FP6-IST-002020

COGNIRON

The Cognitive Robot Companion

Integrated Project

Information Society Technologies Priority

D4.2.1

Report on Evaluation of Experiments on Correspondence Problem in Robot Learning on How to Imitate Humans

Due date of deliverable: 31/12/2004
Actual submission date: 31/01/2005

Start date of project: January 1st, 2004

Duration: 48 months

Organisation name of lead contractor for this deliverable:
University of Hertfordshire

Revision: final
Dissemination Level: PU

Executive Summary

This report describes the initial experiments carried out with the JABBERWOCKY system developed as part of the University of Hertfordshire's first phase of the program of research for the COGNIRON project. The JABBERWOCKY system generates appropriate action commands that can be used by a robotic companion to imitate the tasks demonstrated by a human user, solving the *correspondence problem* of mapping observed behaviour to the robot's own embodiment.
The report is complemented by a demonstration video of the JABBERWOCKY system using captured data from a human demonstration to generate corresponding actions targeted to two different (dissimilarly embodied) imitator platforms (in simulation), that imitate by matching the arrangement of objects on a workspace, starting from dissimilar initial positions and according to combinations of different *effect metrics* (see Appendix A).
The experimental results provide initial validation for the JABBERWOCKY system, and illustrate the qualitatively different imitative behaviours that result from using different metrics to match the *effect* aspects of the demonstration of object arrangement.
The work was carried out to satisfy the requirements of Research Area (RA) 4 as outlined in Work Package (WP) 4.2 to meet the 12 month target; Deliverable 4.2.1. Details of the RAs, WPs, Key-Experiments (KEs), COGNIRON Functions (CFs) and Deliverables can be found in the respective COGNIRON project documents.

Role of Evaluation of Experiments on Correspondence Problem in Robot Learning on How to Imitate Humans in COGNIRON

The work presented in this report for WP4.2 (*"Correspondence mapping across dissimilar bodies (How to imitate)"*) helps to provide rigorous scientific foundations for COGNIRON functions CF-RG:*"Learning to reproduce gestures"* and CF-LIF:*"Learning important features of a task"*, and develops in synergy with workpackage WP4.1 (*"Sub-goals extraction and metrics of imitation performance"*) by (1) providing a broad theoretical and practical systematic scientific framework from which to select what aspects of behaviour to imitate (*goal and sub-goal metrics for states, actions and effects*) going beyond the current state-of-the-art, and (2) by implementing a multi-targetable architecture which allows the specific features extracted from human demonstrations to be imitated on different platforms.

Relation to the Key-Experiments

The work presented in this report is relevant to KE3 script: *Learning Skills "Arranging and interacting with objects"*. The script stresses the robot's ability to learn from implicit (imitation learning) and explicit (verbal interaction) teaching. As the embodiment of the human user and the robotic companion will be dissimilar, a *correspondence problem* must be solved, defined by the imitator's embodiment, the sub-goal granularity, and the metrics used to match aspects of behaviour. Extensions of the JABBERWOCKY system presented this report could be used to generate action commands that a robotic companion could use to imitate the tasks demonstrated by a human user. Since the robotic platform to be used in the COGNIRON KE3 is not yet fixed, and also for maximum applicability, WP4.2 has adopted a generic approach to the correspondence problem, targeting multiple different potential platforms. The corresponding solutions produced by the system can be targeted to multiple robotic platforms and adapt to arbitrary starting conditions.

Contents

1 Introduction

For a robot to learn tasks and skills from a naive human, e.g. in a home inhabited by people who neither can nor want to learn programming languages, inverse kinematics, nor artificial intelligence techniques, the robot's capabilities must implement general imitation and social learning. A robotic companion learning tasks and skills by demonstration from the human user is a powerful paradigm for transferring knowledge within a social context. But, as the robot and the human will not have the same embodiment and not the same access to object affordances, the robot must have the capability to discover appropriate mappings of its own *actions* so as to reproduce *states* and *effects* that match the task demonstrated by the human. This is called the *correspondence problem* in imitation and must be solved by the robot in order to successfully imitate.
Different aspects of matched behaviour are required for different types of social learning/robot programming by demonstration. We review the relevant theoretical concepts with emphasis on different metrics and classes of correspondence problems defined by the sub-goal granularity and the choice of which aspects of behaviour are significant and should be matched.
Towards solving the correspondence problem and allowing a robotic companion to match the behaviour of the human user, the JABBERWOCKY system is developed. The system uses captured data from a human demonstrator to generate appropriate action commands that can be targeted for various software and hardware platforms. These actions should allow the imitating agent to achieve corresponding actions, states and effects, depending on

- the metrics used to evaluate the similarity between the observed demonstration and the imitative behaviour,
- the sub-goal granularity,
- the embodiment restrictions and constraints (as imposed by the targeted imitator platform) and
- the possibly different initial state of the objects in the environment.

The system is intended to be able to generalize across variations in all of the above items.
The resulting character of the imitative behaviour will greatly depend on the metrics used to match aspects of the human demonstrated behaviour. In the initial 12 months we have concentrated on *effect* metrics, since they are essential for imitation of object manipulation and arrangement (a capability assumed important for a cognitive robotic companion) and also since they can be employed by a variety of imitators with very dissimilar embodiments. We defined a set of important effect metrics for use in the initial experiments with the JABBERWOCKY system. These are demonstrated and evaluated for two target imitator platforms in experiments reported here.
Further directions, including extensions to using *state* and *action* metrics, different target imitator platforms, the characterization of the *spaces of metrics*, and the relation to goal extraction are also discussed.

2 Overview

The remainder of this report is organized as follows:
Sections 3 and 4 give theoretical background on the *agent-based perspective* in imitation and the *correspondence problem* of how to map actions modelled by a human to those of an autonomous imitator.
Section 5 gives further background by presenting an overview of the ALICE (**A**ction **L**earning via **I**mitating **C**orresponding **E**mbodiments) generic framework for solving the correspondence problem. The ALICE framework provides a functional architecture that informs the design of robotic systems that can learn socially from a human demonstrator.
Section 6 gives details on the relevance of the work presented in this report to the COGNIRON functions and key-experiments.
One system informed by the ALICE framework is JABBERWOCKY, developed for the COGNIRON project, and presented in section 7. The JABBERWOCKY system uses captured data from a human demonstrator to generate appropriate action commands that can be targeted for various software and hardware platforms.
Section 8 defines a selection of metrics that are used to match the *effects* aspect of the captured demonstrated tasks by the JABBERWOCKY system in the experiments presented in this report.
The experimental setup is given in section 9, while the experimental results are presented and discussed in section 10.
Finally, section 11 gives a summary, conclusions and discusses future work.

3 The Agent Based Perspective on Imitation

In order to identify the most interesting and significant problems on robot-human imitation, an agent-based perspective must be used [Dautenhahn and Nehaniv, 2002]. In this perspective, imitation is best considered as the behaviour of an autonomous agent in relation to its environment, including other autonomous agents. The mechanisms underlying imitation are not separated from the behaviour-in-context, including the social and non-social environments, motivations, relationships among the agents, their embodiments, the agent's individual and learning history, etc. Such a perspective helps unfold the full potential of research on imitation and helps in identifying challenging and important research issues. The agent-based perspective has a broader view and includes five central questions that the imitating agents must address: *who* to imitate, *when* to imitate, *what* to imitate, *how to* imitate and how to *evaluate* a successful imitation. A systematic investigation of these research questions can show the full potential of imitation from an agent-based perspective [Dautenhahn and Nehaniv, 2002].

3.1 The "Big Five" Questions

An autonomous agent, in order to be able to imitate must address the following five questions.

***Who* to imitate?** It is important that the imitating agent chooses its model[1] in such a way that engaging in an imitating behaviour would benefit the imitator in some way. The interacting agents might have different goals and receive different rewards. The imitator must first examine whether the model actions are beneficial or relevant to his own tasks and then extract a possible behaviour to imitate. If the agent has to choose among several models, some evaluation of the performance of the appropriate behaviour(s) by the possible candidate models is required before a choice is made (cf. [Dautenhahn, 1994]). Note, an agent is not required to be aware of the fact that the performed behaviour is to be imitated by another agent, although it might help to provide feedback on the success of the imitative behaviour.

For the work described in this report, it will be assumed, in general, that the relationship is given (demonstrator = human user, imitator = robotic companion) and not to be discovered by the imitating agent. It will also be assumed that the human is aware of her or his role as a demonstrator and will not try to deceive or confuse the robot, but instead try to demonstrate the task in the most possibly clear way.

***When* to imitate?** Deciding when to engage in imitating behaviour, the appropriate time, situation, context etc. is another important question that must be resolved. *Immediate imitation* leads to synchronous behaviour, with the agents sharing the same context and using the same objects. This can be beneficial in speeding the process of learning. *Deferred imitation* might occur later, even at the absence of the model but within a relevant context.

[1] In this report the term *demonstrator* will be used to indicate an agent that has an active 'teacher' role, i.e. is aware that a 'student' is observing the demonstration and will try to accommodate it. In contrast, the term *model* is used in a more generic case to indicate any agent that performs a behaviour that could be imitated by another agent.

Synchronic imitation occurs at the same time as the model's behaviour, using objects similar to the ones the model is using [Nielsen and Dissanayake, 2003].

The question of *when* also includes the issue of *why* to imitate. The agent should be motivated to imitate the model instead of imitating indiscriminately all the time. For example there might be a benefit for imitating, either for the individual, or for its kin/group.

For the work described in this report, the imitator might perform immediate or synchronic imitation while learning (depending on whether the same or corresponding objects are used) and deferred imitation when the task that has been learned needs to be performed at a later time.

***What* to imitate?** There are several aspects of a demonstrated behaviour that could be imitated. It may be preferable to imitate *actions*, *states*, or desirable *effects* of an observed behaviour [Nehaniv and Dautenhahn, 1998, 2001, 2002]. The structure of the knowledge transferred presents another problem, as there can be a distinction between different modes of imitation. Richard Byrne and Anne Russon propose two different kinds of imitation, *program level* and *action level* as opposite ends of a spectrum, i.e. copying the organizational structure of the behaviour versus copying the surface form of behaviour. As a general consequence of this, an agent is required to have the ability to build hierarchical structures in order to exhibit program level imitation [Byrne and Russon, 1998].

The purpose of the imitation relates to the degree of its success. Using the example of imitating a dance instructor, one might choose to imitate only the end result, e.g. reaching a specific location on the dance floor. This can be done either by following the path that the instructor used or maybe ignoring it (partially or totally). The student can also try to do the dance steps that the instructor performed while moving on that path. In this example choosing to ignore either the path or the actions that need to be performed in the exact order of sequence will result in poor imitation according to most dance judges. This is especially true because as external observers they can be unaware of the goals and priorities that the imitator has chosen. Using a different *granularity*, i.e. distinguishing which are the important aspects of the model performance and dividing them into a number of sub-tasks, will produce different results.

In the case of imitating the painting of a wall, the situation is quite different. The net result, i.e. covering the entire wall with paint is what is important and can be achieved in numerous ways. The order of paintbrush strokes in the sequence is not important. Nor is replicating the exact actions. The use of a brush of different width or type than the one used by the model is also not important here, nor is e.g. the use of a ladder to make reaching the higher part of the wall easier. The imitator is free to vary all three aspects and can still succeed. Replicating exactly the same hand movements as the model, but being far from the wall without the brush making actual contact with the surface will result in failure due to misinterpretation of the desired result. Using the same example it can also be pointed out that even if the entire wall is finally covered, any paint stains on the floor or the rest of the room will undermine the overall result. Therefore, taking into account the (both desired and undesired) effects on the environment can affect the success of the imitation.

The perspective of the observed actions - whether it is absolute or relative - poses another important issue. If any of the actions performed by the model are not mirror-symmetric (as is most usual), choosing among the possible versions will affect the imitation result, e.g. if a robot is to imitate somebody that waves her right hand should the robot raise its right or left hand? (Not to mention what should the robot should do if it has a gripper but no hands!)

The complementary question of *what* to imitate is addressed by WP4.1 (“Sub goals extraction and metrics of imitation performance”). Later in the COGNIRON project, work from WP4.1 will be integrated with the work stemming from WP4.2 (“Correspondence mapping across dissimilar bodies (How to imitate)”, presented in this report) and provide metrics and granularity to be used in the experiments using the JABBERWOCKY system described in section 7. For the work presented in this report, in order to perform the *how to imitate* experiments until month 18 and from the start of the first year (where the WP4.1 results were not yet available), the demonstrated behaviour aspects that are to be imitated (actions, states and/or effects) are given in advance, and not discovered by the agents.

***How* to imitate?** Once the agent has decided *who* and *what* to imitate, appropriate mechanisms must be employed to achieve the necessary imitating actions. The embodiment of the agent and its affordances will play a crucial role. It is possible that the model and imitator agents will have dissimilar embodiments with different number of parts, limbs, joints, degrees of freedom (DOF). Even small differences in the required affordances might affect the performance of the imitation behaviour. In order to decide which muscles or motors and actuators to use in order to move its body parts, the agent must (at least partially) solve[2] the *correspondence problem* (see section 4), the main focus of COGNIRON WP4.2 (“Correspondence mapping across dissimilar bodies (How to imitate)”) and the research presented in this report.

How to *evaluate* the imitative behaviour? For an attempt made to imitate the model behaviour, there needs to be a measure of evaluating the behavioural matching. Explicit metrics are probably not used by animals and humans, but in artificial systems, the choice of an appropriate metric is very important, as it will be used to capture the notion of the difference between performed and desired actions and also the difference between attained and desired states [Nehaniv and Dautenhahn, 2001, 2002, Alissandrakis et al., 2002, 2003a, 2003b]. The evaluation can be performed either by the imitator, the model or an external observer.

The question of *evaluating* the imitative behaviour is closely linked to the question of *what* to imitate; the metrics used will have to measure the similarity of the chosen aspects to imitate (actions, states and/or effects). For example, if the *states* of the demonstrated behaviour are to be imitated, appropriate *state metrics* must be used.

For the work presented in this report, as the aspects to imitate are given and not discovered by the imitating agents, the metrics will also be given, instead of found from observing the

[2]This can also involve learning.

task demonstration. For the experiments described in this report, the agents are imitating the *effects* aspect of the demonstration and we have defined our own *effect* metrics[3] (see section 8).

Each question presents its own difficulties and research problems. An integrated approach to these questions must be the ultimate goal of work on imitation in adaptive systems, in particular for robots learning skills and tasks from naive human users.

[3]Also, towards a characterization of the *space of effect metrics*, we are using these metrics to explore absolute/relative angle and displacement aspects, focusing on overall arrangement and trajectory of manipulated objects (see section 8).

4 The Correspondence Problem in Imitation

A fundamental problem when learning how to imitate is to create an appropriate (partial) mapping between the actions afforded by particular embodiments to achieve corresponding states and effects by the model and imitator agents [Nehaniv and Dautenhahn, 1998b]. For similar embodiments, this seems to be straightforward (although it actually involves deep issues of perception and motor control). But once the assumption that the agents belong to the same 'species', i.e. have sufficiently similar bodies and an equivalent set of actions, is dropped, as with a robot imitating a human, the problem becomes more difficult and complex. Even among biological agents, individual differences in issues of perception, anatomy, neurophysiology, and ontogeny can create effectively dissimilar embodiments between members of the same species. A close inspection of seemingly similar artificial agent embodiments can yield similar conclusions due to issues like individual sensor and actuator differences (hardware) or the particular representations and processing that these agents employ (software). In the COGNIRON setting, it will be desirable to have different kinds of agents in the learning process, i.e. humans and robots interacting socially.

The following statement of the *correspondence problem* [Nehaniv and Dautenhahn 2000, 2001, 2002] draws attention to the fact that the model and imitator agents may not necessarily share the same morphology or may not have access to the same affordances:

> Given an observed behaviour of the model, which from a given starting state leads the model through a sequence (or hierarchy [or program]) of *sub-goals* in *states*, *action* and/or *effects*, one must find and execute a sequence of actions using one's own (possibly dissimilar) embodiment, which from a corresponding starting state, leads through corresponding sub-goals - in corresponding states, actions, and/or effects, while possibly responding to corresponding events.

In this approach, through a correspondence[4] an imitator can map observed actions of the model agent to its own repertoire of actions as constrained by its own embodiment and by context [Nehaniv and Dautenhahn 2000, 2001, 2002]. Qualitatively different kinds of social learning result from matching different combinations of matching actions, states and effects at different levels of granularity [Nehaniv, 2003] (see subsection 4.1).

Artificial agents that have the ability to imitate may use (perhaps more than one) metrics to compare the imitator agent's own actions, states and effects with the model's actions, states and effects, in order to evaluate the imitative behaviours and discover corresponding actions that they can perform to achieve a similar behaviour. The choice of metrics used is therefore very important as it will have an impact on the quality and character of the imitation. Many aspects of the model behaviour may need to be considered, as the metrics capture the notion of the salient differences between performed and desired actions and also the difference between attained and desired states and effects [Nehaniv and Dautenhahn 2001, 2002]. The choice of metric determines, in part, *what* will be imitated, whereas solving the correspondence problem concerns *how*

[4]In the context of the work presented in this report, a *recipe* is an embodiment-independent overall (loose) plan, taking into account the metrics, sub-goal granularity and initial conditions. Each recipe can be adapted according to a targeted platform's own embodiment restrictions and context, to be executed by the imitator.

Table 1: **A taxonomy of social learning, imitative and matched behaviour.** Extending [Call and Carpenter, 2002], different theoretical terms of social learning, imitative and matched behaviour can be associated with each combination of *goals*, *actions* and *results*. Note that by using the term social *learning*, Call and Carpender assume at least a degree of novelty. However the classification can be used just as well also for imitative or matched behaviour when there is no novelty.

Goals *(understood)*	**Actions** *(copied)*	**Results** *(reproduced)*	**Theoretical term**
yes	yes	yes	Imitation
yes	yes	no	Failed Imitation
yes	no	yes	Goal Emulation
yes	no	no	Goal Emulation
no	yes	yes	Mimicry
no	yes	no	Mimicry
no	no	yes	Emulation
no	no	no	no Social Learning

to imitate [Dautenhahn and Nehaniv, 2002]. In general, aspects of action, state and effect as well as the level of granularity (*what to imitate*) do all play roles in the choice of metric for solving the problem of *how to imitate* [Nehaniv and Dautenhahn, 2001, Alissandrakis et al., 2002, Billard et al., 2004]. On-going research is thus addressing the complementary problem of how to *extract sub-goals* and *derive suitable metrics* automatically from observation [Nehaniv and Dautenhahn, 2001, Nehaniv, 2003, Billard et al., 2004, Calinon and Billard, 2004] and COGNIRON D4.1.1.

4.1 Taxonomies of Imitation, Social Learning and Matched Behaviour

Josep Call and Malinda Carpender in [Call and Carpenter, 2002] associate the theoretical terms *imitation*, *goal emulation mimicry* and *emulation* within a common framework based on *goals*, *actions* and *results* (see Table 1).
To illustrate the distinction of goals, actions and results (as used in [Call and Carpenter, 2002]), let us assume the imitator is observing the model manipulating a box[5]. If the box is transparent and an object, such as piece of candy, can be seen inside, the imitator may understand that the *goal* of the model is to retrieve it. If afforded by the box, the imitator might open the box in a different way, instead of copying the exact model *actions*, for example moving a sliding side instead of lifting the lid. Opening the box will reproduce the *result* of having an open box. The

[5]This box could be similar to the ‘artificial fruits’ used by [Whiten, 2002] for investigating the imitation of sequential and hierarchical actions in human children and chimpanzees.

goal could be "misunderstood" as simply opening the box. Alternatively, the imitator might instead crush the box achieving the goal of obtaining the candy inside.

The work presented in this report does not use this taxonomy by [Call and Carpenter, 2002], which was concerned with terminological issues in psychology and biology, but a related one developed also for AI applications. For the research presented in this report we will use the systematic taxonomy presented in [Nehaniv, 2003] shown in Table 2 that uses the combination of metrics that measure the similarity in a given aspect of the model's behaviour and the granularity, to define classes of imitation *correspondence problems*. This taxonomy is broad and can be used in both psychology and biology, and also in computer science and artificial intelligence.

Call and Carpenter's taxonomy merges aspects of *what* to imitate and *how* to imitate, but from the perspective of the correspondence problem, it is useful to separate these issues. Nehaniv offers a taxonomy strictly for the latter. Although *actions* are also used in Nehaniv's framework, the *results* are partitioned into *states* (of the agent bodies) and *effects* (changes in the environment). The notion of *goals* (and to some extent *results*) as used by Call and Carpenter is a separate issue related to *what to imitate*. Inferring another agent's goals is a difficult problem which will not be addressed in this report. Compared to humans and animals, for artificial agents, recognizing *intentionality* (understanding the goals of the model) remains a largely unsolved problem. The choice of metrics and granularity that the agent should use to imitate is another complementary problem addressed by on-going research on *goal extraction* [Nehaniv and Dautenhahn, 2001, Billard and Schaal, 2002, Nehaniv, 2003, Billard et al., 2004, Calinon and Billard, 2004].

Rather, in the system presented and the experiments discussed in this report (see sections 7 and 9) goals (or sequences of sub-goals), metrics and granularity are not extracted by the imitator, but are given.

Cognitive companions learning tasks and skills in the home by imitating human demonstrators will need to handle aspects in all these classes. For the work presented in this report we will concentrate in the *effects* aspect (classes 3 to 5), as effects on objects in the environments (their arrangement and manipulation) are likely to be of primary interest to a human user in a home environment.

In the future, the work will extend to consider *states*, *actions*, and combinations of them.

Table 2: **Classes of correspondence problems arising from particular combinations of metrics and granularities.** Each class is defined by the combination of metrics that measure the similarity in a given aspect of the model's behaviour and the granularity used. [Nehaniv, 2003] relates this taxonomy to theoretical terms traditionally used to classify types of social learning and matching behaviour.

Metrics			*Granularity*	**Class**
–	–	–	end	0
–	–	–	coarse	1
–	–	–	fine	2
–	–	effects	end	3
–	–	effects	coarse	4
–	–	effects	fine	5
–	states	–	end	6
–	states	–	coarse	7
–	states	–	fine	8
actions	–	–	end	9
actions	–	–	coarse	10
actions	–	–	fine	11
–	states	effects	end	12
–	states	effects	coarse	13
–	states	effects	fine	14
actions	–	effects	end	15
actions	–	effects	coarse	16
actions	–	effects	fine	17
actions	states	–	end	18
actions	states	–	coarse	19
actions	states	–	fine	20
actions	states	effects	end	21
actions	states	effects	coarse	22
actions	states	effects	fine	23

5 The ALICE framework

A mechanism that solves the correspondence problem is an essential part of any artificial system with successful imitating behaviour capabilities. In previous work we have developed ALICE (**A**ction **L**earning via **I**mitating **C**orresponding **E**mbodiments), a generic framework for solving the correspondence problem [Alissandrakis et al., 2002, 2004, Alissandrakis, 2003]. The ALICE framework assumes that an *imitator* agent is exposed to a *model* agent, which is performing a sequence of *actions*, these actions comprising the *behaviour* of the model. By performing an action, the agents can change their *state* and cause some *effects* on their environment. The ALICE framework builds up a library of actions from the repertoire of an imitator agent that can be executed to achieve corresponding actions, states and effects to those of a model agent (according to given metrics and granularity) (see Figure 1).

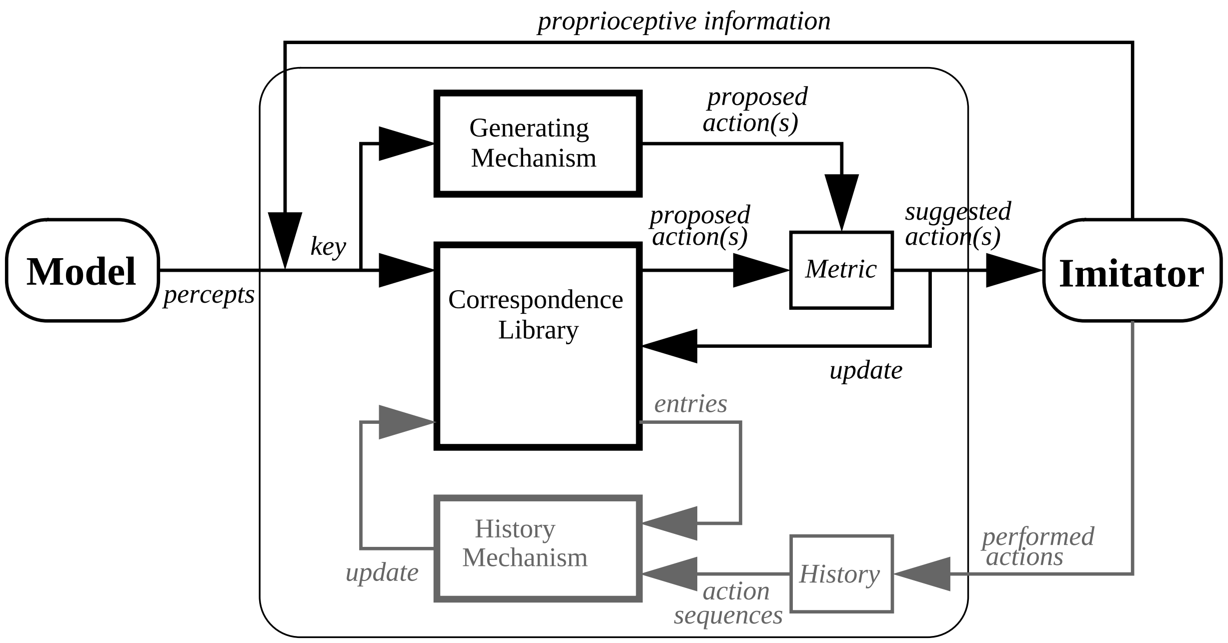

Figure 1: **The ALICE framework.** The *percepts* of the imitator arising from the model's behaviour (actions, states and effects) and *proprioceptive information* (state) of the imitator form a *key* that is used by the *correspondence library* (if it matches any of the existing entry keys at that stage of the library's growth) and the *generating mechanism* to produce a sequence of one or more *proposed action(s)*. These are evaluated using a *metric*, and the correspondence library is updated accordingly with the resulting *suggested action(s)* for the imitator. In parallel (shown in the figure using a grey color), the *history mechanism* can be used to discover any *action sequences* from the *history*, that can improve any of the existing library entries. The sequence of all the actions performed so far by the imitator composes the history.

The ALICE framework is comprised of two major components on top of any *generating mechanism* (see section 5.3) of imitative behaviours. An overview of the framework is shown in Figure 1. The first component builds-up a *correspondence library* (see section 5.1), using perceptual (both exteroceptive and proprioceptive) data to relate the model actions, states and effects to corresponding actions that can be performed by the imitator and result in similar [6] states and effects. The second component, the *history mechanism* (see section 5.4), addresses any limitations of the generating mechanism used by the agent, by discovering additional alternative correspondences from the imitator's action *history*.

The Correspondence Library When observing the model (at a certain sub-goal granularity), the imitator agent constructs a *key*, consisting of the perceived action(s), state(s) and/or effects of the model. The key can also include proprioceptive information (state of the imitator). This key is compared with the *entry keys* of the correspondence library so far. Depending if the key matches another existing entry key, the following procedure will be taken:

- **Create a new entry** If the key doesn't match any other corresponding library key, then a new entry is created in the library, with this key as the entry key. This entry will initially contain a corresponding action of the imitator, found by the *generating mechanism*.
- **Update an existing entry** If the key matches an existing entry key, then either:
  - **Use the entry** The entry will contain one (or more) corresponding imitator action sequences. If more than one exists, choose one from the list according to some criteria, e.g. the shortest imitator's action sequence. Or:
  - **Modify the entry** If the generating mechanism can produce a better (according to the metric) imitator action, modify the entry to contain it.

The imitator can then use this imitator action to imitate (changing its state and resulting in effects on the environment).

The History Mechanism In addition, the imitator agent can examine its history (a list of sequential imitator actions *performed* so far).

- **Update an existing entry** If an imitator's action (or a sequence) in the history results in the same (or sufficiently similar) imitator's state and/or effects, as compared to the model's state and effects of an existing entry key, then modify that relevant entry in the correspondence library, by adding this imitator's action(s) to its list of actions under that key.

There are a variety of existing machine learning techniques addressing experience-based learning, for example, reinforcement learning [Sutton and Barto, 1998], case-based reasoning [Kolonder, 1993], inductive learning [Sammut et al., 1992, Quinlan, 1993], or learning of behavioural histories [Michaud and Matarić, 1999], and others. The intention is not to develop a new and efficient machine learning algorithm, but instead a framework for learning *how to imitate* by

[6]Unless otherwise stated, comparison and similarity depend upon the metrics used in each context. Matching comprises either a *perfect* match, or a *loose* match that can depend on a similarity threshold.

solving the correspondence problem is proposed and systematically studied. The ALICE framework provides a functional architecture that informs the design of robotic systems that can learn socially from a human demonstrator. Clearly, this framework can easily be combined with these or many other machine learning techniques.

5.1 The Correspondence Library

The correspondence library[7] consists of a number of entries, each identified by a *key*. The key can be any combination of perceptual (model's action, state and effects) and proprioceptive (imitator's state) data. These data relate to aspects of the model's behaviour, and the imitator's state when those aspects were first observed. Each entry will contain one (or a list of) imitator's action(s) that if performed when the key is next observed, should achieve corresponding imitator's state and/or effects. Note that a metric value associated with the performance of any of these imitator's actions is not kept in the entries, as it can be dependent of the imitator context (imitator's state).
A *similarity threshold* may be used to determine if a key has been observed before. Using a *looser matching* will result in a smaller number of entries in the library. As these entries are more generic, they will be updated more frequently (if the imitator is repeatedly exposed to the model), resulting in faster rate of learning. On the other hand, the imitating performance may or may not be acceptable depending on the minimum granularity required by the model behaviour.

5.2 Metrics and Granularity

The choice of metric and granularity will in part determine what the agent will imitate and affect the overall character of the imitation. The metrics used should capture all the necessary and important aspects of the actions that the agents can perform, the states they can be in, and possible effects on the environment.
The comparisons may involve all three aspects of the behaviour, *actions*, *states* and *effects*, or any combination of them. The combination used (together with the choice of granularity) will define the correspondence problem *class* (see section 4.1, table 2). A *matching or similarity threshold* can be used, to determine how 'similar' they need to be in order to match.

5.3 The Generating Mechanism

A *generating mechanism* is used by the ALICE framework to produce the contents of the correspondence library; it can be *any* algorithm or mechanism that generates actions (or sequences) that are valid (i.e. within the agent's repertoire) and possible in the context of the imitator.
The fact that an imitative action – even an accidental one – may receive positive feedback could increase the animal's motivation and tendency to imitate (cf. [Dautenhahn and Nehaniv, 2002]).

[7]At each stage in its growth, a library of correspondences is an example of a (partial) relational homomorphism between the abstract automata associated with the model and the imitator [Nehaniv, 1996, Nehaniv and Dautenhahn, 2001, Nehaniv and Dautenhahn, 2002].

Moreover, this can serve to draw attention toward salient aspects of the environment and reveal affordances of actions and objects useful for survival in the course of ontogeny. In the ALICE framework, no direct feedback from the model is used, instead the metrics are used to evaluate the imitative behaviours.
In the ALICE overview, the generating mechanism remains underspecified, as the performance of the ALICE framework does not depend on the precise implementation of the generating mechanism as long as this mechanism returns a valid action that can be performed within the current context by the imitator, from the entire action-space of the agent.
Generally, depending on the size and dimensions of the action-space, the generating mechanism would more efficiently consider the states of the model and the imitator agents (also the effects of the model), and use them to construct the selection of the corresponding actions (e.g. when using inverse kinematics). Alternatively, either a less precise generating mechanism, or a generating mechanism that produces entirely random actions might serve just as well the purpose of exploring the action-space when solving the correspondence problem, especially if the state-space is sufficiently ‘small’. For a large action search space, a ‘smarter’ and more sophisticated generating mechanism is required.
Depending on the position where it lies on this spectrum, the generating mechanism may range from inadequate if used on its own (e.g. returning just random sequences of moves) to a highly specialized complex algorithm that can return the most appropriate correspondence possible without need for learning. For real-world applications, one would expect that the more efficient the generating mechanism used, the faster the learning rate will then be.

5.4 The History Mechanism

If only the actions found by the generating mechanism are used to build-up the correspondence library, the performance of the imitator would be directly limited by the choice of the algorithm. Moreover, some of the stored actions, although valid solutions to the correspondence problem related to the actions of the model, may become invalid in certain contexts (state of the imitator). The *history mechanism* helps to overcome these difficulties: The imitator can examine its own history to discover further correspondences without having to modify or improve the generating algorithm used. These correspondences will be *sequences of* actions since, no matter how simplistic, the generating mechanism is required to be able to explore the entire search-space of single actions.
The agent’s *history* is defined as the list of imitator’s actions that were performed so far by the agent while imitating the model together with their resulting imitator’s state and effects. This kind of history provides valuable experience data that can then be used to extract useful mappings to improve and add to the correspondence library created up to that point. The contents of the history are useful, since the results (imitator’s resulting state and effects) of imitator actions are known from past experience given a certain context (imitator’s previous state), without need for *prediction of performance*[8].
The methods for extracting this information from history can vary depending on the particular

[8]This remains a hard problem, especially for physical robotic systems.

realization, and managing the found sequences of actions can depend on additional metrics (e.g. keeping only the shortest sequence that can achieve the desired state and/or effects, or keeping only the top five sequences according to a performance measure). The size of the history will also depend on the actual implementation and/or the task context.

6 Relevance to COGNIRON Functions and Key-Experiments

The work presented in this report for WP4.2 (*"Correspondence mapping across dissimilar bodies (How to imitate)"*) helps to provide rigorous scientific foundations for COGNIRON functions CF-RG:*"Learning to reproduce gestures"* and CF-LIF:*"Learning important features of a task"*, and develops in synergy with workpackage WP4.1 (*"Sub-goals extraction and metrics of imitation performance"*) by (1) providing a broad theoretical and practical systematic scientific framework from which to select what aspects of behaviour to imitate (*goal and sub-goal metrics for states, actions and effects*) going beyond the current state-of-the-art, and (2) by building a multi-targetable architecture which allows the specific features extracted from human demonstrations to be imitated on different platforms. The work is also relevant to one of the COGNIRON key-experiments, KE3:*"Learning Skills and Tasks"*.

6.1 CF-RG: Learning to Reproduce Gestures

The robotic companion in COGNIRON will have to be able to replicate a demonstrated task by its user (imitate), although its embodiment and affordances will differ from those of a human. The work presented in this report is focused on addressing the *correspondence problem* in imitation (see section 4) and the question of *how to imitate* (see section 3.1, page 5).
The JABBERWOCKY system presented in section 7 uses captured data from a human demonstrating a task and given metrics and sub-goals, provides multi-platform targeted solutions for the correspondence problem. These solutions can be converted into command actions that multiple imitator robotic platforms (currently in simulation only, in the future extended to include hardware) can execute to achieve a successful imitation of the task, depending on the particular imitator's embodiment and context.

6.2 CF-LIF: Learning Important Features of a Task

In order to perform initial experiments with the current implementation of the JABBERWOCKY system (described in section 7), a series of metrics is defined (see section 8) that are *given* to the imitating agents for matching the demonstration behaviour aspects. These defined metrics also contribute towards a characterization of the *space of metrics*, that will be useful in guiding various types of robotic imitation and social learning from human demonstrators.
In the future, results from WP4.1 (*"Sub-goals extraction and metrics of imitation performance"*) will be used to allow the imitating agents to learn what metrics to use (and also how to extract sub-goals) by observing the demonstration by the user (*what to imitate*).

6.3 KE3: Learning Skills and Tasks

The key-experiment *Learning Skills and Tasks* stresses the learning and reasoning capabilities for the robot to acquire knowledge about goals and tasks. At the current stage it is planned to demonstrate and assess the following skills which are implemented on a robot platform:

- Learning goals from observations (RA4, RA5, RA6).
- Reproduction of the goal for arbitrary starting conditions (RA4, RA6).

The work presented in this report is relevant to KE3 script: *Learning Skills "Arranging and interacting with objects"*. The script stresses the robot's ability to learn from implicit (imitation learning) and explicit (verbal interaction) teaching, and is envisioned as follows: The robot learns new skills to manipulate objects and, by so doing, it learns a new task. The robot will watch a human demonstrator performing a task of manipulating some objects. The demonstration will be repeated several times. Each demonstration will be slightly different from the others. The order that the objects are manipulated may change, as well as the relative positions and displacements of the objects. The absolute position of the objects may vary as well. While watching the demonstrations, the robot will learn the invariants of the task (relative position and orientation of the objects with respect to one another) and new skills such as object-actions relations (how to grip an object). Once the demonstrations are finished, the robot will try to reproduce the task. While doing so, the robot might query the user if some demonstration were ambiguous and its choice is non deterministic. The user might stop and correct verbally the robot during the reproduction, if the robot makes important mistakes.
As the embodiment of the human user and the robotic companion will be dissimilar, a *correspondence problem* must be solved, defined by the imitator's embodiment, the sub-goal granularity, and the metrics used to match the behaviour aspects. Extensions of the JABBERWOCKY system presented in section 7 could be used to generate action commands that a robotic companion could use to imitate the tasks demonstrated by a human user. Since the robotic platform to be used in the COGNIRON KE3 is not yet fixed and also for maximum applicability, WP4.2 has adopted a generic approach to the correspondence problem, targeting multiple different potential platforms. The corresponding solutions produced by the system can be targeted to multiple robotic platforms and adapt to arbitrary starting conditions.

7 Functional System Architecture

This section presents the JABBERWOCKY system developed for the COGNIRON project, addressing the *correspondence problem* in imitation (see section 4). The design of the JABBERWOCKY system is informed by the ALICE framework (see section 5).

7.1 System Overview

The JABBERWOCKY system uses captured data from a human demonstrator to generate appropriate action commands that can be targeted for various software and hardware platforms. These actions allow the imitating agent to achieve corresponding *actions*, *states* and *effects*, depending on the (relevant to the demonstrated task and context) metrics and granularity (provided by a *what to imitate* and *sub-goal extraction* module), embodiment restrictions and constraints (imposed by the targeted imitator platform), and possibly different initial state of the objects in the environment (see Figure 2).
The corresponding actions, states and effects as performed by the imitator can also be captured and used as a demonstration for another imitating agent, allowing for a form of cultural transmission.
The system bears some similarity to the one presented in [Kuniyoshi et al., 1994], but with the main differences that it can use *any* given metric and granularity and is designed to be able to generate action commands targeted for a variety of platforms, both in software and hardware to match different behaviour aspects and achieve various types of social learning.
The JABBERWOCKY implementation described in this report and used for the experiments (see section 9) was implemented using the *Swarm simulation system*[9] (coded in Obj-C) and MATLAB.
The targeted imitator platforms used so far were implemented and simulated using Webots^{TM}; the *Xanim* humanoid robot simulator is also under consideration.

7.2 Demonstrator (Model Agent)

The system uses captured data from a human demonstrator. The demonstrated behaviour is captured using motion sensors (Polhemus LIBERTY^{TM} motion capture system). By attaching the motion sensors on the arms, hands and torso of the human, as well as on the objects that the demonstrator is manipulating, we can obtain the *actions* (e.g. hand movements, gestures), *states* (e.g. arm and body postures) and *effects* (e.g. positioning, displacement, rotation of objects in the workspace) of the demonstrator.

7.2.1 The Polhemus LIBERTY^{TM} Motion Capture System

The motion capture system used is the LIBERTY 240/8 with eight sensor channels. It has 240 Hz update rate per sensor (simultaneous samples) and 3.5 msec latency. The resolution is 0.038

[9]Swarm is a software package for multi-agent simulation of complex systems. The official homepage of the Swarm Development Group can be found at `http://wiki.swarm.org`.

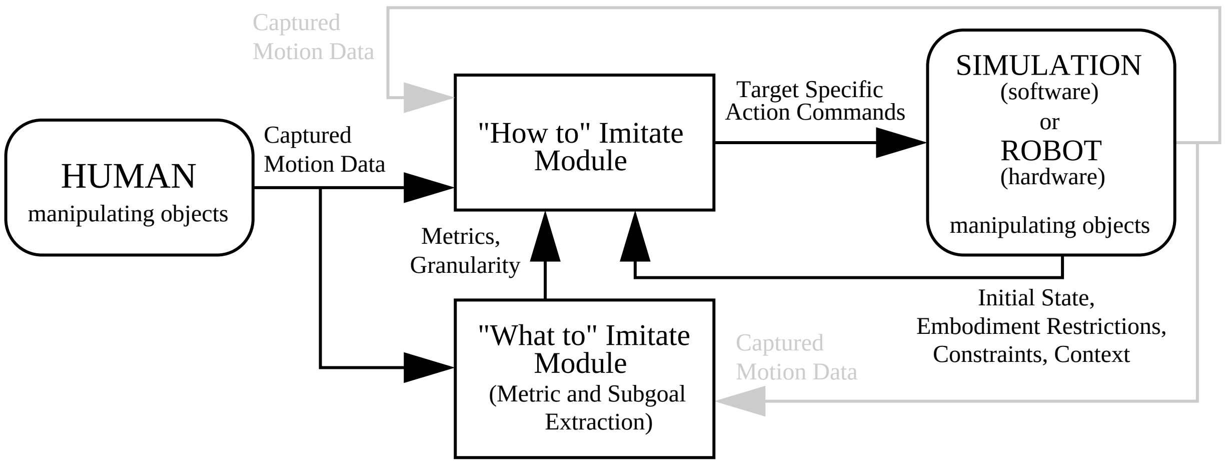

Figure 2: **The JABBERWOCKY system architecture.** Using data captured from a human and given appropriate metrics and sub-goal granularity, the multi-target system can produce action command sequences that when executed by a software or hardware agent can achieve corresponding actions, states and/or effects. The corresponding actions, states and effects as demonstrated by the imitator can also be captured and used as a demonstration for another imitating agent. Differently embodied and constrained target systems in various contexts need to be supported.

mm at 30 cm range; 0.0012° orientation. The static accuracy is 0.08 cm RMS for X, Y or Z position; 0.15° RMS for sensor orientation. The range is 90 cm at above specifications, with useful operation in excess of 180 cm (the range used in the experiments was typically 50 cm). The LIBERTY unit is connected to a Linux operating PC via the USB I/O port. The data format used is ASCII in metric units. Depending on the experimental run, 1 to 8 sensors can be used.

7.3 Imitator (Target Platform)

The system is addressing the correspondence problem for dissimilarly embodied imitators, so the *how to imitate* module produces action commands that can be used by multiple different target platforms as imitator agents, both in simulation (software) and hardware (robots).
Each particular target platform will pose different embodiment restrictions and constraints to the actions, states and effects it can achieve, and eventually to the quality and character of the imitation.
The demonstrator and the imitator might share the same workspace or they might operate in different ones. Even in the same workspace, unless the objects and agents positions are arranged back into the same initial configuration before the imitative behaviour, the context will be different and the imitator therefore has to take that into consideration when imitating.
Once the imitator performs the actions using the same (or corresponding) objects in its workspace, its actions, states and effects could also be captured and used as a demonstration for another agent, allowing for a form of *cultural transmission*.

7.3.1 The Webots^{TM} Simulator

The Webots mobile robotics simulation software provides users with a rapid prototyping environment for modelling, programming and simulating mobile robots. The included robot libraries enable users to transfer their control programs to many commercially available real mobile robots (including Aibo©, Lego© Mindstorms©, Khepera©, Koala© and Hemisson©) [Michel, 2004].
For WP4.2, the Webots platform was used to implement simulations of two different target imitator platforms (see figures 3 and 4) and described in section 9.4:

- Multiple mobile robots, each corresponding to one of the manipulated objects (described in section 9.4.1).
- A robotic manipulator arranging the corresponding objects (described in section 9.4.2).

7.3.2 The Xanim Simulator

Another possible target platform evaluated for JABBERWOCKY is the *Xanim* simulator created by Stefan Schaal [Schaal, 2000]. *Xanim* is a dynamic simulation of the 30 degrees of freedom humanoid robot *DB*, located at the Advanced Telecommunication Research Institute (ATR) in Kyoto (see figure 5).
Substantial extension of the physical modeling of objects and their manipulation environment in *Xanim* would be required to exploit the extent of effect matching now available with JABBERWOCKY, although this limitation does not preclude its possible use for state and action imitation of a (stationary) human (e.g. see [Billard et al., 2004, Calinon and Billard, 2004]).

7.4 *What to Imitate* Module

The *what to imitate* module will use the captured demonstration data to extract appropriate subgoals (granularity) and also discover what metrics must be used to capture the appropriate aspects of the particular demonstration.
As noted in section 3.1, page 4, the question of *what* to imitate is addressed by WP4.1. Currently, both metrics and the sub-goal granularity are given, as the results from WP4.1 are not yet available to be integrated in the implemented system.
So far, the work on WP4.2 has concentrated on solving the correspondence problem for the *effects* aspect of the demonstrated behaviour, that is the manipulation of objects in the workspace. Several different *effect* metrics have been defined (see section 8) that are used in the experiments presented in this report. In the future the work will be extended to consider the *state* and *action* aspects of a demonstration.
In the current implementation of the JABBERWOCKY system, the sub-goal granularity is given by finding the *critical points* in the trajectories of the manipulated objects. A critical point occurs when the direction of the captured trajectory and/or the orientation of an object changes by more than a certain threshold.

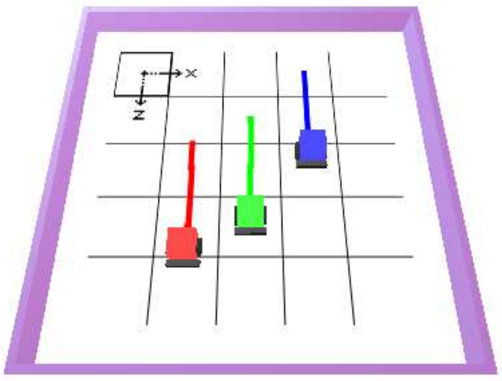

Figure 3: **The *three robots as objects* platform.** This platform (described in detail in section 9.4.1) is implemented and simulated using Webots. Each robot (red, green and blue) corresponds to an object from the demonstrator's workspace (according to the color), and leaves a trail to help visualizing the imitative behaviour trajectory.

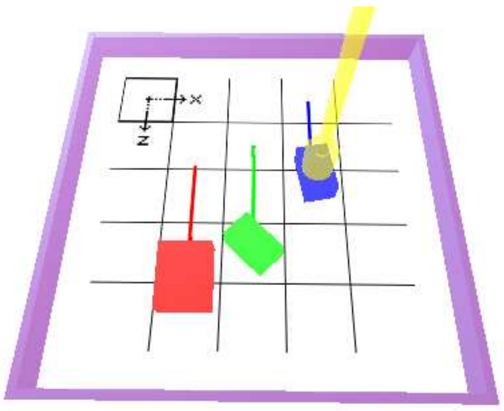

Figure 4: **The *manipulator and three objects* platform.** This platform (described in detail in section 9.4.2) is implemented and simulated using Webots. The manipulator (yellow) is positioned above the workspace and able to move and rotate the three colored objects (in this case, same size as the corresponding ones manipulated by the demonstrator). When moved, each object leaves a colored trail to help visualize the imitative behaviour trajectory.

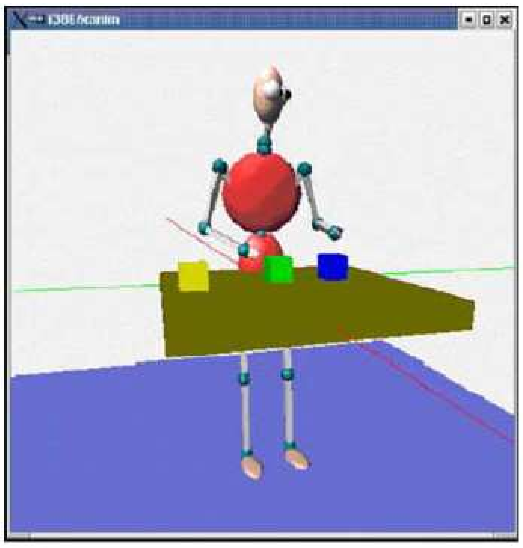

Figure 5: **The Xanim simulator.**

7.5 *How to Imitate* Module

The *how to imitate* module uses the captured data from the demonstration, the metrics and the sub-goal granularity[10] discovered by the *what to imitate* module to produce a sequence of action commands for an agent to execute and imitate. These action commands are made target specific by taking into account the particular embodiment, affordances and restrictions of the target imitator agent, and also contextual information (including the initial state of) for both the agent and the environment.

Concentrating on the *effects* aspect of the demonstrated behaviour to be imitated only, an embodiment-independent solution to the correspondence problem can be found, taking into account the *effect* metrics and the sub-goal granularity. For example consider a human opening a cupboard, removing an object, closing the cupboard and placing the object on a table. This sequence of events can be achieved by agents of varying embodiments, ignoring *state* aspects like e.g. which hand was used to open the cupboard or how the object was held (or grasped) or even *action* aspects like e.g. the way the human walked (gait) across the room. Any agent that can open the cupboard, transport the object and place it on the table can potentially imitate the effects of this particular demonstration. But for this solution to be useful to an imitating robotic companion, it must be converted to action commands that take into account its embodiment and also the context (e.g. the cupboard is already open, the object is located on a different shelf in the cupboard, the table is in another room), so that the imitator uses its motors and actuators to achieve the desired effects of the task.

The choice of initially concentrating on *effects* is guided by the assumption that the manipulation of objects will be the most important aspect of the demonstrated behaviours that users would like

[10]In the current system implementation both the metrics and the sub-goal granularity (critical points) are given.

a robotic companion to imitate in a home environment (e.g. fetching objects or arranging them in particular ways). In the future, the work will be extended to consider also the *state* and *action* aspects of a demonstration.

8 Metrics

To evaluate how similar its actions, states and effects are to the model's actions, states and effects, an imitator agent uses different metrics (addressing the *how to evaluate the imitative behaviour* question, see section 3.1, page 3.1). When the value of the metric(s) used is minimised, then that imitator action, state or effect (or a sequence of) is "optimized", i.e. is the most similar to the perceived model's behaviour.
As we concentrate on the effect aspect of the agent's behaviour, the metrics defined and used in this work will be various *effect metrics*, evaluating the similarity between the effects on the environment of the model and the imitator. We do not presently consider the state or the actions aspects of the a model's behaviour.
The reason for this prioritization is that in general, unless the embodiments are "very similar" (i.e. a humanoid robot and a human), although an action or state metric can be defined and used, the actions or states that minimize it will be qualitatively very different to those of the model. For example consider a mobile robot on wheels, also having a single arm equipped with a gripper, as an imitator and a human as a model. This robot will perhaps be able to achieve similar arm postures (states) and perform similar gestures (actions) to the human, but only the ones that require a single arm. It will be able to move to the same position in a room (state) but the way it reaches the destination (action, using its wheels) will differ to that of the human (action, walking). Still, this robot is able to arrange and manipulate objects (effects) but will probably achieve them going through a very different sequence of states and actions. Trying to force the robot to e.g. use a specific way to grasp an object, when another way is possible (and perhaps more efficient) can be restricting depending on its particular embodiment and access to affordances. In future work, we will consider states and later actions.

8.1 Effect Metrics

Towards a characterization of the *space of effect metrics*, we are exploring absolute/relative angle and displacement aspects and focus on the overall arrangement and trajectory of manipulated objects. Looking at how objects can be manipulated (in a non-destructive and combining way), there are two different perspectives: how the object was displaced and how it was rotated. The displacement can be either relative or absolute related to the final position in the workspace, or relative to the other objects within the workspace. The rotation can be also be relative or absolute related to the final orientation of the object. To fully describe the manipulation of an object, both displacement and angular effect aspects must be considered. We consider these aspects in a two-dimensional workspace, such as a table surface.
If the initial configuration of the (same or corresponding) objects is the 'same' for both the model and the imitator agents, then there is no observable distinction between using either the absolute and relative displacement/rotation or the relative position (if the objects are manipulated in the same order). But if the agents are active in a different workspace starting from a different initial configuration of objects, or the timing and the order of the manipulations is not the same, it will be impossible to satisfy simultaneously all the aspects. Therefore choosing to satisfy one particular aspect will result in a qualitatively different effect than if another one was chosen,

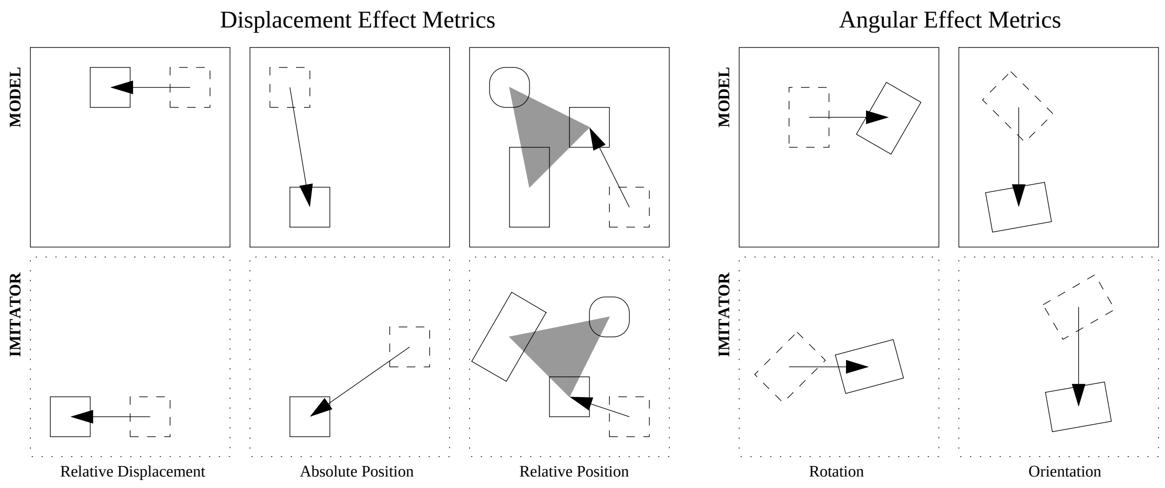

Figure 6: **Some *displacement* (left) and *angular* metrics (right).** To evaluate the similarity between object displacements, the *relative displacement*, *absolute position* and *relative position* effect metrics can be used. To evaluate the similarity between object rotations, the *rotation* and *orientation* effect metrics can be used. The second row shows the way the corresponding object (in a different workspace) needs to be moved or rotated by an imitator to match the corresponding effects. The grey triangles are superimposed to show that for the *relative position* effect metric, the relative final positions of the objects are the same.

but still satisfy those similarity quantitative criteria. This is shown in the experimental results presented and discussed in section 9.

8.1.1 Displacement Effect Metrics

The model is moving an object from position X_M to position X'_M on the workspace, achieving an object displacement $\Delta X_M = X'_M - X_M$, where $X_M = \begin{bmatrix} x_M \\ y_M \end{bmatrix}$, $X'_M = \begin{bmatrix} x'_M \\ y'_M \end{bmatrix}$, and $\Delta X_M = X'_M - X_M = \begin{bmatrix} x'_M - x_M \\ y'_M - y_M \end{bmatrix}$. The imitator should move the same (or corresponding) object from position X_I to position X'_I on the workspace, with a displacement $\Delta X_I = X'_I - X_I$, such that a displacement metric is minimised (see Fig. 6, left).

Relative Displacement Effect Metric is minimized if $\Delta X_I = \Delta X_M$ and

$$X'_I = X_I + \Delta X_M = \begin{bmatrix} x_I \\ y_I \end{bmatrix} + \begin{bmatrix} x'_M - x_M \\ y'_M - y_M \end{bmatrix} = \begin{bmatrix} x_I + x'_M - x_M \\ y_I + y'_M - y_M \end{bmatrix}.$$

Absolute Displacement Effect Metric is minimized if $X'_I = X'_M$ and

$$\Delta X_I = X'_M - X_I = \begin{bmatrix} x'_M - x_I \\ y'_M - y_I \end{bmatrix}.$$

Relative Position Effect Metric This metric is minimized if the object is moved to a similar position relative to other objects in the workspace. The *relative position* effect metric is defined here for three objects in the workspace.

The center of the manipulated object is defined as $A = \begin{bmatrix} x_A \\ y_A \end{bmatrix}$, and the centers of the other two objects as $B = \begin{bmatrix} x_B \\ y_B \end{bmatrix}$ and $C = \begin{bmatrix} x_C \\ y_C \end{bmatrix}$. The imitator must move the same (or corresponding) object to form a triangle $\hat{ABC}$ so that it is the "same" as the triangle formed by the model, i.e. the angles $C\hat{A}B$, $A\hat{B}C$ and $B\hat{C}A$ are equal. The triangle sides $\bar{AB}$, $\bar{BC}$ and $\bar{CA}$ can be equal only if the objects start from the same initial configuration for both agents and are manipulated in the same order, so only the equality of the angles can be used in general[11].

The *relative position* effect metric is minimized if

$$X_I' = A \text{ and } \Delta X_I = A - X_I = \begin{bmatrix} x_A - x_I \\ y_A - y_I \end{bmatrix}.$$

8.1.2 Angular Effect Metrics

The model is rotating an object from orientation θ_M to orientation θ_M' on the workspace, with a rotation $\Delta\theta_M = \theta_M' - \theta_M$. The imitator should rotate the same (or corresponding) object from orientation θ_I to orientation θ_I' on the workspace, with a rotation $\Delta\theta_I = \theta_I' - \theta_I$, such that a displacement metric is minimised (see Fig. 6, right).

Rotation Effect Metric is minimized if $\Delta\theta_I = \Delta\theta_M$ and $\theta_I' = \theta_I + \Delta\theta_M$.

Orientation Effect Metric is minimized if $\theta_I' = \theta_M'$ and $\Delta\theta_I = \theta_M' - \theta_I$.

[11]Given $C\hat{A}B_M$, $A\hat{B}C_M$, $B\hat{C}A_M$ and $\overline{BC}_I$, we can find the other two sides $\overline{AC}_I = \sqrt{\frac{(1-cos^2(A\hat{B}C_M))\times\overline{BC}_I^2}{(1-cos^2(C\hat{A}B_M))}}$ and $\overline{AB}_I = \sqrt{\frac{(1-cos^2(B\hat{C}A_M))\times\overline{BC}_I^2}{(1-cos^2(C\hat{A}B_M))}}$, to satisfy the equalities $C\hat{A}B_I = C\hat{A}B_M$, $A\hat{B}C_I = A\hat{B}C_M$ and $B\hat{C}A_I = B\hat{C}A_M$.
Assuming that side $\overline{BC}_I$ lies on the $(0, +\infty)$ x-axis with points $\mathcal{B} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\mathcal{C} = \begin{bmatrix} |\overline{BC}_I| \\ 0 \end{bmatrix}$ corresponding to B_I and C_I, we can then find a point $\mathcal{A} = \begin{bmatrix} \frac{\overline{BC}_I^2 - \overline{AC}_I^2 + \overline{AB}_I^2}{2\times\overline{BC}_I} \\ \frac{\sqrt{(-\overline{BC}_I+\overline{AC}_I-\overline{AB}_I)\times(-\overline{BC}_I-\overline{AC}_I+\overline{AB}_I)\times(-\overline{BC}_I+\overline{AC}_I+\overline{AB}_I)\times(\overline{BC}_I+\overline{AC}_I+\overline{AB}_I)}}{2\times\overline{BC}_I} \end{bmatrix}$ corresponding to A_I, such that the equalities $\overline{\mathcal{AB}} = \overline{AB}_I$, $\overline{\mathcal{BC}} = \overline{BC}_I$ and $\overline{\mathcal{CA}} = \overline{CA}_I$ are satisfied.
To find A_I we need to rotate and translate $\mathcal{A}$ in respect to the actual co-ordinates of $B_I = \begin{bmatrix} x_B \\ y_B \end{bmatrix}$ and $C_I = \begin{bmatrix} x_C \\ y_C \end{bmatrix}$ in the imitator's workspace: $A = \begin{bmatrix} x_A \\ y_A \end{bmatrix} = \begin{bmatrix} cos\phi & sin\phi \\ -sin\phi & cos\phi \end{bmatrix} \times \mathcal{A} + \begin{bmatrix} x_B \\ y_B \end{bmatrix}$, where $\phi = tan^{-1}\left(\frac{y_C - y_B}{x_C - x_B}\right)$.

8.1.3 Other Effect Metrics

Depending on the initial configuration of the corresponding objects in the imitator's workspace, or the particular task that the imitator would like to achieve, it might be desirable to use also other metrics that take into account mirror symmetry, both positional and angular, to features of the environment or other agents. For example:

Mirror Displacement Effect Metric is minimized if $\Delta X_I = -\Delta X_M$ and

$$X'_I = X_I - \Delta X_M = \begin{bmatrix} x_I \\ y_I \end{bmatrix} - \begin{bmatrix} x'_M - x_M \\ y'_M - y_M \end{bmatrix} = \begin{bmatrix} x_I - x'_M + x_M \\ y_I - y'_M + y_M \end{bmatrix}.$$

Mirror Rotation Effect Metric is minimized if $\Delta\theta_I = -\Delta\theta_M$ and $\theta'_I = \theta_I - \Delta\theta_M$.

Parallel Orientation Effect Metric is minimized if $\theta'_I = \vartheta$ and $\Delta\theta_I = \vartheta - \theta_I$, where ϑ is the orientation of a feature in the environment (e.g. one edge of the table). If the features in the workspace of the imitator are the same as the model's, then $\vartheta \equiv \theta'_M$ and this metric becomes equivalent to the *orientation effect metric*.

8.2 Combinations of Effect Metrics

To evaluate both the movement and the orientation of an object, both metric types must be used. To match the observed effect, the (corresponding) object needs to be moved on the workspace according to the displacement given by the displacement effect metric and rotated according to the angular effect metric used.
A weighted combination of more than one displacement metric can also be used, by averaging the displacement vectors that minimise each metric. For example, if $\Delta X_i = \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix}$ is the displacement that minimises a displacement effect metric i, and ω_1, ... , ω_n are the weights of the n displacement effect metrics to be combined, the displacement that minimizes this composite metric is then given by $\Delta X = \begin{bmatrix} |\Delta X| \times cos(\phi) \\ |\Delta X| \times sin(\phi) \end{bmatrix}$, where $|\Delta X| = \omega_1 \times \sqrt{\Delta {x_1}^2 + \Delta {y_1}^2} + ... + \omega_n \times \sqrt{\Delta {x_n}^2 + \Delta {y_n}^2}$ and $\phi = \omega_1 \times tan^{-1}\left(\frac{\Delta y_1}{\Delta x_1}\right) + ... + \omega_n \times tan^{-1}\left(\frac{\Delta y_n}{\Delta x_n}\right)$.

9 Experimental Setup

9.1 Demonstrated Task

In the work described in this report, the demonstrated task consists of three block objects (colored red, green and blue) arranged in a 2D workspace surface by a human who acts as the demonstrator. The workspace is a square grid 50 cm by 50 cm, and the sizes of the objects are: 10 cm by 8 cm (red) and 8 cm by 5 cm (green and blue).
The current work focuses on the *effects* aspect of the demonstrated behaviour, so only the position and orientation of the objects as they are manipulated by the demonstrator are captured, omitting the demonstrator's actions (arm movements) and states (body posture). Using the Polhemus LIBERTY^{TM} motion capture system (described in section 7.2.1), a sensor is attached on top of each object, giving the position of the object's center and also the object's orientation relative to the origin.
In ongoing work, three (or more) additional sensors will be used, one attached to the human torso and one at each hand, providing additional information about the demonstrator's states and actions. Taking into account the *states* aspect would help the JABBERWOCKY system solve possible ambiguities when producing the corresponding actions for imitation. For example, a humanoid robot imitator, considering the states of the demonstrator would obtain possibly useful information e.g. which hand to use (left or right) to reach an object from its current configuration, based on the choice of hand used by the demonstrator[12].
In section 10, the JABBERWOCKY system is evaluated for combinations of effect metrics on two target platforms when presented with human demonstrated manipulations of objects.

9.2 Realization of the *What to Imitate* Module

In the current implementation of the JABBERWOCKY system the metrics and the sub-goal granularity are given, instead of discovered by the imitator agents based on the observed demonstrated task. The *what to imitate* module provides a choice of metrics and granularity based on the task and context of the demonstration, although there might not always be a unique, "correct", choice. Here, the various possible metrics and granularity have been selected in advance, and as will be shown, depending on the choice, the character of the resulting matched *effects* can be very different.

9.2.1 Sub-goal Granularity

For each demonstration the system captures a total of up to 1000 frames[13], each containing the position and orientation of the objects. The duration of the task will be usually less than that, so duplicate frames (where no motion or rotation occurs) must be removed, and the critical points in the object trajectories found. Going through the frames and looking at position and orientation

[12] Assuming that the demonstrator uses her/his embodiment in a meaningful way and does not complicate the task unnecessarily.
[13] The total duration of the capture will be approximately 10-15 sec.

data for each object, a critical point for an object is a frame where the direction of the motion or the orientation of that object changes more that a certain threshold. The critical points will usually occur at different frames for each object during the demonstration. After the critical points for each object are found, they are 'synchronized' so that if one or more objects has a critical point at a certain frame, then this frame is also a critical point for the rest of the objects. The displacement and rotational information contained in the critical point frames discovered in this way will be used by the JABBERWOCKY system as the sub-goal granularity for imitating the task. By changing the thresholds (one for the direction and one for the rotation) we can get different levels of granularity ranging from fine to coarse, depending on how many critical points are generated.
This is a very simplistic, yet sufficient for our initial experiments, way to define the sub-goal granularity of the demonstration in a generic way. In the future, other methods resulting from the research in WP4.1 will replace the way the sub-goals are extracted from a captured demonstration.

9.2.2 Metrics

Combinations of *displacement* and *angular* effect metrics (defined in section 8) are used together with the sub-goal granularity to define a correspondence problem class (see table 2 in section 4.1) that an imitator must solve in order to imitate successfully.
The choice of *effect* metrics to use is currently given to the JABBERWOCKY system by a human user, and not discovered by the imitator agents. In the future, algorithms and methods developed in WP4.2 will be integrated, allowing for automatic discovery of appropriate metrics for each demonstrated task.

9.3 Realization of the *How to Imitate* Module

The *how to imitate* module considers the given *effect* metrics and sub-goal granularity, together with the (possible dissimilar) initial configuration of the objects in the imitator's workspace (also given) to produce initially an embodiment-independent correspondence solution (since only the *effects* aspect are considered).
To discover this correspondence, the JABBERWOCKY system currently uses a simple simulation of the 2D workspace that can handle various 'block' objects moving and rotating around, accounting for object collisions and workspace confines. This simulation can replay the captured model data at a given granularity, displaying the trajectory and orientation of the objects as they move and rotate on the workspace, from the initial configuration to the final captured frame. In parallel, starting from a different initial configuration of the same (or different) corresponding objects on the imitator's workspace, the simulation produces a sequence of changes to displacement and rotation for each object, that minimize the given effect metrics.
For example if the effect metric used is the *relative displacement* effect metric, and the demonstrator moved an object 10 cm to the right, then in order to minimize the metric, the corresponding object in the imitator's workspace must be also moved 10 cm to the right. But some displacements or rotations, although minimizing the metric, might be invalid because the path or final

position is occupied by other objects or agents, e.g. if the corresponding object is less than 10 cm away from the right edge of the workspace (because the initial position was different), the entire move cannot be performed. The *how to imitate* module will then have to discover an alternative way in the given context (including other agents, static or dynamic obstacles) to achieve the same *effects* according to the metric. In this case it might be acceptable to move the object up to the right edge and then continue the rest of the imitative behaviour. In another context, it might be preferable not to move the object at all. This contextual information should be ideally provided by the *what to imitate* module, based on observations of the currently demonstrated task and not pre-defined. In the current JABBERWOCKY implementation, the system attempts to move (or rotate) the objects until they reach an obstacle (based on simple 2D object collision detection), and then stop, instead of considering another path to reach the position (and/or achieve the orientation) that minimizes (if possible) the metric used.
In each case, very different correspondences may result from trying to match different aspects of the demonstration. In the experiments presented in this report (see sections 10.1 and 10.2), the use of different *effect* metrics is shown to result in qualitatively different imitative behaviours. The confines of the workspace and the obstruction of the path by other objects can also influence the imitative behaviour.
To imitate and achieve similar *effects* as the model, an imitator agent will have to adopt this (largely) embodiment-independent correspondence solution to move and rotate the objects, using a generated sequence of action command instructions. These action commands will be targeted to multiple imitator platforms, taking into account the embodiment constraints and restrictions of imitator embodiments.

9.4 Targeted Imitator Platforms

After generating the embodiment-independent effect correspondence solutions for the corresponding objects in the imitator's workspace, the JABBERWOCKY system will convert each of them to a sequence of action commands to be executed by multiple imitator platforms.
Two such targeted platforms are used in the current realization of the system, both implemented using the Webots^{TM} robot simulation software.

9.4.1 Three Robots As Objects

In the first imitator platform, the imitator's workspace contains no objects. Instead, the imitator is 'embodied' as three mobile robots, each corresponding to one of the objects manipulated by the demonstrator. Each robot is square 4cm by 4cm (so in this case, besides dissimilar demonstrator-imitator embodiments, there is also dissimilar object correspondence, mapping the objects to mobile robots). The robots can follow the individual trajectories of the objects as arranged by the demonstrator, but cannot match the orientation (while moving) because they are differential wheel robots. Therefore the *angular* effect aspect will be ignored when they imitate, matching only the *displacement* effect aspect.
In the simulation, as the robots move around the workspace, they leave behind a colored trail (of same color as themselves and their corresponding objects) to help visualize the imitated

trajectories (see figure 3).
To convert the effect correspondences into a sequence of action commands for this target imitator platform, each robot is given a sequence of way-points, depending on its corresponding object. For each of these way-points, the robot must use its differential wheel embodiment to move in a straight line up to that position in the workspace, and after reaching the target position, move on to the next.
All three objects will have the same number of critical points (see section 9.2, above) and as a result the number of way-points will be the same[14] for each robot. But the time it takes each robot to achieve its current target position will not be the same, depending on the distance it has to travel. This can result in a 'synchronization' problem; the robots may either go through their sequence of way-points ignoring what the other robots are doing (unless they are obstructing their way) or the robots can try to synchronize their imitative behaviour by waiting until all three have reached their current target positions before moving to the next one. In the experiments presented in this report, the robots are synchronized.

9.4.2 Manipulator and Three Objects

In the second targeted imitator platform, the imitator's workspace contains three objects, of the same size and color as the corresponding objects in the demonstrator's workspace. The imitator is embodied as a single arm manipulator, positioned above the workspace and able to pick-up, move and rotate the three objects. This embodiment, although dissimilar to the one of the human demonstrator, is nevertheless able to match both *displacement* and *angular* effect aspects of the demonstration.
As the objects are moved (and rotated) around the workspace by the manipulator in the simulation, they leave behind a colored trail (of same color as themselves) to help visualize the imitated trajectories. The manipulator is shown as a vertical yellow cylinder mounted at the end of a bar positioned above the workspace (see figure 4).
The action sequence produced by the JABBERWOCKY system will consist of a continuous path[15], with way-points above the current (and future) positions of the objects. When the manipulator is above an object that must be moved, the manipulator picks it up, then moves (together with the object) to the target position and places the object down (while also, if required, rotating it), before continuing to the next object.
To match the *effects* at each critical point, the order the manipulator approaches the objects is the same (red object, green, blue). If no displacement or rotation is required for an object during each of these turns, that object is ignored, simplifying the manipulator's path. If more than one objects were moved at the same time during the demonstration (by the human using both hands) resulting in a sequence of critical points, this imitator will only be able to match this by moving (and/or rotating) each object in turn at each critical point before continuing to the next, as it has a single manipulator.

[14] Although there probably will be duplicates within the sequence, indicating that the object has to be still.
[15] For this platform, the path will be closed, starting and ending at a position at the upper left corner of the workspace.

10 Evaluation of Experiments

Two experimental runs (each using a different captured demonstration of an object arrangement task by a human) will be discussed in this report. In each run, after finding the critical points, a combination of different effect metrics was used (either the same *effect* metric for all the objects or a different one for each object) together with dissimilar initial configuration of the objects (for the imitator's workspace) to define several different correspondence problems. The JABBERWOCKY SYSTEM produced corresponding action commands that minimized the effect metrics in each case, targeted at the two imitator platforms described above in section 9.4. Using these action commands, the imitator agents were able to imitate the appropriate effect aspects of the demonstration and the resulting imitative behaviours were captured from the Webots simulation. Two human demonstrations are discussed here and imitated on both target platforms with various effect metrics guiding what aspects to match. (The second experimental run is shown in the demonstration video - see Appendix A).
The results shown provide examples of the JABBERWOCKY system producing solutions to the *correspondence problem* for multiple targeted platforms and also illustrate the qualitative differences resulting from using different *effect* metrics.

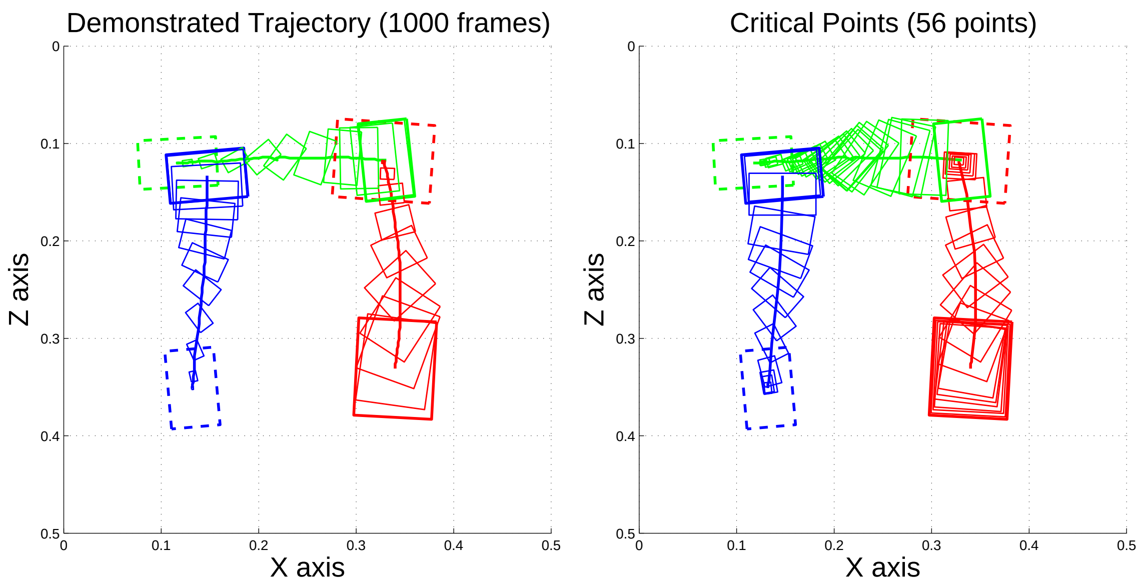

Figure 7: **First captured demonstration (left) and the extracted critical points (right).** The colors (red, green and blue) indicate the three different objects. The dotted outlines indicate the initial position and orientation of the objects, while the solid thick outline the final. For the demonstration data, the intermediate object's position and orientation is shown with solid thin outlines, linearly scaled (at intervals equal to one tenth of the overall trajectory only, for clarity) to indicate the direction of the movement. For the critical points, each object's position and orientation is shown at every critical point, again linearly scaled.

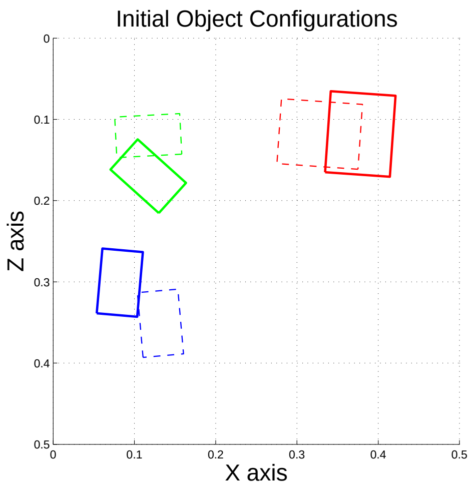

Figure 8: **An example of dissimilar initial object positions.** The dotted outlines indicate the initial position and orientation of the objects in the demonstrator's workspace (from the demonstration shown in figure 7, left) and the solid outlines the (dissimilar) initial configuration of the objects in the imitator's workspace.

The first human demonstration is visualized in figure 7. A human user moved (and rotated) the red object downwards, then the green object to the right and finally the blue object upwards in his workspace. The captured data are shown in figure 7, left. Using the method described in section 9.2.1, the critical points were found and are shown in figure 7, right.
The objects in the imitator's workspace need not have the same initial position and orientation. Figure 8 shows the initial configuration in the imitator's workspace (solid outlines) compared to the initial configuration in the demonstrator's workspace (dotted outlines). Relative to the initial object configuration in the demonstrator's workspace, the red object is translated to the right and rotated by 90°, the green object is translated downwards and rotated by 45°, and the blue object is translated sideways up and to the left, also rotated slightly.
Using the *relative displacement* effect metric, the JABBERWOCKY system produces corresponding action commands that are visualized in figure 9 (left) for the *three robots as objects* imitator platform. Each robot (the different colors indicate the object correspondence) must move along the way-points (indicated by the dots) from its initial position (dotted outline) to the final position (solid outline). The initial and final positions are visualized in the figures as circles (of relative size to the robots) to show that the orientation of the robots is not considered. Since the initial positions are different, but the matched displacements must be the same (due to the metric), the robots should move with the resulting paths displaced accordingly (red to the right, green downwards and blue to the left and up). The corresponding actions are performed by the robots in the Webots simulation environment and the resulting imitative behaviour is captured and shown in figure 9 (right).
Using the *relative displacement* and *rotation* effect metrics, the JABBERWOCKY system also produces the corresponding action commands shown in figure 10 (left) for the *manipulator and*

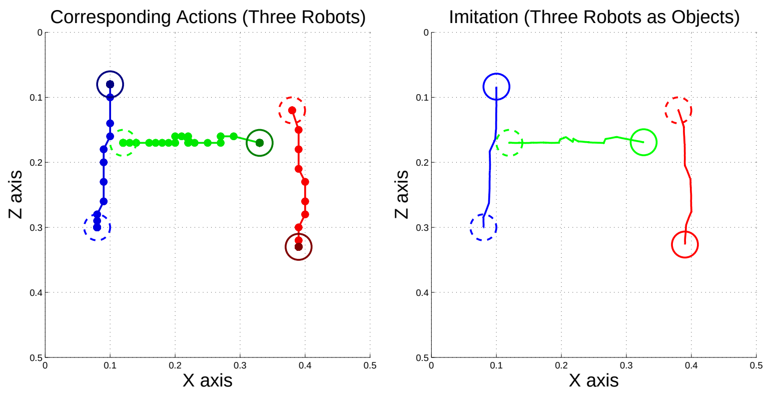

Figure 9: **An example of corresponding action commands for the *three robots as objects* imitator platform (left) and the resulting imitative behaviour (right).** Using the critical points shown in figure 7, starting from the initial positions shown in figure 8, and minimizing the *relative displacement* effect metric, each of the robots must move along the way-points shown (left). The initial (dotted outline) and final (solid outline) positions are shown as circles, indicating that the orientation of the robots is not considered (the actual robots are square, but of equivalent size). Each way-point is indicated as a dot. The robots then perform an imitative behaviour (in Webots) and the captured results from the simulation are shown in the right plot.

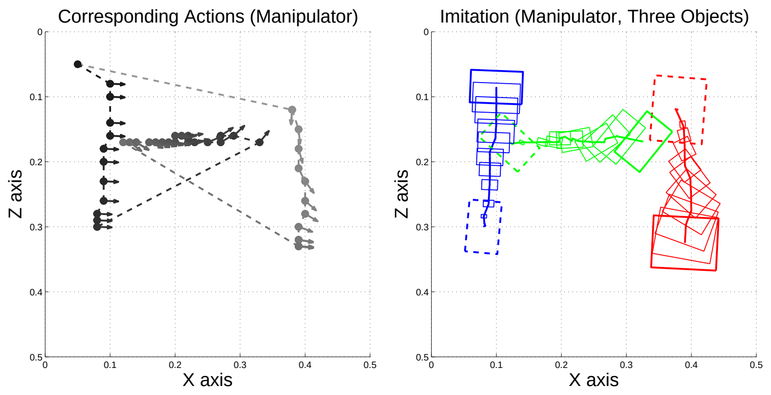

Figure 10: **An example of corresponding action commands for the *manipulator and three objects* imitator platform (left) and the resulting imitative behaviour (right).** Using the critical points shown in figure 7, starting from the initial positions shown in figure 8, and minimizing the *relative displacement* and *rotation* effect metrics, the manipulator must follow the continuous closed path (starting and ending at the left top corner of the workspace) shown as a dotted line (left). The line in drawn using a gray to black color gradient to indicate the direction of the path. When reaching an object, the orientation that the object must be rotated to is shown by a small arrow. The manipulator then performs an imitative behaviour (in Webots) and the captured results from the simulation are shown in the right plot.

three objects imitator platform. The path of the manipulator is shown as a dotted line (drawn using a gray to black color gradient to indicate the direction), with way-points indicated as dots. Since this platform can also match the *angular* effect aspects, the orientation that the objects should have at each way-point is indicated by an arrow. The manipulator starts from, and returns back to a position in the top left corner after all the *effects* have been achieved. The object displacements will be similar to the ones shown in figure 9, with the paths translated according to the dissimilar initial object positions. The objects will rotate by the same amount as in the captured demonstration (due to the metric used) but will not match the same orientation. For example, the red object in the imitator's workspace starts at a right angle relative to its initial orientation in the demonstrator's workspace. As a result, the final (and intermediate) orientation during the imitative behaviour will also be at a right angle relative to the captured demonstration.

10.1 First Experimental Run: Same Effect Metrics Used for All Objects

To illustrate the different *effects* resulting from the different correspondence problems defined by the choice of metrics, we produced action commands using the JABBERWOCKY system and we captured the imitative behaviours simulated in the two targeted platforms.

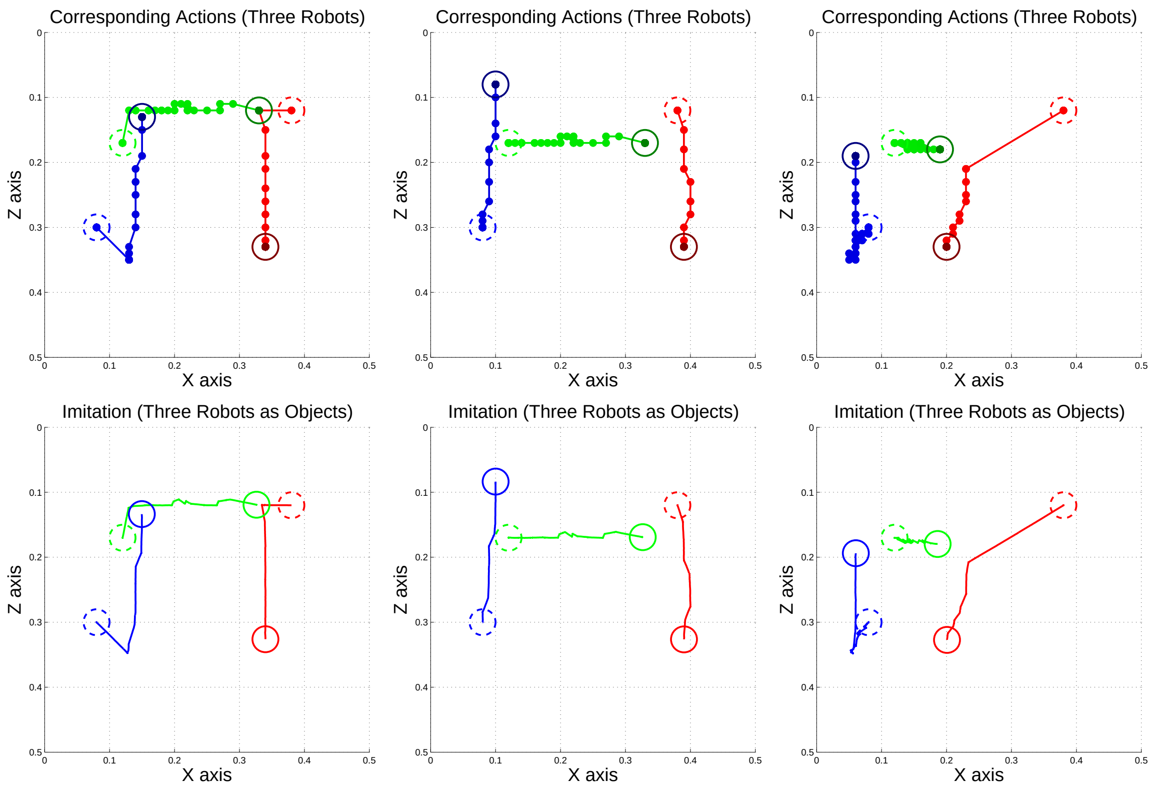

Figure 11: **Corresponding action commands for the *three robots as objects* imitator platform (top row) and the resulting imitative behaviours (bottom row).** Using the critical points shown in figure 7, starting from the initial positions shown in figure 8, and minimizing the *absolute displacement* (left column), *relative displacement* (middle column) and *relative position* (right column) effect metrics, each of the robots must move along the way-points shown (top row). In each case, the robots then perform an imitative behaviour (in Webots) and the captured results from the simulation are shown in the bottom row.

The corresponding action commands and the resulting effects from using each of the different *displacement* effect metrics (in each case the same metric is used for all the objects) are shown in figure 11, for the *three robots as objects* imitator platform. The captured demonstration, critical points and initial imitator's object configuration are shown in figures 7 and 8. The corresponding *effects* resulting from using the *relative displacement* effect metric (figure 11, middle) were discussed in the previous subsection. Using the *absolute displacement* effect metric (figure 11, left), the robots must make an initial adjustment of their positions to reach the corresponding object's initial position in the demonstrator's workspace. For the rest of the imitative behaviour, their positions (and also their displacements) match precisely those of the corresponding objects from the demonstration. But using the *relative position* effect metric (figure 11, right), results in qualitatively very different trajectories to the ones captured. Nevertheless, the robots move in such a way to match the relative position effect aspect as the corresponding objects in the demonstration. The reason for this trajectory distortion is the dissimilar initial object configuration.

The corresponding action commands and the effects resulting from using combinations of each of the different *displacement* and *angular* effect metrics (in each case the same metrics are used for all the objects) are shown in figures 12 and 13 respectively, for the *manipulator and three objects* imitator platform. The captured demonstration, critical points and initial imitator's object configuration are shown in figures 7 and 8. The corresponding trajectories resulting from using each of the *displacement* effect metrics are similar the ones shown and discussed above for the *three robots as objects*, but with the added aspect of the object's orientation. When the *rotation* effect metric is combined with the *displacement* effect metrics (figures 12 and 13, middle columns), the orientation of the objects will be offset from the initial object orientation as seen in figure 8. When the *orientation* effect metric is combined with the *displacement* effect metrics (figures 12 and 13, left columns), the orientation of the objects will be initially corrected to align with the object orientation as seen in figure 8, and for the rest of the imitative behaviour it will match orientation captured in the demonstration. If no *angular* effect metric is used (figures 12 and 13, right columns), then the objects will not rotate at all during the imitative behaviour, and will conserve their initial orientation.

If no *displacement* effect metric is used, then the objects will simply stay still in their initial positions or (if an *angular* effect metric is used) spin around.

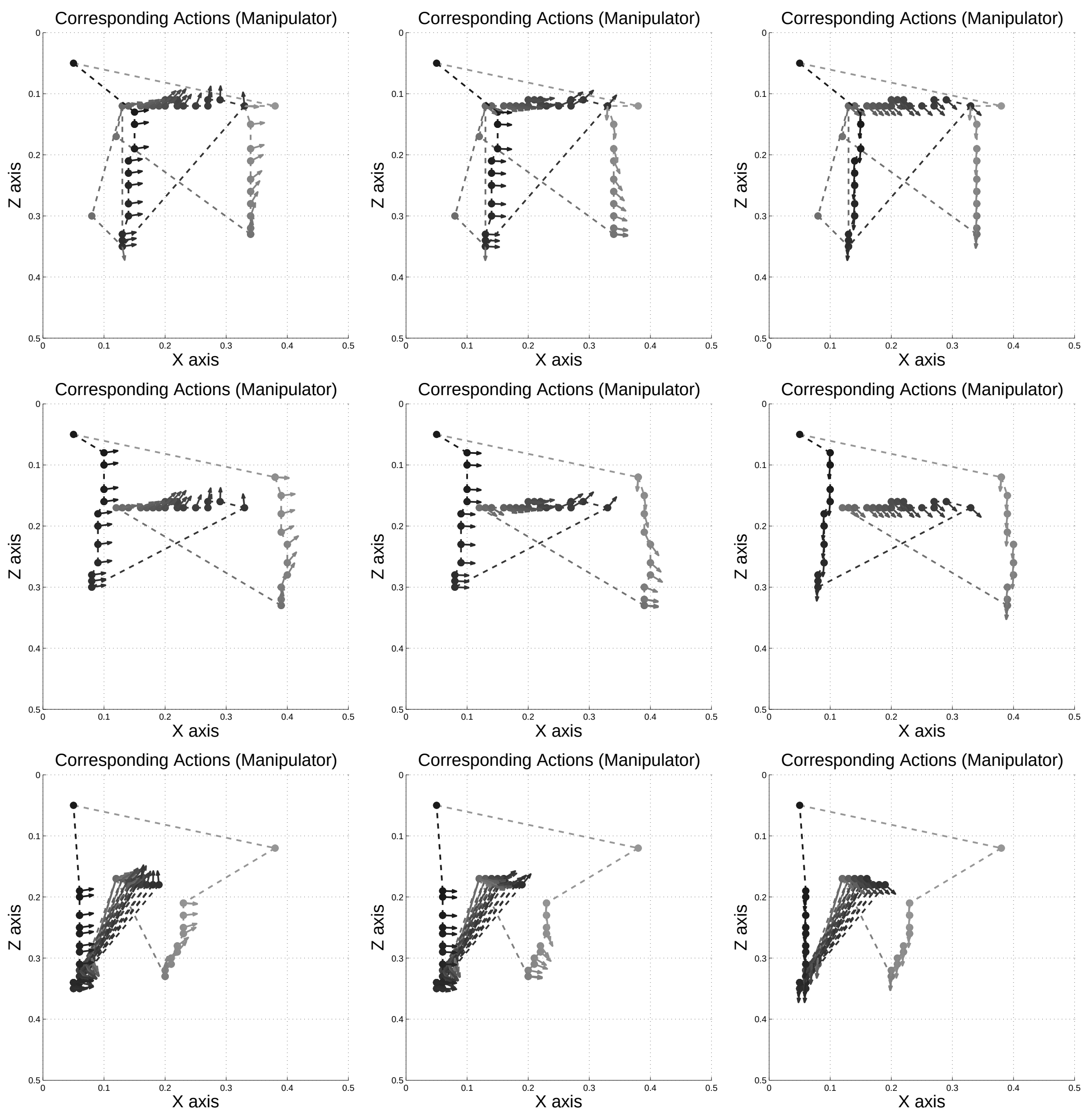

Figure 12: **Corresponding action commands for the *manipulator and three objects* imitator platform.** Using the critical points shown in figure 7, starting from the initial positions shown in figure 8, and minimizing the *absolute displacement* (top row), *relative displacement* (middle row), *relative position* (bottom row), *orientation* (right column) and *rotation* (middle column) effect metrics, the manipulator in each case must follow the continuous closed path, moving and rotating the objects accordingly. Note that in the left column, no angular effect metrics are used, only displacement effect metrics.

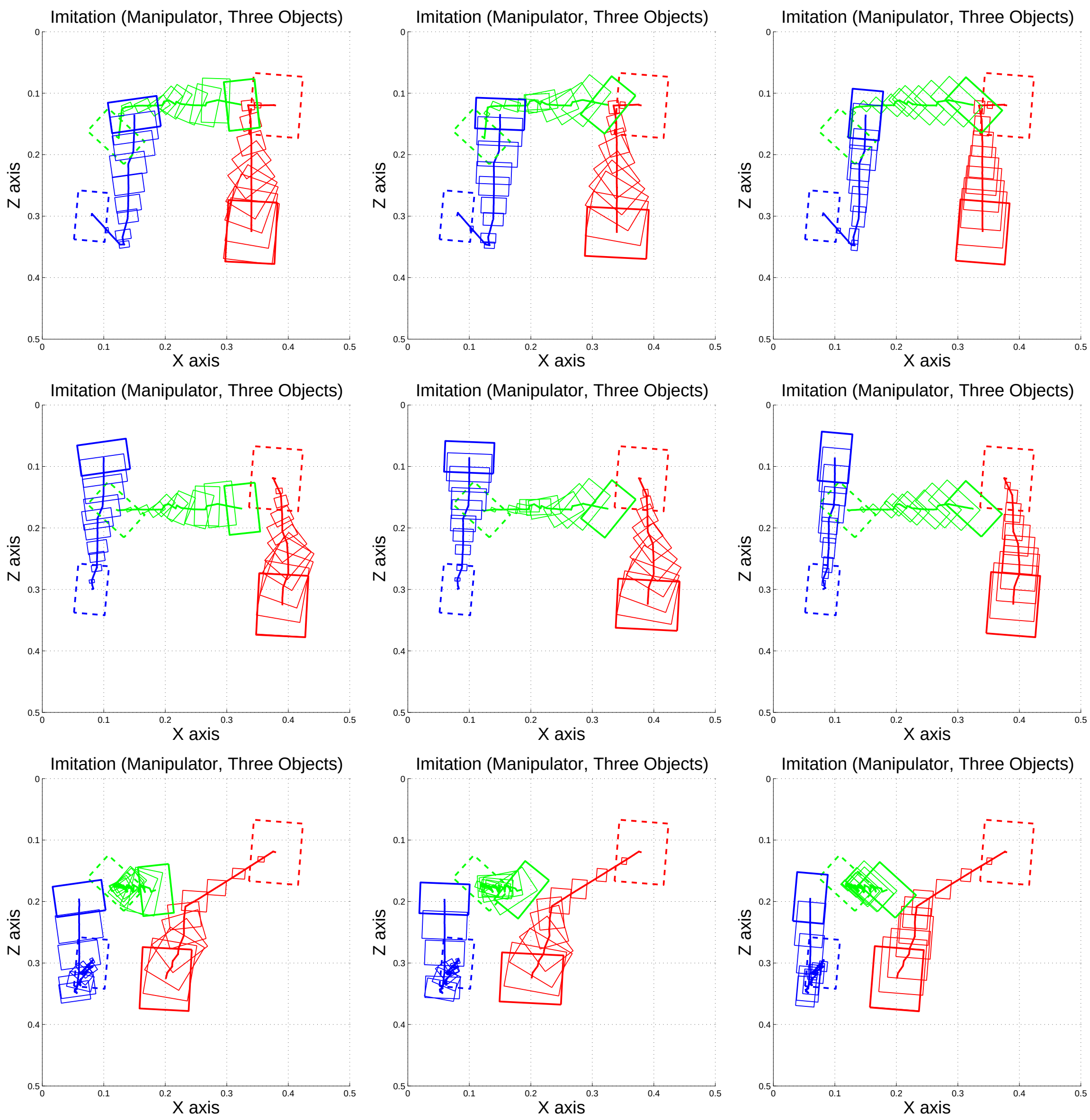

Figure 13: **Resulting imitative behaviours using the corresponding action commands shown in figure 12 for the *manipulator and three objects* imitator platform.** Using the critical points shown in figure 7, starting from the initial positions shown in figure 8, and minimizing the *absolute displacement* (top row), *relative displacement* (middle row), *relative position* (bottom row), *orientation* (right column) and *rotation* (middle column) effect metrics, the manipulator in each case performs an imitative behaviour (in Webots) using the corresponding action commands shown in figure 12 and the captured results from the simulation are shown. Note that in the left column, no angular effect metrics are used, only displacement effect metrics.

10.2 Second Experimental Run: Combination of Different Effect Metrics for Each Object

In contrast to the experimental run presented and discussed in the previous subsection, where the same metrics were used to match the *effects* of each object, in this subsection a different metric will be used for each object.
This experimental run is also shown in the demonstration video (see Appendix A).

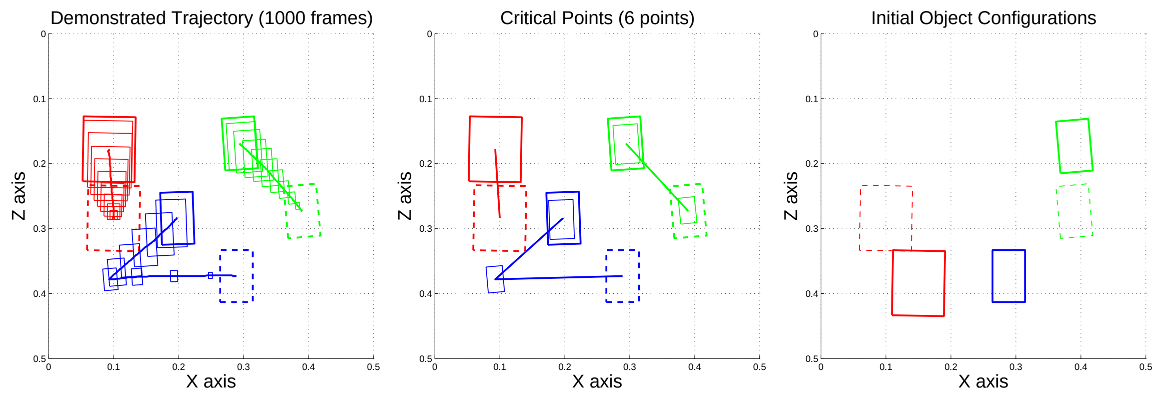

Figure 14: **Second captured demonstration (left), extracted critical points (middle) and initial object configurations (right).**

The demonstration that is used in this experimental run is shown in figure 14 (left). The human demonstrator first moves the red object upwards, then the green object sideways to the right and finally the blue object in such a way that in the last two positions, it forms first a *right* and then an *isosceles* triangle (approximately) with the other two objects. The critical points are found and shown in figure 14 (middle). The granularity is more coarse in this case compared to the first demonstration, as a larger threshold (for direction change) was used to define the critical points. The initial position of the objects in the imitator's workspace is shown in figure 14 (right). The red object is offset sideways down and to the right, and the green object upwards. The blue object occupies the same initial position.
Three different combinations of *displacement* effect metrics are used, shown in figures 15 and 16. Since the objects were not rotated by the demonstrator, no *angular* effect metrics were used. First, the red object matches the *absolute displacement* effect metric, the green object the *relative displacement* effect metric and the blue object the *relative position* effect metric (shown in figures 15 and 16, left columns).

- The final position of the red object (that performs only a single move) will be the same as one of the corresponding red object in the demonstrator's workspace (due to the absolute displacement metric).
- The green object will move sideways to the left, but the trajectory is translated upwards (due to the relative displacement metric).

- The blue object will move as to form first a *right* (when the red and green objects stop moving) and finally an *isosceles* triangle (due to the relative position metric). Because of the other two objects movements in the early stage of the imitative behaviour, the blue object will also move (when at the same stage the corresponding blue object remain still in the demonstration), trying to preserve the relative positions.

Secondly, the red object matches the *relative displacement* effect metric, the green object the *relative displacement* effect metric and the blue object the *relative position* effect metric (shown in figures 15 and 16, middle columns).

- The red object will move a shorter distance, matching the trajectory (translated) of the corresponding red object in the demonstrator's workspace (due to the relative displacement metric).
- The green object will move as in the previous case.
- The blue object will again move to conserve the relative positions and then form the two triangles, but now due to the position of the red object (different from to the previous case) the trajectory will be different.

Thirdly, the red object matches the *relative position* effect metric, the green object the *relative displacement* effect metric and the blue object the *relative displacement* effect metric (shown in figures 15 and 16, right columns).

- The red object will move as to conserve its relative position to the other two objects, resulting in more than one moves (due to the relative position metric).
- The green object will again move as in the two previous cases.
- The blue object, starting from the same initial position, will move using the same trajectory as the corresponding blue object in the demonstrator's workspace (due to the relative displacement metric).

As a result of the particular combination of metrics and the order of the object displacements, the same *right* and *isosceles* triangles will not be formed in the last case as in the first two. So, the choice of a metric will lead to matching of aspects associated with that particular metric, but may or may not result in matching of other aspects.

10.3 Summary Evaluation

The experiments show the diverse character of different successful imitative behaviours optimized to match particular aspects of the effects of demonstrated human manipulation of objects. Aspects captured by metrics for *absolute displacement*, *relative displacement*, *relative position*, *rotation* and *orientation* could all successfully be matched. The results illustrate the multi-platform targetability of the JABBERWOCKY system to map human demonstrated manipulations to matching robotics manipulations (in simulation), generalizing to different initial object configurations.

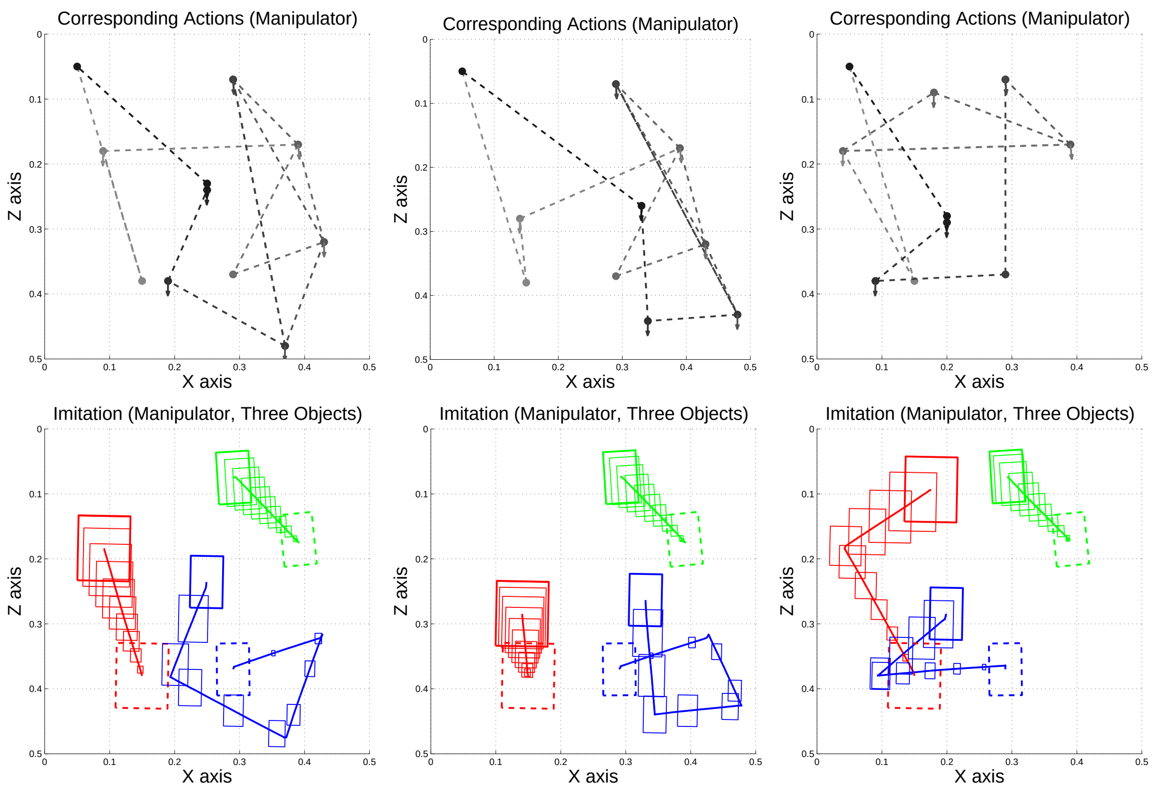

Figure 15: **Corresponding action commands for the *manipulator and three objects* imitator platform (top row) and the resulting imitative behaviours (bottom row).** Using the critical points shown in figure 14 (middle) and starting from the initial positions shown in figure 14 (right), a combination of displacement effect metrics (different for each object, see text for details) are minimized. In each case, the manipulator then performs an imitative behaviour (in Webots) and the captured results from the simulation are shown on the bottom row.

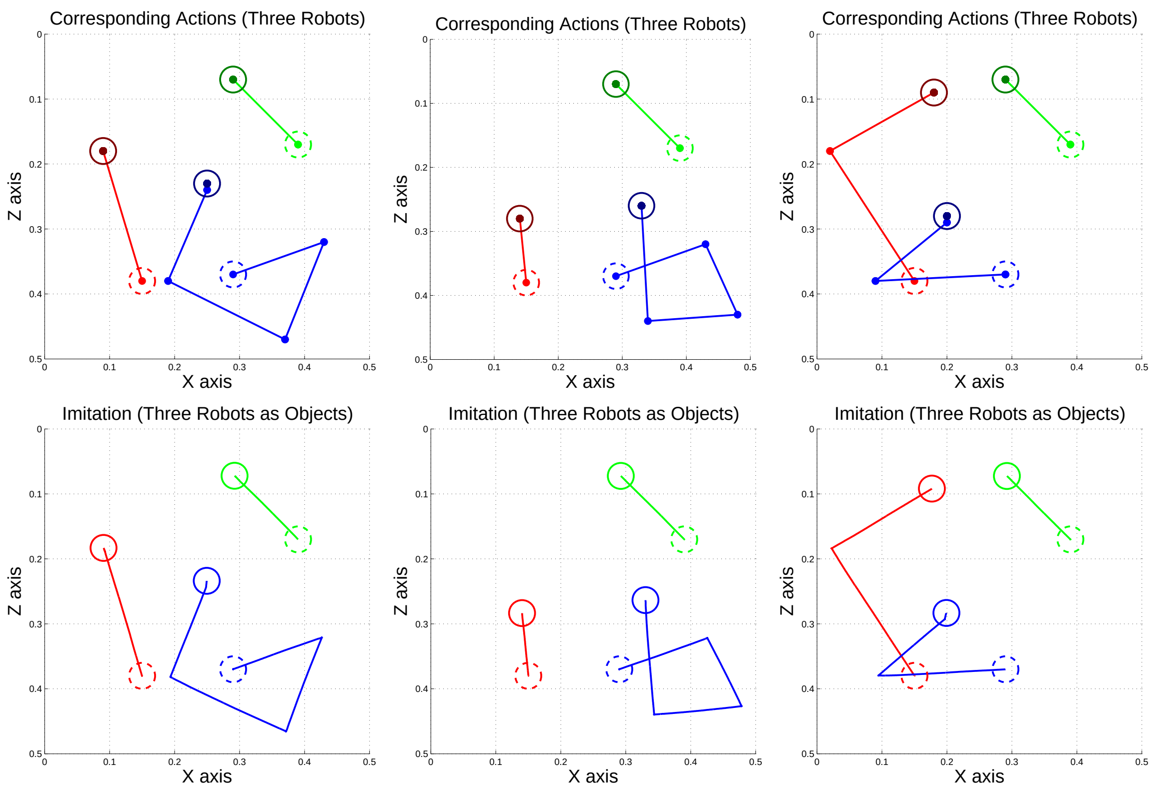

Figure 16: **Corresponding action commands for the *three robots as objects* imitator platform (top row) and the resulting imitative behaviours (bottom row).** Using the critical points shown in figure 14 (middle) and starting from the initial positions shown in figure 14 (right), a combination of displacement effect metrics (different for each object, see text for details) are minimized. In each case, the robots then perform an imitative behaviour (in Webots) and the captured results from the simulation are shown on the bottom row.

11 Summary, Conclusions and Future Work

11.1 Summary

Qualitatively different kinds of social learning result from matching different combinations of matching *actions*, *states* and *effects* at different levels of granularity.
Metrics capture similarity of robot- and human-achieved *effects* on the environment, including e.g. sequences in object manipulation; *states* of the body; or effector *actions*; or dynamically varying combinations of these aspects, in order to match demonstrator behaviour (where the latter two aspects depend more strongly on details of robot embodiment than the former). Metrics do this by providing formal measures for the degree of matching between two different effects, states or actions. A choice of metrics and the granularity determines *what to imitate*, and guides machine learning and solution of the *correspondence problem*, i.e. allowing an imitator robot to match appropriate features of the behaviour of (e.g. human) demonstrator in order to socially acquire skills or learn how to perform demonstrated tasks.
This report presents an initial characterization of a *space of effect metrics* to use in guiding various types of robotic imitation and social learning from human demonstrators.
A multitargetable system (JABBERWOCKY) has been developed that can be used with captured motion data from a human demonstrator to solve the corresponding problem for an imitator agent (in simulation), according to given effect metrics and granularity. The system generates high-level, largely embodiment-independent (as currently only the *effects* aspect is considered) solutions to the correspondence problem, which are then adapted so a particular target imitator can map sequences of observed effects of the demonstrator on objects to its own repertoire of actions as constrained by its own embodiment and by context (possibly different initial configurations of environment and manipulated objects). This work is validated in simulation experiments (including a demo - see Appendix A) and documented in this report.

11.2 Discussion and Conclusions

The experiments with the two target platforms show that depending on which metric is used to match aspects of demonstrated behaviour, different successful imitative behaviours result.
The experimental results illustrate the multi-platform targetability of the JABBERWOCKY system to map human demonstrated manipulations to matching robotics manipulations (in simulation), generalizing to different initial object configurations.
Using the captured demonstrated behaviour and given a fixed effect metric and granularity, it is clear that the system will be able to generalize human-demonstrated arrangement of objects across different starting object configurations. For example, for any initial configuration of objects in a table place-setting, using a previously demonstrated arrangement of the objects and the *absolute displacement* and *orientation* metrics, JABBERWOCKY would be able to generate appropriate actions that could be used by any imitator platform (capable of moving and rotating objects) to reproduce the observed arrangement.

11.3 Future Work

Building on results from WP4.1 and WP4.2, research on the characterization of *spaces of metrics* and their exploitation will be advanced for use in guiding various types of robotic imitation and social learning from human demonstrators.

Part of the future work on metrics will further document the space of *effect* metrics for largely embodiment-independent imitation. This will provide a more comprehensive characterization of the scope of what it is possible to imitate, guiding collaborators in the design of task and skill learning systems that employ robot programming by demonstration with a broadened, yet constrained, scope and algorithmic and software tools to compute the metrics and generate behavioural recipes that can be targeted to multiple, dissimilar imitator platforms.

Multi-platform targeted solutions for the correspondence problem (*how to imitate*) given metrics and sub-goal structures will be developed further building on WP4.2 results, by extending from implemented simulation systems towards robotic target platforms.

Work on the correspondence problem using the JABBERWOCKY system will allow an imitator agent to learn how to reproduce gestures (states, actions) and also simple manipulation of objects (effects).

Work on the characterization of the space of metrics will aid the design of systems that solve the *what to imitate* (i.e. learning the important features of a demonstrated task) by providing well-characterized spaces of rigorous metrics to capture essential task features.

The social and physical context of imitation will also be treated in forthcoming work at increasing levels of complexity, taking into account e.g. selection of a model, static and dynamic observation, as well as timing constraints and turn-taking interaction.

A Appendix: Notes on the Demonstration Video

An *.avi* file showing the experiments described in section 10.2 can be downloaded from `http://homepages.feis.herts.ac.uk/~comqaa1/cogniron-D421.avi`.
A human is shown arranging three objects (a red, a green and a blue block) on a grid surface workspace. The *effects* of this demonstration are captured (see figure 14, left) and the critical points are found (see figure 14, middle). Starting from a dissimilar initial configuration of the objects in the imitator's workspace (see figure 14, right), corresponding actions are generated by the JABBERWOCKY system for each of the two target imitator platforms (described in sections 9.4.1 and 9.4.2), for three different combinations of effect *displacement* metrics (see figures 16, left and 15, left). The imitator platforms are then shown (simulated in Webots^{TM}) to imitate the demonstration using the corresponding actions produced by the JABBERWOCKY system, starting from their dissimilar initial configuration and matching each effect metric combinations (see figures 16, right and 15, right).
The video is encoded using DivX 5.2.1 for Windows. A free version of the codec can be downloaded from `http://www.divx.com/divx/download/`.

References

[Alissandrakis, 2003] Alissandrakis, A. (2003). *Imitation and Solving the Correspondence Problem for Dissimilar Embodiments - A Generic Framework*. PhD thesis, University of Hertfordshire.

[Alissandrakis et al., 2002] Alissandrakis, A., Nehaniv, C. L., and Dautenhahn, K. (2002). Imitation with ALICE: Learning to imitate corresponding actions across dissimilar embodiments. *IEEE Trans. Systems, Man & Cybernetics: Part A*, 32(4):482–496.

[Alissandrakis et al., 2003a] Alissandrakis, A., Nehaniv, C. L., and Dautenhahn, K. (2003a). Solving the correspondence problem between dissimilarly embodied robotic arms using the ALICE imitation mechanism. In *Proc. Second International Symposium on Imitation in Animals and Artifacts – Aberystwyth, Wales, 7-11 April 2003*, pages 79–92. Society for the Study of Artificial Intelligence and Simulation of Behaviour.

[Alissandrakis et al., 2003b] Alissandrakis, A., Nehaniv, C. L., and Dautenhahn, K. (2003b). Synchrony and perception in robotic imitation across embodiments. In *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '03)*, pages 923–930.

[Alissandrakis et al., 2004] Alissandrakis, A., Nehaniv, C. L., and Dautenhahn, K. (2004). Towards robot cultures? – Learning to imitate in a robotic arm test-bed with dissimilar embodied agents. *Interaction Studies: Social Behaviour and Communication in Biological and Artificial Systems*, 5(1):3–44.

[Billard et al., 2004] Billard, A., Epars, Y., Calinon, S., Cheng, G., and Schaal, S. (2004). Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, 47:2-3.

[Billard and Schaal, 2002] Billard, A. and Schaal, S. (2002). Computational elements of robot learning by imitation. In *American Mathematical Society Central Meeting, Madison, October 12-13, 2002*.

[Byrne and Russon, 1998] Byrne, R. W. and Russon, A. E. (1998). Learning by imitation: a hierarchical approach. *Behavioral and Brain Sciences*, 21:667–709.

[Calinon and Billard, 2004] Calinon, S. and Billard, A. (2004). Stochastic gesture production and recognition model for a humanoid robot. In *IEEE/RSJ Intl Conference on Intelligent Robots and Systems (IROS)*.

[Call and Carpenter, 2002] Call, J. and Carpenter, M. (2002). Three sources of information in social learning. In Dautenhahn, K. and Nehaniv, C. L., editors, *Imitation in Animals and Artifacts*. MIT Press.

[Dautenhahn, 1994] Dautenhahn, K. (1994). Trying to imitate – a step towards releasing robots from social isolation. In Gaussier, P. and Nicoud, J.-D., editors, *Proc. From Perception to Action Conference, Lausanne, Switzerland*, pages 290–301. IEEE Computer Society Press.

[Dautenhahn and Nehaniv, 2002] Dautenhahn, K. and Nehaniv, C. L. (2002). An agent-based perspective on imitation. In Dautenhahn, K. and Nehaniv, C. L., editors, *Imitation in Animals and Artifacts*, pages 1–40. MIT Press.

[Kolodner, 1993] Kolodner, I. L. (1993). *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA.

[Kuniyoshi et al., 1994] Kuniyoshi, Y., Inaba, M., and Inoue, H. (1994). Learning by watching: Extracting reusable task knowledge from visual observations of human performance. *IEEE Trans. Robot. Automat.*, 10:799–822.

[Michaud and Matarić, 1999] Michaud, F. and Matarić, M. J. (1999). Representation of behavioral history for learning in nonstationary conditions. *Robot. Auton. Syst.*, 29:187–200.

[Michel, 2004] Michel, O. (2004). Webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42.

[Nehaniv, 1996] Nehaniv, C. L. (1996). From relation to emulation: The covering lemma for transformation semigroups. *Journal of Pure & Applied Algebra*, 107(1):75–87.

[Nehaniv, 2003] Nehaniv, C. L. (2003). Nine billion correspondence problems and some methods for solving them. In *Proc. Second International Symposium on Imitation in Animals and Artifacts – Aberystwyth, Wales, 7-11 April 2003*, pages 93–95. Society for the Study of Artificial Intelligence and Simulation of Behaviour.

[Nehaniv and Dautenhahn, 1998a] Nehaniv, C. L. and Dautenhahn, K. (1998a). Embodiment and memories - algebras of time and history for autobiographic agents. In *Proceedings of 14th European Meeting on Cybernetics and Systems Research EMCSR'98*, pages 651–656.

[Nehaniv and Dautenhahn, 1998b] Nehaniv, C. L. and Dautenhahn, K. (1998b). Mapping between dissimilar bodies: Affordances and the algebraic foundations of imitation. In Demiris, J. and Birk, A., editors, *Proceedings European Workshop on Learning Robots 1998 (EWLR-7), Edinburgh, 20 July 1998*, pages 64–72.

[Nehaniv and Dautenhahn, 2000] Nehaniv, C. L. and Dautenhahn, K. (2000). Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications. In Demiris, J. and Birk, A., editors, *Interdisciplinary Approaches to Robot Learning*, pages 136–161. World Scientific Series in Robotics and Intelligent Systems.

[Nehaniv and Dautenhahn, 2001] Nehaniv, C. L. and Dautenhahn, K. (2001). Like me? - measures of correspondence and imitation. *Cybernetics and Systems*, 32(1-2):11–51.

[Nehaniv and Dautenhahn, 2002] Nehaniv, C. L. and Dautenhahn, K. (2002). The correspondence problem. In Dautenhahn, K. and Nehaniv, C. L., editors, *Imitation in Animals and Artifacts*, pages 41–61. MIT Press.

[Nielsen and Dissanayake, 2003] Nielsen, M. and Dissanayake, C. (2003). Synchronic imitation as pre-linguistic social interaction. In *Proc. Second International Symposium on Imitation in Animals and Artifacts – Aberystwyth, Wales, 7-11 April 2003*, pages 131–137. Society for the Study of Artificial Intelligence and Simulation of Behaviour.

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA.

[Sammut et al., 1992] Sammut, C., Hurst, S., Kedzier, D., and Michie, D. (1992). Learning to fly. In *Proc. Ninth International Conference on Machine Learning*, pages 385–393. Morgan Kaufmann.

[Schaal, 2000] Schaal, S. (2000). The SL simulation and real-time control software package. USC.

[Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

[Whiten, 2002] Whiten, A. (2002). Imitation of sequential and hierarchical structure in action: Experimental studies with children and chimpanzee. In Dautenhahn, K. and Nehaniv, C. L., editors, *Imitation in Animals and Artifacts*, pages 191–210. MIT Press.