BIO-INSPIRED ADAPTIVE TUNING OF HUMAN-ROBOT INTERFACES

By

Bakur A.H. AlQaudi

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2018

Acknowledgments

I would like to express my sincere thanks to my advisor Prof. Frank L. Lewis for his guidance, support, and motivation during my doctoral research. I would forever be indebted to him for introducing me to the fields of the control system and model reference control of the Human-robot interaction. In addition to the science, and research ethics I have learned insights from him in my research, and life.

Also, I would like to express my sincere thanks to my co-advisor Prof. Daniel S. Levine for his supervision, care, and enthusiasm during my doctoral research. I learned a lot from him, especially in his area of expertise in cognitive modeling, and neural networks. This research affected me personally, learning about the human brain, and its functions. In addition to the science, and research ethic I have learned insights from him in my research, and life.

I am thankful to my committee members, Dr. William E. Dillan, Dr. Yan Wan and Dr. Michael T. Manry, for their insightful comments. I thank my collaborators, Dr. Hamidreza Modares for giving me a new perspective on reinforcement learning, and mathematical proof. Dr. Bahare Kiumarsi for introducing me to the field of multiple model systems, and Q-learning mathematical foundations. Also, I would like my coauthors of the publication Prof. Kay-Yut Chen, Dr. Isura Ranatunga, Shaikh M. Tousif, Dr. Dan O. Popa. I am also extending my appreciation to my research group at UTARI, including Dr. Mohammed Abouheaf, Dr. Giulio Binetti, Raghavendra Sriram, and Patanjali Joshi.

Last but not least, I would want to thank my parents, Awdah and Maryam, for their unconditional support and encouragement during my student time at the University of Texas at Arlington. My beloved wife, Hanin Almuzaini, for her patience and unconditional support. My brothers, and sisters, especially my late brother Ibrahim, who left before seeing the completion of this work. My amazing friends, especially, Mousa Almotairi, and Sakher Ghanem.

Table of Contents

Table of Figures

Abstract

Supervising professor: Frank L. Lewis.

Motivated by recent advancement in neurocognitive in brain modeling research, multiple model-based Q-learning structures are proposed for optimal tracking problem of time-varying discrete-time systems. This is achieved by utilizing a multiple-model scheme combined with adaptive resonance theory (ART), and dopamine-like model. In the ART algorithm , dopamine-like model and generates sub-models based on the match-based clustering method utilizing. A responsibility signal governs the likelihood of contribution of each sub-model to the Q-function. The Q-function is learned using the batch least-square algorithm. Simulation results are added to show the performance and the effectiveness of the overall proposed control method.

A novel enhanced human-robot interaction system based on model reference adaptive control is presented. The presented method delivers guaranteed stability and task performance and has two control loops. A robot-specific inner loop, which is a neuroadaptive controller, learns the robot dynamics online and makes the robot respond like a prescribed impedance model. This loop uses no task information, including no prescribed trajectory. A task-specific outer loop takes into account the human operator dynamics and adapts the prescribed robot impedance model so that the combined human-robot system has desirable characteristics for task performance. This design is based on model reference adaptive control, but of a nonstandard form. The net result is a controller with both adaptive impedance characteristics and assistive inputs that augment the human operator to provide improved task performance of the human-robot team. Simulations verify the performance of the proposed controller in a repetitive point-to-point motion task. Actual experimental implementations on a PR2 robot further corroborate the effectiveness of the approach.

Chapter 1 Introduction

1.1 Background And Motivation

Optimal control encompasses the design of a control policy that satisfies a tracking or regulation control objective while simultaneously minimizes a performance function. A sufficient condition to find a feedback solution to an optimal regulation problem is to solve the Hamilton-Jacobi-Bellman (HJB) equation. For linear systems with quadratic performance function, the HJB equation reduces to the algebraic Riccati equation (ARE). For the case of optimal tracking problem, however, traditional solutions are composed of two components; a feedback term obtained by solving an HJB equation and a feedforward term obtained a priori by either solving a differential equation. The feedback term tries to stabilize the tracking error dynamics and the feedforward term tries to guarantee faultless tracking. Algorithms for computing the feedback and feedforward terms are traditionally based on offline solution methods which require complete knowledge of the system dynamics.

Another approach is to combine complexity reduction using multiple model architecture that contains identification models operating in parallel; may either be fixed or may be tuned from an initially chosen value. Coupled with an optimal model to solve the Hamilton-Jacobi-Bellman (HJB) equation. For linear systems with quadratic performance function, the HJB equation reduces to the algebraic Riccati equation (ARE). The purpose of these models identifies the operation point of the environment, and for computing the feedback and feedforward terms for optimality.

An advantage of these combined system model-free RL algorithms for systems require measurement of the system states. However, it is not possible to measure the full states of the systems in many practical situations. Realizing how humans underlying neural activity in the brain gives rise to

emergent conformity to general laws of decision making is a central focus of many research institutes. Furthermore, there is a thrust to apply the understanding of these neuro-cognitive research in an area like cognitive control, financial markets, and economic studies. Biological brains can select actions which are most of the time are based on either past experiences, or results that the results might hold. The functionality of the brain is about making choices which yield better results.

This work is aimed at providing links between neuroscience, psychology and control systems. Detailed study of mechanism of computations and decisions in human brain has been presented. It is further strengthened with findings from a psychological perspective. Architectures for learning and control which are inspired through, and use all these findings are presented so that an integrated compilation has been prepared on basis of which faster, more efficient decisions and control structures can be designed for various autonomous systems.

1.2 Contribution and Outline

In chapter two, a brief literature review of the recent cognitive studies is discussed. This include the discussion of the Human Factor approach where the human body is modeled, then Neuro-Cognitive Approach is explored including Neuroscience and psychological models, Cognitive brain-like models and multiple model approach.

In chapter three multiple model-based Q-learning structures are proposed for optimal tracking problem of time-varying discrete-time systems. This is achieved by utilizing a multiple-model scheme combined with adaptive resonance theory (ART), where generates sub-models based on the match-based clustering method utilizing.

In chapter four, the same multiple model-based Q-learning structures are proposed for optimal tracking problem of time-varying discrete-time systems, utilizing dopamine-like model to evaluate the contribution of each model.

In chapter 5, An enhanced human-robot interaction system based on model reference adaptive control is presented. The presented method delivers guaranteed stability and task performance and has two control loops. A robot-specific inner loop, which is a neuroadaptive controller, learns the robot dynamics online and makes the robot respond like a prescribed impedance model.

1.3 Publications resulted from this work

- Modares, H., Ranatunga, I., Alqaudi, B., Lewis, F. L., & Popa, D. O. Intelligent human–robot interaction systems using reinforcement learning and neural networks. In Y. Wang & F. Zhang (Eds.), Trends in control and decision-making for human–robot collaboration systems (pp. 153–176). Berlin: Springer.
- B. Alqaudi, H. Modares, I. Ranatunga, S. M. Tousif, F. L. Lewis, D. O. Popa, "Model reference adaptive impedance control for physical human-robot interaction", Control Theory and Technology, vol. 14, pp. 68-82, 2016.
- B. Alqaudi, B Kiumarsi, D.S. Levine, F. L. Lewis, Optimal Control Using Multiple Adaptive Resonance Theory and Q-Learning, submitted to neurocomputing
- B. Alqaudi, D.S. Levine, F.L. Lewis, "Neural network model of decisions on the Asian disease problem", Proceedings of International Joint Conference on Neural Networks 2015, pp. 1333-1340.
- D. S. Levine, K. Y. Chen and B. Alqaudi, "Neural network modeling of business decision making," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 206-213.

Chapter 2 Literature review

Every day the control engineering is evolving, its limits have persistently extended. It is enormous leaps from classical regulation with simple proportional-integral-derivative (PID) loops, to model-based control and multivariable structures, to modern control theory, to hybrid, hierarchical, reinforcement architectures, and most recently understand systems as networks and control them using other existing methods like graph theory, and game theory. The advancement is equally spread in theoretical foundations and application scope have seen extraordinary progress.

A natural extension of these leaps in the control theory literature is the cognitive control. It arises from the fact that current automatous systems function performs excellently in environments they are designed for namely around their nominal operating environments. Moreover, most of the previous, and current system function adequately in environments with foreseeable uncertainties as in the advanced adaptive and robust control structures without the presence of operators. Yet, control systems of currently necessitate substantial human intervention when confronted by a novel and unanticipated environment conditions. Figure 1-1 shows a generic architecture of cognitive control system. Such conditions can arise from extreme changes in the environment, extreme disturbances, structural changes in the system0. Furthermore, to illustrate and to give an example consider an autonomous robot in search and rescue operations, encounter novel situations that require perception, reasoning, decision making and most importantly adaptive learning. Such cognitive control aspects play a crucial role in autonomous systems and will advance control system to new leaps.

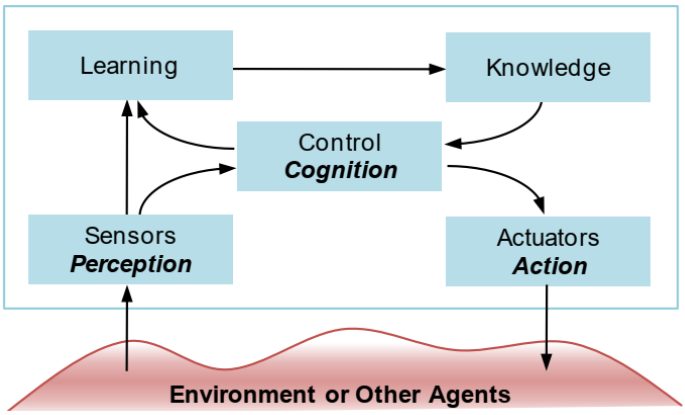

Figure 2-1: Cognitive Control System Architecture

Moreover, cognitive control arises to include the human factor; in today's control systems for applications such as aircraft, industrial factories, and so on. This human-agent interaction aspect proposes another essential focus for cognitive control. Multi-agent coordination, cooperation, and control help in the execution of complex tasks, effective operation in mixed competitive-cooperative situations that require participating agents to have cognitive-like capabilities[2].

There are a lot of definitions of cognitive control; however, one of the best is the one defined by the researcher of the euCognition project funded by the European Commission. In their report, the described a cognitive system by a system that exhibits goal-oriented behavior in sensing, reasoning, and action. A cognitive system requires flexibility to changes its goals and function depending on situational context and experience, and ability to act in unstructured environments without human intervention and robustly responds to surprise; or able to interact with humans and other cognitive systems to jointly solve a complex task [3]. Cognitive control draws its main influences from; first, Systemic Neuroscience, Cognitive Science, and Neuro-cognitive Psychology where the developed computational models of control, perception, and control policies are applied based on experimental studies. Second, Information processing technology where algorithms realize cognitive capabilities essentially in inference and reasoning. Lastly, engineering technologies in mechatronics, controllability, stability, model/knowledge

representation, is utilized to implement robust cognitive abilities in guaranteed performance constraints. Figure 1-2 illustrates the interrelation of these three influences.

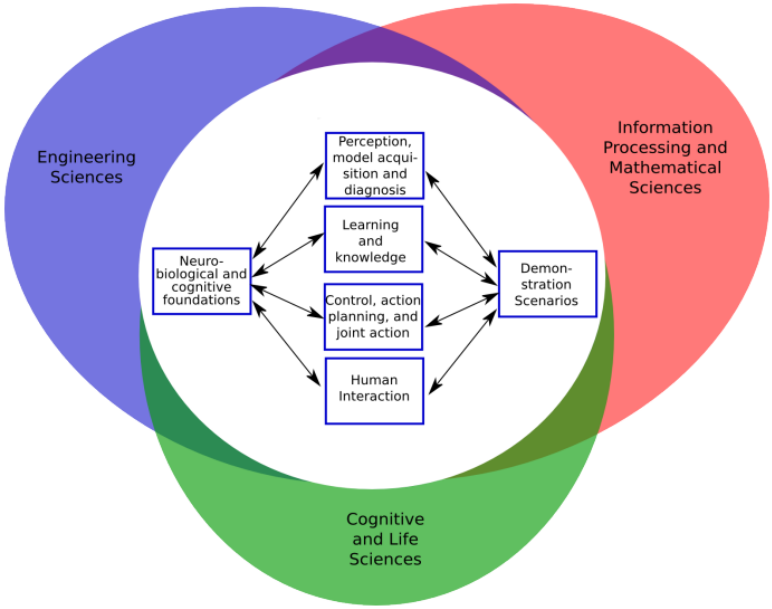

Figure 2-2: Cognitive Control Influences

Furthermore, there is a great deal of intermixing between cognitive control and artificial indigence; and here ones want to highlight the differences between the two. While the two terms have been used numerously to describe each other, and there are a lot of similarities between the two areas of research. Nevertheless, artificial intelligence research does not involve mimicking of human thought processes; Instead, AI systems are concerned with the optimality, best possible algorithms for solving a given problem as the primary outcome. In order to illustrate the problem, consider the example of the autonomous car, AI system starts with the goal of avoiding collisions and staying on course, not mimic the process of the human brain in driving. On the other hand, cognitive control does not make decisions for humans but instead supplements our human decision-making process. Furthermore, create a human-like control to simplify complex tasks[4]. The cognitive control provides human-like system to achieve optimal decisions, or control human embedded system to achieve optimality and stability by augmenting the

system to human liking; while AI is rooted in the idea that system can perform better control decisions on the human behalf.

There are significant efforts to develop systems in which humans and autonomy are responsible for cooperative sensory data acquisition, perception, cognition, and decision-making. Such cooperative operation is an inevitability, especially for assistive robotics, for example, the autonomy exists to support functionality that the human users cannot perform. Embedding a human as a user, source or decision aid in the operation of autonomous systems enlarges the difficulty. Although humans offer higher cognitive capabilities that complement system functionalities, the impact is depending on the system's ability to infer the intent, preferences, and limitations of the human. With all these efforts in place, there is a significant gap in theory and tools for the design of human-embedded systems and human-like systems. In the literature today, there are two answers for these efforts. First, the research that aims at modeling the human in the loop as standard control transfer function, this can be summarized as human factor studies. Second, the research which is derived from neuroscience, psychological and psychological studies that start with the existing models in those fields. In section 1-1, examples of human factor approach studies are given, and in section 2-2 a discussion of neuro-cognitive approach is discussed.

2.1 Human Factor approach

The first group of research aimed to achieve a transfer function of a human operator in various situation and embed that model in a more extensive control system. This research is mainly performed to determine a human transfer function that allows evaluation and prediction of the performance of manually controlled systems. Significant of the studies in this area was to control manual and robotics systems. one of the early examples of that is the lunar-landing simulator to determine the effect of the drive system characteristics on the performance of the pilot-vehicle combination. In [5] they surveyed the method of two hundred studied that concern this topic. However, most of the current studies involve

finding the human transfer function for a specific task. In controlling joystick or moving in x-y table-like movement. In [6][7] find the human transfer function simple linear first-order system with high bandwidth. This type of control structure is usually accompanied with estimator to increase the accuracy of the human realization in the control loop. [8].

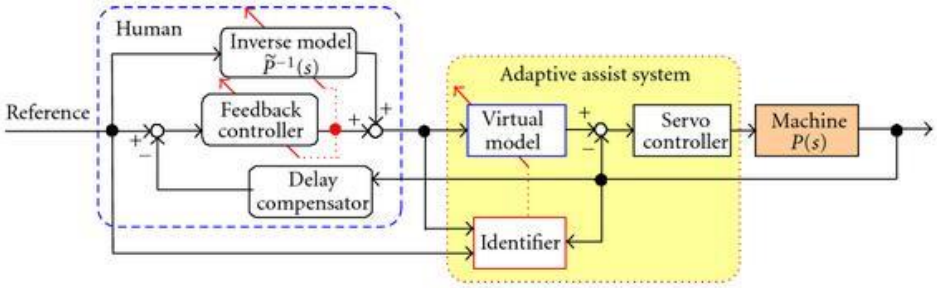

Figure 2-3 : Adaptive Assistive control structure

An excellent example of this is the work of [7], where adaptive assistive control consists of three subsystems: a servo controller, an online identifier of the operator's control characteristics, and a variable dynamics control using adaptive control. The adaptive control utilizes a Lyapunov candidate function using a haptic interface device composed by an XY-stage as shown in figure 1-3.

2.2 Neuro-Cognitive Approach

Realizing how humans and animals make decisions considering the underlying neural activity in the brain gives rise to emergent conformity to general laws of decision making is a central focus of many research institutes. Furthermore, there is a thrust to apply the understanding of these neuro-cognitive research in an area like artificial intelligence and cognitive control, financial markets, and economic studies. Biological brains can select actions which are most of the time are based on either past experiences, or results that the results might hold. The functionality of the brain is about making choices which yield better results. Although functionality is not fully understood, there are proposed and proven

theories that address intelligence, various learning and decision- making processes performed by various parts of the brain.

Within the neurocognitive approach, there are three primary schools of thoughts to answer the question of creating human-like cognitive control system. These three approaches are serious attempts to understand and replicate the human in a generalized, flexible, and adaptive kind of capability that we see in the human brain. The first approach is strictly in-line with the cognitive finding, biologically plausible and limit the capability to the research on neuroscience, and psychology; it is discussed in subsection 1-2-1. The second approach inspires from the existing research and modify it to achieve a generalized structure of control, and it is discussed in subsection 1-2-2. The third approach is the multiple models' approach which started as a computational model and then enhanced by the development of the neurocomputational studies, namely multiple models approach, and it is discussed in subsection1-2-3.

2.2.1 Neuroscience and psychological models

2.2.1.1 Piaget's and Hebbian learning

Jean Piaget argued that human learning development progresses chronologically through stages. Piaget defined learning as the ability to adapt to the environment. Adaptation takes place through assimilation and accommodation, with the two processes interacting throughout life in different ways, according to the stage of learning development. In assimilation, the individual absorbs new information, fitting features of the environment into internal cognitive structures. In accommodation, the individual modifies those internal cognitive structures to adapt to the new information and meet the demands of the environment[9]. Hebbian learning algorithms consider a wide range of behaviors and changes throughout development. These include critical periods, learning of statistical regularities in the environment, development of object knowledge, and development of flexible behaviors. The A basic Hebbian learning rule, and the update of the weight in learning takes the following form:

$$w_{ij}(t) = w_{ij}(t-1) + \Delta w_{ij} \tag{2.1}$$

Where the weight from unit i to unit j at any given time t is the weight value from the previous time step plus the change in weight resulting from the activity of units i and j, and $\Delta w_{ij} = \gamma a_i a_j$ where . γ is the learning rate a_i and a_j are denote the activation levels of units i and j respectively

While the Piaget's and Hebbian learning theories have it is own application, the overwhelming evidence suggests that the brain nervous system in the must be more complicated to implement behavioral plans yet is flexible and responsive to unexpected changes. The evidence-based experiments purposed the demands for mechanism representing features or attributes of short-term learning and another mechanism representing categories of long-term learning. As an answer to this demand, more frameworks have emerged such as Gated Dipole Network, Adaptive Resonance Theory (ART), and Fuzzy Trace Theory (FTT).

2.2.1.2 Gated Dipole Theory

Like the Hebbian learning, the gated dipole theory was spurred by an effort to compare current values of reinforcement variables with recent past values of the same variables. Further, the gated dipole also explained response associated with the absence of a punishing reinforcement. The network is designed so that shutting of an input to one channel leads to transient activation of the other channel. Say, if the two channels represent positive and negative effects or 'on' and 'off' states, the offset of an effectively negative stimulus can produce positive affect and vice versa. In the figure1-4, J is an input which is the phasic reinforcement signal, I nonspecific arousal is tonic which is represented all the time, w_1 and w_2 are synapses and ' x_i 's are activity nodes. After J is shut off, $w_1 < w_2$, so $x_3 < x_4$. By competition, x_5 is activated, enhancing a motor output suppressed by J.

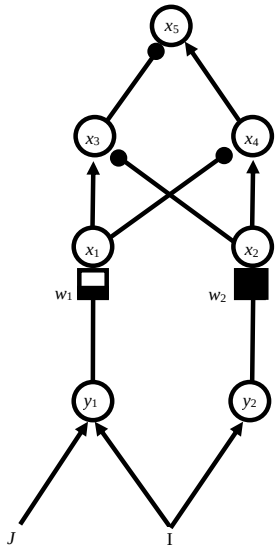

Figure 2-4: Gated Dipole Circuit

Figure 1-4 shows a schematic gated dipole, which obeys Equations 1-2. The synapses w1 and w2, marked with squares are transmitter that tends to be depleted with activity, Other terms in those equations denote new transmitter production[11].

$$\begin{aligned}
&\frac{dy_1}{dt} = -ay_1 + I + J \qquad && \frac{dy_2}{dt} = -ay_2 + I \\
&\frac{dw_1}{dt} = b(c - w_1) - ey_1 w_1 && \frac{dw_2}{dt} = b(c - w_2) - ey_1 w_2 \\
&\frac{dx_1}{dt} = -fx_1 + gy_1 w_1 && \frac{dx_2}{dt} = -fx_2 + gy_2 w_2 \\
&\frac{dx_3}{dt} = -hx_3 + k(x_1 - x_2) && \frac{dx_4}{dt} = -hx_4 + k(x_2 - x_1) \\
&\qquad\qquad \frac{dx_5}{dt} = -mx_5 + k(x_3 - x_4)
\end{aligned} \tag{2.2}$$

where a, b, c, e, f, g, h, k and m are all positive constants. Equation 1-2 reflect a symmetry between the positive and negative channels, they have the decay rates (*a, f,* and *h*), the same depletion rate (*e*), the same transmitter recovery rate (*b*), the same bounded depletable rate (*c*), and the same coefficients for signal transmission between levels (*g* and *k*).

2.2.1.3 Adaptive resonance theory (ART)

ART was developed by Stephen Grossberg and Gail Carpenter on based in the research of the mechanism in which the brain processes information. It has many variations describes several neural network models which use supervised and unsupervised learning methods, and address problems such as classification, clustering, and prediction. Contrary to most of the neural network map models like Kohonen self-organizing map which is pure feed-forward layers projecting unidirectionally to higher processing layers, Grossberg argued that pure feedforward coding or categorization could be unstable, and lack plasticity. This led to the idea of adaptive resonant feedback between two layers of nodes, corresponding to the extensive feedback connections in the visual system in the cortex to lateral geniculate.

The ART network is considering two layers; first, F1 layer is assumed to consist of nodes that respond to input features, analogous to a sensory area of the cerebral cortex. Second, The F2 layer is assumed to consist of nodes that respond to categories of F1 node activity patterns. Synaptic connections between the two fields are modifiable bidirectionally according to two different learning laws. It utilizes competition as a standard tool in neural networks; thus, only the F2 node receiving the most significant signal from F1 becomes active. The simplest form of competition is winner take all is made: only the F2 node receiving the largest signal from F1 becomes active. Then, mapping from F2 field to the F1 field utilizing gain control node is performed. This lead to First, it prevents F2 activity from always exciting F1, and second, it shuts off most neural activity at F1 if there is a mismatch between the input pattern and the active category. If a match occurs, then F1 activity is large because many nodes are simultaneously excited by input and prototype. If a mismatch occurs F1 activity is not sufficient to inhibit a chosen node which thereby becomes active. The selected node activity leads to F2 reset which shuts off the active category node as long as the current input is present. A mismatch reset occurs, and a new category node is chosen if the vigilance criterion is not met. If sufficient match based on the vigilance criterion, the choice

is made more permanent; this is the resonance, and it is called adaptive because the prototype resonating with the input reflects the learning of previous inputs by the node at F2 [12] .

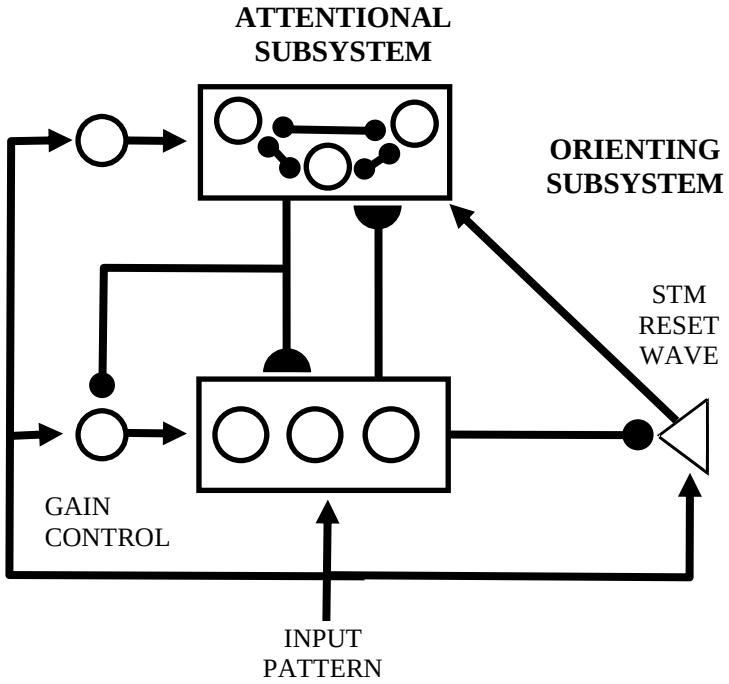

Figure 2-5 : ART 1 architecture

Adaptive resonance theory networks are a massive family including a wide range of precise architectures. The most essential classifications within ART are unsupervised versus supervised and analog versus binary. Figure 1-5 illustrates the basic ART model where Short-term memory at the feature level F1 and category level F2 and bottom up and top down interlevel long term memory traces, are modulated by other nodes. The orienting system generates a reset wave to F2 when bottom up and top down patterns mismatch at F1, that is, when the ratio of F1 activity to input activity is less than a vigilance level.

2.2.1.4 Fuzzy trace theory (FTT)

Fuzzy-trace theory embraces inconsistencies in human reasoning by assuming opposing dual processes. Contrary to the Piaget's theory and bases on counterintuitive data on how memory

development influences the development of reasoning, Fuzzy trace theory propose that there are two different types of learning in human brain. FTT suggests that we encode traces of experiences we learn in two different ways: 'verbatim' (literal meaning) and 'gist' (essential meaning) encoding. The ability to grasp the gist of a problem and ignore relatively minor details facilitates our ability to recognize the problem, in the other hand others encountered before and thereby drawn on our memory of those problems along with increased efficiency. Gist encoding tends to reduce the relative attractiveness of sure losses and enhance the relative attractiveness of sure gains in comparisons with risky alternatives. It can arise from emotions, learning, information or any combinations of these things. These two encodings are heavily influenced by how we frame the decision, and mathematically represented as probabilistic choices. The dual-process assumption of FTT has also been used to explain common biases of probability judgment, including the conjunction and disjunction fallacies. The conjunction fallacy occurs when people mistakenly judge a specific set of circumstances to be more probable than a more general set that includes the specific set. Yet FTT is incomplete as it does not answer what kinds of gist are extracted under what circumstances. So, it is combined with ART and a very close neural network model for brain has been developed for Gambling modeling, Asian decease problem[13], and more recently to retail problem[14].

2.2.2 Cognitive brain-like models

2.2.2.1 Werbos' three generations of Brain-like models

Werbos' work explains the basic mathematical principles and their relation to the most important features of a mammal brain–how intelligence works so that a control system can be designed that can learn to perform the complex range of tasks. Based on the fact there is no artificial system capable of learning to perform the complex range of tasks that the mammal brain can learn to perform. The provided roadmap for how to reach that point, utilizing a neuroscience help by providing a series of qualitative but quantifiable theories of how intelligence works in the mammal brain.

From an engineering and mathematical prospective of the mammalian brain, it is not enough to use the old optimization rule of Hamilton and Jacobi. Rather there is a need to consider the stochastic case because mammals cannot predict our environment in a deterministic way. The foundation for optimization over time in the stochastic case is the Bellman equation, a great breakthrough developed by Bellman. Based on the observation that human brains are not optimal all the time, a possible answer is that mammal brains are designed to learn approximate optimal policy with bounded computational resources. Approximate Dynamic Programming (ADP) models of brain intelligence have been developed[15]. An efficient method to compute an optimal strategy or policy of action for a general nonlinear decision problem over time subject to noise is to use Bellman equation as follows:

$$J(\underline{\mathbf{x}}(t)) = \max_{\underline{\mathbf{u}}(t)} \left\langle U(\underline{\mathbf{x}}(t), \underline{\mathbf{u}}(t)) + J(\underline{\mathbf{x}}(t+1)) \right\rangle / \left(1+r\right) \tag{2.3}$$

where $\underline{\mathbf{x}}(t)$ is the state of the environment at time t, $\underline{\mathbf{u}}(t)$ is the choice of actions, U is the cardinal utility function, r is the interest or discount rate, the angle brackets denote expectation value and J is the function that must be solved in order to derive an optimal strategy of action. A system can learn to approximate this policy by using a neural network to approximate the J function and other key parts of the Bellman equation as shown in figure 1-6.

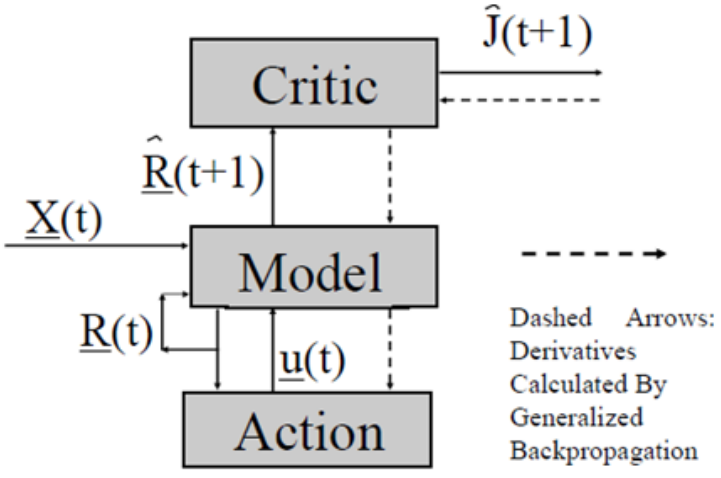

Figure 2-6: First Generation Humanlike ADP model

The second generation of brain-like mechanism [16]is extended utilizing the use of critic networks and model networks which are far more powerful than feedforward neural networks. It requires using the recurrent networks along with networks which “settle down” over many cycles of an inner loop calculation before emitting a calculation. In turn that needs a relatively low sampling rate; about 4-8 frames per second which is the rate observed for the cerebral cortex, in response to new inputs from thalamus-the “movie screen” watched by the upper brain it is shown in figure 1-7.

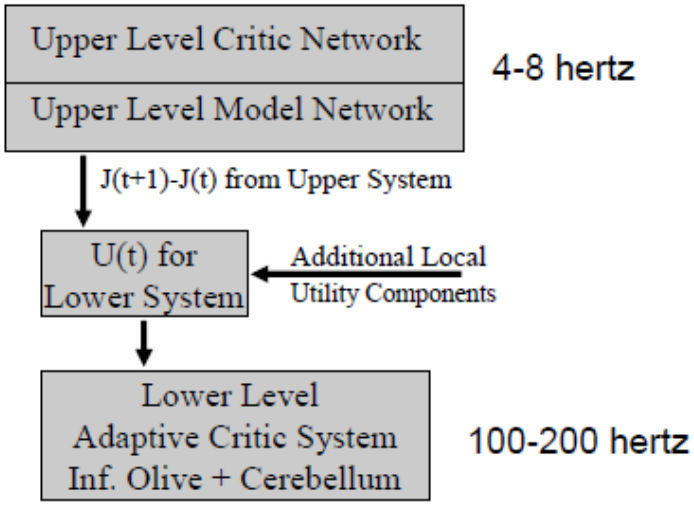

Figure 2-7 Second Generation Humanlike ADP Model

A third generation of human-like intelligence[17] was proposed utilizing temporal complexity. Combining the key capabilities of a Simultaneous Recurrent Network and a “conformal'' network was proposed, to allow This immediate prediction and control and navigation through complex two-dimensional problems. While it provides far more complex than a Multilayer perceptron network, it was not as popular, in part because the learning was slow and it is not easy for people to take advantage of the great brain-like power of such networks. Figure 1-8 shows an architecture of such network.

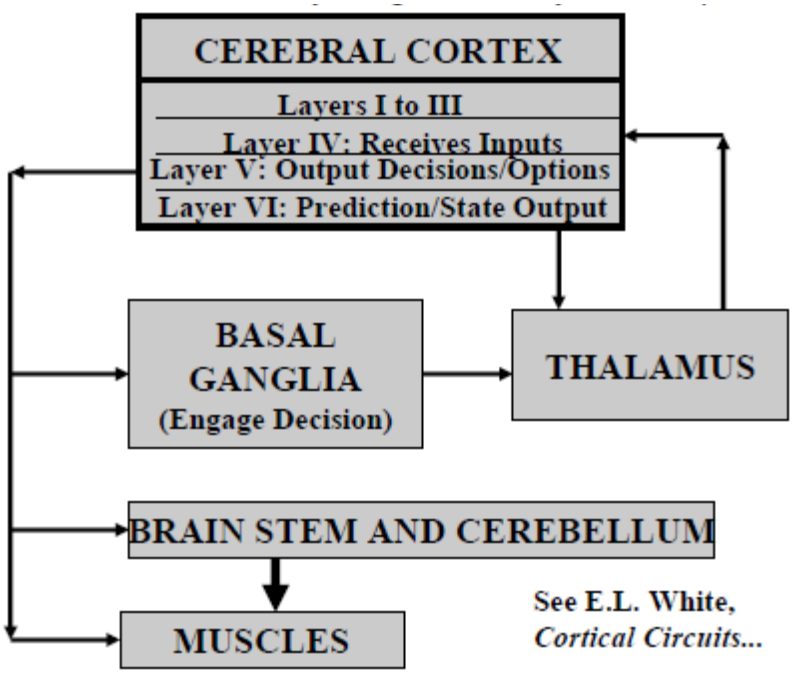

Figure 2-8 : Third Generation Humanlike Model

2.2.2.2 Suri-Schultz Model

Unlike the famous Sutton and Barto model versions of temporal difference TD did not include specific roles for brain regions such as the dopamine, basal ganglia, and prefrontal cortex. There were more explicit simulations of brain regions were gradually included in later extensions of the TD model, starting with Suri and Schultz[18]. The simulations using TD in the Suri-Schultz articles encompassed sequence learning, delayed response, and anticipatory dopamine neuron activity. Their work noted that previous versions of the TD model had predicted that dopamine cell activity would be depressed not only if reward is delivered later than expected, which is supported by data, but also if reward is delivered earlier than expected, which is not supported by data.

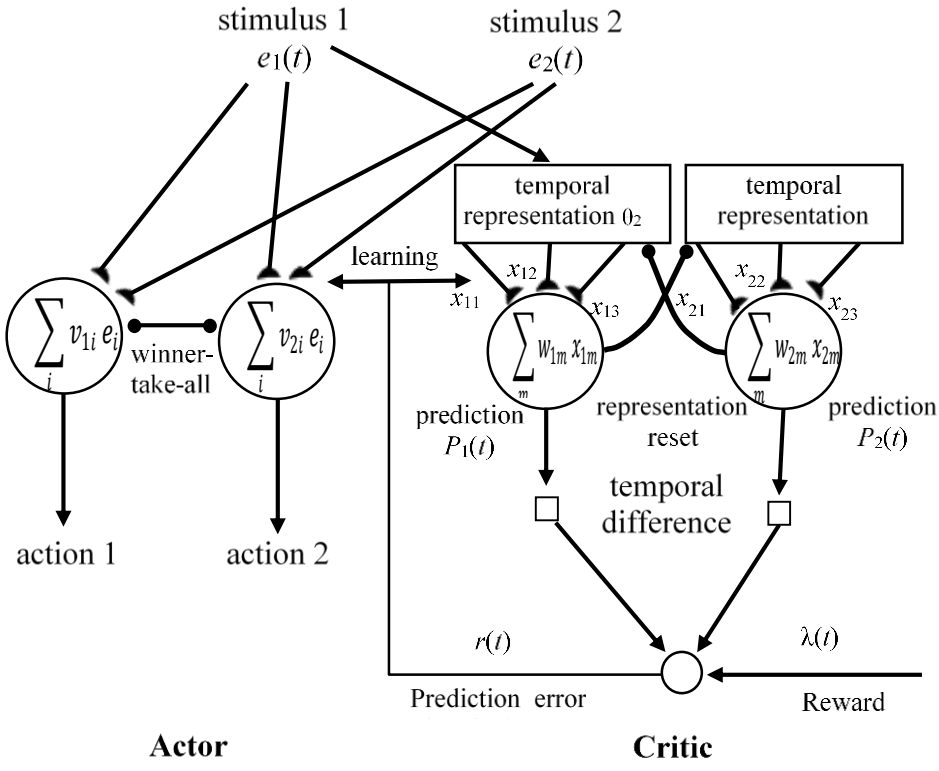

Figure 2-9 : Suri and Schultz model

Figure 1-9 shows a version of the Suri and Schultz TD model. Actor and Critic receive input stimuli 1 and 2 which are coded as functions of time. The Critic computes the effective reinforcement signal $r(t)$ which modifies the weights v_{nl} of the Critic and the weights w_{lm} of the Actor which yield that both actor and critic learn through time. The Critic associates input stimuli 1 and 2 with Signal $r(t)$. Every stimulus is represented as a series of components x_{lm} of different durations, each of which influence the reward prediction signal according to its own adaptive weight w_{lm}. The prediction $P_1(t)$ is computed as the weighted sum of these components. Winner-take-all competition between predictions $P_1(t)$ of different stimuli sets all but one representational component to zero. The change in the prediction of a stimulus is computed by taking the temporal difference between successive time steps. The temporal difference is summed over all stimuli and added to the primary reinforcement signal input coding the reward, leading

to the effective reinforcement signal which codes reward prediction error. The Actor learns to associate stimuli with behavioral actions.

2.2.3 Multiple model approach

It is amazing the multiple model approach originated in both computational\ intelligent control, and cognitive studies at the same time. Advances in study of brain suggests that are multiple control structures working simultaneously in the brain. That has led to the use multiple controllers and/or predictors used in the new controllers. Here, some of them are discussed which use multiple reinforcement learning structures, multiple neural networks and multiple adaptive controllers.

Adaptive Control systems were traditionally designed around a single fixed or slowly adapting the model of the system. This design scheme inherently implies that the operating environment is either time invariant, or varies gradually, within bounded limits, with time. In practice, complex systems operate in multiple operating environments which may change abruptly from one operating point to another. The speed and accuracy with which a controller responds to sudden and substantial changes may be considered as a measure of its adaptiveness and intelligence. While robust adaptive control is restricted to sufficiently small ranges of variations, traditional adaptive control reacts too slowly to abrupt changes, resulting in large transient errors before convergence. This need to robustness, consequently need to be addressed.

Narendra and Balakrishnan [20] proposed multiple model architecture that contains identification models operating in parallel; may either be fixed or may be tuned from an initially chosen value. The purpose of these models identifies the operation point of the environment. Corresponding to each of this identification model is a parameterized with a controller, whose parameter vector is chosen such that the corresponding controller achieves the control objective for each identification model. At each iteration and time step, one of the identification-controller models is selected based on switching control rule. The control problem is to determine suitable switching control rule and tuning these parameters to yield the

best performance for the given objective while assuring stability. The architecture applies to both linear and nonlinear systems. It is illustrated in figure 1-10.

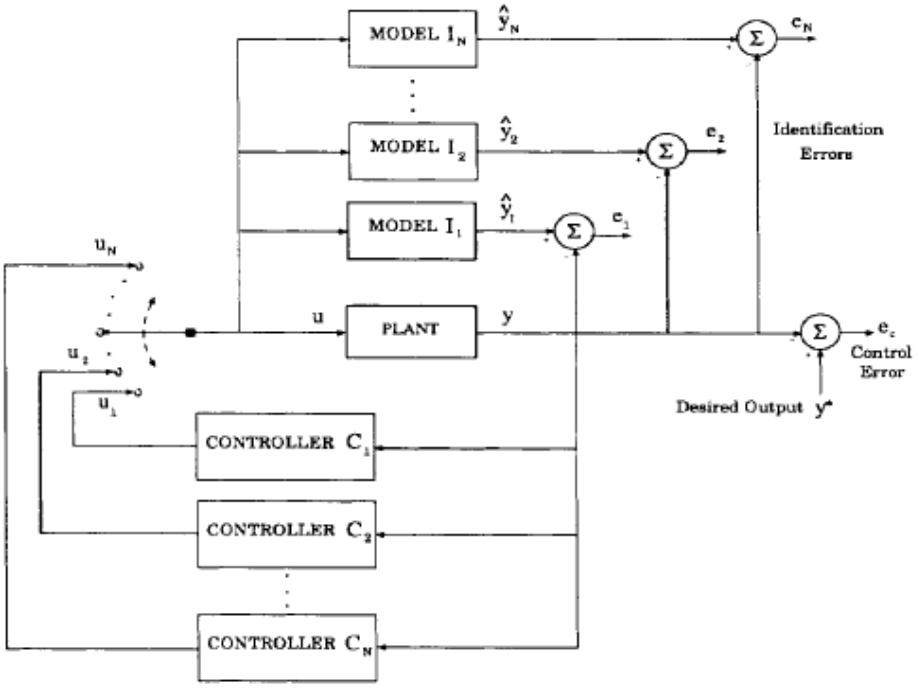

Figure 2-10: multiple -model architecture

Narendra proposed that the switching rule is to be determined by evaluating performance cost indexes for each controller and switch to the one with the minimum index at every time step. The performance of any candidate controller can be evaluated after each utilization of the model. On the other hand, the performance of all the identification-controller models are evaluated in parallel at every time step. Therefore, the indexes must be based on the performance of the identification-controller models rather than the controllers. The proposed performance index proposed as follows:

$$J_i(t) = \alpha e_j^2(t) + \beta \int_0^t e^{-\lambda(t-\tau)} e_j^2(\tau) d\tau, \alpha \geq 0, \beta, \lambda > 0 \qquad (2.4)$$

where α and β can be chosen to yield a desired combination of instantaneous and long-term accuracy measures. The forgetting factor λ determines the memory of the index in rapidly switching environments and ensures boundedness of $J_i(t)$ for bounded error e_j.

Kenji Doya et al[21] and based on the work of Suri and Schultz propose a modular reinforcement learning architecture for nonlinear, non-stationary control tasks, which is called Multiple Model-based Reinforcement Learning (MMRL). The basic idea is to decompose a complex task into multiple domains in a spatial and timely manner based on the environmental dynamics operating points. The utilize the use of a predictor of the environment. The Value of each sub-model is calculated using standard value estimator. The system is composed of multiple modules, each of which consists of a state prediction model and a reinforcement learning controller. The switching between the model is based on responsibility signal λ_i of each model, which is given by the softmax function of the prediction errors. Then the responsibility signal is assigned to each predictive model depending on the likelihood of the current observed state and the reliability of the past predictions. Finally, it is used to weight the outputs of predictive modules, to gate the learning of the prediction models, to weight the action outputs and to gate update of the reinforcement learning controllers. The overall system is illustrated in figure 1-11.

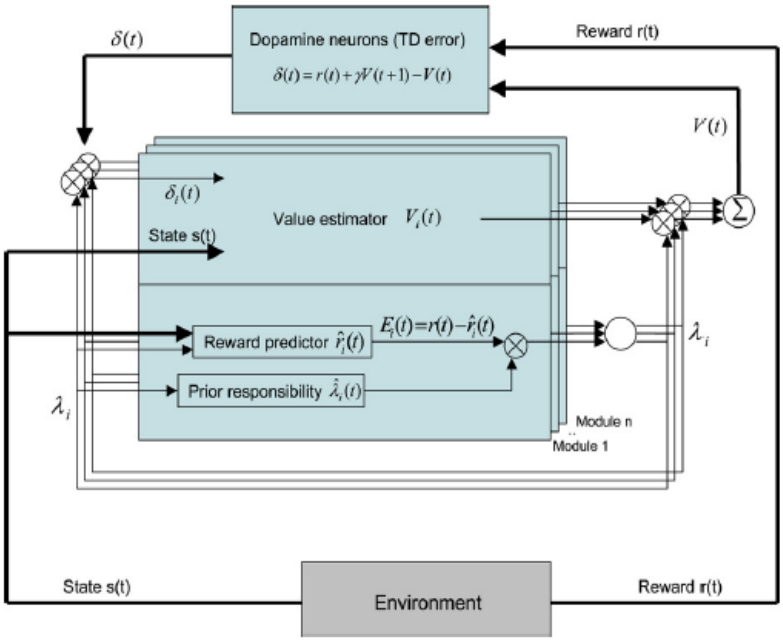

Figure 2-11: Multiple Model-based Reinforcement Learning model

Chapter 3 Optimal Control Using Multiple Adaptive Resonance Theory and Q-Learning

3.1 Introduction

The tracking control problem has gained significant attention in the control system community, due to its numerous applications. The objective is to design a control law to ensure stability of the control systems, as well as tracking a desired reference trajectory in an optimal fashion, by minimizing a performance function. Traditional solutions to the optimal tracking problem aims at finding two components [22], namely, a feedback term obtained by solving a Hamilton-Jacobi-Bellman (HJB) equation and a feedforward term obtained by solving a noncausal difference equation [23][24]. The feedback and feedforward terms are mainly found separately, and the solution is commonly obtained in an offline fashion, which requires complete knowledge of the system dynamics. Reinforcement learning (RL)[25][28], as a class of machine learning methods, has been widely used to find the online solution to the optimal tracking problem of time-invariant discrete-time systems. These methods mainly use neural networks (NNs) to approximate the value function and consequently find the optimal control solution. This application of NNs essentially extends traditional adaptive control capabilities to more advanced optimal adaptive learning feedback controllers. There is a large gap between such neural adaptive learning feedback controllers and the manner in which the human brain functions. In neuro-cognitive psychology, it is observed that the human makes quick decisions based on association of existing external variables or cues with responses learned and stored based on previous experiences [29][30]. If there is a match between current observed circumstances and previously stored responses, the human executes the previously stored response that most closely corresponds to current observations. Otherwise, the human generates a new response to the new conditions.

Likewise, in many controls applications, the system operates in different environments, with each environment requiring a dynamic description of the dynamics. Such is the case in multiple-model adaptive control [31][32], fault tolerant control, and elsewhere. In these applications, neural networks must provide high-level functions such as classification, clustering, and so on. There are similarities between such applications in different environments and the operation of the brain in using previously stored responses.

Learning networks can be used to encapsulate previous experiences into categories of stored responses that correspond to system response in different environments. Motivated by the sensory information handling in parts of the cerebral cortex in the human brain, numerous methods of data clustering have been developed to match current experiences to previously stored experiences [29][29]. In self-organized clustering, the categories are determined automatically based on various criteria for determining similarity, and new categories may be created if current data does not match with previous experience[30]. In k-mean clustering[34][35], the goal is to partition the inputs into a predetermined number k of clusters. In the self-organizing map [36][38], previously experienced data is stored as representative categories in the interconnection weights of a neural network, which is trained to produce a low-dimensional discretized mapping of the input space to achieve clustering with dimensionality reduction. Such methods of self-organized clustering are subject to unstable categorization when the distances between categories are too large, and to temporal instability in stored memory when categories are updated and do not retain their distinct characteristics. Methods for adding or resetting categories based on new information in the incoming data and for pruning or removing categories that are not used are generally ad hoc and do not have performance guarantees. An adaptive self-organizing map [23]-[38] was applied to feedback control applications in by sorting observed data into previously defined categories, within each of which a feedback controller based on prior experiences is stored. Nevertheless, this method can exhibit temporal instability in stored memory since representatives of several categories

can be tuned simultaneously. Therefore, improved methods of self-organized clustering [38] are needed that have more stable categories and temporal behaviors for applications in automatic feedback control for different environments.

Adaptive Resonance Theory (ART) is a match-based clustering approach that provides an explanation of human cognitive information processing [37]. It represents several NN models which use supervised and unsupervised learning methods to address problems such as pattern recognition and prediction in the human brain. The basic concept is to categorize the input data into categories, based on a vigilance criterion. Once some categories are established, the new input data is matched with existing categories, which is called the internal memory of an active code. ART matching leads either to a resonant state or a rest state. This matching state is established if the vigilance criterion is met. If the resonance is not achieved, the learning process takes place and an alternative category search is to be established. If the search ends, i.e. no resonance with all the categories, a new category, active code, is created. This match-based learning process is the foundation of ART code stability.

In this chapter, motivated by recent neurocognitive models of mechanisms in the brain, a model-free Q-learning based algorithm is presented to find the optimal solution to the tracking problem of time-varying discrete-time systems. The proposed algorithm combines RL with the ART algorithm to monitor changes in the dynamics of the environment. The system starts with several sub-models based on a prior knowledge. A new sub-model is then added due to the vigilance once a mismatch, no resonance, is established in ART. The ART responsibility signal indicates the likelihood of the current input belonging to the existing sub-models in the time space. Based on the presented Q-function, an optimal control for each subsystem is found online using only measured data along the system trajectories.

This chapter is organized as follows a new formulation for the optimal tracking problem of time-varying discrete-time systems using Q-learning is presented in Section 2-2. Adaptive resonance theory is presented in Section 2-3. Simulation results of the mentioned algorithms are presented in Section 2-4.

3.2 Optimal Tracking Control Problem

Consider the linear system dynamics as

$$x_{k+1} = A_j x_k + B_j u_k \tag{3.1}$$

where $x_k \in \mathbb{R}^n$ is state vector of the system, and $u_k \in \mathbb{R}^m$ is control input. It is assumed that the system operates in multiple environments which may change abruptly from one sub-model to another. That is, $(A_j, B_j) \subset M = \{(A_1, B_1), (A_2, B_2), \ldots, (A_N, B_N)\}$ with N is the number of sub-models which is generally unknown.

The goal is to design the control input u_k to assure that the states of the system track the reference trajectory r_k in an optimal manner by minimizing a predefined performance function as

$$J(x_k, r_k) = \sum_{i=k}^{\infty} \gamma^{i-k} \left[(x_i - r_i)^T S (x_i - r_i) + u_i^T R u_i \right] \tag{3.2}$$

with $S \geq 0$ and $R = R^T > 0$. $0 < \gamma \leq 1$ is a discount factor.

The desired reference trajectory is defined as

$$r_{k+1} = F r_k \tag{3.3}$$

with $r_k \in \mathbb{R}^n$.

The rest of this section assumes there is only one sub-model. This assumption is relaxed in the next section. Based on the system dynamics (3.1) and the reference trajectory dynamics(3.3), construct the augmented system as

$$X_{k+1} = \begin{bmatrix} x_{k+1} \\ r_{k+1} \end{bmatrix} = \begin{bmatrix} A_j & \mathbf{0} \\ \mathbf{0} & F \end{bmatrix} \begin{bmatrix} x_k \\ r_k \end{bmatrix} + \begin{bmatrix} B_j \\ \mathbf{0} \end{bmatrix} u_k \equiv T_j X_k + B_{1j} u_k \tag{3.4}$$

where the augmented state is $X_k = \begin{bmatrix} x_k^T & r_k^T \end{bmatrix}^T$.

The performance function (3.2) in terms of the state of the augmented system for the sub-model j can be written as

$$\mathrm{V}_j(X_k) = \sum_{i=k}^{\infty} \gamma^{i-k} \left[X_i^T \bar{S}_j X_i + u_i^T R_j u_i \right] \tag{3.5}$$

where

$$\bar{S}_j = \begin{bmatrix} S_j & -S_j \\ -S_j & S_j \end{bmatrix}$$

It is shown in [18] that the value function is quadratic in terms of the states of the system (3.5) as

$$V_j(X_k) = X_k^T P_j X_k \tag{3.6}$$

Substituting (3.5) into (3.6) yields the Bellman equation corresponding to $j-th$ sub-model as

$$X_k^T P_j X_k = X_k^T \bar{S}_j X_k + u_k^T R_j u_k + \gamma X_{k+1}^T P_j X_{k+1} \tag{3.7}$$

Then, the optimal control input corresponding to sub-model j is given as

$$u_k^* = -\gamma (R_j + \gamma B_{1j}^T P_j B_{1j})^{-1} B_{1j}^T P_j T_j X_k \tag{3.8}$$

where P_j is obtained by solving the following algebraic Riccati equation (ARE)

$$\bar{S}_j - P_j + \gamma T_j^T P_j T_j - \gamma^2 T_j^T P_j B_{1j} (R_j + \gamma B_{1j}^T P_j B_{1j})^{-1} B_{1j}^T P_j T_j = 0 \tag{3.9}$$

Eq. (2.9) is the optimal control input when one has just one model to show the behavior of the system. However, the time-varying systems have different dynamics for each kind of fault or change in

the environment, each type of these dynamics is a sub-model and we should take into account all these sub-models in the value function and control input. In the next section, adaptive resonance theory (ART) for self-organized clustering is proposed to determine the contribution of each sub-model in the general value function using a signal extracted by ART. The overall system is highlighted in Figure.2.1.

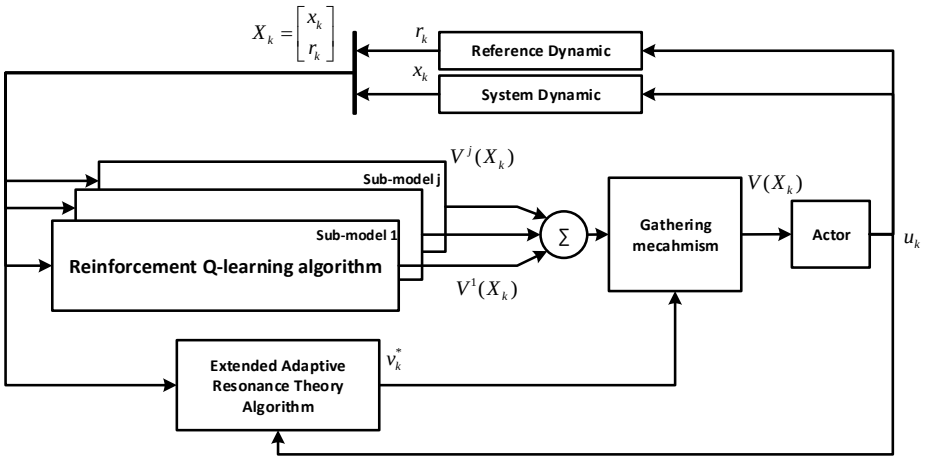

Figure 3-1: Overall System

3.3 Adaptive Resonance Theory and Value Function for Multiple-model Systems

In this section, adaptive resonance theory (ART) is used to approximate value function for multiple-model systems which are used to model uncertain time-varying systems. The fundamentals of ART are first presented, and it then is combined with multiple model and Q-learning to approximate the optimal value function and consequently the optimal solution for uncertain time-varying systems.

3.3.1 Adaptive Resonance Theory

Adaptive resonance theory (ART) remedies many of these defects noted above by employing two maps between two spaces [30][33][40]. An input data space F1 is called the short-term memory, and a category feature space F2 is called the long-term memory. One map is from the input data space F1 to the category feature space F2. This is termed the bottom-up map. A second dual map is from the category

feature space F2 back to the input space F1, known as the top-down map. The process of matching the observed data to a stored category occurs as a result of the interaction of these two maps. The model postulates that 'top-down' expectations mapped back to the input space take the form of a category prototype which is then compared, using certain metrics, with the actual input data as detected. When the difference between the actual input and the category prototype does not exceed a threshold called the vigilance parameter, the input is considered a member of the expected class. ART thus offers a solution to the plasticity versus stability paradox [40][41]. Furthermore, in the ART network, pruning and resetting of categories is furthermore formally defined using certain parameters. Furthermore, the number of categories is not predefined. There are many different forms of the ART theory which utilize different methods or metrics for determining the match between the input signal and the category prototype.

These notions are captured mathematically as follows. The complete ART algorithm is detailed in Table 2.1. The situation of our concern is when the observed input data for the ART are the inputs and outputs of a dynamical system (3.1). The input data for the ART are defined as the vector of past controls and states

$$d_k = [x_k^T \quad x_{k-1}^T \quad \dots \quad \dots \quad x_{k-\delta_x}^T \mid \quad u_{k-1}^T \quad u_{k-2}^T \quad \dots \quad u_{k-\delta_u}^T]^T \tag{3.10}$$

with δ_x, δ_u determined by the user based on experience. These input data are viewed as residing in the F1 layer, and a mapping to the category feature space, or F2 layer, is provided by

$$v_k = \frac{W^T d_k}{\left\| W^T d_k \right\|} = \left[v_k^1 \quad \dots \quad v_k^j \quad \dots \quad v_k^N \right]^T \tag{3.11}$$

where matrix W is interpreted as the weight matrix of a neural network. Define $W = \left[W_1 \quad W_2 \quad \cdots \quad W_J \right]$ where the columns W_j, $j = 1,...,N$ are the category representatives currently stored in the ART network. As such, (3.12) is an inner product that compares the input data d_k to each

category column of W, with elements v_k^j larger for categories W_j that are closer to d_k in Euclidean norm.

The next step in ART is to use competitive learning to choose a winning category to which v_k most closely belongs. To accomplish this, sort the entries v_k^j in F2 in descending order of magnitude to define the ordered list $j_1, j_2, ..., j_N$. Define $j^* = j_1$, the largest index as the chosen category F2 winner, Define the F2 vector v_k^* as a vector of zeros with an entry of 1 in position $j^* = j_1$. Define the dual map from F2 to F1 as

$$d_k^* = Wv_k^* \tag{3.12}$$

Then, $d_k^* = W_{j^*}$ is the expected or hypothesized category prototype in F1 space. That is, d_k^* is the ideal data signal that would produce category representative W_{j^*} using the map (3.11).

The key step in ART now occurs, namely matching the input data to a stored category. To accomplish this, compare the input data to the category prototype $d_k^* = W_{j^*}$ in F1 space. Many norms have been proposed for this matching test, including techniques from fuzzy logic. We use simply the Euclidean norm condition

$$\left\| d_k - d_k^* \right\| < \nu \tag{3.13}$$

where ν is known as the vigilance parameter, specified by the user. If this condition is satisfied, then column $W_{j^*} = W_{j_1}$ is declared the winning category and resonance is said to occur. Then, the weights in column W_{j^*} are updated to more closely fit the observed input data. This is accomplished by using the adaptive learning algorithm

$$W_{k+1}^{j^*} = \alpha d_k + (1-\alpha) W_k^{j^*} \tag{3.14}$$

To avoid unstable categorization and temporal instability in the stored categories, only one step of this update is performed. Note that non-winning category representatives are not updated. This preserves stability of categorization in ART. If condition (3.13) does not hold, then no winning category is declared, and the next closest category to v_k is selected. That is, one sets the trial value $j^* = j_2$. Then the map from F2 to F1 (3.12) and the matching test (3.13) are repeated. If match occurs, then column $W_{j^*} = W_{j_2}$ is declared the winning category and resonance is said to occur. Then, the weights in column W_{j^*} are updated by one step of adaptive learning algorithm (3.14). Again, if condition (3.13) does not hold, then no winning category is declared, and the next closest category to v_k is selected, namely $j^* = j_3$. The steps (3.12), and (3.13) are repeated until a winning category is found. If no winning category is found, then a new category is declared, and the input data itself is added as the last column of NN weight matrix W, so that the number of categories increases to $J+1$. Thus, adding new categories is automatic in ART, whereas in other clustering methods, it is usually ad hoc.

In summary, the ART network consists of four stages. First, the preprocessing stage wherein the input data d_k is mapped to a vector v_k in category space F2. Second, is the choice stage of selecting a winning category in F2, namely j^*, which is the category to which the data most closely belongs. Third, the match stage, where the winning category is mapped back to F1 to obtain the hypothesized category

Table.2.1. Extended ART Algorithm

	Select λ, ρ, $0 \le \kappa \le J$,
	Initialize $k=0$, x_0, $\tau_0 = 0$, $f_0 = \lambda$
(1)	$k = k+1$
	$x_{k+1} = Ax_k + Bu_k \quad x_k \in \mathbb{R}^n$
	Form d_k $d_k \in \mathbb{R}^{n(\delta x+1)m\delta u}$ $d_k = \dfrac{d_k}{\|d_k\|}$
	$\tau_{k+1} = \max(\tau_k - 1, 0)$, Where τ Is Refractory Index
	$f_{k+1} = \max(f_k - 1, 0)$, And f Fading Index
	If $f_j(k) = 0$ Remove Column W_j
	$v_k = \dfrac{W^T d_k}{\|W^T d_k\|} = \begin{bmatrix} v_1(k) & \dots & v_j(k) & \dots & v_J(k) \end{bmatrix}^T$
	Order $v_j(k)$ In Descending Order 1 To -1, Call the Ordered Indices $j_1, j_2 \dots j_J$
	Set $l = 1$
(2)	Set $j^* = j_l$
	Define $v(k)^* = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T$ As A Zero Vector With 1 In Position j^*
	If $\tau_{j^*}(k) = 0$ Then $d_k^* = W_{j^*}$
	Else Go To (3)
	If $\|d_k - d^*_k\| < \nu$ Then
	Train W_{j^*} As ($W_{j^*}(k+1) = \alpha d_k + (1-\alpha) W_{j^*}(k)$) Set $f_{j^*}(k+1) = \lambda$
	Set $\tau_{j_i}(k+1) = \rho$, $\forall j_i < l$ Start Refractory
	Else Go To (3)
(3)	Set $l = l+1$
	If $l > \kappa$ Go To (2)
	Set $W_{J+1} = d(k)$
	Go To (1)

prototype d_k^*, which is compared to the current observed data d_k. If there is a match between and d_k and d_k^*, resonance is said to occur. Then, in a fourth adaptation stage, the winning category representative is update to more closely match the current observed data.

The complete ART algorithm is detailed in Table 1. Several details remain about the functioning of ART. The first is the specification of how many categories to try before the condition of 'no winning category' is declared. This number is called κ in Table 2.1. Next, if a category is selected as the winning category $j^* = j_l$ in F2, and if this choice fails the match test (3.13) in F1, then a refractory time period ρ is initiated for the category j_l and it cannot be used again until the refractory period is over. This is kept track using a refractory index $\tau_{j_l}(k)$ in Table 2.1. Finally, ART includes a formal method for deleting categories that are never used. This is accomplished using a fading index *J*-vector f_k in Table 1, where the j-th entry of f_k is the fading index for category j. If category j is not used during the fading period λ, then category column W_j of the NN weight matrix is removed, and the number of categories J is set to $J-1$

3.3.2 New Value Function Structure Using ART

In this subsection, a general value function approximation for multiple-model linear systems using ART is presented. In the proposed value function approximation scheme, each sub-model contributes to the value function using a responsibility signal. In fact, the general value function is given by

$$V(X_k) = \sum_{j=1}^{N} v_k^j V_j(X_k) = \sum_{j=1}^{N} v_k^j X_k^T P_j X_k \tag{3.15}$$

where $v_k^j \quad j = 1,...,N$ are the responsibility signals which determine the contribution of each sub-model to the general value function.

Considering (3.4) and (3.15) in (3.2), yields the Bellman equation for time-varying systems

$$\sum_{j=1}^{N} v_k^j X_k^T P_j X_k = X_k^T \bar{S} X_k + u_k^T R u_k + \gamma \sum_{j=1}^{N} v_k^j X_{k+1}^T P_j X_{k+1} \tag{3.16}$$

and the Hamiltonian is defined as

$$H(X_k, u_k) = X_k^T \bar{S} X_k + u_k^T R u_k + \gamma \sum_{j=1}^{N} v_k^j X_{k+1}^T P_j X_{k+1} - \sum_{j=1}^{N} v_k^j X_k^T P_j X_k \tag{3.17}$$

Applying the stationarity condition $\partial H(X_k, u_k)/\partial u_k = 0$ yields the optimal control input as

$$u_k^* = \gamma \left(\sum_{j=1}^{J} v_k^j (R + \gamma B_{1j}^T P_j B_{1j}) \right)^{-1} \left(\sum_{j=1}^{J} v_k^j B_{1j}^T P_j T_j \right) X_k \tag{3.18}$$

where $P_j \quad j = 1,...N$ are obtained by solving a set of AREs (3.9).

Remark 1. Note that complete knowledge about the augmented system dynamics is required to find the optimal control input (3.18). In the next section, reinforcement learning is used to find the solution to the optimal tracking problem without requiring any knowledge about the system dynamics.

3.3.3 Q-learning to Solve Optimal Tracking Problem of Multiple-model Systems

The solution to the optimal multiple-model tracking control problem needs complete knowledge about the system dynamics and reference trajectory dynamics. In this section a Q-learning algorithm is developed that solves this problem online without requiring any knowledge of the augmented system dynamics.

Based on the Bellman equation (3.7), the discrete-time Q-function for j-th sub-system is defined as

$$Q_j(k) = X_k{}^T \bar{S}_j X_k + u_k{}^T R_j u_k + \gamma X_{k+1}{}^T P_j X_{k+1} \tag{3.19}$$

Substituting the augmented system (3.4) in (3.19) yields,

$$
\begin{aligned}
Q_j(X_k,u_k) &= X_k{}^T\bar{S}_j X_k + u_k{}^T R_j u_k + \gamma(T_i X_k + \mathrm{B}_1 u_k)^T P_j (T_i X_k + \mathrm{B}_1 u_k) \\
&= \begin{bmatrix} X_k \\ u_k \end{bmatrix}^T \begin{bmatrix} \bar{S}_j + \gamma T_j^T P_j T_j & \gamma T_j^T P_j B_1 \\ \gamma B_1^T P_j T_j & R_j + \gamma B_1^T P_j B_1 \end{bmatrix} \begin{bmatrix} X_k \\ u_k \end{bmatrix} \\
&= \begin{bmatrix} X_k \\ u_k \end{bmatrix}^T \begin{bmatrix} H_j^{xx} & H_j^{xu} \\ H_j^{ux} & H_j^{uu} \end{bmatrix} \begin{bmatrix} X_k \\ u_k \end{bmatrix} \\
&= Z_k{}^T H_j Z_k
\end{aligned}
\tag{3.20}
$$

For the multiple-model systems, the general Q function is defined as

$$
Q(k) = \sum_{j=1}^{N} v_k^j Q_j(X_k) \tag{3.21}
$$

By substituting the quadratic form (3.20) in (3.21), one has

$$
\begin{aligned}
Q(k) &= \sum_{j=1}^{N} v_k^j Q_j(X_k) = v_{k1}^1 Z_k^T H_1 Z_k + v_k^2 Z_k^T H_2 Z_k + \ldots + v_k^N Z_k^T H_J Z_k \\
&= Z_k^T (\sum_{j=1}^{N} v_k^j H_j) Z_k \\
&= Z_k^T \begin{bmatrix} \sum_{j=1}^{N} v_k^j H_j^{xx} & \sum_{j=1}^{N} v_k^j H_j^{xu} \\ \sum_{j=1}^{N} v_k^j H_j^{ux} & \sum_{j=1}^{N} v_k^j H_j^{uu} \end{bmatrix} Z_k \\
&= Z_k^T \mathrm{H} Z_k
\end{aligned}
\tag{3.22}
$$

Eq. (3.22) shows that the general Q-function for multiple-model systems is quadratic in terms of the states of the augmented system and control input.

Applying the stationarity condition $dQ(k)/du_k = 0$ yields,

$$
u_k^* = \left(\sum_{j=1}^{N} v_k^j H_j^{uu} \right)^{-1} \left(\sum_{j=1}^{N} v_k^j H_j^{ux} \right) X(k) \tag{3.23}
$$

Now, we can present a Q-learning algorithm to solve the optimal tracking control problem of multiple-model systems online without knowing the augmented system dynamics (T_j, B_{1j}).

The Bellman equation (3.16) in terms of Q-function is given as

$$Q(X_k, u_k) = X_k^T \bar{S} X_k + u_k^T R u_k + \gamma Q(X_{k+1}, u_{k+1}) \tag{3.24}$$

Substituting (3.22) into (3.24), the Q-function Bellman equation (3.24) becomes

$$Z_k^T \mathrm{H} Z_k = X_k^T S_1 X_k + u_k^T R u_k + \gamma Z_{k+1}^T \mathrm{H} Z_{k+1} \tag{3.25}$$

Policy iteration is especially easy to implement in terms of the Q-function, as follows.

Algorithm 1. Policy Iteration using Q-function

1. Policy evaluation

$$Z_k^T \mathrm{H}^{i+1} Z_k = X_k^T \bar{S} X_k + (u_k^i)^T R (u_k^i) + \gamma Z_{k+1}^T \mathrm{H}^{i+1} Z_{k+1} \tag{3.26}$$

2. Policy improvement

$$u_k^{i+1} = \left(\sum_{j=1}^{N} v_k^j (\mathrm{H}_j^{uu})^{i+1} \right)^{-1} \left(\sum_{j=1}^{N} v_k^j (\mathrm{H}_j^{ux})^{i+1} \right) X(k) \tag{3.27}$$

Q-Learning attempts to learn the cost of the current category state and taking a specific action toward minimizing the performance index. The advantage of Q-learning is that convergence guarantees can be given even when function approximation is used to estimate the action values.

3.4 Simulation

To show the effectiveness of the proposed method, simulations have been carried out on a mass-spring-damper system. The system dynamics is

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + x_{2,k} \\ x_{2,k+1} &= -\frac{k}{m} x_{1,k} + (1 - \frac{b}{m}) x_{2,k} + \frac{1}{m} u_k \end{aligned} \tag{3.28}$$

Three different system behaviors are considered for this simulation. The parameters set of each time variant period are provided in Table 2.2. These parameters change the behavior system dynamics (3.28). For each time interval a system is activated for the corresponding parameters.

Table 2.2: The parameters of three system dynamics for three-time intervals

Time Interval	System Parameters
$0 < t \leq 300$	$k_1 = 10 \quad b_1 = 10 \quad m_1 = 90.$
$300 < t \leq 600$	$k_2 = 30 \quad b_2 = 15 \quad m_2 = 90$
$600 < t \leq 1000$	$k_3 = 50 \quad b_3 = 50 \quad m_3 = 90.$

The extended ART parameters are chosen as $\lambda = 100$, $\rho = 0.78$, $\kappa = 20$, and $\delta_x = \delta_u = 4$. Figure.2-2 shows the norm of the difference between the optimal control gain and the computed gain. It is obvious from the figure that the gain converges to the optimal value. Spikes is seen in the first iterations with system 1, a smaller spike is seen when the dynamic changing at 300 points, and tiny spike at 600. This because the extended ART sub-models are carried through the three systems, and the changing in the system dynamic is not very huge. The optimal gains and the computed gain are shown in table 2.3.

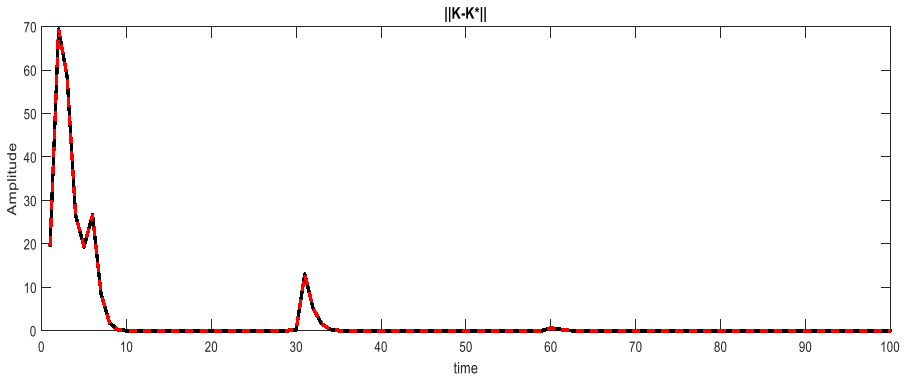

Figure 3-2 The norm of the difference between the optimal control gain and the computed gain

Table 2.3: The optimal gains vs the Computed gains

	Optimal Gains	Computed Gains
System 1	$K^*_{Sys1} = \begin{bmatrix} -4.292 & 23.686 \end{bmatrix}$	$K_{Sys1} = \begin{bmatrix} -4.292 & 23.686 \end{bmatrix}$
System 2	$K^*_{Sys2} = \begin{bmatrix} -0.127 & 3.019 \end{bmatrix}$	$K_{Sys2} = \begin{bmatrix} -0.127 & 3.019 \end{bmatrix}$
System 3	$K^*_{Sys3} = \begin{bmatrix} -0.489 & 1.297 \end{bmatrix}$	$K_{Sys3} = \begin{bmatrix} -0.489 & 1.297 \end{bmatrix}$

3.5 Conclusion

In this chapter, ART clustering algorithm is combined with RL to find the optimal solution to the tracking problem of time-varying discrete-time systems. The changes in the system behavior is taken into account using multiple-model approach. ART algorithm generates sub-models based on the clustering match-based method. A Q-learning based algorithm is then used to find the optimal solution online and without requiring any knowledge of the system dynamics. Each sub-model contributes into Q-function through a responsibility signal generated by ART.

Chapter 4 An Incremental Optimal Q-Learning Model for Biologically inspired dopamine-like reinforcement signal for a spatial delayed response task

4.1 Introduction

Optimal control is set to find a control law for a given dynamical system such that a certain optimality criterion is achieved. A control problem includes a performance function that is a function of state and control variables. Optimal control is a set of differential equations describing the paths of the control variables that minimize the performance function. The Optimal tracking problem of nonlinear systems has always been the key focus in the control field in the last several decades. The objective is to design a control law to ensure the stability of the control systems, as well as optimally tracking a desired reference trajectory, by minimizing a performance function. Traditional solutions to the optimal tracking problem aims at finding two components [22] , namely, a feedback term obtained by solving a Hamilton-Jacobi-Bellman (HJB) equation and a feedforward term obtained by solving a noncausal difference equation [23][24]. The feedback and feedforward terms are mainly found separately, and the solution is commonly obtained in an offline fashion, which requires complete knowledge of the system dynamics. Reinforcement learning (RL) [25][28], as a class of machine learning methods, has been widely used to find the online solution to the optimal tracking problem of time-invariant discrete-time systems.

These methods mainly use a function approximation scheme to approximate the value function and consequently find the optimal control solution. Function approximation methods employ the different problems such as regression, classification, fitness approximation which work to a certain degree and received unified treatment as supervised learning problems. However, there is a trend for the application of NNs essentially extends traditional adaptive control capabilities to more advanced optimal

adaptive learning feedback controllers. This trend focuses on two different aspects stability and plasticity. The area of studying these types of networks has a relative shortage. The shortage is generated because of the divergence between cognitive brain discovery and the computational model. The huge advancement in computational power field yield very decent function approximator which focuses on optimality, and stability but not plasticity. This spark some criticism from the cognitive, and neuroscientist on such learning method. The human makes quick decisions based on association of existing external variables or cues with responses learned and stored based on previous experiences [29][30]. This induced a big issue in the existing application of reinforcement learning (RL) to real-world control problems is how to deal with nonlinearity and nonstationarity. For a nonlinear, high-dimensional system, the conventional discretizing approach necessitates a huge number of states, which makes learning very slow. Standard RL algorithms can perform badly when the environment is nonstationary or has hidden states. These problems have motivated the introduction of modular or hierarchical RL architectures. The main theme to solve the curse of the dimensionality model is to use one of two methods One approach to this "curse of dimensionality" is to "divide and conquer" as in the renowned work of bellman [18] or the cognitive - neuroscientific work of Ghahramani [19] and Doya [20] subdividing a complicated problem into simpler subproblems. To simplify this consider a subject in a visuomotor task might realize that rewards depend solely on eye movements, whereas other tasks reward multiple effectors independently, like driving while talking on the phone. Work in computational RL [21-22]. However, these models also considered stability and optimality. Other neuroscience model considered only modeling the brain functions in such a situation, namely consider the dopamine receptor, and synapses.

It is needless to say that all these models use variations of temporal difference model. TD models are more efficient than most conventional reinforcement models in learning a wide variety of behavioral tasks, from balancing a pole on a cartwheel [23] to playing world class backgammon [24] Robots using TD algorithms learn to move about two-dimensional space and avoid obstacles or reach and grasp[25]. In

biological applications, TD models replicate the foraging behavior of honeybees,[26] simulate human decision making [27] and learn eye movements [28-29]. In fact, the dopaminic association is, in essence, some form of TD learning.

Furthermore, most models lack the consideration of the temporal difference of the reward. In neuroscience short-latency response of reward, dopamine neurons, to primary rewards and conditioned. It lacks the consideration of reward-predicting signal might constitute a biological implementation of a TD reinforcement signal. In addition, most models do not take into the account the negative reward. In biology, this occurs when the prediction of the reward is high, and the reward is omitted. Similar to the Effective Reinforcement Signal, activations of reward, dopamine neurons occur only following rewards that are unpredictable and not following fully predicted rewards. Depressions occur when a predicted reward is omitted. This short-latency response is largely restricted to appetitive prediction error and rarely occurs with surprise or predicated aversive reward. In addition, most models consider the learning in the critic, but rarely in the actor. Suri and Schultz showed that the learning in reinforcement situation occurs in both the critic as the actor**Error! Reference source not found.** . Cognitive RL theories hypothesize that learning is driven by a "prediction error" (PE), typically assumed to be a single signal broadcast by dopaminergic projections [30-32]. This single signal has a striking similarity with the computational work of Kalman in prediction covariance matrix in Kalman filtering technique.

Suri and Schultz proposed the celebrated model of the dopamine reward circuitry in the brain which has some explanation for all the above problem. The model bridges the gap between the computational models and the neuroscientific models. The model consists of an Actor component and a Critic component. Actor and Critic receive input stimulus from all the models which are coded as functions of time, respectively. The critic computes the Effective Reinforcement Signal which serves to modify the critic learning weights and the actor weights mimicking adaptive synapses. The function of the Critic is to associate the input stimulus with the Effective Reinforcement Signal. Every stimulus is represented as a

series of time-spatial components of different durations. Each of these components influences the reward prediction signal according to its adaptive weight. This form of temporal stimulus representation allows the Critic to learn the correct duration of the stimulus-reward interval. For every stimulus, a specific prediction is computed as the weighted sum of the representational components. A winner-take-all competition between predictions of different stimuli sets all representational components to zero except the strongest one. The change in the prediction of a stimulus is computed by taking the temporal difference between successive time steps of the previous prediction the current prediction with a discounting factor. The temporal difference between successive predictions is summed over all stimuli and added to the primary reinforcement signal input coding the reward. The result of this summation is the Effective Reinforcement Signal which codes the error in the prediction of reward. The Actor learns to associate stimuli with behavioral actions. A winner-take-all rule prevents the Actor from performing two actions at the same time. Notice that the learning in the critic is linear combination of all evaluated reward, and the learning in the critic is a winner-take-all scheme.

In this paper, motivated by neurocognitive models of mechanisms described in Suri Schultz model in the brain, a model-free Q-learning based algorithm is presented to find the optimal solution to the tracking problem of time-varying discrete-time systems. The proposed algorithm combines RL with the Suri Schultz to monitor changes in the dynamics of the environment. The system starts with a fixed number of sub-models based on prior knowledge. The Suri Schultz effective reward signal contribution of each critic to the overall learning, and the contribution of each actor in the reduction in the time-space. Based on the presented Q-function, an optimal control for each subsystem is found online using only measured data along the system trajectories.

This paper is organized as follows. Q-learning method is highlighted in Section 2. New the Suri Schultz model is explained in Section 3. The formulation for the LQR using Q-learning is presented in Section 4. Simulation results of the mentioned algorithms are presented in Section 5.

4.2 Optimal Tracking Control Problem

Consider the linear system dynamics as

$$x_{k+1} = A_j x_k + B_j u_k \tag{4.1}$$

where $x_k \in \mathbb{R}^n$ is state vector of the system, and $u_k \in \mathbb{R}^m$ is control input. It is assumed that the system operates in multiple environments which may change abruptly from one sub-model to another. That is, $(A_j, B_j) \subset M = \{(A_1, B_1), (A_2, B_2), ..., (A_N, B_N)\}$ with N is the number of sub-models which is generally unknown.

The goal is to design the control input u_k to assure that the states of the system track the reference trajectory r_k in an optimal manner by minimizing a predefined performance function as

$$J(x_k, r_k) = \sum_{i=k}^{\infty} \gamma^{i-k} \left[(x_i - r_i)^T S (x_i - r_i) + u_i^T R u_i \right] \tag{4.2}$$

with $S \geq 0$ and $R = R^T > 0$. $0 < \gamma \leq 1$ is a discount factor.

The desired reference trajectory is defined as

$$r_{k+1} = F r_k \tag{4.3}$$

with $r_k \in \mathbb{R}^n$.

The rest of this section assumes there is only one sub-model. This assumption is relaxed in the next section. Based on the system dynamics (3.1) and the reference trajectory dynamics(3.3), construct the augmented system as

$$X_{k+1} = \begin{bmatrix} x_{k+1} \\ r_{k+1} \end{bmatrix} = \begin{bmatrix} A_j & \mathbf{0} \\ \mathbf{0} & F \end{bmatrix} \begin{bmatrix} x_k \\ r_k \end{bmatrix} + \begin{bmatrix} B_j \\ \mathbf{0} \end{bmatrix} u_k \equiv T_j X_k + B_{1j} u_k \tag{4.4}$$

where the augmented state is $X_k = \begin{bmatrix} x_k^T & r_k^T \end{bmatrix}^T$.

The performance function (3.2) in terms of the state of the augmented system for the sub-model j can be written as

$$\mathrm{V}_j(X_k) = \sum_{i=k}^{\infty} \gamma^{i-k} \left[X_i^T \bar{S}_j X_i + u_i^T R_j u_i \right] \tag{4.5}$$

where

$$\bar{S}_j = \begin{bmatrix} S_j & -S_j \\ -S_j & S_j \end{bmatrix}$$

It is shown in [18] that the value function is quadratic in terms of the states of the system (3.5) as

$$V_j(X_k) = X_k^T P_j X_k \tag{4.6}$$

Substituting (3.5) into (3.6) yields the Bellman equation corresponding to $j-th$ sub-model as

$$X_k^T P_j X_k = X_k^T \bar{S}_j X_k + u_k^T R_j u_k + \gamma X_{k+1}^T P_j X_{k+1} \tag{4.7}$$

Then, the optimal control input corresponding to sub-model j is given as

$$u_k^* = -\gamma (R_j + \gamma B_{1j}^T P_j B_{1j})^{-1} B_{1j}^T P_j T_j X_k \tag{4.8}$$

where P_j is obtained by solving the following algebraic Riccati equation (ARE)

$$\bar{S}_j - P_j + \gamma T_j^T P_j T_j - \gamma^2 T_j^T P_j B_{1j} (R_j + \gamma B_{1j}^T P_j B_{1j})^{-1} B_{1j}^T P_j T_j = 0 \tag{4.9}$$

Eq. (3.8) is the optimal control input when one has just one model to show the behavior of the system. However, the time-varying systems have different dynamics for each kind of fault or change in

the environment, each type of these dynamics is a sub-model and we should take into account all these sub-models in the value function and control input. In the Suri-Schultz learning model is proposed to determine the contribution of each sub-model , critic, and actor in the general value function. The overall system is highlighted in Figure 3.1

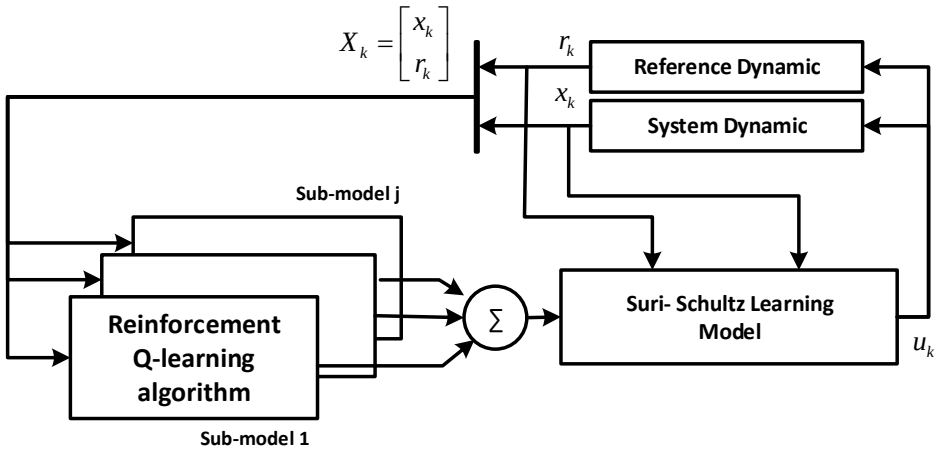

Figure 4-1 Overall system

4.3 Dopamine-like reinforcement Model Value Function for Multiple-model Systems.

In this section, Dopamine-like reinforcement Model is used to approximate value function for multiple-model systems which are used to model uncertain time-varying systems. The fundamentals of Dopamine-like reinforcement Model are first presented, and it then is combined with multiple model and Q-learning to approximate the optimal value function and consequently the optimal solution for uncertain time-varying systems.

4.3.1 Dopamine-like reinforcement Model

Suri-Schultz model [100] was originated to investigated how the simulated response of dopamine neurons to reward-related stimuli could be used as a reinforcement signal for learning a spatial delayed response task in the primate brains. This work gave the cognitive model to the computational learning

theory and probably approximately correct learning, (PAC) frameworks for mathematical analysis of machine learning. It was proposed in 1984 by Leslie Valiant [102][1][102]. Spatial delayed response tasks assess the functions of the frontal cortex and basal ganglia in short-term memory, movement preparation and expectation of environmental events. In these tasks, a stimulus appears for a short period at a particular location, and after a delay the subject moves to the location indicated. In general learning situation, this analogous to the trajectory of learning. In these frameworks, the learner receives samples and must select a generalization function, generally called the hypothesis, from a certain class of possible functions. The goal is that, with high probability, the selected function will have low generalization error. These models are more immune against learning noise as a reward. In fact, the PAC model was later extended to remedy noise (misclassified samples).

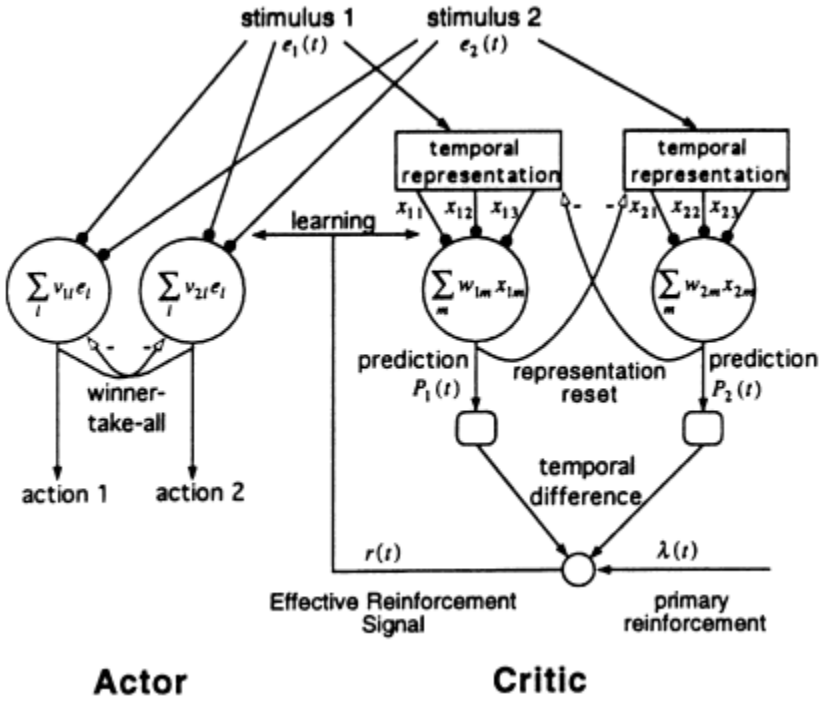

Figure 4-2: Suri-Schultz model

In Suri-Schultz model dopamine neurons, reward signals, are activated by unpredicted rewards and reward-predicting stimuli, are not influenced by fully predicted rewards, and are depressed by

omitted rewards. Thus, they appear to report an error in the prediction of reward, which is the crucial reinforcement term in formal learning theories. This scheme gives an explanation of the stability and the plasticity in the brain, however, in a computational optimal setting, the fully predicated reward should be considered. Computationally, the calculation of the reward and the prediction of the rewards yield the learning trajectory. Theoretical studies on reinforcement learning have shown that signals similar to dopamine responses can be used as effective teaching signals for learning. The reinforcement signal was modeled according to the basic characteristics of dopamine responses to novel stimuli, primary rewards and reward-predicting stimuli. A Critic component analogous to dopamine neurons computed a temporal error in the prediction of reinforcement and emitted this signal to an Actor component which mediated the behavioral output. The spatial delayed response task was learned via two subtasks introducing spatial choices and temporal delays, in the same manner as monkeys in the laboratory. In all three tasks, the reinforcement signal of the Critic developed in a similar manner to the responses of natural dopamine neurons in comparable-learning situations, and the learning curves of the Actor replicated the progress of learning observed in the animals. Omission of reward induced a phasic reduction of the reinforcement signal at the time of the reward and led to the extinction of learned actions. A reinforcement signal without prediction error resulted in impaired learning because of perseverative errors. Loss of learned behavior was seen with sustained reductions of the reinforcement signal, a situation in general comparable to the loss of dopamine innervation in Parkinsonian patients.

These notions are captured mathematically as follows. The situation of our concern is when the observed input data for the Suri-Shultz are the inputs and outputs of a dynamical system (3.1) and the reference trajectory of (3.3). The function e_k^j, describing the physical salience of one of these stimuli, took value one during presentation of the stimulus and zero otherwise, this constitute the input data for the Suri-Shultz are defined:

$$e_{k+1}^{j} = 1 - \left(\frac{x_k - r_k}{\left\| x_k - r_k \right\|} \right) \tag{4.10}$$

The stimulus is the phasic system error between the system states x_k , and the reference trajectory r_k .

The stimulus trace of a stimulus e_k^j is decaying with the parameter δ is defined as decay of trace and action trace as in (3.10). This is reignited if the error is reduced. The augmented phasic system error is normalized, and it is always 0<e<1, where 1 as no error.

$$\bar{e}_{k+1}^{j} = h(e_k^j + \delta e_{k-1}^j). \tag{4.11}$$

Where

The function *h*() limited the argument to values smaller or equal to one and was defined as

$$h(y) = \begin{cases} y & y<1 \\ 1 & \text{else} \end{cases} \tag{4.12}$$

Here we limit the system response time a period of time to ensure the repose is not overlearned or under learned. For most system include the primates three trials is time between learning and action, also this sometime is defined as the refectory signal.

$$a_k^j = \begin{cases} 1 & a_{k+1}^j \\ 0 & a_{k+3}^j \end{cases} \tag{4.13}$$

If there was no ongoing action a'^j_k, a_k^j is computed from the weights v_k^j and the stimulus trace, the augmented phasic system error $\bar{e}_k^j$.

$$a'^j_k = \left[\sum\nolimits_j v_k^j \bar{e}_k^j \right] g(e_k^j). \tag{4.14}$$

This means that the action a'^{j}_{k} of this sub- is a candidate optimal actions, for the Where the function g() is designed allowed actions only if a reward was presented

$$g(e^{j}_{k}, e^{j}_{k-1}) = \begin{cases} 1 & e^{j}_{k} - e^{j}_{k-1} > 0 \\ 0 & else \end{cases} \tag{4.15}$$

The action a'^{j}_{k} and selected using competition between actions in the form of a winner-take-all rule as follows:

$$a^{j}_{k} = \begin{cases} 1 & a'^{j}_{k} > a'^{m \neq j}_{k} \\ 0 & else \end{cases} \tag{4.16}$$

The representation of selected action was extended over time with the traces

$$a^{j}_{k} = h(a^{j}_{k} + \delta a^{j}_{k}). \tag{4.17}$$

Where the limiting function h() is defined in, and δ is describing the decay of the trace.

computation of the temporal representation x^{m}_{k} from a stimulus e^{m}_{k}. For a single stimulus e^{m}_{k}, the temporal representation x^{m}_{k} depended only on the onset of this stimulus e^{m}_{k} and not on its offset. x^{m}_{k} was defined with the recursive function,

$$X^{m}_{k} = f(e^{m}_{k}, x^{m}_{k-1}) \tag{4.18}$$

Three cases were distinguished for the definition of function *f*():

Temporal Representation 1: Onset of stimulus e^{m}_{k} elicited the first representation component

$$x^{1}_{k}(t) = e^{1}_{k} \tag{4.19}$$

Temporal Representation 2: The slower components followed one iteration:

$$x^{m}_{k} = \rho \cdot e^{m}_{k-1} \tag{4.20}$$

Where ρ is Onset activations decrease of the stimulus representation components

Temporal Representation 3: More than one iteration after the onset of stimulus e_k^m, the components:

$$x_k^m = c_k^m x_{k-1}^m / \gamma \tag{4.21}$$

Where temporal representation increased gradually with discount factor γ. This increase was chosen in order to make the time-course of the representation components proportional to the time-course of the desired prediction signal. The decays of the representation components were implemented with the function c_k^m :

$$c_k^m \begin{cases} 0 & n=\max(x_{k-1}^m) \\ & n \in 1...m \\ 1 & else \end{cases} \tag{4.22}$$

which set the largest component of x_{k-1}^m to zero.

Changes in the temporal representation were defined:

$$\Delta x_k^m = \left\lfloor x_{k-1}^m - \gamma x_k^m \right\rfloor_+ \tag{4.23}$$

$\lfloor \ \rfloor_+$ indicates that negative values of the argument were set to zero and positive values of the argument remained unchanged.

The adaptive critic weights w_k are used to associate the temporal stimulus representation x_k^m of stimulus with the reward prediction:

$$P_k^m = \sum w_k x_k^m . \tag{4.24}$$

For several temporal representation, the prediction P_k is the sum over the reward predictions associated with all temporal representation

$$P_k = \sum_m P_k^m \tag{4.25}$$

The response of dopamine neurons is modeled as the Effective Reinforcement Signal:

$$r_k = d + \lambda_k + \gamma P_k - P_{k-1} \tag{4.26}$$

where λ_k is primary reward, d is positive constant, and γ is discount factor.

The learning in the Critic weights is updated as follows:

$$w_k^m = w_{k-1}^m + \eta_c (r_k - d) \left\lfloor x_{k-1}^m - \gamma x_k^m \right\rfloor_+ \tag{4.27}$$

η_c as the learning rate. The brackets $\lfloor\ \rfloor_+$ indicate that a negative number is set to zero and a positive number remains unchanged.

The learning in the actor weight is updated as follows:

$$v_k^m = v_{k-1}^m + \eta_a (r_k - d) a_k^m e_k^m. \tag{4.28}$$

η_a as the learning rate.

4.3.2 New Value Function Structure Using Dopamine-like reinforcement Model

In this subsection, a general value function approximation for multiple-model linear systems using modified is presented. In the proposed value function approximation scheme, each sub-model contributes to the value function using a responsibility signal. In fact, the general value function is given by

$$V(X_k) = \sum_{j=1}^{N} a_k^j V_j(X_k) = \sum_{j=1}^{N} a_k^j X_k^T P_j X_k \tag{4.29}$$

where $a_k^j \quad j=1,...,N$ are the responsibility signals which determine the contribution of each sub-model to the general value function.

Considering (3.4) and (3.15) in (3.2), yields the Bellman equation for time-varying systems

$$\sum_{j=1}^{N} a_k^j X_k^T P_j X_k = X_k^T \bar{S} X_k + u_k^T R u_k + \gamma \sum_{j=1}^{N} a_k^j X_{k+1}^T P_j X_{k+1} \tag{4.30}$$

and the Hamiltonian is defined as

$$H(X_k, u_k) = X_k^T \bar{S} X_k + u_k^T R u_k + \gamma \sum_{j=1}^{N} a_k^j X_{k+1}^T P_j X_{k+1} - \sum_{j=1}^{N} a_k^j X_k^T P_j X_k \tag{4.31}$$

Applying the stationarity condition $\partial H(X_k, u_k)/\partial u_k = 0$ yields the optimal control input as

$$u_k^* = \gamma \left(\sum_{j=1}^{J} a_k^j (R + \gamma B_{1j}^T P_j B_{1j}) \right)^{-1} \left(\sum_{j=1}^{J} a_k^j B_{1j}^T P_j T_j \right) X_k \tag{4.32}$$

where $P_j \quad j=1,...N$ are obtained by solving a set of AREs (3.9).

Remark 1. Note that complete knowledge about the augmented system dynamics is required to find the optimal control input (3.18). In the next section, reinforcement learning is used to find the solution to the optimal tracking problem without requiring any knowledge about the system dynamics.

4.3.3 Q-learning to Solve Optimal Tracking Problem of Multiple-model Systems

The solution to the optimal multiple-model tracking control problem needs complete knowledge about the system dynamics and reference trajectory dynamics. In this section a Q-learning algorithm is developed that solves this problem online without requiring any knowledge of the augmented system dynamics.

Based on the Bellman equation (3.7), the discrete-time Q-function for j-th sub-system is defined as

$$Q_j(k) = X_k{}^T \bar{S}_j X_k + u_k{}^T R_j u_k + \gamma X_{k+1}{}^T P_j X_{k+1} \tag{4.33}$$

Substituting the augmented system (3.4) in (3.19) yields,

$$\begin{aligned} Q_j(X_k, u_k) &= X_k{}^T \bar{S}_j X_k + u_k{}^T R_j u_k + \gamma (T_i X_k + \mathrm{B}_1 u_k)^T P_j (T_i X_k + \mathrm{B}_1 u_k) \\ &= \begin{bmatrix} X_k \\ u_k \end{bmatrix}^T \begin{bmatrix} \bar{S}_j + \gamma T_j^T P_j T_j & \gamma T_j^T P_j B_1 \\ \gamma B_1^T P_j T_j & R_j + \gamma B_1^T P_j B_1 \end{bmatrix} \begin{bmatrix} X_k \\ u_k \end{bmatrix} \\ &= \begin{bmatrix} X_k \\ u_k \end{bmatrix}^T \begin{bmatrix} H_j^{xx} & H_j^{xu} \\ H_j^{ux} & H_j^{uu} \end{bmatrix} \begin{bmatrix} X_k \\ u_k \end{bmatrix} \\ &= Z_k{}^T H_j Z_k \end{aligned} \tag{4.34}$$

For the multiple-model systems, the general Q function is defined as

$$Q(k) = \sum_{j=1}^{N} a_k^j Q_j(X_k) \tag{4.35}$$

By substituting the quadratic form (3.20) in (3.21), one has

$$\begin{aligned} Q(k) &= \sum_{j=1}^{N} a_k^j Q_j(X_k) = a_{k1}^1 Z_k^T H_1 Z_k + a_k^2 Z_k^T H_2 Z_k + \ldots + a_k^N Z_k^T H_J Z_k \\ &= Z_k^T (\sum_{j=1}^{N} a_k^j H_j) Z_k \\ &= Z_k^T \begin{bmatrix} \sum_{j=1}^{N} a_k^j H_j^{xx} & \sum_{j=1}^{N} a_k^j H_j^{xu} \\ \sum_{j=1}^{N} a_k^j H_j^{ux} & \sum_{j=1}^{N} a_k^j H_j^{uu} \end{bmatrix} Z_k \\ &= Z_k^T \mathrm{H} Z_k \end{aligned} \tag{4.36}$$

Eq. (3.22) shows that the general Q-function for multiple-model systems is quadratic in terms of the states of the augmented system and control input.

Applying the stationarity condition $dQ(k)/du_k = 0$ yields,

$$u_k^* = \left(\sum_{j=1}^{N} a_k^j H_j^{uu} \right)^{-1} \left(\sum_{j=1}^{N} a_k^j H_j^{ux} \right) X(k) \tag{4.37}$$

Now, we can present a Q-learning algorithm to solve the optimal tracking control problem of multiple-model systems online without knowing the augmented system dynamics (T_j, B_{1j}).

The Bellman equation (3.16) in terms of Q-function is given as

$$Q(X_k, u_k) = X_k^T \bar{S} X_k + u_k^T R u_k + \gamma Q(X_{k+1}, u_{k+1}) \tag{4.38}$$

Substituting (3.22) into (3.24), the Q-function Bellman equation (3.24) becomes

$$Z_k^T \mathrm{H} Z_k = X_k^T S_1 X_k + u_k^T R u_k + \gamma Z_{k+1}^T \mathrm{H} Z_{k+1} \tag{4.39}$$

Policy iteration is especially easy to implement in terms of the Q-function, as follows.

Algorithm 1. Policy Iteration using Q-function

Policy evaluation

$$Z_k^T \mathrm{H}^{i+1} Z_k = X_k^T \bar{S} X_k + (u_k^i)^T R (u_k^i) + \gamma Z_{k+1}^T \mathrm{H}^{i+1} Z_{k+1} \tag{4.40}$$

Policy improvement

$$u_k^{i+1} = \left(\sum_{j=1}^{N} a_k^j (\mathrm{H}_j^{uu})^{i+1} \right)^{-1} \left(\sum_{j=1}^{N} a_k^j (\mathrm{H}_j^{ux})^{i+1} \right) X(k) \tag{4.41}$$

Q-Learning attempts to learn the cost of the current category state and taking a specific action toward minimizing the performance index. The advantage of Q-learning is that convergence guarantees can be given even when function approximation is used to estimate the action values.

4.4 Simulation

To show the effectiveness of the proposed method, simulations have been carried out on a mass-spring-damper system. The system dynamics is

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + x_{2,k} \\ x_{2,k+1} &= -\frac{k}{m}x_{1,k} + (1-\frac{b}{m})x_{2,k} + \frac{1}{m}u_k \end{aligned} \tag{4.42}$$

Three different system behaviors are considered for this simulation. The parameters set of each time variant period are provided in Table 2.2. These parameters change the behavior system dynamics (3.28). For each time interval a system is activated for the corresponding parameters.

Table 2.2: The parameters of three system dynamics for three-time intervals

Time Interval	System Parameters
$0 < t \leq 300$	$k_1 = 10 \quad b_1 = 10 \quad m_1 = 90.$
$300 < t \leq 600$	$k_2 = 30 \quad b_2 = 15 \quad m_2 = 90$
$600 < t \leq 1000$	$k_3 = 50 \quad b_3 = 50 \quad m_3 = 90.$

The Dopamine-like reinforcement Model parameters are chosen as

Initial weights wlm= 0.2 , Critic Learning rate ηc=0.08 , Baseline of Effective Reinforcement Signal d=0.5 , Discount factor γ=0.98 , Onset activations decrease of the stimulus representation components ρ=0.94, Actor Initial weights vnl = 0.4 , Actor Learning rate ηa = 0.1 , Maximum of random distribution σ=0.5, Decay of stimulus trace and action trace δ=0.96

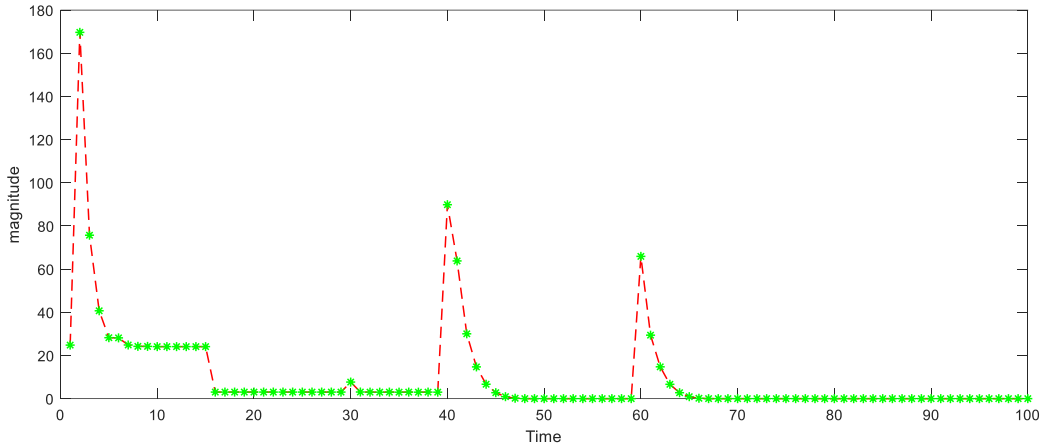

Figure 4-3: The norm of the difference between the optimal control and computed gain

Figure.3-2 shows the norm of the difference between the optimal control gain and the computed gain. It is obvious from the figure that the gain converges to the optimal value. Spikes is seen in the first iterations with system 1, a smaller spike is seen when the dynamic changing at 300 points, and tiny spike at 600. This because the extended ART sub-models are carried through the three systems, and the changing in the system dynamic is not very huge. The optimal gains and the computed gain are shown in table 3.3.

Table 3.3: The optimal gains vs the Computed gains

	Optimal Gains	Computed Gains
System 1	$K^*_{Sys1} = \begin{bmatrix} -4.292 & 23.686 \end{bmatrix}$	$K_{Sys1} = \begin{bmatrix} -4.292 & 23.686 \end{bmatrix}$
System 2	$K^*_{Sys2} = \begin{bmatrix} -0.127 & 3.019 \end{bmatrix}$	$K_{Sys2} = \begin{bmatrix} -0.127 & 3.019 \end{bmatrix}$
System 3	$K^*_{Sys3} = \begin{bmatrix} -0.489 & 1.297 \end{bmatrix}$	$K_{Sys3} = \begin{bmatrix} -0.489 & 1.297 \end{bmatrix}$

4.5 Conclusion

In this chapter Dopamine-like reinforcement Model is combined with RL to find the optimal solution to the tracking problem of time-varying discrete-time systems. The changes in the system behavior is considered using multiple-model approach. Dopamine-like reinforcement Model generates sub-models based on the clustering match-based method. A Q-learning based algorithm is then used to find the optimal solution online and without requiring any knowledge of the system dynamics. Each sub-model contributes into Q-function through a responsibility signal generated by Dopamine-like reinforcement Model.

Chapter 5 Model Reference Adaptive Impedance Control for Physical Human-Robot Interaction

Physical human-robot interaction (HRI) and cooperation has become significantly more important in recent years and is now of a major focus in robotics and control society. The empirical evidence suggests that physically embodied interactions are preferred by human operators over virtual or remote teleconference interactions [42]. Unlike ordinary industrial robotics where the environment is structured and known, in HRI systems, the robots interact with humans who have very different skills and capabilities. Therefore, it is of paramount importance for robots to adjust themselves to the level of the skills and capability of the human and compensate for possible human mistakes due to fatigue, stress, etc.

Control of industrial robots has often focused on following a desired trajectory in a well-known and structured environment. For robot manipulators with unknown nonlinear dynamics, modeling inaccuracies, and disturbances, nonlinear adaptive robot controllers have often been designed based on computed torque control [43] and/or feedback linearization [44][45] to yield guaranteed trajectory following. Adaptive control using neural networks (NNs) has been successfully employed for control of uncertain robot systems in the literature. These mentioned adaptive control methods, however, do not consider the interaction between the robot and the environment or the human. When the robot is in contact with an object or a human, it must be able to control not only positions, but also forces.

Impedance control has been widely studied in robotics as a control technique to perform robotic contact tasks. The purpose of impedance control is to provide stable tracking during robot contact with the external environment [45]-[50] by regulating the mechanical impedance response of a robot to a desired reaction according to a given task. In trajectory following, the important feature is the tracking error dynamics. Therefore, impedance control in these applications has focused on making the tracking

error dynamics behave like a prescribed impedance model [50]-[56]. Adaptive impedance control can be used to guarantee stable contact with unknown environments and specify the desirable response of the robot to an external force profile. This can potentially be used to regulate the interactions between a robot and a human operator while dynamically performing a task. Various considerations have been taken into account to tune the impedance parameters. In [52], an adaptive impedance feedforward term was used based on task requirements. In [54] adaptive controllers based on neural networks were designed in which the error dynamics parameters were tuned to become closer to a prescribed error dynamics model.

Most existing adaptive NN-based controllers and adaptive impedance controllers focus on tracking error dynamics, and/or make the tracking error dynamics have a prescribed impedance characteristic. Moreover, the control torques derived in most work has been done in the literature depends on the prescribed impedance model parameters. The objective of trajectory following with an error dynamic having prescribed impedance properties often restricts the applications of these approaches in human-robot interactive systems. Modern human-robotic interactive systems must be capable of performing a wide range of tasks. Applications in industry, military, aerospace and the gaming industry focus on semi-autonomous features of robotic systems in interacting with humans. This requires that task-specific controls include the effects of both the robot dynamics and the human dynamics, and their interactions. In this setting, trajectory following design for robot torque controllers is not suitable and limits system performance to a narrow range of tasks. In HRI systems, any trajectory tracking objectives cannot be implemented solely by the inner robot control loop because the human dynamics must be included in task trajectory following objectives.

This chapter is motivated by the human factor studies, and as opposed to most existing results does not design a robot torque controller for trajectory following. The purpose of this chapter is (1) to avoid the need for the human to learn robot-specific models, so he can focus on the task and (2) to adapt

the robot performance to assist the human-robot system in performing the task. The contributions of this chapter are as follows. An inner-loop torque controller is first designed to make the robot behave like a prescribed impedance mode l from the human force input to the robot motion coordinates. This means the human does not need to learn an inverse dynamics model to compensate for robot nonlinearities and is a completely different philosophy than making a trajectory error dynamic follow a prescribed impedance model [52]. Then, a task-specific outer-loop controller is designed, taking into account the human transfer characteristics, to tune the robot impedance model to assist the human in effectively performing the task. The outer-loop task-specific controller is designed to make the combined transfer function of the human and the robot resemble a desirable performance model based on task requirements. Techniques from model reference adaptive control are modified to accommodate the fact that the tunable impedance model appears after the plant, not before as in standard model-reference adaptive control (MRAC). This task control loop incorporates a human dynamics system identifier. Adaptive tuning algorithms are given for the robot impedance model parameters and proofs of performance are formally presented. Novel extensions to MRAC are made in the design of both the robot-specific inner loop and the task-specific outer loop controller design.

This chapter is organized as follows. Section 3.1 provides an overview of the design philosophy in this chapter. Section 3.2 designs a neural network adaptive torque controller that makes a robot dynamic appear like a prescribed robot impedance model. This design is not based on trajectory following. In Section 3.3 an outer-loop controller is designed using a novel MRAC structure that takes into account both the human dynamics model and the prescribed robot impedance model to ensure the effective performance of a task. Adaptive methods are given for tuning the robot impedance model to assist the human in the performance of the task. Section 3.4 gives simulation results and implementation results on a PR2 robot are given in Section 3.5.

5.1 Structure of Adaptive Human-Robot Interaction.

In this section we preview the overall control architecture developed in this chapter. Two control loops are designed. These control loops are motivated by human factors studies [58]-[61] that show a human operator learns two components in performing tasks with a robotic system. He learns a robot-specific inverse dynamics model to compensate for the nonlinearities of the robot. This appears to occur in the cerebellum, where supervised learning is used to learn the environment [61]. Simultaneously, he learns a task-specific feedback control component that is particular to the successful performance of the task. Some recent work in adaptive impedance control follows this approach of robot-specific impedance control inner loop design followed by a task-specific outer loop design that includes the human dynamics [57].

In this chapter, a robot-specific inner loop is first designed to make the robot dynamics from the human operator input to the robot motion appear as a prescribed robot impedance model. The robot-specific inner-loop controller appears in Figure 2-1 and is developed in Section 3.3. The objective in this loop is to design the controller torque τ to make the error between the robot position, i.e. q, and the prescribed impedance model position, i.e. q_m, go to zero. That is to design design τ to make $e_m = q - q_m$ go to zero. The input to both robot and impedance model is the human torque τ_h. This is not the same as the bulk of the work in robot impedance control [47] and neural network adaptive control [50]-[63] which is directed towards making a robot follow a prescribed trajectory, and causing the trajectory error dynamics to follow a prescribed impedance model [52]. In our approach, no trajectory information and no information of the prescribed impedance model is needed for the inner loop design. This leaves the freedom to incorporate all task information in an outer loop design. It will be seen that the robot torque input does not depend on the impedance model parameters. This contrasts with other adaptive impedance control approaches which have a trajectory following objective [52].

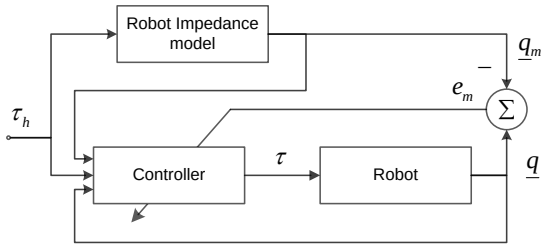

Figure 5-1 : Inner-loop robot-specific Model Reference Neuroadaptive Control

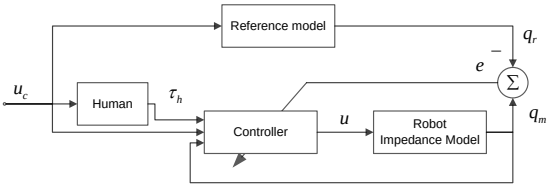

Figure 5-2: Outer-loop task-specific MRAC for Adaptive Human-Robot Interaction

An outer task-specific loop is next designed that considers the human operator dynamics. All task performance details are relegated to this outer-loop design. The task-specific outer loop design is shown in Figure.3.2 and designed in Section 3.4. The objective is to tune the robot impedance model, which is performed by designing the control input u , as described later, to make the position of impedance model tracks the position of a reference model, i.e. q_r . It is a novel form of MRAC of a different sort than Figure.3.1. The application of MRAC must be modified since the tunable parameter robot impedance model appears *after* the unknown human plant model, not before it as in standard MRAC design. Human-robot interactive systems can perform a variety of quite general tasks. In this chapter, we consider the task to be following a desired trajectory, as in point-to-point motion control by a human operator [57],[62] Then, the task reference input $u_c(t)$ in Figure. 3.2 is interpreted as the desired task trajectory to be followed by the combined man-robot system. The outer-loop design has two components. An assistive input is generated that helps the human in task performance and the prescribed robot impedance model in Figure.3.1 is adapted to enhance the human in task performance. This design must take into account

the unknown human dynamics as well as the desired overall dynamics of the human-robot system, which depends on the task.

5.2 Inner-Loop Control Design

In this section, the inner-loop torque controller for the robot manipulator shown in Figure.3.1 is derived to make the robot dynamics from human operator input to robot motion appear like a prescribed robot impedance model. A neural network approximator is used to compensate for the unknown nonlinear robot dynamics. We call this approach neuroadaptive control. The detailed result of this design is shown in Figure .3.3. No task trajectory information is needed in this design, so that this work is different from most existing work in robot control and neural network control [64].

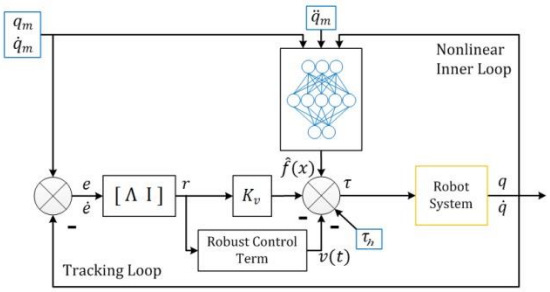

Figure 5-3 Model Reference Neuroadaptive Controller

5.2.1 Robot Impedance Model and Model-Following Error Dynamics

In this section we formulate a novel control objective for an inner-loop robot controller that does not involve trajectory tracking.

The robot dynamics equation is adapted from [43]

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = \tau + \tau_h \tag{5.1}$$

where $q \in \mathbb{R}^n$ are the robot positions, $M(q)$ is the inertia matrix, $V(q,\dot{q})$ is the Coriolis/centripetal forces, $G(q)$ is the gravity vector, and $F(\dot{q})$ is the friction term. The disturbance is

$\tau_d \in \mathbb{R}^n$ and the human operator input is τ_h. Control torque τ is to be designed to fulfil the control objective outlined above and detailed below.

Equation (5.1) can be considered as being either in joint space or Cartesian operational space. If it is in the joint space, the inputs τ_d, τ_h are torques. If it is in Cartesian space, the inputs τ_d, τ_h are forces. Forces f and torques τ are related by $\tau = J^T f$ where J is the robot Jacobian matrix. The Cartesian inertia, Coriolis/centripetal forces, friction and gravity terms are likewise determined from their joint space counterparts by using the Jacobian matrix, according to standard techniques [43].

Select the prescribed robot impedance model whose dynamics are to be followed by the robot as

$$M_m \ddot{q}_m + D_m \dot{q}_m + K_m q_m = \tau_h \tag{5.2}$$

where $q_m(t)$ is the model trajectory, M_m is the desired mass matrix, D_m is the desired damping matrix, and K_m is the desired spring constant matrix. The impedance parameters M_m, D_m, and K_m will be designed in Section 3.4 in an outer task-specific loop that takes into account both the human operator dynamics and the task objectives.

Robot-Loop Control Design Objective. Design a robot torque controller that makes the robot dynamics (5.1) from the human input τ_h to the manipulator motion $q(t)$ behave like the prescribed impedance model (5.2).

To this end, define the model-following error

$$e_m = q_m - q \tag{5.3}$$

and the sliding mode error

$$r = \dot{e}_m + \Lambda e_m \tag{5.4}$$

where Λ is a symmetric, positive definite design parameter matrix. Since (5.4) is a stable system, considering $r(t)$ as input and $e_m(t)$ as output, the control torque τ in (5.1) is now designed to guarantee that $r(t)$ is bounded. This guarantees bounded model-following error $e_m(t)$.

Using (5.1), (5.3), and (5.4) the dynamics of the sliding mode error are given by

$$M(q)\left(\ddot{q}_m - (\dot{r} - \Lambda \dot{e}_m)\right) + V(q,\dot{q})\left(\dot{q}_m - (r - \Lambda e_m)\right) + F(\dot{q}) + G(q) + \tau_d = \tau + \tau_h \tag{5.5}$$

which yields the sliding error dynamics

$$M(q)\dot{r} = -V(q,\dot{q})r + f(x) + \tau_d - \tau - \tau_h \tag{5.6}$$

where

$$f(x) = M(q)(\ddot{q}_m + \Lambda \dot{e}_m) + V(q,\dot{q})(\dot{q}_m + \Lambda e_m) + F(\dot{q}) + G(q) \tag{5.7}$$

is a nonlinear function of robot parameters which is assumed unknown. It is important to note that $f(x)$ does not depend on the impedance model parameters M_m, D_m, and K_m in (5.2). This is in contrast to impedance control robot controllers that have a trajectory following objective [52] where a tracking error is used instead of the model-following error (5.3).

5.2.2 Neuroadaptive Model-Following Controller

In this section, a control structure is given which uses a neural network (NN) to approximate the unknown function $f(x)$ in (3.7) and guarantees the stability of the model-following error(5.3). Therefore, the robot dynamics (5.1) with human input τ_h appears as the prescribed impedance model (5.2). We call this a neuroadaptive model-following controller. The use of NN in robot control is a standard approach used by many prior works [64]. In contrast to almost all these standard approaches, there is no trajectory-following objective here, so that a desired reference trajectory is not needed by the neuroadaptive controller.

To provide an approximation for the unknown function $f(x)$ in (5.7), a neural network (NN) is introduced. According to the NN approximation property [67]-[71] the nonlinear function in (3.7) can be approximated by

$$f(x) = W^T \sigma(V^T x) + \varepsilon \tag{5.8}$$

where W and V are unknown ideal NN weights and $\sigma(.)$ is a vector of activation functions. The NN input vector is $x = [e_m{}^T \;\; \dot{e}_m{}^T \;\; q_m^T \;\; \dot{q}_m^T \;\; \ddot{q}_m^T]^T$. It is known that the NN approximation error ε is bounded on a compact set. Assume the ideal weights are bounded by a constant positive scalar Z_B according to

$$\|Z\|_F = \left\| \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \right\|_F \le Z_B \tag{5.9}$$

with $\|.\|_F$ the Frobenius norm. Define matrix $\hat{Z}$ commensurately with the definition of Z.

To make the model-following error defined in (3.3) stable and consequently make the robot dynamics (5.1) behave like the prescribed impedance model, the control torque is designed as

$$\tau = \hat{W}^T \sigma(\hat{V}^T x) + K_v r - v - \tau_h \tag{5.10}$$

where $K_v r$ is a proportional-plus-derivative loop with $K_v = K_v{}^T$ a gain matrix, and

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x) \tag{5.11}$$

is the NN approximation for the unknown function $f(x)$, and

$$v(t) = -K_z \left(\left\| \hat{Z} \right\|_F + Z_B \right) r \tag{5.12}$$

with $K_z > 0$ a scalar gain is a robustifying signal that compensates for unmodeled and unstructured disturbances.

It is shown in Theorem 1 (Section 3.4) how to tune the NN weights $\hat{V}$ and $\hat{W}$ such that the control torque in (5.10) makes the model-following error (5.3) bounded and consequently the robot dynamics (5.1) from human input τ_h to the output $q(t)$ behaves like the prescribed robot impedance model (5.2).

Remark 1. The structure of the robot controller designed here is given in Figure.3-3. It is important to note that this controller guarantees model-following behavior of the robot dynamics (3.1) given the prescribed robot impedance model (5.2), based on the model-following error (5.3). There is no objective for tracking a desired trajectory. This is in contrast to almost all existing work in robot control [64] . Second, the impedance model parameters M_m , D_m , and K_m do not appear in the control law (5.10) or in the function $f(x)$ in (5.7), so that the NN does not need to identify the already-known impedance model parameters. This is reflected in Figure.3.3, where the prescribed impedance model (5.2) does not appear. This is contrast to the work on adaptive impedance control based on a trajectory tracking error dynamic [52]. As a result, the approach given here cleanly decouples the robot-specific control design given here from the task-specific control design which is given in the next section. This is in keeping with human factor studies [58] which indicate that the human learns two control components in task performance, one to compensate for nonlinear robot dynamics and one to assure task performance.

5.3 Outer-Loop Model Reference Adaptive HRI Controller

In this chapter, the task-specific outer loop controller is designed using extensions of model-reference adaptive control. The pioneering research work for model reference adaptive control (MRAC) was carried on during the 1960s by H. P. Whitaker, P. V. Osburn and A. Keze. Initial work in MRAC depended on gradient descent algorithms, including the MIT rule [72]. More rigorous Lyapunov designs

for MRAC were proposed by P. C. Parks [73]. In [74]-[79] general approaches to MRAC design and its applications were developed. Seminal work was done by [78], and others.

The objective in this section is to design the Human-robot Interaction task-specific controller in Figure.3.2 that takes into account the human dynamics, which are unknown, and the task objectives. The detailed result is in Figure.3.4. It will be seen that this task-loop controller performs two functions. It adapts the parameters of the robot impedance model (5.2) so that the task performance of the human-robot system is improved, and also provides assistive inputs that enhance the human's task performance. No robot-specific information is needed in the task loop design presented in this section. This decoupling of control objectives goes along with human factors studies in [58].

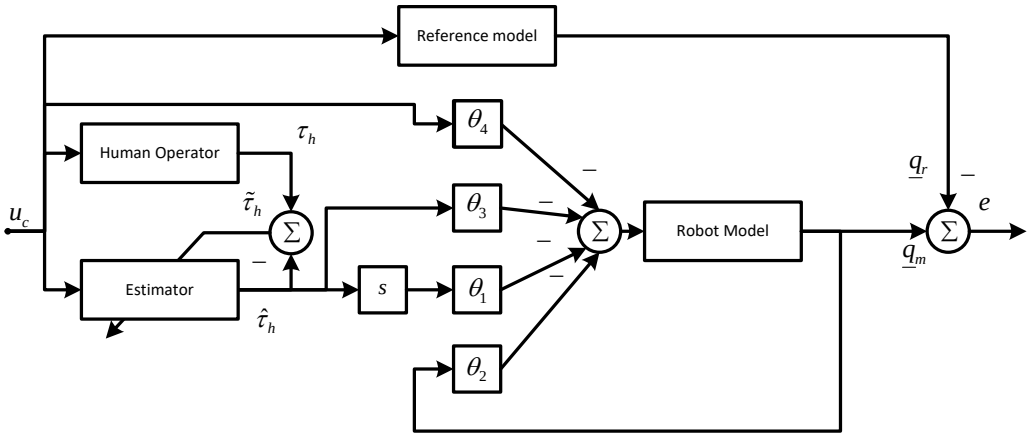

Figure 5-4: *Overall system of Model Reference Adaptive Control*

5.3.1 Model Reference Adaptive Control (MRAC) Formulation of Adaptive HRI

The problem of adapting the robot impedance model in Figure. 3.2 to assist the human in performing a task is now formulated as a nonstandard MRAC problem. The challenge to be overcome is that the tunable parameter compensator is the prescribed robot impedance model in Figures 3.2 and 3.4, which occurs *after* the unknown plant (the human dynamics), not before, as in standard MRAC. This problem is overcome by adding a system identifier for the human dynamics.

The prescribed robot impedance model (5.2) has $q_m(t) \in \mathbb{R}^n$, with *n* the number of degrees of freedom of the robot. It is assumed here that the robot dynamics (3.1) are in Cartesian task space, so that for *n=6* degrees of freedom, the vector *q*(*t*) has three position components and three angular rotation components [43]. Regarding the human transfer characteristic, it is known from human factors neurocognitive studies [58] that, in human-robot interactive task performance, the human adapts itself to compensate for robot dynamics nonlinearities and also learns task-specific controls. However, after learning, it has been observed that the expert operator exhibits the transfer characteristics of a simple linear model with a time delay. For many tasks, this human operator model is a first-order linear system of the form [62].

$$\tau_h = B(sI + A)^{-1} u_c \tag{5.13}$$

In this section, it is assumed that the human transfer matrices *A* and *B* are unknown. It is observed in human task studies there is a reaction time delay τ that is independent of the particular operator once the task has been learned, and is almost constant at 0.4s [57],[62]. Therefore, it can be compensated for, so that, without loss of generality, the delay τ can be taken as zero in (5.13) by shifting the measured time signals.

Regarding the task reference model in Figure.3.2, it is further observed in human-robot interactive task learning studies that the human operator adapts to make the overall transfer characteristic of the human-robot system appear as a simple linear first-order system with high bandwidth. This is known as the crossover model. Specifically [62], the skilled operator in a man-machine system adapts his own dynamics to make the total system transfer characteristic of the human-plus-robot remain unchanged over wide variations in the robot dynamics. The total man-robot transfer characteristic is therefore prescribed here as the task reference model

$$q_r = B_m(\mathrm{s}I + A_m)^{-1} u_c \tag{5.14}$$

with prescribed matrices A_m and B_m. These parameters are selected based on the specific task.

The class of tasks depicted in Figure.3.2 includes trajectory-following tasks where the human operates the robot to follow a prescribed trajectory. This includes point-to-point motion tasks in force fields as studied in [57],[62]. This class of tasks can be considered as having a model-following objective based on the overall task reference model (5.14), and given the unknown human dynamics (5.13) and the robot response detailed by the robot impedance model (5.2).

Remark 2. It is noted that if the task is trajectory following by the man-machine system, the parameters of the task reference model (5.14) should be selected so that $A_m = B_m$. This has low-frequency gain of 1, so that the trajectory is followed with zero steady-state error. Matrix A_m should be selected based on desired transient response characteristics of the man-machine system. This choice of task reference model does not restrict the objective to following constant trajectories. If the trajectory is time varying, suitable choice of the time-constant matrix A_m^{-1} will still result in good trajectory following.

In Section 3.3 it was assumed that the prescribed robot impedance model (5.2) is of second order. However, the model does not appear in the control design given in Theorem 1. Only the model motion trajectory $q_m, \dot{q}_m,$ and $\ddot{q}_m$ is needed in the design of the robot-specific controller there. Therefore, in this section we take a nominal prescribed robot impedance model as

$$q_m = B_n(\mathrm{s}I + A_n)^{-1}u \tag{5.15}$$

where A_n and A_n are initial nominal matrices. It is shown in the following how the overall prescribed impedance model will be changed and tuned by MRAC design to assist the human to perform a task.

Based on the above and referring to Figures 3.2 and 3.4, consider the dynamics for the human, nominal robot impedance model, and task reference model, given respectively by

$$\dot{\tau}_h = -A\tau_h + Bu_c \tag{5.16},$$

$$\dot{q}_m = -A_n q_m + B_n u \tag{5.17},$$

$$\dot{q}_r = -A_m q_r + B_m u_c \tag{5.18},$$

Here, the prescribed task trajectory is $u_c(t)$ and an MRAC control law is to be designed for the control input $u(t)$ in (5.17).

5.3.2 Adaptive Impedance Control and Human-Assistive Inputs Using Lyapunov Design

Given this setup, the basic concept of Model Reference Adaptive Control MRAC [74]-[78] can be used in this section to confront the design of the task loop of Figure.3.2. The dynamics for the human, robot impedance model, and task reference model, given respectively by (5.16), (5.17),(5.18).

Unfortunately, applying MRAC to this problem is complicated by the fact that in standard MRAC, the tunable controller appears before the unknown plant dynamics and provides its control input so that the plant has the transfer characteristics of the reference model. By contrast, in adaptive impedance control for human-robot interaction (Figure.3.2), the tunable impedance model occurs *after* the unknown human dynamics. This causes some complications and requires the introduction of a system identifier for the human dynamics. The overall setup for adaptive HRI using MRAC approach is given in Figure.3.4. The approach given here provides a formal model-following stability proof using Lyapunov techniques, and formalizes the human dynamics identifier approach used in [62].

Task-Loop Control Design Objective. Design an MRAC for control input $u(t)$ so that the combined human-robot transfer function is equal to the prescribed task reference model (5.18). See Figure.3.2.

It will be seen that the MRAC for $u(t)$ has two components. One component tunes the parameters of the robot impedance model (5.17). Then, the robot impedance model (5.17) provides the model reference trajectory $q_m, \dot{q}_m$, and $\ddot{q}_m$ used in the inner-loop torque controller of Figure 3-1 and

Theorem 1, through the sliding mode error (5.4) and the NN input vector $x = [e^T \ \ \dot{e}^T \ \ q_m^T \ \ \dot{q}_m^T \ \ \ddot{q}_m^T]^T$. The second component of the MRAC p gives assistive inputs that augment the operator's output $\tau_h(t)$ to enhance his task performance. See comments at the end of Theorem 2.

The human transfer function (5.16) is unknown. Therefore, a system identifier is introduced as

$$\dot{\hat{\tau}}_h = -\hat{A}\hat{\tau}_h + \hat{B}u_c \tag{5.19}$$

for the human response. Define the human response estimation error $\tilde{\tau}_h = \tau_h - \hat{\tau}_h$. Then, the estimation error dynamics becomes

$$\begin{aligned} \dot{\tilde{\tau}}_h = \dot{\tau}_h - \dot{\hat{\tau}}_h &= -A\tau_h + Bu_c + \hat{A}\hat{\tau}_h - \hat{B}u_c \\ &= -A\tilde{\tau}_h + \tilde{A}\hat{\tau}_h - \tilde{B}u_c \end{aligned} \tag{5.20}$$

where the identifier parameter errors are $\tilde{A} = \hat{A} - A$, and $\tilde{B} = \hat{B} - B$. Now, consider the control law

$$u = -\theta_1\dot{\hat{\tau}}_h - \theta_2 q_r - \theta_3\hat{\tau}_h - \theta_4 u_c \tag{5.21}$$

where $\theta_1, \theta_2, \theta_3$ and θ_4 are tunable matrices of appropriate dimension. Then, the overall system is illustrated in Figure.3.4. To derive tuning laws for the parameters $\theta_1, \theta_2, \theta_3 \ \ \theta_4 \ \hat{A}$, and $\hat{B}$ such that the control objective is achieved, define the model-following output error as

$$e = q_m - q_r \tag{5.22}$$

Then

$$\begin{aligned} \dot{e} &= \dot{q}_m - \dot{q}_r \\ &= -A_n q_m + B_n u + A_m q_r - B_m u_c \end{aligned} \tag{5.23}$$

Substituting the control law (5.21) into this equation and manipulating yields the model-following error dynamics as

$$\dot{e} = -A_m e - [B_n\theta_2 + A_n - A_m]q_m \quad -[B_m + B_n\theta_1\hat{B} + B_n\theta_4]u_c \quad + B_n[\theta_1\hat{A} - \theta_3]\hat{\tau}_h \tag{5.24}$$

The next result provides tuning laws for the control parameters in (5.21), the human dynamics identifier (5.19) and the neural network weights for the inner-loop controller in (5.10) that make the overall human-robot system behave like prescribed reference model (5.14).

Theorem 2. Consider the prescribed impedance model (5.2), and the robot dynamics (3.1) with control input (5.10) for the inner-loop controller. Consider the unknown human dynamics (3.16), the robot impedance model (5.17), and the outer-loop control input (5.21). Tune the NN weights in the inner-loop controller (5.10) as

$$\dot{\hat{W}} = F\hat{\sigma}r^T - F\hat{\sigma}'\hat{V}^T x r^T - \kappa F\|r\|\hat{W} \tag{5.25}$$

$$\dot{\hat{V}} = Gx(\hat{\sigma}'^T\hat{W}r)^T - \kappa G\|r\|\hat{V} \tag{5.26}$$

where F and G are symmetric positive definite matrices and $\kappa > 0$ is a small design parameter. Tune the outer-loop control parameters in (5.21) according to

$$\begin{aligned}
\dot{\theta}_1 &= \gamma_1 B_n^{-1} P_m e u_c^T \\
\dot{\theta}_2 &= \gamma_2 B_n^{-1} P_m e q_m^T \\
\dot{\theta}_3 &= \gamma_3 P_m B_n^T e\hat{\tau}_h^T + \gamma_1 \hat{A} B_n^{-1} P_m e u_c^T - \gamma_4\theta_1 P_h\tilde{\tau}_h^T\hat{\tau}_h^T \\
\dot{\theta}_4 &= \gamma_1(\hat{B} + B_n^{-1}) P_m e u_c^T
\end{aligned} \tag{5.27}$$

with $P_m > 0$ and $P_h > 0$, and the parameters in the human system identifier (5.19) according to

$$\dot{\hat{A}} = -\gamma_4 P_h\tilde{\tau}_h\hat{\tau}_h^T, \dot{\hat{B}} = \gamma_5 P_h\tilde{\tau}_h^T u_c^T \tag{5.28}$$

Then, the inner-loop model-following error $e_m(t)$, the outer-loop model following error $e(t)$ and the human response estimation error are bounded, so that the product of the human dynamics and robot dynamics follows the task reference model (5.18).

Proof: Define a Lyapunov function as:

$$
\begin{aligned}
L = & \frac{1}{2} r^T M(q) r + tr(\tilde{W}^T F^{-1} \tilde{W}) + tr(\tilde{V}^T F^{-1} \tilde{V}) + \frac{1}{2} e^T P_m e \\
& + \frac{1}{2\gamma_2} tr([B_n \theta_2 + A_n - A_m]^T [B_n \theta_2 + A_n - A_m]) \\
& + \frac{1}{2\gamma_1} tr([B_m + B_n \theta_1 \hat{B} + B_n \theta_4]^T [B_m + B_n \theta_1 \hat{B} + B_n \theta_4]) \\
& + \frac{1}{2\gamma_3} tr([\theta_1 \hat{A} - \theta_3]^T [\theta_1 \hat{A} - \theta_3]) + \frac{1}{2} \tilde{\tau}_h^T P_h \tilde{\tau}_h^T \\
& + \frac{1}{2\gamma_4} tr(\tilde{A}^T \tilde{A}) + \frac{1}{2\gamma_5} tr(\tilde{B}^T \tilde{B})
\end{aligned}
\tag{5.29}
$$

where the weight estimation errors are $\tilde{W} = W - \hat{W}$, $\tilde{V} = V - \hat{V}$. Differentiating this Lyapunov function and using (5.24) yields

$$
\begin{aligned}
& \dot{L} = r^T M(q) \dot{r} + \frac{1}{2} r^T \dot{M}(q) r + tr(\tilde{W}^T F^{-1} \dot{\tilde{W}}) + tr(\tilde{V}^T F^{-1} \dot{\tilde{V}}) + e^T \{ -\frac{1}{2} (A_m{}^T P_m + P_m A_m) e \\
& - P_m [B_n \theta_2 + A_n - A_m] q_m - P_m [B_m + B_n \theta_1 \hat{B} + B_n \theta_4] u_c + P_m B_n [\theta_1 \hat{A} - \theta_3] \hat{\tau}_h \} \\
& + \frac{1}{\gamma_2} tr([B_n \theta_2 + A_n - A_m]^T B_n \dot{\theta}_2) + \frac{1}{\gamma_1} tr([B_m + B_n \theta_1 \hat{B} + B_n \theta_4]^T (\hat{B} B_n \dot{\theta}_1 + B_n \dot{\theta}_4) \\
& + \frac{1}{\gamma_3} tr([\theta_1 \hat{A} - \theta_3]^T [\hat{A} \dot{\theta}_1 + \theta_1 \dot{\hat{A}} - \dot{\theta}_3]) + \tilde{\tau}_h^T P_h \dot{\tilde{\tau}}_h + \frac{1}{\gamma_4} tr(\tilde{A}^T \dot{\tilde{A}}) + \frac{1}{\gamma_5} tr(\tilde{B}^T \dot{\tilde{B}})
\end{aligned}
\tag{5.30}
$$

Since $-A_m$ is Hurwitz, there exists a $Q_m > 0$ such that $A_m{}^T P_m + P_m A_m = Q_m$

The robot manipulator dynamics (5.1) is assumed to be unknown and therefore the function f in (3.7) is unknown and approximated online by (3.11). Then, the closed-loop filtered error dynamics (3.6) becomes

$$
M(q) \dot{r} = -V(q, \dot{q}) r + \hat{W}^T \phi(\hat{V}^T z) + \tau_d - \tau - \tau_h + \tilde{f} \tag{5.31}
$$

where $\tilde{f}(x) = f(x) - \hat{f}(x)$ is the estimation error. Substituting τ form (3.10) in (3.31) gives:

$$M(q)\dot{r} = -V(q,\dot{q})r - K_v r + \tau_d + \tilde{f}(x) + v(t) \tag{5.32}$$

On the other hand, since

$e^T\{P_m[B_n\theta_2 + B_n - B_m]q_m - P_m[B_m + B_n\theta_1\hat{B} + B_n\theta_4]u_c + P_m B_n[\theta_1\hat{A} - \theta_3]\hat{\tau}_h\}$ is scalar, one has

$$\begin{aligned} &e^T\{P_m[B_n\theta_2 + A_n - A_m]q_m - P_m[B_m + B_n\theta_1\hat{B} + B_n\theta_4]u_c + P_m B_n[\theta_1\hat{A} - \theta_3]\hat{\tau}_h\} = \\ &tr([B_n\theta_2 + A_n - A_m]^T P_m e\, q_m{}^T - [B_m + B_n\theta_1\hat{B} + B_n\theta_4]^T P_m e\, u_c{}^T + [\theta_1\hat{A} - \theta_3]^T P_m B_n{}^T e\,\hat{\tau}_h^T) \end{aligned} \tag{5.33}$$

Using (3.25), (3.26), 3.32 and (3.33) into (3.30) gives:

$$\begin{aligned} \dot{L} &= -r^T K_v r + \frac{1}{2} r^T(\dot{M}(q) - 2V(q,\dot{q}))r + tr\{\tilde{W}^T(F^{-1}\dot{\tilde{W}} + \hat{\sigma}r^T - \sigma'\hat{V}^T x r^T)\} \\ &+ tr\{\tilde{V}^T(G^{-1}\dot{\tilde{V}} + x r^T\hat{W}^T\hat{\sigma}')\} - \frac{1}{2}e^T Q_m e - tr([B_n\theta_2 + A_n - A_m]^T P_m e\, q_m{}^T) \\ &- tr([B_m + B_n\theta_1\hat{B} + B_n\theta_4]^T P_m e\, u_c{}^T) + tr([\theta_1\hat{A} - \theta_3]^T P_m B_n{}^T e\,\hat{\tau}_h^T\} \\ &+ \frac{1}{\gamma_2} tr([B_n\theta_2 + A_n - A_m]^T B_n\dot{\theta}_2) \\ &+ \frac{1}{\gamma_1} tr([B_m + B_n\theta_1\hat{B} + B_n\theta_4]^T(\hat{B}B_n\dot{\theta}_1 + B_n\dot{\theta}_4) \\ &+ \frac{1}{\gamma_3} tr([\theta_1\hat{A} - \theta_3]^T[\hat{A}\dot{\theta}_1 + \theta_1\dot{\hat{A}} - \dot{\theta}_3]) + \\ &\tilde{\tau}_h^T[-\frac{1}{2}(A^T P_h + P_h A)\tilde{\tau}_h + P_h\tilde{A}\hat{\tau}_h - P_h\tilde{B}u_c] \\ &+ \frac{1}{\gamma_4} tr(\tilde{A}^T\dot{\tilde{A}}) + \frac{1}{\gamma_5} tr(\tilde{B}^T\dot{\tilde{B}}) \end{aligned} \tag{5.34}$$

Noting that since $-A$ is stable there exists a $Q_h > 0$ such that $A^T P_h + P_h A = Q_h$, using $\tilde{\tau}_h^T[P_h\tilde{A}\hat{\tau}_h - P_h\tilde{B}u_c] = tr(\tilde{A}^T P_h\tilde{\tau}_h\tilde{\tau}_h^T + \tilde{B}^T P_h u_c\,\tilde{\tau}_h^T)$ and using the tuning rules for the inner- and outer- loop controllers gives

$$\begin{aligned} \dot{L} &= -r^T K_v r - \frac{1}{2}e^T Q_m e - \frac{1}{2}\tilde{\tau}_h Q_h\,\tilde{\tau}_h{}^T + k\|r\| tr\{\tilde{W}^T(W - \tilde{W})\} \\ &+ k\|r\| tr\{\tilde{V}^T(V - \tilde{V})\} + r^T(w + v) \\ &= -r^T K_v r + k\|r\| tr\{\tilde{Z}^T(Z - \tilde{Z})\} + r^T(w + v) \end{aligned} \tag{5.35}$$

Since

$tr\{\tilde{Z}^T(Z-\tilde{Z})\} = <\tilde{Z}, Z> - \left\|\tilde{Z}\right\|_F^2 \le \left\|\tilde{Z}\right\|_F \|Z\|_F - \left\|\tilde{Z}\right\|_F^2$ One has

$$\begin{aligned}
\dot{L} &\le -r^T K_v r - e^T A_m e - \tilde{\tau}_h^T a \tilde{\tau}_h + k\|r\| \cdot \left\|\tilde{Z}\right\|_F (\mathrm{Z}_B - \left\|\tilde{Z}\right\|_F) - \mathrm{K}_Z(\left\|\hat{Z}\right\|_F + Z_B)\|r\|^2 + \|r\| \cdot \|w\| \\
&\le K_{v_{\min}} \|r\|^2 + k\|r\| \cdot \left\|\tilde{Z}\right\|_F (\mathrm{Z}_B - \left\|\tilde{Z}\right\|_F) - \mathrm{K}_Z(\left\|\hat{Z}\right\|_F + Z_B)\|r\|^2 + \|r\|\left[C_0 + C_1\left\|\tilde{Z}\right\|_F + C_2\|r\| \cdot \left\|\tilde{Z}\right\|_F\right] \qquad (5.36) \\
&\le -\|r\|\left\{K_{v_{\min}}\|r\| - k \cdot \left\|\tilde{Z}\right\|_F (\mathrm{Z}_B - \left\|\tilde{Z}\right\|_F) - C_0 - C_1\left\|\tilde{Z}\right\|_F\right\}
\end{aligned}$$

where $K_{v_{min}}$ is minimum singular value of K_v and the last inequality $\dot{L}$ is negative as long as the term in braces is positive. Defining $C_3 = Z_B + C_1 / k$ and completing the square yields

$$K_{v_{\min}}\|r\| - k \cdot \left\|\tilde{Z}\right\|_F (\mathrm{Z}_B - \left\|\tilde{Z}\right\|_F) - C_0 - C_1\left\|\tilde{Z}\right\|_F = k(\left\|\tilde{Z}\right\|_F - \mathrm{C}_3/2)^2 + K_{v_{\min}}\|r\| - C_0 - kC_3^{\,2}/4$$

which is guaranteed positive as long as either

$$\|r\| > \frac{C_0 + kC_3^{\,2}/4}{K_{v_{\min}}} \text{ or } \left\|\tilde{Z}\right\| > \frac{\mathrm{C}_3}{2} + \sqrt{\frac{C_0}{k} + \frac{C_3^{\,2}}{4}}$$ This completes the proof. ■

This result provides a method for tuning the robot impedance model (3.17) to provide a desired model reference output $q_r(t)$ such that the human (3.16) plus robot impedance model follows

The prescribed task reference model (3.18). This output is sent as $q_m(t)$ to the inner robot control loop in Figure.3.3 to compute the inner-loop model following error (3.3). The human input in Figure.3.3 and in (3.10) is $\tau_h(t)$. These relationships are shown in Figure. 3.2 and Figure.3.4.

It is interesting to examine the operation of the control input (3.21). After convergence of the human system identifier, one has $\hat{\tau}_h(t) = \tau_h(t)$. Then, the closed-loop robot impedance model is

$$
\begin{aligned}
q_r &= \frac{B_n}{s+A_n}u \\
&= \frac{B_n}{s+A_n}(-\theta_1 \dot{\hat{\tau}}_h - \theta_2 q_r - \theta_3 \hat{\tau}_h - \theta_4 u_c) \\
&= \frac{-B_n}{s+(A_n+B_n\theta_2)}[(\theta_1 s+\theta_3)\tau_h + \theta_4 u_c] \\
&\equiv \frac{-B_n}{s+(A_n+B_n\theta_2)}\overline{\tau}_h
\end{aligned}
\tag{5.37}
$$

where $\overline{Y}(t)$ can be considered as a modified human force defined, according to (3.13) , by

$$
\overline{\tau}_h = [\frac{B(\theta_1 s+\theta_3)}{s+A} + \theta_4]u_c \tag{5.38}
$$

Therefore, control parameter θ_2 modifies the robot impedance model time constant, whereas control parameters θ_1, θ_3, and θ_4 provide a proportional-plus-derivative controller that augments the human force signal τ_h . This can be viewed as an assistive term that aids the human so that task performance is improved. In fact, it is observed in [62] that the expert human operator, after learning to accomplish a task, incorporates a PD controller that seems to come from a task model learned in the cerebellum [58].

5.4 Simulation

In this section, the results from simulating the proposed controllers on a 2-link robotic arm in MATLAB are presented. The 2-link robot arm is a revolute-revolute planar arm described in [43] Example 3.2-2. First are shown the simulation results for the outer-loop controller in Figure.3.2 and Figure.3.4 that adapts the parameters of the prescribed robot impedance model. Next are shown the simulation results for the inner-loop model reference neuroadaptive controller in Figures 3.1 and 3.3.

5.4.1 Outer-loop Simulation

This simulation is for the outer task loop shown in Figures 3.2 and 3.4. In this simulation the prescribed robot impedance model (3.2) is chosen to have $q_m(t) \in \mathbb{R}^2$, with *n=2* the number of degrees of

freedom of the robot. It is assumed here that the robot dynamics (3.1) are in Cartesian task space. The nominal robot impedance model (3.17) is chosen for each degree of freedom as $\dot{q}_m = -3q_m + 3u$. These nominal time constants and gain parameters are modified through the action of the adaptive control (3.12). See the discussion after Theorem 2. The task reference model (3.18) is taken as $\dot{q}_r = -12q_r + 12u_c$, where u_c is the desired trajectory to be reached in a point-to-point motion task. The unknown human dynamics model (3.16) is chosen as $\dot{\tau}_h = -1\tau_h + 0.5u_c$. The human dynamics model is unknown, and the human system identifier model (3.19) is designed to adaptively identify the human in the loop.

The performance of the outer task loop MRAC in Figure3.4 is shown in Figures 3.5,3.6,3.7. A square wave is selected for the task reference input $u_c(t)$. This is interpreted as a point-to-point motion task where the human-robot system is required to cycle from one point to another point repetitively. Figure.3.5 shows the output of the robot impedance model $q_m(t)$ and the output $q_r(t)$ of the task reference model. It is seen that $q_m(t)$ closely follows $q_r(t)$, with performance improving after several cycles. This shows the adaptive improvement of the controller as the robot impedance model is tuned and the assistive inputs to the human are learned. See discussion after Theorem 2. The effectiveness of the human system identifier is revealed in Figure.3.6, which shows the output of the human transfer function $\tau_h(t)$ and the output of the human identifier $\hat{\tau}_h(t)$, which follows $\tau_h(t)$ more closely with each motion cycle. The convergence of the human identifier parameters to the actual human model parameters is shown in Figure.3.7.

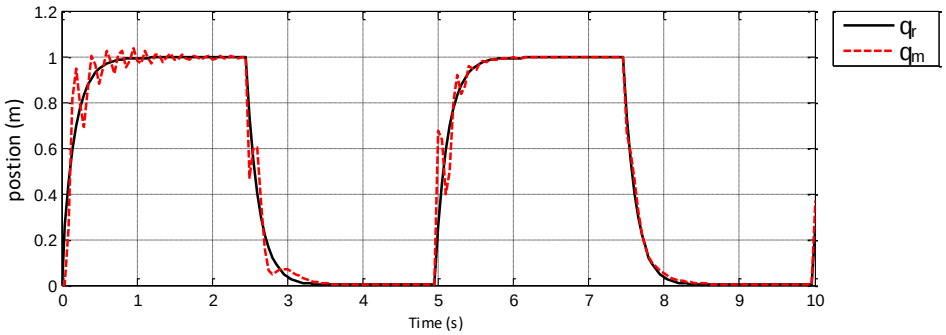

Figure 5-5 Robot Impedance Model qm(t)Output and Prescribed Task Reference Output qm(t

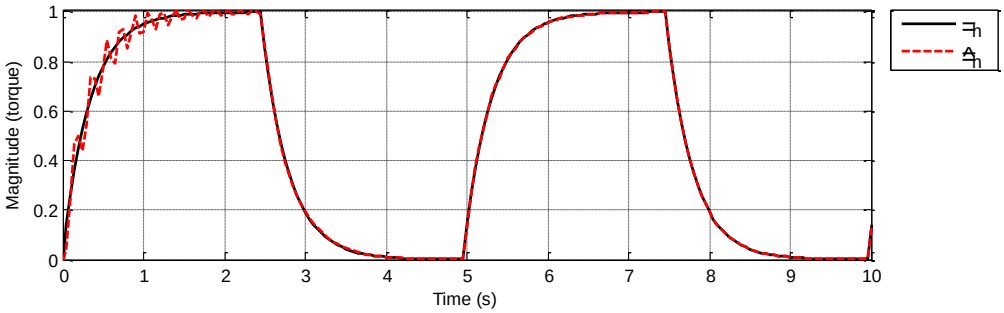

Figure 5-6 *Human Output* $\tau_h(t)$ *and Human Identifier Output* $\hat{\tau}_h(t)$

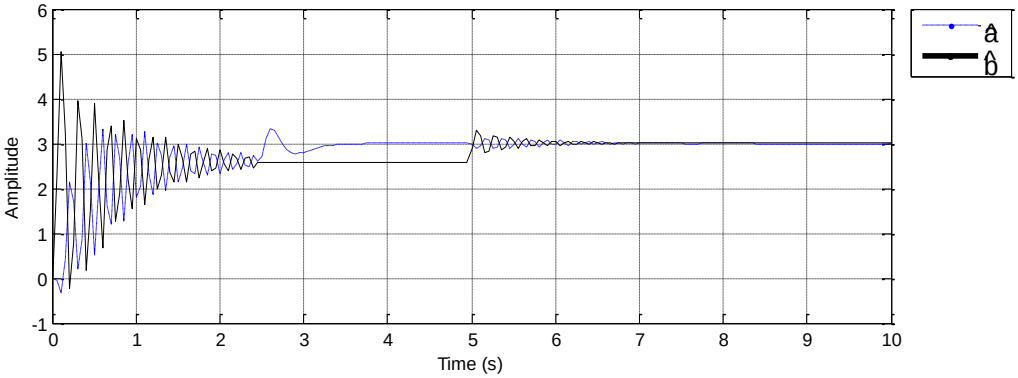

Figure 5-7 Parameter Convergence of Adaptive Human Identifier Model

5.4.2 Inner-loop Simulation

This simulation is for the inner robot control loop of Figures 3.1 and 3.3. The outer-loop design just described generates the human operator signal $\tau_h(t)$ and the robot impedance model trajectory $q_m(t)$. Two parallel outer loops were used, one for each joint of the 2-link robot arm simulated here.

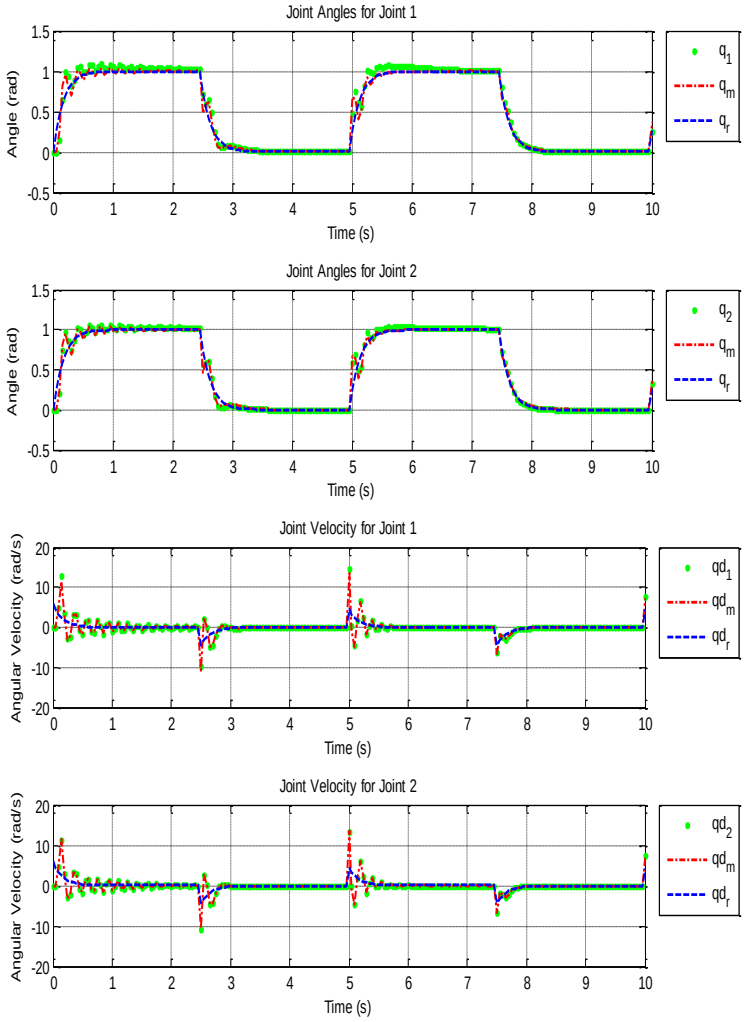

Figure 5-8 Inner-loop simulation

The robot dynamics (3.1) used for this simulation was the 2-link revolute-revolute planar robot arm described in [43] Example 3.2-2. The arm parameters are selected as $m_1 = 0.8kg$, $m_1 = 2.3kg$, $l_1 = 1m$, $l_2 = 1m$ and $g = 9.8m/s^2$. The controller parameters used in Theorem 1 were $K_v = I_2$, $\Lambda = 5I_2$,

$F = 100I_2$, $G = 20I_2$, κ = 0.07, K_z = 5, and Z_B = 100, where I_2 is the 2×2 identity matrix. A two-layer Neural Network was used with 10 inputs, including a constant bias input, 20 hidden layer neurons and 2 outputs. The sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ was used for the activation functions. The weights $\hat{W}$ and $\hat{V}$ of the network were randomly initialized.

The simulation results for both links are shown in Figure.3.8, where $q_{1d}(t), q_{2d}(t)$ denote the 2 components of the task trajectory $u_c(t)$. It is observed that, after a short transient learning period of a few cycles of the square wave task trajectory, the motion $q_m(t)$ generated by the robot impedance model and the robot motion $q(t)$ are identical. This verifies the performance of the model reference neuroadaptive controller in making the robot arm behave like the robot impedance model.

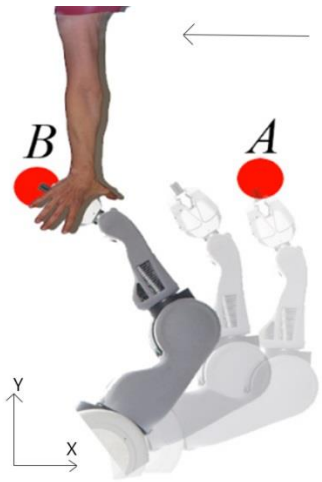

Figure 5-9 Experiment Layout

5.4.3 Overall Performance of the Proposed Controller

It can be seen from the simulation results that the two controllers, inner robot loop and outer task loop, achieve the objectives of the design. The outer loop assists the human in achieving the task by providing two assistive components and tuning the robot impedance model. The robot specific inner-loop

controller compensates for the robot nonlinearities and makes the robot behave like this robot impedance model.

5.5 Experimental Case Study

In this section a case study of a practical experiment to evaluate the controllers of the Human-Robot interaction system is presented. The experiments were conducted at the University of Texas at Arlington Research Institute on a PR2 robot. Figure.3.9 shows the experimental layout and Figure.3.10 shows the PR2 robot. The controller was implemented in real-time using the real-time controller manager framework of the PR2 in ROS Groovy. The real-time loop on the PR2 runs at 1000Hz and communicates with the sensors and actuators on an EtherCAT network. Human force is measured using an ATI Mini40 FT Sensor attached between the gripper and forearm of the PR2.

Figure 5-10 PR2 Robot at UTARI

The experiment involves the seven degree-of-freedom arm of the PR2 robot in a point-to-point motion (PTP) task. PTP manipulation is an increasingly popular task, both in the game industry and in industrial applications.

In this experiment a human applies a force on the right arm of the PR2 to follow the PTP motion trajectory, as shown in Figure.3.9. The experiment is setup with a human operator and the PR2 arm across from each other as seen in Figure.3.10. The human operator was then asked to hold the gripper of the PR2 to perform PTP motion between point A and B along the y axis. The human is assumed to be working in open-loop without considering the visual feedback of the current location and the target location of the gripper. The desired target location to be reached is switched every 5 seconds.

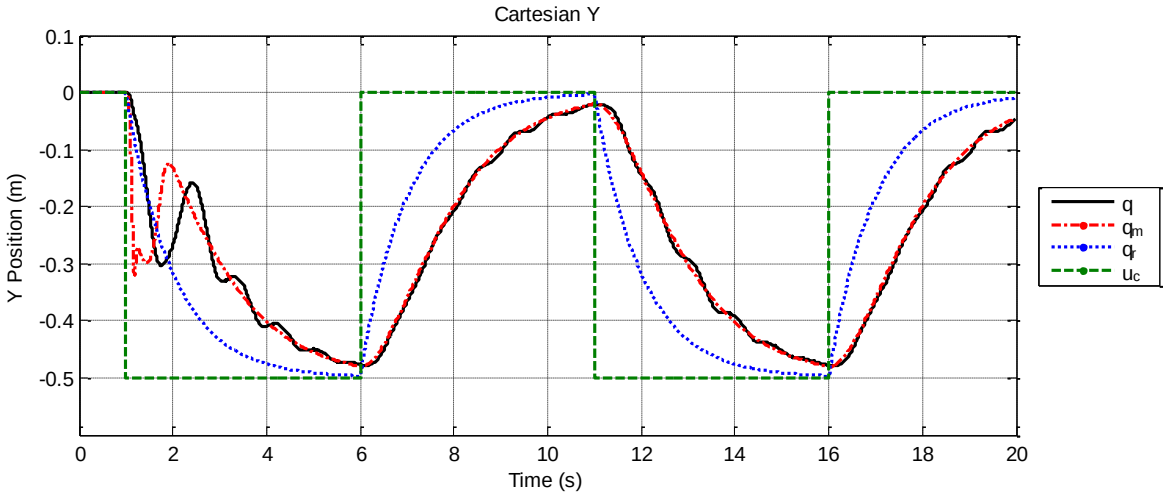

Figure 5-11 SIMULATION RESPONS INTERACTION OF HUMAN-ROBOT INTERACTIVE SYSTEM

The controller parameters used were $K_v = 5I_6$, $\Lambda = 20I_6$, $F = 100I_6$, $G = 200I_6$, κ = 0.3, K_z = 0.001, and Z_B = 100,

where I_6 is the 6×6 identity matrix. A two-layer Neural Network was used with 35 inputs, including the bias input, 10 hidden layer neurons, and 7 outputs. The sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ was used for the activation functions. The weights $\hat{W}$ and $\hat{V}$ of the network were randomly initialized.

The result of the whole human-robot interaction system is shown in Figure.3.11. The task trajectory (in green) gives the target point locations, which cycle every five seconds. The task reference model output is shown (in blue) followed by the robot impedance model output (in red) and the real robot

output (in black). It is seen that the inner-loop Neuroadaptive controller makes the robot (in green) follow the robot admittance model output (in red), and the outer-loop MRAC makes the human-robot interactive team follow the prescribed task model. This is accomplished after a short transient learning time where the adaptation mechanism tunes the whole system in the first 6 seconds. There is a small-time delay of 0.4 sec due to the human reaction time.

5.6 Conclusion

This chapter presented a novel method of enhancing human-robot interaction based on model reference adaptive control. The method presented delivers guaranteed stability and task performance and has two control loops. A robot-specific inner loop is a Model Reference Neuroadaptive Controller that learns the robot dynamics online and makes the robot responds like a prescribed impedance model. This loop uses no task information, including no prescribed trajectory. A task-specific outer loop takes into account the human operator dynamics and adapts the prescribed robot impedance model so that the combined human-robot system has desirable characteristics for task performance. This design is also based on model reference adaptive control, but of a nonstandard form. The net result is a controller with both adaptive impedance characteristics and assistive inputs that augment the human operator to provide improved task performance of the human-robot team. Simulations verify the performance of the proposed controller in a repetitive point-to-point motion task. Actual experimental implementations on a PR2 robot further corroborate the effectiveness of the approach.

Chapter 6 Conclusions and Future Work

AS a future work, chapter three will be presented as a journal paper, with some modification. In addition, the rest of the model parameters are going to be further explored. The learning for acquisition, the learning for extinction parameter, and Reset duration modification for dynamical models is needed to be explored.

Also, the application of the adaptive resonance theory, and the dopamine like model for application such as multiagent system, and game theory application is another area to be explored.

The work in multiagent system specifically can be extended using the presented dopamine like model by switching each sub-model into an agent, the control structure is to have a consensus on task trajectory reference.

Appendix

Acknowledgements:

Reprinted by permission from Licensed Content Publisher Springer Nature , Licensed Content Publication Control Theory and Technology , Licensed Content Title Model reference adaptive impedance control for physical human , robot interaction , Licensed Content Author Bakur Alqaudi, Hamidreza Modares, Isura Ranatunga et al , Licensed Content Date Jan 1, 2016 , License Number 4400720890178

Refences

[1] Beetz, Michael, Martin Buss, and Dirk Wollherr. "Cognitive technical systems—what is the role of artificial intelligence?." *Annual Conference on Artificial Intelligence*. Springer, Berlin, Heidelberg, 2007.

[2] Z. Shi, S. Hirche, W. X. Schneider and H. Muller, "Influence of visuomotor action on visual-haptic simultaneous perception: A psychophysical study," *2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Reno, NE, 2008, pp. 65-70.

[3] euCognition, the European Network for the Advancement of Artificial Cognitive Systems [Online], 2010. Available at http://www.eucognition.org/euCognition_2006-2008/definitions.htm.

[4] Buss, M.; Carton, D.; Gonsior, B.; Kuehnlenz, K.; Landsiedel, C.; Mitsou, N.; Nijs, R. de; Zlotowski, J.; Sosnowski, S.; Strasser, E.; Tscheligi, M.; Weiss, A.; Wollherr, D., "Towards proactive human-robot interaction in human environments," 2011 2nd International Conference on Cognitive Infocommunications (CogInfoCom), Budapest, 2011, pp. 1-6.

[5] R. G. Costello and T. J. Higgins, "An Inclusive Classified Bibliography Pertaining to Modeling the Human Operator as an Element in an Automatic Control System," in IEEE Transactions on Human Factors in Electronics, vol. HFE-7, no. 4, pp. 174-181, Dec. 1966.

[6] H. Kobayashi, Y. Ohyama, J. H. She, M. Hosaka and H. Hashimoto, "Transfer function representation of situated human's controller," 30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004, 2004, pp. 653-656 Vol. 1.

[7] S. Suzuki, K. Furuta. Adaptive impedance control to enhance human skill on a haptic interface system. Journal of Control Science and Engineering .

[8] Liang Gong ; Changyang Gong ; Zhao Ma ; Lujie Zhao ; Zhenyu Wang ; Xudong Li ; Xiaolong Jing ; Haozhe Yang, "Real-time human-in-the-loop remote control for a life-size traffic police robot with multiple augmented reality aided display terminals," 2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM), Hefei, 2017, pp. 420-425.

[9] Hans G. Furth, "Biology and Knowledge. An Essay on the Relations between Organic Regulations and Cognitive Processes. Jean Piaget ," The Quarterly Review of Biology 47, no. 2 (Jun., 1972): 203.

[10] Munakata, Y. and Pfaffly, J., Hebbian learning and development. Developmental Science, 2004,7: 141-148.

[11] Daniel S. Levine, Introduction to Neural and Cognitive Modeling, L. Erlbaum Associates Inc., Hillsdale, NJ, 2000.

[12] Grossberg, S. Adaptive pattern classification and universal recoding: Parallel development and coding of neural feature detectors. Biological Cybernetics, 1976,23, 121 134.

[13] AlQaudi, B., Levine, D. S., & Lewis, F. L. (2015). Neural network model of decisions on the Asian disease problem. Proceedings of International Joint Conference on Neural Networks 2015, 1333-1340.

[14] D. S. Levine, K. Y. Chen and B. AlQaudi, "Neural network modeling of business decision making," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 206-213.

[15] Paul Werbos, “What is Mind? What is Consciousness? How Can We Build and Understand Intelligent Systems?”, Werbos’ website.

[16] Paul J. Werbos, “Neural networks and the human mind: New mathematics fits humanistic insight”, IEEE International Conference on Systems, Man, and Cybernetics, vol. 1, 1992, pp. 78-83.

[17] Paul J. Werbos, “Intelligence in the brain: A theory of how it works and how to build it”, Neural Networks 22, 2009, pp. 200-212.
[18] Suri, R. E., & Schultz, W. Learning of sequential movements by neural network model with dopamine-like reinforcement signal. Experimental Brain Research, 1998, 121, 350-354.
[19] Suri, R. E., & Schultz, W. A neural network model with dopamine-like reinforcement signal that learns a spatial delayed response task. Neuroscience, 1999, 91, 871-890.
[20] Kumpati Narendra, Jeyendran Balakrishnan, “Adaptive control using multiple models”, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 42, NO. 2, 1997, pp. 171-187.
[21] K. Doya, Kazuyuki Samejima, Ken-ichi Katagiri, Mitsuo Kawato, “Multiple Model-Based Reinforcement Learning”, Neural Computation, 2002, pp. 1347-1369.
[22] F.L. Lewis, K. Liu, and A. Yesildirek, Neural net robot controller with guaranteed tracking performance, IEEE Trans. Neural Networks, 6 (3) (1995) 703-715.
[23] B. Kiumarsi, F. L. Lewis, D. S. Levine, Optimal control of nonlinear discrete time-varying systems using a new neural network approximation structure, Neurocomputing, 156 (2015) 157-165
[24] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, A. G, Adaptive linear quadratic control using policy iteration, Proceedings of IEEE American control conference, Baltimore, Maryland, (1994) 3475-3479.
[25] A. Al-Tamimi, F.L. Lewis, M. Abu-Khalaf, Model-free Q-learning designs for linear discrete-time zero-sum games with application to HH-infinity control, Automatica, 43 (3) (2007) 473–481.
[26] Q. Wei, R. Song, Q. Sun, Nonlinear neuro-optimal tracking control via stable iterative Q-learning algorithm, Neurocomputing, 168 (2015) 520-528.
[27] F.L. Lewis, D. Vrabie, K.G. Vamvoudakis, Reinforcement learning and feedback control using natural decision methods to design optimal adaptive controllers, IEEE Systems Magazine, 32 (6) (2012) 76–105
[28] K.G. Vamvoudakis, F.L. Lewis, Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica 46 (5) (2010) 878–888.
[29] P.J. Werbos, A menu of designs for reinforcement learning over time, Neural networks for control, MIT Press, Cambridge, MA (1991), pp. 67–95.
[30] D. S. Levine, Neural dynamics of affect, gist, probability, and choice, Cognitive System Research 16 (2012) 57-72.
[31] K.S. Narendra, J. Balakrishnan, Improving transient response of adaptive control systems using multiple models and switching, IEEE Trans. Automatic Control, 39 (9) (1994) 1861–1866.
[32] K. Pawelzik, J. Kohlmorge, K.R Muller, Annealed competition of experts for a segmentation and classification of switching dynamics, Neural Computation, 8 (1996) 340–356.
[33] S. Grossberg, Competitive learning: From interactive activation to adaptive resonance. Cognitive science, 11 (1) (1987) 23-63.
[34] J. A. Hartigan, M. A. Wong, Algorithm AS 136: A K-Means Clustering Algorithm, Journal of the Royal Statistical Society. Series C (Applied Statistics), 28 (1) (1979), 100–108.
[35] , M.C. Naldi, R.J.G.B. Campello, Comparison of distributed evolutionary k-means clustering algorithms, Neurocomputing, 163 (2015) 78-93.
[36] T. Kohonen, Self-organizing Maps, Springer, Berlin and Hiidelberg, 1995.
[37] F. Coleca, A.State, S.Klement, E.Barth, T.Martinetz, Self-organizing maps for hand and full body tracking, Neurocomputing, 47 (2015) 174-184.

[38] S. Grossberg, Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors & II: Feedback, expectation, olfaction, and illusions, Biological Cybernetics, (1976) 187-202.
[39] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M-B. Naghibi, Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. Automatica, 50 (4) (2014) 1167-1175.
[40] T. Frank, K. F. Kraiss and T. Kuhlen, Comparative analysis of fuzzy ART and ART-2A network clustering performance, IEEE Transactions on Neural Networks, 9 (1998) 544-559.
[41] K. Oweiss, R. Jin, Y. Suhail, Identifying neuronal assemblies with local and global connectivity with scale space spectral clustering, Neurocomputing, 70 (2007) 1728-1734.
[42] Wainer, J, Feil-Seifer, D.J, Shell, D.A, and Mataric, M.J,"The role of physical embodiment in human-robot interaction", The 15th IEEE International Symposium on Robot and Human Interactive Communication, 2006.
[43] Lewis, F.L, Dawson, D, Abdallah, M. and Chaouki,T, Robot Manipulator Control: Theory and Practice , CRC Press, 2003.
[44] Slotine, J. J E, and Li,Weiping, Applied Nonlinear Control ", Prentice Hall, 1991.
[45] Jamshidi,M, Oh,B. J, Oh, and Seraji,H,"Two adaptive control structures of robot manipulators", Journal of Intelligent and Robotic Systems, Vol. 6, No. 2-3, pp. 203-218, 1992.
[46] Hogan, N, " Impedance Control: An Approach to Manipulation ", American Control Conference 304-313, 1984.
[47] Anderson, R, Spong, " Hybrid Impedance Control of Robotic Manipulators ", Journal Robot Automation, Vol. 1, No. 5, pp. 549–556, 1988.
[48] Kawasaki, H. and Taniuchi, R. "Adaptive Control For Robotic Manipulators Executing Multilateral Constrained Task". Asian Journal of Control,2003.
[49] Hanmei, Wu, Wenkang, Xu, Chenxiao, " Adaptive impedance control in robotic cell injection system ", International Conference on Methods and Models in Automation and Robotics pp.268-275, 2012.
[50] Ge,S.S, Hang,C.C, Lee,T.H, and Zhang,T, Stable Adaptive Neural Network Control, Kluwer academic. Boston, 2001.
[51] Vukobratovic, M.K, Rodic, A.G, and Ekalo, Y, "Impedance control as a particular case of the unified approach to the control of robots interacting with a dynamic known environment". J Intell Robot Syst, Vol. 18, No.2,pp.191–204,1992..
[52] Gribovskaya,E, Kheddar,A, and Billard,A, " Motion learning and adaptive impedance for robot control during physical interaction with humans ", IEEE International Conference on Robotics and Automation 4326–4332, 2011.
[53] Huang, L,Ge,S.S, and Lee,T.H,, " Neural Network Based Adaptive Impedance Control of Constrained Robots ", Proceedings of the IEEE International Symposium 615-619, 2002.
[54] Wang, C, Li,Y,Shuzhi,Ge, S,Peng Tee,Keng, Tong Lee,Heng, " Continuous critic learning for robot control in physical human-robot interaction ", 13th International Conference on Control, Automation and Systems , pp. 833-838, 2013.
[55] Yanan,L, Ge, S.S, and Chenguang Y, " Impedance control for multi-point human-robot interaction ", 8th Asian Control Conference , pp. 1187-1192, 2011.

[56] Tsuji, T, and Tanaka, Y, " Tracking control properties of human-robotic systems based on impedance control ", IEEE Trans.on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 35, No. 4, pp. 523-535, 2005.
[57] Suzuki,S, Kurihara,K,Furuta, K,Harashima,F, and Pan,Y, " Variable Dynamic Assist Control on Haptic System for Human Adaptive Mechatronics ", IEEE Conf. Decision Control, and European Control Conf, , pp. 4596-4600, 2005.
[58] Wolpert,D, Miall,M, Chris,R. and Kawato,M, " Internal models in the cerebellum ", Trends in Cognitive Sciences, Vol. 2, No. 9, pp. 338-347, 1998.
[59] Kleinman,D. Baron, L, Baron,S, and Levison,W.H, " An Optimal Control Model of Human Response Part I: Theory and Validation ", Automatica, Vol. 6, No. 3, pp. 357–369, 1970.
[60] Miall, R.C, Weir, D.J, Wolpert, D.M, and Stein, J.F, " Is the Cerebellum A Smith Predictor? ", Journal of Motor Behavior, Vol. 25, No. 3, pp. 203–216, 1993.
[61] Doya, K, Kimura, H. and Kawato, M, " Neural mechanisms of learning and control ", IEEE Control Systems, Vol. 21, No. 4, pp, 2001.
[62] Suzuki,S, and Furuta, K, " Adaptive impedance control to enhance human skill on a haptic interface system ", Journal Control Science Eng. , pp. 1-10, 2012.
[63] Chen,F.C. and Khalil,H.K, " Adaptive control of nonlinear systems using neural networks ", International Journal Control, Vol. 55, No. 6, pp. 1299-1317, 1992.
[64] Lewis,F.L, Jagannathan,S, and Yesildirek,A, Neural Network Control of Robot Manipulators and Nonlinear Systems , Taylor and Francis, London, 1992..
[65] Ge, S.S, Lee, T.H, and Harris,C.J, " Adaptive Neural Network Control of Robotic Manipulators, World Scientific. Singapore, 1998.
[66] Christodoulou, G.A. and Rovithakis M.A, " Adaptive Control of Unknown Plants Using Dynamical Neural Networks ", IEEE Trans. Systems, Man, and Cybernetics, Vol. 24, No. 3, pp. 400-412, 1994..
[67] Yeşildirek,A, and Lewis,F.L, " Feedback linearization using neural networks ", Automatica, Vol. 31, No. 11, pp. 1659-1664, 1995.
[68] Polycarpou,M.M, " Stable adaptive neural control scheme for nonlinear systems ", IEEE Trans. Automat. Control, Vol. 14, No. 3, pp. 447-451, 1996.
[69] Poznyak, A.S, Yu,W,Sanchez,E.N, Perez,J.P, " Nonlinear Adaptive Trajectory Tracking Using Dynamic Neural Networks ", IEEE Trans. Neural Networks, Vol. 10, No. 6, pp. 1402-1411, 1996.
[70] Rovithakis, G.A, " Performance of A Neural Adaptive Tracking Controller for Multi-Input Nonlinear Dynamical Systems ", IEEE Trans. Systems, Man, Cybern. Part A, Vol. 30, No. 6, pp. 720-730, 2000.
[71] Hunt, R. and Zbikowski, K.J, Neural Adaptive Control Technology , World Scientific. Singapore, 1996.
[72] Osburn, P.V, Whitaker,H.P, and Kezer,.A, " New Developments in the Design of Adaptive Control Systems ", Institute of Aeronautical Sciences, 1961.
[73] Parks, P," Liapunov Redesign of Model Reference Adaptive Control Systems ", IEEE Trans. on Automatic Control, 362-367, 1966.
[74] Åström, K.J. and Wittenmark, B, " A Survey of Adaptive Control Applications ", Proceedings of the 34th IEEE Conference On decision and Control,pp. 649-654,1995.
[75] Aseltine, J. and A. Mancini, S, " A Survey of Adaptive Control Systems ", Transactions on Automatic Control, Vol. 6, No. 1, pp. 102-108, 1958.
[76] Unbehauen, H, " Adaptive dual control systems: a survey ", Adaptive Systems for Signal Processing, Communications, and Control Symposium, , pp. 171-180, 2000.

[77] Filatov, N.M. and Unbehauen H, " Survey of Adaptive Dual Control Methods ", IEEE Proceedings Control Theory and Applications, Vol. 147, No. 1, pp. 118-128, 2000.
[78] Åström,Karl J. and Wittenmark, B, Adaptive Control, Pearson Education. India, 2001.
[79] Landau, I. D, " A Survey of Model Reference Adaptive Techniques Theory and Applications ", Automatica, Vol. 10, No. 4, pp. 353-379, 1974.
[80] Singh, S. P. (1992). Transfer of learning by composing solutions of elemental sequential tasks. Machine Learning, 8, 323–340.
[81] Sutton, R. S. (1988). Learning to predict by the methods of temporal difference. Machine Learning, 3, 9–44.
[82] Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning. Cambridge, MA: MIT Press.
[83] Sutton, R., Precup, D., & Singh, S. (1999). Between MDPS and semi-MDPS: A framework for temporal abstraction in reinforcement learning. Artifcial Intelligence, 112, 181–211.
[84] Parr, R., & Russel, S. (1998). Reinforcement learning with hierarchies of machines. In M. I. Jordan, M. J. Kearns, & S. A. Solla (Eds.), Advances in neural informationprocessingsystems,10(pp.1043–1049).Cambridge, MA:MIT Press.
[85] Morimoto, J. & Doya, K. (2001). Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. Robotics and Autonomous Systems, 36, 37–51.
[86] Dayan, P., & Hinton, G. E. (1993). Feudal reinforcement learning. In C. L. Giles, S. J. Hanson, & J. D. Cowan (Eds.), Advances in neural information processing systems, 5 (pp. 271–278). San Mateo, CA: Morgan Kaufmann.
[87] Doya, K. (2000). Reinforcement learning in continuous time and space. Neural Computation, 12, 215–245.
[88] Bellman RE (1957) Dynamic programming. Princeton, NJ: Princeton UP.
[89] Ghahramani Z, Wolpert DM (1997) Modular decomposition in visuomotor learning. Nature 386:392–395.
[90] Doya K, Samejima K, Katagiri K, Kawato M (2002) Multiple model-based reinforcement learning. Neural Comput 14:1347–1369.
[91] Chang Y, Ho T, Kaelbling LP (2003) All learning is local: multi-agent learning in global reward games. Paper presented at 17th Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, December.
[92] Russell S, Zimdars AL (2003) Q-decomposition for reinforcement learning agents. Paper presented at International Conference on Machine Learning, Washington, DC, August.
[93] Barto A. G., Sutton R. S. and Anderson C. W. (1983) Neuronlike adaptive elements that can solve difficult learning control ploblems. IEEE Trans on Systems, Man, and CyberneticsSMC-13, pp. 834–846.
[94] G. Tesauro TD-Gammon, a self-teaching backgammon program, achieves master-level play Neural Comput., 6 (1994), pp. 215-219.
[95] Fagg, A. H. (1993) Reinforcement learning for robotic reaching and grasping. In New Perspectives in the Control of the Reach to Grasp Movement (eds Bennet K. M. B. and Castiello U.), pp. 281–308. North Holland.
[96] P.R Montague, P. Dayan, C. Person, T.J Sejnowski Bee foraging in uncertain environments using predictive hebbian learning Nature, 377 (1995), pp. 725-728
[97] P.R Montague, P. Dayan, T.J Sejnowski A framework for mesencephalic dopamine systems based on predictive hebbian learning J. Neurosci., 16 (5) (1996), pp. 1936-1947

[98] K.J Friston, G. Tononi, G.N Reeke Jr., O. Sporns, G.M Edelman Value-dependent selection in the brain: simulation in a synthetic neural model Neuroscience, 59 (1994), pp. 229-243
[99] Daw ND, Doya K (2006) The computational neurobiology of learning and reward. Curr Opin Neurobiol 16:199–204
[100] Schultz W (1998) Predictive reward signal of dopamine neurons. J Neurophysiol 80:1–27.
[101] Schultz W, Dayan P, Montague PR (1997) A neural substrate of prediction and reward. Science 275:1593–1599.
[102] L. Valiant. A theory of the learnable. Communications of the ACM, 27, 1984.
[103] K. Pawelzik, J. Kohlmorge, K.R Muller, Annealed competition of experts for a segmentation and classification of switching dynamics, Neural Computation, 8 (1996) 340–356.
[104] S. Grossberg, Competitive learning: From interactive activation to adaptive resonance. Cognitive science, 11 (1) (1987) 23-63.
[105] J. A. Hartigan, M. A. Wong, Algorithm AS 136: A K-Means Clustering Algorithm, Journal of the Royal Statistical Society. Series C (Applied Statistics), 28 (1) (1979), 100–108.
[106] , M.C. Naldi, R.J.G.B. Campello, Comparison of distributed evolutionary k-means clustering algorithms, Neurocomputing, 163 (2015) 78-93.
[107] T. Kohonen, Self-organizing Maps, Springer, Berlin and Hiidelberg, 1995.
[108] F. Coleca, A.State, S.Klement, E.Barth, T.Martinetz, Self-organizing maps for hand and full body tracking, Neurocomputing, 47 (2015) 174-184.
[109] S. Grossberg, Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors & II: Feedback, expectation, olfaction, and illusions, Biological Cybernetics, (1976) 187-202.
[110] T. Frank, K. F. Kraiss and T. Kuhlen, Comparative analysis of fuzzy ART and ART-2A network clustering performance, IEEE Transactions on Neural Networks, 9 (1998) 544-559.
[111] K. Oweiss, R. Jin, Y. Suhail, Identifying neuronal assemblies with local and global connectivity with scale space spectral clustering, Neurocomputing, 70 (2007) 1728-1734.

Biographical Information

Bakur ALQAUDI received the B.Sc. degree in Electronics Commination and Electrical Automation from the Yanbu Indus- trial College, Yanbu, Saudi Arabia, and M.Sc. degree in Electrical Engineering focusing in Bio robotics, Control and Cybernetics from Rochester Institute of Technology, Rochester, NY, U.S.A., in 2008 and 2012, respectively. He is currently pursuing the Ph.D. degree with the University of Texas at Arlington, Arlington, TX, U.S.A. He joined Yanbu Industrial College as an Instructor, from 2008 to 2009, and received the King's scholarship for Gas and Petroleum track in 2009. His current research interests include physical human-robot interaction, adaptive control, reinforcement learning, robotics, and cognitive-psychological inspired learning and control.