

Enhancing Robustness through Mechanical Cognitization

Gideon Avigad*, Wei Li[†], and Avi Weiss*

*Department of Mechanical Engineering
ORT Braude College of Engineering, Karmiel, Israel
Email: {gideon, avi}@braude.ac.il

[†]Department of Automatic Control and Systems Engineering
University of Sheffield, Sheffield, UK
Email: wei.li11@sheffield.ac.uk

Abstract—The common approach for training robots is to expose them to different environmental scenarios, training their controllers to have the best possible commands when untrained scenarios are encountered. When humans train they do the same. They try new manipulations by performing within different environments. However, humans training (and in fact development from infancy to maturity) also includes a type of training which, although claimed to improve cognitive capabilities, has not, to date, been adopted for the training of robots. This type of training involves the restriction of manipulation capabilities while performing different tasks, e.g., climbing with just one hand. Recently a research that facilitates functions instead of a mechanical systems that aims at exploring the invigorating idea that such training, would enhance the robustness of robots, has been published. This type of training has been termed as Mechanical Cognitization. In the current paper, the preliminary published results are detailed and more elaborated examples are given. Specifically, it is shown that the Mechanical Cognitization based training improves the performances when performing within untrained environments and when malfunctions occur. The advantages of the suggested training are highlighted through facilitating a comparison between two schemes that include a common neural net (with no training of restricted modes) and the recently introduced Mechanical Cognitization based neural net for which the training includes training of restricted modes. The results highlight the advantages of Mechanical Cognitization based training in enhancing robustness.

Keywords—Cognitive robotics, developmental robotics, evolutionary algorithms

I. INTRODUCTION

The use of robots in performing industry-related tasks and operating within hazardous environments has become ubiquitous. Yet robots are rarely used in everyday tasks that are performed mainly by humans. Indeed, humans exhibit truly amazing competencies in performing arduous and complicated tasks that may involve changing working conditions (scenarios), while controlling and maneuvering their multi-degrees-of-freedom body. By repeatedly executing different tasks, the human brain learns how to control the complex human body. Two human activities are of interest to the current research. The first is associated with training for participation in sports and athletic activities. For example, when training, climbers often use different techniques such as climbing with one hand tied behind the back, climbing sloping walls with no

hands, or climbing blindfolded. Clearly, such actual climbing conditions are not expected. Rather, these are all training techniques intended to improve climbers' sense of balance and movement skills. Restriction of movement as a training method is found in other sports as well, among them swimming (e.g., swimming with just one hand or without using the legs) and the martial arts (Fighting blindfolded). The second human training activity of interest here is also related to restricted movement and involves the way human capabilities develop from birth. The training of babies' minds begins on a body that is not yet fully developed. In contrast to new-born calves or horses, for example, human babies cannot stand, walk or run. Evolution has dictated a slow rate of development among humans and has forced the use of restricted capabilities. Could this be because in many situations, only some of the body's competencies are used so that the body must also be trained for these sub-manipulations? Note that many sports advocate starting young in order to let the body and mind adapt to the demands of the sport. In [1], we suggested exploring the novel idea of enhancing the robustness of robots by training them while taking into consideration both their final bodies/embodiments and their restricted modes (less capable versions). Such training has the potential to enhance the robustness of robots in performing untrained maneuvers as well as in coping with malfunctions and unexpected working conditions.

For elucidating the idea behind Mechanical Cognitization (MC), suppose that a robotic climber (CR) needs to be developed, so that it is capable of climbing a wall with poles sticking out of it. The left panel of Figure 1 depicts one possible mechanical configuration (body) for such a CR. The CR now must be trained to maneuver up and grasp one of the poles (A or B). The idea suggested here is that during the training of the controller of this CR, not only should this body be utilized but also its restricted modes. Two such restricted modes are depicted in the middle and right panels of the figure. Clearly, performing such a maneuver by utilizing one of the restricted modes might be associated with degraded performances (e.g., larger integral of the square error, measured while considering the planned and actual maneuver performed). Restricted modes may include the following restrictions: a) using only some of the mechanical capabilities, such as preventing some of the

links from moving – that is, if the robot has four arms/links, it will be restricted to use only two or three of them or will be prevented from using its gripper; b) restricting the movement of the arms/links to less than their full possible extent; c) deliberately imposing friction at the joints; d) changing the stiffness of the links; or e) restricting the actuator performance, for example by reducing the power supply to the actuators or using weaker actuators (smaller motors).

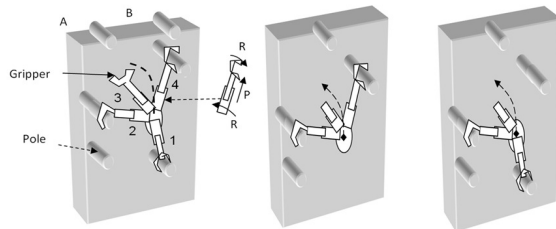


Fig. 1. CR having four links is trained using all of them (left panel) and using restricted modes (middle and right panels). This figure was from [1].

The work in [1] explains the idea, which is demonstrated by using functions. In the current paper, more elaborated examples are given. It is shown that the MC based training improves the performances when performing within untrained environments and when malfunctions occur. The paper is organized as follows. Section II discusses the needed background, which includes cognitive architectures and developmental robotics. The methodology is given in Section III. It includes the way to train and test the different MC related schemes. Next, in Section IV, the success of MC in enhancing robustness is demonstrated through using a basic example (in Subsection IV-A) and further elaboration (Subsection IV-B). A discussion and envisaged future work are presented in Section V.

II. BACKGROUND

Over the past several decades, a great deal of research attention has been directed at cognition and its implementation for artificial brains. The inspiration provided by human beings toward producing a machine that will copy human abilities is evident. Different models of cognition have been adopted to produce artificial cognitive systems or cognitive architectures. Cognitive architectures [2] represent attempts to create unified theories of cognition, i.e., theories that cover a broad range of cognitive issues, among them attention, memory, problem-solving, decision-making and learning. These theories consider several aspects, including psychology, neuroscience, and computer science. Examples of such architectures are the Soar system [3],[4], and ACT-R [5]. Some of these architectures have been claimed to be more adequate than others for use as cognitive brains for robots. This distinction (see, e.g., [6]), is rooted in the differences between the “cognitivist” and the “emergent” philosophies of cognition. The philosophy of emergent cognition contends that the relationship between the cognitive architecture and the body it is controlling (e.g., robots) is essential to the development of cognition, which is

not the case for the “emergent” philosophy. The current paper deals with the “emergent” philosophy, because the learning directly depends on the availability of models describing the controlled entity. An associated philosophy is embodied cognition [7],[8], which states that cognition can be influenced and biased by states of the body and that abstract cognitive states are grounded in states of the body. Among the architectures that facilitate this view is the biologically plausible brain-inspired neural-level cognitive architecture proposed by Shanhahan [9], in which cognitive functions such as anticipation and planning are realized through internal simulation of interaction with the environment. Burghart et al. [10] proposed a hybrid cognitive architecture for a humanoid robot that is based on the interaction of parallel behaviour-based components and a long-term memory sub-system utilizing a variety of representational schemas, including object ontologies and geometric models, Hidden Markov Models, and kinematic models. For a comprehensive survey of many of the approaches to model cognition and the resulting cognitive architectures, see [6].

Several approaches have been proposed to improve the response of artificial entities to specific stimulations by circumventing complex cognitive architecture. For example, the computational model of perception and action for cognitive robots discussed in [11] embraces the view that there is a direct route from perception to action that may bypass cognition [12]. A related approach is morphological computing (see, e.g., [13],[14],[15]), in which the idea is to design the mechanical structure to respond directly to a stimulus. This response is a result of the special morphology (shape, materials inter-relation among parts) of the structure. For example, in [16] the special features of a hand (Yoki hand) partially built from flexible deformable materials enable it to easily grasp different objects with no need for controller feedback. This notion has gained a great deal of interest, and for the past several years workshops have been dedicated to considering different aspects of morphological computing, such as artificial skin and stretchable sensors, compliant actuators and mechanisms, and soft materials in robotics.

In contrast the proposed research focuses on the enhancement of cognition by considering the mechanical structure, as is the case in morphological computing. Here, however, the cognitive architecture is of vital importance, and the mechanical structure and its possible restricted modes (permutations of the final structure) are utilized for training the cognitive architecture. This means that the mechanical structure is the driving force for the enhancement of cognition/learning. This enhancement of cognition by facilitating the mechanical structure of the robot has been termed as Mechanical Cognition (MC) [1].

Most relevant to the current paper are studies conducted by Mark Lee’s group at Aberystwyth, UK. Their research is related to Developmental Robotics [17]. According to this approach, which is rooted in the way babies develop, cognitive development is achieved through staged growth of cognition as the sensorimotor competencies are gradually and sequentially improved. In several publications [18],[19],[20] Lee’s group introduced and developed what they term as ‘constraint lifting’. At each stage, learning takes place with

certain constraints imposed on the sensomotoric system. At the next stage, some of these constraints are removed or ‘lifted’. For example, learning hand-eye coordination in manipulating a robotic arm has been investigated. In that case, as leaning progressed, constraints imposed on moving parts of the robot (e.g., using the fingers) were ‘lifted’.

The current paper and the MC idea involve several basic differences from the works such as [19]: a) In contrast to the sequential staged growth, MC may be enhanced simultaneously. b) In the proposed approach, constraining manipulations may take place any time along the robot’s life time and c) In one of the hereby proposed schemes, the knowledge gained through training in a restricted mode, may be preserved separately, and utilized when needed. As discussed above, cognition involves among other issues, learning, anticipation, conceptualization etc. The current paper deals just with the learning phase. For this reason and for the sake of focusing the current study on proving the applicability and impact of MC on robustness, we chose to utilize architectures, which are merely neural nets as was done in [1].

Previous studies on embodied cognition concentrated on how to construct a sophisticated artificial cognitive architecture, by utilizing one embodiment of the controlled entity. In contract here, we elaborate on the MC idea, which is aimed at fully exploiting the capabilities of any controller (artificial brain), by facilitating the learning of its embodiment including its related restricted modes (less capable embodiments). Here we further elaborate on the results attained in [1], through considering different combinations of restricted modes, allowing a more in depth look into the suggested learning approach.

III. METHODOLOGY

In order to elucidate the MC idea and to demonstrate its potential, mathematical functions are used here as was done in [1], instead of a CR or any other mechanical system. Representing the “environment” (the climbing wall) to which the CR has to adapt to (able to climb in the best way), is a polynomial function $Y(x)$, of order m where x is a vector of inputs (e.g., location of poles). In other words $Y(x)$ may be viewed as a planned route for the robot to follow. The CR’s controller is a neural net (NN) for which the outputs are coefficients of a polynomial of order n , $y(x) = a_1x^n + a_2x^{n-1} + \dots + a_n$. Each output may be viewed as a control signal (here it is a coefficient) to a motor of a manipulator that moves a robotic arm. The sum of the arm’s movements results (through kinematics) in the location of the CR on the wall. Here, this summing is represented by $y(x)$. The correlation between the CR case and the function representation, is summarized using Figure 2.

A. Training schemes

The training of the NN may be enhanced by, e.g., minimizing the square Error, $Error = (Y(x) - y(x))^2$ averaged over the available function’s points. In order to train the net to give an output, which is adequate to the environment (the function of order n), the artificial learning system was set as depicted in Figure 3.

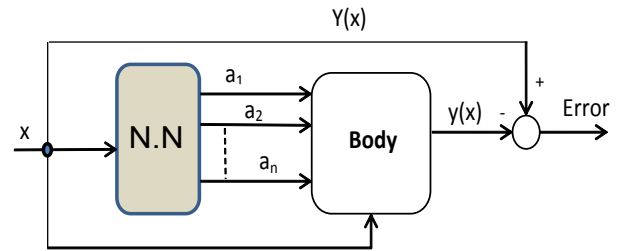


Fig. 2. The correlation between the CR and the related function representation.

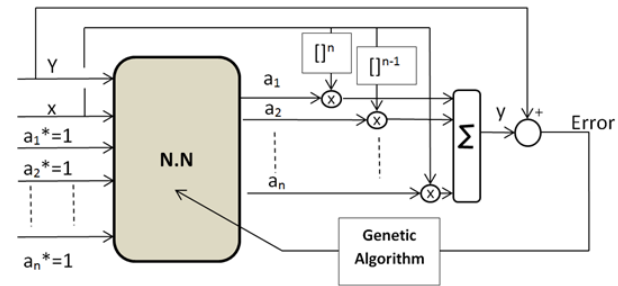


Fig. 3. The artificial NN with no restricted modes.

The input to the NN, is a list of k , x and corresponding $Y(x)$ values, which are fed sequentially to the net. The Net has extra n inputs (flags), namely: $A = [a_1^*, a_2^*, \dots, a_n^*]$. Each flag may be assigned a binary value. If $a_i^* = 1.0$, it means that the net’s i -th output is not prohibited and the related coefficient (link or DOF) participates in evaluating $Y(x)$. If $a_i^* = 0.0$, the i -th coefficient would be disregarded and the net is trained to produce just $n - 1$ outputs in order to still fit, in the best way, to the original function (environment), which is of order m .

Definition: If $\neg \exists a_i^* = 0$ then the net is training/operating in a non-restricted mode. If $\exists a_i^* = 0$ then the net is training/operating in a restricted mode.

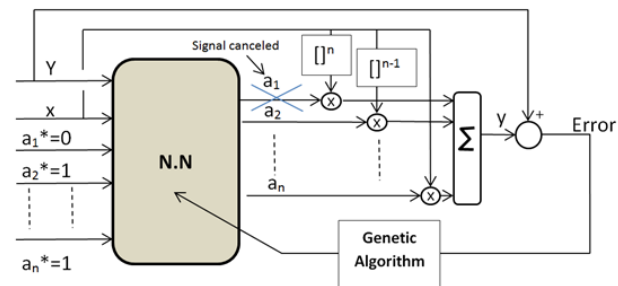


Fig. 4. The artificial NN training setting for a restricted mode.

While training, two NN schemes are considered. The first is merely the common NN, termed here as CNN, for which all flags’ values are one. The other training involves a series of training sessions, which include a training in the non-restricted mode as well as a series of different restricted modes training. The same net is trained for accommodating all of

the restricted modes and therefore this trained net is termed here as Amalgamated Modes Neural Net (AMNN). In [1], for each restricted mode, the training of the AMNN exploited not more than the system's available resources (K pairs). K/n of the inputs, were pairs fed to the net together with all the flags set to one, as was in the non-restricted mode. For the next K/n of inputs, the output a_1 has been prohibited and the corresponding flag was set to zero, $a_1^* = 0$ and so forth. In the current paper this approach is not maintained. Here, the training at each mode, facilitates the entire set of available examples. The reasons for this change include: *a*) It is possible that the number of available training points is limited, leading to insufficient training resources for each restricted mode, *b*) It is conceivable that training the restricted modes and the non-restricted modes would require more resources. When human train, it is acknowledged that for getting better (e.g., by climbing with one hand) more training time and commitment is mandatory. The training of the different neural nets was done using the $(\mu + \lambda)$ evolution strategy with self-adaptive mutation strengths [21], where $\mu = 50$ and $\lambda = 50$.

IV. TESTING THE SUCCESS OF MC IN ENHANCING ROBUSTNESS

Testing the different schemes for their robustness, is done here through considering two different uncertainties that involve untrained-for changes. These changes include, *a*) Malfunction: where one of the coefficients (robot's links/DOF) is prohibited. In such a case a more robust scheme would be the one for which the error with respect to the original function is smaller. This means that although malfunction occurs, the system is aiming at doing its "job" in the best way, and *b*) Environmental change: where the function is no more the original trained-for function (a new climbing route etc.). While testing the different schemes the following is assumed. The MC related neural net namely the AMNN has a feedback from the net's output (robot's links) such that if one or more malfunction, their related flags are changed to zero. In the case of the AMNN system, deliberately setting flags to zero for restricting movements along specific DOF, may be done. Decision on such a deliberate restriction may be done by a higher level controller that, e.g., uses vision to assess the accessibility to the target point. This means that a rational decision on, which of the modes to use may be done. When using just two fingers to lift a small object instead of using all fingers, humans are also using vision to make such a decision. For point *b* above another scenario may be envisaged. In such a scenario, manipulation takes place using the non-restricted mode. If the function is not satisfactorily estimated (target is not reached), another mode may be tried (the robot may retrieve to its initial configuration and retry). For the functions case, this means that the original function alters to a new one (new route) and a scheme that is more robust would be the one that adapts to the changed environment and acts to follow it with less error.

A. Example 1: Basic results

In this example an NN, which serves as the controller is to approximate a function of order two ($m = 2$), which

was arbitrarily chosen to be: $Y(x) = 3x^2 + 2x + 1$. The approximating function has been chosen to be of order three ($n=3$): $y = a_1x^3 + a_2x^2 + a_3x + a_4$. It is noted that justifying this redundancy for the current case is rather hard and for now the importance of redundancy may be only borrowed from the correlation to the fact that human mechanics is redundant. The CNN's NN is a forward neural network with five inputs (two for the x and corresponding $y(x)$ and three for the flags a_1^*, a_2^* and a_3^*), four hidden neurons (tansig activation functions were used), and four output neurons (for a_1, a_2, a_3 and a_4). For finding the best net, one hundred evolutionary runs, involving each 5000 generations, were run for each training mode. The best net is chosen such that the average of the fitness function, over all training points, is the smallest. Here $k = 10$ such that: $x = [0.2, 0.4, \dots, 2.0]$. Figures 5(a), 5(b) depict the approximation of the target function by the CNN and AMNN schemes, respectively.

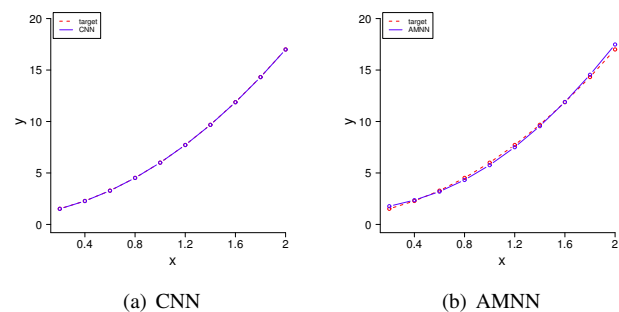


Fig. 5. This plot shows the performance of the a) CNN system and b) AMNN system with all flags set to be one.

It can be seen that the CNN approximation is somewhat better than that of the AMNN. This is not surprising because the the AMNN is trained using different restricted modes, some of which are not adequate for the function at hand (e.g., training when $a_2^* = 0.0$). The superiority of the CNN over the AMNN is further depicted in Figure 6, where the fitness value over 5000 generations of the evolutionary strategy run, is shown. As can be seen in the figure, the AMNN training needs more time to attain a reasonable good performance, while for the CNN, good performance is achieved rather quickly. Nevertheless, it will be shown that the merit of using the AMNN will be apparent when robustness to untrained scenarios would be tested.

1) *Malfunction in example 1:* The first test of robustness involves testing the robustness of the two schemes to malfunctions. For the current example a malfunction means prohibiting one of the DOF (one coefficient of $y(x)$ is set to zero). The performances of the two schemes are tested while both are to reach the training points. The performances of these schemes are compared using a box plot that are depicted in Figure 7. In all of the following box plots the line inside the box represents the median of the data. The edges of the box represent the lower and the upper quartiles (25-th and 75-th percentiles) of the data, whereas the whiskers represent the lowest and the highest data points that are within 1.5 times the inter-quartile range from the lower and the upper

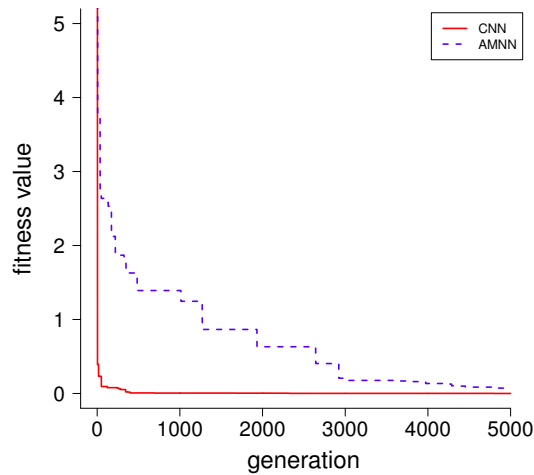


Fig. 6. The fitness of the best trained individual in each generation for the CNN and AMNN schemes.

quartiles, respectively. Circles represent outliers. In Figure 7, each box depicts the statistical data attained by one of the schemes over all training points, where the performances are the squared errors computed for each data point by: $Error(x) = (Y(x))^2 - (y(x))^2$.

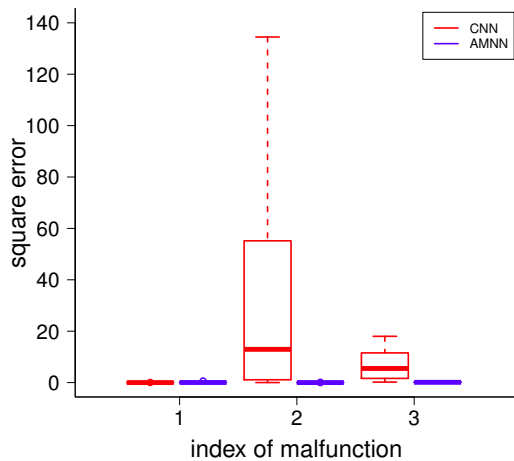


Fig. 7. The performance of CNN and AMNN system when malfunctions occur (index 1/2/3 corresponds to set the coefficients of third/second/first order of the function to zero). The p values for this figure (for the second and third DOF) are 0.0115 and 4.33e-05, respectively.

Restricting the first DOF (index “1” in Figure 7), namely a_1 , does not result in any superiority of one scheme over the others. This is not surprising due to this DOF being of higher order than needed. However, when prohibiting the other two DOF, a_2 and a_3 , which are designated by “2” and “3”, respectively, the enhanced robustness of the AMNN, is highlighted. This is especially profound when a_2 is prohibited.

The AMNN system performs significantly better than CNN system in the second and third case of malfunction (two-sided MannWhitney test, 5 percent significance level). Clearly, this advantage is built on training on restricted modes and preparing the scheme for such cases. Nevertheless, this training is just part of the training and the non-restricted mode is also part of the training.

2) *Environmental Changes in example 1:* For examining the robustness of the two schemes to environmental changes, the same x points as those of the training points are used, however, $Y(x)$ does not stay the same. For a mechanical system this would mean for the input, the output changes due to unexpected influences such as friction. To simulate such unexpected changes, the original function $Y(x)$ has been altered by changing its powers. The statistical data of changing the power two (second order) from one until three (skipping the original power, two) in steps of 0.2 and changing the power one (first order) from zero until two (skipping the original power, one), is depicted in Figure 8. Each data point in that figure is computed by computing the squared error between that function value at that point and the point reached by the scheme, averaged over all x points. The training followed the approach explained in Section IV.

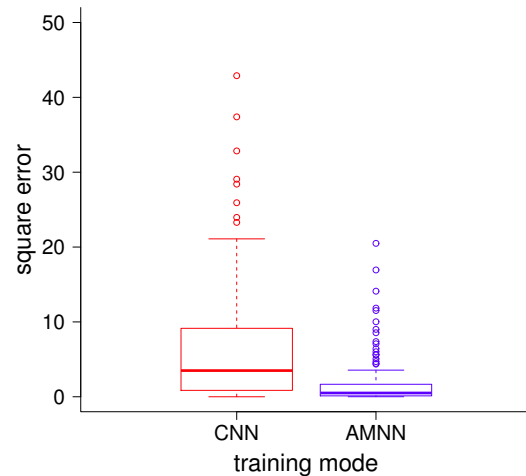


Fig. 8. The average fitness value of the CNN and AMNN system when changing the powers of the original function. The p value for this figure is 2.605e-11.

Depicting the results that are represented in Figure 8, it is clear that the MC related idea, which is realized through the AMNN scheme, promotes the robustness of the system when dealing with environmental changes as they are presented here (assuming the correlation between environmental changes and function changes).

For further testing the robustness to environmental changes, the original function is subjected to different changes, this time, instead of altering the powers, the coefficients are changed. In this case, the coefficients corresponding to the second order are changed from two to four in steps of 0.2, and the coefficients corresponding to the first order are

changed from one to three. There are total of $11 * 11 = 121$ combinations. The combination of three and two for the second and first coefficients, respectively, which corresponds to the original function, is excluded from the statistics (thus just 120 combinations are presented). The statistical data is presented in Figure 9. It can be easily seen that here again the AMNN performed significantly better than the CNN (two-sided MannWhitney test, 5 percent significance level).

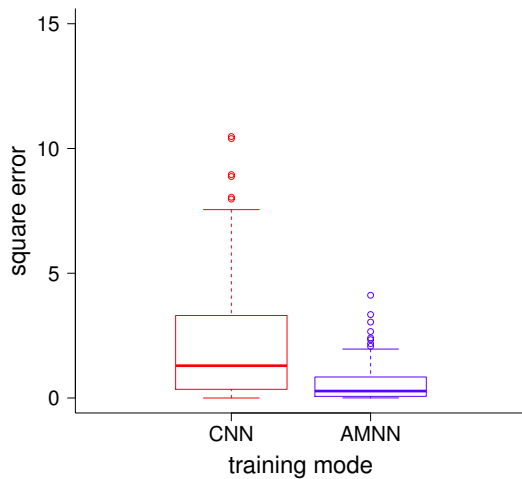


Fig. 9. The average fitness value of the CNN and AMNN system when changing the coefficients of the original function. The p value for this figure is $5.011e-11$.

For elucidating the enhanced robustness of the AMNN scheme, Figure 10 depicts the performances of the AMNN scheme, designated in green and the CNN scheme, designated by blue and the changed environment related function shown by the dotted red curve. Here the original (target) function has been altered to $Y = 3x^{1.5} + 2x^{0.5} + 1$.

The improved robustness of the AMNN is highlighted through its adaptation to the newly introduced function. Figure 11 depicts the same but for the case for which new coefficients are represented into the original function, namely, the function is altered to be $Y = 3.5x^2 + 2.5x^1 + 1$.

Again the competency of the AMNN scheme to respond to the needed changes, which is rooted in its ability to choose and activate/prohibit DOF, is highlighted.

B. Example 2: Elaborated results

In this section, the basic results, which were presented in Section IV-A, are elaborated and the approach is further tested. The main investigation concerns the effect of what restricted modes are utilized for the training phase. Consideration is given to the original CNN scheme and to three different training settings used for the AMNN scheme. The settings differ one from the other by the restricted modes that are used for the training. In all of these training settings, the non-restricted mode serves as one training mode and the other modes are as follows. AMNN1 is trained by a setting that is similar to the setting used for training the AMNN of

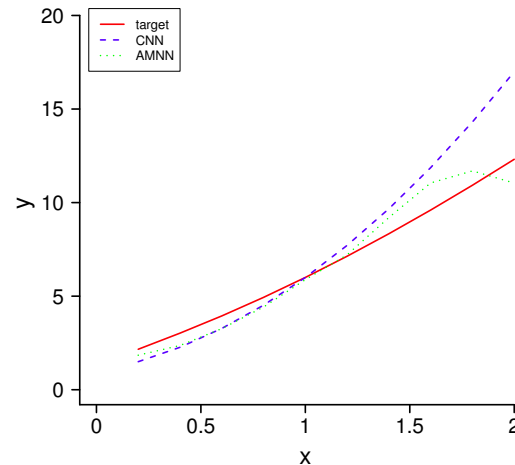


Fig. 10. One example showing how the CNN and AMNN system approximate a new function with different powers (simulating environmental changes).

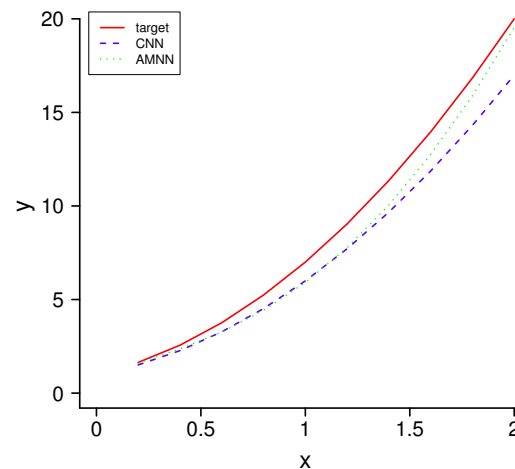


Fig. 11. One example showing how the CNN and AMNN system approximate a new function with different coefficients (simulating environmental changes).

Section IV-A. This means that three restricted modes are used ($k = 3$). In each of the restricted modes, one coefficient in y is cancelled. For AMNN2, each training involves restricting two modes. Because three coefficients are involved (a_1, a_2 and a_3 , there are three restricted modes that are trained (again $k = 3$). For the AMNN3 setting the trainings used for both the AMNN1 and the AMNN2 are amalgamated. This means that restricting both one coefficient and two coefficients is practised, resulting in six restricted modes training ($k = 6$).

1) *Normal Situation:* Figure 12 shows the output (blue curves) of the best individuals (with the lowest fitness value) in the 5000th generation for the CNN and AMNNs training in normal situation (In normal situation, all DOF of the robot are enabled, that is, all flags are set to 1.0.). Clearly, all train-

ing schemes approximate the target function in a reasonable accuracy. The fitness values of the the CNN scheme and the three AMNN's settings (AMNN1, AMNN2 and AMNN3) are $6.0 \cdot 10^{-5}$, 0.052, 0.125, and 0.072, respectively. The lower the fitness is, the better is the training. Therefore, the CNN training performs best in the normal situation. The advantage of the CNN scheme over all of the AMNN settings may be further highlighted by depicting Figure 13. The figure shows the fitness value of the best individuals in each generation during the evolutionary process for the CNN and AMNNs settings. As may be depicted, the fitness of CNN is always the lowest among the schemes in each generation. The training time required for the CNN is also much shorter than that of the AMNN settings. Comparing within the three AMNN settings, it may be observed that the fewer the DOF (i.e., k is smaller) is restricted during the training, the better average performance is attained.

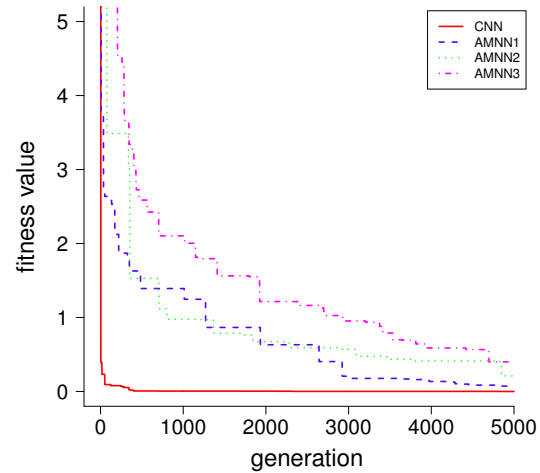


Fig. 13. This plot shows the fitness value of the best individuals in each generation of the evolution for the CNN and AMNNs training schemes.

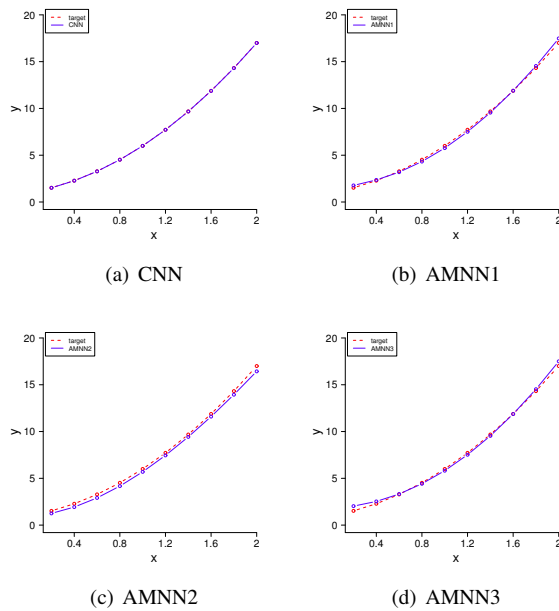


Fig. 12. This plot shows the output (blue curves) of the best individuals in the 5000th generation for the CNN and AMNNs training to approximate the target function. The red (dotted) curve represents the target function.

In the following, the robustness of the different schemes and settings are examined based on the two unexpected changes, which were discussed in Section IV, namely malfunction and environmental changes.

2) *Malfunction in example 2:* For testing the robustness of the different schemes and related settings to malfunctions, each of them will be tested for different malfunction combinations. This means that the performances will be evaluated by restricting one coefficient at a time as well as when prohibiting more than one coefficient. Figure 14(a) shows the performance of each scheme when malfunctions occur using box plot. When the third DOF malfunctions (a_1 is prohibited), all the training modes can still approximate the target function well, since this DOF is redundant. For all the other cases of malfunction, the CNN scheme does not perform well, as may be comprehended from the relatively large square error

shown in Figure 14(a). For the AMNN1, when only one DOF malfunctions, the system performs well. Surprisingly, it still performs very well when the third and first DOF malfunction (a_1 and a_3 are prohibited). Note that the system is not trained for this case. For the AMNN2 training, the system can handle more malfunction cases (1, 3, 4, 5 and 6). For the AMNN3, in all of the malfunction cases the trained system can perform well. It seems that the more degrees of freedom are restricted during the training, the higher robust the trained system is, at least when dealing with malfunctions.

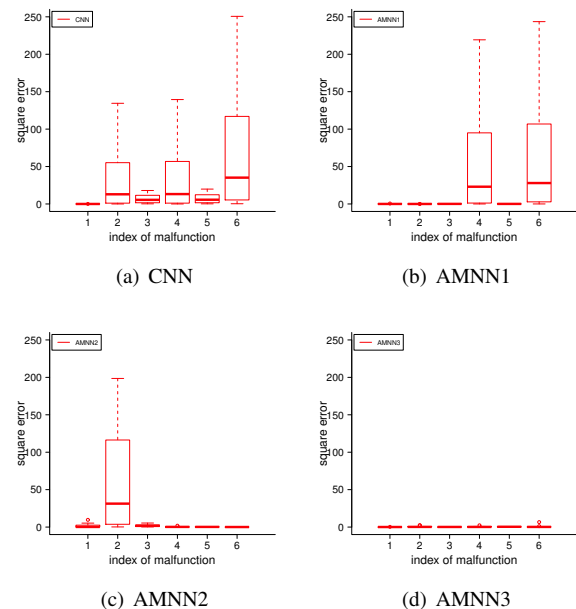


Fig. 14. This plot shows the performance of the trained system when malfunctions occur for the CNN and AMNNs training. 1/2/3: only the third/second/first DOF malfunctions. 4: the third and second DOF malfunction; 5: the third and first DOF malfunction; 6: the second and first DOF malfunction.

Generally saying, it is hard to envisage what would be the success of each training before analysing the statistical data. Moreover, it is rather unclear how to relate a success or failure of one specific training with respect to a specific DOF (e.g., the failure of AMNN2 with respect to the second DOF). Clearly, the non-linearity of the equations, the tansig function etc. hinders success in such a forecast.

3) *Environmental Change for example 2:* Here, the environmental change is simulated by changing the powers or coefficients of the trained function, i.e., the trained system has to approximate different, untrained-for, functions. Figure 15 and Figure 16 show the performance of the CNN and AMNNs training schemes when the powers and coefficients of the target function are changing, respectively. The x coordinates are the same as those in the 10 training points, but the y coordinates have changed depending on the testing functions.

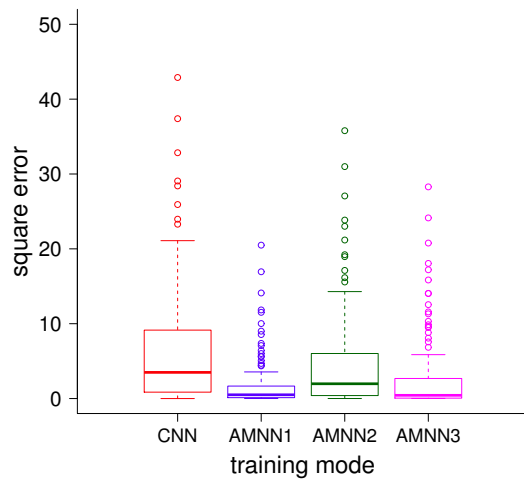


Fig. 15. This plot shows the performance of the CNN and AMNNs training when changing the powers of the target function.

For the case of changing powers, the second order of the target function is changing from $[1.0, 1.2, \dots, 3.0]$, and the first order is changing from $[0.0, 0.2, \dots, 2.0]$. We deleted the combination of 2.0 and 1.0, which corresponds to the same target function used for training. For each new testing function, we tested the performance of CNN and AMNNs using the new inputs. To approximate each point in the testing function, we selected the best mode operation that can obtain the least square error. For the CNN, we only have one operation mode. For the AMNN1, AMNN2 and AMNN3 training, the number of selected operation modes is four (one for the non-restricted mode and three for the one DOF restriction modes), four (one for the non-restricted mode and three for the two DOF restriction modes), and seven (one for the non-restricted mode three for the one DOF restriction and three for the two DOF restriction modes), respectively. For each testing function, we assigned an fitness, which is the average square errors for all the 10 points. The case of changing coefficients is similar. In this case, the coefficients corresponding to the second order is changing from $[2.0, 2.2, \dots, 4.0]$, and the

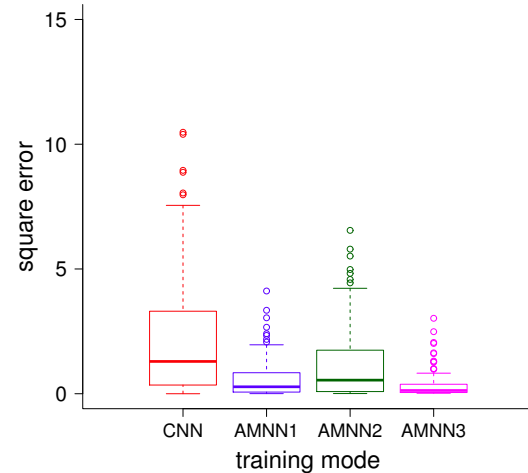


Fig. 16. This plot shows the performance of the CNN and AMNNs training when changing the coefficients of the target function.

coefficients corresponding to the first order is changing from $[1.0, 1.2, \dots, 3.0]$. We also deleted the combination of 3.0 and 2.0, which corresponds to the original, trained-for, function.

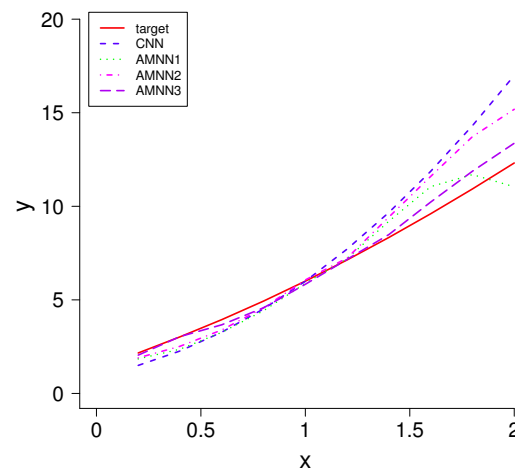


Fig. 17. This plot shows an example of how the CNN and AMNNs approximate a new function when changing the powers of the target function.

It is clear that the AMNN'S settings perform significantly better than that of the CNN, when environmental changes occur (two-sided Mann–Whitney test, 5% significance level). The p values are smaller than 0.02. In the case of changing powers and coefficients, the AMNN1 and AMNN3 outperforms the AMNN2 training. Although we find that restricting two degrees of freedom in the training (AMNN2) benefits more on the malfunctions comparing with restricting only one DOF. In the case of environmental changes, AMNN1 benefits more, as it has more degrees of freedom to approximate the new functions. Comparing the performance of AMNN1 and

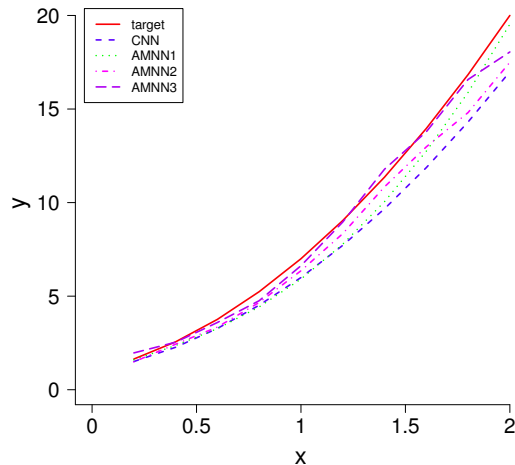


Fig. 18. This plot shows an example of how the CNN and AMNNs approximate a new function when changing the coefficients of the target function.

AMNN3, there is no significant difference in the case of changing the power, but AMNN3 performs significantly better in the case of changing the coefficients. From the results, we can see that the system trained with all combinations of restricted modes possesses the highest robustness.

Figure 17 and Figure 18 show an example of the two cases of environmental changes. The testing functions are still $y = 3 * x^{1.5} + 2 * x^{0.5} + 1$ and $y = 3.5 * x^2 + 2.5 * x^1 + 1$ respectively. As we can see, since the AMNNs have the advantage of choosing different operation modes to approximate the new functions, they perform better than the CNN scheme.

V. DISCUSSION AND FUTURE WORK

The Mechanical Cognitization idea has been further tested and elaborated here. Although the purpose of MC is the enhancement of robots' robustness, functions are still facilitated here. Clearly, using functions to prove a concept that is to enhance robustness of robots, which are not modeled as functions, requires a leap of faith on behalf of the reader. Nevertheless, much progress on the utilization of neural nets was instigated by using functions and therefore they are used as a fundamental base for our planned study. It has been shown in the paper that learning that includes both non-restricted modes as well as restricted modes, enhances robustness to environmental changes and to malfunctions. Although the training is done on the same training set as trained by the common scheme, the MC based schemes attain improved robustness. The improved robustness is attained through embedding a set of flags that open the way for deliberately restricting modes to attain improved performances. The enhanced robustness comes on the expense of extended training time. In the current paper, an insight is gained on the influence of choosing the restricted modes to be trained with. From analysing the results, it is evident that as more restricted modes are trained for, so does the robustness improves. As for future work, clearly

the next step would be proving the MC idea by utilizing a mechanical system such as manipulator/robot. Further research should take place for finding an optimization approach that will optimize the number of DOF (size of n) such that the robustness would be maximized. A multi objective approach could be also taken in order to maximize robustness and training time. Due to the contradiction among these objectives that was highlighted in the current paper, a trade-off set might be found. The utilization of neural nets as the controller's architecture should be also revisited and more sophisticated ones should be considered to serve as the cognitive, learning entity.

VI. ACKNOWLEDGEMENTS

This research was supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme.

REFERENCES

- [1] G. Avigad and A. Weiss, "Mechanical Cognitization," in *Proceedings of the 6th International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE 2014)*. IARIA, 2014, pp. 116–119.
- [2] A. Newell, "The knowledge level: presidential address," *AI magazine*, vol. 2, no. 2, p. 1, 1981.
- [3] P. S. Rosenbloom, J. E. Laird, and A. E. Newell, *The Soar papers: Research on integrated intelligence, Vols. 1 & 2*. The MIT Press, 1993.
- [4] D. E. Kieras and D. E. Meyer, "An overview of the EPIC architecture for cognition and performance with application to human-computer interaction," *Human-computer interaction*, vol. 12, no. 4, pp. 391–438, 1997.
- [5] P. Langley, "An adaptive architecture for physical agents," in *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE, 2005, pp. 18–25.
- [6] D. Vernon, G. Metta, and G. Sandini, "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 151–180, 2007.
- [7] L. A. Shapiro, "The mind incarnate," 2004.
- [8] —, *Embodied cognition*. Routledge London, 2011.
- [9] M. Shanahan, "Consciousness, Emotion, and Imagination," in *Proceedings of the 2005 AISB Workshop: Next Generation Approaches to Machine Consciousness*, 2005, pp. 26–35.
- [10] C. Burghart, R. Mikut, R. Stiefelwagen, T. Asfour, H. Holzapfel, P. Steinhaus, and R. Dillmann, "A cognitive architecture for a humanoid robot: A first approach," in *Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2005, pp. 357–362.
- [11] P. Haazebroek, S. Van Dantzig, and B. Hommel, "A computational model of perception and action for cognitive robotics," *Cognitive processing*, vol. 12, no. 4, pp. 355–365, 2011.
- [12] J. R. Simon and A. P. Rudell, "Auditory SR compatibility: the effect of an irrelevant cue on information processing," *Journal of Applied Psychology*, vol. 51, no. 3, p. 300, 1967.
- [13] R. Pfeifer, F. Iida, and G. Gómez, "Morphological computation for adaptive behavior and cognition," in *International Congress Series*, vol. 1291. Elsevier, 2006, pp. 22–29.
- [14] R. Pfeifer and G. Gómez, "Morphological computation connecting brain, body, and environment," in *Creating Brain-Like Intelligence*. Springer, 2009, pp. 66–83.
- [15] T. M. Kubow and R. J. Full, "The role of the mechanical system in control: a hypothesis of selfstabilization in hexapedal runners," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 354, no. 1385, pp. 849–861, 1999.
- [16] H. Yokoi, A. H. Arieta, R. Katoh, W. Yu, I. Watanabe, and M. Maruishi, "Mutual adaptation in a prosthetics application," in *Embodied Artificial Intelligence*. Springer, 2004, pp. 146–159.
- [17] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida, "Cognitive developmental robotics: a survey," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 1, pp. 12–34, 2009.

- [18] M. H. Lee, Q. Meng, and F. Chao, "Developmental learning for autonomous robots," *Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 750–759, 2007.
- [19] M. Hülse, S. McBride, and M. Lee, "Fast learning mapping schemes for robotic handeye coordination," *Cognitive Computation*, vol. 2, no. 1, pp. 1–16, 2010.
- [20] J. Law, M. Lee, M. Hülse, and A. Tomassetti, "The infant development timeline and its application to robot shaping," *Adaptive Behavior*, vol. 19, no. 5, pp. 335–358, 2011.
- [21] H.-G. Beyer, *The theory of evolution strategies*. Springer, 2001.