Lehrstuhl für Steuerungs- und Regelungstechnik
Technische Universität München
Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss

# Intention Recognition in Dynamic Field Theory

**Laith M. H. Alkurdi**

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

**Doktor-Ingenieurs (Dr.-Ing.)**

genehmigten Dissertation.

Vorsitzender:

Prof. Dr.-Ing. Bernhard Seeber

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Angelika Peer
2. Prof. Dr.-Ing. Sami Haddadin

Die Dissertation wurde am 06.03.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 29.03.2019 angenommen.

# Foreword

This work would not be possible without the help of the many great people at the Chair of Automatic Control Engineering (LSR) at the Technische Universität München (TUM). Firstly, my gratitude is directed to my supervisor Prof. Angelika Peer for the never ending support and for all the fruitful discussions and guidance that made this work possible. I am also grateful to Prof. Martin Buss and Prof. Dirk Wollherr for their great support and for giving me the opportunity to pursue my research career at the LSR. My thanks extend to Frau Schmid for all her patience and help.

My time at LSR was an exciting scientific experience that was enriched with professional colleagues who were eager to discuss ideas and help in any possible way. Special thanks to my colleagues in the BEAMING and MOBOT projects, Milad Geravand and Stefan Klare, with whom I had the joy of working, sharing knowledge and sharing the road on our many trips. I would like to thank my colleagues who made the time at LSR enjoyable and enriching. I would like particularly to thank Mohammad Abu-Alqumsan, Philine Donner, Daniel Althoff, Andreas Lawitzky, Ken Friedl, Markus Kühne, Roderick de Nijs, Annemarie Turnwald, Sotiris Apostolopoulos, Muhammad Sheraz Khan, Alexander Pekarovskiy, Christian Landsiedel and Markus Schill. In terms of technical help, Wolfgang Jaschik, Thomas Lowitz, Tobias Stoeber, Domenik Weilbach and Kilian Weber for all their efforts. I thank you all greatly.

I would like to sincerely thank my students who were a great part of this work. They have consistently showed passion and scientific vigor in discussing and studying many questions this work has addressed. Thank you Christian Busch, Andre Christ, Tommy Schau, Lucas Falch...

This work is dedicated to the love of my life, my wife Sadia, who has supported me continuously. I dedicate this work to my family who have always pushed me to aspire to my dreams, thank you dad, thank you mom, thank you Ahmad, Luai and Haitham.

Munich, January 2018 Laith Alkurdi

## Abstract

To achieve seamless human-robot interaction, each agent should adapt to the anticipated state of the other. Inferred intentions should be a decisive factor on how a robot makes high-level decisions. The large state space of possible solutions to what action the human intends to perform in the environment, as well as the timing constraints to achieving a solution, renders the task of intention recognition nontrivial. However, a robot with abilities of estimating the intended actions of humans in its environment can anticipate their needs and plan accordingly. In this thesis, we present a control approach to intention recognition based on Dynamic Field Theory (DFT). We present this cognitive control architecture as a dynamical model that enforces concepts of embodied embedded cognition (EEC) where (generation and understanding) intelligent behavior is a product of the interaction between the agent's body, cognitive abilities, and the environment that it is situated in. The proposed work that is based on DFT estimates the driving force behind an agent's action by introducing an integration between top-down and bottom-up processes. In order to achieve intention recognition, we construct the control problem as a dynamic decision-making system. The first level of the top-down process tries to understand the context behind the observed kinematic movements of a human. The second level of the top-down process compares the observed the trajectory against a range of learned movements for recognition. The overall intention is recognized by understanding the performed actions in a top-down direction by mixing signals from the inference blocks as mentioned above. Explicitly, these two processes make sense of the observed action by parsing the trajectory of the movement on the one hand and the contextual meaning behind the movement on the other. This involves controlling traveling peaks in the integro-differential field equations and stabilizing the solutions of those traveling peaks. Finally, internal simulation allows the system to be predictive in the task of intention recognition. This internal simulation of movement generation represents the bottom-up process. The framework is tested in an environment within which perform high-level actions. Using the solutions as mentioned above, the system can come to a decision on what action the human is performing and what the underlying context is. Furthermore, the internal simulation bottom-up process is validated on a two-dimensional musculoskeletal arm model.

# Contents

# Notations

## Abbreviations

| | |
|---|---|
| AUA | Action Recognition Architecture |
| BN | Bayesian Networks |
| CARS | Conditional Action Recognition System |
| CARM | Conditional Action Recognition Module |
| CoS | Condition of Satisfaction |
| CN | Canonical Neurons |
| CNS | Central Nervous System |
| DBN | Dynamic Bayesian Networks |
| DFT | Dynamic Field Theory |
| DPA | Distribution of Population Activity |
| DNF | Dynamic Neural Fields |
| EPH | Equilibrium Point Hypothesis |
| HAMMER | Hierarchical Attentive Multiple Models of Execution and Recognition |
| HMM | Hidden Markov Models |
| IR | Intention recognition |
| ISI | Interstimulus Intervals |
| MNS2 | Mirror Neuron System 2 model |
| MNS | Mirror Neuron System |
| MOSAIC | MOdular Selection And Identification for Control |
| MSI | Mental State Inference |
| MT | Middle Temporal Visual Area |
| PAIRS | Plan, Action and Intention Recognition Systems |
| PLA | Point-light Animations |
| POMDP | Partially Observable Markov Decision Problem |
| RNNPB | Recurrent Neural Networks with Parametric Bias |
| RoS | Range of Satisfaction |
| STS | Superior Temporal Sulcus |
| TARS | Trajectory Action Recognition System |
| TARM | Trajectory Action Recognition Module |
| TCT | Threshold Control Theory |
| ToM | Theory of Mind |

# 1 Introduction

One of the goals of developing robotic systems is to produce machines that are capable of sharing an environment with people and ultimately assisting them with their everyday life tasks. This, however, requires that these robots be endowed with abilities of action understanding and intention recognition. Having such abilities in their arsenal would allow intelligent systems to understand what the observed actor is currently doing, why this action is being done and to what end. Moreover, it would allow these systems to assess in which way assistance could be given, and at which point this assistance could be introduced. The challenge then is building computational models that can aid intelligent systems in understanding manipulation or locomotion actions of an observed actor. In this work we present a biologically inspired dynamic approach towards intentional action and plan understanding.

Within the embodied situated cognition stance, intelligent human behavior can be understood as the adaptive response an agent produces due to the tight coupling between the agent's body, the environment and the agent's decision making processes [1, 2]. As the agent's decision making processes develop an intention, a series of intentional actions are sequentially produced to fulfill that intention within the current environment. Intention recognition can then be understood as the task of extracting the meaning behind the series of these actions [3, 4], and is tightly related to action and (action) plan understanding [5–8]. Human action understanding can then be understood as the task of relating a stream of human-related multimodal data (motion, audio, contextual, etc.) to the environment and classifying this data into semantic terms. Philosophically, the problem of mental state (intentions, desires, beliefs, etc.) attribution has been addressed by the theory of mind (ToM) [9, 10], within which actions are understood based on one's own understanding of the decision making processes.

The above ideas are also observed on a neuronal level within the mirror neuron system (MNS), where specific regions in the brain are active both when an agent produces and observes an action. The tasks of action and plan recognition are hypothesized to be one of the functions of the MNS [11, 12] in which the observed behavior (the trajectory of a reaching action) is mapped to one's own as explained by the direct matching hypothesis (or the motor resonance hypothesis) [13]. It is unknown at which level the observed behavior is mapped to one's own as explained by the direct matching hypothesis (or the motor resonance hypothesis). Possible non-exclusive options include a high-level intention level, an emulation level in which the motor code of the final goal of the behavior is matched or an imitation level in which the motor code of the trajectory of the behavior is matched [13]. The link between environment and behavior has also been observed in a different neuron system called the canonical neuron system (CNS) that seems to encode action possibilities directed towards different objects.

## 1.1 Intention recognition

Intention and intention recognition (IR) as terms are often used synonymously with concepts such as intentionality, desires, and beliefs. In the following, we define intention and intentionality from a philosophical point of view, and how they are related to desires and beliefs. Furthermore, we present studies that focus on human intention and the process of intention recognition. We present next how the concepts within human intention recognition motivate intention recognition systems within robotics and present classifications, characteristic, possible applications, challenges and limitations to such systems.

Intentionality is a technical term that could be attributed to mental states that are goal-directed [14]. Furthermore, it is a property attributed to actions such they could be called purposeful [15]. Intentions and intentionality are complex terms that are applied only when specific conditions are met. In order to attribute intentionality to an action, the agent is required to have *awareness* of performing such an action and has the *skills* required to perform such an action [16]. An intention, on the other hand, is a mental state that represents such an action [15]. An agent develops an intention once it develops a certain *desire* and a certain *belief* [16]. Explicitly stated, given an action $A$, one could give that action an intention $I$ if one grants an agent a desire for some outcome $O$ and a belief that $A$ will likely lead to $O$ [3].

In that sense, a desire is a prerequisite for an intention [3]. To eliminate confusion in understanding a desire and how it relates to an intention we describe three features discussed in philosophy on the distinction between the two concepts. First, intentions are directed at the intender's own action whereas desires can be directed at an abstract concept or object. Secondly, intentions are based on some amount of reasoning whereas desires are typically the input to such reasoning. Finally, intentions come with a characteristic commitment to perform the intended action whereas desires do not. In the following section, we expand on the concepts of human intention and present studies aimed at understanding the mechanisms that humans employ to achieve intention recognition.

#### Studies in human detection of intentions and intentionality

Through a set of psychological experiments, it had been shown that humans recognize other people's intentions and plan accordingly [17, 18]. Early work by Schmidt et al. in the late 1970's aimed at developing a system called BELIVER that was used to test the psychological theory behind recognizing intentions, goals, and plans of an agent through observing its actions [18]. Through these experiments it was seen that people were able to predict the next action or even the ultimate goal after been given a list of behaviors; moreover, they were able to attribute these actions as a part of a plan. Bartman later developed a framework for understanding intentions [19]. He concluded that intentions are the characterizing feature of a humans' actions and that they could be used to explain the reason behind them.

In general, it is thought that three stances exist to identify, explain and predict the behavior of systems [20]. The first stance is the physical stance, in which the actual physical construction of the system, that governs its operation, is used to understand and predict the behavior of the system. The second stance is called the design stance. It is

used to bypass the complexity that could be encountered in the physical stance. The basic premise here is that systems are designed to perform certain tasks and it should conform to the actions to ultimately achieve these tasks. Thus, by understanding the design of the system we can predict and understand behavior. The final stance and the most relevant to our discussion here is the intentional stance. The intentional stance is employed when it is unpractical to consider the physical and design stance for action prediction and behavioral understanding. Within this stance, the beliefs and desires of the agent, who is assumed to be rational, are analyzed based on the environment in which the agent is situated in and how the agent interacts with it. Once this information is available, the future actions that the agent might take to achieve his goals can be predicted. Within this work, we focus on modeling the intentional stance of human behavior understanding in a dynamic architecture that takes inspiration from the human ToM.

People read the intentions underlying the behavior of others readily and with little conscious effort. How do people so effortlessly detect intentions within the dynamic behavior stream and so readily apprehend its content? In psychology, it was determined that people directly perceive others' intentions on witnessing their actions [6, 21]. In other words, intentionality and the specific intentions at play are thought to be within the behavioral stream waiting to be detected. Concretely put, people directly employ ToM to perceive intentions of others by recognizing their actions and movement.

Theory of mind is defined as the ability to attribute mental states such as beliefs, desires, and intentions to oneself and others, as well as understanding why others may have different mental states than one's own [22, 23]. There exist two main mentalistic (and opposing) approaches to explain ToM in the philosophy literature. Namely, theory-theory and simulation theory [24, 25]. A third approach known as the "teleological stance" also exists, however, this is a non-mentalistic view of representing, explaining and predicting goal-directed actions [26, 27].

The basic idea in theory-theory is that the observer attributes mental states to the actor based on theoretical observations about the actor's behavior and the states of the environment that has an effect on the actor's behavior and mental state. Mental states can also be attributed to some previous knowledge of the target's mental states, and they can be used to attribute mental states such as intentions. The primary ingredient in theory-theory is the use of theoretical reasoning to attribute mental states. Theory-theory is thus a combination of two theses. First, common-sense psychological concepts can be considered theoretical concepts, and these concepts are employed similarly to how laws of physical science are considered. Second, people detect psychological states in themselves and others by making theoretical inferences accordingly.

On the other hand, in simulation theory, the observer puts himself in the actor's shoes and pretends to be in certain states that the actor might be in. He feeds inferred starting states into appropriate cognitive decision-making mechanisms to develop an internal understanding of the observed agent. In other words, the observer tries to make his own mind emulate the mental sequence the observed agent will go through. The observer, in this case, tries to reproduce what transpires in the agent, this is in contrast to theory-theory where this is not taken into account. The process of mental reproduction or simulation substitutes the theoretical reasoning by psychological laws which are unnecessary under

the simulation heuristic. This is not to say that simulation does not include any reasoning; the opposite is more accurate. Practical reasoning is involved, but perhaps not exactly theoretical reasoning. There might be cases where theoretical reasoning might be involved, such as when a simulation is trying to understand a theoretical thinker such as a scientist.

A third contrasting view to both the aforementioned mentalistic theories is the developmental psychological view of teleological stance [26, 27]. This stance is considered a non-mentalistic, reality-based view that aims at understanding intention by exploiting the relationship between the observed actions, possible goal states and the *situational constraints* (current state of the environment). It is contrasting to the mentalistic views as it can reason about the observed goal-directed actions by "making a reference to the relevant aspects of reality" without attributing mental states to the observed agent's mind [28].

The specific evidence for the evolutionary origins of mental simulation can be found in the discovery of particular neurons that fire both when performing a specific goal-oriented action and the observation of this action being perceived [29]. These neurons are called the mirror neurons (MNs), and they were first observed in the brain of the Macaque monkey. All mirror neurons discharge during specific goal-related motor actions such as grasping, manipulation or holding an object. Once an intention is developed, a specific set of neurons will fire to achieve that action. The observation of an intention and a goal-oriented action will fire those neurons as well, however at a reduced rate. Specifically, when a person observes an action, he will generate a plan to perform that exact action or imagine himself doing it. However, this plan is never allowed to be put online and is thus inhibited. Specifically put, the developed plan never yields motor output.

The human mirror neuron system (MNS) is a brain region that is active during the execution of a set of actions and is also activated during the observation of these actions. In terms of functions, MNS serve the purpose of imitation [30, 31], language evolution [32], and importantly in our case it is suggested that they endow the functionality of intention recognition [12] and action understanding [11]. Direct matching or motor resonance has been identified as the mechanisms behind the functions of intention and action understanding [33, 34]. Three interpretations are prominent as to what exactly is encoded and matched by MNS (for the function of direct matching) [13]: The first level is a detailed motor parameter that describes the action; this could be a trajectory of the hand itself. This level supports the role of imitation for MNS [35, 36]. The second level is also a motor encoding that describes the schema level motor plan. This level supports the role of MNS in emulation (goal imitation) [37]. The third level is a decoding of the intention driving the set of actions. This level supports MNS for intention recognition [38, 39].

The problem of mental state attribution is fundamentally uncertain as there exists no one-to-one mapping between different intentions and the actions they produce and vice versa. Furthermore, mental states are not directly observable. Modeling human cognitive ability of intention recognition is not a straightforward task. We have followed in this thesis the philosophical view that promotes both an action as well as a plan based understanding of intention. Explicitly stated, the account in this work is based on the view that intention recognition follows first the recognition of primitive actions (or what is refereed to as intention-in-action [40], immediate action [41], present-directed intentions [19] or proximal intention [42]). The second is the recognition of intention of the plan behind these actions

(or prior intentions [40], prospective intention [41], future-directed intention [19] or distal intention [42]). Both views are modeled in dynamics system theory and serve as a robotic cognitive decision making architecture. In the following, we introduce intention recognition within robotics, and highlight classifications, characteristics, possible applications, challenges and limitations of such systems.

### Intention recognition systems in robotics

Robots endowed with the ability of intention recognition would be able to facilitate human-robot interaction, overcome shortcomings with the communication channel as well as adjust its parameters to comply with human actions. For these reasons, intention recognition becomes an integral part of any human-machine interaction interface.

Intention recognition can be seen as the intersection of human-machine interaction, machine learning, and cognitive science. Intention recognition aims to infer the aims and goals of an agent through the understanding of its actions and the impact of these action plan on the environment. It also uses the observations of the environment state to make those inferences. An agent in this context is an autonomous entity situated in an environment in which it can act upon [43]. Intention recognition has many applications ranging from assisted technologies to interactive storytelling and computer games. They have been successfully introduced in system intrusion detections as well as observing military movements and riot control in urban environments.

Having an intention means that the system transcends from the realm of acting reactively, and starts to plan a sequence of actions towards a final goal and state. The recognition path becomes essential for deducting an agent's ultimate goal from his observed states and being able to predict what his next possible state would be. Intention recognition is important since:

- it enables pro-active cooperation and promotes cooperation; furthermore it preempts danger [44].
- it makes the interaction between human and machine almost as normal as that of human/human interaction [44].
- it counteracts possible communication problems in instances where [43]:
  - communication is not available in the agents due to it not being implemented or because of hardware restrictions.
  - communication is not reliable: such as problems with a temporary drop in communication or delays in the communication channel.
  - communication is uneconomic.
  - communication is in-agent: While communication is a very important part of the design of agents, having maximum reliability on communication messages takes away from the autonomy of agents in such that they are unable to acquire the information themselves but rather require this information to be sent to them; making any loss of data fatal to their operation.

  - communication is undesirable: for example in adversarial scenarios where it is a matter of security to reveal information, and communication channels could be undesirable.
  - communication is unrealistic: when trying to implement human scenarios, there exist certain constraints that can limit the quality and the quantity of information shared over a communication channel. This has to be taken into account and thus renders some of the methods of information sharing unrealistic.
  - communication is not understandable: unless there is a clear and common standard communication protocol, agents might not be able to understand each other's messages.

Intention recognition systems can be classified into four main classes [17]. The different classes and their definitions are as follows:

- Intended: the observed agent gives clear signals for his actions to covey his intentions.
- Keyhole: the observed agent does not intend for his actions to be observed or does not care. This case could lead to partial observability. This is the case with help systems that provide unsolicited guidance. e.g. ambient intelligence systems at home [45].
- Adversarial: the agent is hostile to his action being observed.
- Diversionary: the observed agent is trying to conceal his intentions by performing misleading actions.

Regarding characteristics, a successful IR system should be able to:

- Deal with uncertainty [46].
- Dive conclusions before a single plan/ action is fully recognized and defined [46].
- Not jump to conclusions if complete information is not available [46].
- Take temporal ordering as a strict constraint for plan and intention recognition [46].
- Handle actions occurring at the same time [46].
- Consider a single action for two different plans [46].
- Be customized to the agent whose intentions and plans are being recognized. Each agent's previous actions and preferences should be taken into consideration individually and used for intention recognition [47].
- Filter actions that do not have a direct impact on the current intention, from those actions that are an important part of the current intention [48].

- Handle the dynamic nature of the environment, while intentions are assumed to have the property of future-directness (which refers to the fact that if an intention is chosen by an agent then a set of actions within a plan are also chosen to be executed to achieve this intention in the future), the world is changing between the time the intention is conceived to the moment it is achieved. It is within that time the intention recognition system should be robust to dynamic changes [19].

Building a sophisticated IR system with the characteristics listed above is not without challenges. We list the most serious problems in intention and plan recognition as discussed in [49–52]:

- Expressiveness of plans: the system should be able to represent plans clearly for system interpretability and scalability.
- Sensitivity to Noise: Noise can be exemplified in adversarial settings, e.g., in which the observed agent is either trying to conceal his actions or deliberately trying to show other intentions. Noise can also be in the form of external actions that have nothing to do with the current intentions [53], previously defined as background actions. Noise can be in the form of external actions that are not part of the intention but elementary to other actions that are important within the plan and intention [54].
- Interrupted & interleaved plans: accounting for the case where the agent has many intentions, and it interleaves the executions of his actions to achieve his plans.
- Plan libraries & scalability: building algorithms that scale up to larger domains and different environments.
- Prior probabilities & performance: effective discrimination in the face of different possible hypothesis.
- Novel plans: the system should be able to both handle original plans and save them in the used plan library.
- Exploration: accounting for the case where the same agent is trying out different actions to achieve the same plan.
- Multi-agent: the explosion in complexity when an IR system encounters multiple (co-operating) agents.

Many limitations could hinder intention recognition as discussed in [43]:

- There might not be an intention at all. The agent under consideration might lack the ability or the control schemes to formulate an intention and a long-term plan. The agent might be reactive in nature. Agents under study might also not have a possible way of exhibiting intentions.
- Intentions are too complex to be conveyed through a communication channel and require a large number of parameters to be sent over.

- Intentions are dependent on the context that it is performed in, environmental variables should be taken into consideration to be able to infer the correct intentions.

- Agents performing intention recognition should fully define the amount and depth of information needed from the observed agent. This will expand the intention recognition problem from simple communication to a complex conversation between the agents to fully define each other's states.

## 1.2 State-of-the-art of intention recognition in robotics

Intention recognition systems aim to infer the intention of an agent given two primary inputs. The first input is the set of observed actions. The second input is a plan library that encodes a set of plans and the possible set of actions and their interdependencies within each plan. According to the intention inference approach, IR systems can be categorized broadly into consistency, probabilistic and dynamic approaches. In the following section, we give an overview of the state-of-the-art in consistency, probabilistic and dynamic approaches towards IR.

### Consistency approaches

Consistency approaches have been powerful tools in intention recognition research. They aim at sequentially removing possible intentions (and plans) that are inconsistent with the set of observed actions. The primary reasoning mechanisms that are usually used within consistency approaches are abduction and causal theories.

Causal decision theory describes the set of rational actions that are available in a specific scenario based on their expected causal consequences. The set of actions constitute a plan that best achieves a specific intention or goal. The same logic could be applied in an inverse-planning setting such as to recognize intentions and final goals.

Abduction, on the other hand, is a form of defeasible reasoning, often used to provide explanations for observations [55]. For example, if the room is hot and the heating is on then through abduction we can say that heating is on from observing the heat of the room. As abduction can provide more than one hypotheses to explain the intentions, Charniak and McDormott in [56] suggest some criteria for choosing between the competing hypotheses. Firstly, a hypothesis is most preferred when it uses the most specific characteristics of the observed action. For example, an interaction with a newspaper would indicate reading it rather than swatting a fly with it. Secondly, a hypothesis that requires fewer additional assumptions is most preferred. For the same example above an indication of reading the newspaper is preferred as it needs no further assumptions, when the indication of swatting a fly would require the observation of a fly. To rank hypothesis, generally two approaches can be used: i) global and ii) local. Global criteria prefer explanations that are minimal in some sense; i.e., the number of facts required to conclude the intention. Local criteria associate some form of evaluation metric with each rule in the background theory and provide a hypothesis metric which can be measured and compared against by combining the evaluation metrics of the rules that were used within the hypothesis.

Examples of consistency approaches can be found in Kautz's work in [46] where a formal theory of plan recognition was introduced. Lesh and Etzioni in [57] presented consistency graphs in which actions and schemas are described. Pruning rules are applied to this graph to reason about the possible intentions and plans. Additionally, consistency approaches have been used in human-computer interaction applications such as that in the *COLLAGEN* system as presented in [58].

Generally, in terms of advantages, consistency-based approaches tend to be highly expressive since the plan libraries are usually constructed manually [52]. Additionally, and since the plan libraries are constructed manually, arbitrary constraints between the actions composing the plans can be imposed [59]. Furthermore, temporal order and logical relations between actions can equally be represented.

Regarding disadvantages consistency approaches generally fail to consider intentions/action priors [52]. This forces the system to search the entire plan library initially, which can be computationally expensive. The reliance on plan libraries in itself is a primary challenge as these plans should be constructed manually and it would be hard to guarantee their completeness and their correctness [50]. Reliance on manually generated plan libraries within consistency-based approaches would render the system incapable to make a valid decision when presented with novel plans. Additionally, consistency approaches are rather sensitive to the observation of noisy actions that are not necessarily part of the agents intention [52]. In the same manner, consistency-based approaches are sensitive to the partial observability of actions and are heavily reliant on continually classifying actions accurately [50, 51]. Furthermore, they are inherently unable to handle interrupted and interleaved plans without further processing [44, 59, 60].

Concerning challenges, consistency-based approaches suffer from handling cases of abandoned intentions and cases where individuals are irrational and incompetent [60]. Furthermore, consistency-based approaches fail to handle cases where users are performing reactive actions or the case where multiple agents are cooperating towards a common intention [50–52].

Within intention recognition, we would always look for the intention that is most consistent with the observed actions. The challenge, however, is to consolidate the case in which actions are consistent with more than one intention [60]. Consistency approaches cannot directly choose between those intentions without information loss. As such it is helpful to have some probabilistic framework to work around this problem, which motivates probabilistic approaches to intention recognition presented in the next section.

#### Probabilistic approaches

Probabilistic approaches applied to intention recognition mainly cluster around the use of Bayesian Networks (BN) and restricted versions of Hidden Markov Models (HMMs). They have proven successful as they do not have the problems that the consistency approaches suffer from. Probabilistic approaches are capable of finding the most probable intentions given a set of current observations from accumulated statistical evidence or simply subjective beliefs encoded in a Bayesian network or a Markov model. Probabilistic approaches aim at quantifying the uncertainty of each of the users' possible intentions and ranking them in a probabilistic manner.

First models were built by Charniak and Goldman [61]. In their work, a library of plans was given, and BNs are built from the library using a knowledge-based model. The posterior probability is inferred to obtain explanations. There have been new advancements to this method notably to include the cases where the agent has many intentions or follows interleaved plans simultaneously or when the system fails to observe actions or addresses partially ordered plans [62].

A context-dependent Bayesian approach was used in [63], although this model is not incremental. This was applied to traffic monitoring, and it was shown that the contextual information is necessary to recognize the driver's intentions.

Bayesian Networks, in general, have been attractive for IR modeling as they are employed to summarize the general statistical evidence. They allow heuristic information to be linked with situation-specific information to reason about which logical action can occur and decide on possible actions to be performed [64]. As BNs are directed acyclic graph structures, the structure shows the conditional in/dependencies between the random variables represented by the graph nodes. A conditional probability distribution table gives information at every node. Regarding advantages, BNs are flexible in representing probabilistic dependencies as well as being efficient inference methods [65]. Concerning disadvantages, the probability updates within BNs are not sensitive to the ordering of the observed actions. Additionally, BNs' complexity increase exponentially as the number of observations increase.

Next to BNs, Hidden Markov Models (HMMs) are commonly employed to recognize intentions. They provide stochastic models for collecting information sequences over time to make estimates on hidden states. Fernandez et al. in [66] presents an HMM intention recognition model to enhance the active cooperation between a robot and a human in transporting rigid objects. Han et al. in [67] utilize HMMs to recognize the intention of observed robots so that an observer robot can act accordingly in a robotic-soccer playing application. Additionally, Yu et al. in [68] present an approach using HMMs to assist human motion in a remote environment by combining human movement intention recognition with real-time environment information. Kelly et al. in [69] discuss an approach for human intention recognition performed by a robot in which concepts in Theory of Mind inspire HMM intention recognition. Regarding advantages, HMMs can handle partially observable states as well as states that can not be identified with high certainty [59]. Concerning disadvantages, HMMs require a large training set as well as sufficient understanding of the problem domain [70].

Dynamic Bayesian Networks (DBN) are also used to model intentions and intention recognition as it is a probabilistic model that provides the ability to reason under uncertainty. Furthermore, it is a causal forward model that allows for subsuming temporal information of successive measurements. Therefore it can encode temporal ordering as opposed to regular BNs that are insensitive to ordering. DBNs are directed acyclic graphs with nodes representing actions and edges representing causal dependencies among these variables. The causal dependencies are modeled using conditional densities. As DBNs capture the development of the network over time; edges are used to connect nodes (models) from one time step to another. These edges are used to represent dependencies from $t$ to $t + 1$. Within this network there is one intention state at each time step, this state

is hidden and discrete as there are classes of intentions we aim to differentiate between. As intentions are affected by the environment, this is captured by a node containing domain knowledge. Duchaine and Gosselin in [71] model intention recognition using DBNs to achieve natural interaction between humans and robots. Tahboub in [72] also models the problem of intention recognition using DBNs. The work aims at introducing an IR module to aid a human as he/she controls a mobile robot through a joystick. Regarding advantages, DBNs allow the modeling, representation, and learning of complex intentions while taking the temporal nature of actions into account. Regarding disadvantages, DBNs suffer from increased computational complexity [59].

Intention recognition is a dynamic process that would benefit from modeling the adaptive interaction between the observed agent and its environment. As such it is helpful to utilize a dynamic framework to address the challenges of intention recognition, which motivates dynamic approaches to intention recognition presented in the next section.

#### Dynamical models approaches

Dynamical models have successfully been used in literature to model IR systems (and more generally computational models of Mirror-neuron systems) as they address the temporal aspects of behavior generation. For example, the MOdular Selection And Identification for Control (MOSAIC) model [73–75] is a decentralized, learning-based, adaptive, dynamic controller that relies on switching between different learned internal models that model the dynamics of motion generation (forward/inverse models, predictor/controller pairs ) to best achieve a given movement task. A similar model that functions at a higher behavior level and is used primarily for imitation is the Hierarchical Attentive Multiple Models of Execution and Recognition (HAMMER) [76–78]. A dynamic system model for imitation, learning and action generation has been proposed using a Jordan recurrent neural network with parametric bias (RNNPB) [79–81]. A Jordan network is a recurrent neural network with the context unit fed-back from the output layer to the context units in the input layer. Recurrent neural networks of the Jordan type are also used in another model called the Mirror Neuron System 2 (MNS2) [82]. MNS2 is an extension of MNS1 which was implemented using simple feedforward neural networks. Generally, neural networks learn the mapping between the observed e.g. actions, trajectories, etc. and its respective intention, actions, etc. accordingly. Neural networks require a large training set and as well as extensive tuning of the hyper-parameters of the network itself. Furthermore, neural networks are treated as black boxes and their results are usually hard to interpret [83].

Dynamic systems theory was explicitly employed in the cognitive framework of dynamic field theory (DFT) as a decision-making system to understand intentions (and actions) in the work of Bicho et al. in [84] and recently in the work of [85]. DFT allows the use of different dynamic neural fields to model the different neural populations that are responsible for the various functions within the task of intention recognition. DNFs seems to model behavior very well due to the dynamic, continuous interaction between the different neural populations involved in the decision making.

### Summary

In the following, we present a summary of the different state-of-the-art methods. The comparison between the different approaches is summarized in Table 1.1 where each criterium (expressiveness of plans, sensitivity to noise, interrupted plans, plan libraries and prior probabilities) is given a score [++, +, 0, −, −−] ranging from a strong advantage (++) to a strong disadvantage (−−) given each approach.

Compared to probabilistic approaches, consistency approaches cannot handle the case when a specific action relates to many intentions. In those specific cases, they are unable to select between possible intentions and in some cases, they might not be able to logically infer an intention at all [59]. Consistency approaches are rather expressive as they are usually constructed manually [52]. Furthermore, they are unable to deal with ambiguity or cases in which the agent attempts original plans or when noisy actions are observed [52]. Additionally, consistency approaches are unable to handle partial observability of actions [51]. Consistency approaches are not capable of neither detecting interleaved plans, that are being performed in parallel, nor interrupted plans [60]. Plan libraries are considered as an input to consistency approaches and required to be available beforehand[49]. Consistency approaches do not consider prior probabilities and thus require a complete initial search of the solution space once an initial action is observed [52]. Consistency approaches require a few actions to be observed to give an initial indication of the possible observed intention; rendering the approach slower compared to other probabilistic approaches [44].

Compared to consistency approaches, probabilistic approaches address the issue as mentioned earlier of resolving multiple intentions relating to one actions [44]. Regarding expressiveness, Markov models are considered less expressive and are capable of predicting next possible action steps [52]. Markov models are less expressive when compared to Bayesian networks where the dependencies between a set of actions within a plan/intention are explicitly defined [44]. Probabilistic approaches are better equipped at handling observations of noisy actions, as each observation counts as evidence of a specific intention [52]. However, as Markov models operate using transition probabilities, a noisy action could lead to inaccurate predictions which require further processing. Furthermore, Bayesian probabilistic approaches address the case where an agent might have multiple or interleaved intentions, or the case when actions are not observed, as well as partially ordered plans. This is in contrast to consistency approaches that are unable to handle these cases [44]. Regarding the generation of plan libraries, probabilistic approaches are well equipped to generate and learn the structure of the, e.g., network or the parameters of the plan library incrementally from examples [59].

Compared to purely consistency/probabilistic approaches, dynamical models are highly expressive and are less sensitive to noisy observations due to the constant update of sensory information. Plan libraries are learned through a manageable dataset of examples using simplified assumptions, e.g., normality and robust statistics, e.g., median and mode. Furthermore, priors can be defined as the initial conditions of the different states within a dynamical systems approaches. Additionally interrupted plans can be inherently handled due to the dynamic adaption to new observations. Due to the favorable comparison compared to consistency and probabilistic-based approaches, we opted to use the dynamical systems approach for modeling IR. Explicitly we have chosen to utilize Dynamic Field

Theory (DFT) to model human action/plan/intention understanding.

**Table 1.**Comparison between consistency, probabilistic and dynamic approaches

| | Comparison criteria | | | | |
|---|---|---|---|---|---|
| Approach | Expressiveness of plans | Sensitivity to noise | Interrupted plans | Plan libraries | Prior probabilities |
| Consistency | ++ | −− | − | 0 | −− |
| Probabilistic | −/+ | + | + | + | ++ |
| Dynamic | ++ | ++ | + | + | ++ |

## 1.3 Problem statements and challenges

In this work, we present a dynamic intention recognition architecture that follows the motivation given so far. Intention recognition is decomposed in this work into two specific steps; proximal intention understanding and distal intention understanding. Proximal intention understanding refers to the understanding an agent's action and its immediate effect on the environment. Distal intention, on the other hand, aims at understanding the intention behind the sequences of actions performed by an agent and reasoning about the agent's plan. Furthermore, proximal intention is decomposed into two steps as motivated by findings in MNS and as described in ToM. The first step is a top-down understanding of the observed kinematics and the agent's interaction with the environment. The second step is a bottom-up internal simulation of the predicted movement. In the following subsections, the challenges are broken down for the specific topics.

### Top-down proximal intention understanding

The first challenge within top-down proximal intention understanding is to identify the methods that humans employ when observing other humans which are acting in their immediate environment. Explicitly, the challenge is to identify the signals that humans rely on to understand intentions, plans, and actions of others around them. Furthermore, these signals are to be represented and modeled in a coherent dynamic framework that promotes top-down action understanding in a manner that is both biologically and philosophically plausible. Additionally, this top-down approach should model the immediate environment in the same dynamic framework. The top-down approach should also be able to resolve the spatiotemporal variability that is observed across different examples of the same actions.

### Bottom-up proximal intention understanding

With the top-down proximal intention recognition as the first step, the second challenge is to model the bottom-up approach. The bottom-up approach provides a reinforcement step to the understanding of the observed action in a similar manner to that which is observed in the Mirror neuron systems. The challenge here again is identifying the biological signals that are responsible for producing specific movements and modeling them in a dynamic

manner. Furthermore, other challenges include validating the resulting kinematics using a musculoskeletal system and comparing it against human-generated movements. Finally, the bottom-up approach has to be biologically and philosophically plausible and is capable of explaining the interaction between the movement and the environment in a coherent framework.

#### Distal intention understanding

Once the observed kinematics are linked to the immediate environment, and the proximal intention is understood, a different challenge arises. The challenge here is to understand the distal intention behind the series of observed atomic actions that the agent is performing in the immediate environment. Distal intention understanding presents its own set of challenges. The primary challenge is to make sense of a string of actions in a framework that is consistent with the proximal intention recognition approach, however at a different level of abstraction. Explicitly, the same methods used within proximal intention recognition should be used again in the distal intention understanding step as motivated by findings in MNS. Additionally, the overall system should be able to predict the next action and reason about the possible set of actions that could occur in the future. Finally, the system should dynamically adapt to new actions observed and react accordingly.

## 1.4 Main contributions and outline of this dissertation

With regards to the challenges stated in the previous section, the following contributions are part of this dissertation:

#### Action understanding from observed movements and context

In chapter 2, the Action Understanding Architecture (AUA) is formalized within dynamic system theory [1, 2]. The main components of this architecture are introduced in chapter 3, where we present the concepts of contextualization and trajectory comparison as the basis of action understanding. The contextual action recognition system (CARS) and the trajectory action recognition system (TARS) are modeled using Dynamic Field Theory [2]. The AUA is based on three hypotheses: firstly, to understand human action, one performs a predictive step to understand the context of the movement. In this predictive step, an observer (trying to understand the actions of an actor) would shift his attention towards an object the acting agent might direct his actions towards. Here we assume that the observing agent directs his gaze towards an object given the direction, and speed of acting agent's end-effector. This contextual prediction step is supported by studies of directly observed behavior in which the relationship between an actor's end-effector and the observer's gaze is described to be predictive [86]. Secondly, once the context of movement is understood, and the object is defined, the affordances of this object are read out. Affordances are used to define the action possibilities that are available by a specific object [87, 88]. Finally, once the possible actions towards the objects are known, the potential trajectories towards the objects could be loaded in preparation for the comparison. This is in accordance with biological studies indicating that the kinematic features are central to the understanding

of human action [89–91]. The comparison is performed by comparing the current movement trajectory against a set of learned trajectories of different movements maintained in long-term memory structures. This is shown in Fig. 1.1, within the proximal intention understanding block.

#### Internal simulation of reaching motion

In chapter 4, we introduce the concept of internal simulation of a movement as an alternative to comparing observed motion against saved memories. Assuming that the understanding of reaching movement is explained by the direct matching hypothesis within MNS, we answer the question of how the internal simulation of a reaching movement is performed given the initial movement information and the context of the motion. Explicitly, we model the dynamically generated internal simulation signal models with the reciprocal $R$ command of the end-effector as explained by the Threshold Control Theory (TCT) [92]. This $R$ command is modeled using a dynamic attractors system and is also validated within our work on a musculoskeletal arm model as explained by the threshold control theory. This complies with descriptions within MNS in which the internally simulated motion should be identical to the one generated when performing the action as opposed to just understanding it. Therefore we additionally make use of the dynamically generated $R$ command as well as the $C$ command to calculate the equilibrium points to generate motion in a musculoskeletal arm model towards the goal object as validation. A comparison can also be dynamically performed against an internally simulated movement of the possible action. This is shown in Fig. 1.1, within the proximal intention understanding block.

#### Plan understanding

In chapter 5, the description of the plan understanding systems is given. We transfer the different components discussed in chapter 3 that were used in the task of action understanding to the higher abstraction of plan understanding. Explicitly, we discuss how affordances are integral in understanding plans and model it within dynamic field theory and discuss how plans can be dynamically generated given new observations of actions. Additionally, we discuss how the comparison between learned plans and the dynamically generated plans, that are based on observations, can be accomplished. We explain how the comparison within the plan understanding part utilizes the same methods as that in action understanding part albeit at a different abstraction layer. This is shown in Fig. 1.1, within the distal intention understanding block.

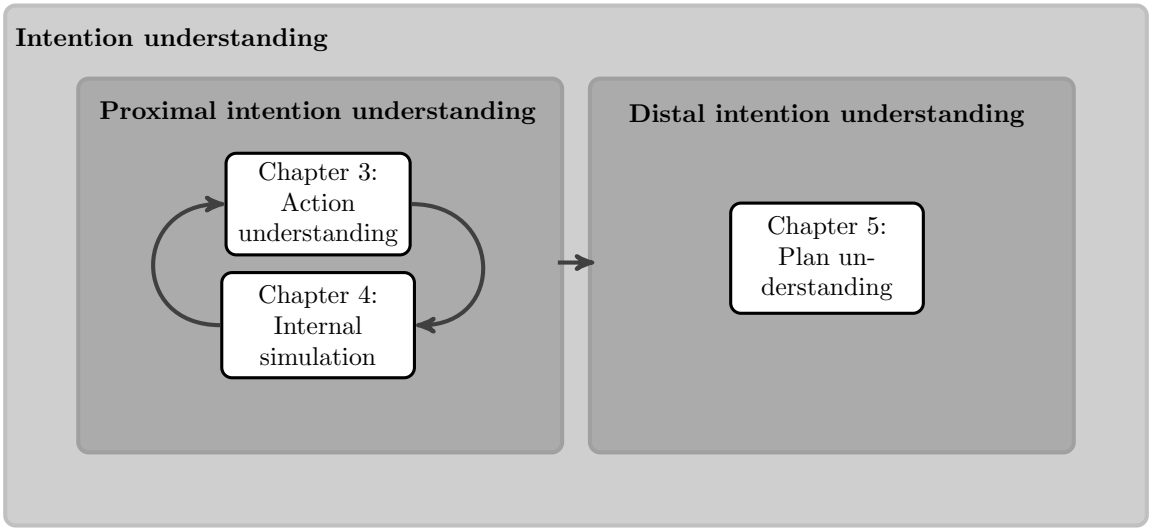


**Figure 1.1:** Visualization of the various components in the thesis and the structure of the thesis.

# 2 Action understanding architecture

Proximal intention understanding refers to the understanding of an agent's action and its immediate effect on the environment. Within this thesis, we decompose the problem of proximal intention understanding into two steps as motivated by findings in MNS and as described in ToM. The first step is a top-down understanding of the observed kinematics and the agent's interaction with the environment. The second step is a bottom-up internal simulation of the predicted movement. The two explicit steps, action understanding and internal simulation, are shown in Fig. 2.1 where the different blocks and their motivation is discussed next.

In this work we motivate our approach by descriptions of cognition in which cognition is said to be *enacted* in the sense that cognition arises for the purpose of adaptive actions [1], and the objects in the environment are represented to reflect their action possibilities and affordances [87, 93, 94]. When observing acting agents in the environment, an observing agent uses its body to understand the observed agent's behavior [95]. Additionally, the observing agent perceives information directly from the environment and uses the context for understanding and making decisions accordingly. Explicitly, an agent perceives object affordances and biological motion. Indeed a major theme in socially-situated cognition is reserved to the idea that the movement and the environmental state of the agents around us are mapped onto the perceiver's body [96].

In this chapter we present a novel architecture that models the environment through the concept of affordance to understand (or simulate) the kinematics of an acting agent in a manner that is consistent with definitions within situated embodied embedded cognition. The AU architecture (AUA) presented in this work is a deterministic model that reacts to the input and produces decisions dynamically, as a computational mirror neuron systems model, in the consistent framework of dynamic field theory within dynamic systems theory.

The common theme among computational models based on mirror and canonical neurons is to incorporate a forward model and inverse model to understand an action. The forward model predicts the expected sensory outcome given a motor command. The inverse model on the other hand maps the sensory input to the motor command. This is observed in the HAMMER family of architectures [77, 78, 97–100] and the MOSAIC model for motor control [73–75] that are mainly designed for imitation and motor control. The theme is also observed in the family of architectures composed of the Mirror Neuron System (MNS) model, the Mental State Inference (MSI) [101] and the MNS2 model [82]. This family of architectures focuses on the modeling of the development of the monkey mirror neuron system for grasping. None of the above families of architectures use the concept of affordances (nor model canonical neurons) for the task of imitation/action understanding. A bio-robotic model for the mirror neuron system was proposed in [102, 103]. In contrast to the models discussed before, the bio-robotic model incorporates a canonical neuron component that aids in the selection of the motor plan that should be active when observing

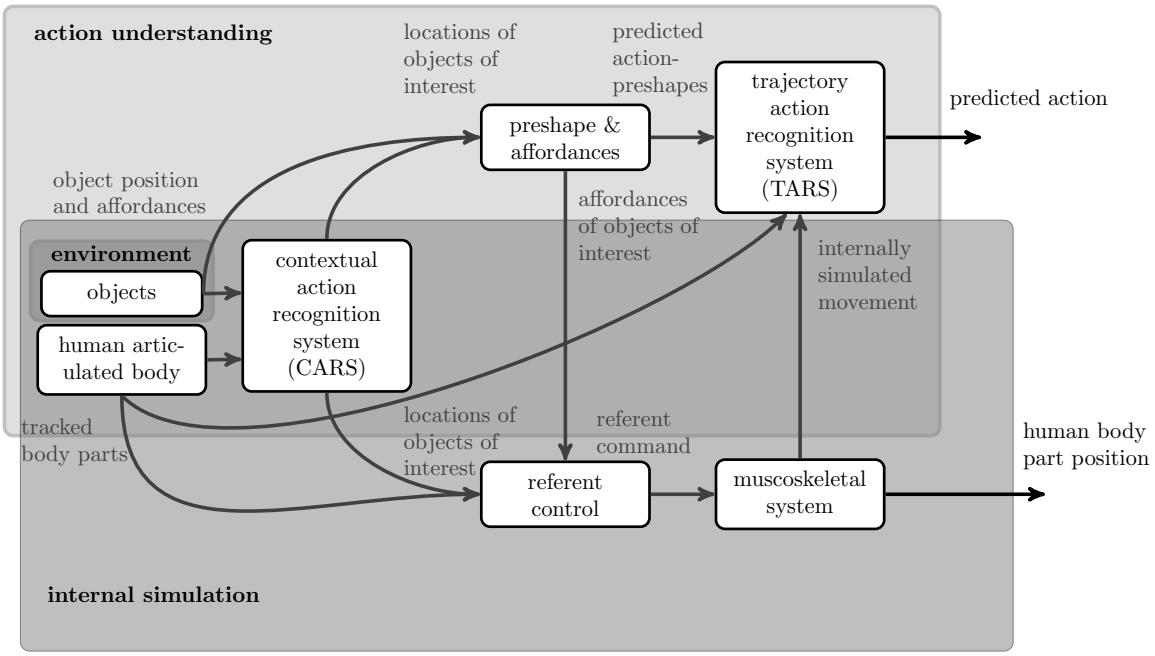


**Figure 2.1:** The AUA can be decomposed into a top-down and bottom-up direction. The top-down direction is represented by the action understanding group of blocks and is discussed in chapter 3. The bottom-up direction is represented by the internal simulation group of blocks and is discussed in chapter 4.

an action. The bio-robotic model also utilizes a forward model to predict the motor command given the visual input of the observed agent. In contrast to the bio-robotic model for the MNS the AUA presented here utilizes the affordance logic to read out the action possibilities given the contextual understanding of the observed motion. Furthermore, the affordance logic block mainly reduces the number of candidate trajectories used for the comparison against the observed motion rather than determining which motor plan should be active when an agent's action is being observed.

The modeling of action understanding systems using DFT has been addressed recently in literature. A neural dynamic approach for parsing a sequence of actions was presented in [85] by Lobato et al. The authors present a neural-dynamic architecture that is capable of detecting and representing a sequence of actions, namely reaching/grasping/dropping objects on a table-top scenario. Trajectory recognition was not considered but rather three-dimensional positions of hands and objects were used to calculate whether the hand was approaching the object or not. The overall architecture is capable of memorizing a string of actions for overall action understanding. Similar work was also presented within neural fields in the work of Bicho et al., in which the focus was on integrating verbal and nonverbal communication in a joint-assembly task in which the sequence of actions were given [84]. In contrast to the work presented by Lobato et al. and Bicho et al. we extend the application area of DNFs towards representation and recognition of temporally extended actions using context and movement information. Furthermore, while the work presented in Lobato et al. and Bicho et al. deals with only table-top scenarios, we present systems that are general enough for understanding locomotion, manipulation and actions in free-space.

Overall, the AU architecture in this work presents, for the first time, a novel predictive system within DFT that models attention-shifts and pairs up with a trajectory parsing system in a second step. Furthermore, the system models the internal simulation of movements. The trajectory parsing system takes account of spatial as well as temporal variations that are usually problematic when understanding actions. Special attention is given on how objects and the environment are integrated in the overall architecture and on how they can drive action understanding.

Compared to the state-of-the-art, the AU architecture in this work combines both context recognition and trajectory recognition rather than opting for either contextual recognition alone or trajectory parsing by itself for the task of action understanding. Additionally it uses concepts of internal simulation of movements as inspired by the MNS. Furthermore compared to the related work within DFT we explicitly model objects and their affordances in a manner that is consistent with definitions in the situated, embodied view of cognition that DFT is built upon. The application domain of this model ranges from scenario understanding to human-robotic interaction scenarios where intelligent systems are expected to assist humans in a meaningful manner. [104, 105]. The model's strength stems from the interaction between the contextual systems (CARS), the trajectory parsing system (TARS) and the affordance system, such that a wide range of actions (manipulation, locomotion and free-space actions) could be understood. The model suffers from a few limitations in the current status. Firstly, the model makes use of a few algorithmic shortcuts that are not biologically plausible. Secondly, the current technical implementation is restrictive

(e.g. due to slow offline template generation).

Explicitly stated, the AUA presented in this chapter combines, for the first time, predictive contextual understanding, trajectory parsing and object affordances using the cognitive dynamic framework of DFT and in a manner that is consistent within the situated embodied embedded cognition stance. Furthermore, the AUA models the internal simulation of movement using dynamic systems theory using attractor dynamics and threshold control theory. Specifically, we can classify our AUA approach as a dynamic, single-layered exemplar-based sequential method, that depends on contextual information when choosing/simulating the example (template).

We expand on the concepts that motivate our approach next. Explicitly we introduce the concepts of the situated embodied embedded cognition first. Next we explain what is meant by affordances. Then, we introduce the concept of biological motion perception that motivate biologically-inspired features used within this work and link it to concepts within threshold control theory that explains how motion is generated. Finally, we explain how these different concepts are related to findings in neuroscience to emphasize our biologically-inspired approach.

### Situated embodied embedded cognition

The basic hypothesis behind situated cognition is that behavior is a product of the dynamic interaction between the agent and its environment, and is inseparable from the context that it emerges from [1, 96, 106]. Information is thought to be a product of the coupling between the agent and its environment rather than an a priori representation in the agents brain as proposed by traditional views of cognition. Situated cognition shares ideas with ecological psychology [88] and intentional dynamics [107]. Moreover, cognition as defined here is understood as a continuous state in which motor-sensory systems interact dynamically and thus can be described naturally using ideas from dynamic system theory [2]. As cognition in this view is a continuous state that affects motor and sensory systems, dynamic system theory is the framework that situated embodied cognition can be described.

Situated cognition as a scientific stance is built on several theses which we present in the following. Firstly, cognition is said to be embodied and situated in the sense that it arises and is a function of the tight coupling between the agent's body and the environment that it is in. Secondly, cognition is situated in its social context in the sense that it arises from the coupling between the agent and its social environment. Thirdly, cognition is enacted in the sense it arises for action due to the agent's adaption of intentionality. Taking these three points into consideration, cognition can be thought as distributed across the objects in the environment and the context that the social agents are situated in. It is therefore, the way that the environment is perceived that influences behavior rather than an internal representation an agent might house in its decision making systems.

The aforementioned line of thought is at the bases of the AUA described in this work. We explicitly design systems (e.g. contextual action recognition system and affordance logic system) and integrate concepts (e.g. internal simulation via threshold control theory) within a cognitive framework (implemented via dynamic field theory) that respect the theses that situated cognition is based up on for the purpose of action understanding.

As mentioned before, situated cognition shares ideas with the field of ecological psy-

chology, specifically with the concept of affordances which defines the action potentials of objects in the environment. We have alluded to this term in our introduction and gave a brief definition. In the following section, we give a more formal definition of this term.

### Affordances

The term affordances was introduced by Gibson as a general and powerful concept to explain what the environment can afford for an agent, and what existing action possibilities are [87, 88, 108]. It is a product of Gibson's ecological approach to cognition which stresses the strong connection between perception and action. In this ecological cognitive approach, affordances are the central perception element. Using affordances, goal-directed action possibilities are then perceived directly from the environment.

The exact definition of affordances has been a point of dispute since it was introduced by Gibson himself, leading to a range of attempts to formalize the concept [109–113]. In this work we take inspiration from the previous references and define affordances as agent-relative, perception-independent, action-invariant activity-potentials an agent directly perceives from the environment it acts in. They are agent-relative in the sense that affordances are attributed to environmental objects with respect to agent parameters (e.g height, width, etc), as an example, an infant's chair might not afford sitting on for an adult and so on [114]. They are also agent-relative in the sense of ability. An affordance disappears if an agent finds himself unable to make use of it. Affordances are also perception-independent as they exist regardless of whether they are perceived or not. They are also action-invariant meaning that affordances do not change in relation to the agents action goals [115]. Historically, the term was influenced by the work of gestalt psychologist such as Koffka who stated that objects have "demand character" that demands the agent to interact with it in a specific way [116]. Using this mind set, one can think of a cup as what it says to the person, namely "drink from me". Concepts of valence and invitation also influenced the idea of affordances.

Affordances can be understood by their properties. Gibson describes affordances to be objective, real and physical. He also describes it as being a "fact of environment and fact of behavior" [88]. The ecological approach to cognition is built on the ideas of affordances and direct perception. Affordances in that sense are perceived directly from the optic array by picking up sensed invariants.

Affordances in this work are hypothesized to be the driving force behind action planning and action production. Furthermore, we hypothesize that it is an essential part of human action and plan understanding/recognition. Namely, the contextual information of the available affordances of an object towards which an arm movement is directed give hints to what the action in itself means. This statement is the motivation behind the contextual action recognition system presented in section 3.4. Its role in the overall architecture is illustrated in Fig. 2.1.

Context can not function alone, and for the task of action recognition, goal-directed movement should be also recognized. In the following we discuss what biological motion perception is and how biological systems are thought to understand movements. Biological motion perception inspires our goal-directed motion recognition system that is presented in section 3.5.

### Biological motion perception

There exists features of body gestures, facial expressions, and eye movement that normally accompany social interaction. The recognition of these changing features and body movements is called "biological motion" and it plays an important role in action/intention recognition and movement anticipation.

The kinematics of human movements are the most visible and important form of visual information available for an observer. Recognition of actions can be as straight forward as observing the kinematics of an actor. In visual sciences action perception from kinematics is studied using the point-light animations (PLA). Studies on PLAs have exploded since Johanson's seminal paper in which he showed that people can identify actions by observing moving PLAs and that static PLAs have no significance in such a task [89]. Specifically, PLA are an important tool to study action recognition because they allow action kinematics to be dissociated from static information about the human form.

Ever since Johansson's [89] contribution in biological motion, where he studied the significance of form information in action recognition without regard to shape information, there has been an influx of studies that showed that there exists a large amount of information that could be inferred by perceiving simple stimulus encoding biological movement. In his original experiments, Johansson attached light point sources to the joints of actors and used cameras to record their movement against a dark background and without ambient light to make these PLAs. Later when test subjects were shown the PLAs, they were able to tell the movement pattern (walking) without trouble even though they were given no prior information about the shown recordings. Moreover, experiments showed that humans observers are skilled at identifying gender, identity, age, emotional state and even personality characteristics from the movement patterns of their acting counterparts [117, 118]. It is also interesting to report that humans also similarly attribute internal states to objects that show animate movement such as 2D triangles [119]. The perception of PLAs is robust and has been shown to function well even if the displays were out of focus or their contrast polarity was made to be different over time. Moreover, it has been shown that correct detection was even possible even when the PLAs were embedded in dynamic noise [120, 121].

There is no doubt that the movement of biological agents houses a considerable amount of information an observer might use to infer actions and intentions. Indeed, the focus on intentional action recognition research has been focused on bottom-up factors [122]. This has been shown in machine learning as well as biological action recognition research groups where extracting features from the observed motion stream is central to perform action classification. In this school of thought, extraction of structure from biological motion [89] has been the main driving force behind experimental setup and research dedicated to intentional action parsing, human behavior understanding as well as mental state attribution [123]. We direct the reader to the following reviews that address biological motion perception more thoroughly and the neural mechanisms behind the recognition of biological movements [90, 124, 125].

Humans are also able to process static images of dynamic movements very similarly. Experiments where subjects were shown single images of actors performing a dynamic movement elicited a response comparable of that when the complete motion was perceived,

while on the other hand images of actors not performing a dynamic movement did not [126, 127]. Moreover, it was shown that presenting a sequence of two static images was enough for subjects to perceive human action [128]. These experiments had also shown in PLA walkers where the accuracy of detection increased with the number of frames in the sequence [129]. The conclusion of the previous work showed that indeed biological motion was "perceived from a sequence spatiotemporally sampled samples". Thus, the recognition of human movement can be achieved by a temporal concatenation of static body postures in what is known as the template-matching model [130]. The above ideas inspire our decisions within our novel trajectory-based action recognition system that is presented in section 3.5. Its role in the overall architecture is illustrated in Fig. 2.1.

#### Threshold control theory: referent control

The Equilibrium Point Hypothesis (EPH), also referred to as threshold control theory (TCT) [92], describes the cognitive and neuro-physiological nature of motion control and provides solutions to the multi-muscles system. TCT has been used in several biomechanical models [131–134].

Within our embedded cognitive action understanding architecture, where the tight coupling between the agent and the environment is respected, TCT provides a natural explanation to how motor actions are obtained. The role of the nervous system, as explained within TCT, is to shift the threshold positions $R$, given information from the environment. This is in contrast to what is usually discussed in robotics where the nervous systems is thought to directly specify the motor commands and mechanical variables. Instead, within TCT, the movement trajectory, muscle activations, forces, torques and equilibrium positions are emergent variables due to the neural specification of the threshold positions $R$.

There exists many forms of threshold position control within TCT depending on the level of neuromuscular system that is of interest. Within our cognitive framework the link between environment (objects, their locations and their affordances) and agent is of major importance. Furthermore, actions in our work are described as goal-directed towards (the positions of) objects and given their affordances. Therefore, we define the threshold position to be the referent configuration of the hand $R_h$. This would aid in the association of the current hand position $Q_h$ with the objects in the environment.

To produce an intentional action, the current hand position $Q_h$ is shifted dynamically to the threshold position $R_h$ such as to interact with an object based on its affordances. An equilibrium trajectory is formed due to the shifts occurring in the equilibrium position of the hand. The equilibrium reference trajectories that are observed for fast reaching motions are characterized to be spatially similar to the actual trajectory of the hand and ending around where the physical arm usually achieves its peak velocity[135, 136].

Threshold control theory and referent control influences our decisions in chapter 4 where we implement dynamic referent control shifts that can also take into account obstacles avoidance. The internal simulation step as well as the referent control block is shown in Fig. 2.1.

#### Relation to Neuroscience

Action and generally context understanding is also observed on a neuronal level in biological agents e.g. as functions of the mirror neuron system (MN) and the Canonical neuron system (CN), respectively.

MNs are specific neurons in the agent's brain that fire not only when the agent is performing an action, but also when the same goal-directed action is observed. Their proposed function is to represent an embodied process that allows action and intention recognition [11, 12] as well as Theory of Mind [137]. The mechanisms the MN system uses to achieve these functions are usually explained by the direct *matching hypothesis* or *motor resonance* in which the encoded neural code of what is observed is matched with a generated neural code of how that movement could be executed [33, 34]. What is being matched could be, high-level abstraction of the intentions, a motor code encoding the plan to emulate a goal-oriented action, or a detailed motor code of the action itself encoding the trajectory of the movement and how to imitate it [13]. Additionally, it has been shown that there exist specific neurons in the MN system that have large specificity towards the way the action is performed and the final goal accomplished, while other neurons lack this level of specificity and the relationship is restricted to the action goal. Other properties of MNs are that they do not activate when observing objects alone, nor when the movement alone is shown [138].

Canonical neurons on the other hand seem to encode action possibilities directed towards objects and motivates our incorporation of affordances in a biological model for AU [138–142]. Indeed, action can be understood given both the motion and the goal towards which the action is directed [143].

## 2.1 Action understanding system architecture

Specifically put, our hypothesis for modeling the understanding of human action is as follows: the robotic (intelligent) system projects its perspective to that of the acting agent - whose action is to be understood. The robot perceives the affordances directly, relative to the acting agent's body and the environment (objects and their properties). The agent's brain controls the body to localize itself towards objects and to perform manipulation actions. The brain can also observe the own performed actions or of other acting agents. The same models and principles (brain) are shared among the cognitive agents (both the robot and the human) that share the same environment. Therefore, the same processes are assumed to be shared and what a robot simulates is similar to what a human plans. We show an illustration of this work flow in Fig. 2.2(a).

The abstract blocks and connections, motivated from cognitive studies and neuroscience, illustrated in Fig. 2.2(a) are translated into the proposed AUA. Furthermore these blocks are illustrated in Fig. 2.2(b) where the connections between the perception blocks (body and (virtual) objects), the contextual action recognition system (CARS), the affordance logic, the trajectory action recognition system (TARS) and the (internal) simulation block are shown.

As discussed in the introduction, the ability to understand actions of others (moti-

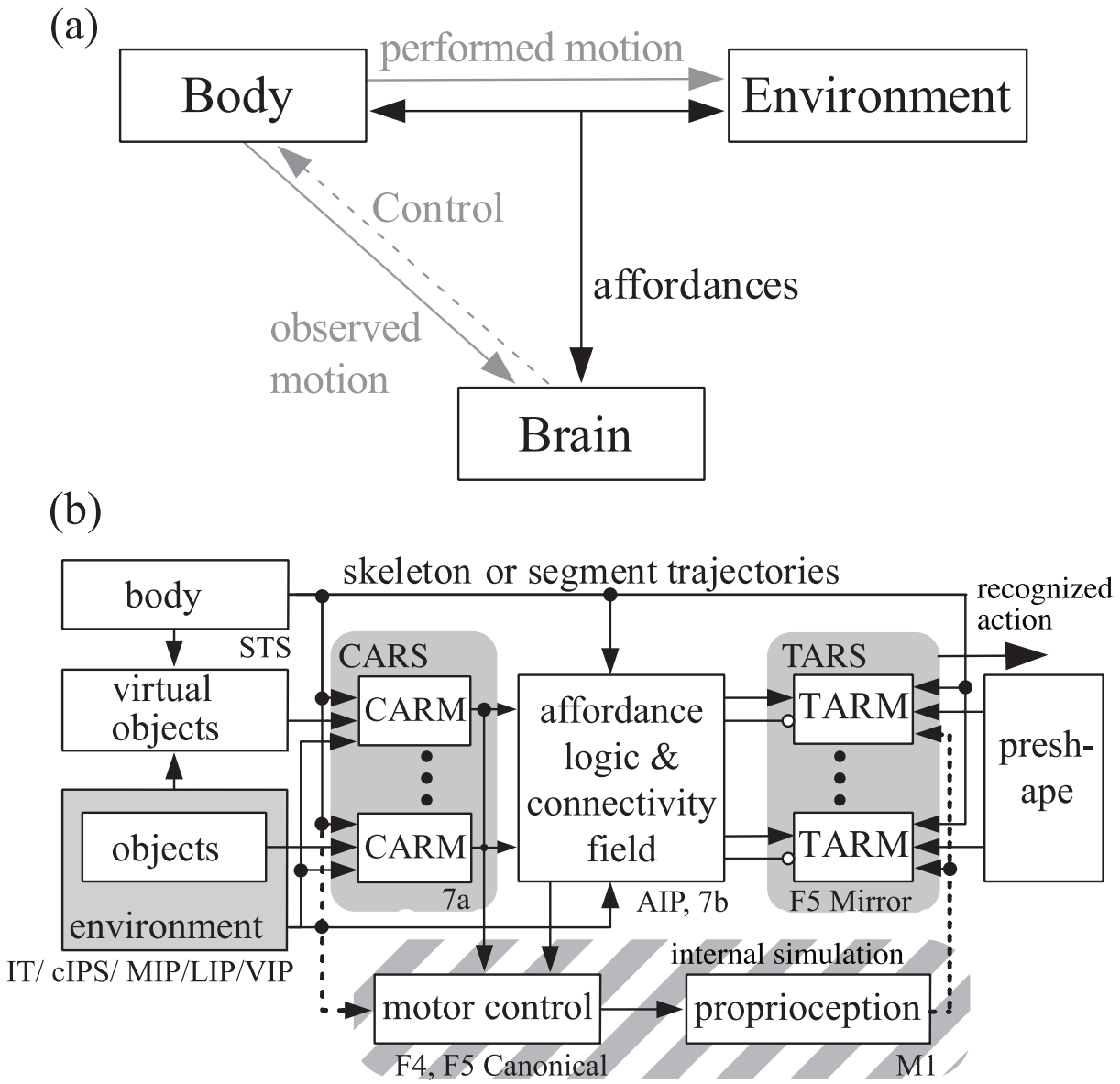


**Figure 2.2:** (a) Illustration of the interactions between brain, body and world or environment based on the contextual affordance input as well as the trajectory input information. (b) Connection of contextual- and trajectory based action recognition system.

vated within situated embodied embedded cognition) is the combination of understanding the action possibilities of the goal-directed objects to which manipulations are aimed at (motivated by concepts of affordances), and the spatiotemporal comparison of observed movements to memorized experiences of movement classes (motivated by concepts within biological motion perception). The memorized experiences of movements that are shown in the preshape block in Fig. 2.2(b) can be dynamically expanded into internally simulated movement generation (motivated by threshold control theory). Information from the environment and observed agents are projected onto the observers body, this processing happens in the *body* block. This block has functional equivalence in the superior temporal sulcus (STS) region in the brain that is responsible for motion detection. The perception of the environment occurs in the *objects* block and has functional equivalence in the superior temporal sulcus (STS) region in the brain. Our basic hypothesis within this block is that the movements of the actor are seen as the observer's own and the objects around the actor are also projected around the observer [106].

When the actor's movement is directed towards an object, the contextual action recognition system (CARS) uses information of optical flow (speed and direction of e.g. the wrists/pelvis) and predicts the object that is to be manipulated. The CARS block resembles the function of the 7a area that is thought to be responsible for the analysis of hand-object spatial relationships, additionally the information of optical flow models the fynamica interaction between the optic flow sensitive regions V3A, V6, and hMT+ and the hippocampus, retrosplenial cortex, posterior parietal cortex, and medial prefrontal cortex [144].

The available affordances of objects (reasoned by an affordance logic block) gives an idea of what the meaning of that movement is, this is presented in section 3.6. This block resembles the function of the 7b and anterior intraparietal (AIP) areas that are responsible for determining the association between object and end effector and extracting object affordances respectively [145, 146]. The trajectory action recognition modules (TARMs) load a memory of a similar movement experienced/learned previously with the help of the preshape block, and compare the observed movement to that memory. Each TARM represents a specific action, and thus several of these modules are combined to make the TARS.

Internal simulation could also be attained using a dynamic motor control block (shown in the hashed block in Fig. 2.2(b)), rather than long-term memories currently stored in the preshape block. The internal simulation of reaching movement generation is discussed thoroughly in Section 4. If the action memory is finally validated, then this action is actually being observed and the system is reset to wait for the next movement.

There exists many options to implement the architecture proposed in Fig. 2.2(b), we chose dynamic neural fields to model the different subsystems as compared to purely consistency/probabilistic approaches DFT is highly expressive, less sensitive to noisy observations and allows modeling priors as discussed in section 1.2. The AUA and its connections will be illustrated in the different upcoming chapters in this thesis.

## 2.2 Conclusions

In this chapter we presented a novel action understanding architecture (AUA) that, in comparison to the state-of-the-art methods discussed in section 1.2, combines for the first time findings in situated embodied embedded cognition, affordances and biological motion perception as well as threshold control theory into a coherent, dynamic cognitive architecture. The action understanding architecture computationally models the MNS using dynamic systems theory and dynamic field theory. Explicitly the AUA is decomposed into two directions to achieve proximal intention understanding. The top-down direction is discussed next in chapter 3. While the bottom-up direction is presented in chapter 4. Finally we describe how the ideas in chapter 3 and 4 are linked to distal intention understanding in chapter 5.

# 3 Action understanding from observed movements and context

Action understanding (AU) can be defined as the task of classifying a stream of human-related multimodal data (motion, speech, etc.) into semantic terms suitable for influencing the future intelligent behavior to support the human agent in a meaningful manner. Intelligent systems face several challenges in AU, these include the spatio-temporal variation within a class of actions, as well as interclass and intraclass variation in how persons perform actions. Spatio–temporal variation here refers to the fact that similar actions might vary in duration and path followed across agents and trials. Another major challenge is the large search space of actions available to an agent in any environment [147, 148]. To be able to understand an action, intelligent systems are required to solve the spatio–temporal variation problem by a robust trajectory recognition system, and the large search space problem by incorporating the context of the action.

In our quest towards an end-to-end biologically-inspired architecture for human action understanding, we present two systems that address the aforementioned challenges and that we hypothesize to be central for the task of AU. This is presented in Fig. 3.1, where we extract the action understanding systems from the overall AUA architecture as illustrated previously in Fig. 2.1. The two systems are inspired by processes observed within human behavioral studies, as discussed previously. The main challenges addressed in this work are the context understanding of an observed movement and the trajectory parsing of the movement. Additional secondary challenges addressed in this work include how the context understanding interacts with trajectory parsing, and how visual information of motion can be used as an input in a manner consistent with the complete system. The work presented is inspired by definitions within the embodied situated cognition stance. The context understanding is based on definitions of affordances, and the trajectory parsing follows ideas of biological motion perception. The embodied situated cognition stance, the concept of affordances as well as the ideas within biological motion perception have been presented in Chapter 2. The different novel systems are modeled using the cognitive framework developed within dynamic field theory (DFT).

As there exists many ways to understand actions an agent might perform, this renders a large search space for an AU system. We address this problem of context understanding by modeling three processes into a contextual action understanding system. Firstly, we model the detection of goal-directed movements. Secondly, we model the shifting of attention from joints (end-effector) to objects in the line of action of the joint movements. Finally, we model the context understanding of the movement given the affordances of the objects towards which the attention was shifted. The term affordances relates to the action possibilities that an object might allow [88]. The context of the movement based on affordances is understood using a novel contextual action recognition system (CARS), as illustrated in Fig. 3.1. The function of the CARS is to pick the most relevant subset

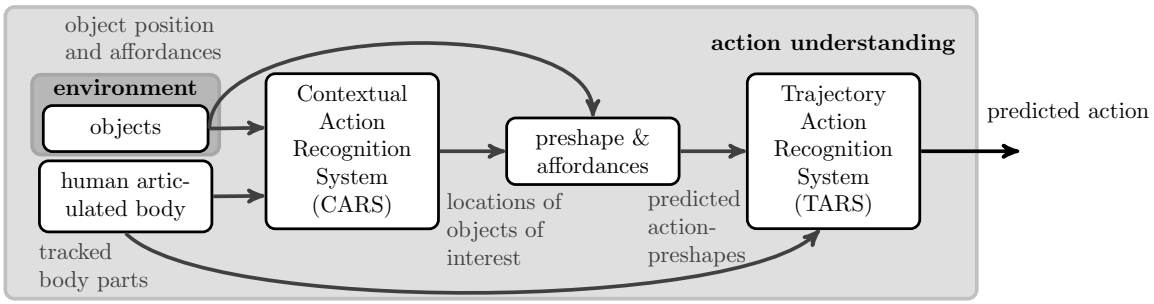


**Figure 3.1:** The action understanding system presented in this chapter and their connections

of templates, in a pre-learned database of templates that represent movement features. A separate affordance logic block aids in this selection, and is further discussed in section 3.6.

The second AU challenge addressed in this work is trajectory parsing. This online comparison is performed within the trajectory action recognition system (TARS), as illustrated in Fig. 3.1. This system allows for spatiotemporal variation between the template and the observed motion, and outputs a positive result if they are matched. The transformation from visual input of joint movements into biologically-inspired features for comparison purposes is considered and further discussed in section 3.3. The contribution in this chapter lies in presenting TARS and CARS and their integration within the AUA for the task of top-down action understanding (proximal intention understanding) and formalizing them within dynamic systems theory. We depend on the mathematical formalization of dynamic field theory to model the *cognitive building blocks* in the AUA.

## 3.1 Related work

Biologically-inspired AU architectures are usually presented as computational models for Mirror neuron systems. Examples of such computational models are the MOdular Selection And Identification for Control (MOSAIC) model [73–75], and the Hierarchical Attentive Multiple Models of Execution and Recognition (HAMMER) [76, 100] that were primarily developed for imitation and later extended for action recognition [77, 78]. The Mental State Inference (MSI) model [101] as well as the Recurrent Neural Networks with Parametric Bias (RNNPB) [79–81] and the Mirror Neuron System 2 model (MNS2) [82], all model the MNS for the purpose of AU.

Other cognitive action understanding systems in literature that do not explicitly model neuronal processes include the work of Yang et. al in [149], in which context-free grammar and parsing algorithms were proposed for the understanding of goal-directed manipulation actions. The architecture uses a depth image to obtain an articulated model of the user's end-effector as input. The depth image is also used to obtain information about the labels of the objects and their position on a table-top. The hand model is transformed into a set of bio-inspired features which then are used to classify the grasp type using a Naive-Bayes

classifier. Additionally, hand tracking produces trajectory profiles for trajectory-based action recognition. The classes were obtained by using a combination of principal component analysis and k-means clustering. An attention model, comparable to our proposed CARS, makes use of bottom-up processes to identify potential fixation points in an image frame as well as top-down attention mechanisms based on the hand location. The spatial intersection of fixation points and the hand location shifts the attention towards an object for monitoring. A new observation consists of a triplet: subject, action, objects. A context-free manipulation action grammar is proposed and using parsing algorithms, a tree group is updated when a new observation is given, and dissolved automatically. The tree output can be then passed to an intelligent agent for decision making and further operations. Yang et. al do not utilize the concept of affordances in their work. Furthermore, in comparison to the work presented in this thesis, the model presented by Yang et. al is not grounded by concepts in cognition, nor uses a common cognitive framework for the modeling of the different subsystems.

Other work presented by Aksoy et al. in [150], describes a complex action by combining descriptors that analyze the relation between the series of manipulated objects with action-related information such as trajectory segments, pose and object information. The combination of these descriptors allows for a better comparison of observed actions, and therefore enriches the meaning behind each action. The work describes how observed actions are either understood as new actions or known ones. The new actions are accommodated for by creating a new novel schemata, while the known ones, if slightly different are assimilated with the representative schemata. Compared to the work proposed in this thesis, Aksoy et al. do not explicitly utilize action trajectories nor contextual information when preforming the task of action understanding.

A neural dynamic approach for parsing a sequence of actions was recently presented in [85] by Lobato et al. The authors present a neural-dynamic architecture that is capable of detecting and representing an even of actions, namely reaching/grasping/dropping objects on a table-top scenario. Trajectory recognition was not considered but rather three-dimensional positions of hands and objects were used to calculate whether the hand was approaching the object or not. The overall architecture is capable of memorizing a string of actions for overall action understanding. Similar work was also presented within neural fields in the work of Bicho et al., in which the focus was on integrating verbal and nonverbal communication in a joint-assembly task in which the sequence of actions were given [84]. In contrast to the work presented by Lobato et al. and Bicho et al. we extend the application area of DNFs towards representation and recognition of temporally extended actions using context and movement information. Furthermore, while the work presented in Lobato et al. and Bicho et al. deals with only table-top scenarios, we present systems that are general enough for understanding locomotion, manipulation and actions in free-space. To the best knowledge of the authors, the systems developed to address the aforementioned challenges of AU are novel within DFT.

Overall, the AU architecture in this work presents, for the first time, a novel predictive system within DFT that models attention-shifts and pairs up with a trajectory parsing system in a second step. The trajectory parsing system takes account of spatial as well as temporal variations that are usually problematic when understanding actions. Special

attention is given on how objects and the environment are integrated in the overall architecture and on how they can drive action understanding. Compared to the state-of-the-art presented in this section, the AU architecture in this work combines both context recognition and trajectory recognition rather than opting for either contextual recognition alone or trajectory parsing by itself for the task of action understanding. Furthermore compared to the state-of-the-art we explicitly model objects and their affordances.

## 3.2 Dynamic field theory

At the core of the modules that make up TARS, CARS, the affordance logic system, and the internal simulation of movement generation blocks are decision making processes that dynamically evolve with the tightly coupled input. These systems all require cognitive abilities to achieve their functions. The CARS as well as the internal simulation block requires the cognitive abilities of object detection, motion prediction and goal selection. The TARS, on the other hand requires feature detection and comparison. Finally, the affordance logic system requires the abilities of dynamic selection and long-term memory. In the following we present the dynamic cognitive framework of DFT, and elaborate on the building blocks that are used within the different systems in this work.

### 3.2.1 Dynamics and instabilities

Dynamic field theory (DFT) provides the mathematical and theoretical framework, that builds on dynamic neural fields (DNFs), to model the embodied, situated view of cognition [2]. DNF is a cognitive mathematical model of the dynamic neuronal activation on a population level. It describes decision making inspired by the pattern formation within the cortical neural populations. It is the stable states (localized-bumps) that dynamically evolve (and devolve) in time, given dynamic perceptual input into the neural fields, that provide a unit of representation. These units of representations are a function of the complex interaction between the neurons in the population, and are the basic units to describe cognitive properties within the neural fields. The strong recurrent connections between these neurons produce patterns that model detection, selectivity and working memory. The dynamics are mathematically described in the following integro-differential equation that was initially proposed in [151]

$$\tau \dot{u}(x,t) = -u(x,t) + h + \int f\big(u(x',t)\big)\omega(x-x')dx' + S(x,t) \tag{3.1}$$

$$\omega(x-x') = c_{\text{exc}} \exp\left(\frac{(x-x')^2}{2\sigma_{\text{exc}}^2}\right) - c_{\text{inh}} \exp\left(\frac{(x-x')^2}{2\sigma_{\text{inh}}^2}\right) \tag{3.2}$$

$$f\big(u(x,t)\big) = \frac{1}{1+\exp\big(-\beta u(x,t)\big)} \tag{3.3}$$

in which the activation of the field $u(x,t)$, as given in (3.1), describes the activity over the metric dimension $x$ at time $t$. Here, $x$, represents a behavioral dimension that the

underlying neuronal populations respond to. This behavioral dimension corresponds to a space of features and properties that the neurons encode. The amount of activation of the field $u$ can then be understood as the presence or lack of information about a space of features along the behavioral dimension $x$. The time scale $\tau$ describes the relaxation of the field and the negative constant $h$ defines the resting level of the field. The term $S(x,t)$ describes an external input to the neural field. The integral term conveys the interaction between different field locations. Sufficiently activated field locations contribute to the neural interaction by way of the interaction kernel $\omega$ given in (3.2). That is, the output of the sigmoid function $f$, given in (3.3), modulates the activation contribution, given by $\omega$, to other field locations. The sigmoid function with slope $\beta$ is shown in Fig. 1(a). An example of an interaction kernel $\omega$ could be a symmetrical homogeneous interaction kernel with short-range excitation (determined by the amplitude factor $c_{exc}$, with an area of influence determined by $\sigma_{exc}$) and a long-range inhibition (determined by the amplitude factor $c_{inh}$, with an area of influence determined by $\sigma_{inh}$) [152]. Four interaction kernels are shown in Fig. 1(b). The choice of the kernel is usually dependent on the kind of cognitive behaviour to be shown.

Analysis of (3.1) leads to the characterization of different states. In the following we describe these states and their significance [2, 151, 153].

In the case where no external input is present, the field has constant level of activation, equal to the negative resting level $h$, along the field dimension. This non-peak attractor state, referred to as a *sub-threshold solution*, maintains its stability under weak external input $S(x,t)$. In the case that the activation level exceeds a threshold level where the lateral interaction $\omega(x-x')$ and the non-linearity $f(u(x',t))$ become active, the neural field is driven in a different dynamic domain. In this case, a *localized peak* develops in the field due to the increase of activation in the field locations where the external input is the largest [153].

Starting from a *sub-threshold solution* a *detection instability* can occur in which peaks evolve at positions of sufficient activation. These positions were successful at accumulating enough activation to overcome the activation threshold of the field. In other words, the probability, or stimulus strength of that feature-space at that position was significant. It is possible to have enough activation at several locations within the field and develop localized activity peaks that provide a representation of the existence of the underlying feature-space values. The interaction kernel labelled with number 2 in Fig. 3.2(b) is an example of a kernel that is used for the detection instability. Furthermore, an example is given in Fig. 3.2(c) and Fig. 3.2(d). Figure 3.2(c) shows an input at a feature position with stimulus strength (solid grey line) that is not sufficient enough to activate the complete field (dashed black line), therefore no information is represented in that field. In Fig. 3.2(d) the stimulus is strong enough to produce a bump in the field, giving a representation of the existence of information which can be read out for further processing. The interaction kernel used in this example is the second kernel in Fig. 3.2(b), and that is shown by the fact that the output takes shape of the kernel around the input's location. Another example of the *detection instability* is illustrated in Fig. 3.3(a). These two positions received enough input such as to show stable peaks in the activation field and appear as an output.

The second case that can be observed is known as the *selection instability*, in which

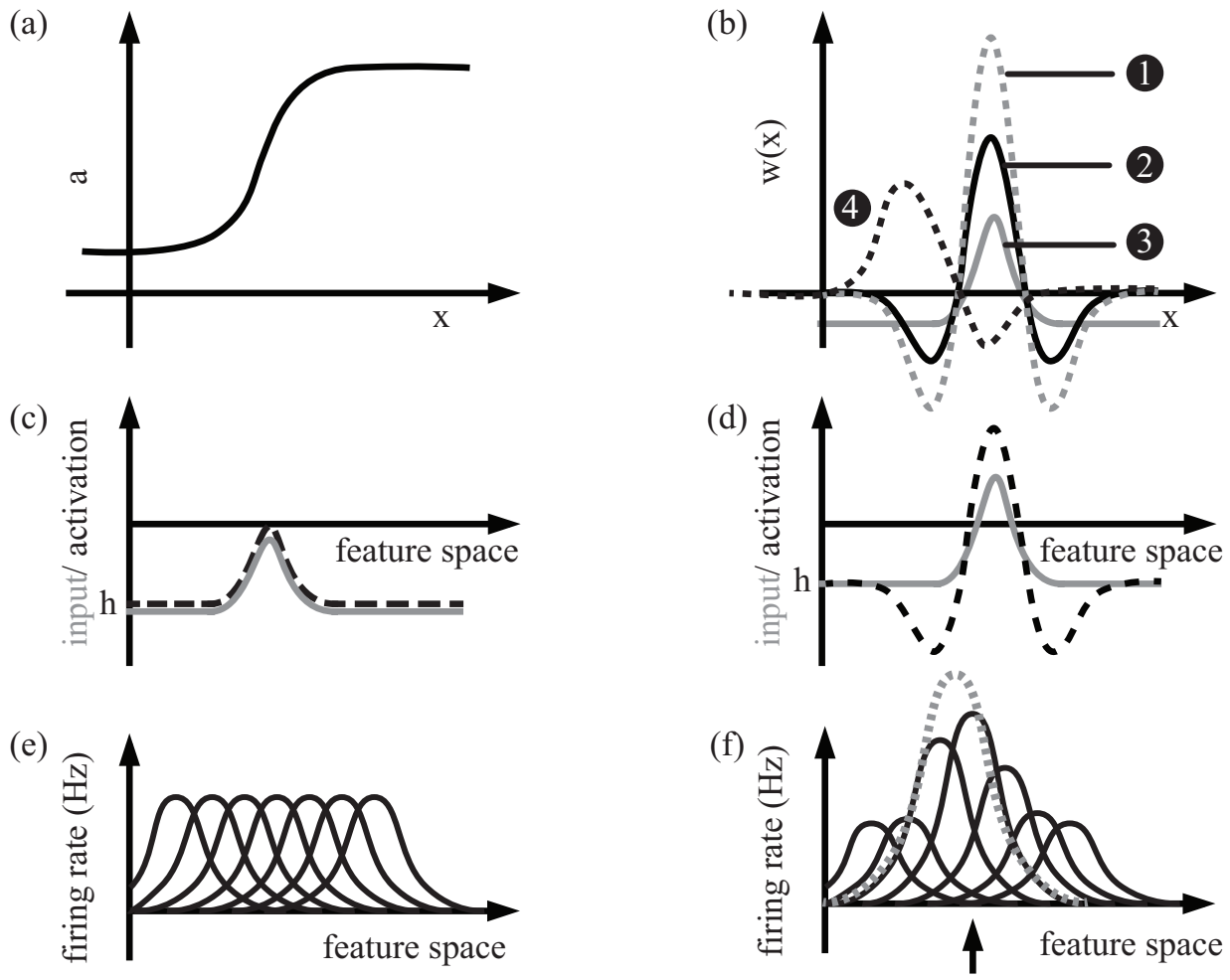


**Figure 3.2:** Dynamic neural field components and distribution of population activation. (a) The sigmoid function. (b) Examples of the interaction kernels: 1) An interaction kernel used to model a working memory instability. 2) An interaction kernel used to model the detection instability. 3) An interaction kernel used to model the selection instability. 4) An interaction kernel used to produce a traveling wave transient state. (c) Subactivation solution within the DNF. (d) An activated field with a stable solution around the input. (e) A group of tuning curves spanning over the features space with no response to a stimulus. (f) The distribution of population activation solution (dashed grey line) to a feature input (indicated at position of the black arrow).

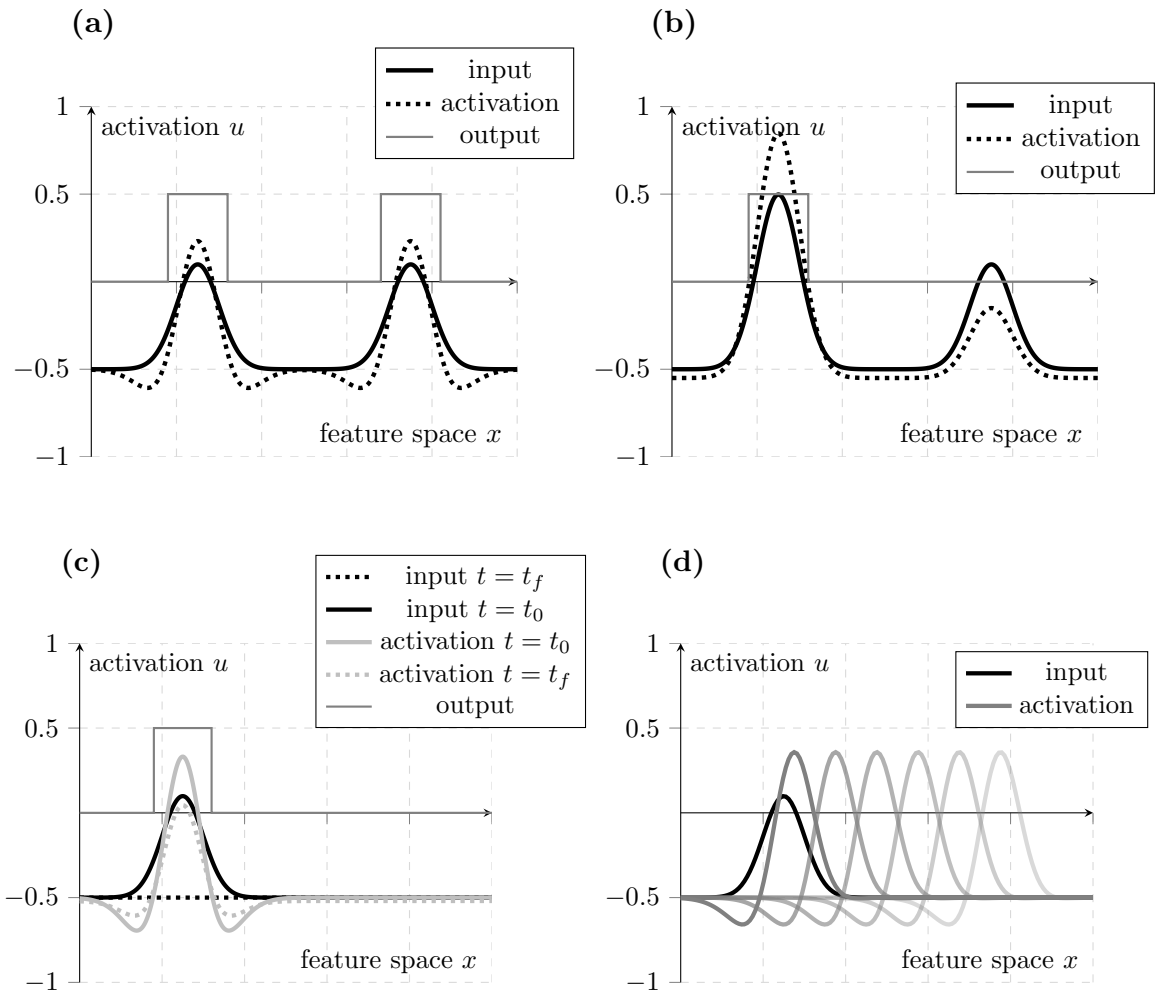


**Figure 3.3:** (a) An example of the detection instability in which the field is activated with a stable solution around the input. (b) An example of the selection instability in which the field is only activated with a stable solution around the strongest peak in the input. (c) An example of the memory instability in which the field is activated with a stable solution around the peak in the input at time $t_0$, and remains activated even when it dies out at a later time $t_f$. (d) An example of the traveling wave instability in which the field is swept with an activation peak starting with the initial position defined by the input location.

only one stable peak can evolve in the field and any subsequent activation at different locations in the field are inhibited. Only large enough activation (one that can accumulate enough activation to overcome the global inhibition induced by the first peak as well as the field's threshold) can appear and inhibit the original stable peak. When two positions of a quiescent field, that shows the selection instability, show activation at the same time, the one with higher activation develops the peak, and inhibits the other positions. Thus, showing a selection of two options. In the case when two or more positions have similar

activation values in a field showing the selection instability, noise in the field plays a role in selecting one of the locations to develop a peak. Positions where peaks of activations are developed are meaningful as units of representation, and they indicate the existence of an important underlying value given the selected feature-space. The interaction kernel labelled with number 3 in Fig. 3.2(b) is an example of a kernel that is used for the selection instability. An example of the instability itself is shown in Fig.3.3(b), where only the strongest of the input peaks is allowed to transpire at the output, while the second peak is inhibited. It is interesting to note that in the case where the peaks at the input are exactly the same amplitude, noise in the field usually helps in making a decision between the two feature values.

An important case that can also be observed in the analysis of (3.1) is one that models *working memory*. This instability can be observed when sufficient interactions are existent in the field to sustain an input even when these inputs cease to exist. This instability aids in modeling decision/features that were made/observed in the past. The interaction kernel labelled with number 1 in Fig. 3.2(b) is an example of a kernel that is used for modeling working memory instability. The working memory instability ultimately leads to a self-sustained activation that represents working memory. An example of this process is shown in Fig.3.3(c) where the input at the initial time step $t_0$ showed a peak (solid black line) that later disappeared at time $t_f$ (dotted black line). This however translates to a slightly lowering of the activation field from $t_0$ (solid gray line) to $t_f$ (dotted gray line).

In the same way that peaks can be stabilized, they can be unstabilized by introducing a negative input to the peak position or by reducing the excitation there. This is referred to as the *reverse detection instability* or *forgetting instability*.

### 3.2.2 Dynamic neural fields and distribution of population activity

These elementary forms of cognition (detection, selection and working memory) discussed so far operate on patterns of neural activity representing sensory stimuli or motor control information. To establish this link between neural activity and external stimuli and internal motor actions, the concept of neural tuning is commonly used. The way a DNF can be related to an activity of neural population is through the concept of Distribution of Population Activity (DPA) [154]. The basic idea is that every neuron in a stimulus-sensitive population is characteristically sensitive to a specific value of that stimulus. The neuron then contributes to the population with a functional form that is usually centered around the stimulus value it is characteristically tuned for. This function of how each neuron responds to a stimulus, and which represents the average firing rate is called a tuning curve. An example is given in Fig. 3.2(e), where 7 (Gaussian approximated) tuning curves span the feature space. The sum of tuning curves, over all neurons in a population, weighted by each of their mean firing rates (understood as the activation level in DNFs) explains the activation of the population of neurons given a stimulus. The DPA is calculated using the following equation

$$\mathrm{DPA}(x,t) = \Big(\sum \mathrm{tuning}_x \times \text{firing rate}(i,t)\Big)/N, \tag{3.4}$$

where $N$ is the number of neurons whose tuning curves at positions $x$ are multiplied by

their activation (firing rate), at time $t$. The final result of a DPA is shown in Fig. 3.2(f) where given a feature value, several neurons respond with their own firing rate (solid black lines). The final result is visualized with the DPA (dashed grey line). The lateral interaction between those neurons by their activations give way to dynamics within the field as discussed in section 3.2.1.

### 3.2.3 Learning within dynamic field theory

The input that might be used in a field could be processed into a decision, or it could be used to maintain a memory trace over the feature space as a simple form of learning. Learning in DNFs can be understood using what is known as a preshape or a memory trace [2, 153]. It is a formalization that allows retention of stimuli information in long-term memory form. The memory trace, which equation is

$$\begin{aligned}\tau_l \dot{P}(x,t) =& \lambda_{build}\Big( - P(x,t) + f\big(u(x,t)\big)\Big) f\big(u(x,t)\big) \\ & -\lambda_{decay} P(x,t)\Big(1 - f\big(u(x,t)\big)\Big),\end{aligned} \tag{3.5}$$

takes input from a DNF with $u(x,t)$, and builds up activation $P(x,t)$ towards the attractor solution (activation-bump) from the input with a time constant $\tau_l/\lambda_{build}$ that is slower than the underlaying DNF. This built up information is lost at a rate that is even slower, $\tau_l/\lambda_{decay}$, when there is no activation present and models long-term memory. Here, $\lambda_{decay}$ and $\lambda_{build}$ are the rates at which the preshape decays or builds up. The constant $\tau_l$ is the time constant of learning in the preshape field.

The memory trace is used as a non-activating input to other decision DNFs. It thus acts as a sub-threshold solution to the field, preshaping (biasing) the locations in the DNF and allowing for easier activation if an input at those specific positions are later introduced into the preshaped DNF. Alternatively, a positive homogeneous input to the field (also known as a boost input) would activate those sub-threshold activations in the field.

### 3.2.4 Comparisons within dynamic field theory

It is essential to compare different DNFs (e.g. memory trace field and perceptual fields that hold the current input from the environment) to model the recognition of specific, meaningful features in the environment. In addition to the recognition of features in the environment, comparison is essential to obtain a level of satisfaction regarding the completion of an action command that was sent to an intelligent system. To that end, the concept of condition of satisfaction (CoS) was introduced to check if a field had reached a predefined level of activation on one or more feature values [155–157]. A CoS consists of three components: an action/preshape field, a perception field and CoS field. This is further illustrated in Fig. 3.4(a,b), where Fig. 3.4(a) shows the case when the CoS does not detect a match between the input from the perception field and the pre-activation from the action field. In the general case where an intelligent system is a part of the action/perception loop, the action field represents a desired action to be fulfilled. This action field effects the intelligent system by providing set points for the satisfaction of

the action. The level of satisfaction is dynamically calculated in the CoS field where the action/preshape field is constantly compared against the perception field. In contrast, in Fig. 3.4(b), the stimulus in the prescription CoS field matches the learned preshape in the action field and a decision bump appears in the CoS field, prompting an activation to be detected. The action and perception fields are an input to a CoS field that gives an indication if there is a match or not. The CoS field is augmented with a node that gives a logical value of detection or not as shown in Fig. 3.4(a,b).

In our work we expanded the concept of CoS to accommodate the fact that human motion can not be represented by a single lower limit configuration but rather a range around a configuration that all could indicate an informative part of a moment (e.g. arm configuration for a handshake). The more general idea here is to expand CoS to using a range $R$ which we refer to as range of satisfaction (RoS). In this RoS formulation, the action field is used as a pre-activation for both the upper and lower CoS fields. The upper CoS field is also pre-activated with a global negative input with a value that equals the desired range $-R/2$. In the same manner the lower CoS field is pre-activated with a global positive input with a value that equals the desired range $R/2$. This allows the detection of a feature in the metric space earlier in the lower CoS. Furthermore, it would allow for detecting if the observed features was within a specific range of activation levels. For example, if the as the upper CoS field would activate (detecting that the feature is above this value) the RoS neuron would deactiveate. This deactivation aids in checking for the next feature which is an important function when comparing a time-continuous movement such as a reaching motion. An illustration of the function of the RoS is shown in Fig. 3.4(c).

### 3.2.5 Prediction within dynamic field theory

So far, we have discussed several cognitive properties of DFT that can be used as building blocks in any cognitive architecture. We have expanded on the function of CoS to better suite the application of action recognition. However, the *prediction* capabilities within DFT are rather limited. Yet, they are vital in an online dynamic application of action understanding. That is why in the following we argue for the need of a mechanism that is able to look ahead in a feature space and provide predictive capabilities. A transient state that could provide these capabilities can be found in *traveling waves*.

Dynamic behavior of traveling activation pulses in the cortical sheets of the brain had been observed [158, 159] and modeled in DNFs [151]. Such dynamics in the neural field has been exploited for intelligent behavior generation [160] and for influencing robotic arm control [161]. Further research on traveling bumps in neural fields have since been conducted and solutions for their collision has been modeled [162].

In the following, we provide a mathematical derivation of the wave transient, for a complete derivation we direct the reader to the work presented in [160]. The initial equation is the dynamic field (3.1). Now, it is assumed that the field, which is used to generate the moving peak, has a local excitation (peak solution) (see $a$-solution [151]) and that the input signal $S(x,t) \equiv 0$. In order to generate movement, an asymmetric interaction kernel $w_a = w_e + w_0$, consisting of a symmetric kernel part $w_e$ overlapped with an asymmetric function $w_0$, is developed. The shape of the function $w_0$, which is necessary to generate

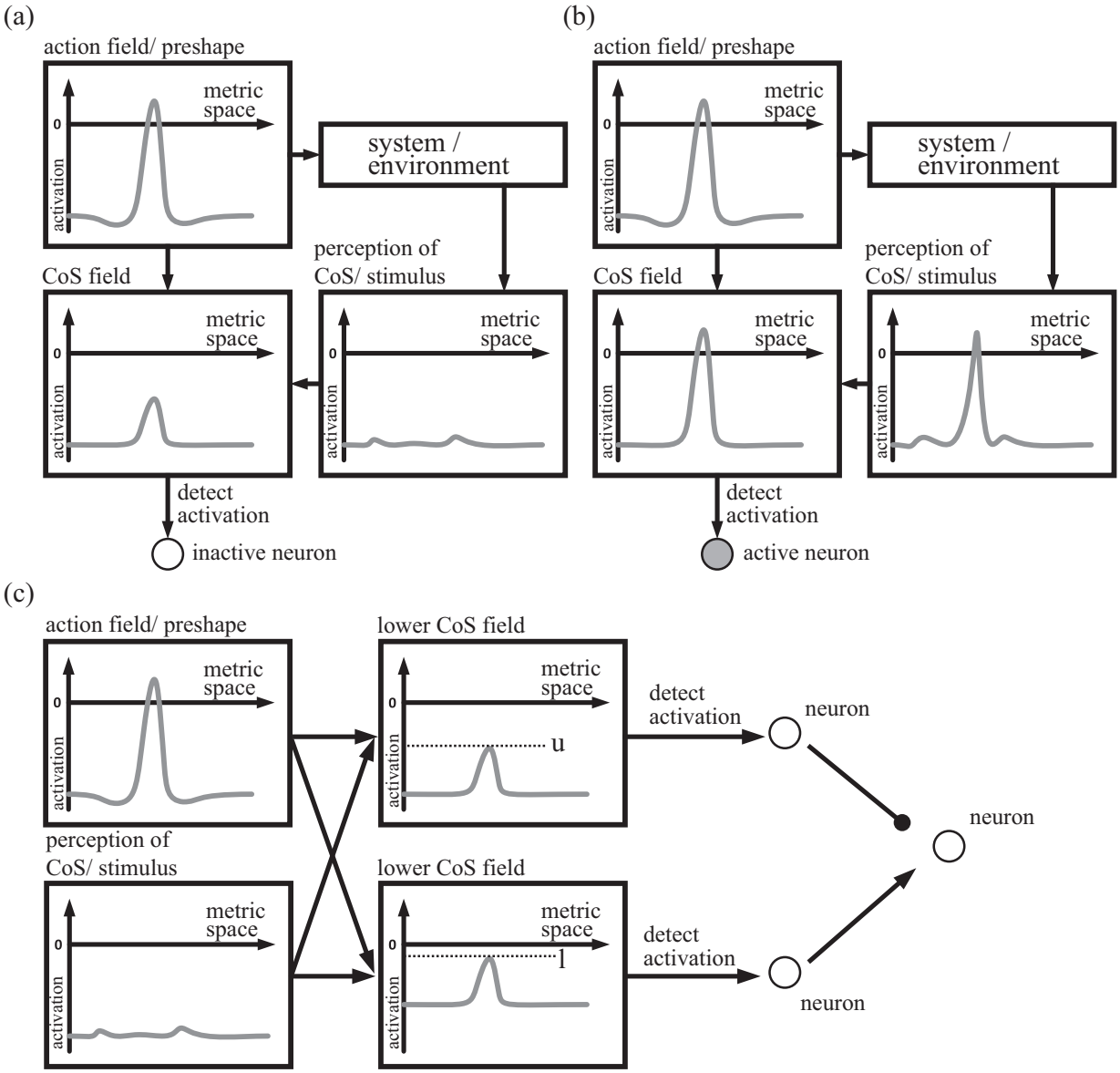


**Figure 3.4:** Illustration of the *Condition of Satisfaction* (CoS) approach. (a) Preshaped *CoS field* without corresponding input from the *Perception of CoS*. (b) Matching input resulting in an activation in the *CoS field*, which can be used to activate a neuron. (c) Illustration of the *Range of Satisfaction* (RoS) concept. The preshape and stimulus are used as input for both, the *lower field* and *upper field*. Further, the range boundaries are illustrated within the fields.

the movement, is determined in the following. By taking the previous assumptions as well as the asymmetric kernel $w_a$ into account, the dynamic field equation (3.1) results into

$$\tau \dot{u}(x,t) = -u(x,t) + h + \int f(u(x',t)) w_a(x-x') \mathrm{d}x'. \tag{3.6}$$

Assuming that there is an initial stable peak solution within the field at time $t = 0$, meaning $U(x) = u(x, 0)$. Thus, the excitation distribution, for any time instance $t > 0$, is given by

$$U(x,t) = U(x + \int_0^t v(\eta) \mathrm{d}\eta), \tag{3.7}$$

whereby $v(t)$ represents the velocity of the moving peak. Equation (3.7) can be used to calculate an equation providing information about the relation of $w_0$ and $v(t)$. Plugging (3.7) into the right side of (3.6) we obtain

$$\tau \dot{u}(x,t) = \tau U' \frac{d}{dt}(x + \int_0^t v(\eta) d\eta) = \tau U' v(t). \tag{3.8}$$

Plugging (3.7) into the left side of (3.6) results in

$$-U + \int_{-\infty}^{\infty} w_a(x,y) f(U(y)) dy + h = \int_{-\infty}^{\infty} w_o(x,y) f(U(y)) dy, \tag{3.9}$$

given the knowledge about the equilibrium solution under $w_e$ is

$$\int_{-\infty}^{\infty} w_e(x,y) f(U(y)) dy = U - h. \tag{3.10}$$

Finally, combining the left and right side we obtain

$$\tau U' v(t) = \int_{-\infty}^{\infty} w_o(x,y) f(U(y)) dy. \tag{3.11}$$

It can be seen that the relation between $w_0$ and $v(t)$ is not as simple as may be expected. However, by setting $w_0 = p(t) w'_e$, and given the knowledge of (3.10) the complex relation simplifies to

$$v(t) = \frac{p(t)}{\tau}. \tag{3.12}$$

Here, $p(t)$ is a time-depending factor and $w'_e$ the spatial derivation of the symmetric kernel part. Now, (3.12) allows to control the speed of the moving peak, whereas the shape of the kernel influences the direction. An example of this kernel is shown in Fig. 3.2(b) (black dashed line labelled with number 4). The traveling wave of activation itself is

illustrated in Fig. 3.3(d). We utilize this instability extensively for two main tasks. Firstly, prediction purposes where we would project current values forward in the feature space given perceptual information. Secondly, scanning the field in one direction for comparison purposes as a time keeping method.

## 3.3 From moving bodies to biologically motivated features

An observer perceives an acting agent as well as the environmental state (in terms of the object in the actors vicinity and how he interacts with them) to infer about this actor's mental states of actions, (action) plans and intentions. In the following we present our decisions for modeling the perception of the moving body in a manner consistent with what is presented in neurally-focused studies. Specifically, we discuss our choices for how the body is perceived, what are the required transformations, what are the features extracted for the classification task and finally, how these features can be used in a neural population approach that is compatible with the DFT.

### 3.3.1 Embeddedness and egocentric coordinates

Complying with the embeddedness concept, the observing agent projects the skeleton of the perceived acting agent on his own. Studies have shown that biological motion might be perceived by projection on egocentric coordinates and this might aid guiding behavior and understanding [106, 163–165]. Similarly studies in neuroscience and mirror neurons have shown evidence of egocentric action understanding [13, 166]. Therefore, the first step in our action understanding architecture is the projection of the actors frame of reference onto the observer's frame of reference. An illustration of the desired transformation is shown in Fig. 3.5(a).

### 3.3.2 The body joint extension and projected relative angle features

Moreover, when observing an acting agent, the observer's visual system focuses on the joints of the acting agent [167]. Out of all the joints, studies have shown that there was a focus on the upper body joints, namely the head, left and right wrists [167]. In our work, we have also integrated the pelvis joint as well as the left and right ankle joints, which are also essential to the understanding of locomotion actions.

The positional information extracted from these joints are then projected onto the transverse and sagittal planes of the observer (after the whole skeleton of the actor had been transformed onto the observers body frame) [168]. We implemented these transformations mathematically with no regard to possible neural mechanisms behind it, however transformation-capable DFT systems were discussed also in literature [169] that could also be extended to egocentric coordinate frame transformations for the purposes of motion perception.

Following from the previous paragraphs, we decided for two *feature types* to be extracted from the projected view for the purpose of action recognition. The first feature type is the *Body Joint Extension*. It is a non-circular feature (linear feature space, 0-100%) which

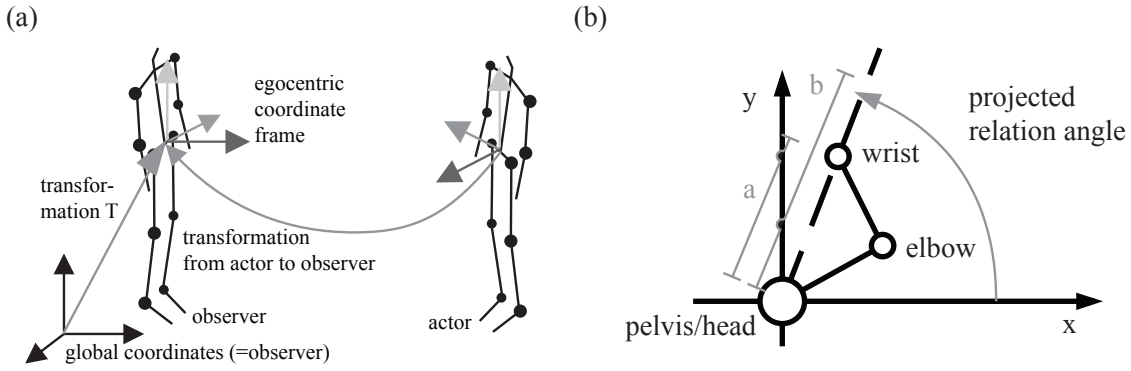


**Figure 3.5:** (a) Transformation into egocentric coordinate frame. (b) Illustration of the projected relation angle between wrist and pelvis with respect to the $xy$-plane. The ratio $a/b$ represents the extension percentage of the arm and is the second used feature.

measures the percentage extension between two joints that are not shared by the same bone. For example the wrist-shoulder body joint extension equals 100% when the arm is fully extended, and 50% when the elbow joint makes a 90-degree angle. We used average human dimensions as given in [170], and calculated the full extension values for a 1.8 meter male for simplification. The second feature type is the *Projected Relative Angle*. It is a feature with a circular feature space (0–360-degrees) which measures the projected relative angle between two joints. Both feature types are described to be view-centered as they are dependent on the position of the viewer relative to the perceived objects (different joints). View-centered representation is one of two major types of descriptions (the other being object-oriented representation) suggested to model the ability of extracting information from the projection of 3D object on retinal images [171–173]. Overall, and given different joints that could be used logically, we propose 39 different features that are calculated for any motion within this work, this accounts for different joints and different plane projections. The full list of features is given in Appendix A. Several combinations of these features can be made depending on the class of the action and the level of joint involvement in that specific movement. Within our work, the temporal evolution of these features are learned from multiple examples in order to compose a memory or an experience that preshapes a comparison dynamic neural field. A memory is learned for each class of action and can be thought as a memorized trace to which the features extracted from the observed action is compared against within the TARM.

### 3.3.3 Formulating features as distribution of population activation

The previous features should be provided as an input to the DFT system in a manner that is neuronally consistent, using formulations within DPA, this process is illustrated in Fig. 3.6. Specifically a pool of receptor cells (neural population) is modeled for each feature type. Each population responds to a specific feature. Each neuron within the population has a specific tuning curve that is centered around a value that it is most sensitive to.

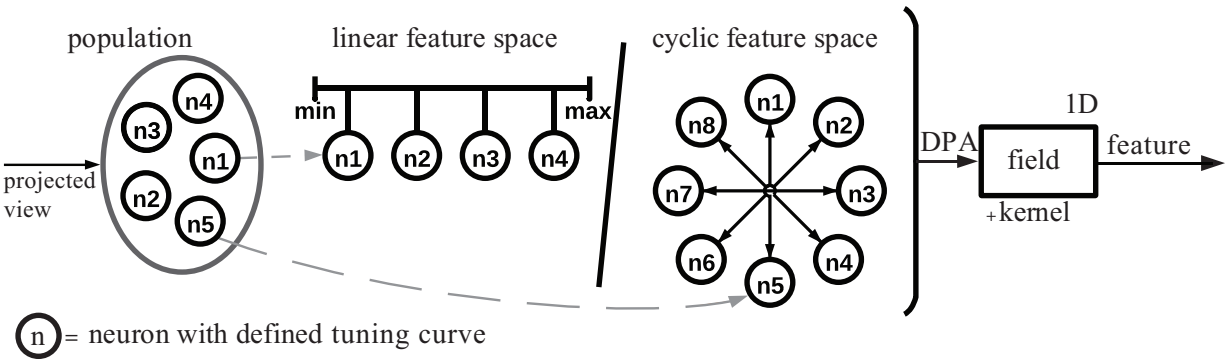


**Figure 3.6:** This figure illustrates how features are represented as input stimuli for the overall system (for a linear feature space (left) and cyclic feature space (right)). The tuning curves and optimal response values of the neurons (circles) within the population are defined. Distribution of population activation (DPA) is used to determine the population activation, which is further processed by a DNF to produce the final output.

This value is called the optimal response value. The neurons participate in the population with tuning curves modulated by their level of activation - which is maximum around their optimal response values. The combined activation of a population of neurons provides the required features to the DFT system. The features are generated using the distribution of population activation.

The specific choice of the aforementioned two features is motivated by studies of the neural mechanisms behind intentional reaching movements [174–176]. These studies indicate that a reaching motion is decoded from neural populations of directionally tuned cells. Each ensemble of directionally tuned cells is tuned towards a preferred direction of movement. Each ensemble within the population contributes to the population by a vector directed towards the preferred direction of movement specific to ensemble of cells and is weighted by the cells' change in activity. The final sum of the population is called the *neural population vector* and points to a direction close to the observed direction of movement. The intensity of the neural population vector was also shown to be related to the speed or amplitude of the movement. The mirror neuron system suggests that the same mechanisms involved in action generation are the same as those in action perception, therefore it follows that features for action understanding should be mapped onto the direction and intensity of movement [13, 166]. The projected relative angle is a general representation of the direction of movement, while the body joint extension represents the calculation of the intensity of the movement.

### 3.3.4 Parameter choice for the DPA feature formulation

Tuning curves, centered around the optimal response value, can be modeled using different shapes. For example, they can be Gaussian tuning curves, cosine tuning curves and sigmoidal tuning curves [177]. The shapes and the parameters of each tuning curve is usually dependent on the specific neuron and stimulus. This can be further specified

given neuronal studies performed on lower primate species such as the rhesus macaque monkeys. However, the results can still be used for (human) biologically-motivated cognitive systems similar to ours [178]. We highlight the work performed by Perret et al. in [179] and the work of Newsome and Salzman in [178] that investigated the firing patterns in reaching motions, and which we base our work upon. We extracted their results and used the functions they proposed in designing our Gaussian functions that represent the tuning curves for motion sensitive neurons. The work of Newsome and Salzman focused on the direction discrimination in monkeys. They measured the visual response from the direction column in the middle temporal visual area (MT). We investigated their recorded data that presented the intensity of response given the direction of motion of the shown stimuli. After initially testing with cubic spline fitting, and parameter minimization using different family of curves, we settled on a representation using a Mexican hat function $\psi(x, \sigma, c)$, with width (standard deviation) of $\sigma$ offset $c$. The parameters of which were decided by solving an argument minimization problem (equation (3.13)) that minimized the Euclidean distance between the fitted spline $s(x)$ and the Mexican hat $\psi(x, \sigma, c)$ given in equation (3.14)

$$\underset{\sigma,c\in\mathbb{R}}{\operatorname{argmin}} \sum_{x\in[-180,180]} |s(x) - \psi(x, \sigma, c)| \tag{3.13}$$

$$\psi(x, \sigma, c) = \frac{2}{\sqrt{3\sigma}\pi^{\frac{1}{4}}} \left(1 - \frac{x^2}{\sigma^2}\right) \exp\left(\frac{-x^2}{2\sigma^2}\right) + c. \tag{3.14}$$

The work in Perrett et. al. also provides measured tuning curves and analysed them. We investigated results in their work in which they record neuronal responses to different head orientations and used their data in our modeling. The results showed that body parts are represented using view-centered descriptions. Furthermore, cells can be described as broadly, bimodally or narrowly tuned. We used the cell response information to model the tuning curves using a modified version of the fitting function (equation (3.15)) used in their original work. Perrett et al. argue for their choice of this equation stating that "it makes few assumptions about the nature of view tuning" [179]. Our modified version (equation (3.19)) guarantees symmetrical tuning curves and was used to solve the optimization problem in equation 3.16. Firstly however, the parameters $\beta_{1-5}$ that compose equation (3.15) have to be approximated given the extracted data $d(x)$ using the minimisation equation (3.16). Therefore modifying equation (3.15) to fulfil the condition $R(x) = R(-x)$ we get equation (3.19) in the following steps:

$$R(\theta) = \beta_1 + \beta_2 \cos(\theta) + \beta_3 \sin(\theta) + \beta_4 \cos(2\theta) + \beta_5 \sin(2\theta) \tag{3.15}$$

$$\underset{\beta_{1-5}\in\mathbb{R}}{\operatorname{argmin}} \sum_{x\in[-180,180]} |d(x) - R(x, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)| \tag{3.16}$$

$$\begin{aligned} &\beta_1 + \beta_2 \cos(\theta) + \beta_3 \sin(\theta) + \beta_4 \cos(2\theta) + \beta_5 \sin(2\theta)... \\ &= \beta_1 + \beta_2 \cos(-\theta) + \beta_3 \sin(-\theta) + \beta_4 \cos(-2\theta) + \beta_5 \sin(-2\theta) \\ &= \beta_1 + \beta_2 \cos(\theta) - \beta_3 \sin(\theta) + \beta_4 \cos(2\theta) - \beta_5 \sin(2\theta) \end{aligned} \tag{3.17}$$

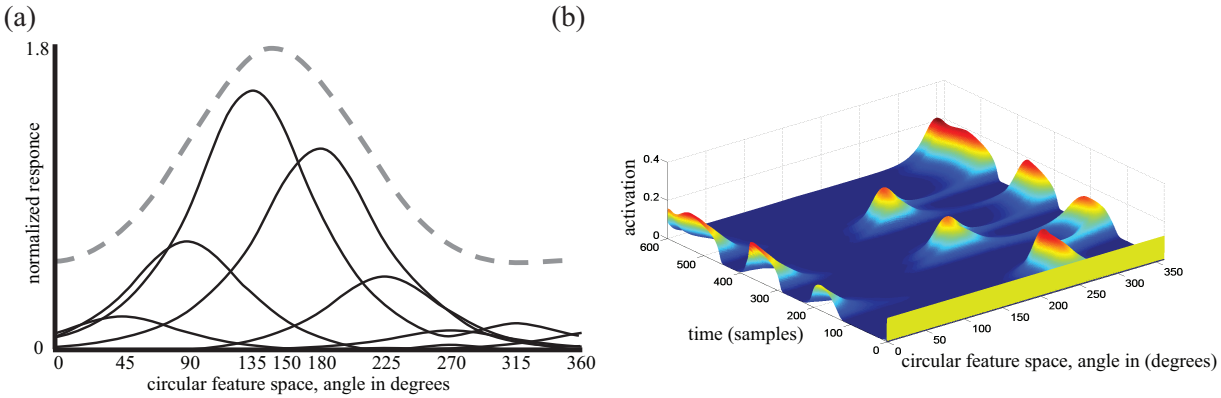


**Figure 3.7:** (a) The DPA response for a specific time step given an observed *projected relative angle* of 150 degrees. (b) The 2D memory trace of the *projected relative angle* between pelvis and right foot in the $x-y$-plane for forward walk action.

$$\beta_3 \sin(\theta) + \beta_5 \sin(2\theta) = -\beta_3 \sin(\theta) - \beta_5 \sin(2\theta) \tag{3.18}$$

$$R_s(\theta) = \beta_{s1} + \beta_{s2} \cos(\theta) + \beta_{s3} \cos(2\theta) \tag{3.19}$$

As previously discussed, a neuron contributes to the population at a specific time to a stimulus with its tuning curve that is centered around an optimal response value [154, 180]. For our cyclic features of orientation, we chose 8 equidistant neurons representing the feature space. Specifically, the optimal response of neuron is $n_i = f_i, i = 1, 2, ..., 8$ where $f = \{0\,^\circ, 45\,^\circ, 90\,^\circ, 135\,^\circ, 180\,^\circ, 225\,^\circ, 270\,^\circ, 315\,^\circ\}$. The cyclic features' shape (tuning curves) are modeled after the viewer-centered narrow tuned cell response [179]. For the linear feature space of distance, we used 6 neurons. The optimal response of each of the neurons was equidistant covering the complete feature space $0 - 100\%$. The tuning curve of each of the neurons was modeled using a Gaussian function with a wide standard deviation. The Gaussian functions were adjusted using the standard deviation to resemble the results of the fitted tuning curves discussed in the Appendix B, and were finally used as they are the common standards in the DNF framework [153]. The transition from discrete neurons to continuous feature space can be described by the DPA and is used as an input in our work to our DFT architecture. An example is shown in Fig. 3.7. A stimulus of arm configuration where the projected relative angle was $150\,^\circ$ was presented. The dashed grey line in Fig. 3.7 (a) shows the response of the population, while the individual black lines show the individual responses of the individual neurons in the population. While Fig. 3.7 (a) shows the response for a specific time step, Fig. 3.7 (b) shows the evolution over time in a neural field. It was observed in our results that the output of the DPA usually gives broadly tuned responses to different stimuli. Therefore, we used a DFT block that adjusts the input using a Neural field with a tighter kernel with a small time constant that does not cause delay in information propagation. This DFT block represents the interaction between the neurons within the population.

### 3.3.5 Summary

We have presented our biologically motivated model for motion perception that serves as a pre-processing block for the TARS. The CARS, on the other hand, takes the end effector's/pelvis' direction and speed as an input.

Our choice of features used to encode the 2D traces, shown in Fig. 3.7(b), was motivated by neuronal optimal response studies [168]. These studies showed that the orientation and distance travelled of observed objects (in our case hand and ankle joints) are encoded neuronally for the purpose of motion perception [181]. Optical flow, which also encodes a vector of direction and distance of moving interest points, has been shown to be significant of biological movement perception. This is also compliant to what is believed to encode motor commands (preferred population vector for a movement direction), enforcing the notion that the same code that encodes action generation is used also for action recognition [176]. These 2D traces are either saved as long-term memories, or provided online for internal comparison with saved memories. The saved long-term memories (preshapes) represent experiences of observing a specific action class [168]. The comparisons are performed in the TARS, however, as the number of actions can be large (the number of memories loaded at one time for comparison can be computationally expensive), we provide the CARS which we discuss in detail in the next section.

## 3.4 The contextual action recognition system

The search space to apply a meaning to an arm extension is rather large. It could be reaching to grasp an object, it could be to press a button, it could be to throw a punch. In order to restrain the search space, and to obtain the context of the movement, we propose a contextual system that aids in action understanding. Our hypothesis in this section is that an intelligent system can extract context from goal-directed motion performed by a human actor by observing the relationship between end-effector (hand) movement and the objects in the near vicinity and their action potentials. In this subsection we propose an attention-shift model and explain how it was implemented using DNFs.

### 3.4.1 Motivation and overview

Eye movement has been shown to react to goal-directed movements. Moreover, the relationship between the eye gaze of an observer and the hand of an actor is predictive [182]. Specifically, in CARS we model the attention shift by the (robotic) observer eyes, from the hands/hip of the actor to the object towards which the movement is directed. The CARS has additional significance since the robotic observer has no option to sense gaze shifts without expensive, invasive gaze detection sensors. This CARS is composed of several contextual action recognition modules (CARMs). Several CARMs are used as one CARM is needed for every item of interest (e.g. end-effector) we might want to track. Following from the work in [182], and as the gaze of the observing agent follows the actors end-effector, the chosen feature for the CARM is the optical flow information of actor's end-effector. Optical flow here specifically refers to the direction of motion tracking information. The

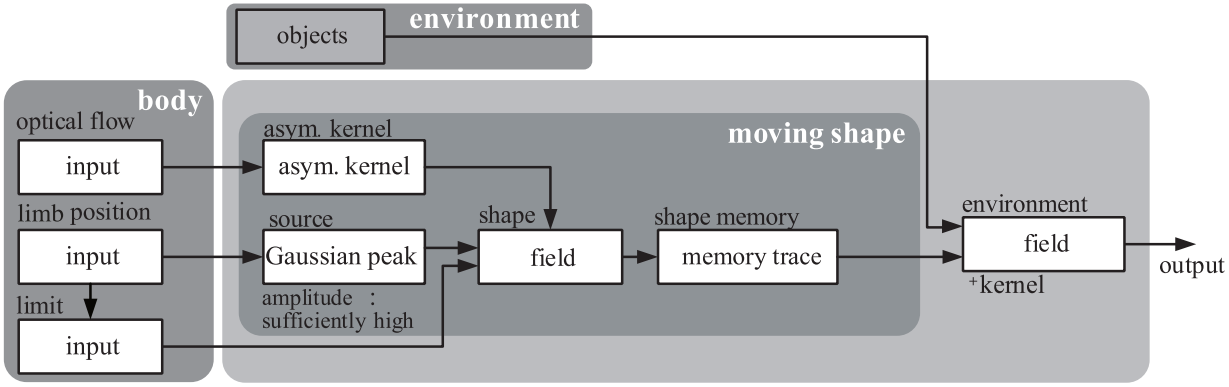


**Figure 3.8:** Architecture of the contextual action recognition module

optical flow information consisting of the actor's end effector (and hip) direction and speed [183] is used as an input for the CARM.

This information is fed to a moving shape (peak), shown in Fig. 3.8, which in turn feeds into a neural field that represents the environment that the actor is performing his actions in. This moving shape is initially located at the end effector's starting position and has a specific limit (set by the limit input block) that it is allowed to travel to before it fades away. The environment neural field is preshaped by the locations (given from the objects block) of manipulable objects in the environment, and is activated if the peaks *shot* from the actors hand (position given by the position input block using the direction/speed of the hand calculated using the optical flow input block) hits constantly a preshaped location. The peaks that are shot, are calculated in the shape neural field block, the speed of which is controlled by manipulating the parameters in the asymmetric kernel block. The term *preshaped* here refers to the fact that the provided activation of the objects is not sufficient to drive the neural field into activation, the field is then said to be subactivated at those locations or preshaped. In this sense, the environment block does not directly encode the environment per-se but the interaction between actor and environment. In the next two paragraphs we explain the different objects that preshape the environment field and the function of the moving shape field.

### 3.4.2 Physical and virtual objects

The object information that is fed into the environment field can encode physical objects that preshape the field at the same $x, y$ location they are observed. The same ideas are extended for locomotion actions (e.g., walking, turning, stepping left, stepping right, etc.) and free-space motions that are not necessarily goal-directed (e.g. waving, dancing, etc.). Virtual objects are imagined around the actor, and motion directed by the feet or the hands towards those virtual objects would read out their virtual affordances to give a hint of what the possible action is. Waving for example is an action in which an arm is extended towards the top and to the right of the head, followed by an arm extension to the left of that initial position and then to the right again and so on. The initial top right position can be thought as a virtual object that has the affordance of waveability. Just like a virtual

object towards the front of a person's centre can encode the ability of hand-shakability. We extend this to locomotion actions in which we can understand the direction of ankle movements towards a virtual object to the front of the feet can mean stepping forward and so on. While the use of virtual object is a simplification of how locomotion and free-space movement could be understood, it allows these two classes of movement to be assigned virtual affordances and be integrated into the overall architecture.

### 3.4.3 The shape field

Central to the moving shape module is the shape field. The output of the shape field is forwarded to a memory field that generates the path from the moving peak, and using this memorized trajectory an object is activated. The shape field has two inputs: a 2D Gaussian peak which is called the source input and a limit input. The source input is controlled by the position of a joint (e.g. wrist) $\mathbf{p}(t)$- relative to the shape field dimensions (that is, egocentric coordinates are respected here too) and is always kept at an amplitude sufficient to cause a permanent activation in the shape field. The source field is centered at the actor's pelvis. The source however could be fixed to the wrist or pelvis depending on the action type that is observed. We consider the left and right wrist joints for manipulation movements (alongside the information of the physical objects) and pelvis for locomotion movements (alongside their respective virtual objects). The combination of the source and the asymmetric kernel define the movement and activation within the shape field. Specifically, as the Gaussian peak position is defined by the joint position and not influenced by the shape field, an activation peak separates from the source (position of the joint) in the direction of the optical flow (joint movement) until it vanishes. The optical flow is calculated as follows

$$o(\mathbf{p}(t)) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot (\mathbf{p}(t) - \mathbf{p}(t-1)). \tag{3.20}$$

Within this work, a 2D Gaussian function $g(x, y, \mu_x(t), \mu_y(t))$ with maximum amplitude at the current *position* input $\mathbf{p}_{pos}(t)$ and a concentrated Gaussian is used (3.21). Accordingly, the expected value $\mu$ equals the *position* input $\mathbf{p}(t)$. Depending on the resting level of the *moving shape* field, the Gaussian has to be shifted by $c$ in order to prevent activation within the field:

$$\begin{aligned} &g(x, y, \mu_x(t), \mu_y(t)) = \\ &A \cdot \exp\left(-\left(\frac{(x-\mu_x(t))^2}{2\sigma_x^2} + \frac{(y-\mu_y(t))^2}{2\sigma_y^2}\right)\right) + c. \end{aligned} \tag{3.21}$$

The calculation of the asymmetric interaction kernel $w_{asym}(x, y, \mathbf{o})$ is presented in (3.22).

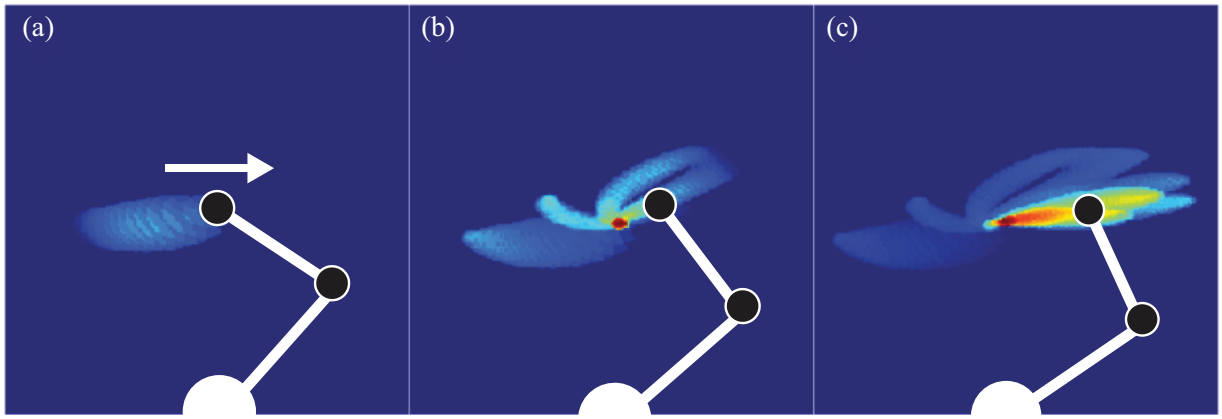


**Figure 3.9:** Example of a *moving shape* output. This figure shows three snapshots (order from left to right) of the output produced by a *moving shape* module in the case of non-zero optical flow.

The basis shape is defined by a 2D Gaussian as described in (3.21) but without shift:

$$\begin{aligned} w_{asym}(x, y, \mathbf{o}) =& g(x, y, \mu_x(t), \mu_y(t)) \\ &+ o_x(t)\frac{\partial g(x, y, \mu_x(t), \mu_y(t))}{\partial x} \\ &+ o_y(t)\frac{\partial g(x, y, \mu_x(t), \mu_y(t))}{\partial y}. \end{aligned} \tag{3.22}$$

The limit input is introduced to add control over the vanishing time and the distance that the separated activation peaks travel. This input preshapes the shape field and restricts the traveling of the peaks to certain areas given information passed from the current position (white arrow in Fig. 3.8).

A moving shape is shown in Fig. 3.9. This figure illustrates an arm moving towards the left. What this would translate to within the moving shape module are the waves seen in the figure. A moving peak centered at the wrist position would propagate given the information of the optical flow. Accumulating waves would build up activation while noise generated from the movement would die out as shown in Fig. 3.9. To summarize, the inputs to the moving shape module are the *optical flow* input, the *position* input and the *limit* input. The output of the moving shape module is the memory trace activation in the *shape memory* field. The moving shape module contains two fields. The first field is the *shape field* that takes an algorithmic inputs of the asymmetric kernel, and gaussian peak calculation. The second field is the *shape memory* memory trace that accumulates the output of the *shape field*. Both fields are defined of the metric space field spanning the immediate environment in meters.

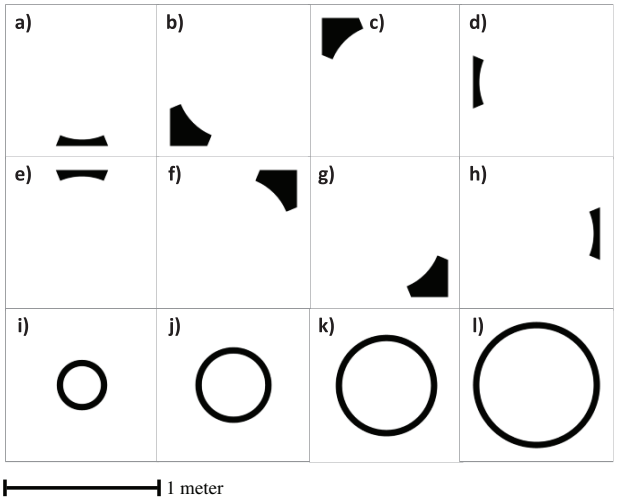


**Figure 3.10:** Virtual objects for direction and movement detection (a)-(h) show the eight directions, each covering $45^\circ$ (i)-(l) show different sized circles for movement detection. Each square has a length of 1 meter and is enough to cover the workspace of an arm.

### 3.4.4 The environment field

Finally, the environment field is a decision field that performs a selection given the (virtual/physical) objects that preshape it and the moving shape module's output that also provides a preshaping input. The field is defined over the feature space representing the environment (in meters). The output is the location of the objects that the observed agent is predicted to manipulate. It is important to note that physical objects in our implementation encode both furniture and manipulable objects. Virtual objects encode positions around the body used for both direction and magnitude (the intensity of the motion) detection. The assumed virtual objects surrounding the body are shown in Fig. 3.10.

A stable peak in the environment field is an indication of which object the actor is intending to interact with and where this interaction is being (will be) performed. For the virtual objects, it gives an indication of what kind of locomotion movement is being performed and intensity/direction of the movement.

### 3.4.5 An example of a CARM

An example of the processes within a CARM is shown in Fig. 3.11. The bottom layer, Fig. 3.11(a) represents the body input in which the gray arrow represents the instantaneous value of the optical flow at the wrist joint. There exists three objects in the environment illustrated using gray circles. A source peak is built over the wrist and peaks are shot from that source given the optical flow information of the wrist joint as shown in Fig. 3.11(b). The activation of the waves are maintained in the memory field illustrated in Fig. 3.11(c). The activation serves to add activation to the environment field on one of the preshaped locations that represents the underlaying objects, as illustrated in Fig. 3.11(d). The accumulated activation at one of the object locations allows for selection as shown in

Fig. 3.11(e) The movement is directed towards one of these objects.

The affordances of that specific object can be read out and preshape the TARS which in turn validates the type of affordance on a movement level. We discuss the modules that make up the TARS in the following section.

## 3.5 The trajectory recognition system

When an acting agent performs an action, his/her movement kinematics provide an abundance of information a human observer could use to recognize the action. In terms of movements, human action varies constantly. That is, for the same action, a person performs movements differently across multiple runs. The time it takes to complete the same action varies also from one trial to another and from one person to another, depending on the task and the kinematics of the actor. In this section, we provide a DNF model of motion trajectory comparison for the purpose of action recognition that acts independently of environmental information. These different blocks that compose the trajectory recognition module are visualized in Fig. 3.12. We explain how we achieve spatial and temporal invariance and provide insights on how the intrinsic properties of the DNF could be used to dynamically adapt the fitting between stored memories and the observed data and give it a *better chance* to get a positive fit. We also discuss our implementation for producing and processing these stored memories (templates). It is worth emphasizing that the TARS is composed of several trajectory action recognition modules (TARMs) specific for each action to be recognized.

In compliance with the template-matching model, biological systems depend on a stream of features (stimulus) produced by static views of the body to perceive and classify movement patterns [130]. These features can be thought as form cues of a specific body configuration, similar to the concept of snapshots presented in [125]. They are called snapshots of interest within our work. The existence of a specific sequence of snapshots encodes a specific action/movement. We refer to this sequence as sequence of interest. However for classification, we need a reference sequence of interest to be matched against. We rely on a set of stored memories (templates) for the observation of different actions as well as a comparison model. Templates are learned in our DNF model by applying an activation of motion features over time in a DNF that represents a template. The classification of actions here would be similar to other single layered exemplar-based sequential approaches that depend on a sequence of feature vectors to perform the classification [147]. We discuss template generation in section 3.5.1. This template has to be adaptive to account for the challenges of AU, for that we present our dynamic template solution in section 3.5.3. From the previous overview, the TARM can be composed into an input side and a preshape side, and they are compared against each other with a *comparison* block, this is discussed in section 3.5.2.

Due to the challenges of AU discussed, the differential speed between the input and the template should be controlled for purposes of correct recognition, and this is done by a *controller block* which is discussed in detail in section 3.5.4. Specifically the *controller block* controls the speed (and the time intervals) at which the traveling wave propagates through the preshape field, as the stimulus is fed online as the movement is observed. In

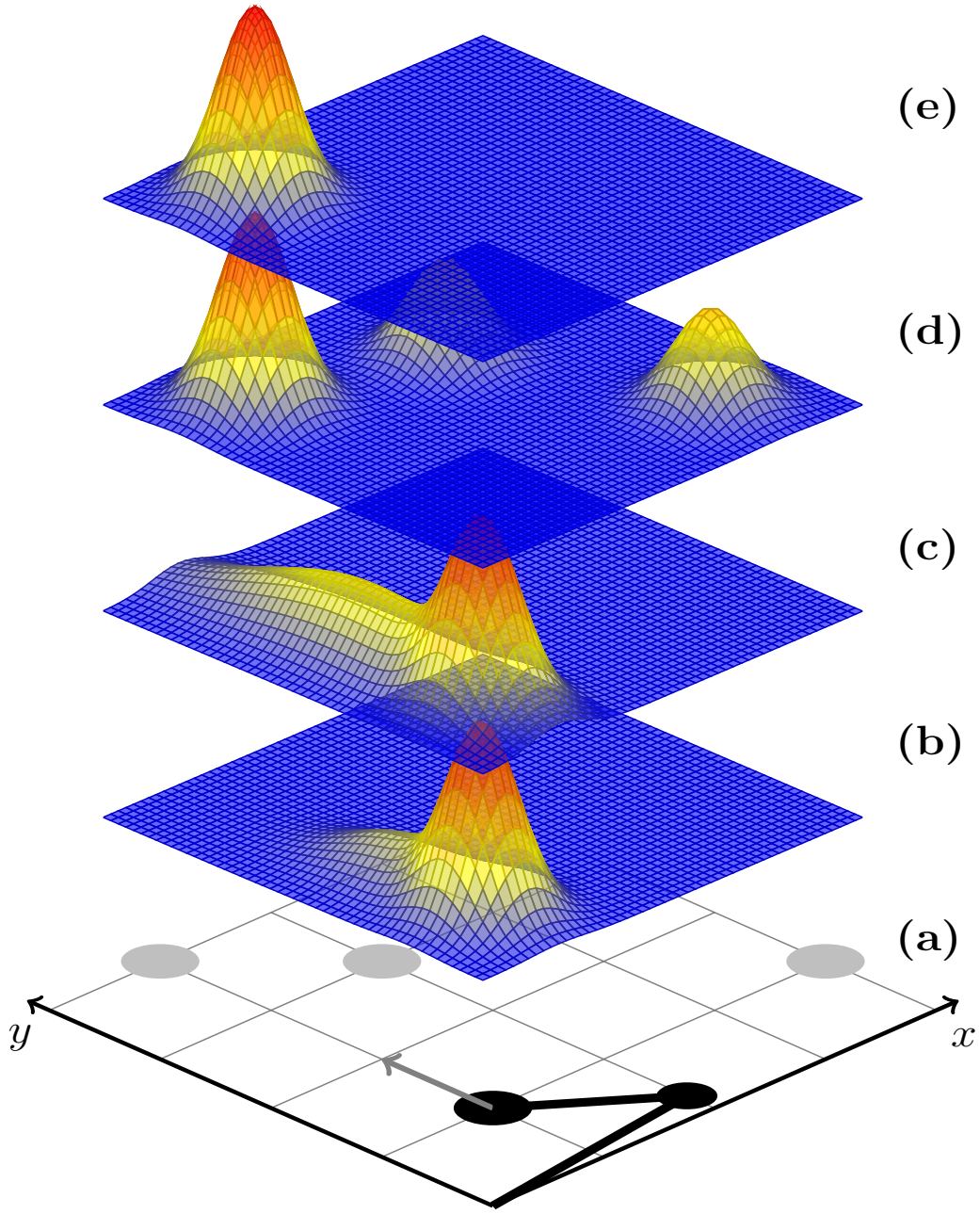


**Figure 3.11:** Example of the operation of a CARM (a) The kinematics of an actor is observed as he tries to reach towards an object and away from obstacles (b) Gaussian peaks are shot from the source that is centered around the right wrist. (c) The activation from the shot Gaussian peaks are maintained in a memory field. (d) The accumulated activation allows for the selection of a peak out of many possible preshaping peaks. The preshapes represents objects existing in the environment. (e) A decision is made in the final selection layer. This decision represents the object that the observer predicts the actors's movements are directed towards.

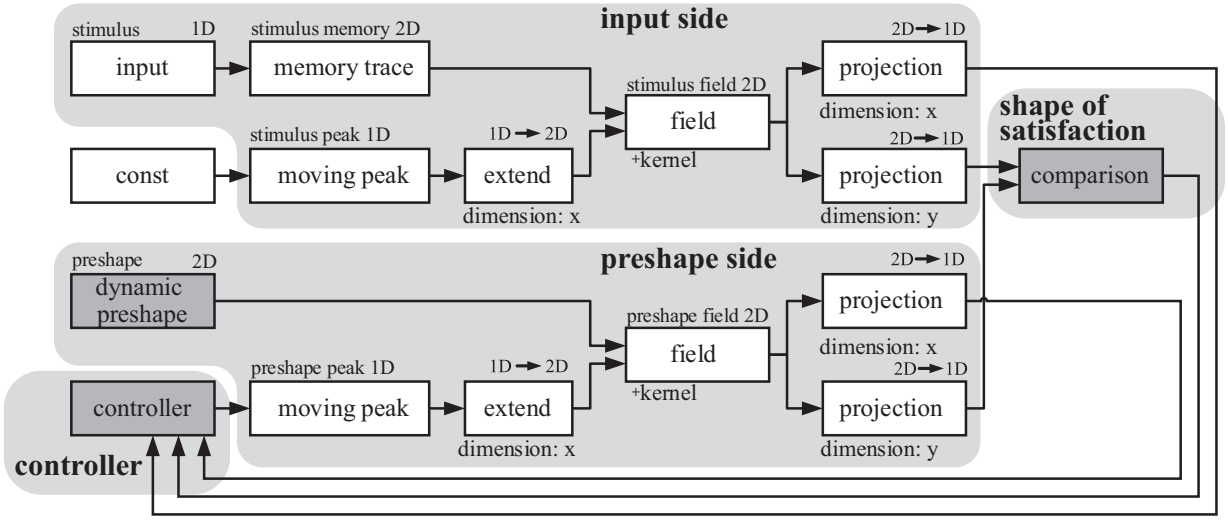


**Figure 3.12:** Overview of the trajectory action recognition module. Dark grey blocks represent blocks that are explained in detail in separate sections. Feedback signals are given to the controller from projection fields and the comparison field.

the following, we discuss the template generation process which is illustrated in figure 3.13.

### 3.5.1 Template generation

The core mechanism here is the accumulation of a memory trace from multiple samples. The samples, which are represented in feature format, are accumulated within a field with a memory trace. The features, as discussed in section 3.3, encode ego-centric distances and angles between the pose of the head or hip (reference) and the wrists and ankles (end effector) in the sagittal, cornal and transverse planes [163, 167, 168, 184]. The choice of wrists and ankles are because they indeed move the most [185]. The observed agent is projected onto the body frame of the observer such as to achieve view (spatial) invariance and model the internal simulation behind action recognition [165]. The DPA model discussed in section 3.2.2, was used to model a set of angle and length sensitive neurons at discrete values similar to what is observed in the neural system of the human [175, 183, 186]. At any given time, the produced features (stimulus) are called a snapshot of interest. The activation of these angle/length sensitive neuronal populations over time activate a DNF either for learning a preshape (template) or to be directly fed as an input for the comparison. The process of generating the trajectory templates is shown in Figure 3.13.

Templates were generated by a mean-like approach within a DNF given several examples from a class of actions. The template generation process illustrated in Fig. 3.13 is modeled such that a single observation (in stimulus trajectory form) is appended to the already accumulated motion observations. Our motivation stems from the intuition that an action is observed completely and continuously and is added to overall past experiences dynamically. From multiple examples of an observed action recorded in our dataset, we pick one random sample, and present it in stimulus form. This is done in the *select sample* block. The length (time) of the sample is normalized, in the *preprocessing block* to a length that was pre-calculated. This pre-calculated length represents the average length of this

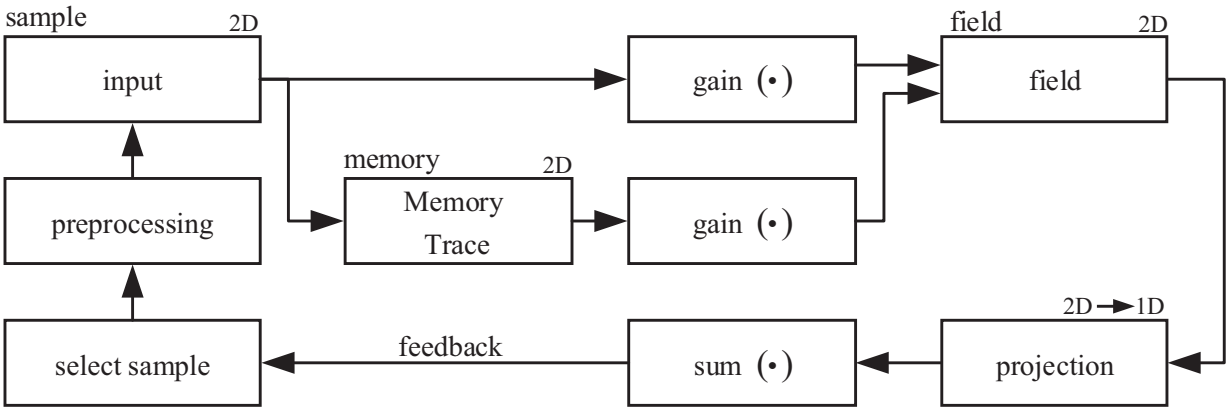


**Figure 3.13:** System architecture to generate trajectory-based templates.

specific action class. This input is then fed into two pathways that again merge into a DNF. The upper pathway multiplies the sample with a gain, while the lower pathway accumulates the observations within a memory and multiplies the output with a gain afterwards. These gains are essential to the learning process. They define how the learned information is changed and when to select a new sample to learn from. The two pathways are merged into a DNF that is projected against time and finally summed up across its activation. A feedback law is then defined such that this activation summation is compared against a threshold value that determines the transition to learn a new example.

### 3.5.2 Comparison block

As the learned preshapes could be substantially shorter or longer in time compared to the observed motion, we propose to use moving peaks to solve the problem of time variability. A peak would propagate in the DNF of both the preshapes templates and the perceived action. The peak in the preshape would jump to *special* locations characterized by fast changes in the feature space. These jumps would be fast in nature. A jump would occur to the next location in the preshape field only if the same feature was observed in the input field that represents the perceived action. This check is performed in the comparison field as shown in Fig. 3.12. As the wave in the preshape field propagates more and more towards the end, the more we are sure that the preshape correctly represents the action we think it is.

This *jump* that occurs from one *snapshot of interest* to the other is determined by allowing the wave to propagate forward at a high speed and detect areas of interest within the preshape. These areas of interest are either zero crossing areas or extrema/saddle points. The snapshots of interest are calculated online by merging both a Gaussian wave that is extended in time and the original preshape in a neural field called the zero crossing field. The online calculation is of vital importance as the area of interest should be allowed to be shifted and adapted during comparison to allow for the best fit between observed and saved values in the features. The two inputs activate the neural field on intersection within the field, this activation is designed to occur around zero crossing points. The projection of this zero crossing field against feature value gives the times at which the

sample has zero crossing points. This can be further expanded against time and fed into as an input alongside the original preshape in a field that presents the sequence of interest. The saddle-extrema points can be calculated similar to the zero crossing points, but after an initial derivation of the preshape has been done. The derivation is done offline. The jump between snapshots of interest could be understood as interstimulus intervals (ISIs) which represent blank intervals between point light display frames which in turn allow low-level influences on human motion perception [185].

Comparison between the snapshots in the preshape and the continuously evolving stimulus input occurs in the comparison block as shown in Fig. 3.12. The comparison block discussed in section 3.2.4 is utilized here. The results of the comparison (match/no match or continuous comparison) are used as feedback signals to the controller block.

### 3.5.3 Dynamic templates

The core mechanism here is dynamically changing the values of the different available parameters (e.g. resting level of the preshape field or the value of the short range excitation of the interaction kernel) within the *preshape field* given the success of the comparison within the TARM. The basic motivation behind the set of tools used in *dynamic templates* approach is twofold. Firstly it is considered a way to to allow for a faster successful comparison result. Secondly, it is a way to allow the generalization of templates.

As the confidence of observing a specific action increases, the more the dynamic preshape is allowed to influence the action recognition process such as to compensate the spatial variation between the preshape and the stimulus. The portion of the preshape that had not been compared against yet, is made to fit the previously observed motion. The compensation is calculated given the past information of the perceived motion. It also allows for the imperfections observed when learning a preshape template and allows some spatial variation between stimulus and preshapes. It aids towards the generalization of the templates. While false positives might be a hindrance due to the use of dynamic templates the use of CARS would limit the number of loaded preshapes such as this drawback is mitigated. This drawback is further shown in the results section.

The dynamic preshape solution we propose is divided into two steps. The changing preshape step aims at manipulating parameters within the preshape generation method. Such changes could be limiting the samples used or manipulating the field to exhibit behavior other than producing a mean-like stimulus trajectory. The changing preshape step alters the shape of the preshape completely and dynamically.

The second adapting preshape step does not change the preshape. It adapts the current preshape given the information seen so far from the stimulus by either shifting it in feature space or influencing its shape slightly. The shape is changed by performing the convolution normally done within the DNF using an adapted 2D Gaussian kernel. The width of the 2D Gaussian kernel is changed depending on the confidence value of the overall trajectory comparison module. This dynamic adaption of the preshape gives a better chance for the fit to occur as we are more confident of our action classification.

### 3.5.4 Controller block

The controller block shown in Fig. 3.12 takes three inputs. These inputs are the temporal positions of the moving waves within the stimulus field and the preshape field as well as the results of the comparison block. The output of this block controls the velocity of the moving preshape wave. This controller block is purely an algorithmic implementation, and is not implemented using neural fields. Furthermore, we assume for the purposes of this control block that the length of the input stimulus, and therefore the temporal position of the stimulus within the currently observed action, is not known. This is a logical assumption since we do not know when the actor will end his action nor at which stage he is currently in. We do however assume that we know the length of the preshape and the position of the traveling wave within the preshape. This is again a logical assumption as we have these preshapes stored as memories within our action understanding system.

The controller, which provides a stop and go signal for the wave, takes a logic input from the comparison module. The controller, which is implemented as an if/else statement, stops the traveling wave (on a snapshot of interest) or allows it to propagate forward with a velocity that is at least as fast as the stimulus' velocity towards the next snapshot of interest. The overall result of the TARM comparison here can be presented as percentage of the current position of the wave within the preshape to the overall length of the preshape. We define this value as the *confidence* value within this document, which serves as an indication of the correct matching preshape. Within our implementation we have also experimented with a continuous controller that slows down or speeds up the traveling wave given the differences between input and preshape. However, we opted for the stop and go controller in our final implementation as the comparison module follows the work on *Condition of Satisfaction*, which provides a logical value of match or not match.

### 3.5.5 An example of a TARM

The TARM takes two trajectories, one from the input side and one from the preshape side (or from an internally generated movement). In order to resolve the temporal variation that could occur, a moving peak travels through both input field (stimulus) and preshape fields. While the speed of the moving peak is continuous in the input side, it is discrete in nature in the preshape side, and is determined by way of a controller. An initial preshape slice is shown in Fig. 3.14(a), while the initial input from the input side is shown in Fig. 3.14(b). The controller checks if the current time slice of the preshape field matches to what is observed in the input side in a comparison field that performs a RoS calculation as discussed earlier. The comparison is shown in Fig. 3.14(c) and indicates a correct comparison. This would allow the preshape moving peak to propagate forward at a fast speed to the next time slice of interest, shown in Fig. 3.14(d). The preshape slice is *stuck* at that position, until the stimulus from the input side catches up and a correct comparison occurs. However, in this example the input illustrated in Fig. 3.14(e) never allows for a correct comparison, as shown in Fig. 3.14(f), therefore the peak is stuck at its previous position. This indicates that this specific TARM is not consistent with the observed motion. When another TARM is successful at understanding the movement, the TARS is reset and the moving peaks start from the initial position.

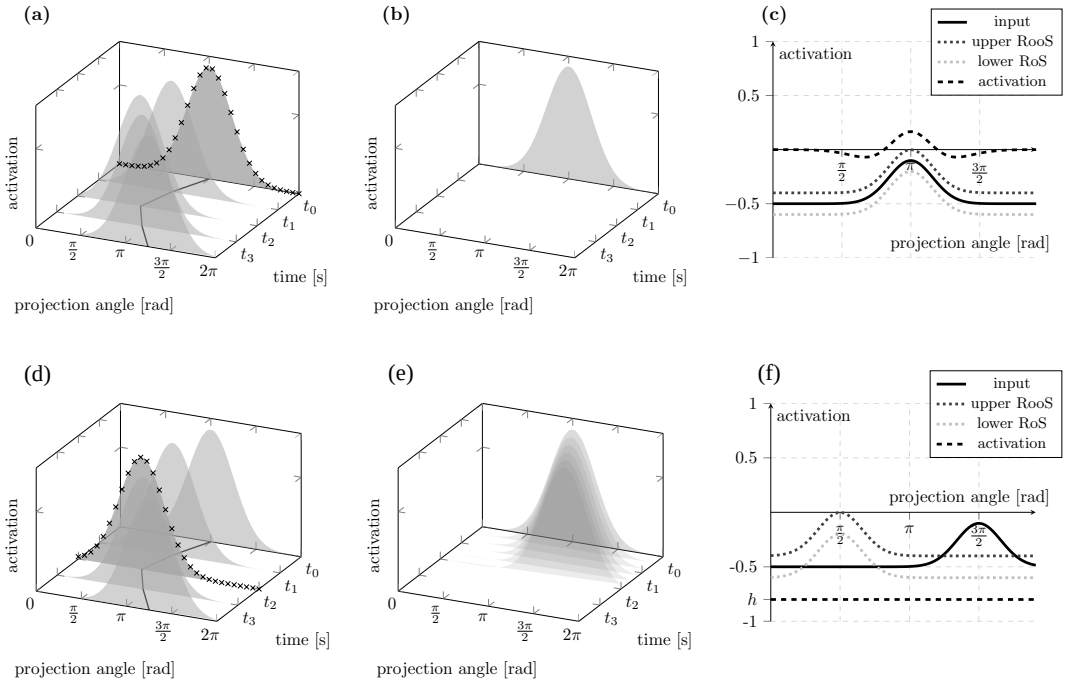


**Figure 3.14:** Example of the operation of TARM (a) An initial preshape slice of the internally simulated neural trajectory (highlighted by the black crosses). (b) An initial input time slice. (c) A RoS comparison between the input and the preshape which shows a correct comparison. (d) A preshape slice of the internally simulated neural trajectory (highlighted by the black crosses) at a position of interest, at a later time step. (e) An input time slice at a later time step. (f) A RoS comparison between the input and the preshape which shows a false comparison at the later time step, leading to the preshape peak to be stuck at that location.

It is important to note that the moving peak in the preshape side *jumps* from one *position of interest* to the next position based on the characteristics of the preshape trajectory. Specifically, it jumps to a point where fast changes in the feature space is observed. Finally, a dynamic preshape allows for local and global shifting in the preshape given correct comparison. In other words, as more comparisons are correct between the input side and the preshape side, the more the dynamic preshape adjusts the original preshape to allow for a better fit.

## 3.6 Affordance logic and connectivity fields

The observing agent identifies and predicts the acting agent's action through understanding its dynamic interaction with the (real or virtual) objects in the agent's immediate environment. The CARS shifts the attention of the observer from the end effector towards real or virtual objects whose affordances can be read for further processing. These affordances constrain the set of all possible actions to a limited subset. This subset is used to

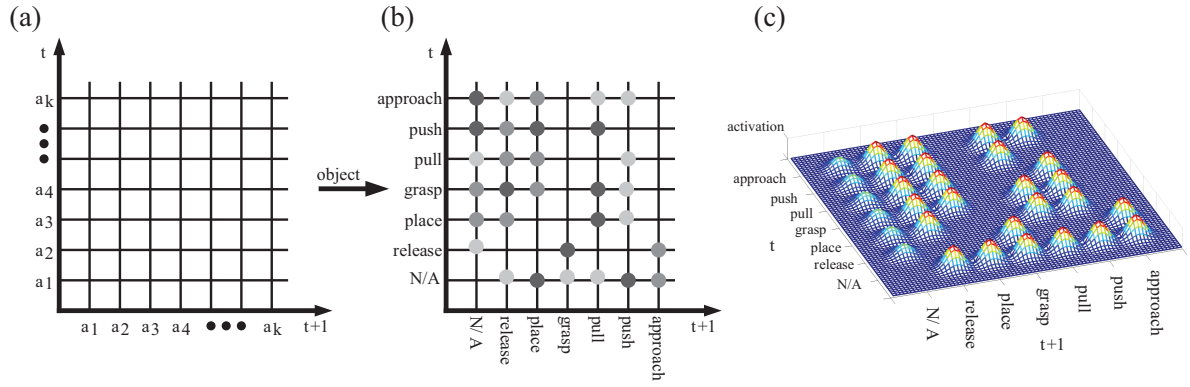


**Figure 3.15:** Connectivity field. a) General structure of the connectivity field for $k$ actions $a$. b) Connectivity field for objects. Gray points represent connections between the current action at $t$ and possible future actions at $t + 1$. Different shades have been chosen in order to represent that different connection strengths are possible.

bias the TARS through a choice of a limited number of preshapes and dismissing the rest. We introduce in the following the concept of *connectivity fields* which aids in achieving the previous ideas.

The connectivity field is a lookup-table-like DNF that encodes future possible object affordances given the object's current affordance state. It houses both ideas of sequential and nested affordances [187]. Each object is represented using its own connectivity fields, which is a 2D DNF with a 2D feature space. The first dimension encodes the current action states of the object and the second dimension contains the action states available in the next time step. As an example, if a glass is being grasped now, it can be released, placed ... etc. as shown in Fig. 3.15. A general structure of connectivity fields is shown in Fig. 3.15 (a) for a connectivity field of $k$ action possibilities $a_{1-k}$ an object might have. A populated connectivity field is shown in Fig. 3.15 (b), in which connections were learned in a 2D memory field. The different shades of peaks in Fig. 3.15 (b) refers to the fact that there exists different probabilities of action transitions encoded in the strength of the connection. Figure 3.15 (c) shows a learned connectivity field.

Within our implementation, we did not integrate abilities of object recognition nor affordance attribution or learning. Object recognition within DNFs has been discussed in [188]. We assumed knowledge of positions, labels and affordances of the objects in the environment to be known. Furthermore, the list of affordances were defined in a complimentary manner to fit the list of action primitives that were recorded in our dataset. This is in accordance with the notion that affordances provide action potentials and provide a logical link between action and environment. These affordances make up the connectivity field.

The connectivity field was realized using a memory trace that saves peaks of activation at connection points between previous and current action state. As the actions are discrete in nature, the input to the memory trace 2D field is an activation of action (neural) population whose tuning curves have zero overlap and have an optimal response value spread equidistantly over the feature space. The learning of how current and future affordances

are connected occurs as follows: when we observe action changes, both feature spaces activate at the locations of these discrete actions, activation at the intersection of both actions emerge within the connectivity matrix field and is finally this peak of activation is saved in the memory trace.

The output of the connectivity matrix can inhibit or excite the saved preshapes of the TARS. When an action is observed it influences the connectivity field. That is, an activation is spread horizontally at the location of that action. This activation is sufficient to activate preshaped peaks (learned in the previous step). These activations are read out by projecting the 2D field onto the next action state axis. These activations go on to excite preshapes in the TARS and the rest remain inhibited.

## 3.7 The action understanding task

For the human action understanding task, we had set up an apartment environment within our laboratory and invited ten participants to perform high-level scenarios as well as short precise movements we refer to as primitives. The goal of the primitives is to provide our system with learning examples of how simple movements were performed. The concatenation of several simple movement primitives (e.g. walk forward, turn, step forward, reach, grab, pull, ...etc) add up to a higher level intention. The primitive actions could be separated into two main categories: manipulation and locomotion actions. The locomotion actions that were recorded were: step (forward, left, right and back), walk (forward and backward), turn (right/left, 90/180 degrees), standing up and sitting down. The manipulation actions that were recorded were: approach (reaching action without a grasp, such as turn on the light switch), grasp (a reaching action with a grasp), push, pull, place, open and close door.

We designed the high-level scenarios that portray a specific intention such that a series of aforementioned primitives were used to execute them. The scenarios we set up were: *pick up the remote to watch TV*, *pick up a snack to eat*, *go to work*, *get up on a vacation day* and *tidy up*. The ten participants were instructed to perform the high-level scenarios as naturally as possible, and were not told to follow a specific order in their execution. We assume that recorded primitives would give us a wide range of movements and allow us to recognize them within the execution of a high-level scenario (intention). By performing the recording session of primitives first, we would prime the participants to using those specific primitives in the high-level scenarios, however this priming effect was not measured nor analyzed. It was observed however, that some participants employed creativity and added a lot of character into the high-level scenarios, as one of the instructions they were given was to *act* as if they were in their own apartment. As an example, some chose to do stretching movements in the get up scenario.

For the motion recording we used an Xsens MVN full-body inertial motion capture (MoCap) suit. The sensor fusion scheme of the Xsens MVN suit gives the kinematic information (position, velocity, acceleration, orientation, angular velocity and angular acceleration) of each body segment as an output [189]. We opted for a motion capture suit as extracting a skeleton of video frames is not the focus in our work. Furthermore, having MoCap data of movement allows us to model the observing robot anywhere within the

apartment environment without being restricted to a certain view point or having to deal with occlusion.

The *grab a snack* task will be evaluated to discuss the results of the CARS in section 3.8.1. Then, the *pick up remote* scenario will be used to give initial results for the integration of the TARS and CARS in section 3.8.3. In the following, we will discuss the CARS and TARS individually and then introduce how they can be integrated into an action understanding system.

## 3.8 Results

The previous section focused on presenting the individual modules of the overall architecture. Many TARMs could be recruited depending on the number of actions to be recognized and build the combination of which composes the TARS. Likewise, many CARMs could be used depending on the end effectors that are of interest and the combination composes the CARS. In the following we present our results of the dynamic systems, CARS and TARS, see section 3.8.1 and section 3.8.2. Additionally, we present initial results of the integrated system in section 3.8.3. The high-level scenario that we used to produce the results in the CARS section is the *Pick up a Snack* scenario. Finally, we evaluate the integrated system with the *Pick remote* scenario. Figure 3.16 shows the 2D reconstruction of our apartment environment. The *Pick up a Snack* scenario consists of getting up from the couch, walking towards the kitchenette, picking up the apple and walking back to the couch to sit there. The *Pick remote* consists of getting up from the couch, walking towards the TV table, picking up the remote and walking back to the couch to sit there and place the remote on the coffee table. The ground truth of the *Pick up a Snack* scenario is given in Table 3.1. The ground truth of the *Pick remote* scenario is given in Table 3.3. The complete architecture was built using MATLAB/Simulink environment using a modified version of the open source toolbox COSIVINA [190].

### 3.8.1 Contextual action recognition system

Three CARMs are running at all times. One for the right wrist, one for the left wrist and one for the pelvis (results for the pelvis are not shown). The virtual objects necessary to be loaded for the function of the pelvis CARM are only loaded when the optical flow information of the pelvis is above a certain magnitude (0.8-millimetres). This threshold was calculated with a decision tree classifier using the magnitude of the optical flow information of the pelvis as the distinctive feature. The moving shape field for the right wrist and the left wrist was shown earlier in this article in Fig. 3.9, in which a right wrist is simply moving right. The field would be preshaped with objects that would allow prediction of interaction. The results are tabulated in Table 3.2.

For our example, the environment houses both furniture items as well as objects. Contextual information of what object and at which location it was manipulated can be inferred using the CARS given the movement of both wrists. We show the results of the CARMs for the right/left wrist interacting with furniture in Fig. 3.17-left ordinate, and the right/left wrist interacting with objects is also shown in Fig. 3.17-right ordinate. The right wrist

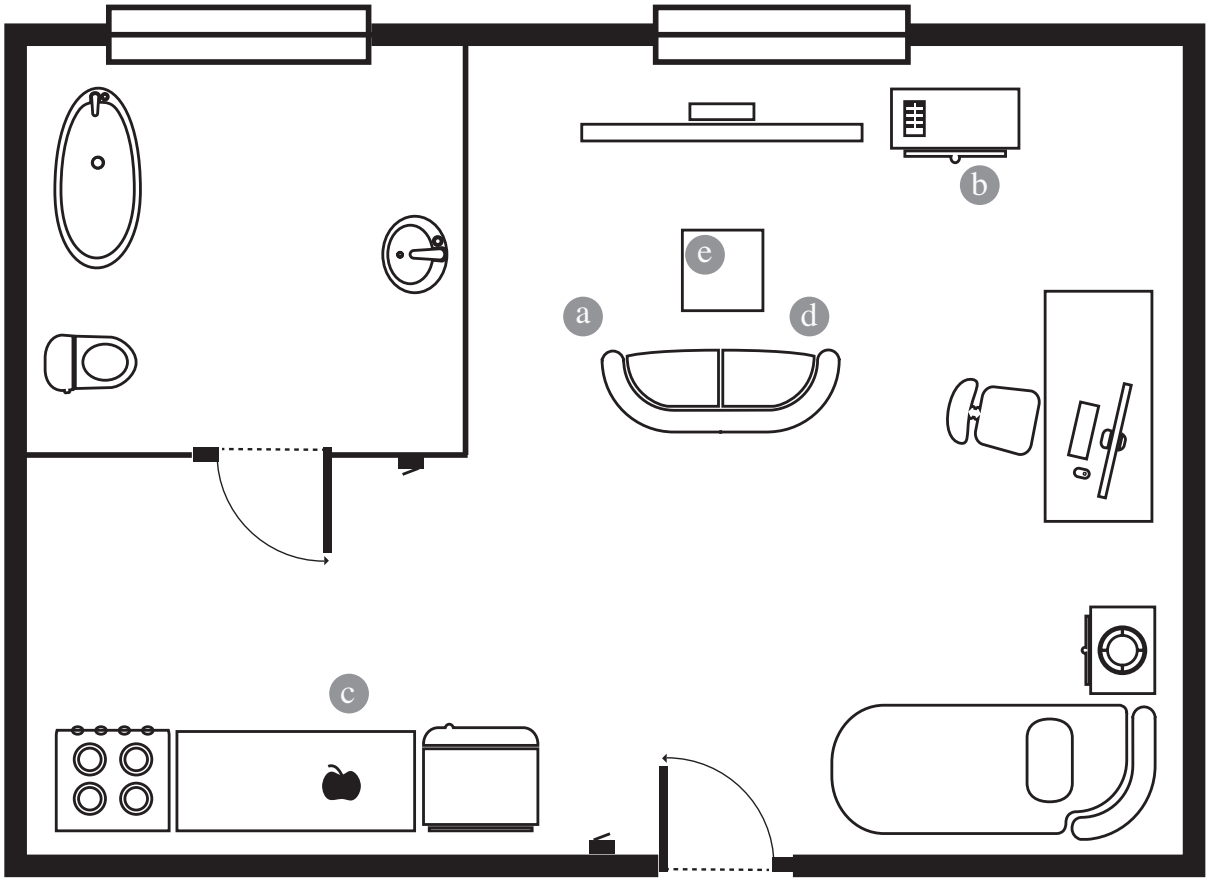


**Figure 3.16:** The apartment environment that was used to record the high-level scenarios. (a) The couch (start position). (b) TV table and remote positions. (c) Kitchenette and apple positions. (d) The couch (end position). (e) Coffee table.

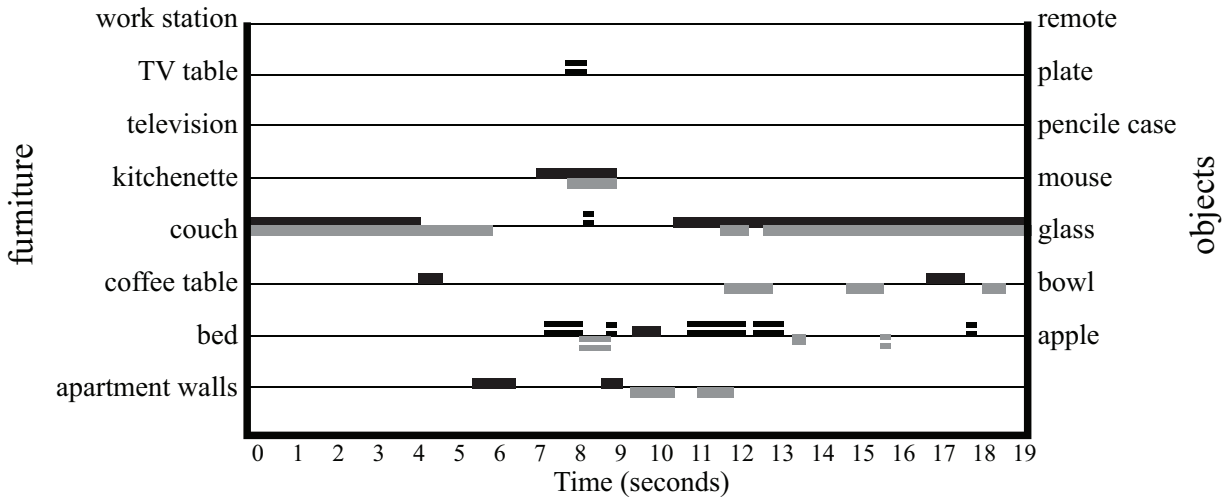


**Figure 3.17:** Interaction with the apartment furniture (listed on the left ordinate and read with the solid lines) and apartment objects (listed on the right ordinate and read with the horizontally dashed lines) for the right wrist (solid/dashed black lines) and left wrist (solid/dashed grey lines). The abscissa represents time in seconds. Detected interaction is illustrated by lines. An example of reading the figure would be: the right wrist was interacting with the glass (dashed black line) at the kitchenette (black solid line) around the 8.5 second mark.

contextual information can be read using the solid/dashed black lines while the grey lines are referring to the contextual information of the left wrist. Figure 3.17 shows an initial interaction with the couch (initial sitting position), then as the subjects stands up his/her movement is towards the coffee table and near the apartment walls later he/she interacts with the kitchenette and walks back towards the couch where he/she places the apple on the coffee table. In terms of objects, Fig. 3.17 shows longer activation with the apple, as the subject reaches, grabs and walks back to the couch with the apple. As can be seen in the results, the CARS only makes a selection of objects/furniture that are predicted to be manipulated, while suppressing the other objects/furniture. The CARS as expected gives contextual information of *what* and *where* interactions take place. The CARS also gives context of locomotion movements necessary to understand such motion.

### 3.8.2 Trajectory action recognition system

Multiple TARMs are running the whole time. One for each action and their respective features. They benefit from the output of the CARS computationally as only a subset of TARMs are excited at each time, the others are inhibited. In the following we show results for the TARS separately and explain why simple trajectory comparison does not aid in a dynamical action understanding architecture.

Figure 3.18 shows a comparison of a generated mean for a step forward action, for the feature of projection distance in the $x-z$ plane between right and left foot. Figure 3.18(a) shows the generated template while Fig. 3.18(b) shows the corresponding mathematical

**Table 3.1:** The pick a snack scenario: ground truth

| Start (seconds) | End (seconds) | Furniture | Object |
|---|---|---|---|
| 0 | 4.8 | couch | |
| 9 | 10.5 | kitchennete | apple |
| 16.5 | 19 | couch | |

**Table 3.2:** The pick a snack scenario: right hand results

| Start (seconds) | End (seconds) | Furniture | Object |
|---|---|---|---|
| 0 | 4 | couch | |
| 5.2 | 6.5 | apartment walls | |
| 7.2 | 9 | kitchenette | apple |
| 9 | 10 | bed | |
| 11 | 13 | couch | apple |
| 10.2 | 19 | couch | |

mean template. Figure 3.18(c) illustrates the difference of (a) and (b). Thereby, dark red represents the maximum value whereas dark blue represents the minimum. This comparison shows our approach is comparable to the mathematical formulation of a mean template.

This mean template can be adapted dynamically given the results of recognition confidence. The way that the template is adapted within DNF is shown in Fig. 3.19. Finally, in Fig. 3.20 we show the results of comparing the step forward action against all other action primitives. Only the step variants reach 100% finally, a fault that can be resolved if the CARS was also connected to suppress templates that represent large movements in the forward direction. However, the confidence level reaches a high level of confidence late, and recognition could be confused earlier across many actions. As these results are obtained by employing TARS alone, the CARS provide means to eliminate a large portion of these actions and allow for a better comparison as will be discussed in the next section that presents the results of the integrated system.

### 3.8.3 Integration of context and trajectory recognition

In the following section, we show our initial results of the CARS and TARS for the *pick the remote* scenario. Within this example, the participant stands up from the couch, takes a few steps forward towards the television table, picks up the remote, sits back down on the couch and places the remote on the coffee table in front of him. The "pick the remote" scenario's ground truth is given in Table 3.3. Figure 3.21(a) shows the results of the CARS and the objects the observer predicts given the participants right wrist movements. The affordances of the objects that are predicted in the CARS step runs several TARMs at

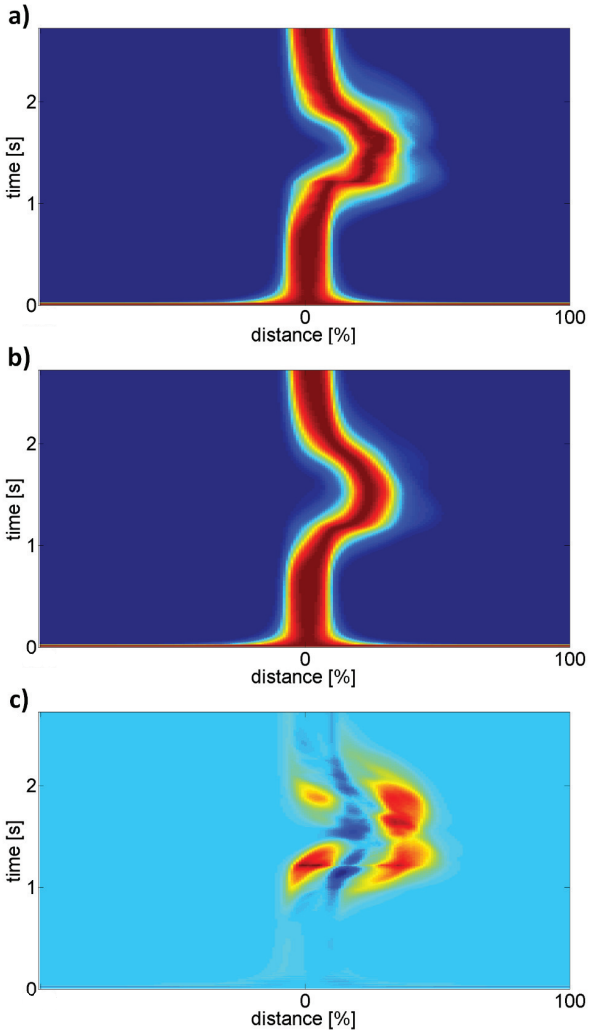


**Figure 3.18:** Comparison of a generated mean template with corresponding mathematically calculated equivalent. the chosen example is: *STEP_FORWARD*, projection distance $xz$ between right and left foot . a) Generated template. b) Corresponding mathematical mean template. c) Difference of a) and b). Thereby, dark red represents the maximum value whereas dark blue represents the minimum.

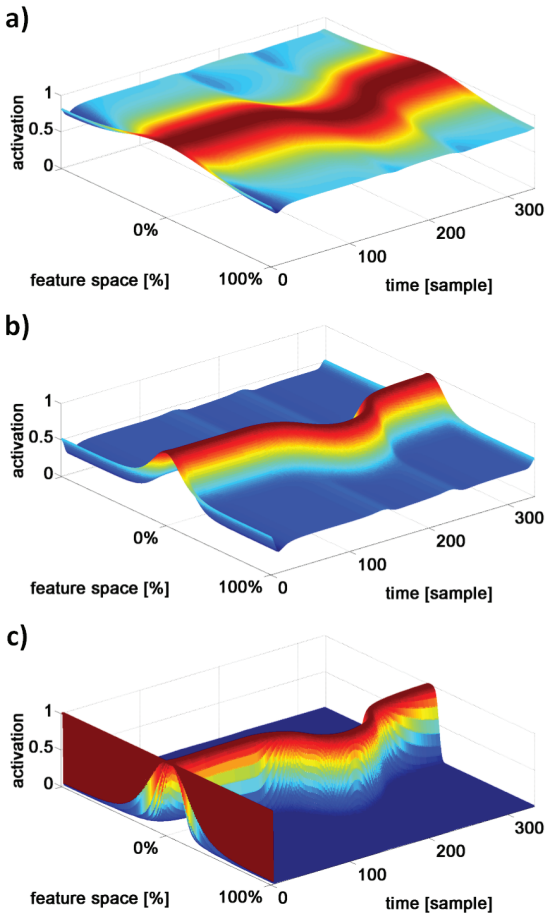


**Figure 3.19:** Influence of the adapting kernel for increasing confidence. a) Preshape adapted with a kernel having almost $0\%$ confidence input. b): Confidence is increased to $50\%$. Finally, c) shows the adapted preshape by $100\%$ confidence, which corresponds the original preshape.

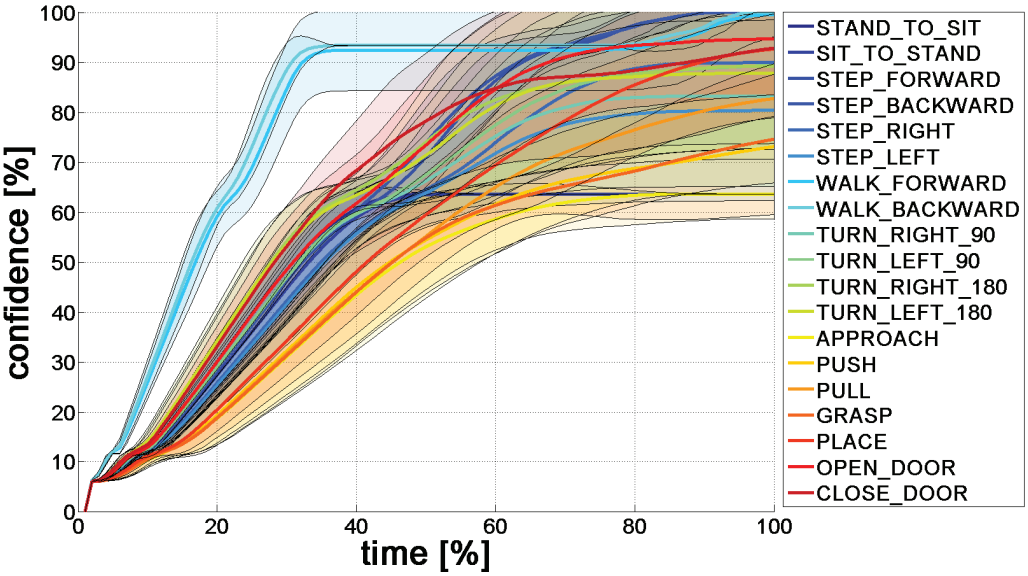


**Figure 3.20:** Comparison result of all primitive action against STEP_FORWARD. The colored lines represent the mean confidence of the corresponding actions (see legend). The shaded areas around each mean shows the variance. Time has been normalized with respect to the length of the recordings.

the same time as can be shown in Fig. 3.21(b). As one TARM reaches a confidence of over 0.8, a decision is made and an action is then recognized (as shown in the instances marked by the red ovals). The combination of the CARS and TARS then gives a semantic understanding of what are the actions that are being observed. The results of the action understanding system are given in Table 3.4. In this example the system understands the movements as follows: stand up at couch (0-2.4 seconds) then step forward by the coffee table (2.4-3.6 seconds), turn stepping left towards the TV table and approach and grasp the remote (3.6-5.2 seconds), then step forward towards the couch (5.2-6.5 seconds) and finally sitting down on the couch and simultaneously pushing to place the remote over the coffee table to end the movement (6.5-7.2 seconds).

The combination of the two systems alongside the dynamic affordance logic system allows for an end-to-end biologically-inspired architecture for human action understanding. The complete system would benefit from an extensive validation given a large human behavior dataset as well as human behavioral studies in intention and action understanding. However, due to space limitations, in this work we focused on presenting the building blocks (TARS and CARS) and their interconnection. We tested the blocks individually and provided initial results of the integration of these systems to provide an insight on the dynamics of decision making. Future work would focus on an extensive validation of the overall architecture. Validation should avoid static representations such as confusion matrices and focus on using new dynamic metrics that measure the conflict between different competing hypotheses of action understanding. Further metrics should measure the interaction between the TARS and CARS modules and measure the benefit to complexity

**Table 3.3:** The pick the remote scenario: ground truth

| Start (seconds) | End (seconds) | Action | Furniture | Object |
|---|---|---|---|---|
| 0 | 0.84 | sit | couch | |
| 0.85 | 2.35 | sit-to-stand | | |
| 2.36 | 3.28 | step-forward | coffee table | |
| 3.29 | 4.32 | step | coffee table | |
| 3.29 | 4.32 | grasp | TV table | remote |
| 4.33 | 5.22 | step | | |
| 5.23 | 6.09 | step forward | | |
| 6.1 | 6.44 | turn right 90 | | |
| 6.45 | 7.59 | stand to sit | couch | |
| 7.6 | 8.83 | sit | couch | |

**Table 3.4:** The pick the remote scenario: results

| Start (seconds) | End (seconds) | Action | Furniture | Object |
|---|---|---|---|---|
| 0 | 2.4 | sit-to-stand | couch | |
| 2.41 | 3.6 | step-forward | coffee table | |
| 3.6 | 5.2 | step left | | |
| 3.29 | 5.2 | approach | TV table | remote |
| 4.6 | 5.3 | step left | coffee table | |
| 5.2 | 6.8 | step forward | couch | |
| 6.1 | 6.44 | pull | coffee tabele | remote |
| 6.8 | 7.5 | stand to sit | couch | |

ratio of combining both signals for a correct and early action understanding. Thus, the proper evaluation of the developed system and definition of metrics constitutes an own research question which will be addressed in our future work.

## 3.9 Discussion

There is an infinite set of intentional descriptions consistent with any given behavior stream. However, even though there exists a large state space of possible interpretations, adults seem to be skilled at agreeing about the semantics of an observed action to a detailed description [5, 8]. Even from a young age, we are able to understand actions (e.g. grasping, pointing and gazing) and attribute a meaning behind them accordingly [191]. These social abilities of action, plan and intention understanding that we posses as humans allow us to socially interact with others around us. We introduced an attention shift model that has an application in the CARS and a trajectory comparison model that has applications in TARS. We also introduced how the link between CARS and TARS could be logically motivated using the concept of affordances and connectivity fields.

A biologically motivated approach for feature selection and generation was discussed. While the features in this work were calculated for a generic 1.8 m tall male, given the true height and weight of the observed actor, the whole anthropometric measures (and thus the features) can be derived by means of correlation formulas [192]. The features calculated, encode relations between different joints in the body. It would be beneficial to devise a system that dynamically switches between different feature sets for enhanced recognition and reduced computational load. Considering features that encode end-effector–objects relations could also be in line with the current work and would enhance recognition rates. Overall the implemented 39 features were sufficient to test the current system and produce the results as seen in the results section.

The features themselves were represented and fed into the DFT architecture using a biologically motivated approach, namely the DPA method which integrates naturally with the concept of dynamic neural fields [193]. We focused on representing the tuning curves in a way that is consistent with neural response studies in literature. The assumption that tuning curves are the same across the population is a limiting one. Indeed, it might be the case that the shape of the tuning curve can be different. Moreover, our assumption of equally distributed tuning curves across the feature space is simplistic and may be not biologically plausible. We assumed that the tuning curves are the same across the population as well as being equally distributed over the feature space as a simplification. Further, work on how and what it means for the optimal response values (both in value and quantity) to be optimally distributed along the feature space might allow for a more meaningful stimulus generation for the DFT architecture.

An attention-shift model was developed for the purpose of context understanding in action recognition tasks. The bias introduced by the CARS aims to reduce the overall computational power of the system. The idea that an observer's expectation of a movement effects how the intention behind it is understood has been shown previously in literature [194]. Furthermore, the need for a top-down mechanism to constrain intentions of an actor has been discussed in [122] where the Gricean pragmatic analysis of language (specifically

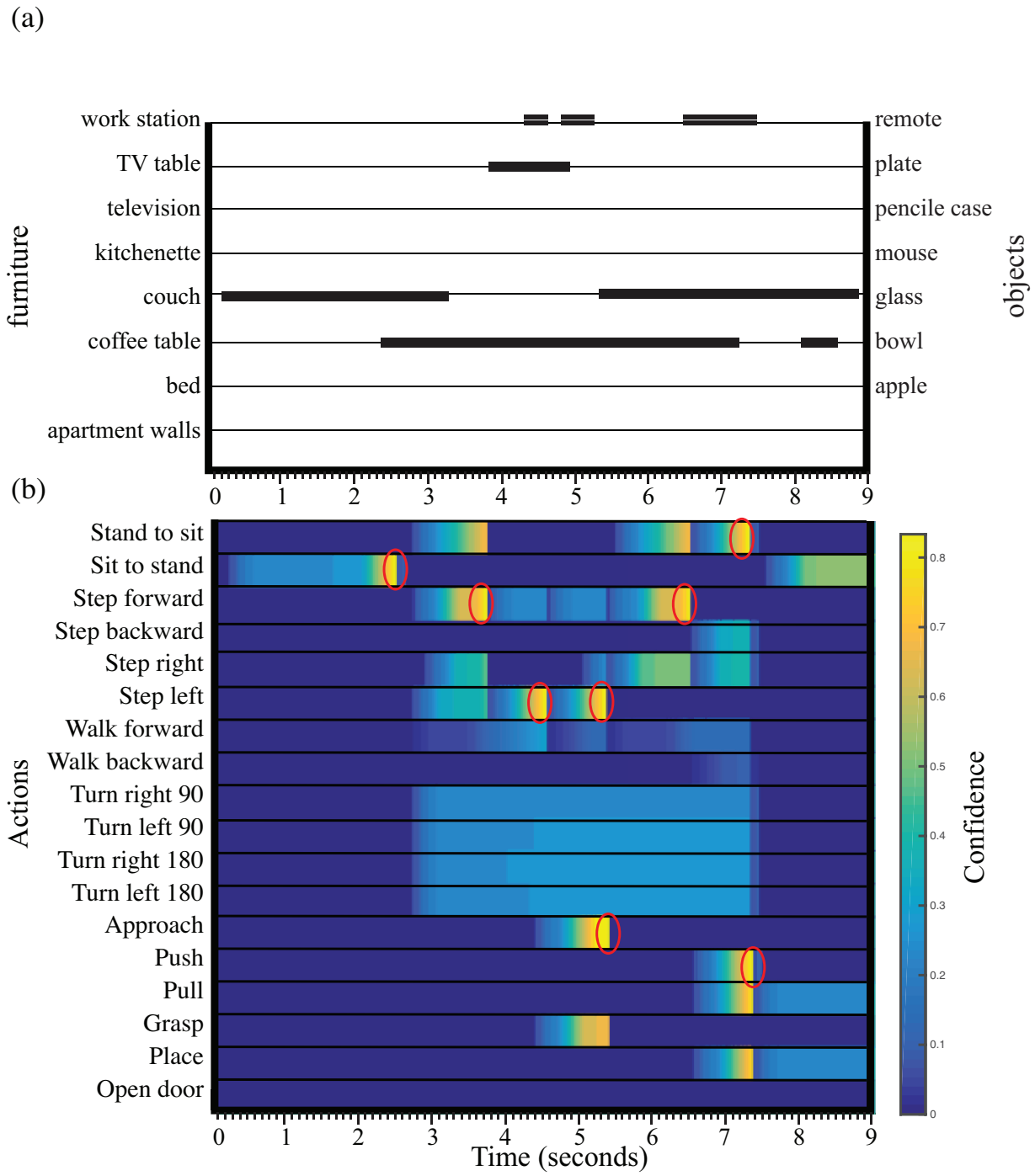


**Figure 3.21:** Results for the *pick up remote* scenario (a) Results of the CARS indicating the interaction of the right wrist with the furniture (left, solid lines) and objects (right, solid dashed lines). This indicates that there was interaction with the couch at the beginning and the end of the complete action, with interactions with the coffee table and the TV table in the middle of the complete action. (b) Results of the TARS. Many TARMs are online and comparing the observed movement dynamically, once one of the systems in competition achieves an accuracy of over 0.8, then all systems are reset and wait for CARS to bias the next round of comparisons. The red ovals indicate a decision made.

the reality and and cooperative principles) were used as the constraint to the understanding of simple, goal-oriented actions.

In terms of neural plausibility, the online computation of the optical flow is problematic however. This is because the online calculation of the optical flow would require rapid and precise plasticity in the synapses that implement lateral interactions. As such, the implementation of CARS should be seen as an algorithmic shortcut for a more complex neural system that could generate moving peaks as described in the CARS implementation.

The TARS subsequently load only a few preshapes that are dependent on the input from CARS. Furthermore, as internal comparison is the basis of the TARS, the current implementation depends on a learned memory of how the motion evolves. The template generation methods produce preshapes that are useful for the comparison process. However, two major issues with the production of template preshapes had been observed and been tackled, namely the branching and widening effects. Branching occurs when there are multiple ways of performing an action kinematically (in contrast to having one way with small variances in motion). In this occurrence we can observe a branch in the preshape that starts from a common point and ends separately. The branching effect has been solved by post-processing these preshapes into a DNF that ultimately picks between branches (the one with most activation/ in this case that has been seen most) and eliminates the other. The other issue is widening, which refers to the fact that the preshape can take wide range of features at some parts due to large variations in the performance of an action. These wide areas usually survive in the post processing procedures and could facilitate faulty detection. This has many limitations, specifically an action recognition system can not house all possibilities for the same action (different speeds/ extensions) that could encode the same action class. We have tackled this problem by trying to adapt the preshape dynamically as well as using a temporally invariant comparison method (traveling waves and extracting snapshots within the learned memory/ preshape).

The CARS and the TARS are brought together such as to limit the search space using ideas of affordances embedded in the connectivity matrix. Using affordances, however, is not without complexity. Further, work should focus on how objects' action potentials are perceived and modeled within DNFs, similar to that presented in [195]. Furthermore, detecting and learning new action abilities (primitives) is also not implemented. However, if affordances could be attributed in future work, new actions that trigger these affordances could be learned online given affordance understanding.

Given the above discussion, we describe in the following how the different modules interact with each other using the "pick up remote" scenario presented in the results section. Initially, as the observed agent moves around in the environment, its skeleton is transformed onto the observer's egocentric coordinate frame. Furthermore, the *Body Joint Extension* and the *Projected Relative Angle* features are calculated.. As the pelvis and wrists of the agents move, they provide input to their individual CARMs to detect a manipulation movement (towards an object/ furniture) or a locomotion movement (towards a virtual object). In our example the agent is interacting with the *couch*. The CARS makes a decision that the couch is being *manipulated*. This affects the affordance logic block to activate the TARMs that are related to the couch e.g. sit-to-stand action or stand-to-sit. The TARS loads the appropriate TARMS (sit-to-stand action and stand-to-sit), allowing

the prehshapes of each action across the different features to be loaded. The comparison occurs in each of the TARMS relating to the action/feature pairs against the observed motion as discussed in the comparison block in section 3.5.2. A decision is achieved within the TARS as one of the TARS achieves an accuracy of 0.8 or above. This resets the system and waits for the CARS for the next input. In this case it is recognized as a locomotion action (the attention shift was towards a virtual object), which forces several locomotion TARMS to turn on as well as the new affordance of the coach (sittability, now that the couch is available to be sat on again). The next round of TARS detects that the agent is stepping forward, and so on until the end of the complete series of intentional actions.

Compared to MNS computational models, our model resembles the HAMMER architecture in that we do not emphasize a motor control role in the current implementation. This is in contrast to the MOSAIC model that was conceived for purposes of dynamic motor control.

In terms of input, the kinematics of certain joints of interest is used in our model similar to the MSI model. However, unlike other implementations we explain how features can be represented in population of neurons for the purpose of action recognition and the generation of long-term memories for each class of actions.

Similar to the MNS and MSI models, we present a model that gives a central role to the objects in the environment and adopts a object-centered representation. We give this representation further importance and build the CARS to extract information of attention shifts towards objects, select them, read out their affordance and allow this information to bias the TARS. Goal-setting then is a focus in our model, while it is not addressed in MOSAIC, HAMMER and RNNPB models. While other models might allow for goal-setting explicitly it is not an automatic procedure by any means and the link to the object affordances and motion parsing is not well established, which is what we focus on in our implementation.

Projection of the acting agent to the observer is a main block in the TARS which allows the system to be agent-independent and complies with the ideas of "internal simulation" and "motor resonance". This self-observation mechanism is also shared with the MNS and MSI models [196]. However, unlike its use in the feedback-loop for action generation in the MSI model, our implementation uses self-observation in our implementation such as to associate the observed stimulus in an associative memory manner to achieve action understanding. We also address how spatio-temporal variance between the stored long-term memories and the observed data could be handled using dynamic neural fields and to obtain a correct classification of the correct motion. Tackling this spatio-temporal variance/similarity between same/different class of actions has not been tackled in the mirror neuron computational models and is vital for the correct classification.

All of the discussed models employ a metric to calculate the similarity between the observed or generated (learned representation) of the action. While RNNPB operates on a parameter space, similarity is calculated based on the distance between the calculated and observed actions. The HAMMER architecture defines similarity based on the completion of the goal. The MSI model, similar to the MOSAIC architecture, simply calculates instantaneous error (or what is called the responsibility signals in the HAMMER model) based on the difference between the predicated and observed movement. These three architectures,

namely HAMMER, MSI AND MOSAIC, in contrast to the RNNPB operate on trajectory space and thus are able to calculate the similarity metrics based on the observed/generated motion trajectories [197]. In our model, we obtain a classification decision in two steps. First using the CARS selection of the object and the actual possible action affordances available at that time step. Secondly, the motion trajectory is parsed in a second step and a decision is made based on the overall activation of a neuron population representing the stored memories of the actions, and how far the traveling wave propagates in that structure.

The setup we proposed within our model allows for online action recognition. Online recognition can also be achieved in the MSI and HAMMER architectures. It can also be achieved in the MOSAIC architecture given the possibility of comparison between different responsibility signals. In terms of verification, our model evaluates the results on real data of an everyday life scenario. Out of the models reviewed, RNNPB and the HAMMER approach used real data as opposed to simulated data used by the other models.

Compared to similar work in literature, we presented a novel predictive system within DFT that models attention-shifts and pairs up with a trajectory parsing system in a second step. Special focus has been given to how kinematic trajectories are introduced into DFTs and how comparison could be performed regardless of possible spatiotemporal variations between the performed and saved representations of the actions.

## 3.10 Conclusions

Overall, the AU architecture in this work presents, for the first time, a novel predictive system within DFT that models attention-shifts and pairs up with a trajectory parsing system in a second step while utilizing an affordance logic system. Compared to the state-of-the-art, the AU architecture in this work combines both context recognition and trajectory recognition alongside affordance logic rather than opting for one or the other solely for task of action understanding. Furthermore, the action recognition systems presented in this chapter, and their underlying subsystems, are modeled using concepts within DFT rather than combining different methods as is prevalent in other approaches in the literature.

The action understanding systems proposed in this chapter were realized using DNFs and are novel within DFT. The first of which, TARS, takes information of movement kinematics. The CARS on the other hand takes information of movement kinematics, object locations as well as affordances in the environment. The TARS produced cognitive decisions that answer questions of what is the action that is being performed, where it is being performed and towards which object. The success of the two systems stems from the tight, dynamic coupling between the environment and the decision making units. This allowed for the production of contextual information necessary for further processing. The initial results generated using the integration of the two systems provide an important step towards a robotic cognitive ability of mental state estimation and intention understanding. Further work should focus on further validating the complete system using the recorded dataset as well as extending the realized system with action production modules to augment the long-term memory templates that are currently being used within TARS.

# 4 Internal simulation of reaching motion

Within the embodied situated cognition stance, intelligent human behavior can be understood as the adaptive response an agent produces due to the tight coupling between the agent's body, the environment and the agent's decision making processes [1, 2]. As the agent's decision making processes develop an intention, a series of intentional actions are sequentially produced to fulfill that intention within the current environment. This mutual interaction between the agent and the environment dynamically influences the equilibrium state of the agent. Accordingly, these shifts in the equilibrium states (neural control signals) that are shared between the agent and the environment results in voluntary, intentional motor actions. This is the basic premise of the equilibrium point hypothesis (EPH), more generally referred to as threshold control theory (TCT) [92, 198]. The result of the top-down systems, discussed in the previous chapter, are used to decide on the object of most and current interest. In this chapter we present an algorithm that calculates the emerging trajectories that bring the hand towards this object. Explicitly, this chapter presents the concept of internal simulation of a reaching motion using TCT as an alternative to comparing observed motion against saved memories (which was discussed in the previous chapter). Assuming that the understanding of reaching movement is explained by the direct matching hypothesis within MNS, we answer the question of how the internal simulation of a reaching movement is performed given the initial movement information and the context of the motion.

Specifically, we model the dynamically generated internal simulation signal with the reciprocal $R$ command of the end-effector as explained by the threshold control theory [92]. This $R$ command is modeled using a dynamic attractor system and is also validated within our work on a musculoskeletal arm model as explained by the threshold control theory. This is in compliance to descriptions within MNS in which the internally simulated motion should be identical to the one generated when actually performing the action as opposed to just understanding it. Therefore we model the *referent control block* that dynamically generates the signals required to produce a reaching motion in a *musculoskeletal arm model* given the *objects* in the environment and the *hand position*. This is presented in Fig. 4.1, where we extract the internal simulation block from the overall AUA as illustrated previously in Fig. 2.1.

Compared to the related work in [136, 199–203], we present for the first time, a novel biologically inspired referent control formulation using dynamic system theory [1, 2], given information of the initial direction of the end effector during a reaching movement as well as the positions of goal/obstacles in the environment. We explicitly use attractor dynamics to define the equilibrium hand trajectories. The model links the formulation of the equilibrium trajectory with the environment, allowing elements of obstacle avoidance and controls the duration of shifts according to task completion.

The rest of this chapter is organized as following. We present the related work and

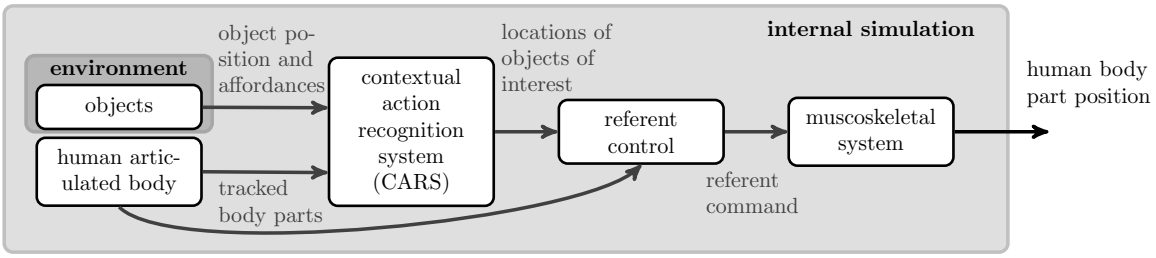


**Figure 4.1:** Reaching motion simulation modules and their connections.

highlight our contribution with regards to it in section 4.1. Next, we present our contribution in a novel biologically inspired referent attractor dynamics system to model the dynamic equilibrium trajectory as explained within referent control and TCT and highlight how obstacle avoidance can be achieved in the same formulation as simple point-to-point movements. This is presented in section 4.2. We validate this equilibrium trajectory formulation using a musculoskeletal system simulation and present the results of the simulation in section 4.3. Finally, we give a discussion in section 4.4 and present the conclusions in section 4.5.

## 4.1 Related work

Within TCT, the role of the central nervous system (CNS) is not to specify motor commands such as trajectories, forces, stiffness, velocity etc. but it is rather to define a central command, consisting of the referent configuration that influences ranges in which neuromuscular elements are active. The referent configuration is generally composed of the threshold position $R$ and referent coactivation command $C$. The threshold position $R$ is defined in a field of possible spatial configurations within a frame of reference $RF$. There exists many frames of reference at different levels of abstraction depending on the desired action in the environment e.g. whole body movements are defined under the referent body configuration $R_b$, hand grasping action are defined under referent hand aperture $R_h$, etc. For example, the referent arm configuration $R_a$ is the origin of a personal $FR$ that consists of all possible arm configurations $Q_a$ in the immediate environment. Ultimately it is the difference between the actual arm configuration $Q_a$ and the reference arm configuration $R_a$ within this $FR$ that leads to shifts in the referent arm location, which in turn decreases the threshold lengths $\lambda$ of muscles. This generates activity and forces leading to a reaching action in the environment. Within this setting, the referent $C$ command contributes in defining a range around the threshold position $R_a$ in which the antagonistic group of muscles are active. The description given above is the basic premise of the *minimization rule* that explains how the redundancy problem is addressed within TCT. Explicitly that there is no redundancy in choosing the specific muscles required to perform any specific action, the solution to which e.g. muscles are required and level of activation emerges automatically as a function of the current $RF$, $Q$, and the control of referent variables $R$, $C$ and $\lambda$ [204].

The idea of referent control is generalized over many actions by defining a *referent body configuration*. Within this work we focus on intentional arm reaching movements in the transverse plane. Explicitly, we focus on the relation between referent, equilibrium and hand trajectories. We briefly explain the difference in the following. During an intentional reaching action an equilibrium point (EP) trajectory is generated. The EP characterizes the set of joint angles and load torques required to produce this reaching motion. The gradual shift of the equilibrium position of the hand (EP component) generates an equilibrium trajectory during this motion. The equilibrium trajectory itself, just like the actual position of the hand, is not predetermined. They are emergent properties of referent control, and arise due to the dynamic tendency of the arm to follow the equilibrium trajectories that emerge due to the interaction of the body with the environment. There has been many attempts to formalize theses equilibrium trajectories leading to a debate on how complex or simple they might be. It has been shown within TCT that the equilibrium trajectories are emergent properties of the dynamic reaching action, that are not isochronous with the actual hand movement (taking 1/3 of the time) and can range from short-last monotonic movements in simple point-to-point movements to complex non-monotonic in more complex reaching movements such in the case of e.g. obstacle avoidance [204].

Previous work focusing on modeling equilibrium trajectories can be found in [136, 199–203]. Generally it was shown that fast point-to-point reaching motions result from fast, monotonic simple trajectories. Flash calculated static stiffness values and simulated reaching motions using an arm model in [199]. The form of the hand equilibrium trajectory produced by the neuromuscular system was assumed to follow a simple linear model. Results showed that this formulation predicted hand trajectories that were similar in qualitative features and quantitive kinematic characteristics.

Dynamic mechanisms were incorporated in the work of Flanagan et.al. in [202] to generate the equilibrium trajectories. This was evaluated by comparing the simulated and actual trajectories of movement. The results suggested different rates of shifts for the $R$ and $C$ command that can be utilized in different portions of the work space.

A monotonic ramp-shaped model for the formulation of the $R$ command was used in [203]. Different movement distances were simulated by modulating the duration of shift in the equilibrium state. The results showed that both the empirical and model data were similar and that the neural control process generating such shifts in equilibrium states preceded the end of the actual movement. Furthermore, neither the timing nor the amplitude of electromyographic signals are pre-planned, but rather an emergent response of the central, reflex and mechanical components of the system that emerge dynamically due to the shift in the equilibrium state.

The timing pattern of the $R$ and $C$ commands were discussed in [136]. Results showed that the equilibrium shifts indeed terminate ahead of movement completion.

As discussed earlier the idea of threshold position is generalized to many types of actions an agent might perform to influence his/her immediate environment or as a reaction to changes in the environment. Taking that into account, the range of threshold positions can be expanded towards goal-directed movements in this environment. As such, the position of the hand is to be coupled with the environment such that decisions on manipulating specific objects drive the action of e.g. reaching. To that end, we utilize a meaningful

decision making system by using formulations within Dynamic Field Theory (DFT) that integrates seamlessly to perform selection tasks on objects in the environment. The output of this decision framework defines the set points (attractors and repellers) that are required by the attractor dynamic system which models the equilibrium trajectory that guides the neuromuscular system towards the reaching action and away from obstacles.

Related work in which DFT was connected to neuronal dynamics for goal-directed movement generation was proposed in [205]. The model integrates timing patterns within DFT for motion generation, as well as feedback of the sensed joining configuration. The movement predicted by the model was compared against experimental data collected from participants. Results indicate the presence of self-motion that does not move the end effector, and is linked to the curvature of the resulting end-effector movement.

Compared to the related work discussed in this section, we present for the first time, a novel biologically inspired referent control formulation using dynamic system theory [1, 2], given information of the initial direction of the end effector during a reaching movement as well as the positions of goal/obstacles in the environment. The model links explicitly the formulation for the equilibrium trajectory with the environment, allowing elements of obstacle avoidance and controls the duration of shifts according to task completion. We explicitly use attractor dynamics to define the equilibrium hand trajectories. The motivation behind such an approach is to define an attractor point of the dynamic system as the state the system is trying to reach, and allow the actual hand trajectory to emerge dynamically as the solution of this dynamic formulation.

## 4.2 Methods

In this section, the different modules that are used to simulate a reaching motion and their connections are discussed. The modules are shown in Fig. 4.1. The current location of the end-effector (hand) $(x_h, y_h)$ as well as the positions of different objects $(x_{obj}, y_{obj})$ (goals/obstacles) in the environment are used as an input to the decision making system. The decision making system dynamically represents the environment, classifies objects into obstacles and goals $(x_{o,g}, y_{o.g})$, and holds these decisions in the working memory. This process is illustrated in the *decision making* block and is discussed in section 4.2.1 where the contextual action recognition system (CARS), presented in section 3.4, is used as the decision making system for the internal simulation of the reaching motion. Following the information of the locations of the end-effector/goal/obstacles from the *decision making* block, an equilibrium trajectory emerges dynamically due to the process of referent control. The equilibrium trajectory guides the motor control units driving a musculoskeletal arm model to the final desired configuration as explained by TCT. This process occurs in the *referent control* block as shown in Fig. 4.1. The *equilibrium trajectory* $(R_x, R_y)$ that emerges from *referent control* is modeled using attractor dynamics and is discussed in section 4.2.2. The reference trajectory is forwarded to the *musculoskeletal system* block for motion generation. The musculoskeletal arm model that is used to validate the dynamic equilibrium trajectory is discussed in section 4.2.3.

As discussed previously, the referent control command is a dynamically evolving command. We model it using the dynamic approach to behavior generation which is based

on dynamic system theory and neural components from neural fields [206]. This approach on which we base our work upon utilizes attractor dynamics for behavior generation originally developed for planning and controlling autonomous robots [206–208] and robotic arm control [209].

The dynamic approach to behavior generation is based on three concepts. Firstly, behavior is described and constrained by defining a set of task variables (e.g. heading direction of reaching movement, velocity of reaching etc.) called *behavioral variables*. Secondly, behaviors are generated dynamically as solutions of dynamical systems where the set points (attractors/ repellers) represent desired/undesired behaviors along the dimensions of the behavioral dimensions. This concept is referred to as *behavioural dynamics* and motivates our work in *referent control* presented in section 4.2.2. Thirdly, the information about the values of the attractors and repellers are neurally represented by the use of neural field dynamics. This concept is referred to as *neural representation of information* and is discussed in detail in the next section.

### 4.2.1 Decision making using dynamic field theory

Decision making in the context of reaching motion is the process of selecting or withholding an action based on its importance, relevance and effect, given a specific context within the environment the agent is embedded in. The contextual action recognition system, presented in section 3.4, is used for the purpose of decision making within the task of dynamic generation of reaching motions, where stable states of heading direction is to be maintained based on the hand position and environmental information of obstacle/goal locations. The CARS gives an output of the locations of objects of interest required for the calculation of the emerging trajectories. Additionally, sensory information of the locations of obstacle/goal should be maintained in the working-memory for further processing into attractor/repeller states such that the sensory-motor interface is adequately defined.

### 4.2.2 Referent control using attractor dynamics

As discussed in the related work, section 4.1, referent control dynamically shifts the referent hand position $R_h$ to produce a reaching motion. We model the referent hand position $R_h$ using the dynamic approach to behavior generation which is based on dynamic system theory and neural components from neural fields [206–208].

In the following we describe the referent hand position $R_h$ model for reaching motions based on the attractor dynamics approach. Furthermore, we discuss our choices for the behavioral variables as well as the dynamical system that drives the arm towards desired objects and away from obstacles.

#### Dynamic attractor system

The basic premise within this work is to design a referent hand dynamical system around the hand position that directs the arm movement towards the goal objects while taking the whole arm configuration into account. The dynamic referent shifts resulting from the behaviour of the hand acts as the equilibrium trajectory that is fed into the motor control

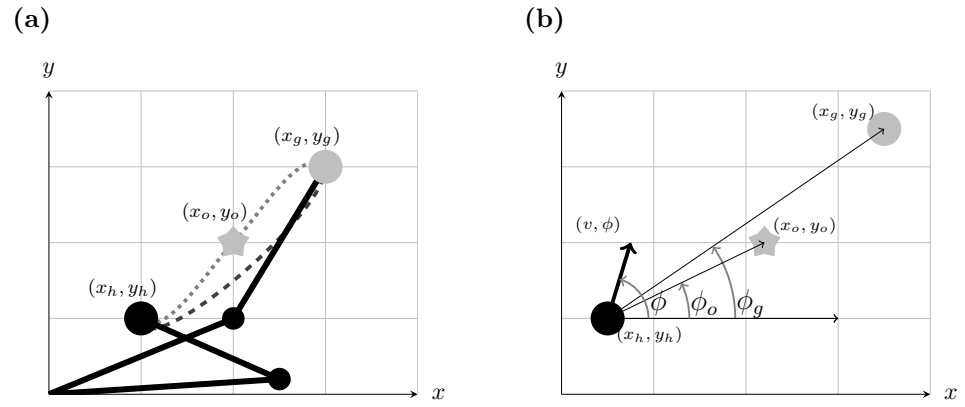


**Figure 4.2:** (a) The setup of the attractor dynamics system. The task of referent hand dynamical system $(x_h, y_h)$ is to drive the arm to the goal position at $(x_g, y_g)$ with a trajectory which is usually characterized by a slightly s-curve. It should also be able to avoid any possible obstacle e.g. at position $(x_o, y_o)$ while taking the whole arm into account. (b) The behavioral variables of the referent hand location.

elements of the musculoskeletal system as described by TCT to be realised by the arm model.

A simple reaching motion is shown in Fig. 4.2(a). It is composed of rotations at both of the shoulder and the elbow to guide the hand location from the initial position $(x_h, y_h)$ to the goal position $(x_g, y_g)$. A point to point reaching motion is characterised by a slightly curved line directed towards the goal. This is illustrated by the grey dotted line in Fig. 4.2(a). In the presence of an obstacle, as shown in Fig. 4.2(a) at position $(x_o, y_o)$, the arm is expected to move towards the goal and avoid collision with the end effector and the arm. The referent hand dynamical system should account for the whole arm such as to avoid collision with the objects as reaching is performed. A challenging setup of hand, goal and obstacle positions is shown in Fig. 4.2(b), in which the obstacle is under the straight line that connects the hand and the goal position. In the case that the width of the referent hand allows a straight line plan, the forearm would collide with the obstacle if the complete arm configuration is not taken into account.

A successful referent controller therefore requires fulfillment of several criteria. Firstly, it should allow for straight line trajectories, while dynamically avoiding obstacles as they come into view as well as adapting to changes in goal position. Secondly, it does not allow for a collision to occur with the rest of the arm. Additionally, and as observed neuronally, the final equilibrium trajectory should spatially close to the actual trajectory. However, it would not be isochronous. That is the equilibrium trajectory would lead the actual trajectory. In the case of simple reaching motion the equilibrium trajectory would be three times faster than the actual trajectory and reaches the goal location as the tangential velocity of the hand reaches its peak velocity [198].

In order to achieve these criteria we designed a dynamical system that would successfully plan the referent shifts. The first design parameter, the behavioral variable, describes

the evolution of the dynamic reaching motion behavior. In order to be compliant with a physical hand and human reaching motions, behavioral variables are chosen to be continuous and observable by a sensor. Motivated by findings in behavioral and neural studies, in which reaching motions are found to be specified with both a direction and amplitude, we pick our behavioral variable to be the heading direction $\phi$ and the movement velocity $v$. [210]. The behavioral variable $\phi$ is shown in Fig. 4.2(b). The desired values for $\phi$ are towards $\phi_g$ and away from $\phi_o$. The amplitude of the movement $v$ (referent speed of hand motion) and the heading angle $\phi$ (referent direction of hand motion) are related to the $R_h$ command given

$$\begin{aligned} \dot{R}_{h,x} &= v\cos(\phi) \\ \dot{R}_{h,y} &= v\sin(\phi). \end{aligned} \tag{4.1}$$

The above formulation transforms the referent direction and speed of hand motion into the $x, y$ direction. This was chosen as we focus in this work on two dimensional reaching motions in the transverse plane. This can be generalized to three dimensional reaching motions by adopting a spherical coordinate system $(r, \theta, \psi)$ as described in neural studies [211] and by accounting for gravity in the formulation of the referent command as discussed in [204].

### Heading angle behavior variable attractor dynamics

The behavioral variable evolves based on solutions of dynamic equations whose steady points are the set of desired/undesired states we would design the system to achieve/avoid. This is mathematically equivalent to the equation:

$$\dot{\phi} = \sum_{1}^{i} f_{tar,i}(\phi(t)) + \sum_{1}^{j} f_{obs,j}(\phi(t)) + \sum_{1}^{k} f_{arm,k}(\phi(t)) + f_{stoc} \tag{4.2}$$

where multiple elementary behaviors are integrated. Namely goal heading, obstacle avoidance and arm awareness in the $f_{tar}, f_{obs}, f_{arm}$ respectively. Stochastic noise is also integrated in the model by way of $f_{stoc}$. The indices $i, j, k$ indicate the possibility of having more than one target, obstacle and arm location, that are integrated into the behavior generation by way of burification theory [212].

The goal heading function $f_{tar,i}$ is designed to be a nonlinear dynamical system with a fixed point at the desired goal location $\phi_{g,i}$, and is mathematically defined in (**??**), it is also illustrated in Fig. 4.3(a). The nonlinear function $f_{tar}$ is a product of a Gaussian function and a linear function. The slope $m$ at the fixed point $\phi_g$ is set to be negative such that an asymptotically stable state is achieved. This would guarantee that when the system is close to the attractor and would always sit in the basin of attraction, such that all initial states of the behavioral variable would converge to the attractor value and tracks it as it changes dynamically

The obstacle avoidance function on the other hand is defined in (4.3), and is illustrated in Fig. 4.3(b). It is a product of three terms. The first term is similar to the attractor function in that it is a positive slope linear function that is limited in range by a Gaussian function. The slope is determined by a factor $1/\Delta\psi$ that represents the angular size

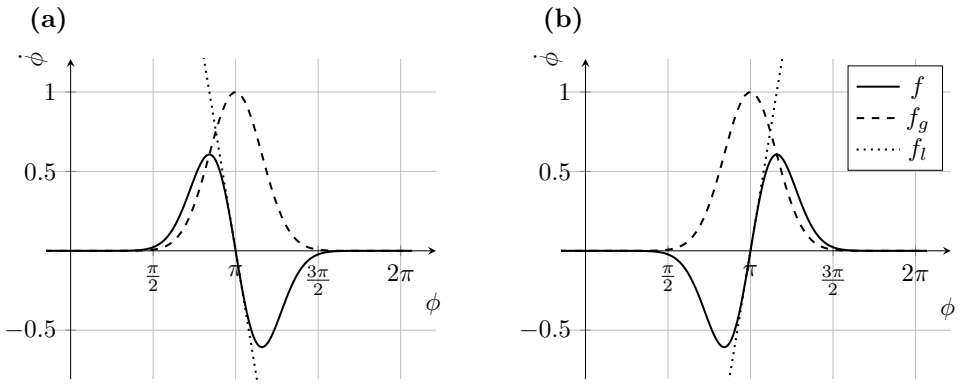


**Figure 4.3:** (a) A nonlinear attractor system that is a product of a linear term (dotted black line) and a gaussian term (dashed black line). The linear term is characterized with a negative slope $m$ and an a zero solution at the desired behavioral variable (in this example $\pi$, where the system would be heading towards a goal). The width of the Gaussian defines the area of effect. (b) A nonlinear repeller system that has a positive slope, as apposed of the negative slope of the attractor. The dynamic system drives the system away from the fixed point $\pi$, which is in this case an undesirable behavior (heading towards an obstacle).

(angular range that the obstacle is active as a repelling force) of the obstacle. The spatial term controls the contribution of the overall repulsion strength as function of the sensed obstacle distance $d_{obs}$. The exponential function's parameters $R_{obs}, R_{hand}$ are the radius of the obstacle and radius of the hand, respectively. While $d_0$ is a form factor that shapes the exponential function. The angular range term on the other hand determines the strength of repulsion based on the visibility of the obstacle. It is a sigmoidal function with model parameters $h_1$ and $\delta$ and its function is to control overall repulsion strength to be nonzero when the hand is facing the obstacle. Otherwise, the repulsion strength would be zero

$$f_{obs}(\phi) = \overbrace{\frac{(\phi - \psi_{obs})}{\Delta\psi} \exp\left(-\frac{1 - |\phi - \psi_{obs}|}{\Delta\psi}\right)}^{\text{repeller term}} \overbrace{\exp\left(-\frac{d_{obs} - R_{obs} - R_{hand}}{d_0}\right)}^{\text{spatial term}} \underbrace{\frac{1}{2}\Big(\tanh\Big(h_1\big(\cos(\phi - \phi_{obs}) - \cos(2\Delta\psi + \delta)\big)\Big) + 1\Big)}_{\text{angular range term}}. \tag{4.3}$$

In order to take the rest of the forearm into account, we design a function $f_{arm}$, defined in function (4.4), that contributes to the heading angle of the hand $\dot{\phi}$. We therefore model several control points that would contribute to the attractor and repeller force observed at the hand. For this task we utilize forward kinematics calculation, as defined in (4.5), to obtain the position of each point $k$ along the upper arm given the joint angles. Each position $(x_k, y_k)$ is calculated using the factor $\eta_k$ that takes values $[0, 1]$ in increments of $1/k$. Inverse kinematics in a first step allows the calculation of the joint angles $\theta_1, \theta_2$ given

the current hand position.

$$f_{arm,k}(\phi(t)) = \overbrace{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\phi-\phi_{g,k})^2}{2\sigma^2}\right) m(\phi-\phi_{g,k})}^{\text{attraction force}} + \underbrace{\frac{(\phi-\psi_{obs,k})}{\Delta\psi} \exp\left(-\frac{1-|\phi-\psi_{obs,k}|}{\Delta\psi}\right) \exp\left(-\frac{d_{obs,k}-R_{obs}-R_{hand}}{d_0}\right)}_{\text{repulsion force}} \tag{4.4}$$

$$\begin{aligned} x_k &= l_1\cos(\theta_1) + \eta_k l_2 \cos(\theta_1+\theta_2) \\ y_k &= l_1\sin(\theta_1) + \eta_k l_2 \sin(\theta_1+\theta_2) \end{aligned} \tag{4.5}$$

where $\psi_{obs,k}$ represents the dynamic angle between the $k$-th referent position on the forearm and the obstacle location. Similarly, $\phi_{g,k}$ is the dynamically changing angle between the $k$-th referent position on the forearm and the goal location. The term $d_{obs,k}$ is the distance between the obstacle and the $k$-th referent position on the forearm, which is also a dynamic variable that varies with time. Then for each arm point $k$ the correspondent $f_{arm,k}(\phi(t))$ is calculated and the contribution is added to the hand's heading angle. The main contribution of $f_{arm,k}(\phi(t))$ is performing the task of obstacle avoidance. The angular range that was used in the hand's obstacle avoidance was disregarded for $f_{arm,k}(\phi(t))$ since the lower arm movement is dependent on the hand itself.

In each time step then, the new position of the hand is calculated through dead reckoning as shown in (4.1). The virtual positions $(x_k, y_k)$ on the lower arm are calculated through inverse/forward kinematics. It is important to note that the behavioral variables and indeed the parameters continuously change in time as the dynamical system evolves. The obstacle and goal direction are dependent on the current position and is calculated such that

$$\phi_g(t) = \arctan\left(\frac{x_h(t)-x_g(t)}{y_h(t)-y_g(t)}\right), \tag{4.6}$$

and

$$\phi_o(t) = \arctan\left(\frac{x_h(t)-x_o(t)}{y_h(t)-y_o(t)}\right). \tag{4.7}$$

The final term of (4.2), $f_{stoc}$, is a stochastic force modeled as Gaussian white noise, that aims to push the system out of a repeller stable state in a limited amount of time.

### Velocity behavior variable attractor dynamics

There are two behavior variables of attractor dynamics, one is heading direction $\phi$, the other one is movement speed $v$. The goal is, to have a smooth and fast movement of the effector to the target, which resembles how a human would reach an object with the hand. The speed of movement should be restricted such that the moving attractors/repeller are capable to be tracked in time. This is important as we expect that the system is near an attractor/repeller at all times to guarantee stability. It was shown that the relation between maximum rate of shift of the attractors/repellers is $\dot{\psi}_{max} = v/d$ where $d$ is the

distance to the attractor/repeller we want to track [213]. By tuning the parameter $\dot{\psi}_{max}$ in (4.8) successful tracking can be guaranteed as discussed in [213]

$$V_i = d_i \dot{\psi}_{max}. \tag{4.8}$$

A dynamical system can then be erected over the behavioral variable $v$ as follows

$$\dot{v} = \overbrace{[-c_{obs}(v - V_{obs}) - c_{tar}(v - V_{tar})]}^{\text{velocity attractor}} \overbrace{\left[\frac{2V_{max}}{1 + \exp^{-\alpha d_g}}, -1\right]}^{\text{spatial term}} \tag{4.9}$$

where $d_g$ is the sensed distance to the goal and $\alpha$ is a shaping parameter for the sigmoidal spatial term. The estimated parameters $V_{obs}$ and $V_{tar}$ describe the maximum passing speed of the objects to the effector, regarding the current distance between them, and a fixed maximum angle velocity $\dot{\psi}_{max}$ as shown in (4.8). The parameter $V_{max}$ is defined as the maximum possible speed of the hand during a reaching motion. The factors $c_{obs}$ and $c_{tar}$ are adjusted in a way that, if the effector is close to an obstacle, $c_{obs}$ dominates, and vice versa. The factors $c_{obs}$, $c_{tar}$ are defined in the following equations

$$c_{obs} = c_{v,obs} \cdot (0.5 + arctan[c \cdot U(\phi)]/\pi), \tag{4.10}$$

$$c_{tar} = c_{v,tar} \cdot (0.5 - arctan[c \cdot U(\phi)]/\pi), \tag{4.11}$$

where the potential function $U(\phi)$ is defined to be

$$U(\phi) = \Sigma_i^n(\lambda_i \sigma_i^2 exp[-(\phi - \psi_{obs,i})^2/2\sigma_i^2] - \lambda_i \sigma_i^2/\sqrt{e}), \tag{4.12}$$

such that $U(\phi)$ takes positive values, if the heading direction leads to strong repeller presence. Otherwise, if the repellers influence is not strong, it takes negative values. It is important to set up the relaxation rate $c_{v,obs}$ and $c_{v,tar}$ as well as the parameter $\lambda_{tar}$ and $\lambda_{obs}$ in the right hierarchy to ensure a compatible behavior with the dynamic of heading direction

$$\lambda_{tar} \ll c_{v,tar}, \quad \lambda_{obs} \ll c_{v,obs}, \quad \lambda_{tar} \ll \lambda_{obs}. \tag{4.13}$$

The result of the velocity model is a steady and fast movement to the target. It is compatible with the dynamics of heading direction and also converges to the targets.

### 4.2.3 The musculoskeletal arm model

In the following we give a description of the musculoskeletal arm model that is used to validate the equilibrium trajectory that emerges from the referent control as described in the previous section. In section 4.2.3 we discuss the two dimensional planar arm model that is used, in terms of muscles, kinematics and dynamics and how it is connected to control signals for force production. Section 4.2.3 discusses muscle torque production around the elbow and shoulder joints that arise due to the referent control signals as described by

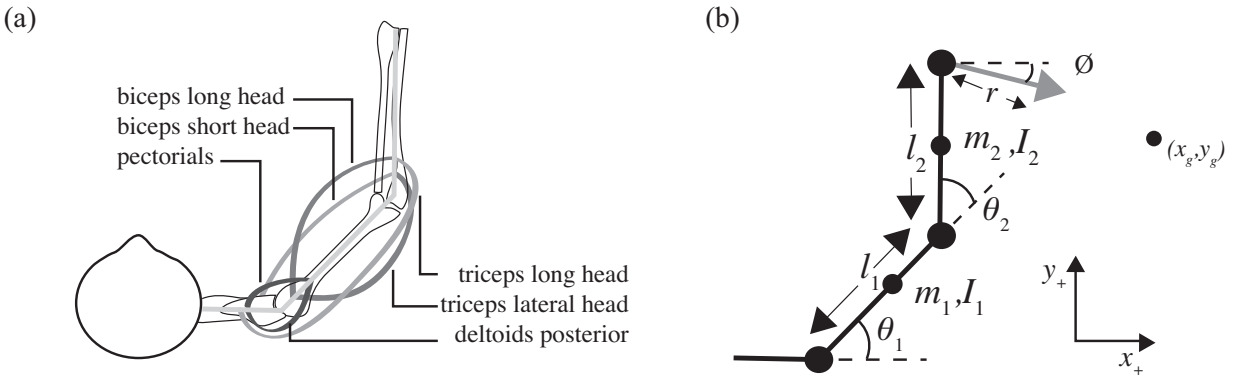


**Figure 4.4:** (a) The arm model with two kinematic degrees of freedom in the horizontal plane. The model includes six muscles, two single joint muscles around the elbow and the shoulder and two double joint muscles. (b) Kinematic and dynamic parameters of the human arm model.

TCT. Finally, the referent control signals are discussed in 4.2.3.

### Arm model

The arm model used in this work is adapted from a model previously presented in [202, 214, 215]. The arm is modeled with two kinematic degrees of freedom in the horizontal plane. It can rotate around the shoulder and elbow joint. The arm is actuated with six muscle groups. The shoulder has single joint extensors and flexors, these are the pectoralis and deltoids respectively. The elbow has both single and double joint extensors and flexors. The biceps long head and triceps lateral head are the single joint extensors and flexors respectively. While the biceps short head and triceps long head are the double-joint flexors and extensors respectively. The muscles are shown in figure 4.4(a).

Muscle insertions were calculated anatomically [216, 217]. The geometrical and inertial constants needed for the equation of motion are shown in figure 4.4(b).

The equations of motion for the arm model in the horizontal plane, that calculates the torques around the shoulder and the elbow $\boldsymbol{\tau} = \left(\begin{smallmatrix}\tau_1\\ \tau_2\end{smallmatrix}\right)$ given the angles $\ddot{\boldsymbol{\theta}} = \left(\begin{smallmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{smallmatrix}\right)$ are:

$$\begin{aligned}\tau_1 &= M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + C_1\\ \tau_2 &= M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + C_2,\end{aligned} \tag{4.14}$$

where the inertial terms $\boldsymbol{M} = \left(\begin{smallmatrix}M_{11} & M_{12}\\ M_{21} & M_{22}\end{smallmatrix}\right)$ and the Coriolis-centrifugal terms $\boldsymbol{C} = \left(\begin{smallmatrix}C_1\\ C_2\end{smallmatrix}\right)$ are:

$$\begin{aligned}M_{11} &= I_1 + I_2 + m_1(l_1/2)^2 + m_2\left(l_1^2 + (l_2/2)^2 + 2l_1(l_2/2)\cos\theta_2\right)\\ M_{12} &= M_{21} = I_2 + m_2\left((l_2/2)^2 + l_1(l_2/2)\cos\theta_2\right)\\ M_{22} &= I_2 + m_2(l_2/2)^2,\end{aligned} \tag{4.15}$$

$$\begin{aligned} C_1 = & \quad -m_2 l_1 \dot{\theta_2}^2 (l_2/2) \sin\theta_2 - 2m_2 l_1 \dot{\theta_1} \dot{\theta_2} (l_2/2) \sin\theta_2 \\ C_2 = & \quad m_2 l_1 \dot{\theta_1}^2 (l_2/2) \sin\theta_2. \end{aligned} \tag{4.16}$$

The inverse dynamics equations can be transformed and solved for $\ddot{\boldsymbol{\theta}} = \begin{pmatrix} \ddot{\theta_1} \\ \ddot{\theta_2} \end{pmatrix}$ to give

$$\ddot{\boldsymbol{\theta}} = \left(\boldsymbol{M^{-1}}\right)(\boldsymbol{\tau} - \mathbf{C}). \tag{4.17}$$

The angular accelerations calculated in (4.17) are double integrated to calculate the joint angles required to calculate current muscle lengths and moment arms as shown in the *biomechanical arm model* and the *moment arm and muscle length calculation* blocks in Fig. 4.5(a).

The muscle lengths (in [mm]) are calculated based on calculations in [218] and are based on the following equation:

$$ML = cst + t_6\theta_2^6 + t_5\theta_2^5 + \cdots + t_1\theta_2 + u_1\theta_1, \tag{4.18}$$

given joint angles $\theta_1, \theta_2$. It is worth noting that there is no distinction between the lengths of the Biceps short head and the Biceps long head, they are considered to have the same length. The same applies to the Triceps lateral head and long head. Similarly, moment arms (in [mm]) of the different muscles are calculated based on calculations in [218] and are based on the following equation:

$$MA = c_5\theta_2^5 + c_4\theta_2^4 + \cdots + c_1\theta_2 + c_0 + d_0, \tag{4.19}$$

given elbow angle $\theta_2$.

The torques, which are the inputs in (4.17), are calculated given the moment arms and the forces generated by each muscle as shown in Fig. 4.5(b).

The muscle model shown in Fig. 4.5(b) generates force depending on the muscle length and the rate at which the muscle is changing its length. Moreover, the graded development of force over time and the passive elastic stiffness of the muscle are also modeled. This is discussed in section 4.2.3 in detail.

### Muscle model

In the following we describe the muscle model used in our work. For a full description we refer the reader to the following sources [133, 219, 220]. The muscle model is shown in figure 4.5(b). The first block is the force generation mechanism. To generate forces each muscle receives an activation $A$, that according to the TCT, depends on the current muscle length $l$ and its derivative $\dot{l}$, and the threshold length $\lambda$ which is centrally defined for motoneuron recruitment. The muscle activation is then defined by

$$A = [l(t-d) - \lambda(t-d) + \mu\dot{l}(t-d) + \rho_r + \epsilon(t)]^+ \tag{4.20}$$

where

$$[x]^+ = \max[0, x] \tag{4.21}$$

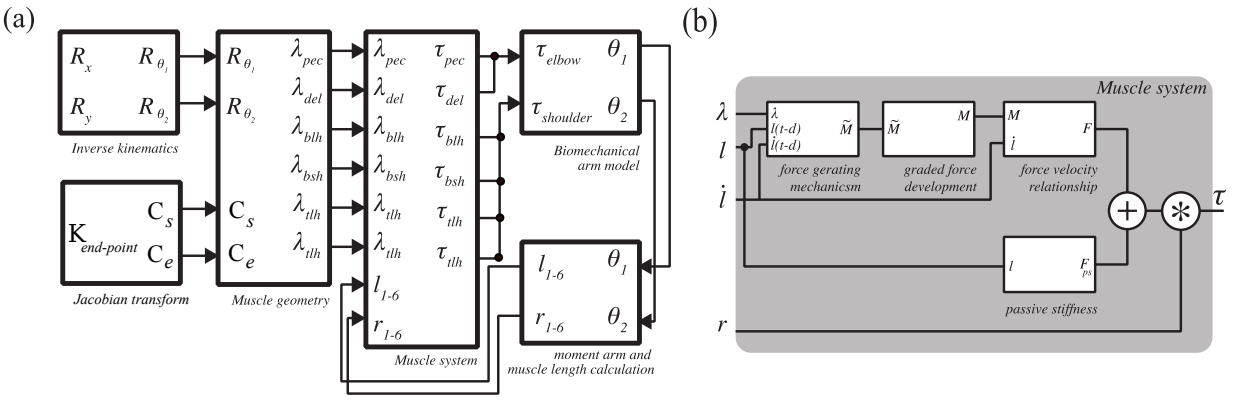


**Figure 4.5:** (a) The musculoskeletal simulation model building blocks. The simulation model is composed of an inverse kinematics block that takes spatial reference commands $R_{x,y}$ and transfers them into joint angle reference commands $R_{\theta_{1,2}}$. The muscle geometry calculates the $\lambda$ commands for each muscle given the desired joint angle. The muscle system block generates the forces in each muscle based on the difference between current and desired muscle length based on the TCT. The biomechanical arm model calculates the joint angles based on the torques around each joint. Finally, the joints angles are used for visualization as well as calculating the current moment arms and muscle lengths requires for the next simulation time step. (b) The muscle model that is used to generate force for each muscle given current and desired muscle length and moment arm. It is composed of four main blocks. The force generating mechanism produces a moment given desired and current muscle length and current muscle velocity. The graded force development models the filter-like properties of a muscle due to calcium kinetics and the force-velocity relationship is modeled in the following block. Passive stiffness is a parallel block that adds to the force. The combined force is transformed into torques around each joints.

The $d$ parameter models reflex delays, and was determined in [133] by estimating delays observed in unloading responses of human arm muscles and is the same for all muscles [221]. The $\mu$ parameter is a damping factor due to proprioceptive feedback, and models the dependency of the muscle's threshold length on the velocity [214]. The parameter $\rho_r$ models the shift in the threshold length resulting from reflex inputs e.g. inter-muscular interaction and cutaneous stimuli. The temporal changes in the threshold resulting from intrinsic properties of motoneurons are modeled in the $\epsilon(t)$ parameter [92].

Muscle forces $f$ result from changes in $A$, in an exponential fashion approximated in

$$f = \rho[\exp(cA) - 1], \tag{4.22}$$

where $\rho$ is a magnitude parameter specified for each muscle.

The $\rho$ parameter is a specific parameter for each muscle, relating to its force-generating capability. It was calculated in relation to the cross-sectional area of each muscle, in which the area was scaled by 1 N/cm$^2$. The form parameter $c$ on the other hand, models the MN recruitment gradient and is fixed for all muscles. This parameter was estimated empirically using regression methods in [220].

The instantaneous muscle force $M$ is then obtained in a second step by processing $\tilde{M}$ with a second-order, critically damped, lowpass filter that models the graded development of muscle force due to calcium kinetics as described in the following:

$$M = \tau^2 \ddot{\tilde{M}} + 2\tau \dot{\tilde{M}} + \tilde{M}. \tag{4.23}$$

The velocity dependency is illustrated in the force-velocity relationship block in figure 4.4(b). The maximal amount of force a muscle can deliver decreases or increases depending on whether it is concentrically or eccentrically contracting (shortening or elongating, respectively). This is in accordance with the sliding filament theory [222]. The relationship is usually captured in a sigmoidal function that saturates at maximal shortening and elongation. The sigmoid function was estimated from empirical data from cat soleus muscle in which the motor nerve was stimulated at different levels. The resulting function is multiplied with $M$ to calculate the active muscle force as shown in equation (4.24).

The final muscle forces also includes a linear term that models passive force in absence of neural input, and as a simplification is linearly dependent on the difference between the current muscle length $l$, and the muscle resting length $r$. Muscle resting lengths were calculated given the initial arm configuration. While the passive stiffness term $k$ was calculated by the force-length relations shown in [223], and are linearly varying with the cross-sectional area of the muscle [133].

The resulting force

$$F = M\left[f_1 + f_2 \cdot \operatorname{atan}(f_3 + f_4 \dot{l})\right] + k(l - r) \tag{4.24}$$

is then dependent on active and passive forces. The shape of the sigmoidal function that represent the active force is dependent on $f_1$ to $f_4$. Forces $f$ are then produced via the muscle model given an input of threshold muscle lengths $\lambda$, which are generated by means of referent control as discussed in the next section.

### Referent control

The neural control signals that are required for the force generation are designed as described in the threshold control theory (TCT). Within TCT, neural control signals set the muscle threshold lengths $\lambda$ for $\alpha$ motoneuron recruitment. A force is then generated as the $\lambda$ values change compared to the current muscle length and the rate at which the muscle is changing in length. The basic premise in TCT is that by setting the $\lambda$ values for all muscles, we achieve a static configuration for the arm. Therefore, by dynamically changing the $\lambda$ values, a smooth movement from one point to another can be generated. The $\lambda_r$ value is calculated given higher level $R_\theta$ commands that set the equilibrium positions for the joint angles. The $R_{\theta_{1,2}}$ can be also computed using inverse kinematic equations from higher level $R_{x,y}$ that represents the spatial equilibrium positions in Cartesian coordinates. The $R_{x,y}$ command is calculated as discussed in section 4.2.2.

Shifts in the threshold position $R$ are produced due to reciprocal control on motorneurons of opposing muscle groups, and produce angular threshold shifts that activate the muscle groups in the same direction as the common threshold angle. In a complimentary fashion, coactivation commands $C$ shift the actuation thresholds of the muscle groups around a specific joint in the opposite direction of the common threshold angle. Therefore the $C$ command defines an area of activation around the $R$ command and moves together with the $R$ command. The $C$ command relates to the task demands of the movement and can be increased or decreased by the CNS [198]. The $\lambda_r$ and $\lambda_c$ are superimposed to form the final $\lambda$ command as follows:

$$\lambda = \lambda_r + \lambda_c - l. \tag{4.25}$$

Both $R$ and $C$ commands have been studied in the context of fast single-joint movements in [224] and have different contributions to reaching motion. It is the difference between the equilibrium joint position defined by $R_\theta$ and the actual position $Q_\theta$ that generates muscle activation and torques for movement generation. The $C$ command on the other hand, provides stability by increasing the stiffness and the damping during the reaching movements and is the main factor behind the acceleration and deceleration towards the final goal position [198, 201].

In our work we use the end-point (hand) stiffness to calculate the $C$ command as it has been shown that the function of the $C$ command is to increase stability through controlling stiffness [225]. Our motivation here is that end-point stiffness relates to the task of the movement and the agent might regulate this depending on the specific task requirements e.g. accuracy as discussed in [226]. The stiffness at the end-point is defined as the change in force with respect to change in position:

$$K_x = \frac{d\mathbf{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{df_x}{dx} & \frac{df_x}{dy} \\ \frac{df_y}{dx} & \frac{df_y}{dy} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}. \tag{4.26}$$

To transfer the end-point stiffness to joint stiffness (stiffness in the joint angles), we need to transform forces $f$ in the Cartesian coordinate system $x$ to torques $\tau$ defined in angular coordinate system $q$. This is done using the Jacobian that describes the relationship

between $\mathbf{x}$ and $\theta$ as follows:

$$J(\boldsymbol{\theta}) = \frac{d\mathbf{x}}{d\boldsymbol{\theta}}, \tag{4.27}$$

which can be used to transform forces in Cartesian coordinate system into torques in the angular coordinate system accordingly using the principle of virtual work as follows:

$$\boldsymbol{\tau} = J(\boldsymbol{\theta})\mathbf{f}. \tag{4.28}$$

The end-point location as a function of joint angles is defined as follows

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{bmatrix}. \tag{4.29}$$

The derivative of the the end-point position with respect to joint angles become:

$$J(\boldsymbol{\theta}) = \frac{d\mathbf{x}}{d\boldsymbol{\theta}} = \begin{bmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}. \tag{4.30}$$

Joint stiffness can then be defined as:

$$K_j = \frac{d\boldsymbol{\tau}}{d\boldsymbol{\theta}} = \frac{d(J(\boldsymbol{\theta})^T)\mathbf{f})}{d\boldsymbol{\theta}} = \frac{d(J(\boldsymbol{\theta})^T)}{d\boldsymbol{\theta}}\mathbf{f} + J(\boldsymbol{\theta})^T \frac{d\mathbf{f}}{d\boldsymbol{\theta}}, \tag{4.31}$$

which can be approximated given small arm displacements, and expanding $d\boldsymbol{f}/d\boldsymbol{\theta}$:

$$K_j \approx J(\boldsymbol{\theta})^T \frac{d\boldsymbol{f}}{d\boldsymbol{\theta}} \approx J(\boldsymbol{\theta})^T \frac{d\boldsymbol{f}}{d\boldsymbol{x}} \frac{d\boldsymbol{x}}{d\boldsymbol{\theta}} \approx J(\boldsymbol{\theta})^T K_x J(\boldsymbol{\theta}). \tag{4.32}$$

We obtain equilibrium torques by multiplying $K_j$ with the angular displacements in the joints. We transform the equilibrium torques into $\lambda_c$ by applying the following set of simplified equations based on (28-33):

First, the equilibrium torque is transferred into force by dividing by the lever arm calculated in (4.19) using

$$F \ = \frac{\tau}{d}. \tag{4.33}$$

Then, the resulting instantaneous muscle force is calculated by solving for $M$ using (4.24) to obtain

$$M \ = \frac{F - k(l-r)}{f_1 + f_2 \tan(f_3)}. \tag{4.34}$$

The activation of the muscle is calculated by solving for A using (4.20), that results in the simplification:

$$A \ = \frac{\ln(\frac{M}{\rho} + 1)}{c}. \tag{4.35}$$

Finally, $\lambda_c$ is calculated given the current muscle length in

$$\lambda_c \ = l - A. \tag{4.36}$$

## 4.3 Results

In the following we present results obtained from our simulation model. This combines high level goal-setting with the attractor dynamics equilibrium trajectory. The resulting equilibrium trajectory provides the reference commands to the TCT model to actuate the musculoskeletal arm model.

### 4.3.1 Experimental-parameters, setup and human-recordings

The model was implemented on MATLAB/Simulink. In terms of the parameter values that are used for the generation of the results in this section, the geometrical and inertial constants needed for the equation of motion (4.14) are given in Table 4.1. The coefficients required to calculate muscle lengths in (4.18) are given in Table 4.2. Additionally, the coefficients used to calculate moment arms in (4.19) are given in Table 4.3. The values of passive stiffness that are used in (4.24) are given in Table 4.4. The value of the reflex delay $d$ as well as the form parameter $c$ and the damping factor $\mu$ that are used in (4.20) are given in Table 4.5. The magnitude parameters $\rho$ for each muscle that is used in (4.22) are given in Table 4.4. The values of the passive stiffness term $k$ for each muscle that is used in (4.24) are also given in Table 4.4. The values of $f_1 - f_4$ that are used in (4.24) and are given in Table 4.5. The choice of the $\tau$ value in (4.23) is also given in Table 4.5 and leads to a critically damped filter with an asymptotic response to a step function in 90 ms [227], which is similar to empirical data observed in human muscles [228] as discussed in [220]. The attractor dynamics parameters used for the simulations, as discussed in section 4.2.2, are given in Table 4.6. The same parameters are used throughout the different simulations in this section.

**Table 4.1:** Geometrical and inertial constants of the arm model

| Arm segment | Mass $/(kg)$ | Length $/(m)$ | Moment of inertia $/(kg\ m^2)$ |
|---|---|---|---|
| Upper arm | 2.1 | 0.34 | 0.015 |
| Lower arm | 1.65 | 0.46 | 0.022 |

In terms of the setup, we had invited twenty participants to record ten simple, untrained, reaching trajectories towards a target located 30cm in the $+y$ direction. We used an Xsens MVN full-body inertial motion capture (MoCap) suit [189]. The average trajectory (and one unit variance around the mean) of these 200 trajectories, were projected on the transverse plane, and are shown in Fig. 4.6(a). The kinematic variables, namely the displacements in the $x, y$ axis and the tangential velocity, are shown in Fig. 4.6(b-d), respectively. A unit variance around the mean is shown using the shaded regions Fig. 4.6(b-d). The average trajectory is slightly curved towards the goal position, and with a unimodal

**Table 4.2:** Coefficients $\theta_1$ and u1 of equation (4.18) for muscle lengths.

| | Muscle group | | | |
|---|---|---|---|---|
| Parameter | Pectoralis | Deltoids | Biceps | Triceps |
| $cst/(\text{mm})$ | 155.19 | 157.64 | 378.06 | 260.05 |
| $t_6 * 10^{11}$ | - | - | - | 6.1385 |
| $t_5 * 10^{8}$ | - | - | - | $-2.3174$ |
| $t_4 * 10^{7}$ | - | - | 5.2156 | 33.321 |
| $t_3 * 10^{5}$ | - | - | $-3.1498$ | $-22.491$ |
| $t_2 * 10^{3}$ | - | - | $-7.9101$ | 5.2856 |
| $t_1 * 10^{2}$ | - | - | $-25.587$ | 40.644 |
| $\theta_1 * 10^{1}$ | $-8.8663$ | 13.743 | $-5.0981$ | 4.4331 |

**Table 4.3:** Coefficients $c_i, c_0$, and $d_0$ of equation (4.19) for moment arms.

| | Muscle group | | | | | |
|---|---|---|---|---|---|---|
| Parameter | Pectoralis | Deltoids | Biceps short head | Biceps long head | Triceps lateral head | Triceps long head |
| $c_5 * 10^{9}$ | - | - | - | - | $-3.5171$ | - |
| $c_4 * 10^{7}$ | - | - | - | - | 13.277 | - |
| $c_3 * 10^{5}$ | - | - | - | $-2.8883$ | $-19.092$ | - |
| $c_2 * 10^{3}$ | - | - | - | 1.8047 | 12.886 | - |
| $c_1 * 10^{1}$ | - | - | - | 4.5322 | $-3.0284$ | - |
| $c_0$ | - | - | - | 14.660 | $-23.287$ | - |
| $d_0$ | 50.80 | $-78.74$ | 29.21 | - | - | $-25.40$ |

**Table 4.4:** Muscle force-generating parameters

| | Muscle group | | | | | |
|---|---|---|---|---|---|---|
| Parameter | Pectoralis | Deltoids | Biceps short head | Biceps long head | Triceps lateral head | Triceps long head |
| $\rho/(\text{N/cm}^2)$ | 14.9 | 14.9 | 2.1 | 11 | 12.1 | 6.7 |
| $k/(\text{N/m})$ | 258.5 | 258.5 | 36.5 | 190.9 | 209.9 | 116.3 |

**Table 4.5:** Force generation model variables

| $d/(\text{ms})$ | $\mu/(\text{s})$ | $c/(\text{mm}^{-1})$ | $\tau/(\text{ms})$ | $f_1/(\text{s/m})$ | $f_2/(\text{s/m})$ | $f_3/(\text{s/m})$ | $f_4/(\text{s/m})$ |
|---|---|---|---|---|---|---|---|
| 10 | 0.06 | 0.112 | 15 | 0.82 | 0.50 | 0.43 | 58 |

**Table 4.6:** Simulation model variables

| $x_i$/(m) | $y_i$/(m) | $x_f$/(m) | $y_f$/(m) | $x_{o,1}$/(m) | $y_{o,1}$/(m) | $x_{o,2}$/(m) | $y_{o,2}$/(m) | $k$ |
|---|---|---|---|---|---|---|---|---|
| 0.15 | 0 | 0.15 | 0.28 | 0.15 | 0.15 | 0.15 | 0.5 | 100 |
| $m$ | $\sigma_a$ | $G_a$ | $\phi_0$/(°) | $\phi_1$/(°) | $r_h$/(m) | $r_o$/(m) | $m_{arm}$ | $\delta$ |
| 100 | 1.5 | 1.5 | 0 | −45 | 0.025 | 0.05 | 75 | 0.1 |
| $d_0$ | $h_1$ | $v_{max}$ | $\psi_{max}$ | $c$ | $c_{v,obs}$ | $c_{v,tar}$ | $\lambda$ | $\sigma$ |
| 0.05 | 10 | 2 | 0.5 | 1 | 0.001 | 5 | 0.01 | 1 |

velocity profile. The average reaching path is seen to be terminating slightly away from the goal position with a displacement of 10cm in the $-x$ direction. This could be for three main reasons. Firstly, the average reaching path is across all trials. Secondly, the participants were untrained. Thirdly, the Xsens MoCap suit was not calibrated with respect to the environment, as such a calibration is not possible. We hypothesize that the third option is the most probable as the unit variance around the mean is rather broad towards the end of the movement, and towards one side of the target, which gives a strong indication of a systematic error and a system inherent problem. Nevertheless, the main kinematic features of a reaching motion (the smooth trajectory and the unimodal velocity profile), which set the standards for the comparison against the simulated trajectories, are rather clear in Fig. 4.6.

### 4.3.2 Simulation of a reaching motion

The result of simulating a similar forward reaching trajectory is shown in Fig. 4.7. The forward reaching trajectory starts from the initial position $(0.15, 0)$ and terminates at the goal position $(0.15, 0.28)$ as given in Table 4.6 where the rest of the parameters used for the simulation is tabulated. The choice of this specific starting location is calculated such that the initial $\theta_1 = 45°$ and $\theta_2 = 90°$. The reference trajectory resulting from the attractor dynamics equilibrium trajectory is shown as the dotted grey line in Fig. 4.7(a). As observed in Fig. 4.7(a), it is a straight line in the $x - y$ plane. The simulated dynamics of the arm, however, show an end-effector trajectory that is curved, initially moving to the right until it finally converges towards the target at the end. For comparison, the resulting solution from the minimum jerk optimization is also shown in Fig. 4.7(a) in the dashed black line. The minimum jerk solution is a straight line directly towards the goal position. The displacement in the $x$ direction of the reference trajectory is shown using the dotted grey line in Fig. 4.7(b), while the end-effector displacement in the $x$ direction is shown using the black solid curve. The minimum jerk solution is also shown in Fig. 4.7(b) using the dashed black line. Compared to the resulting simulation of the attractor dynamics, the minimum jerk trajectory tracks the desired $x$ position very well with no deviations. Similarly, the displacement in the $y$ direction of the reference trajectory (dotted grey), end-effector trajectory (solid black) and the resulting minimum jerk trajectory (dashed black) are shown in Fig. 4.7(c). Finally the tangential velocity of the end-effector position is shown

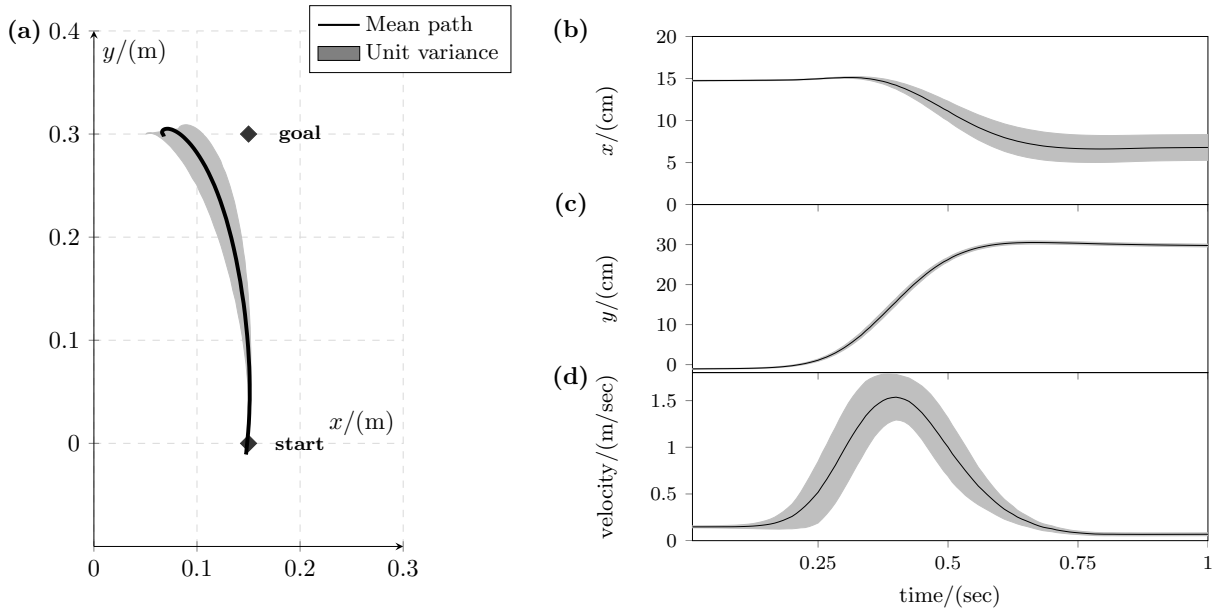


**Figure 4.6:** (a) The average reaching path of $20$ participants performing $10$ trials each, with a unit variance around the mean (shaded grey). (b) The mean trajectory in the $x$ direction (solid black), with a unit variance around the mean (shaded grey). (c) The mean trajectory in the $y$ direction (solid black), with a unit variance around the mean (shaded grey). (d) The mean tangential velocity (solid black), with a unit variance around the means (shaded grey).

in Fig. 4.7(d) as the solid black line. The tangential velocity resulting from the minimum jerk solution is shown using a dashed black line. The results show a comparable end-effector trajectory to that in Fig. 4.6 and a similar unimodal tangential velocity. The reference trajectory, however, as described in literature, terminates around the third of the complete movement, and close to the peak of the tangential velocity. Furthermore, the resulting end-effector trajectory (solid-black) differs from the reference trajectory (dotted-grey) especially in the second half of the motion. This is because the reference trajectory shift completed half-way during the motion and it is the muscle dynamics (inertial and reflex delays, etc.) and reflex properties that are then responsible for the deceleration of the movement and achieving the goal position; which could lead to the simulated musculoskeletal end-effector's trajectory deviation from the reference trajectory as discussed in [198]. The difference between the resulting end-effector trajectory (solid-black) and the minimum jerk trajectory (dashed-black) seems to be a function of maximum velocity. The resulting end-effector trajectory starts later and ends earlier, thus having a higher maximum velocity. The oscillations that are observed in the tangential velocity profile in Fig. 4.7(d) around 400ms can be attributed to the musculoskeletal system's hook motion towards goal position in Fig. 4.7(a). In this instance the final position of the reference trajectory aims at eliminating the movement error and guides the musculoskeletal system towards the goal position by changing direction of movement and slightly increasing the tangential velocity.

The above results show the ability of the attractor dynamics systems to generate neural equilibrium trajectories that can be realized by the musculoskeletal system using the EPH concepts. The resulting neural trajectory provides the $R_{x,y}$ commands that are used within the TCT. The neural attractor dynamics trajectory planner does not depend on complex calculations, and produces simple straight trajectories when no obstacles are observed. The calculations depend on the dynamically evolving behavioral variables, and are a function of the desired goal location, obstacles in the environment and the current arm configuration. The dynamic parameters of the arm model, as well as the muscles lead to a trajectory that is slightly curved as observed usually in a human reaching trajectory.

It is worth noting that the maximum tangential velocity achieved via the simulated movement is higher than that observed through the minimum jerk solution for a similar motion duration of 650ms. The tangential velocity can be increased by setting the movement duration to be less than e.g. 650ms.

### 4.3.3 Simulation of a reaching motion with obstacle avoidance

In presence of obstacles, however, the simple system of the neural attractor dynamics equilibrium trajectory generator adapts, and produces the reference trajectories accordingly. We simulate the case of obstacle avoidance in Fig. 4.8. The forward reaching trajectory starts from the initial position $(0.15, 0)$ and terminates at the goal position $(0.15, 0.28)$. The obstacle is located at position $(0.15, 0.15)$ as given in Table 4.6 where the rest of the parameters used for the simulation is tabulated. The reference trajectory resulting from the attractor dynamics equilibrium trajectory is shown as the dotted grey line in Fig. 4.8(a). As observed in Fig. 4.8(a), it is a curved line in the $x - y$ plane. The solution of the minimum jerk optimization through the via point $(0.3, 0.15)$ is also shown in Fig. 4.8(a) as the dashed black line. The simulated dynamics of the arm, however, as

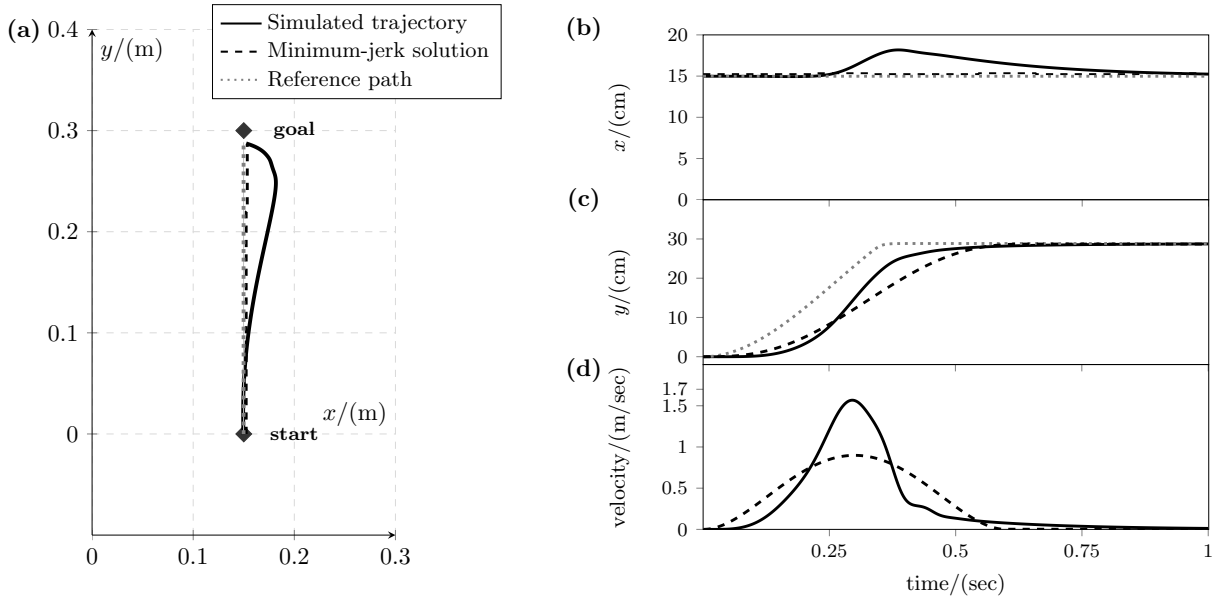


**Figure 4.7:** (a) The reference path (dotted gray) and simulated musculoskeletal end-effector trajectory (solid black) as well as the minimum-jerk solution (dashed black) for the reaching motion in the transverse plane. The reaching motion consists of a forward movement from the black diamond labeled *start* to the black diamond labeled *goal* in the $+y$ direction. (b) The reference trajectory (dotted gray) and simulated musculoskeletal end-effector trajectory (solid black) as well as the minimum-jerk solution (dashed black) in $x$ displacement vs. time. (c) The reference trajectory (dotted gray) and simulated musculoskeletal end-effector trajectory (solid black) as well as the minimum-jerk solution (dashed black) in $y$ displacement vs. time. (d) The tangential velocity of the simulated end-effector (solid black) as well as the tangential velocity resulting from the minimum-jerk solution (dashed black) vs. time.

illustrated in Fig. 4.8(a) using the solid black line, show an end-effector trajectory that can be categorized by three movement portions. The first is a slow forward movement and recovery away from the obstacle, this is highlighted by the marker (1) Fig. 4.8(a). The second portion of the movement is marked by the marker (2) and is characterized by a slightly curved line to avoid the obstacle and finally a curved trajectory towards the goal position that is marked by the marker (3). The displacement in the $x$ direction of the reference trajectory is shown using the dotted grey line in Fig. 4.8(b), while the end-effector displacement in the $x$ direction is shown using the black solid curve. The displacement in the $x$ direction of the solution of the minimum jerk trajectory is shown using the black dashed curve. Similarly, the displacement in the $y$ direction of both the reference (dotted grey) and end-effector (solid black) as well as the resulting minimum jerk (dashed black) trajectories are shown in Fig. 4.8(c). Finally the tangential velocity of the end-effector position is shown in Fig. 4.8(d). The tangential velocity is characterized by three peaks that are consistent with the three movement portions. The tangential velocity resulting from the minimum jerk solution is shown using a dashed black line in Fig. 4.8(d), and is characterized by two distinct peaks. The first portion relates to avoiding the obstacle and

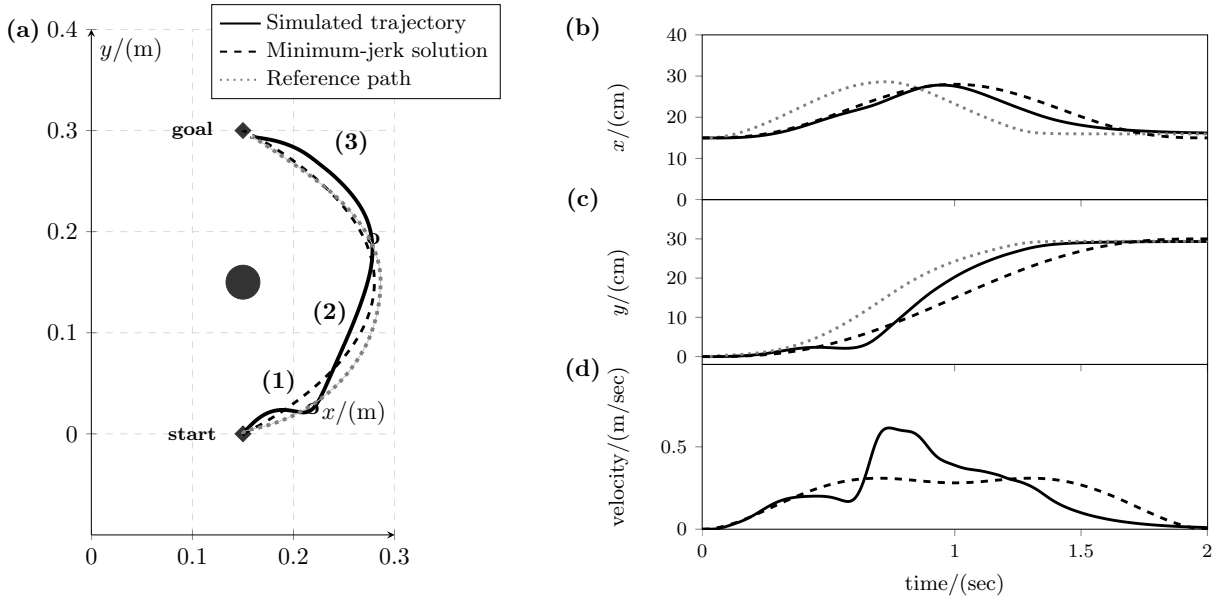


**Figure 4.8:** (a) The reference trajectory (dotted grey) and simulated musculoskeletal end-effector trajectory (solid black) as well as the minimum-jerk solution (dashed black) in the transverse plane.in the transverse plane. The movement simulated is a movement towards the $(+y)$ direction to avoid the obstacle (black circle). Additionally the three distinct portions of the movement are highlighted)(1-3). (b) The reference trajectory (dotted grey), simulated musculoskeletal end-effector trajectory (solid black) as well as the minimum-jerk solution (dashed black) in $x$ displacement vs. time. (c) The reference trajectory (dotted gray), simulated musculoskeletal end-effector trajectory (solid black) as well as the minimum-jerk solution (dashed black) in $y$ displacement vs. time. (d) The tangential velocity of the simulated end-effector (solid black) as well as the tangential velocity resulting from the minimum-jerk solution (dashed black) vs. time

then moving towards the goal in the second portion.

As shown in Fig. 4.8, the equilibrium trajectory dynamically avoids obstacles in relation to the whole arm. We highlight this in the next set of results shown in Fig. 4.9(a-d). In our current implementation, we chose $k$ which is the number of control points along the lower arm to be 100. The initial heading angle $\phi_0$ defines the evolution of the neural trajectory. Neurally, the heading angle is described as the angle of the vector pointing towards the goal position from the current end-effector position [175]. Setting the heading angle $\phi_0 = 0$ would lead to an incorrect overall trajectory since the lower arm collides with the obstacle (grey star), as shown in the simulation result in Fig. 4.9(a). This can be adjusted however, by setting $\phi_1 = -45°$. The simulation, shown in Fig. 4.9(b), shows the arm avoiding the obstacle by correctly passing under it. This suggests however, that the initial value of the behavioral variable $\phi$ should take into account the locations of obstacles and take values that not only avoid these obstacles but avoid collision with the forearm.

In order to solve the problem of initial value setting, similar to what is shown in Fig. 4.9(a), as well as Fig. 4.9(c), we devise a repeller term over the entire arm as dis-

cussed in section 4.2.2. The scenario in Fig. 4.9(c), represents a challenging case as the obstacle is located slightly below its previous position, and with a radius that would allow for the end-effector to safely pass through using a mathematically and logically accurate initial angle of $\phi_0 = 0$. While the simulation results in Fig. 4.9(c) were obtained without the repelling function $f_{arm}$ and results in an incorrect trajectory that leads to a collision with the forearm, the simulation results in Fig. 4.9(d) were obtained with the addition of the $f_{arm}$ term. The results in Fig. 4.9(d) show a correct trajectory that avoids the obstacle by traveling under the obstacle with an initial heading angle of $\phi_0 = 0$.

The kinematic and attractor dynamic terms related to the simulation result shown in Fig. 4.9(d) are shown in Fig 4.10(a-j). The $x$ and $y$ displacements are shown in Fig. 4.10(a) and Fig. 4.10(b) respectively. The displacement show smooth trajectories towards the location of the goal. The heading angle as shown in Fig. 4.10(c) initially takes negative values to avoid the obstacle both at the end effector and forearm locations and raises in values as it avoids the obstacle to head towards the goal. The additive terms of heading angle and obstacle avoidance and arm awareness as well as the stochastic noise term are given in Fig 4.10(a-g) respectively. The $f_{arm}$ term is initially dominant driving the autonomous system to large negative values such as to guarantee the forearms movement below the obstacle. The $f_{tar}$ then increases to drive the system back to achieve the goal position. The stochastic term provides the means to escape any spurious local maxima stable points that might occur that are undesired. The velocity generated given in Fig. 4.10(a-g) depends on the values of the spatial velocity terms $v_{space}$ and the velocity attractor term $v_{atr}$. As can be seen in Fig. 4.10, the equilibrium trajectory end-point velocity is a unimodal velocity profile that resembles that desired end-effector tangential velocity of the arm.

## 4.4 Discussion

The threshold control theory is a special case of dynamics systems theory that is consistent with cognitive theory. Specifically it is consistent with embodied enacted cognition where cognition is said to arise for action and perception. The dynamic shifts of the equilibrium point within threshold control theory is a function of the agent as well as the objects as modeled within this work. While we present a model for the generation of reaching motions, several models in the literature have proposed the use of threshold control theory for standing, gripping and complete body movements [132, 229, 230].

The threshold control theory additionally uses the ideas of attractors indirectly by using incremental equilibrium positions along the desired trajectory. Similar models in literature [136, 199–203] are complex in nature and are unable to adapt dynamically to the environment. Compared to related work, we presented a novel simple attractor dynamics systems, that takes the body configuration and environment state into account, to generate the reference trajectories.

Through the use of the neural trajectory planner, we have shown that dynamic plans towards the goal can be obtained such as to account for obstacles and the kinematics of the arm. This is discussed in the referent control block. This would substantially limit the amount of training data required by other methods in the same class, shifting the analysis to intentional variables such as the heading angle, to dictate the behavior given its initial

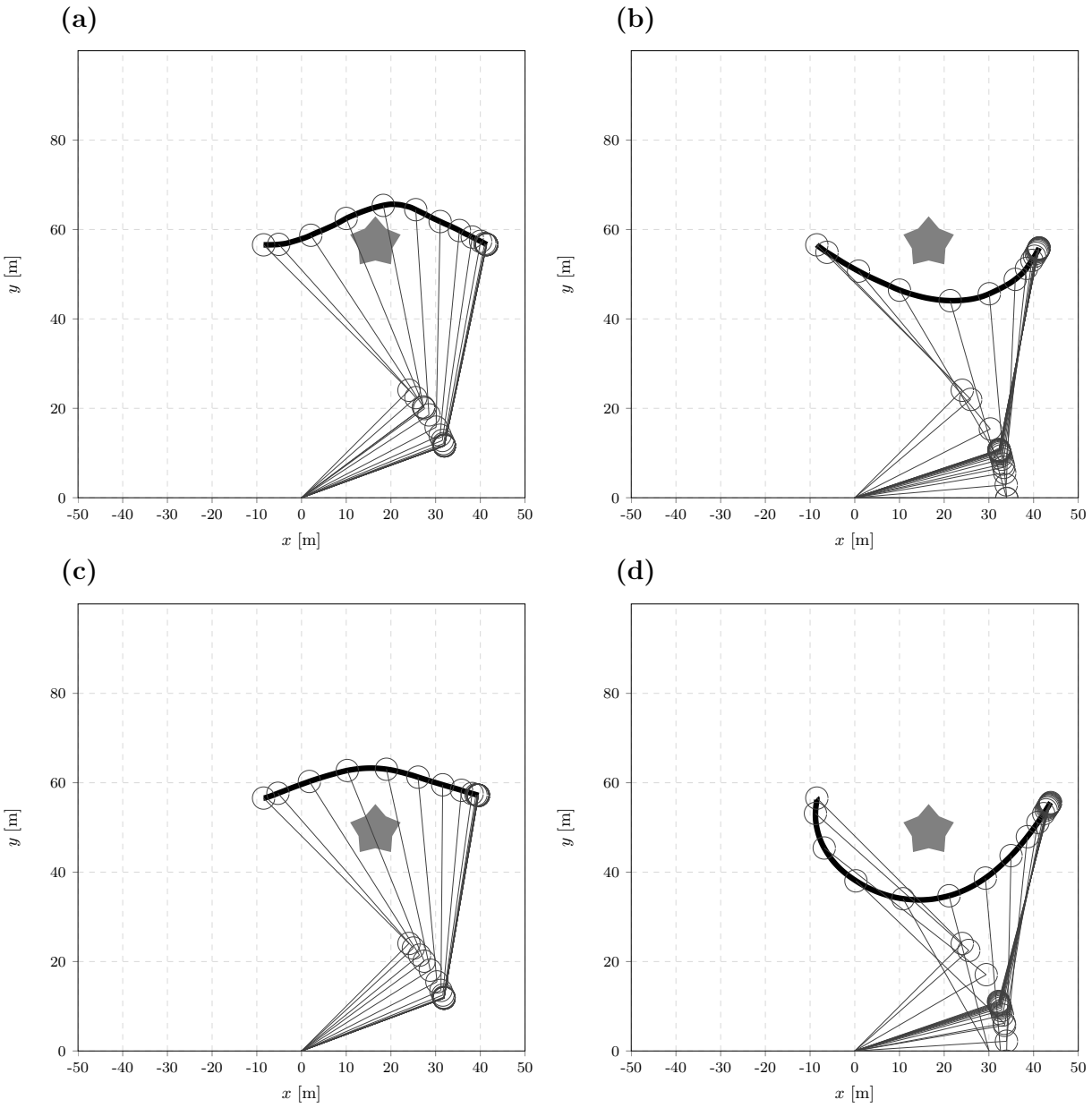


**Figure 4.9:** (a) Simulated neural trajectory result for an obstacle in between the starting and goal locations. The initial value of $\phi$ was equal to zero. (b) Simulated neural trajectory result for an obstacle in between the starting and goal locations where the initial value of $\phi$ was changed to $-45$. (c) Simulated neural trajectory result for an obstacle slightly lower than the middle location between the starting and goal points. The initial value of $\phi$ was equal to zero. The simulation was run without the arm awareness function $f_{arm}$. (d) Simulated neural trajectory result for an obstacle slightly lower than the middle location between the starting and goal points. The initial value of $\phi$ was equal to zero. The simulation was run with the additive arm awareness function $f_{arm}$.

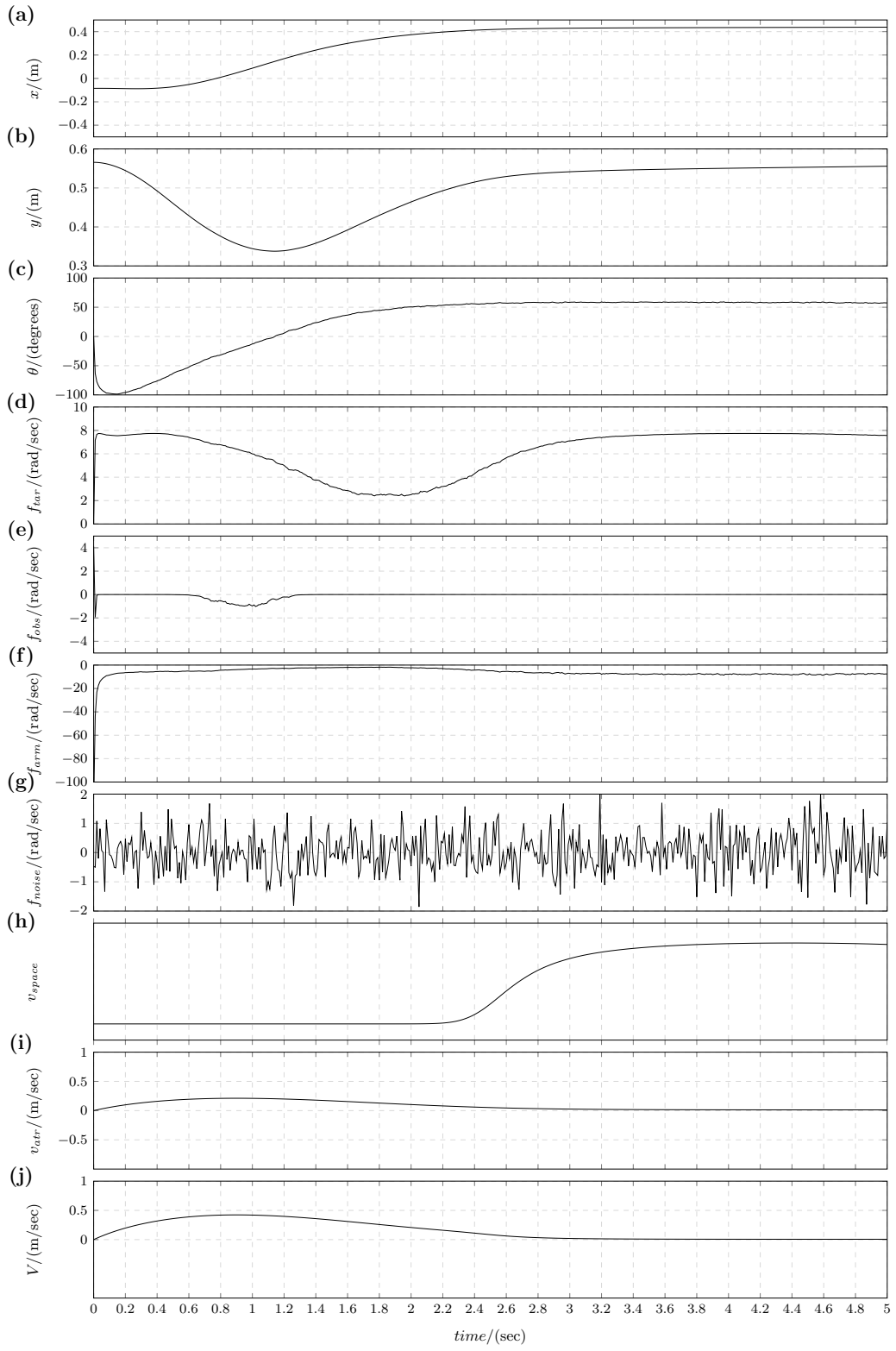


**Figure 4.10:** Temporal development of the kinematic and attractor dynamic terms that resulted in the simulations shown in Fig. 4.9. (a) The $x$ displacement of the neural reference trajectory. (b) The $y$ displacement of the neural reference trajectory. (c) The heading angle of the autonomous robot at the hand location. (d) The goal acquisition term. (e) The obstacle avoidance term. (f) The arm awareness term. (g) The stochastic noise term. (h) The spatial term of of velocity. (i) The velocity attractor term. (j) The velocity of the autonomous robot at the hand location.

value. Finally the last building block discussed was the musculoskeletal arm model that was used to validate the output of the referent control and simulate reaching motions with and without the presence of an obstacle.

The results show the systems ability to successfully complete a forward reaching motion with and without the existence of an obstacle. Furthermore, the results were compared to the output of minimum-jerk optimization. Minimum-jerk was proposed as an explication to how the brain plans and controls movements in an optimal manner in order to solve the redundancy problem on a kinematic level in 2D settings. The cost function within the minimum-jerk optimization problem minimizes the time integral of the square of jerk (rate of change of acceleration). Minimum-jerk was able to predict the regularities in hand paths of multi-joint arm movements. Explicitly, minimum-jerk was able to predict smooth movements, with unimodal, symmetrical velocity profiles. Compared to the minimum-jerk prediction of the obstacle-free movement, the simulated movement using referent control showed similar characteristics. Explicitly, the simulated movement showed smooth movements with unimodal, mostly symmetrical velocity profiles. It is worth noting that the simulated movement seems to have a higher maximum tangential velocity compared to the solution of the minimum-jerk optimization problem, however the movement duration, which is an input parameter to the minimum-jerk optimization problem, was not similar. The discrepancy in movement duration could explain the variation between maximum tangential velocities.

Furthermore, minimum-jerk optimization requires definition of via-points, as well as the condition of passing through those via-points (velocity and acceleration through the via-point) while the attractor dynamics does not. Additionally, the work presented here easily extends to 3D cases where the minimum jerk model holds for 2D scenarios. In the case of the obstacle avoidance problem there was a slight difference between the predictions within the minimum-jerk model and the simulation of the referent control signals. Explicitly, the simulation predicted three segments of motion for obstacle avoidance, while the minimum-jerk optimization predicted two symmetrical motion segments. The minimum-jerk optimization takes the parameter of the via-point position as an input to the model, while the attractor dynamic solution produces the "via-points" dynamically. However, both trajectories in $x$ and $y$ directions were smooth. It is worth noting that the simulated trajectory initially lagging the minimum-jerk model as if to give the arm the best possible chance to avoid the obstacle until second  0.6. Starting then, the simulated movement was leading compared to the prediction of the minimum-jerk model. As the repeller force generating from the attractor dynamic repulsion from the obstacle are overcome by the attraction forces of the goal.

## 4.5 Conclusions

We presented in this chapter, for the very first time, a novel approach to dynamic reaching motion generation through the integration of attractor dynamics with referent control theory. Similar work in literature that model reaching trajectories through referent control suffer mainly from two issues. Firstly, the models are usually complex and hard to explain. Secondly, they are unable to adapt dynamically to the environment.

In contrast to the above, the work presented here offers a novel, biologically-inspired referent control formulation using dynamic systems theory to model a reaching motion that is both simple and adapts to the environment. Results showed the dynamic generation of the reference trajectory that was validated using a musculoskeletal arm model for simulation both in open space and in an obstacle avoidance scenario.

Although not investigated throughly in this work, the dynamics approach for behavior generation benefits from the DNF for high-level decision making (e.g. identifying goals and obstacles). Indeed, future work would focus on investigating dynamic changes in goal setting, and studying the effect of those changes on the simulated paths generated by the musculoskeletal systems. Future work could additionally investigate human-recorded obstacle avoidance to extract common via-points for comparison. Furthermore, future work could investigate varying the number of via-points within the minimum-jerk optimization for comparison purposes as well.

# 5 Plan understanding

So far, we have addressed the problem of proximal intention understanding from two directions. The first step was a top-down understanding of the observed kinematics and the agent's interaction with the environment, as discussed in chapter 3. The second step was a bottom-up internal simulation of the predicted movement. The bottom-up approach was discussed in chapter 4, where we focused on the simulation of reaching motions. Given the understanding of the proximal intention, which gives an indication of the atomic actions an agent is performing, the distal intention of the agent can also be understood. Distal intention understanding refers to understanding the plan of the agent given a series of observed actions. In robotics, the field of plan, action and intention recognition systems (PAIRS) has been active in investigating distal intention understanding and developing intelligent systems to that end.

Research in the field of PAIRS follows a trend in linking actions to plans in order to understand human intentions. Action recognition aims to classify patterns from low-level sensory information, such as cameras or inertial sensors, into semantic labels of human actions. Plan and intention recognition operates on a higher-level and aims to infer action plans and intents by appending a meaning to the temporal relationship between the recognized primitive actions.

It is thought that humans perform plan recognition through utilizing an inferential system. This inferential framework readily accommodates our ability as perceivers to deal with the complex link between actions and intentions [40]. In this system, an intentional plan is determined not only given recognized actions but also from external information. This information includes cues in the immediate context e.g. the setting, location, presence of specific people and equipment. They should also include prior knowledge about the observed agent as well as the script within which the agent's motions are embedded. These additional characteristics allow the inferential system to reduce the search space to a smaller set out of many possibilities. The bottom line here is that the inferential system allows intentions to be recognized given the bigger picture that the action stream is embedded in. Keeping the above points into consideration a set of inputs to intention and plan recognition systems can be defined. First input would be a set of conceivable intentions, secondly a set of plans achieving each intentions given, namely, a plan library that is biased to context, agent and environment.

Given the discussion so far, the interpretation of the sequence (using a plan library and an inferential system) of actions houses vital information that explains how people identify the intentions of others. The important question becomes, how do people choose their sequence of actions? In the field of philosophy, it is argued that in ToM mental states are inferred through the application of the "rationality principle" [20]. This is also in compliance with the "teleological stance" through the psychological principle of rational action [28]. The principle of rational action assumes that (rational) actions emerge to

fulfill an intention (goal-state) by the most efficient way. Furthermore, the goal states are realized by choosing the most rational action (most efficient, least risky, fastest, etc..) currently available given the *constraints of the situation* (current state of the environment). Thus, an explanation of the intention behind observed behavior is said to be acceptable and well-formed if, and only if, the observed action (intention) can be thought to be rational in accordance with a goal state (desire) given the situational constraints (belief). Therefore, for a successful recognition of intentions, both the observer and the actor should share a common understanding of the environment and the what it affords in terms of action possibilities. The situational constraints in our work are modeled using the concepts within affordances.

We give in this chapter an outlook on how the task of plan understanding can be linked with the presented work of action understanding. We introduce a cognitive, neurally inspired model of plan understanding for the purpose of human intention recognition. The architecture models a robotic ToM which follows descriptions from simulation theory and respects the dynamic nature of intentions. The model follows the embedded, situated cognitive stance in which the tight coupling of the agent, the environment and the brain is respected. We build upon our previous work of (primitive) action understanding that models the attention shift of an observer to objects of interest, and uses contextual information of affordances to classify the trajectory information of the observed agent. The model proposed here respects the dynamic nature of intentions and the inputs in the environment as well as the embedded situated stance of cognition and is modeled using system dynamic theory of dynamic neural fields.

We present in this work, for the first time, a DFT-based plan understanding system that extends the different components discussed in chapter 3 to the higher-level abstraction of plan understanding. Explicitly, compared to the state-of-the-art in DFT-based plan understanding presented in [231], we present an affordances-based approach within dynamic field theory that generates plans dynamically given new observations of actions. Furthermore, we present a novel plan comparison approach that extends the TARS system presented in section 3.5. The components of the plan recognition model are presented in section 5.2. The implementation results of plan prediction are presented in section 5.3. Finally we give a discussion in section 5.4.

## 5.1 Related Work

Early work on plan recognition is found in the work of Robert Wilensky [232, 233] and James Allen [234], in which systems were developed for the tasks of narrative plan understanding. The focused shifted from small scale plan understanding into large scale plan understanding for the tasks of speech recognition in the work of VERBMOBIL [235, 236] and TRAINS [237, 238], that was later expanded into TRIPS [239].

Similar to the discussion of state-of-the-art in intention recognition in the introduction of this thesis, logic approaches and probabilistic approaches are dominant in the task of plan understanding. Appelt and Pollak in [240] show an example of formal logic approaches in which weighted abduction was used to extract plans from a set of rules based on observation. On the other hand Konolige and Pollack in [241] utilized a probabilistic approach to reason

about the most likely plan within a specific problem domain. Charniak and Goldman in [61, 62] introduced Bayesian inference for plan understanding, leading to the use of dynamic belief networks as a way to apply Bayesian inference for the task of plan understanding in the work of Albrecht et al. in [53, 242]. Probabilistic plan recognition was also used in the work of Goldman et al. in [243] where partially-ordered plans, multiple interleaved plans and effects of context were taken into consideration.

In addition to probabilistic models, optimization techniques were also used in modeling plan recognition systems. In the work of Sukthankar and Sycara [244] a plan recognition system was developed based on a cost minimization approach for *Military Operations in Urban Terrain* (MOUT). The system aimed at recognizing a plan of an agent that is tasked to achieve a specific goal within a military operation. The proposed approach utilizes an environmental simulator to generate the final results given the solution of the minimum cost optimization.

Recently, Baker and Tenenbaum [245] tackled the problem of cognitive plan recognition by modeling ToM. This follows the same motivation presented in this chapter of modeling human cognition through ToM. Due to the fact, that the mental states of others are unknown, this becomes a difficult task. Different mental states can result in the same action/ behavior, whereas similar mental states can result in different behavior [245]. Since mental states are hidden and there exists no explicit one-to-one mapping between actions and behavior they proposed a Bayesian approach to tackle those issues. The authors propose a Bayesian Theory of Mind framework in which knowledge as well as ontology is integrated at different abstraction levels. Explicitly, they follow a common path within PAIRS in which a generative model of decision making is utilized for inductive reasoning in a second step given observations. The generative model is modeled as a partially observable Markov decision problem (POMDP) in which beliefs about the environment and their uncertainty are probabilistic distributions and the desires/preferences constitute the reward functions. The generative algorithms presented by Baker and Tenenbaum decide on actions that have the highest expected reward. Compared to the work presented by Baker and Tenenbaum, the work presented here predicts the next action step based on a learnt transition between actions. In terms of plan understanding, the work presented by Baker and Tenenbaum performs Bayesian inference on the inverted value function to reason about the agents beliefs and reward function given the agent's observed actions. This work, on the other hand, utilizes direct comparison between expected and observed actions for comparison.

Doshi et. al. in [246, 247] similarly model human condition through ToM using POMDPs for inductive reasoning. The work presented by Doshi et. al. uses ideas of interactive POMDPs such that the observing agent's decision making processes is taken into account as well as the observed agent's plan in a nested manner. This is helpful in adversarial settings as the nested POMDPs allows to model the observed agents's ability to plan his actions given other agents reasoning about that plan. Compared to the work presented by Doshi et. al., we do not model adversarial settings.

The advantages and disadvantages of probabilistic and logical approaches to plan understanding are similar to what has already been introduced in section 1.2. Optimization techniques on the other can be very powerful in finding an optimal solution to a cost function that describes the decision making process of intentional action plans. They suffer

however from several limitations such as finding an appropriate formulation of the cost function as well as the high computational cost of evaluating an optimal solution in a simulation scenario. The work presented in this chapter presents a decision making approach that utilizes simple comparisons to recognize action plans given the affordances of the objects in the immediate environment. Furthermore, it does so by utilizing the same systems used in the task of action understanding in a dynamic manner.

## 5.2 DFT-based plan understanding

The first step to achieve plan understanding is to maintain the observed actions into a time-series that represents the plan observed so far. This time-series can be expanded to include possible future actions to facilitate the task of plan understanding. The generation of future actions to extend the observed action plan could allow for faster comparison (against learnt plans in the plan library) and aid in producing more reliable results. A system that allows for the prediction of possible future actions is required to dynamically adapt to changes in this action plan time-series as new actions are observed. We present DFT to model the situational constraints and affordances in section 5.2.1. Furthermore, we use DFT to perform the task of dynamic plan generation given information of affordances in section 5.2.2. Finally, the task of comparing plans for the task of plan understanding is presented in section 5.2.3.

### 5.2.1 Affordance-based Approach

Affordances are at the heart of plan understanding similar to its function in the action understanding systems discussed in chapter 2. Furthermore, we model the situational constraints of the environment through the use of affordances. Explicitly within this work, we hypothesize that an action plan is represented by a sequential manipulation of affordances in the immediate environments. The same logic-based approach discussed in section 3.6 is utilized here.

The basic premise in this approach is as follows. As an agent produces an intention, it acts on the environment through a series of manipulation actions that ultimately changes the set of available affordances accordingly. The connectivity field models the interaction of the agent with the available affordances and the dynamic effect on the perceived affordances as a consequence of this interaction. The sequential changes in the affordances would allow for reasoning about the set of future actions as well as the required actions (affordance manipulations) required to achieve a certain goal.

Generally for action understanding the difference between the current affordance state of the environment and the final goal allows for the inference of the set of actions required to achieve that specific goal. Actions are then methods required to manipulate, use or change the set of available affordances. As an example, if an agent is sitting on the couch in the living room and develops an intention to drink water, then the affordance of the glass in the kitchen needs to be changed from *approachable* to *graspable* to e.g. *drinkable* and for that the agent needs to perform a locomotion action to locate itself in a grasping reach, and fill the glass with water so it is drinkable such that the agent can perform the

final action of drinking from the glass. In order to utilize this affordance-based approach, the plan understanding system would require to encode knowledge regarding the dynamic nature of affordances when an agent manipulates the different objects. The connectivity field discussed in section 3.6 allows to learn and reason regarding these changes.

Having the ability to model affordances and their dynamic interaction within the connectivity field would allow us to reason about the possible next actions in an observed agent's plan. This would allow us to generate dynamic plans given the series of observed actions. We present this dynamic generation of action plans in the next section (section 5.2.2).

### 5.2.2 Generating dynamic plans

Within this approach, the past, current and future prediction actions are maintained in a consistent neural field that represents the *understood* plan. It is important to note that action sequences discussed in this work are time-independent. That is, action sequences are stored in action time-steps and durations of actions are not of importance as they do not add valuable information for the task of plan understanding.

In the generation of the plan, past actions are addressed first. Past actions are saved within a memory field directly given the input of the action understanding systems. The same applies to current actions observed in the current action time-step. The integration of future actions within the plan generation is not as straight forward as integrating past and current actions. Future actions are generated sequentially and randomly given information from the *connectivity field* and the CARS as introduced in section 3.4.

Given the output of the CARS, the contextual information of which object the action is directed towards is obtained. Once the object of interest is identified, the corresponding *connectivity field* is selected. The set of possible future actions are read out from the *connectivity field* given the current action. A selection kernel is then used to select an action from the set of possible future actions given the activation strength of the different actions in the connectivity field. The selected action then goes through the same process to predict the second possible future plan and so on. This process is repeated given the desired number of look-forward steps desired. This is in line with Lashley's description of sequence learning as discussed in [248].

It is worth noting that a selection is performed at each action step across all possible *connectivity fields* as a function of the possible manipulable objects by way of a selection kernel. Alternatively, it is possible to perform a selection step initially across the different possible objects before performing a selection in the selected object's connectivity field. The process of performing the prediction steps within a *connectivity field* is illustrated in Fig. 5.1.

In the following, we walk through the example given in Fig. 5.1 when predicting the next series of actions given the currently observed actions. The currently observed action, observed at the current action time step $t$, in this example is $a_1$. This is shown in Fig. 5.1(a) on the *action time step* axis. The possible future actions can be read out from the *connectivity field*. In this case actions $a_2$ and $a_3$ are read out. The read out 1D slice of the *connectivity field* is used as an input to the decision field as shown in Fig. 5.1(b). A decision in the decision field is made between $a_2$ and $a_3$ and $a_2$ is selected as illustrated in the output of the selection field in Fig. 5.1(b). The action $a_2$ is used to perform the next

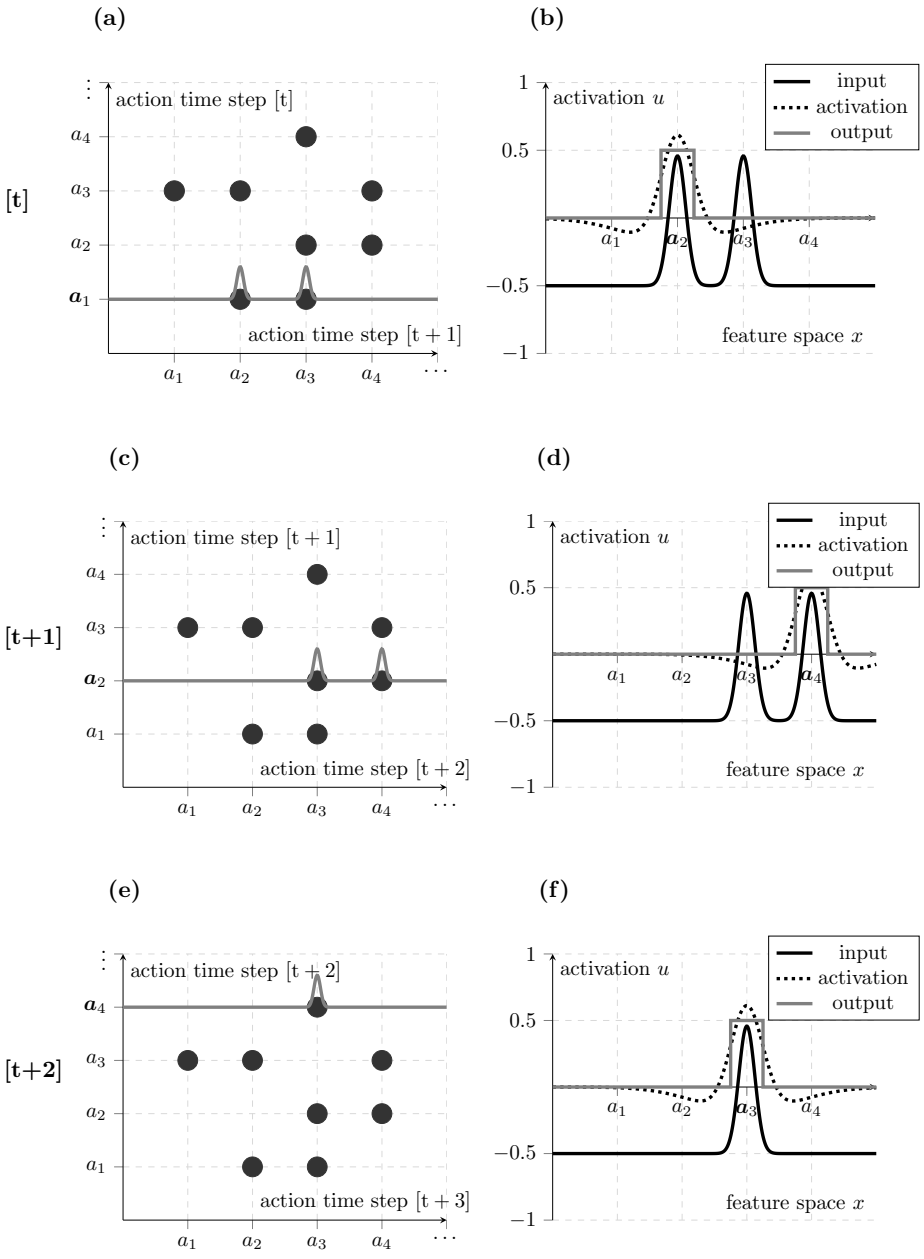


**Figure 5.1:** Plan generation example. a) to d) shows the prediction of an action sequence using a *connectivity field*. A detailed description of this figure is provided in section 5.2.2.

prediction step in Fig. 5.1(c) to read out the possible next actions, which in this case are actions $a_3$ and $a_4$. The same procedure is repeated to perform a selection between $a_3$ and $a_4$ which in turn influences the possible action at *action time step* $[t+3]$ and so on, until we reach our fixed number of look-ahead steps.

Our dynamic plan generation method that reuses the same central affordance logic system modeled within the *connectivity field*. The *connectivity field* couples with a selection field that produces an output of the next possible actions in the next time steps. The previous, current and future actions generate a dynamic plan that is saved in memory for the purpose of comparison and further inference as discussed in the next section.

### 5.2.3 Plan comparison

In this section, we present the plan comparison approach given the dynamically generated plan discussed in the previous section. The basic premise in this work is that a comparison has to be performed between the generated plan and a set of *learnt* plan templates in order to confirm the understood plan. This is similar to our approach for action understanding in which the TARS compares on a trajectory-level the goal-directed movement anticipated by CARS. The plan comparison method presented here reuses the TARS discussed previously in 3.5.

#### Plan comparison based on path trajectory

The *plan comparison approach based on the path trajectory comparison* discussed in this section is based on the TARS approach discussed in section 3.5. The basic premise here is that the generated plan is treated similar to that of the stimulus of observed movements. This approach is motivated by descriptions of plans being a sequence of time-dependent actions as discussed in [248]. It is important to note that a preshape in this case does not encode a movement but rather a sequence of actions. This generated plan is compared against a set of learnt templates of plans similar to how the stimulus of observed movements is compared against a set of preshapes of learnt movements. The learnt template uses a set of action plans that might be very different than one another in terms of the order of actions. Therefore, branching (of parallel actions) within the template generation step is allowed. This ensures that multiple actions are allowed to be represented at a same action time step within one intentional plan. This generalized template would allow different variations of the same distal intention to be taken into account. Within the plan trajectory method, the dynamically generated plan is projected into a memory field and a moving wave is allowed to propagate through it for the purpose of comparison similar to action comparison discussed in the TARS in section 3.5.

The comparison approach, illustrated in Fig. 5.2, is a combination of three distinct logical parts (layers). The first layer is a memory field as described in section 3.2, and is composed of the plan up to and not including the action at action step $t$. The second layer is composed of the current action at action step $t$, and is the output of the AUA. Finally, the third layer is composed of the future actions beyond action step $t$ as discussed in section 5.2.2. Overall the combination of the three parts compose a two dimensional memory trace which constitutes a set of atomic actions against time. The atomic actions

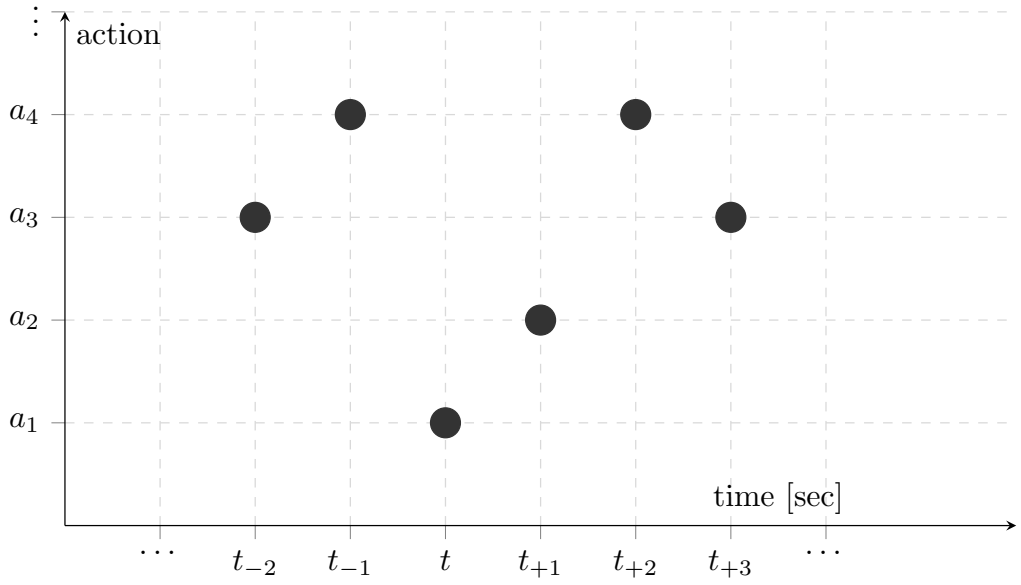


**Figure 5.2:** Plan trajectory comparison: this figure visualizes the generation of a plan in trajectory form. The trajectory is composed of the past, current and predicted actions. Further description of this figure can be found in section 5.2.3.

are imagined as neurons that contribute to the field with their own tuning curve in a manner consistent with DPA descriptions as discussed in section 3.2.2. It is worth noting that unlike the continuous case when angles or distances are used as features in motion trajectories within DFT, the actions do not interact with each other. As such the tuning curves have their own optimal response values that separate the activation peaks accordingly.

The first layer of the plan trajectory is generated by using a traveling peak, as discussed in section 3.2.5, that activates regions in the path trajectory field corresponding to the actions detected. That is, as actions are recognized in the past, traveling waves activate the corresponding neurons dynamically.

As for currently recognized actions, they are integrated through a second layer at an action step $t$. The reasoning here is this layer has a different purpose functionally. It encodes information that is highly dynamic. That is the current action is not fully understood until it is completed. Therefore, this layer is implemented by a DNF that allows dynamic changes.

The third and final layer is integrated in a similar fashion to the currently recognized action in the second layer. The only difference is that the future actions depend on each other in a temporal fashion, that is each DNF decision field requires the decision in the previous DNF to perform the action. Finally all layers are integrated in a final layer by adding up activations from the aforementioned three layers. The activations are added into the final layer when a traveling wave scans through each of the three layers sequentially. For example, to integrate future actions, the traveling wave should be positioned at the action time step $[t + 1]$. The integration of the three layers generates the plan trajectory that is

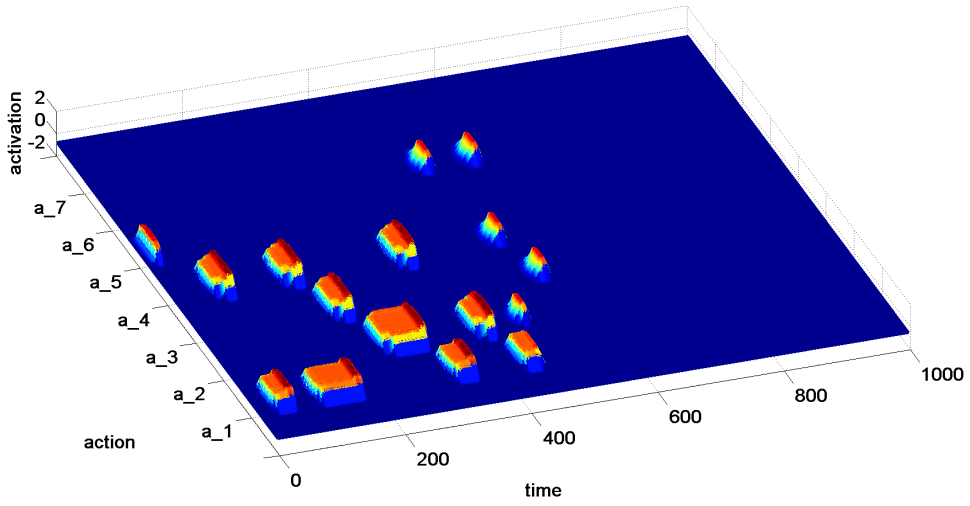


**Figure 5.3:** Dynamic generated plan. Illustrated is the activation or a DNF for different actions $a_1$ to $a_7$ with respect to time.

capable of being compared against a plan template in a manner similar to that performed in TARS, discussed in section 3.5. This approach allows for a dynamic continuous comparison that takes plan changes into account at different (current and future) layers that could occur at every action time step.

## 5.3 Results

In this section, we show the results of the dynamic plan generation. The trajectory-based comparison implements the same TARS, however without the use of the Sequence of interest module. As such, for further discussion and results regarding trajectory-based comparison, we refer the reader to section 3.5 and section 3.8.2 respectively. As for the dynamic plan generation, the same concepts of using a traveling wave for prediction discussed in section 3.2.5 is utilized here. The traveling waves are set to be equidistant with medium speed. We implemented 5 look forward steps to predict the next five actions. The future actions are predicted given the *connectivity field* and a DNF with a selection kernel. The results of a dynamic generated plan is shown in Fig. 5.3.

The *connectivity field* itself is learned. The learning is implemented as follows: as the action understanding systems provide the action sequences observed the *connectivity field* is populated sequentially. As this process occurs over different examples, the transition between different actions is preserved within the *connectivity field* and would thus allow for representations of plans accordingly.

## 5.4 Discussion

We have presented in this chapter a DFT module for dynamic plan generation that memorizes the sequence of understood actions performed by an agent. Furthermore, the dynamic plan generation module allows the dynamic integration of the currently observed action, as well as reasoning about future actions given a learnt *connectivity field*. The dynamics within DFT allows the overall adaption to changing output from the action understanding systems that are dynamic in their nature as well. We have also presented the plan comparison method based on path trajectory that reuses the same concepts from TARS to reason about the observed plans.

Compared to related work that model ToM (e.g. Baker and Tenenbaum's work in [245] as well as Doshi et. al.'s work in [246, 247]), the presented approach allows for dynamical plan generation and comparison. The overall system adapts accordingly to observed actions as the action understanding modules update their beliefs. Furthermore, the generalized template plans themselves are dynamic and allow for different action paths over time to explain the same intentional plan.

The current system however does not generate a plan library of plan templates. A generative model for simulating plan libraries can be considered given the respective *connectivity field*. In that sense, multiple plans can be generated based on the states of the environment and the *connectivity field* of the respective intentional plan. Learning *connectivity fields* could be then an advantage as it allows for the possibility of generating different plans online and comparing them dynamically against the observed action plan in a second step.

Furthermore, the current system does not take feedback signals into account. Explicitly, feedback signals linking the action understanding system to the plan understanding could be integrated. The feedback signal from the dynamically generated plans to the *connectivity field* within AUA could give hints to the AUA which affordances to prune when loading the next preshape (or generating the trajectories of the movement) . It would do so by removing the set of affordances that are not compliant with the predicted plan.

Other challenges should also be taken into consideration within the plan understanding systems. Explicitly, interruptions as well as unrecognized actions should be taken into account. Furthermore, we assume in this work that there is no ordering between the discrete actions along the feature space that defines the different DNFs (e.g. connectivity field). It could be the case that actions could be ordered along the feature space based on their similarity and their underlying connections, which in turn could allow for them to interact meaningfully if their tuning curves were to overlap.

## 5.5 Conclusions

We presented in this chapter a distal intention recognition system that reuses the same modules that were used for action understanding for the purposes of plan understanding. This is inline with descriptions of MNS that promotes an emulation as well as an imitation functionality to the MNS. We presented a module that is capable of dynamically generating plans given the *connectivity field*, and explained how the TARS can be reused for the task of comparing the observed plan against a learnt preshape encoding the intentional action plan.

The presented work is novel in DFT and in contrast to other work in literature, reused the same mechanisms used in the action understanding step to perform plan understanding in a meaningful cognitive framework. Future work should focus on the interaction between the plan understanding systems and the AUA through feedback/feedforward signals. The interaction between the two systems should also be evaluated in terms of e.g. time to decision making and recognition accuracy etc. Further future work could investigate the topics of interruptions and unrecognized actions.

# 6 Conclusions

The task of human intention understanding is a central problem that should be addressed adequately if robotic systems are to be integrated socially into our everyday life. Intention understanding is biologically motivated by findings in MNS and philosophical descriptions within ToM. However, similar to most cognitive tasks that biological systems effortlessly demonstrate, replication on artificial systems is not without complications. The basic premise of this work is to transfer the functions of the MNS at different levels of abstraction into a robotic, dynamic cognitive framework via dynamic systems theory. Dynamic systems theory adequately provides the framework to describe behavior in a top-down and bottom-up approach and across different abstractions in a biologically motivated and cognitively valid manner. In this work, dynamic field theory within dynamic system theory is used to tackle the different challenges of intention recognition in three different chapters. The solutions that were provided were chapter-specific and were discussed accordingly. In the following, we summarize the main contributions of this thesis and provide an outlook for different future research directions.

## 6.1 Summary of contributions

In chapter 2 we presented our motivation and design for the action understanding architecture that tackles the challenges of proximal intention understanding in a top-down, bottom-up approach. The primary focus of this work was aimed at identifying the signals used by humans when detecting/simulating intentional actions of other observed agents, modeling it within DFT and designing a decision-making system that makes use of these signals in a meaningful manner. The AUA presented in this work is a deterministic model that reacts to the input and produces decisions dynamically. This is in contrast to probabilistic models proposed in the literature. Specifically, we can classify our AUA approach as a dynamic, single-layered exemplar-based sequential method, that depends on contextual information when choosing the example (template). Exemplar-based sequential methods have an advantage of requiring less training data to perform recognition when compared to probabilistic methods.

In chapter 3, two explicit systems were introduced to address the top-down approach of action understanding. These are the CARS and TARS which make use of the affordance logic system that models the immediate environment and the available affordances of the objects within this environment. Overall, the AU architecture in this work presents a novel predictive system within DFT which models attention-shifts and integrates with a novel trajectory parsing system in a second step. The trajectory parsing system takes into account the spatial as well as the temporal variations that are usually problematic when understanding actions on a trajectory level. Particular attention is given on how objects and the environment are integrated into the overall architecture and on how they

can drive action understanding. The overall systems were evaluated given a dataset of participants performing high-level intentional actions. Results show that the integration of both the attention-shift system and the trajectory comparison system yields good action understanding results compared to each of them alone.

In chapter 4, we presented a bottom-up approach to aid in the task of action understanding. Explicitly, we introduced the dynamic simulation of action trajectories in the case of reaching movements. We described how the cognitive decision-making system (such as attention shift model) could select objects using DFT which in turn sets attractors that drive the behavior of a reaching motion. We linked the environment to the end-effectors and described the dynamic nature of reaching motion generation as motivated by threshold control theory. The reference command was modeled using attractor dynamics. We validated the resulting referent and coactivation commands on a two-dimensional musculoskeletal arm model and compared the output against examples from participants performing the same movement. Furthermore, we described how attractor dynamics could aid in the task of obstacle avoidance within threshold control theory.

In chapter 5, the plan understanding system was introduced. We explained how the different modules that were presented in chapter 3 are reused in the task of plan understanding. Furthermore, we explained how the affordance logic system that is central to the task of action understanding is also at the heart of plan understanding. Through simulation, we showed how the observations of a few actions would allow the system to reason about the next immediate action and project the possible future actions. The work in this chapter highlights the use of the different systems for the solution of both the action and plan understanding task. This is in-line with descriptions of MNS where the underlying function of MNS could explain imitation (action-level), emulation (goal-level) or intention.

## 6.2 Future directions

The work presented in this thesis motivates directions for future work.

**Biological motivated features:** We have discussed within this work two sets of biologically-inspired features for action understanding; the body joint extension and the projected relative angle features. Additionally, we discussed how they could be neurally represented within concepts of DPA and DFT. Future work could focus on investigating different combinations of features and discuss how they are optimally mixed under different contexts. Furthermore, the context under which specific features operate best could be investigated.

**Learning affordances:** Learning how to perceive affordances is an integral part of the social capabilities of humans. We have discussed at length within this thesis the importance of affordances in predicting and planning intentions. The concept of robotic learning of object affordances has not been discussed within DFT despite being an important concept within the cognitive capabilities of humans. Future work could focus on expanding the simple concept of affordance-based logic fields we present in this thesis such that it encodes the relationships between actions, objects, and effects in an active manner.

**Dynamic affordances:** The concepts of nested affordances and sequential affordances were introduced by Gaver in [187]. Informally, they describe the dynamic nature of affordances in that manipulating a set of affordances changes either the affordance of the manipulated item or that of the other objects in the immediate environment. Future work could focus on the formalization of the concepts of nested and sequential affordances dynamically within DFT.

**Full-body referent control using attractor dynamics:** The concepts of locomotion and manipulation have been investigated under threshold control theory. The attractor dynamics approach described in this thesis could be extended to represent a broader range of goal-directed movements. Future work could focus on defining attractors around different behaviors that could be achieved dynamically using full-body movements.

**Comparison metrics for plan understanding:** We have discussed in this thesis the concept of comparison within DFT. The comparison was performed within the context of a movement which is usually described using continuous variables. Plans, on the other hand, are described by discrete atomic variables (actions) that are unordered. Therefore, the concept of direct comparison is unrealistic. Future work could focus on defining dynamic comparison metrics within DFT between a set of plans to achieve intention understanding.

**Proactive robotic interaction:** Once a robotic system understands the intention of the human agent, that information should be acted upon proactively. Furthermore, the selected proactive robotic action, along with the human's reaction should be taken into account in a feedback loop to reinforce or debunk the understood intention. Future work could focus on the formulation of behavioral feedback loops within DFT and model their effect on the decision-making process within the different intention recognition systems.

## 6.3 Concluding remarks

We have presented in this thesis novel systems to achieve intention understanding within dynamic systems theory, and DFT, that are inspired by the function and mechanisms of the mirror neuron system and ToM. The systems are capable of extracting essential information from movement kinematics and parse objects in the environment into goals and obstacles as well as reason about the agent's action plan. Furthermore, dynamic neural trajectories are planned towards the possible goal in the immediate environment. The presented work advances the state-of-the-art by combining the key concepts of dynamic behavior generation world of Schöner et. al. with the intentional dynamics of Kelso et al. such that the interaction is understood using the affordance concept of Gibson et al. and movement is directed towards them using concepts pioneered by Feldman et al. for cognitive action understanding within the cognitive framework of DFT.

# Bibliography

[1] Eliot R Smith and Gün R Semin. Socially situated cognition: Cognition in its social context. *Advances in experimental social psychology*, 36:53–117, 2004.

[2] Gregor Schöner. Dynamical systems approaches to cognition. *Cambridge handbook of computational cognitive modeling*, pages 101–126, 2008.

[3] Bertram F Malle and Joshua Knobe. The folk concept of intentionality. *Journal of Experimental Social Psychology*, 33(2):101–121, 1997.

[4] G.E.M. Anscombe. *Intention*. Harvard University Press, 2000.

[5] Darren Newtson. Attribution and the unit of perception of ongoing behavior. *Journal of Personality and Social Psychology*, 28(1):28, 1973.

[6] Darren Newtson and Gretchen Engquist. The perceptual organization of ongoing behavior. *Journal of Experimental Social Psychology*, 12(5):436 – 450, 1976.

[7] Dare Baldwin and Jodie Baird. Action analysis: A gateway to intentional inference. In Philippe Rochat, editor, *Early social cognition*, page 215–240. NJ: Lawrence Erlbaum Associates, 1999.

[8] Jeffrey M Zacks and Barbara Tversky. Event structure in perception and conception. *Psychological bulletin*, 127(1):3, 2001.

[9] Robert M Gordon. 'radical'simulationism. In Eds. P. Carruthers & P. K. Smith, editor, *Theories of theories of mind*. Cambridge: Cambridge University Press., 1996.

[10] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(04):515–526, 1978.

[11] Maria Alessandra Umilta, Evelyne Kohler, Vittorio Gallese, Leonardo Fogassi, Luciano Fadiga, Christian Keysers, and Giacomo Rizzolatti. I know what you are doing: a neurophysiological study. *Neuron*, 31(1):155–165, 2001.

[12] Marco Iacoboni, Istvan Molnar-Szakacs, Vittorio Gallese, Giovanni Buccino, John C Mazziotta, and Giacomo Rizzolatti. Grasping the intentions of others with one's own mirror neuron system. *PLoS biology*, 3(3):e79, 2005.

[13] Erhan Oztop, Mitsuo Kawato, and Michael A Arbib. Mirror neurons: functions, mechanisms and models. *Neuroscience letters*, 540:43–55, 2013.

[14] Franz Brentano. *Psychology From an Empirical Standpoint*. Humanities Press, 1874.

[15] Bertram F Malle, Louis J Moses, and Dare A Baldwin. *Intentions and intentionality: Foundations of social cognition*. MIT press, 2001.

[16] Joshua Knobe and Bertram Malle. The folk concept of intentionality. *Journal of Experimental Social Psychology*, 33:101–121, 1997.

[17] Philip R Cohen, C Raymond Perrault, and James F Allen. Beyond question answering. *Strategies for natural language processing*, pages 245–274, 1981.

[18] Charles F. Schmidt, NS Sridharan, and John L. Goodson. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence*, 11(1):45–83, 1978.

[19] Michael Bratman. *Intention, Plans and Practical Reason*. Harvard University Press, 1987.

[20] Daniel Clement Dennett. *The intentional stance*. MIT press, 1989.

[21] Darren Newtson. Attribution and the unit of perception of ongoing behavior. *Journal of Personality & Social Psychology*, 28(1):28–38, October 1973.

[22] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1:515–526, 12 1978.

[23] Robert M. Gordon. 'radical' simulationism. In Peter Carruthers and Peter K. Smith, editors, *Theories of Theories of Mind*. Cambridge University Press, 1996.

[24] Robert M Gordon. Folk psychology as simulation. *Mind & Language*, 1(2):158–171, 1986.

[25] Alvin I Goldman. Interpretation psychologized*. *Mind & Language*, 4(3):161–185, 1989.

[26] Gergely Csibra and György Gergely. The teleological origins of mentalistic action explanations: A developmental hypothesis. *Developmental Science*, 1(2):255–259, 1998.

[27] György Gergely and Gergely Csibra. Teleological reasoning in infancy: The infant's naive theory of rational action: A reply to premack and premack. *Cognition*, 63(2):227–233, 1997.

[28] György Gergely and Gergely Csibra. Teleological reasoning in infancy: The naıve theory of rational action. *Trends in cognitive sciences*, 7(7):287–292, 2003.

[29] Giacomo Rizzolatti, Luciano Fadiga, Vittorio Gallese, and Leonardo Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive brain research*, 3(2):131–141, 1996.

[30] Laurie Carr, Marco Iacoboni, Marie-Charlotte Dubeau, John C Mazziotta, and Gian Luigi Lenzi. Neural mechanisms of empathy in humans: a relay from neural systems for imitation to limbic areas. *Proceedings of the national Academy of Sciences*, 100(9):5497–5502, 2003.

[31] R Christopher Miall. Connecting mirror neurons and forward models. *Neuroreport*, 14(17):2135–2137, 2003.

[32] Giacomo Rizzolatti and Michael A Arbib. Language within our grasp. *Trends in neurosciences*, 21(5):188–194, 1998.

[33] Vittorio Gallese, Christian Keysers, and Giacomo Rizzolatti. A unifying view of the basis of social cognition. *Trends in cognitive sciences*, 8(9):396–403, 2004.

[34] Giacomo Rizzolatti, Leonardo Fogassi, and Vittorio Gallese. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Reviews Neuroscience*, 2(9):661–670, 2001.

[35] Michael A Arbib, James B Bonaiuto, Stéphane Jacobs, and Scott H Frey. Tool use and the distalization of the end-effector. *Psychological Research PRPF*, 73(4):441–462, 2009.

[36] Stéphane Jacobs, Claudia Danielmeier, and Scott H Frey. Human anterior intraparietal and ventral premotor cortices support representations of grasping with the hand or a novel tool. *Journal of Cognitive Neuroscience*, 22(11):2594–2608, 2010.

[37] Frédérique De Vignemont and P Haggard. Action observation and execution: What is shared? *Social neuroscience*, 3(3-4):421–433, 2008.

[38] M. A. Umiltà, L. Escola, I. Intskirveli, F. Grammont, M. Rochat, F. Caruana, A. Jezzini, V. Gallese, and G. Rizzolatti. When pliers become fingers in the monkey motor system. *Proceedings of the National Academy of Sciences*, 105(6):2209–2213, 2008.

[39] Leonardo Fogassi, Pier Francesco Ferrari, Benno Gesierich, Stefano Rozzi, Fabian Chersi, and Giacomo Rizzolatti. Parietal lobe: from action organization to intention understanding. *Science*, 308(5722):662–667, 2005.

[40] John R. Searle. *Intentionality: An Essay in the Philosophy of Mind*. Cambridge University Press, 1983.

[41] Myles Brand. *Intending and Acting*. MIT Press, 1984.

[42] Alfred R. Mele. *Springs of Action : Understanding Intentional Behavior*. Oxford University Press, USA, 1992.

[43] Clint Heinze. Modelling intention recognition for intelligent agent systems. Technical report, Defence Science And Technology Organisation Salisbury (Australia) Systems Sciences Lab, 2004.

[44] Luís Moniz Pereira et al. State-of-the-art of intention recognition and its use in decision making. *AI Communications*, 26(2):237–246, 2013.

[45] Christopher W Geib and Robert P Goldman. Plan recognition in intrusion detection systems. In *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings*, volume 1, pages 46–55. IEEE, 2001.

[46] Henry A. Kautz and James F. Allen. Generalized plan recognition. In *AAAI*, volume 86, 1986.

[47] Liliana Ardissono, Guido Boella, and Leonardo Lesmo. Recognition of problem-solving plans in dialogue interpretation. In *Proc. 5th Int. Conf. on User Modeling*, pages 195–197. Citeseer, 1996.

[48] Neal Lesh. Adaptive goal recognition. In *IJCAI*, pages 1208–1214. Citeseer, 1997.

[49] Sandra Carberry. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1-2):31–48, 2001.

[50] Fariba Sadri. Intention recognition in agents for ambient intelligence: Logic-based approaches. In Tibor Bosse, editor, *Agents and Ambient Intelligence: Achievements and Challenges in the Intersection of Agent Technology and Ambient Intelligence*, volume 12, pages 197–236. IOS Press, 2012.

[51] Fariba Sadri. Logic-based approaches to intention recognition. In *Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives*, pages 346–375. IGI Global, 2011.

[52] Marcelo Gabriel Armentano and Analía Amandi. Plan recognition for interface agents. *Artificial Intelligence Review*, 28(2):131–162, Aug 2007.

[53] DavidW. Albrecht, Ingrid Zukerman, and AnE. Nicholson. Bayesian models for keyhole plan recognition in an adventure game. *User Modeling and User-Adapted Interaction*, 8(1-2):5–47, 1998.

[54] Martha Pollack. Some requirements for a model of the plan-inference process in conversation. *Communication Failure in Dialogue, North Holland*, pages 245–256, 1987.

[55] Charles S. Peirce. *The Collected Papers of Charles Sanders Peirce, Vol. VII: Science and Philosophy*. Harvard University Press, Cambridge, 1958. From the Commens Bibliography | `http://www.commens.org/bibliography/anthology_volume/peirce-charles-s-1958-collected-papers-charles-sanders-peirce-vol-vii`.

[56] Eugene Charniak and Drew McDermott. Introduction to ai. *Reading (Mass.): Addison*, 1985.

[57] Neal Lesh and Oren Etzioni. A sound and fast goal recognizer. In *IJCAI*, volume 95, pages 1704–1710, 1995.

[58] Charles Rich, Candace L Sidner, and Neal Lesh. Collagen: Applying collaborative discourse theory to human-computer interaction. *AI magazine*, 22(4):15, 2001.

[59] Dimitra Papadimitriou, Georgia Koutrika, John Mylopoulos, and Yannis Velegrakis. The goal behind the action: Toward goal-aware systems and applications. *ACM Transactions on Database Systems (TODS)*, 41(4):23, 2016.

[60] Sandra Carberry. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1-2):31–48, 2001.

[61] Eugene Charniak and Robert Goldman. A probabilistic model of plan recognition. In *Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 1*, AAAI'91, pages 160–165. AAAI Press, 1991.

[62] Eugene Charniak and Robert P. Goldman. A bayesian model of plan recognition. *Artificial Intelligence*, 64(1):53–79, 1993.

[63] David V Pynadath and Michael P Wellman. Accounting for context in plan recognition, with application to traffic monitoring. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 472–481. Morgan Kaufmann Publishers Inc., 1995.

[64] Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 256–265. Morgan Kaufmann Publishers Inc., 1998.

[65] Marcus Huber and Richard Simpson. Plan recognition to aid the visually impaired. *User Modeling 2003*, pages 146–146, 2003.

[66] Vicente Fernandez, Carlos Balaguer, Dolores Blanco, and Miguel Angel Salichs. Active human-mobile manipulator cooperation through intention recognition. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2668–2673. IEEE, 2001.

[67] Kwun Han, Manuela Veloso, et al. Automated robot behavior recognition. In *ROBOTICS RESEARCH-INTERNATIONAL SYMPOSIUM-*, volume 9, pages 249–256, 2000.

[68] Wentao Yu, Redwan Alqasemi, Rajiv Dubey, and Norali Pernalete. Telemanipulation assistance based on motion intention recognition. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1121–1126. IEEE, 2005.

[69] Richard Kelley, Alireza Tavakkoli, Christopher King, Monica Nicolescu, Mircea Nicolescu, and George Bebis. Understanding human intentions via hidden markov models in autonomous mobile robots. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 367–374. ACM, 2008.

[70] Yoram Singer and Manfred K Warmuth. Training algorithms for hidden markov models using entropy based distance functions. In *Advances in Neural Information Processing Systems*, pages 641–647, 1997.

[71] Vincent Duchaine and Clement M Gosselin. General model of human-robot cooperation using a novel velocity based variable impedance control. In *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*, pages 446–451. IEEE, 2007.

[72] Karim A Tahboub. Intelligent human-machine interaction based on dynamic bayesian networks probabilistic intention recognition. *Journal of Intelligent & Robotic Systems*, 45(1):31–52, 2006.

[73] Clay B Holroyd and Michael GH Coles. The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. *Psychological review*, 109(4):679, 2002.

[74] Masahiko Haruno, D Wolpert, and Mitsuo Kawato. Mosaic model for sensorimotor learning and control. *Neural computation*, 13(10):2201–2220, 2001.

[75] Daniel M. Wolpert, Kenji Doya, and Mitsuo Kawato. A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society of London*, 358:593–602, February 2003.

[76] John Demiris and Gillian Hayes. Imitation as a dual-route process featuring predictive and learning components; 4 biologically plausible computational model. *Imitation in animals and artifacts*, page 327, 2002.

[77] Yiannis Demiris and Bassam Khadhouri. Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and autonomous systems*, 54(5):361–369, 2006.

[78] Yiannis Demiris and Gavin Simmons. Perceiving the unusual: Temporal properties of hierarchical motor representations for action perception. *Neural Networks*, 19(3):272–284, 2006.

[79] Jun Tani. Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. *Neural Networks*, 16(1):11–23, 2003.

[80] Jun Tani and Masato Ito. Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 33(4):481–488, 2003.

[81] Jun Tani, Masato Ito, and Yuuya Sugita. Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using rnnpb. *Neural Networks*, 17(8):1273–1289, 2004.

[82] James Bonaiuto, Edina Rosta, and Michael Arbib. Extending the mirror neuron system model, i. *Biological cybernetics*, 96(1):9–38, 2007.

[83] Guoqiang Peter Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4):451–462, 2000.

[84] Estela Bicho, Luís Louro, and Wolfram Erlhagen. Integrating verbal and nonverbal communication in a dynamic neural field architecture for human-robot interaction. *Frontiers in Neurorobotics*, 4(5):1–13, May 2010.

[85] David Lobato, Yulia Sandamirskaya, Mathis Richter, and Gregor Schöner. Parsing of action sequences: A neural dynamics approach. *Paladyn, Journal of Behavioral Robotics*, 6(1), 2015.

[86] J. R. Flanagan and R. S. Johansson. Action plans used in action observation. *Nature*, 424:769–770, 2003.

[87] James J. Gibson. *The Senses Considered as Perceptual Systems.* Boston: Houghton Mifflin., 1966.

[88] James J. Gibson. *The Ecological Approach to Visual Perception.* Boston: Houghton Mifflin, 1979.

[89] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973.

[90] Randolph Blake and Maggie Shiffrar. Perception of human motion. *Annu. Rev. Psychol.*, 58:47–73, 2007.

[91] Jeffrey M Zacks, Shawn Kumar, Richard A Abrams, and Ritesh Mehta. Using movement and intentions to understand human activity. *Cognition*, 112(2):201–216, 2009.

[92] Anatol G Feldman and Mindy F Levin. The equilibrium-point hypothesis–past, present and future. In *Progress in Motor Control*, pages 699–726. Springer, 2009.

[93] Arthur M Glenberg. What memory is for: Creating meaning in the service of action. *Behavioral and brain sciences*, 20(01):41–50, 1997.

[94] Lawrence W Barsalou. Language comprehension: Archival memory or preparation for situated action? *Discourse Processes*, 28:61–80, 1999.

[95] Wolfgang Prinz. *A common coding approach to perception and action.* Springer, 1990.

[96] Gün R Semin and Eliot R Smith. Socially situated cognition in perspective. *Social Cognition*, 31(2):125–146, 2013.

[97] Matthew Johnson and Yiannis Demiris. Perceptual perspective taking and action recognition. *International Journal of Advanced Robotic Systems*, 2(4):32, 2005.

[98] Matthew Johnson and Yiannis Demiris. Visuo-cognitive perspective taking for action recognition. AISB, 2007.

[99] Y Derimis and G Hayes. Imitations as a dual-route process featuring predictive and learning components: a biologically plausible computational model. *Imitation in animals and artifacts*, pages 327–361, 2002.

[100] Yiannis Demiris* and Matthew Johnson. Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning. *Connection Science*, 15(4):231–243, 2003.

[101] Erhan Oztop, Daniel Wolpert, and Mitsuo Kawato. Mental state inference using visual control parameters. *Cognitive Brain Research*, 22(2):129–151, 2005.

[102] Giorgio Metta, Giulio Sandini, Lorenzo Natale, Laila Craighero, and Luciano Fadiga. Understanding mirror neurons: a bio-robotic approach. *Interaction studies*, 7(2):197–232, 2006.

[103] Laila Craighero, Giorgio Metta, Giulio Sandini, and Luciano Fadiga. The mirror-neurons system: data and models. *Progress in brain research*, 164:39–59, 2007.

[104] D. Feil-Seifer and M.J. Mataric. Defining socially assistive robotics. In *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pages 465–468, June 2005.

[105] Akin Avci, Stephan Bosch, Mihai Marin-Perianu, Raluca Marin-Perianu, and Paul Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *Architecture of Computing Systems (ARCS), 2010 23rd International Conference on*, pages 1–10, Feb 2010.

[106] Gün Semin and John Cacioppo. Grounding social cognition: Synchronization, coordination, and co-regulation. In Gün R Semin and Eliot R Smith, editors, *Embodied grounding: Social, cognitive, affective, and neuroscientific approaches*. Cambridge University Press, 2008.

[107] Robert E Shaw, Endre Kadar, Mikyoung Sim, and Daniel W Repperger. The intentional spring: A strategy for modeling systems that learn to perform intentional acts. *Journal of Motor Behavior*, 24(1):3–28, 1992.

[108] J.J. Gibson, E.S. Reed, and R. Jones. *Reasons for Realism: Selected Essays of James J. Gibson*. Resources for Ecological Psychology. L. Erlbaum, 1982.

[109] Anthony Chemero. An outline of a theory of affordances. *Ecological psychology*, 15(2):181–195, 2003.

[110] Keith S Jones. What is an affordance? *Ecological psychology*, 15(2):107–114, 2003.

[111] Thomas A Stoffregen. Affordances as properties of the animal-environment system. *Ecological Psychology*, 15(2):115–134, 2003.

[112] Claire F Michaels. Affordances: Four points of debate. *Ecological Psychology*, 15(2):135–148, 2003.

[113] Harry Heft. Affordances, dynamic experience, and the challenge of reification. *Ecological Psychology*, 15(2):149–180, 2003.

[114] William H Warren. Perceiving affordances: visual guidance of stair climbing. *Journal of Experimental Psychology: Human Perception and Performance*, 10(5):683, 1984.

[115] Joanna McGrenere and Wayne Ho. Affordances: Clarifying and evolving a concept. In *Graphics Interface*, volume 2000, pages 179–186, 2000.

[116] K. Koffka. *Principles of Gestalt Psychology*. Number v. 7 in Cognitive psychology]. Routledge, 1999.

[117] Lynn T Kozlowski and James E Cutting. Recognizing the sex of a walker from a dynamic point-light display. *Perception & Psychophysics*, 21(6):575–580, 1977.

[118] Winand H Dittrich, Tom Troscianko, Stephen EG Lea, and Dawn Morgan. Perception of emotion from dynamic point-light displays represented in dance. *Perception-London*, 25(6):727–738, 1996.

[119] Fritz Heider and Marianne Simmel. An experimental study of apparent behavior. *The American Journal of Psychology*, pages 243–259, 1944.

[120] James E Cutting, Cassandra Moore, and Roger Morrison. Masking the motions of human gait. *Perception & Psychophysics*, 44(4):339–347, 1988.

[121] Hanako Ikeda, Randolph Blake, and Katsumi Watanabe. Eccentric perception of biological motion is unscalably poor. *Vision research*, 45(15):1935–1943, 2005.

[122] Dare Baldwin, Jeffery Loucks, and Mark Sabbagh. Pragmatics of human action. In Thomas F. Shipley and Jeffrey M. Zacks, editors, *Understanding events: From perception to action. Oxford series in visual cognition.*, pages 96–129. Oxford University Press, 2008.

[123] Chris D Frith and Uta Frith. Interacting minds–a biological basis. *Science*, 286(5445):1692–1695, 1999.

[124] Sarah-Jayne Blakemore and Jean Decety. From the perception of action to the understanding of intention. *Nature Reviews Neuroscience*, 2(8):561–567, 2001.

[125] Martin A Giese and Tomaso Poggio. Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, 4(3):179–192, 2003.

[126] David E Meyer and Sylvan Kornblum. *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience*, volume 14. Mit Press, 1993.

[127] Zoe Kourtzi and Nancy Kanwisher. Activation in human mt/mst by static images with implied motion. *Journal of cognitive neuroscience*, 12(1):48–55, 2000.

[128] Sheba Heptulla Chatterjee, Jennifer J Freyd, and Maggie Shiffrar. Configural processing in the perception of apparent biological motion. *Journal of Experimental Psychology: Human Perception and Performance*, 22(4):916, 1996.

[129] JA Beintema, K Georg, and M Lappe. Perception of biological motion from limited-lifetime stimuli. *Perception & Psychophysics*, 68(4):613–624, 2006.

[130] Joachim Lange and Markus Lappe. A model of biological motion perception from configural form cues. *The Journal of Neuroscience*, 26(11):2894–2906, 2006.

[131] Anatol G. Feldman Jean-Francois Pilon, Sophie J. De Serres. Threshold position control of arm movement with anticipatory increase in grip force. *Experimental Brain Research*, 181:49–67, 2007.

[132] Michael Günther and Hanns Ruder. Synthesis of two-dimensional human walking: a test of the $\lambda$-model. *Biological cybernetics*, 89(2):89–106, 2003.

[133] Vittorio Sanguineti Rafael Laboissière Paul L. Gribble, David J. Ostry. Are complex control signals required for human arm movement? *Journal of Neurophysiology*, 79:1409–1424, 1998.

[134] Paul L Gribble and David J Ostry. Compensation for loads during arm movements using equilibrium-point control. *Experimental Brain Research*, 135(4):474–482, 2000.

[135] Justin Won and Neville Hogan. Stability properties of human reaching movements. *Experimental Brain Research*, 107(1):125–136, 1995.

[136] M Ghafouri and AG Feldman. The timing of control signals underlying fast point-to-point arm movements. *Experimental brain research*, 137(3-4):411–423, 2001.

[137] Vittorio Gallese and Alvin Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in cognitive sciences*, 2(12):493–501, 1998.

[138] Vittorio Gallese, Luciano Fadiga, Leonardo Fogassi, and Giacomo Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119(2):593–609, 1996.

[139] Scott T Grafton, Luciano Fadiga, Michael A Arbib, and Giacomo Rizzolatti. Premotor cortex activation during observation and naming of familiar tools. *Neuroimage*, 6(4):231–236, 1997.

[140] Luciano Fadiga, Leonardo Fogassi, Vittorio Gallese, and Giacomo Rizzolatti. Visuomotor neurons: Ambiguity of the discharge or 'motor'perception? *International journal of psychophysiology*, 35(2):165–177, 2000.

[141] M Kellenbach, Matthew Brett, and Karalyn Patterson. Actions speak louder than functions: the importance of manipulability and action in tool representation. *Cognitive Neuroscience, Journal of*, 15(1):30–46, 2003.

[142] Consuelo B Boronat, Laurel J Buxbaum, H Coslett, Kathy Tang, Eleanor M Saffran, Daniel Y Kimberg, and John A Detre. Distinctions between manipulation and function knowledge of objects: evidence from functional magnetic resonance imaging. *Cognitive Brain Research*, 23(2):361–373, 2005.

[143] Michael A Arbib and Giacomo Rizzolatti. Neural expectations: A possible evolutionary path from manual skills to language. *Communication & Cognition*, 1996.

[144] Katherine R Sherrill, Elizabeth R Chrastil, Robert S Ross, Uğur M Erdem, Michael E Hasselmo, and Chantal E Stern. Functional connections between optic flow areas and navigationally responsive brain regions during goal-directed navigation. *Neuroimage*, 118:386–396, 2015.

[145] K.J. Friston, C.D. Frith, R.J. Dolan, C.J. Price, S. Zeki, J.T. Ashburner, W.D. Penny, and R.S.J. Frackowiak. *Human Brain Function*. Human Brain Function. Elsevier Science, 2004.

[146] Monica Maranesi, Luca Bonini, and Leonardo Fogassi. Cortical processing of object affordances for self and others' action. *Frontiers in psychology*, 5:538, 2014.

[147] J.K. Aggarwal and M.S. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43(3):16:1–16:43, April 2011.

[148] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, 2013.

[149] Yezhou Yang and Yiannis Aloimonos. A cognitive system for human manipulation action understanding. In *the Second Annual Conference on Advances in Cognitive Systems (ACS)*, volume 2, 2013.

[150] Eren Erdal Aksoy, Minijia Tamosiunaite, Rok Vuga, Ales Ude, C Geib, Mark Steedman, and Florentin Worgotter. Structural bootstrapping at the sensorimotor level for the fast acquisition of action knowledge for cognitive robots. In *Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on*, pages 1–8. IEEE, 2013.

[151] Shun-ichi Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27(2):77–87, 1977.

[152] Samuel A Ellias and Stephen Grossberg. Pattern formation, contrast control, and oscillations in the short term memory of shunting on-center off-surround networks. *Biological Cybernetics*, 20(2):69–98, 1975.

[153] Yulia Sandamirskaya, Stephan KU Zibner, Sebastian Schneegans, and Gregor Schöner. Using dynamic field theory to extend the embodiment stance toward higher cognition. *New Ideas in Psychology*, 31(3):322–339, 2013.

[154] Annette Bastian, Gregor Schöner, and Alexa Riehle. Preshaping and continuous evolution of motor cortical representations during movement preparation. *European Journal of Neuroscience*, 18(7):2047–2058, 2003.

[155] Yulia Sandamirskaya and Gregor Schöner. An embodied account of serial order: How instabilities drive sequence generation. *Neural Networks*, 23(10):1164–1179, 2010.

[156] Y. Sandamirskaya and G. Schöner. Serial order in an acting system: A multidimensional dynamic neural fields implementation. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, pages 251–256, Aug 2010.

[157] Y. Sandamirskaya, M. Richter, and G. Schöner. A neural-dynamic architecture for behavioral organization of an embodied agent. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2, pages 1–7, August 2011.

[158] Fernando Lopes da Silva. Neural mechanisms underlying brain waves: from neural membranes to networks. *Electroencephalography and clinical neurophysiology*, 79(2):81–83, August 1991.

[159] Jörn M. Horschig, Johanna M. Zumer, and Ali Bahramisharif. Hypothesis-driven methods to augment human cognition by optimizing cortical oscillations. *Frontiers in Systems Neuroscience*, 8(119), 2014.

[160] Rainer Menzner, Axel Steinhage, and Wolfram Erlhagen. Generating interactive robot behavior: A mathematical approach. In *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, pages 135–144. The MIT Press/Bradford Books, 2000.

[161] Ioannis Iossifidis and Axel Steinhage. Controlling an 8 dof manipulator by means of neural fields. In *International Conference on Field and Service Robotics*, pages 1–7, 2001.

[162] Yao Lu, Yuzuru Sato, and Shun-ichi Amari. Traveling bumps and their collisions in a Two-Dimensional neural field. *Neural Computation*, 23(5):1248–1260, Feb 2011.

[163] David I Perrett, Mark H Harries, Ruth Bevan, S Thomas, PJ Benson, AJ Mistlin, AJ Chitty, JK Hietanen, and JE Ortega. Frameworks of analysis for the neural representation of animate objects and actions. *Journal of Experimental Biology*, 146(1):87–113, 1989.

[164] Nikolaus F Troje. Reference frames for orientation anisotropies in face recognition and biological-motion perception. *PERCEPTION-LONDON-*, 32(2):201–210, 2003.

[165] Martin A Giese. Biological and body motion perception. *Oxford Handbook of Perceptual Organization*, 2014.

[166] Vittorio Caggiano, Leonardo Fogassi, Giacomo Rizzolatti, Joern K Pomper, Peter Thier, Martin A Giese, and Antonino Casile. View-based encoding of actions in mirror neurons of area f5 in macaque premotor cortex. *Current Biology*, 21(2):144–148, 2011.

[167] Elisabeta Marinoiu, Dragos Papava, and Cristian Sminchisescu. Pictorial human spaces: How well do humans perceive a 3d articulated pose? In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1289–1296. IEEE, 2013.

[168] Isabelle Bülthoff, Heinrich Bülthoff, and Pawan Sinha. Top-down influences on stereoscopic depth-perception. *Nature neuroscience*, 1(3):254–257, 1998.

[169] Sebastian Schneegans and Gregor Schöner. A neural mechanism for coordinate transformation predicts pre-saccadic remapping. *Biological cybernetics*, 106(2):89–109, 2012.

[170] Robert L. Williams II. Engineering biomechanics of human motion. Technical report, Ohio University, 2013.

[171] Jerome A Feldman. Four frames suffice: A provisional model of vision and space. *Behavioral and Brain Sciences*, 8(02):265–289, 1985.

[172] David and Marr. *Vision: A computational investigation into the human representation and processing of visual information*. WH San Francisco: Freeman and Company, 1982.

[173] David Marr and Herbert Keith Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 200(1140):269–294, 1978.

[174] Apostolos P Georgopoulos, John F Kalaska, Roberto Caminiti, and Joe T Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience*, 2(11):1527–1537, 1982.

[175] Apostolos P Georgopoulos, Andrew B Schwartz, and Ronald E Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

[176] Apostolos P. Georgopoulos and Elissaios Karageorgiou. Representations of voluntary arm movements in the motor cortex and their transformations. In Thomas F. Shipley and Jeffrey M. Zacks, editors, *Understanding Events: From Perception to Action*, chapter 9, pages 229–254. Oxford University Press, 2008.

[177] Peter Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, 2005.

[178] W.T. Newsome and C.D. Salzman. The neuronal basis of motion perception. *Ciba Found Symposium*, -:217–230, Jan 1993.

[179] D.I. Perrett, M. W. Oram, M. H. Harries, R. Bevan, J. K. Hietanen, P. J. Benson, and S. Thomas. Viewer-centred and object-centred coding of heads in the macaque temporal cortex. *Experimental Brain Research*, 86(1):159–173, 1991.

[180] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience*, 2(11):1527–1537, November 1982.

[181] M Taira, S Mine, AP Georgopoulos, A Murata, and H Sakata. Parietal cortex neurons of the monkey related to the visual guidance of hand movement. *Experimental brain research*, 83(1):29—36, 1990.

[182] J Randall Flanagan and Roland S Johansson. Action plans used in action observation. *Nature*, 424(6950):769–771, 2003.

[183] Carol L Colby, Jean-René Duhamel, and Michael E Goldberg. Ventral intraparietal area of the macaque: anatomic location and visual response properties. *Journal of neurophysiology*, 69:902–902, 1993.

[184] M Oram and DI Perrett. Responses of anterior superior temporal polysensory (stpa) neurons to "biological motion" stimuli. *Cognitive Neuroscience, Journal of*, 6(2):99–116, 1994.

[185] George Mather, Kirstyn Radford, and Sophie West. Low-level visual processing of biological motion. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 249(1325):149–155, 1992.

[186] Marius V Peelen and Paul E Downing. The neural basis of visual body perception. *Nature Reviews Neuroscience*, 8(8):636–648, 2007.

[187] William W Gaver. Technology affordances. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 79–84. ACM, 1991.

[188] Oliver Lomp, Kasim Terzić, Christian Faubel, JMH du Buf, and Gregor Schöner. Instance-based object recognition with simultaneous pose estimation using keypoint maps and neural dynamics. In *Artificial Neural Networks and Machine Learning–ICANN 2014*, pages 451–458. Springer, 2014.

[189] Daniel Roetenberg, Henk Luinge, and Per Slycke. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. Technical report, XSENS TECHNOLOGIES, 2013.

[190] Cosivina: Compose, simulate, and visualize neurodynamic architectures, an open source toolbox for matlab (accessed: May 27th 2015): https://bitbucket.org/sschneegans/cosivina.

[191] Dare A Baldwin, Jodie A Baird, Megan M Saylor, and M Angela Clark. Infants parse dynamic action. *Child development*, 72(3):708–717, 2001.

[192] Paolo De Leva. Adjustments to zatsiorsky-seluyanov's segment inertia parameters. *Journal of biomechanics*, 29(9):1223–1230, 1996.

[193] Yulia Sandamirskaya. Dynamic neural fields as a step towards cognitive neuromorphic architectures. *Frontiers in Neuroscience*, 7(276):1–13, January 2014.

[194] Jerry Zadny and Harold B Gerard. Attributed intentions and informational selectivity. *Journal of Experimental Social Psychology*, 10(1):34–52, 1974.

[195] B. Duran and Y. Sandamirskaya. Neural dynamics of hierarchically organized sequences: A robotic implementation. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 357–362, Nov 2012.

[196] Erhan Oztop, Mitsuo Kawato, and Michael Arbib. Mirror neurons and imitation: A computationally guided review. *Neural Networks*, 19(3):254–271, 2006.

[197] Barıs Akgün, D Tunaöglu, and Erol Sahin. Action recognition through an action generation mechanism. In *International Conference on Epigenetic Robotics (EPIROB)*, 2010.

[198] Anatol G Feldman. Space and time in the context of equilibrium-point theory. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(3):287–304, 2011.

[199] Tamar Flash. The control of hand equilibrium trajectories in multi-joint arm movements. *Biological cybernetics*, 57(4):257–274, 1987.

[200] ML Latash and GL Gottlieb. Reconstruction of shifting elbow joint compliant characteristics during fast and slow movements. *Neuroscience*, 43(2):697–712, 1991.

[201] Mindy F Levin, Anatol G Feldman, Theodore E Milner, and Yves Lamarre. Reciprocal and coactivation commands for fast wrist movements. *Experimental brain research*, 89(3):669–677, 1992.

[202] J Randall Flanagan, David J Ostry, and Anatol G Feldman. Control of trajectory modifications in target-directed reaching. *Journal of motor behavior*, 25(3):140–152, 1993.

[203] N St-Onge, SV Adamovich, and AG Feldman. Control processes underlying elbow flexion movements may be independent of kinematic and electromyographic patterns: experimental study and modelling. *Neuroscience*, 79(1):295–316, 1997.

[204] Anatol G. Feldman. *Different Forms of Referent Control*, pages 97–128. Springer New York, New York, NY, 2015.

[205] V Martin, John P Scholz, and Gregor Schöner. Redundancy, self-motion, and motor control. *Neural computation*, 21(5):1371–1414, 2009.

[206] Gregor Schöner, Michael Dose, and Christoph Engels. Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and autonomous systems*, 16(2):213–245, 1995.

[207] Gregor Schöner and Michael Dose. A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems*, 10(4):253–267, 1992.

[208] Axel Steinhage and Gregor Schoener. Dynamical systems for the behavioral organization of autonomous robot navigation. In *Sensor Fusion and Decentralized Control in Robotic Systems*, volume 3523, pages 169–180, 1998.

[209] Ioannis Iossifidis and G Schoner. Dynamical systems approach for the autonomous avoidance of obstacles and joint-limits for an redundant robot arm. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 580–585. IEEE, 2006.

[210] Apostolos P. Georgopoulos and Elissaios Karageorgiou. *Representations of Voluntary Arm Movements in the Motor Cortex and Their Transformations*. Oxford University Press, 5 2008.

[211] Apostolos P Georgopoulos, Ronald E Kettner, and Andrew B Schwartz. Primate motor cortex and free arm movements to visual targets in three-dimensional space. ii. coding of the direction of movement by a neuronal population. *Journal of Neuroscience*, 8(8):2928–2937, 1988.

[212] L. Perko. *Differential Equations and Dynamical Systems*. Texts in Applied Mathematics. Springer, 2001.

[213] Estela Bicho, Pierre Mallet, and Gregor Schöner. Target representation on an autonomous vehicle with low-level sensors. *The International Journal of Robotics Research*, 19(5):424–447, 2000.

[214] AG Feldman, SV Adamovich, DJ Ostry, and JR Flanagan. The origin of electromyograms explanations based on the equilibrium point hypothesis. In *Multiple muscle systems*, pages 195–213. Springer, 1990.

[215] Paul L Gribble and David J Ostry. Origins of the power law relation between movement velocity and curvature: modeling the effects of muscle mechanics and limb dynamics. *Journal of Neurophysiology*, 76(5):2853–2860, 1996.

[216] KN An, FC Hui, BF Morrey, RL Linscheid, and EY Chao. Muscles across the elbow joint: a biomechanical analysis. *Journal of biomechanics*, 14(10):659–669, 1981.

[217] Jack M Winters, Savio LY Woo, et al. *Multiple muscle systems*. Springer-Verlag, 1990.

[218] Pascale Pigeon, L'Hocine Yahia, and Anatol G Feldman. Moment arms and lengths of human upper limb muscles as functions of joint angles. *Journal of Biomechanics*, 29(10):1365–1370, 1996.

[219] Felix E Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical reviews in biomedical engineering*, 17(4):359–411, 1988.

[220] Rafael Laboissiere, David J Ostry, and Anatol G Feldman. The control of multimuscle systems: human jaw and hyoid movements. *Biological cybernetics*, 74(4):373–384, 1996.

[221] James C Houk, William Z Rymer, and Patrick E Crago. Dependence of dynamic response of spindle receptors on muscle length and velocity. *Journal of Neurophysiology*, 46(1):143–166, 1981.

[222] Andrew F Huxley. Muscle structure and theories of contraction. *Prog. Biophys. Biophys. Chem*, 7:255–318, 1957.

[223] AG Feldman and GN Orlovsky. The influence of different descending systems on the tonic stretch reflex in the cat. *Experimental neurology*, 37(3):481–494, 1972.

[224] AG Feldman. Superposition of motor programs i. rhythmic forearm movements in man. *Neuroscience*, 5(1):81–90, 1980.

[225] Mark L Latash and Gerald L Gottlieb. An equilibrium-point model for fast, single-joint movement: I. emergence of strategy-dependent emg patterns. *Journal of Motor Behavior*, 23(3):163–177, 1991.

[226] Paul L Gribble, Lucy I Mullin, Nicholas Cothros, and Andrew Mattar. Role of cocontraction in arm movement accuracy. *Journal of neurophysiology*, 89(5):2396–2405, 2003.

[227] Arthur J Miller. *Craniomandibular muscles: their role in function and form*. CRC Press, 1991.

[228] Karl Hainaut, Jacques Duchateau, and Jean Edouard Desmedt. Differential effects of slow and fast motor units of different programs of brief daily muscle training in man. *New developments in electromyography and clinical neurophysiology*, pages 241–249, 1981.

[229] Jean-François Pilon, Sophie J De Serres, and Anatol G Feldman. Threshold position control of arm movement with anticipatory increase in grip force. *Experimental brain research*, 181(1):49–67, 2007.

[230] Kazuhisa Domen, Mark L Latash, and Vladimir M Zatsiorsky. Reconstruction of equilibrium trajectories during whole-body movements. *Biological cybernetics*, 80(3):195–204, 1999.

[231] Estela Bicho, Wolfram Erlhagen, Luis Louro, and Eliana Costa e Silva. Neuro-cognitive mechanisms of decision making in joint action: A human robot interaction study. *Human Movement Science*, 30(5):846 – 868, 2011. {EWOMS} 2009: The European Workshop on Movement Science.

[232] R. Wilensky. Why john married mary: Understanding stories involving recurring goals. *Cognitive Science*, 2(3):235–266, 1978.

[233] R. Wilensky. *Planning and Understanding: A Computational Approach to Human Reasoning*. Artificial Intelligence. Addison-Wesley Publishing Company Advanced Book Program, 1983.

[234] J. F. Allen and R Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15(3):143–178, 1980.

[235] Jan Alexandersson. Plan recognition in verbmobil. In *Proceedings of the IJCAI-95 Workshop "The Next Generation of Plan Recognition Systems: Challanges for an Insight from Related Areas of AI"*, pages 2–7, Montreal, Canada, 8 1995.

[236] Jan Alexandersson, Norbert Reithinger, and Elisabeth Maier. Insights into the dialogue processing of verbmobil. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 33–40. Applied Natural Language, April 1997.

[237] James F. Allen, Bradford W. Miller, Eric K. Ringger, and Teresa Sikorski. A robust system for natural spoken dialogue. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 62–70, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[238] George Ferguson, James Allen, and Brad Miller. Trains-95: Towards a mixed-initiative planning assistant. In *in Proceedings of the 3rd Conference on AI Planning Systems*, 1996.

[239] George Ferguson and James F. Allen. Trips: An integrated intelligent problem-solving assistant. In *In Proc. 15th Nat. Conf. AI*, pages 567–572. AAAI Press, 1998.

[240] Douglas E. Appelt and Martha E. Pollack. Weighted abduction for plan ascription. In *Technical Note 491, SRI International, Menlo Park*, pages 1–25, 1992.

[241] Kurt Konolige and Martha E. Pollack. Ascribing plans to agents. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'89, pages 924–930, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[242] David W. Albrecht, Ingrid Zuckerman, Ann E. Nicholson, and Ariel Bud. Towards a bayesian model for keyhole plan recognition in large domains. In *In Proceedings of the Sixth International Conference on User Modeling*, pages 365–376. Springer-Verlag, 1997.

[243] Robert P. Goldman, Christopher W. Geib, and Christopher A. Miller. A new model of plan recognition. *Artificial Intelligence*, 64:53–79, 1999.

[244] Gita Sukthankar and Katia Sycara. A cost minimization approach to human behavior recognition. In *Proceedings of Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1067–1074, July 2005.

[245] Chris L. Baker and Joshua B. Tenenbaum. Chapter 7 - modeling human plan recognition using bayesian theory of mind. In Gita Sukthankar, Christopher Geib, Hung Hai Bui, David V. Pynadath, and Robert P. Goldman, editors, *Plan, Activity, and Intent Recognition*, pages 177 – 204. Morgan Kaufmann, Boston, 2014.

[246] Prashant Doshi, Xia Qu, Adam Goodie, and Diana Young. Modeling recursive reasoning by humans using empirically informed interactive pomdps. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 1223–1230. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[247] Prashant J Doshi. Decision making in complex multiagent contexts: A tale of two frameworks. *AI Magazine*, 33(4):82, 2012.

[248] Karl Spencer Lashley. The problem of serial order in behavior. In *Cerebral mechanisms in behavior*, pages 112–136. 1951.

## Supervised Students' Theses

[249] Christian Busch. Development of a neural field model for human intention recognition. mathesis, Lehrstuhl fur STEUERUNGS- und REGELUNGSTECHNIK Technische Universitat Munchen, 2014.

[250] Katharina Lehmer. Physics-based modeling of human motion. Bachelor's Thesis, September 2013. Lehrstuhl fur STEUERUNGS- und REGELUNGSTECHNIK Technische Universitat Munchen.

## Author's Publications

[251] Laith Alkurdi, Christian Busch, and Angelika Peer. Dynamic contextualization and comparison as the basis of biologically inspired action understanding. *Journal of behavioural robotics*, 2018.