Improved Attention Models for Memory Augmented Neural Network Adaptive Controllers

Deepan Muthirayan, Scott Nivison and Pramod P. Khargonekar

***Abstract*— We introduced a *working memory* augmented adaptive controller in our recent work. The controller uses attention to read from and write to the working memory. Attention allows the controller to read specific information that is relevant and update its working memory with information based on its relevance, similar to how humans pick relevant information from the enormous amount of information that is received through various senses. The retrieved information is used to modify the final control input computed by the controller. We showed that this modification speeds up learning.**

In the above work, we used a soft-attention mechanism for the adaptive controller. Controllers that use soft attention update and read information from all memory locations at all the times, the extent of which is determined by their relevance. But, for the same reason, the information stored in the memory can be lost. In contrast, hard attention updates and reads from only one location at any point of time, which allows the memory to retain information stored in other locations. The downside is that the controller can fail to shift attention when the information in the current location becomes less relevant.

We propose an attention mechanism that comprises of (i) a hard attention mechanism and additionally (ii) an attention reallocation mechanism. The attention reallocation enables the controller to reallocate attention to a different location when the relevance of the location it is reading from diminishes. The reallocation also ensures that the information stored in the memory before the shift in attention is retained which can be lost in both soft and hard attention mechanisms. Through detailed simulations of various scenarios for two link robot robot arm systems we illustrate the effectiveness of the proposed attention mechanism.

I. Introduction

Even though there is remarkable progress in machine learning, robotics and autonomous systems, humans still outperform intelligent machines in a wide variety of tasks and situations [1]. Central to the functioning of human cognitive system are several functions like perception, memory, attention, reasoning, problem solving, thinking and creativity. Thus, it is only natural to ask, can functions inspired from human cognition improve control algorithms? With this neuro-cognitive science inspiration, we recently introduced the concept of memory augmented neural adaptive controllers in [2], [3]. In this paper, we focus on the notion of attention in this setting.

Deepan Muthirayan and Pramod P. Khargonekar are with the Department of Electrical Engineering and Computer Sciences, University of California, Irvine, CA 92697. Email: {dmuthira, pramod.khargonekar}@uci.edu

Scott Nivison is with the Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL, USA. Email: scott.nivison@us.af.mil.

Supported in part by the National Science Foundation under Grant Number ECCS-1839429.

Attention is the state of focused awareness on some aspects of the environment. Attention allows humans to focus on sensory information that are relevant and essential at a particular point of time. This is especially important considering, at any moment of time, the human takes in enormous amount of information from visual, auditory, olfactory, tactile and taste senses. The human-inspired attention mechanism have been instrumental in machine learning applications such as machine translation [4]. Attention based models have also improved deep learning models like neural networks [5], reinforcement learning (RL) [6] and are the state-of-the-art in meta learning [7] and generative adversarial networks [8]. We believe that the idea of attention holds tremendous potential for learning in cyber-physical systems.

Inspired by human attention and the recent successes in deep learning, we explore attention models for control algorithms in a specific context. We consider the working memory augmented NN adaptive controllers proposed in our recent work [2], [3]. Here, the controller stores and retrieves specific information to modify its final control input. We showed that this modification speeds up the response of the controller to abrupt changes. Attention is relevant here because, the main controller uses an attention mechanism to read and write to the working memory. The natural question to ask is what is a good attention mechanism for such an application?

Our previous work [2] used a specific form of attention called soft-attention, where the central controller reads and writes to all locations in the working memory. Here, the extent to which the memory contents are erased and rewritten or contribute to the final read output depends on the relevance of the content at a particular location in the memory. While soft attention mechanisms can be effective by retrieving relevant information from all locations, the same feature can lead to loss of information.

In contrast, hard attention mechanisms [9] are mechanisms that read and write to only one location at any point of time. A controller that uses hard attention does not lose the information stored in other locations since at any point of time only one location is read from or updated. The disadvantage is that the controller can fail to shift attention when the current information becomes less relevant. In addition, this information can be lost due to continual modifications.

In this paper, we propose an attention mechanism that is a combination of *hard attention* and an *attention reallocation* mechanism. The attention reallocation mechanism allows the controller to shift attention to a different location when the current location becomes less relevant. This also allows the

memory to retain the information stored in it before the shift was forced by the reallocation mechanism. Thus, the mechanism we propose can overcome the limitations of prior hard and soft attention mechanisms.

The setting we consider is a well-studied NN adaptive control setting. This setting comprises of an unknown nonlinear function that is to be compensated. Typically, the neural network is used to directly compensate the unknown function. The literature on NN based adaptive control is extensive [10]–[17]. In the setting here, we consider nonlinear uncertainties that can vary with time including variations that are *abrupt* or *sudden*. The objective for the controller is to adapt quickly even after such abrupt changes. We make the assumptions that (i) the abrupt changes are not large and (ii) the system state is observable.

In Section II, we revisit the *Memory Augmented Neural Network* (MANN) adaptive controller proposed in our recent works [2], [3]. In Section III, we discuss the working memory interface and the proposed attention mechanism. Finally, in Section IV, we provide a detailed discussion substantiating the improvements in learning obtained by using the attention mechanisms proposed in this paper. We do not include the stability proof for the controller described here. The proof outlined in [2] can be trivially extended to the closed loop system discussed here.

II. Control Architecture

In this section, we briefly introduce the memory augmented control architecture for adaptive control of continuous time systems proposed in our earlier work [2]. The central inspiration behind this idea is the working memory in the human memory system.

Proposed Architecture: The general architecture is depicted in Fig. 1. The proposed architecture augments an external working memory to the general dynamic feedback controller. In our previous work, we specialized it to the specific controller that augments an external working memory to a neural network. In this paper, we focus on the attention models for the working memory. We propose several attention mechanisms for the controller. The novelty of our proposal lies in how the controller can reallocate its attention under certain circumstances.

Specialization to NN Adaptive Control: in neural adaptive control, control input u to the plant is computed based on state feedback, error feedback and the NN output. The control input is a combination of base controller u_{bl}, which is problem specific, the NN output u_{ad} and a "robustifying term" v [12], [18]. The final control input is given by

$$u = u_{bl} + u_{ad} + v. \tag{1}$$

Since, it is assumed that the system state is observable, the output of the plant is the system state. The control term u_{bl} is computed based on the output of an error evaluator, for example, the error between the system output and the desired trajectory in a trajectory tracking problem. The control term u_{ad} is the final NN output. This output is typically used to compensate an unknown nonlinear function in the system

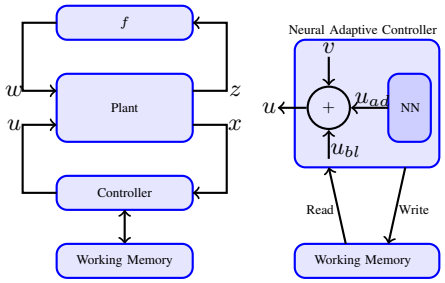

Fig. 1: Left: general controller augmented with working memory. Right: memory augmented neural adaptive controller. f is the uncertainty.

dynamics. The robustifying term is introduced to compensate the higher order terms that are left out in the compensation of the unknown function by the term u_{ad}.

In the memory augmented NN adaptive control we proposed in [2], the working memory acts as a complementing memory system to the NN. The output of the NN block, u_{ad} is computed by combining the information read from the external working memory and the NN. The exact form of this term and how it is modified based on the contents of the working memory is described in Section III. We showed in [2] that this modification speeds up the response of the closed loop system to abrupt changes.

Notation: The system state is denoted by $x \in \mathbb{R}^n$. The estimated NN weight matrices are given by $\hat{W}, \hat{V}, \hat{b}_w$ and $\hat{b}_v$. We denote the NN used to approximate the function $f(x)$ by $\hat{f}$, which is given by $\hat{f} = \hat{W}^T\sigma(\hat{V}^T x + \hat{b}_v) + \hat{b}_w$.

We introduce a function called softmax(.), takes in as input a vector a and outputs a vector of the same length. The ith component of softmax(.) is given by

$$\text{softmax(a)}_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)}. \tag{2}$$

We introduce two other vector functions which appear in the NN update laws. We denote these functions by $\hat{\sigma}$ and $\hat{\sigma}'$ which are defined by:

$$\hat{\sigma} = \begin{bmatrix} \sigma(\hat{V}^T x + \hat{b}_v) \\ 1 \end{bmatrix}$$

$$\hat{\sigma}' = \begin{bmatrix} \text{diag}(\sigma(\hat{V}^T x + \hat{b}_v) \odot (1 - \sigma(\hat{V}^T x + \hat{b}_v))) \\ \mathbf{0}^T \end{bmatrix} \tag{3}$$

where $\mathbf{0}$ is a zero vector of dimension equal to the number of hidden layer neurons.

III. Working Memory and Attention

In this section, we first summarize the two working memory operations, i.e., Memory Write and Memory Read, introduced in our previous works [2], [3]. We then provide a detailed discussion on the various attention mechanisms and the attention reallocation mechanism. Finally, we discuss how the Memory Read output is used to modify the final output u_{ad}.

A. Memory Write:

The Memory Write equation for the working memory is

$$\dot{h}_i = -w_r(i)h_i + c_w w_r(i)h_w + w_r(i)\hat{W}h_e^T, \tag{4}$$

where h_w is the write vector, $w_r(i)$ is the factor that determines whether the memory vector i is eligible for update or not and c_w is a constant. The write vector h_w corresponds to the new information that can be used to update the contents of the memory. The write vector h_w for this interface is given by,

$$h_w = \sigma(V^T\tilde{x} + \hat{b}_v). \tag{5}$$

Specifying the write vector to be the current hidden layer value allows the memory to store and reuse them later to modify the final NN output.

The factor $w_r(i)$s determine the elegibility of the respective locations for update and are determined by an addressing mechanism. The controller generates a query q which could be, for example, the current state or the current hidden layer output itself. Each memory location i is typically associated with a key k_i, which serve as its identifier. The keys k_is are compared with the generated query to deterimine the factors $w_r(i)$s. The factor $w_r(i)$ is set to be 1 for the i whose k_i is closest to the query q and the rest are set to zero. The location whose corresponding factor is one becomes eligible and is updated. In machine learning parlance this type of an addressing mechanism is referred to as *hard attention* [19].

Later, we discuss several attention mechanisms that differ based on how the keys are determined and how the query is specified.

B. Memory Read:

The Memory Read output, h_o, for the interface is given by

$$h_o = hw_r, \tag{6}$$

where $w_r(i)$s are the same set of factors discussed earlier. Essentially, the Memory Read output is a weighted combination of the respective memory contents with the weights being $w_r(i)$s. The factors $w_r(i)$s are a natural choice for the weights because they have a direct correspondence to the relevance of the information in their respective locations.

C. Attention Mechanism

We propose an attention mechanism that comprises of (i) a hard attention and (ii) an attention reallocation mechanism. The attention reallocation mechanism shifts the attention to a different location when the relevance of the information in the current location diminishes. After the shift, the information in the previous location is retained because in the period before the next shift the proposed mechanism uses hard attention. Mechanisms that use hard attention without attention reallocation can fail to shift attention and continue to read from and modify this information, eventually losing this information. Thus, attention reallocation plays a complementary role to hard attention. Below, we discuss two different hard attention mechanisms for the proposed controller (i) where the key is state based and (ii) where the key is representation based. We then discuss the attention reallocation mechanism.

1) State based Hard Attention: In this design, the keys are specified to be a set of points in the state space. The update equations for the keys are specifed to be an asymptotically stable first order dynamic system whose state is the key vector k_i and input is the sub-vector of the state $\underline{\mathrm{x}}$. Thus, the key update equations are given by:

$$\dot{k}_i = -c_k w_r(i)(k_i - \underline{\mathrm{x}}), \tag{7}$$

where the constant c_k is a design constant. The constant c_k determines the response time of this equation and is set such that the final output of this equation is a good representation of the states visited when its corresponding memory location was updated.

The query needs to be set such that the controller can retrieve values relevant to the current state of the system. Given that the keys are points in the state space, we set the query q to be the current state itself, i.e.,

$$q = \underline{\mathrm{x}}. \tag{8}$$

In the simulation examples that we discuss later $\underline{\mathrm{x}}$ is chosen to be the position variables in the state vector.

A hard attention mechanism selects the location whose key is closest to the query. In this design, this location is determined by

$$i^* = \text{argmin}_i \|q - k_i\|_\infty. \tag{9}$$

Given i^*, the factors $w_r(i)$s are naturally given by

$$w_r(i) = \begin{cases} 1, & \text{if } i = i^*, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

2) Representation based Hard Attention: Alternatively, we can specify the keys as a set of points in the hidden layer feature space of the neural network (NN). This is a reasonable approach because the hidden layer features by definition are a representation of the NN input space.

In this design, the key for the respective memory locations are given by

$$k_i = h_i. \tag{11}$$

We choose the memory vectors themselves as the keys for the following reasons. Firstly, the memory vectors provide a dynamic set of points in the feature space. Secondly, a key by design should contain information about the memory content and the scenario it represents and the memory vectors satisfy this criterion. The keys for the respective locations are formally given by

The query q should be such that the controller can retrieve information from the memory that is relevant. Assuming that the current scenario and the scenario that the memory contents correspond to are not very different, the information in the location that is likely to be relevant is the location whose key is closest to the current hidden layer output. The controller can retrieve this information by specifying the

query q to be the current hidden layer output of the NN itself. Hence, we define q as:

$$q = \sigma(V^T \tilde{x} + b_v). \tag{12}$$

In this design, the location that is selected by the hard attention mechanism is given by

$$i^* = \text{argmin}_i \|q - 1/c_w k_i\|_\infty. \tag{13}$$

The factor $1/c_w$ in the above equation accounts for the same factor in the Memory Write equation (4) which is also the key update equation in this case. Having defined i^*, the factor $w_r(i)$s are given by the same equation (10).

D. Attention Reallocation

The attention reallocation mechanism that we propose continually checks whether the content of any of the memory locations is nearer to the current hidden layer output. If, at some point, the hidden layer output deviates from any of the contents beyond this threshold θ then the controller reallocates attention to the least relevant location with its value re-initialized at the current hidden layer value. Such a design is likely to ensure that, at any point of time, there is at least one memory location whose content is 'very' relevant to the current scenario.

Attention reallocation also ensures that the memory does not forget the information stored before the shift in attention. This is because once the shifts occurs, the location where the information is stored is neither updated nor read, at least for a certain period, until the attention shifts back to this location. We note that, in both hard and soft attention, this information is likely to be lost.

The decision to shift attention is determined by

$$a_r = \begin{cases} 0, & \text{if } \exists\ i \text{ s.t. } \|\sigma(V^T \tilde{x} + b_v) - 1/c_w \mu_i\|_\infty < \theta, \\ 1, & \text{otherwise.} \end{cases} \tag{14}$$

When $a_r = 1$, it indicates that the attention is to be shifted. In this design, the location i_s that the attention is shifted to is given by

$$i_s = \text{argmax}_i \|\sigma(V^T \tilde{x} + b_v) - 1/c_w \mu_i\|_\infty. \tag{15}$$

The attention mechanism is initialized with the possible range of selections limited to just one location. The mechanism can expand this range to include other locations progressively if doing so could be beneficial. The decision to include new locations is specified by the same decision rule (14). This ensures that the controller starts with a limited set of memory locations and increases this set only when required. This can avoid unwanted jumps in attention, which could be the case with a mechanism that uses only hard attention.

E. NN Output:

The learning system (NN) *modifies its output* using the information h_o retrieved from the memory. For this memory interface, the NN output is modified by adding the output of the Memory Read to the output of the hidden layer as given below.

$$\text{NN Output: } u_{ad} = -\hat{W}^T \left(\sigma(\hat{V}^T \tilde{x} + \hat{b}_v) + h_o\right) - \hat{b}_w. \tag{16}$$

We postulated and showed empirical evidence in our earlier work [2] that such a modification improves the speed of learning by inducing the search in a particular direction, which facilitates quick convergence to a neural network that is a good approximation of the unknown function. For a detailed discussion on how memory augmentation improves learning we refer the reader to [2]. The computed NN output is fed to the controller (Fig. 1) which then computes the final control input by Eq. (1).

NN Update Law: The NN update law, which constitutes the learning algorithm for the proposed architecture, is the regular update law for a two layer NN [18]:

$$\begin{bmatrix} \dot{\hat{W}} \\ \dot{\hat{b}}_w^T \end{bmatrix} = C_w \left(\hat{\sigma} - \hat{\sigma}' \left(\hat{V}^T \tilde{x} + \hat{b}_v\right)\right) h_e - \kappa C_w \|e\| \begin{bmatrix} \hat{W} \\ \hat{b}_w^T \end{bmatrix},$$

$$\begin{bmatrix} \dot{\hat{V}} \\ \dot{\hat{b}}_v^T \end{bmatrix} = C_v \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix} h_e \begin{bmatrix} \hat{W} \\ \hat{b}_w^T \end{bmatrix}^T \hat{\sigma}' - \kappa C_v \|e\| \begin{bmatrix} \hat{V} \\ \hat{b}_v^T \end{bmatrix}. \tag{17}$$

The variable h_e in (4) is problem specific and depends on the Lyapunov function (without the NN error term).

IV. Simulation Results and Discussion

In this section, we consider several scenarios and show that the controller that uses the proposed attention mechanism results in improved performance. We also provide a detailed discussion on how the attention mechanism proposed here improves the performance. Finally, we provide a brief comparative study of the two dynamic key design approaches.

A. Robot Arm Controller

We briefly discuss the control equations for a typical robot arm controller augmented by an external working memory. The dynamics of a multi-arm robot system is given by, as in [12],

$$M(x)\ddot{x} + V_m(x, \dot{x})\dot{x} + G(x) + F(\dot{x}) = \tau, \tag{18}$$

where $x \in \mathrm{R}^n$ is the joint variable vector, $M(x)$ is the inertia matrix, $V_m(x, \dot{x})$ is the coriolis/centripetal matrix, $G(x)$ us the gravity vector, $F(\dot{x})$ is the friction vector and τ is the torque control input. Let $s(t)$ be the desired trajectory (reference signal), then the error in tracking the desired trajectory is

$$e(t) = s(t) - x(t). \tag{19}$$

Define the filtered tracking error by

$$r = \dot{e} + \Lambda e, \tag{20}$$

where $\Lambda = \Lambda^T > 0$. Then the system equations in terms of the filtered tracking error r, as given in [12], is

$$M\dot{r} = -V_m r - \tau + f, \tag{21}$$

where $f = M(\ddot{s}+\Lambda\dot{e})+V_m(\dot{s}+\Lambda e)+N(x,\dot{x})$ and $N(x,\dot{x}) = G(x) + F(\dot{x})$. Given f, it folows that the input to the NN, $\tilde{x} = [e, \dot{e}, s, \dot{s}, \ddot{s}]$. For this system, the control law,

$$\tau = -u = -u_{bl} - u_{ad} - v, \tag{22}$$

where $u_{bl} = -K_v r$,

$$v = -k_v(\|\hat{W}\|_F + \|\hat{V}\|_F + \|\mu\|_F + Z_m)r,$$

and u_{ad} is the NN output as defined in (16). The NN update laws are the same as (17). The vector $h_e = r^T$ in this case.

B. Two Link Robot Arm System

The system matrices for a typical two-link planar robot arm system are given below.

$$M(x) = \begin{bmatrix} \phi + \rho + 2\psi\cos(x_2) & \rho + \psi\cos(x_2) \\ \rho + \psi\cos(x_2) & \rho \end{bmatrix}, \tag{23}$$

$$V_m(x,\dot{x}) = \begin{bmatrix} -\psi\dot{x}_2\sin(x_2) & -\psi(\dot{x}_1+\dot{x}_2)\sin(x_2) \\ \psi\dot{x}_1\sin(x_2) & 0 \end{bmatrix}, \tag{24}$$

$$N(x,\dot{x}) = \begin{bmatrix} \phi\gamma\cos(x_1) + \psi\gamma\cos(x_1+x_2) \\ \psi\gamma\cos(x_1+x_2) \end{bmatrix}. \tag{25}$$

In the examples we consider here, the initial masses of the two links are set as: $m_1 = 0.8$ Kg, $m_2 = 2.3$ Kg. Their arm lengths are set as: $l_1 = l_2 = 1$ m. The parameters in the matrices, in terms of the link masses and the link lengths are: $\phi = (m_1+m_2)l_1^2 = 3.1$ Kgm^2, $\rho = m_2 l_2^2 = 2.3$ Kgm^2, $\psi = m_2 l_1 l_2 = 2.3$ Kgm^2, $\gamma = g/l_1 = 9.8$ s^{-2}.

The control algorithm parameters are set as: $c_w = 3/4, \theta = 0.2, K_v = 20, k_v = 10, \kappa = 0, C_w = C_v = 10$. The number of hidden layers, $N = 10$. To start with the number of memory vectors is initialized to one, i.e., $n_s = 1$. The memory is allowed to include up to a maximum of 5 memory locations. For the memory interface, dynamic representation based key is used. Later, we draw comparison between this approach and the dynamic state based key approach. In all the scenarios described below, the attention reallocation is on only during the initializing phase.

C. Illustration of Attention Reallocation

In the scenario we consider here, masses undergo the following sequence of abrupt changes:

$$\begin{aligned} &m_i \to 2m_i \text{ at } t = 10,\ m_i \to \sqrt{2}m_i \ t = 20, \text{ and} \\ &m_i \to 1/\sqrt{2}m_i \text{ at } t = 40 \end{aligned} \tag{26}$$

The link lengths are: $l_1 = 1, l_2 = 2$. The values of other parameters are as given above. In the first set of results the attention reallocation is on only during the initial phase (case 1). In the second set of results the reallocation mechanism is on throughout (case 2).

Case 1: Fig. 2 shows the response of joint angles for the controller that uses soft attention and the attention mechanism proposed here. Figure 4 shows the response of joint angles for the controller that uses hard attention and a regular NN controller with $N = 14$ hidden layer neurons. We find that the responses for the controller that uses soft attention and hard attention show large and sustained oscillations after every abrupt change compared to the mechanism proposed in this work. From the figures it is clear that this is caused by the large change and the oscillations observed in h_o after every abrupt change for these two controllers.

The attention mechanism we proposed in this work shows an improved performance for the following reason. In the initial phase, the mechanism reallocates attention if the memory content deviates beyond the thershold θ and continues to reallocate if the deviations continue to occur. This continues till the number of memory locations grows up to 5, the maximum that the memory can include. After this initial phase, the reallocation mechanism is turned off. But the information in the memory vector just before the reallocation is still retained and so the attention can shift back to this location if any subsequent jump exceeds the limit implied by this information. As a result, the Memory Read output is restricted from any deviation that exceeds the limit implied by the information in the memory contents even after the initial phase. This results in the diminished oscillations and thus the improved performance that we observe for the controller that uses hard attention and attention reallocation in the initial phase.

Case 2: Fig. 3 shows the response of joint angles for the controller that uses attention reallocation throughout. The response shows almost negligible oscillations unlike the previous case. This is clearly attributable to the reallocation mechanism being active throughout, which restricts the deviation of the memory contents much more than the previous case, as is evident from the h_o and σ plots shown in Fig. 3.

But, we observe that the initial peak after every abrupt change exceeds that of the controller that uses soft attention and the peak for the same controller in the previous case. This is observed because the quick correction provided by the second update term (or the third term) in the Memory Write equation (4), which necessarily results in a deviation, is restricted more than necessary in this case. Thus, having the attention reallocation on throughout reduces the oscillations but exaggerates the initial peak just after the abrupt change.

In the following, we present many scenarios to illustrate the effectiveness of the mechanism we proposed in this work. In all these scenarios, the major contributing factor to why the mechanism we proposed performs better is the explanation that we provided in this section.

1) Scenario 1: In this scenario, the values of the parameters are as given above. The attention reallocation is active only during the initial phase. It is set this way in all the scenarios we consider below.

The command signals that the two arm angles have to track are: $s_1 = \sin(0.5t), s_2 = 0$ and the masses undergo the

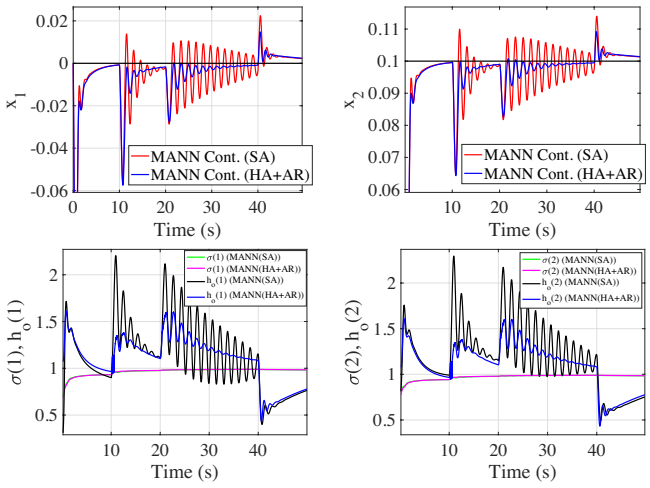

Fig. 2: Above: respone of first and second joint angle, x_1 and x_2, reallocation is on only during the initial period. Below: plot of σ, h_o.

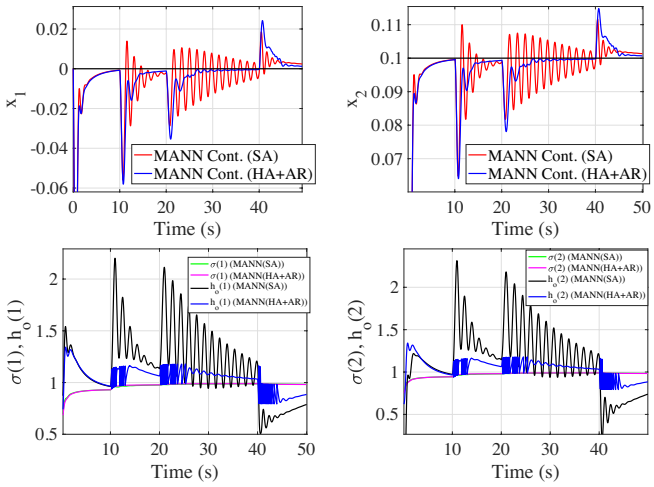

Fig. 3: Above: respone of first and second joint angle, x_1 and x_2, reallocation is on throughout. Below: plot of σ, h_o.

following sequence of abrupt changes:

$$
\begin{aligned}
&m_i \to \sqrt{2}m_i \text{ at } t = 5,\ m_i \to \sqrt{2}m_i \text{ at } t = 25,\\
&m_i \to \sqrt{2.5}m_i \text{ at } t = 50,\ m_i \to 0.63m_i \text{ at } t = 75,\\
&m_i \to \sqrt{0.5}m_i \text{ at } t = 90,\ m_i \to \sqrt{0.5}m_i \text{ at } t = 110,\\
&m_i \to \sqrt{0.1}m_i \text{ at } t = 130,\ m_i \to \sqrt{10}m_i \text{ at } t = 150,\\
&m_i \to \sqrt{2}m_i \text{ at } t = 170,\ m_i \to \sqrt{5}m_i \text{ at } t = 190,\\
&m_i \to \sqrt{0.2}m_i \text{ at } t = 210,\ m_i \to \sqrt{0.5}m_i \text{ at } t = 230,\\
&m_i \to \sqrt{0.1}m_i \text{ at } t = 250,\ m_i \to \sqrt{10}m_i \text{ at } t = 270,\\
&m_i \to \sqrt{2}m_i \text{ at } t = 290,\ m_i \to \sqrt{5}m_i \text{ at } t = 310. \qquad (27)
\end{aligned}
$$

In the results below, we compare the performace of this controller with the MANN controller discussed in our earlier work [2] (uses soft attention) and an equivalent NN controller. An equivalent NN controller is a controller that has the same number of parameters as the MANN controller, where the parameter count for the MANN controller includes

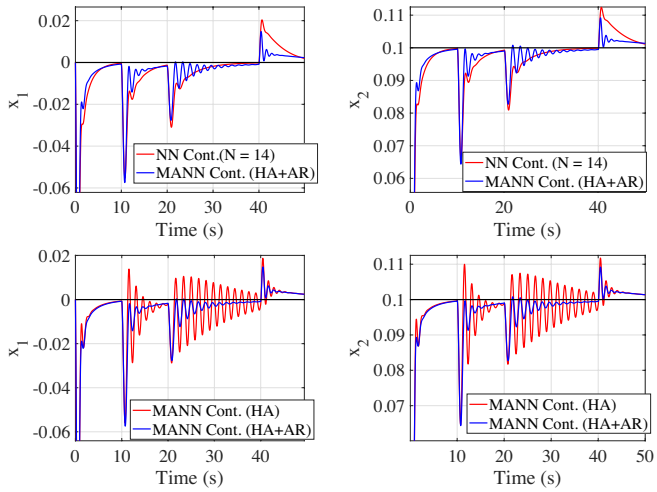

Fig. 4: Above: respone of first and second joint angle, x_1 and x_2. Comparison with a NN controller with $N = 14$. Below: respone of first and second joint angle, x_1 and x_2. Comparison with the MANN controller that uses hard attention.

the size of the memory, which is $n_s \times N$. For the robot arm system considered in this section, the NN controller that is equivalent to a MANN controller with $N = 10$ hidden layer neurons and $n_s = 5$ number of memory vectors is a network with $N = 14$ number of hidden layer neurons.

Table I provides the values of the sample root mean square error (SRMSE) for the two joint angles. The SRMSE values clearly indicate that the MANN controller version that uses hard attention (HA) and attention reallocation (AR) outperforms the other controllers by a notable margin.

TABLE I: Robot Arm System, SRMSE $\times 10^3$, Scenario 1

Joint Angle	1	2
NN Cont. (N = 14)	10.2	4.4
MANN Cont. (soft att., N = 10) - I	8.8	3.8
MANN Cont. (this paper, N = 10) - II	8.2	3.6
% Reduction (From I to II)	7.3 %	5.3 %

2) Scenario 2: In this scenario, the command signal for the joint angle 2, $s_2 = 0.1$, instead of 0. The masses undergo the same sequence of changes as given in (27). The MANN controller that uses soft attention is provided with $n_s = 5$ memory vectors and the equivalent NN controller is provided with a NN that has $N = 14$ number of hidden layer neurons. The rest of the setting for this scenario is similar to that of scenario 1. For lack of space, we do not provide the plots of the responses for this scenario. Table II provides the values of the sample root mean square error (SRMSE) for the two joint angles. We note that the proposed MANN controller outperforms the other controllers by a notable margin.

3) Scenario 3: In the scenarios considered so far the sequence of changes were periodic in nature. The scenario we consider here is the same as scenario 1 except that the abrupt changes the masses undergo result in a monotonic

TABLE II: Robot Arm System, SRMSE $\times 10^3$, Scenario 2

Joint Angle	1	2
NN Cont. (N = 14)	10	5.2
MANN Cont. (soft att., N = 10) - I	9.4	4.9
MANN Cont. (this paper, N = 10) - II	8.3	4.5
% Reduction (From I to II)	11.7 %	8.2 %

increase of the mass values. More specifically, we consider the following sequence of abrupt changes:

$$m_i^2 \rightarrow m_i^2 + 0.2m_i^2(0), \text{ after every } 20\text{s}. \tag{28}$$

Table III provides the values of the sample root mean square error (SRMSE) for the two joint angles. We note that, here too, the proposed MANN controller outperforms the other controllers by a notable margin.

TABLE III: Robot Arm System, SRMSE $\times 10^3$, Scenario 3

Joint Angle	1	2
NN Cont. (N = 14)	5.7	2.6
MANN Cont. (soft att., N = 10) - I	5.6	2.5
MANN Cont. (this paper, N = 10) - II	5.2	2.2
% Reduction (From I to II)	7 %	12 %

4) Scenario 4: In this scenario, the intial masses of the links are given by, $m_1 = 3$ and $m_2 = 2$. Rest of the setting considered here is the same as that of scenario 1. The parameters for the interface and in the control and update laws are also equal to the values used for scenario 1. Table IV provides the values of the sample root mean square error (SRMSE) for the two joint angles. The SRMSE values indicate that the proposed controller is only marginally better than the controller that uses soft attention in this scenario. We reported this scenario to illustrate the point that the improvements are scenario dependent.

TABLE IV: Robot Arm System, SRMSE $\times 10^3$, Scenario 4

Joint Angle	1	2
NN Cont. (N = 14)	12.0	3.6
MANN Cont. (soft att., N = 10) - I	10.2	3.1
MANN Cont. (this paper, N = 10) - II	10.0	3.0
% Reduction (From I to II)	2 %	3 %

5) Scenario 5: In this scenario, the setting is similar to scenario 1 except that the link lengths are different. The links lengths are: $l_1 = 1$ and $l_2 = 2$. The command signals are: $s_1 = 0$ and $s_2 = \sin(0.5t)$. Table V gives the SRMSE values for the three controllers. Here too, we observe that the controller that uses the attention mechanism proposed achieves a performance that is better compared to soft attention.

D. Comparison with Hard Attention

Here, we report several scenarios where the controller that uses the proposed attention mechanism is noticeably better than the controller that uses hard attention.

TABLE V: Robot Arm System, SRMSE $\times 10^3$, Scenario 5

Joint Angle	1	2
NN Cont. (N = 14)	11.0	7.6
MANN Cont. (soft att., N = 10) - I	9.8	7.4
MANN Cont. (this paper, N = 10) - II	9.3	6.8
% Reduction (From I to II)	5 %	8 %

First, we consider scenario 6. In this scenario the masses undergo the same set of abrupt changes as given in scenario 1. The arm lengths of the robot arm are: $l_1 = 1$, $l_2 = 2$. The command signals are: $s_1 = 0, s_2 = 0.1$. The threshold for attention reallocation, $\theta = 0.25$. Table VI gives the SRMSE values for the two joint angles. We only report the SRMSE values of the signals after the first 10s in order to report the values without the initial peak. We find that the mechanism proposed here shows a noticeable improvement in the performance. Similar responses are observed in scenario 4 when the command signals are: $s_1 = 0$ and $s_2 = 0.1$.

Next, we consider scenario 5. Table VII gives the SRMSE values for the two joint angles. The error values are reported for the responses starting from 10s to report the values without the initial peak. Clearly, the SRMSE values for the controller that uses the proposed attention mechanism show a noticeable improvement relative to the controller that uses hard attention.

We find similar performance improvements in scenarios 1 and scenario 4 when the link lengths are set as: $l_1 = 1$, $l_2 = 2$. In scenarios $1, 2$ and 3, the responses for the two controllers turned out to be similar. In scenario 4, we found the controller that uses hard attention to be marginally superior.

TABLE VI: Robot Arm System, SRMSE $\times 10^3$ ($t > 10$), Scenario 6

Joint Angle	1	2
NN Cont. (N = 14)	9.4	5.7
MANN Cont. (hard att., N = 10) - I	7.8	4.9
MANN Cont. (this paper, N = 10) - II	7.3	4.5
% Reduction (From I to II)	6.4 %	8.2 %

TABLE VII: Robot Arm System, SRMSE $\times 10^3$ ($t > 10$), Scenario 5

Joint Angle	1	2
NN Cont. (N = 14)	9.7	6.9
MANN Cont. (hard att., N = 10) - I	7.9	6.3
MANN Cont. (this paper, N = 10) - II	7.5	5.9
% Reduction (From I to II)	5 %	6.3 %

E. Discussion on Key Design Methods

In this section, we provide a comparative study of representation based key design and state based key design. Table VIII gives the SRMSE values for the two key design approaches in scenarios 1 and 4 respectively. It is clear that

the dynamic state based key design is better than dynamic representation based key design in scenario 4 but is worser in scenario 1. This suggests that a clear conclusion cannot be drawn on which key design is superior.

TABLE VIII: Robot Arm System, SRMSE $\times 10^3$

Joint Angle	1	2
scenario 1		
MANN Cont. (soft att.)	8.8	3.8
MANN Cont. (dyn rep key)	8.2	3.6
MANN Cont. (dyn state key)	8.5	3.7
scenario 4		
MANN Cont. (soft att.)	10.2	3.1
MANN Cont. (dyn rep key)	10.0	3.0
MANN Cont. (dyn state key)	9.5	2.9

V. Conclusion

In this work, we proposed a much improved attention mechanism for working memory augmented neural network adaptive controllers. The attention mechanism we proposed is a combination of a hard attention and an attention reallocation mechanism. The attention reallocation enables the memory to reallocate attention to a different location when the information in the location it is reading from becomes less relevant. This shifts the attention to a new location which is initialized with the most relevant information and also retains the older information in the location prior to the shift. Thus, the memory is able to overcome the limitations of prior soft and hard attention mechanisms. We showed through extensive simulations of various scenarios that the attention mechanism we proposed in this paper is more effective than the other standard attention mechanisms.

References

[1] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, 2017.

[2] D. Muthirayan and P. P. Khargonekar, "Memory augmented neural network adaptive controllers: Performance and stability," *arXiv preprint arXiv:1905.02832*, 2019.

[3] ——, "Memory augmented neural network adaptive controller for strict feedback nonlinear systems," *arXiv preprint arXiv:1906.05421*, 2019.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[5] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[6] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, p. 471, 2016.

[7] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," *arXiv preprint arXiv:1707.03141*, 2017.

[8] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.

[9] S. A. Nivison and P. Khargonekar, "A sparse neural network approach to model reference adaptive control with hypersonic flight applications," in *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 0842.

[10] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.

[11] A. Yeşildirek and F. L. Lewis, "Feedback linearization using neural networks," *Automatica*, vol. 31, no. 11, pp. 1659–1664, 1995.

[12] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 388–399, 1996.

[13] K. S. Narendra and S. Mukhopadhyay, "Adaptive control using neural networks and approximate models," *IEEE Transactions on neural networks*, vol. 8, no. 3, pp. 475–485, 1997.

[14] C. Kwan and F. L. Lewis, "Robust backstepping control of nonlinear systems using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 30, no. 6, pp. 753–766, 2000.

[15] A. J. Calise, N. Hovakimyan, and M. Idan, "Adaptive output feedback control of nonlinear systems using neural networks," *Automatica*, vol. 37, no. 8, pp. 1201–1211, 2001.

[16] L. Chen and K. S. Narendra, "Nonlinear adaptive control using neural networks and multiple models," *Automatica*, vol. 37, no. 8, pp. 1245–1255, 2001.

[17] S. S. Ge and C. Wang, "Adaptive neural control of uncertain mimo nonlinear systems," *IEEE Transactions on Neural Networks*, vol. 15, no. 3, pp. 674–692, 2004.

[18] F. Lewis, S. Jagannathan, and A. Yesildirak, *Neural network control of robot manipulators and non-linear systems*. CRC Press, 1998.

[19] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.