, such as ER, using evolutionary approaches. The evolution will evolve a neural network controller that will serve as a cognitive model on board of a robot.

4.2.2 Development

Prototypes will serve as a proof of concept for sub-components of the scenario described previously. With every prototype a set of posed requirements will be satisfied and this will guide the development process.

4.2.3 Communication

Described scenario proposes to use gestures as a communication medium between the robots.
Indirect communication is proposed as opposed to other alternative ways of communication, such as, for example, direct communication. Indirect communication has advantages over direct communication with which information is unambiguously passed between the robots. Gestures may encapsulate a lot of information or may disambiguate communicated information by, for example, pointing to a specific object using very simple gesture.

Sign language is a form of indirect communication and it allows constructing gestures with infinite possibilities, where gestures are combined together to create new meanings.

4.2.4 Evaluation

HCI techniques will be used for the evaluation of functional and non-functional requirements. This will include empirical testing of prototypes and gathering of quantitative and qualitative data.

Quantitative data will tell how many times the robot succeeded in recognising gestures and correctly associating colour with the perceived gesture. Overall, the quantitative data will allow to measure the quality of the evolved controller and how good the controller is at remembering the sensorimotor mapping.

Qualitative data will give meaning to quantitative data. In particular, having collected the observed measurements of how reliable and satisfactory the controller is, qualitative data will correct these results by looking at the behaviour of robots.

Ultimately, having collected quantitative and qualitative data, it will be put through statistical analysis to measure how sufficient the developed system is.
5 Past Work

Proposed project will utilise parts of previously implemented unpublished gesture recognition system [41]. Mentioned system consists of multiple layers, where every layer performs some input data (visually perceived gesture) manipulation or transformation and outputs probability of an input for every learned gesture.

![Figure 2: Overall view of existing gesture recognition system](image)

- a) extracted features from sequence of frames in a video stream
- b) String of Feature Graphs (SFG) [26] for the video stream
- c) affinity matrix for the SFG
- d) HyperNEAT [54] evolved feature detectors (artificial neural networks)
- e) classifier artificial neural network
- f) resulting classification of the video stream in a)

The existing system is not currently tested fully and, therefore, cannot be used directly as a part of the real-time vision-based learning system. For the proposed project, only part a) from the Figure 2 will be used, which is responsible for vision-based feature extraction.

The system is designed to detect and recognise gestures of a human, therefore, the system searches for human body parts (head and torso) by using Haar feature-based cascade classifier [46]. Once the head and the upper body are detected, the system performs background subtraction and searches for moving objects that originate from the upper body, such as limbs (e.g. arms).

Existing system will not be described fully as only a small fraction of it will be used in the proposed project.
5.1 Vision-Based Feature Extraction

Feature extraction begins with region of interest (ROI) detection, which detects where action in video is happening by performing background subtraction, followed by illumination reduction and edge enhancement with the help of erosion and dilation. Features are extracted from detected moving objects that originate from the upper body of identified human in a video. Features are represented by local kinematic features that contain the most relevant information needed for gesture representation. In theory, capturing kinematic features is less noisy than capturing more complex features, such as blobs (as in [10]) or salient spots (as in [14]) that contain collection of features, as the amount of noise is proportionate to the amount of data collected [40]. Potential limbs are analysed by looking at hull and convex defects to find break points (e.g. elbows) and smaller details (e.g. fingers).

![Figure 3: ROI Detection and Feature Extraction](image)

a) Torso b) Torso and limbs c) Torso, limbs, and limb details detection

All the implementation is done in C++ programming language with the help of OpenCV\(^1\) image processing library.

---

\(^1\)OpenCV library [http://opencv.org/](http://opencv.org/)
6 Organisation

As discussed previously, the aim of this project is to prove that combining ER with PbD can make real-time learning more efficient. The next subsections will describe in details what is needed to achieve this goal.

6.1 Objectives

<table>
<thead>
<tr>
<th>#</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Primitive programming</td>
</tr>
<tr>
<td>2</td>
<td>Gesture recognition algorithm</td>
</tr>
<tr>
<td>3</td>
<td>EA evolved cognitive architecture</td>
</tr>
<tr>
<td>4</td>
<td>Integrated System</td>
</tr>
</tbody>
</table>

Table 1: Objectives

Gestures are a part of the scenario, given in Section 4. Robots have no pre-programmed innate abilities to perform complex gestures as hugging or attempting to hit an opponent. Therefore, these primitives must be programmed in advanced before proceeding onto development of gesture recognition algorithm.

Having primitives, gestures will be encoded and a model will be trained to recognise different gestures based on perceived visual information.

Cognitive architecture will be evolved and updated in real time on board of a robot.

All developed and tested components will be integrated in a single system and executed on board of NAO.
### 6.2 Requirements Analysis

<table>
<thead>
<tr>
<th>#</th>
<th>Requirement</th>
<th>Objective</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Functional Requirements</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Primitive execution</td>
<td>1</td>
<td>H</td>
</tr>
<tr>
<td>2</td>
<td>Extract features</td>
<td>2</td>
<td>H</td>
</tr>
<tr>
<td>3</td>
<td>Encode features</td>
<td>2</td>
<td>H</td>
</tr>
<tr>
<td>4</td>
<td>Model training</td>
<td>3</td>
<td>H</td>
</tr>
<tr>
<td>5</td>
<td>Model testing</td>
<td>3</td>
<td>H</td>
</tr>
<tr>
<td>6</td>
<td>Correct classifications</td>
<td>3</td>
<td>H</td>
</tr>
<tr>
<td>7</td>
<td>Re-trainable model</td>
<td>3</td>
<td>H</td>
</tr>
<tr>
<td>8</td>
<td>EA algorithm for ANN</td>
<td>3</td>
<td>H</td>
</tr>
<tr>
<td>9</td>
<td>Processing ANN input</td>
<td>3</td>
<td>H</td>
</tr>
<tr>
<td>10</td>
<td>EA trigger</td>
<td>3</td>
<td>H</td>
</tr>
<tr>
<td>11</td>
<td>Single system</td>
<td>4</td>
<td>M</td>
</tr>
<tr>
<td>12</td>
<td>System executable on NAO</td>
<td>4</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td><strong>Non-Functional Requirements</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Clear features</td>
<td>2</td>
<td>H</td>
</tr>
<tr>
<td>14</td>
<td>Consistent features</td>
<td>2</td>
<td>H</td>
</tr>
<tr>
<td>15</td>
<td>Simple encoding</td>
<td>2</td>
<td>M</td>
</tr>
<tr>
<td>16</td>
<td>Expressive encoding</td>
<td>2</td>
<td>M</td>
</tr>
<tr>
<td>17</td>
<td>Fast model training</td>
<td>2</td>
<td>M</td>
</tr>
<tr>
<td>18</td>
<td>Fast recognition</td>
<td>2</td>
<td>H</td>
</tr>
<tr>
<td>19</td>
<td>Fast evolution</td>
<td>3</td>
<td>M</td>
</tr>
<tr>
<td>20</td>
<td>No execution delays</td>
<td>4</td>
<td>L</td>
</tr>
</tbody>
</table>

**Table 2:** Requirements  
H — High, M — Medium, L — Low
Compiled requirements are split into two groups with functional requirements in one group and non-functional requirements in another. Functional requirements usually describe the overall system in terms of its functions and are usually prioritised higher. Non-functional requirements describe system from the perspective of usability and comfort. Although non-functional requirements that deal with delays and stability are important, these are usually prioritised lower when the system is in its conceptual phase.

Identified requirements are listed together with the objective that they belong to and the priority (high, medium, low).

6.2.1 Functional Requirements

Robot shall be able to execute a certain primitive on demand. Primitive are innate to robots and are encoded directly. Robot shall be able to extract features from the vision data. The system shall be able to encode extracted features in a suitable way. The system should be able to train as well as to test a model using encoded features. Trained model should be able to correctly classify similar gestures. A model should be dynamic enough to be retrained using different gestures without losing information about previously learned gestures. EA algorithm should be capable of evolving an ANN, which will be used for classification purposes. EA should be triggered at any time new information becomes available to evolve a better ANN classifier. All the components of the system should be integrated into a single self-organising system and uploaded on the robot.

6.2.2 Non-Functional Requirements

Extracted features should be clear enough and not include any unnecessary noise that could potentially require more processing power, which is to be minimised. Certain features should be chosen by the designer and not changed at any time of the execution. The encoding of the features should be simple yet expressive to reduce the amount of processing power needed and still contain lots of information. The training of the model, the classification/recognition,
the evolution, and the execution should be fast, so that the system can function in real time.

6.3 Research Plan

Research will consist of (i) primitives programming (ii) developing robot gesture recognition algorithm based on data from visual sensing, (iii) development of cognitive architecture that will be able to classify gestures based on the extracted data from each gesture in real time, (iv) extensive experimentation and data gathering for evaluation, (v) evaluation, and (vi) reporting.
Figure 4: Project Plan
The project starts with the first prototype that consists of programming a set of primitives (basic innate abilities) for robots that will be hard-coded and uploaded onto robots. Demonstrator is taught how to perform such actions as hitting and hugging, while the learner is taught how to evade by moving to a side and how to perform a hug.

The output of the second prototype is a gesture recognition algorithm. Gesture recognition will consist of the feature extraction using vision data, encoding of features as hidden Markov chains, training the hidden Markov model (HMM) with Baum-Welch algorithm for parameter estimation, and testing the model using test data. Prior to training/testing, the data has to be collected. This will be done in parallel with the gesture recognition algorithm development when the feature extraction will become available. The algorithm will be tested directly on NAO robots in order to get as realistic data as possible.

For the third part of the system, an artificial neural network will be developed using evolutionary algorithm, developed specifically for this application. No freely available open source implementation of algorithms (e.g. NEAT [56], HyperNEAT [55] or their variations) will be used for this task. Basic evolutionary algorithm will be developed to test basic operators as opposed to complex evolutionary pathways that are present in, for example, HyperNEAT. After the development of the EA, an update mechanism will be put in place that will trigger evolution as the new data will become available, leading to a real-time adaptation of the algorithm.

After all sub-components will be developed, they will be merged into a single self-organising system on board of a learner robot. Thereafter, the system is uploaded on to the robot, a set of test-cases from scenario, described in Section 4, will be executed and the data will be collected for analysis.

All the activities will be documented in the report, that will be submitted at the end of the project. The documentation will start after the prototype 2 will be available, describing the technical part of the implementation. The report will be updated alongside the project activities.
6.4 Performance Assessment

<table>
<thead>
<tr>
<th>#</th>
<th>Prototype</th>
<th>Date</th>
<th>Requirements</th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Primitives programming</td>
<td>20th of May</td>
<td>1</td>
<td>Robots can perform every gesture on demand</td>
</tr>
<tr>
<td>2</td>
<td>Gesture recognition algorithm</td>
<td>17th of June</td>
<td>2-3, 13-18</td>
<td>High accuracy on collected testing data set is high</td>
</tr>
<tr>
<td>3</td>
<td>EA evolved cognitive architecture</td>
<td>22nd of July</td>
<td>4-10, 19</td>
<td>Demonstrate correct gesture classification</td>
</tr>
<tr>
<td>4</td>
<td>Integrated system</td>
<td>5th of August</td>
<td>11-12, 20</td>
<td>Demonstrate real time evolution of cognitive architecture</td>
</tr>
</tbody>
</table>

Table 3: Prototypes Assessments

Primitives programming will be assessed by demonstrating the selected primitives (attacking to hit, hugging, and evading) are executed correctly every time they are requested.

Gesture recognition algorithm will be assessed by showing that certain accuracy over some chosen threshold is achieved, signifying correctness of the algorithm.

Evolved cognitive architecture will be assessed by showing that the data, outputted by the gesture recognition algorithm is fed into evolved neural network that performs correct classification of the perceived gesture. Confusion matrix will be constructed to specify the accuracy of the classification on a set of trials.

Finally, integrated system will be assessed by demonstrating the evolving cognitive architecture on NAO robot in real time that will be capable of associating gestures with emotions of the demonstrator robot.
## 6.5 Risk Analysis

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>L</th>
<th>I</th>
<th>Resolution Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Robot breaks or does not work</td>
<td>L</td>
<td>H</td>
<td>Request a new robot meanwhile continue working in Webots simulator</td>
</tr>
<tr>
<td>2</td>
<td>Robot is not capable of extracting consistent features from visual cues</td>
<td>L</td>
<td>H</td>
<td>Choose other features or add an external camera/kinect</td>
</tr>
<tr>
<td>3</td>
<td>Classification is consistently wrong</td>
<td>M</td>
<td>H</td>
<td>Test the implementation for errors or choose other classification method</td>
</tr>
<tr>
<td>4</td>
<td>Training is too slow</td>
<td>H</td>
<td>M</td>
<td>Simplify the model or use additional resources</td>
</tr>
<tr>
<td>5</td>
<td>EA is too slow on the robot</td>
<td>H</td>
<td>M</td>
<td>Simplify the algorithm, use additional resources, or look for alternative solutions</td>
</tr>
<tr>
<td>6</td>
<td>Integration of sub-systems is impossible</td>
<td>L</td>
<td>M</td>
<td>Look for workarounds that would allow pseudo-integration</td>
</tr>
<tr>
<td>7</td>
<td>System is not executable on NAO</td>
<td>L</td>
<td>H</td>
<td>Fallback onto a PC solution, substitute components that are causing this behaviour</td>
</tr>
<tr>
<td>8</td>
<td>Lack of time to implement everything</td>
<td>M</td>
<td>M</td>
<td>Reduce the scenario and use existing solutions for sub-systems</td>
</tr>
</tbody>
</table>

| Table 4: Enumerated Risks |
| L — Likelihood, I — Impact |

Robot-related risks have low likelihood as the robots are relatively stable. In case of a possible issue, the issue will be reported and alternative hardware/robots will be used. If no other robots or hardware is available, simulators, such as Webots\(^2\) will be used as a fallback solution.

Risks with the functional requirements have medium likelihoods due to the difficulty of predicting the amount of work needed for a particular feature. The impact is medium because such issues can be resolved relatively fast with the help of the peers in the community or with extra timing, diverted from other project activities.

\(^2\)Webots development environment https://www.cyberbotics.com/overview
Qualitative risks are marked with high probability. Slow execution of the system is expected due to the low amount of resources present on NAO robots. The impact is marked from medium to high, because some of the requirements may not be met due to the limitation in processing power. Developed algorithms may be simplified in order to achieve the fastest possible processing on board of robots.

7 Future Applications

Software–based real-time learning is becoming as widely applied as adaptive control methods in intelligent systems [4]. Algorithms that employ real-time learning are used, among other related areas, in human–robot interaction (e.g. facial expression, speech, gesture recognition [23]), autonomous robots operations (e.g. flexible manufacturing, space exploration [24]), object tracking [37], robot navigation [34], expert systems [48], etc.

Successful completion and findings for the proposed in this report project will be used in further human–robot interaction studies with the focus on sign language processing and recognition. The aim of the further study will be to look into how real-time machine learning methods may improve efficiency of human–robot interaction using sign language. Some of thus far reviewed literature (e.g. [33], [38], [45], [47], [65], [67]) indicates that research in sign language recognition is not robust enough to suit complexity of a given sign language. Since real-time learning allows system to cope with change and adapt as well as self-organise given unseen inputs, it is believed that sign language recognition systems may become more robust by using real-time learning methods to learn new signs as well as user patterns in real time.
8 Conclusion

Report showed the evolution of real-time learning and techniques used. It focused on a subset of related work that studies the implications of the real-time learning in social robotics. The Report presented hypothesis, aims, objectives, and methods of the project, justifying the choices made based either on personal preferences, state-of-the-art solutions or mere interest in certain approaches. Initial work had been shown and a plan had been made for the future work.

The research is expected to provide some insides into how a social robot may understand user’s emotional state and react in a such a way as to increase its likeability and thus affect the emotional state of the user. This may lead to further study of acceptance of robots and such phenomenon as uncanny valley in human-robot interaction.
References


