Evolving Robust Controller Parameters using Covariance Matrix Adaptation

Geruf K. M. Pedersen
Department of Psychology
University of Würzburg
97070 Würzburg, Germany
gerulf@psychologie.uni-wuerzburg.de

Martin V. Butz
Department of Psychology
University of Würzburg
97070 Würzburg, Germany
butz@psychologie.uni-wuerzburg.de

ABSTRACT

In this paper, the advantages of introducing an additional amount of tests when evolving parameters for specific purposes is discussed. A set of optimal PID-controller parameters are sought for an exemplary system, which simulates a human-like robotic arm. When evolving the controller parameters, the number of different movements included in the optimization process is varied. By including extra movements to the optimization process, the time it takes to evolve the parameters does increase, but the uncertainty due to noise is correspondingly lowered. Additionally, it is shown that the added movements, which improve robustness of the system, do not significantly lower the overall performance of the resulting system, when utilizing the evolved parameters.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Control theory*; I.2.9 [**Artificial Intelligence**]: Robotics—*Manipulators*; J.2 [**Computer Applications**]: Physical Sciences and Engineering—*Engineering*

General Terms

Algorithms, Design, Performance

1. INTRODUCTION

The continuing development toward more automated and advanced systems, both in industrial as well as research contexts, increases the need for more and better controllers. In order to live up to the demand for accuracy, those controllers must be either designed explicitly to a given part of the system or they must include some form of adaptive algorithm that can tune them according to their desired purpose. The major disadvantage of the former case is, that it is time consuming to design a new controller for every specific application. However, also the latter case of an adaptive algorithm is not without problems, as the adaptation algorithm needs to be flexible enough to handle the adaptation of controllers for a large variation of purposes. Also, configuring such an algorithm to optimize a given controller in the right way can be very difficult.

Another issue is that when controllers are developed for many systems today, the parameters are usually optimized using a model of the system. However, using models for optimization is not without problems. First of all, good models of various systems can be difficult to obtain if at all possible. Secondly, even if a good model is available, the methods for finding the optimal controller parameters may not apply appropriately, as most control design methods of the day still rely heavily on the model being linear, at least locally around the operating point for which the optimization is performed. This also applies to the many cases where a controller has been developed for a robotic system, e.g. when using LQG and LQR design methods[13, 8]. The impact of these limitations is that even though effective methods exist for optimizing controller parameters, the applicability of those methods are often limited.

In this paper, an adaptation algorithm will be used to optimize the parameters of several low-level controllers of a non-linear system, namely a simulated robotic arm. Of the many available optimization algorithms available, one algorithm that has proved itself successful repeatedly for various optimization problems is the Covariance Matrix Adaptation (CMA) algorithm [5]. It has previously been used for such varied applications as learning of neural structures in visuomotor control[11], complete-basis-function parametrization for shaping of laser pulses [10], filter design for sensor applications [1], optimization of dynamic molecular alignment[12], aerodynamic design [6], estimation of rotation parameters for a kinematic hand model [2], optimization of neural network structures [7], and for computing Nash equilibria [9].

The aim of this work is to use CMA to optimize the low-level controller parameters over the entire movement range of a simulated robotic arm and not just a few specific movements or movements limited to a subset of the arm's range. In connection with this optimization, it will also be investigated how the use of different movement regions during the optimization process will affect the final performance of the system. The performance when reaching for a specific target, located anywhere in the range of the arm, is not expected to outperform a dedicated controller that was trained specifically for such a movement. However, on average, for all possible targets in the range, it is expected that

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'10, July 7–11, 2010, Portland, Oregon, USA.
Copyright 2010 ACM 978-1-4503-0072-8/10/07 ...$10.00.

the evolved controller will perform better than any dedicated controllers.

The low-level controller parameters that will be optimized in this paper are the proportionate-, integral-, and derivative-parameters of the classical PID-controller.

In the next section, a detailed description of the simulated arm will be given, including details about the controllers used. In section 3 the algorithmic setup and the fitness function will be presented along with the considerations with regard to multiple start/goal combinations. Then the obtained results will be shown in section 4 followed by the conclusion in section 5.

2. SYSTEM DESCRIPTION

The arm that is used for the simulations in this paper is a 7DOF human-like arm, consisting of a 3 degrees-of-freedom (DOF) shoulder joint, a 2DOF elbow joint, and a 2DOF wrist joint. However, to keep the number of simulations at a feasible level, see 3.2, only the 4 major degrees of freedom are considered. The medial/lateral rotation of the shoulder joint, the pronation/supination of the elbow joint, as well as the radial/ulnar deviation of the wrist joint are ignored in this paper, leaving the adduction/abduction and flexion/extension movements of the shoulder, and the flexion/extension movement of both elbow and wrist joints. A sketch of the system configuration is shown in Figure 1.

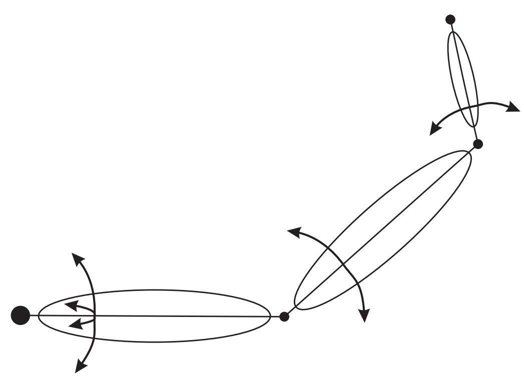

Figure 1: Sketch of the considered system with 4DOF (2DOF in the shoulder joint and 1DOF in the elbow and wrist joints).

The arm and controllers are simulated using appropriate kinematic and dynamic parameters in a physics-based simulator[1]. The simulator is not ODE-based, but uses the sum of forces and torques to calculate the movement of the individual links of the arm. The movement updates are performed at $1ms$ intervals and the effect of gravity was not included in any of the simulations.

2.1 Controllers

Each of the controllable joints of the arm is controlled by a classical PID-controller. The functionality of a PID-controller can be described as

$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de(t)}{dt} \ , \qquad (1)$$

[1]The length of upper and lower arm, l_1 and l_2, are 0.4m and the length of the hand, l_3, is 0.2m. The masses of the limbs are 4kg, 3kg, and 1kg respectively.

Parameter	Joint 1	Joint 2	Joint 3	Joint 4
K_p	50	50	40	10
T_I	1	1	1	1
T_D	0.08	0.1	0.06	0.05

Table 1: Initial controller parameters.

where $u(t)$ is the controller output at time t, $e(t)$ is the error between the actual and the desired system value, and K_p, K_i and K_d are weightings of the proportional, integral and derivative terms respectively. Depending on the choice of parameter values for K_p, K_i and K_d the controllers response to a given error signal will vary significantly.

For the use in the simulation, the controller equation has been discretized using the Backward rectangular rule. This was done in order to ensure that the performance of the controller in the system would not exceed what would be available to a real controller for a corresponding real life system.

3. CMA

The CMA algorithm used for this work is a Java implementation of the algorithm described in [4] which is freely available at `http://www.lri.fr/~hansen/cmaes_inmatlab.html`. The phenotype of the algorithm consists of the proportional gain, integral time, and rate time for the controllers of each of the four controlled joints, resulting in a total of 12 optimization parameters. The initial values of each parameter are based on values found manually, and they are listed in Table1, where joint 1 corresponds to the shoulder flexion/extension, joint 2 is the shoulder adduction/abduction, joint 3 is the elbow, and joint 4 is the wrist movement.

The only limitations on the parameters are that they must remain positive in order to assure correct functionality of the controllers. No stability checks or other limitations have been included.

CMA is allowed to run for a maximum of 500 iterations. This limit was chosen as no significant improvement was observed after that time in a number of experiments that were run for up to 5000 iterations. The rest of the CMA parameters use the default values.

3.1 Fitness Function

Defining the fitness function in the right way is key for obtaining a usable result from any evolutionary computation algorithm, and attempting to find the low-level controller parameters for a robotic arm using CMA is no exception. Any weakness in the fitness function will usually be exploited and results in a set of evolved parameters that are useless for the intended application.

The fitness function that was finally used for finding the controller parameters is given by the cost function

$$J(\vec{x}, \vec{u}) = \int_0^T \vec{x}(t)\mathbf{Q}\vec{x}(t)^T + \vec{u}(t)\mathbf{R}\vec{u}(t)^T dt \ , \qquad (2)$$

where $\vec{x}(t)$ is the vector of positional errors for all joints, $\vec{u}(t)$ contains the control effort exerted by the different joints, $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices, and T is the termination time of the simulation. The termination time of a simulation was set to $3500 \cdot d$, where d is the euclidean distance between initial and goal locations in posture space. The cost function is similar to those used when finding an LQR controller[3].

For all of the experiments in this paper the matrices **Q** and **R** were diagonal with identical entries of 1 and 0.003 respectively.

As several different start/goal combinations were simulated for each set of possible controller parameters θ, the costs for all combinations were summarized to yield the final fitness contribution

$$f_{tot}(\theta) = \sum_{i=1}^{N} J_i(\vec{x}, \vec{u}) \ , \qquad (3)$$

with N specifying the total number of start/goal combinations.

3.2 Multiple Start/Goal Combinations

The fitness function used in this paper is based on multiple start/goal combinations, such that the evolved controller parameters will be able to perform well over the entire movement range and not only be specialized for one particular movement or movement within a limited range. Thus, it is attempted to trade a potential good performance within a limited range for a slightly worse performance that is applicable to the entire movement range.

As a result, the movement range of each joint is divided into one or more regions. The combination of these joint regions produce a set of goal regions for the entire arm, and it is then possible to include the movement to each of these regions from all the other regions, when finding the optimal controller parameters. An obvious limitation to this scheme is that an increase in the number of regions per joint will result in an exponential increase in the number of start/goal combinations. It is thus necessary to keep the number of regions per joint low, but not so low that the resolution of the movement range becomes too coarse.

To avoid the trap of overspecialization to specific movements, the start and goal locations within each region are chosen randomly at the start of each generation. This ensures that the fitness comparison within each generation will not be affected by the random sampling. However, for comparison between generations, it is expected that the sampling of this dynamically changing fitness landscape will have an effect.

To investigate the effect of the number of regions on the quality of the solution, a number of experiments with a varying number of regions are conducted. The simplest experiment tests movements between 2 regions which is achieved by splitting the movement range of one joint into two equally large regions. The joints, whose movement ranges are not split, have their start and target angles uniformly distributed over the entire range of obtainable angles by that joint. Another experiment is also conducted, where the movement range of another joint is split into two ranges, resulting in 4 distinct regions with 12 unique start/goal combinations when moving between them. Finally, another split yields 8 regions with 56 start/goal combinations.

3.3 Investigating Overfitting

During evolution of the parameters, for every 5 iterations of CMA, the best solution is subjected to some additional testing. The results of this testing are, however, kept separate from CMA and do not influence the optimization itself. This testing is used to further assess how CMA might be specializing the movement to the trained targets. As such, when the test is performed, two regions around each current target are designated, and 30 uniformly distributed targets in each region are chosen. By investigating how the fitness score for these random targets develops, it is possible to determine how specialized the movement is for the current targets compared to how the performance degrades for targets further away. The regions are specified as a fraction of the possible movement range of each joint, and are chosen to be 0.1 and 0.2 respectively. Thus, the random targets are located in a rectangular region around each current target. The random targets are uniformly distributed in these rectangular regions. In case one of these additional targets lie outside of the movement range of a given joint, the target is set to the movement limit that it exceeded. Thus, when training targets close to the extreme values of the joint ranges, the additional testing will not always be perfectly uniformly distributed around the target region.

4. RESULTS

The experiments are first performed on the simplest system, where only 2 start/goal regions are used to evolve the controller parameters.

4.1 2-Movement Experiment

For the experiments with 2 start/goal combinations, the movement range of the joint 1 (shoulder flexion/extension) is chosen for the split in order to generate the desired regions. Only movements between the regions are considered when optimizing the controller parameters and performing the extended testing. A total of 30 independent evolutionary runs were conducted for this experimental setting, and the resulting development of the fitness value for the best individual over time during these experiments can be seen in Figure 2.

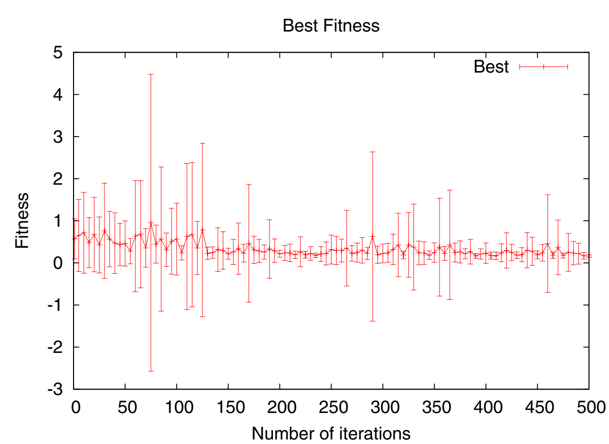

Figure 2: Development of mean and standard deviation of generational best fitness value for 2-movement experiments.

It can be seen that during evolution the average fitness value of the best individual does improve a little, but as the values fluctuate considerably, it cannot be concluded if a set of good controller parameters has been found by the algorithm. The found controller values also fluctuate considerably, which is shown for joint 1 in Figure 3. This makes it hard to draw any conclusion with regard to the quality of the parameters.

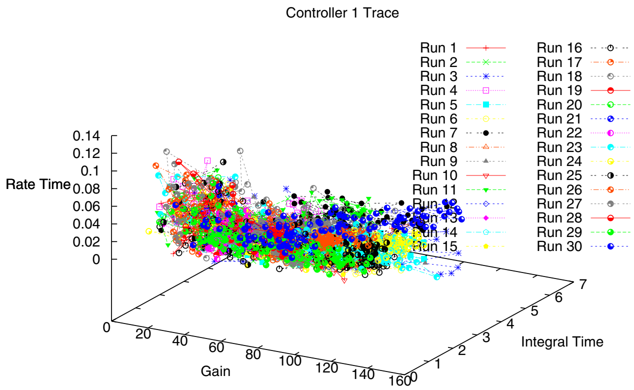

Figure 3: Development of controller parameters for the best individual for 2-movement experiments.

An investigation of the mean value of the extended tests in the defined test regions, shown in Figure 4, does indicate a slight learning tendency during evolution, but still the extensive fluctuations make it hard to draw any final conclusions.

Based on these results, an investigation of the need for further refinement of the start/goal regions seems obvious, and the set of tests using 4 different arm regions are presented next.

4.2 12-Movement Experiments

The 12-movement experiments are based on the 4 regions created when splitting both the movement of joint 1 and joint 3 into halves. For this combination of regions another set of 30 evolutionary runs are conducted, and the development of the best fitness value for these experiments is plotted in Figure 5.

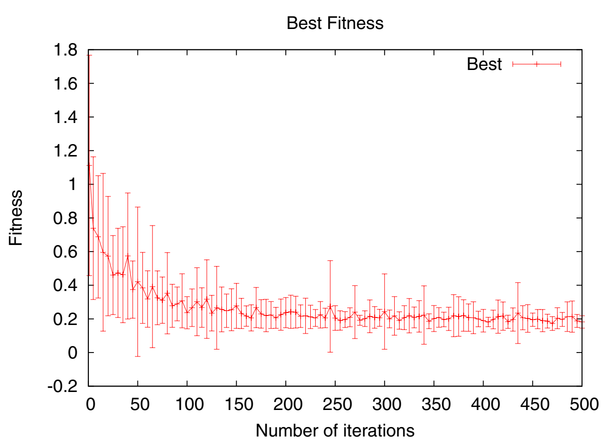

Figure 5: Development of mean and standard deviation of generational best fitness value for 12-movement experiments.

For these experiments the average of the best fitness value clearly improves as evolution progresses, and even though the standard deviation fluctuates somewhat, the fluctuations are small when compared to the average fitness value. In fact, they are significantly lower than for the 2-movement experiments. A decrease in fluctuations was expected due to the larger sample size, since each controller in this case is sampled 12 times compared to the 2 times in the previous experiment. So even though the fluctuations are lower it is hard to draw any final conclusions based on the available data, as the improvement may be due to either the increased number of samples for each controller or the movement between the added number of regions.

For these experiments, the controller parameters of the best individual are spread over a larger area in the space of possible parameter values. The combination of such a large variety of parameters to a relatively small range of fitness values would seem to indicate that the optimization problem is highly multi-modal, which would make it difficult for an algorithm such as CMA to converge to the same set of optimal parameters during different runs.

Also for this set of experiments, the additional tests in the specified test regions were performed and they are shown in figure 6.

These plots also clearly show a learning curve as evolution progresses. Although these plots also benefit from the increased sample size when it comes to lowering the fluctuations, the fact that both the mean value and its standard deviation are low does indicate that the extended number of tests was beneficial when optimizing the PID controller parameters for the system. To investigate this effect further, a set of experiments with 8 distinct reachable regions is performed next.

4.3 56-Movement Experiments

The 8 regions are obtained by extending the 4 regions used in the previous experiments into 8 by splitting the movement of joint 2 into half as well. Unfortunately, as the number of regions increase exponentially, so does the possible movements between regions increase correspondingly, resulting in a total of 56 different movements between regions. Due to the large number of movements and a non-trivial amount of computation necessary to perform the simulations, the number of tests with 56 movements are limited to 7. However, even with this limitation, the sampling of each set of controller parameters will once again have increased considerably.

The average fitness value of the best individual throughout evolution is plotted in Figure 7.

As expected, the fluctuations of the best fitness value has decreased when compared to the previous experiments. Aside from that, the average of the best fitness value corresponds to what was also found in the previous set of experiments. This indicates that using more specific movements for the optimization process does not cause the value of the fitness function to increase. Thus, the additional tests, which should assure an increased robustness of the controller, do not introduce a significant degradation of the controller performance.

The additional tests that were performed on the 56-movement experiments are plotted in Figure 8.

As expected, the fluctuations of the fitness value once again decreased. But even though the mean values for the two test regions have actually decreased a little when compared to the previous experiment, it cannot be used to make any definite conclusions about potential improvements to

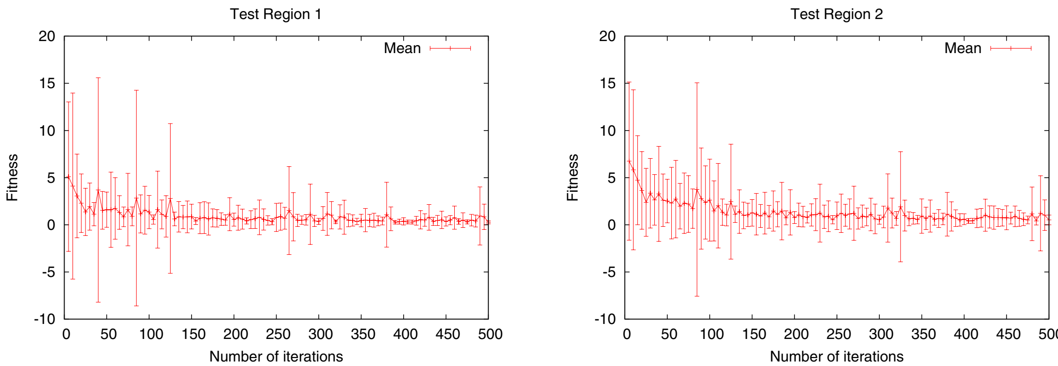

Figure 4: Average fitness value for individuals in small (left) and large (right) extended test region respectively, when moving between 2 distinct regions.

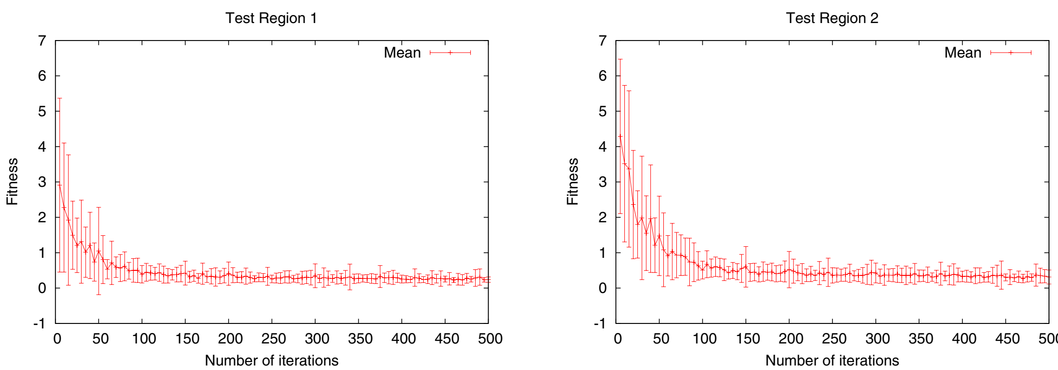

Figure 6: Average fitness value for individuals in small (left) and large (right) extended test region respectively, when moving between 4 distinct regions.

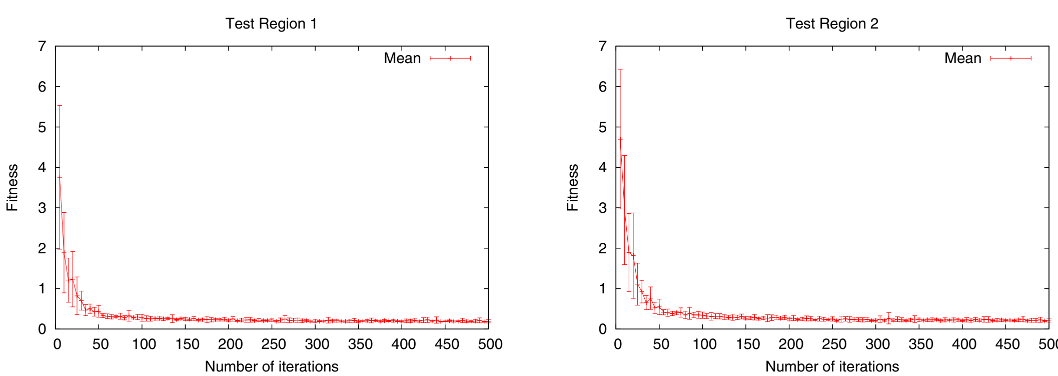

Figure 8: Average fitness value for individuals in small (left) and large (right) extended test region respectively, when moving between 8 distinct regions.

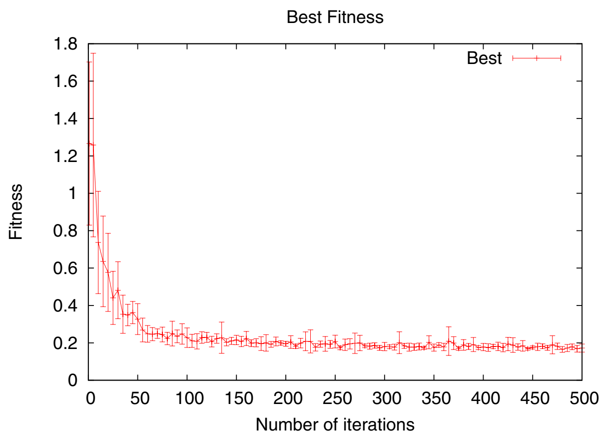

Figure 7: Development of mean and standard deviation of generational best fitness value for 56-movement experiments.

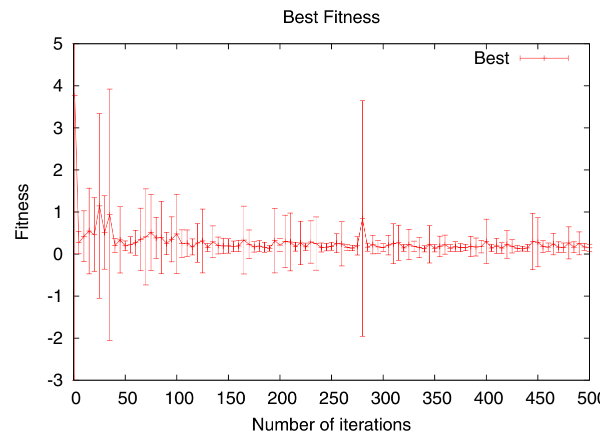

Figure 9: Development of mean and standard deviation of generational best fitness value for single sampling of the entire movement range.

performance, other than reducing the region of uncertainty when it comes to the performance of the evolved parameters. The additional tests also confirm the findings of the original fitness value used in the evolutionary process, that is, the added number of different movements does not seem to worsen the performance of the controllers when utilizing the evolved parameters.

To investigate, to what extent the improvement of the fitness values for more target regions might be caused by the increased sampling of each controller, some additional tests were conducted.

4.4 Investigation of Increased Sampling

In order to test for the effect of the increased sampling effect, two additional experiments are made. The first experiment is one in which only one movement is trained. This means that only one start/goal combination is tested for each controller, and these are found using uniformly random values. The other experiment also considers only one large region as before, but each controller is tested on two start/goal combinations. Thus it is possible to compare how training using a single movement compares to one with two movements as well as one with two movements between specific regions, as found in section 4.1.

4.5 Single Sampling

The single sampling experiment only uses one movement for finding the fitness value of a given controller. Similarly to the prior experiments, 30 runs using this setting is performed. The resulting development of the best fitness value for each generation is shown in Figure 9.

It is very hard to see that the average fitness value actually decreases over time, and that the system thus exhibits learning. Also, there are some large spikes, which would indicate that the dynamically changing fitness landscape, due to the random start/goal postures, does have an effect and punishes any over-specialized controller parameters that might have been evolved. However, as the mean fitness value recovers to approximately the same value after being subjected to the spikes, it is evident that the evolutionary process of CMA is capable of overcoming the challenge posed by this type of dynamically changing fitness landscape.

The resulting fitness values for the large extended test region throughout the evolution are shown in Figure 10. The results from the small extended test region are not shown as they was very similar to that of the large test region.

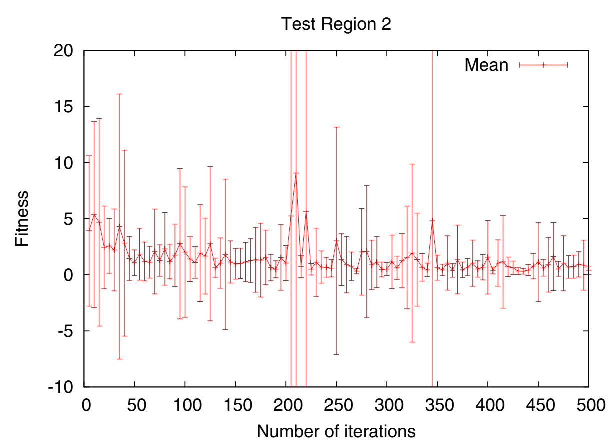

Figure 10: Average fitness value for individuals in the large extended test region, when moving within a single large region.

The results from the extended test regions show that the robustness of the evolved controller parameters are far from robust, as the performance obtained in these regions are extremely noisy. It would thus appear that CMA in some way accommodates too much to the specific movements.

Compared to the 2-movement experiments conducted earlier it can be seen that the noise of the fitness value for the 1-movement experiment generally is much lower, except for one extreme spike. As such, the introduction of the movement regions in this case does seem to have caused more difficulty for the evolution of good controller parameters. However, that was also what was originally expected, as the introduction of the movement regions forces CMA to evolve controllers for situations that are guaranteed to be more different than when the start/goal combinations are chosen completely at random. The extended tests, however, clearly indicate that the robustness of the evolved parameters is considerably worse than for the 2-movement experiments.

4.6 Double Sampling

The only difference when the double sampling experiment is compared to the single sampling experiment, is that each controller is tested on two start/goal combinations located within the entire movement range. The results after running CMA for 30 runs using this method are shown in Figure 11

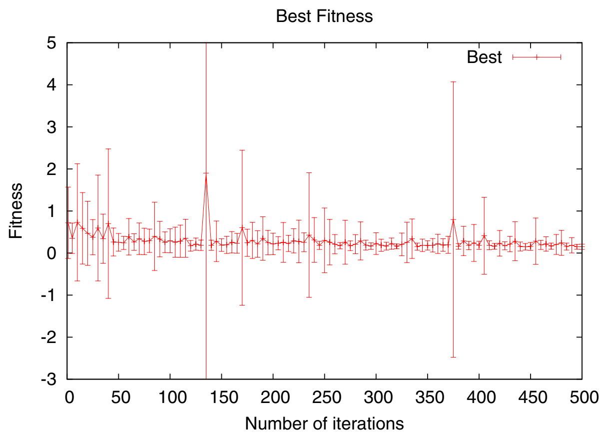

Figure 11: Development of mean and standard deviation of generational best fitness value for double sampling of the entire movement range.

For this case, the learning curve for CMA can be seen a little better. The noise levels when compared to the single sampling case does not seem to have changed significantly. Based on the obtained results it is thus not possible to determine whether additional sampling actually had a beneficial effect on the evolution of the parameters.

The results for the large extended test region are shown in Figure 12.

Again the figure shows that the evolved parameters are not very robust, as the noise level of the fitness values is very high.

When compared to the 2-movement experiments, the conclusions are thus the same as for the single sampling experiments. The noise for the 2-movement case, shown in Figure 4, is higher for the evolved fitness value than what was obtained for these double sampling experiments, so the added restriction caused more problems for the evolution of the parameters. However, for the extended test regions, the noise level is considerably lower for the 2-movement experiments, indicating that the robustness of the parameters found through the 2-movement experiments is better.

5. CONCLUSION

This paper has successfully evolved a varied set of low-level PID controller parameters for a 4DOF human-like arm. In fact, it was shown that the optimization problem, as specified by the fitness function, was highly multi-modal, resulting in a variety of controller parameters. Unfortunately,

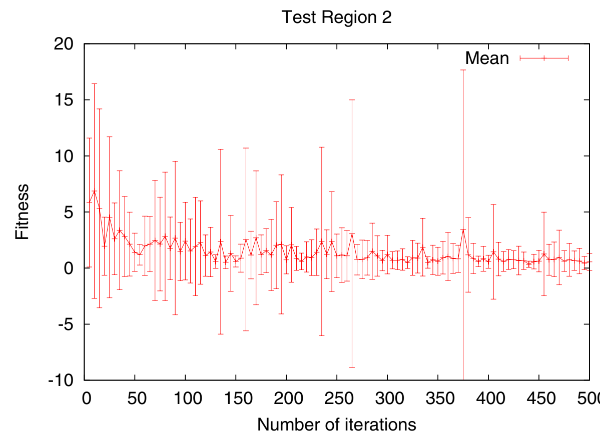

Figure 12: Average fitness value for individuals in the large extended test region, when moving within a single large region, but doubling the number of samples.

the fact that the parameters found during the different evolutionary runs covered such a large area of the parameter space puts a limit to the usage of evolutionary algorithms for this particular purpose, as evolution will not be capable of converging to one unique solution. This will continue to be the case unless the fitness function is modified in some way or constraints are added which can force convergence of the algorithm. Such constraints could specifically emphasize on small overshoots or low steady-state errors of the system. However, despite the large variety in the parameters, the overall performance of the system was found to be more or less consistent, depending on the number of movements included in the optimization process. So, the optimization was solved, but not with a unique solution.

Even though the example used in this paper was that of a simulated human-like robotic arm, whereto a set of optimal PID-controller parameters were sought, the obtained results are not limited to this particular system. Adding additional tests for different movement regions to the optimization of any system, will in such a case assure that the system obtains a larger degree of robustness when performing a given task. For the simulations in this paper, the sampling noise on the evolved fitness values were reduced significantly when including additional movements for the calculation of the fitness values. However, the cause for this improvement is most likely due to the considerable increase in the number of samples for each controller, even though it cannot be verified by the obtained results. Further investigation into this issue is currently underway.

Another significant finding was that the addition of extra movements between regions did not decrease the overall performance of the system, at least when using a large number of movements for determining the fitness of a controller. In fact, the overall fitness values actually improved a little when additional movements were included, when going from the 12-movement to the 56-movement experiments, but as the improvement was very small the topic remains open for further study.

The introduction of the movement regions, and forcing the algorithm to optimize movements between those regions, thus conclusively increased the robustness of the evolved controller parameters for the considered system.

6. REFERENCES

[1] S. Baskar, P. N. Suganthan, N. Q. Ngo, A. Alphones, and R. T. Zheng. Design of triangular fbg filter for sensor applications using covariance matrix adapted evolution algorithm. *Optics Communications*, 260(2):716–722, 2006.

[2] P. Cerveri, N. Lopomo, A. Pedotti, and G. Ferrigno. Derivation of centers and axes of rotation for wrist and fingers in a hand kinematic model: Robust methods and reliability results. *Annals of Biomedical Engineering*, 33(3):402–412, 2005.

[3] G. F. Franklin, J. D. Powell, and M. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, 3rd edition, 1998.

[4] N. Hansen. The CMA evolution strategy: a comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[5] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of 1996 IEEE International Conference on Evolutionary Computation*, pages 312–317, Nagoya, Japan, 1996. IEEE, IEEE Service Center.

[6] M. Hasenjäger, B. Sendhoff, T. Sonoda, and T. Arima. Three dimensional evolutionary aerodynamic design optimization with cma-es. In H.-G. Beyer, U.-M. O'Reilly, D. V. Arnold, W. Banzhaf, C. Blum, E. W. Bonabeau, E. Cantú-Paz, D. Dasgupta, K. Deb, J. A. Foster, E. D. de Jong, H. Lipson, X. Llora, S. Mancoridis, M. Pelikan, G. R. Raidl, T. Soule, A. M. Tyrrell, J.-P. Watson, and E. Zitzler, editors, *Genetic and Evolutionary Computation – GECCO-2005, Part II*, Washington, DC, 2005. ACM Press.

[7] M. Husken, Y. Jin, and B. Sendhoff. Structure optimization of neural networks for evolutionary design optimization. *Soft Computing*, 9(1):21–28, 2005.

[8] L. Márton, A. S. Hodel, B. Lantos, and J. Y. Hung. Underactuated robot control: Comparing lqr, subspace stabilization, and combined error metric approaches. *IEEE Transactions on Industrial Electronics*, 55(1):3724–3730, Oct. 2008.

[9] N. G. Pavlidis, K. E. Parsopoulus, and M. N. Vrahatis. Computing nash equilibria through computational intelligence methods. *Journal of Computational and Applied Mathematics*, 175(1):113–136, 2005.

[10] O. M. Shir, C. Siedschlag, T. Bäck, and M. J. J. Vrakking. The complete-basis-functions parameterization in es and its application to laser pulse shaping. In M. Keijzer, M. Cattolico, D. Arnold, V. Babovic, C. Blum, P. Bosman, M. V. Butz, C. Coello Coello, D. Dasgupta, S. G. Ficici, J. Foster, A. Hernandez-Aguirre, G. Hornby, H. Lipson, P. McMinn, J. Moore, G. Raidl, F. Rothlauf, C. Ryan, and D. Thierens, editors, *Genetic and Evolutionary Computation – GECCO-2006, Part II*, pages 1769–1776, Seattle, WA, 2006. ACM Press.

[11] N. T. Siebel, G. Sommer, and Y. Kassahun. Evolutionary learning of neural structures for visuo-motor control. In A. Kelemen, A. Abraham, and Y. Liang, editors, *Computational Intelligence in Medical Informatics*, Studies in Computational Intelligence, chapter 5, pages 93–115. Springer-Verlag, Berlin, Germany, 2008.

[12] C. Siedschlag, O. M. Shir, T. Bäck, and M. J. J. Vrakking. Evolutionary algorithms in the optimization of dynamic molecular alignment. *Optics Communications*, 264(2):511–518, 2006.

[13] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005 American Control Conference*, volume 1, pages 300–306, 2005.