

# Accelerating Robotic Assembly Parameter Optimization Through the Generation of Internal Models

Jeremy A. Marvel\*, *Student Member, IEEE*, and Wyatt S. Newman, *Senior Member, IEEE*

**Abstract-** The demand for the ability of robotic assembly solutions to self-tune is steadily growing as companies invest more in automation technologies for manufacturing. In this study we present a method that employs computational learning to generate internal models for online optimization acceleration of a Genetic Algorithms (GA) approach of exploring a parameter space for process optimization. By randomly sampling the “gene pool” parameter space, it is possible to successfully generate a mapping of high-dimensional input parameters to their respective resulting performances in the presence of noise, and then use this same map to improve the performance of the learning process by acting as a predictive filter for selectively choosing the child gene sequences most likely to produce improved assembly results. Results are given that demonstrate the advantages of internal model building as it relates to both virtual and physical trials.

## BACKGROUND

Computational intelligence approaches for online learning have shown incredible potential for autonomous assembly parameter tuning [1]. While simple fitness functions aimed at reducing assembly times and the subsequent contact forces are sufficient for online training, the process of optimizing assembly parameters is far from being efficient. Though a GA can be algorithmically tuned by adjusting its learning rates [2] and metrics of internal competition [3], it must first act upon the child sequences before deciding gene succession. Parameters, however ill-fitted for production requirements, must thus be run prior to being discarded. Hence online training can be exceedingly costly in terms of the time needed for convergence toward an acceptable solution and of any damage caused by testing sub-optimal parameters.

Prior attempts to make virtual models for assembly task tuning have relied on pre-built simulators for offline optimization (ex. [4]), or have generated three-dimensional surface mappings of the environment for pre-trial analysis (ex. [5]). These solutions, however, are expensive and time-consuming to generate, must be rebuilt for new product designs, and do not capture the gross uncertainties of the assembly process resulting from environmental fluctuations and part wear. Here we present a stochastic method for automatically optimizing assembly parameters aided by dynamically-generated internal models of the mappings of parameter-space sequences to their expected performances when applied to a robotic assembly task.

## CURRENT RESULTS

For the purpose of this study we define convergence acceleration as the positive motion toward some optimal performance solution in less time (either in terms of an absolute chronological measurement or as a tally of unique trials) than it would take to achieve the same performance with the normal model. Given that the model effectively

predicts the performance of the genes produced by the GA driver program and assuming that each trial has a constant evaluation cost  $c$ , by running only the  $K$  projected best-performing sequences of the  $M$  total genes otherwise necessary to achieve a given performance, we can expect an average convergence enhancement cost of  $cK/M$ .

Initial efforts focused on determining the extent of the benefit gained from simple surface estimates using 1) a numerical model of the linear tangent hyperplane fit of a query point to the data using a least-squared gradient descent method, and 2) a standard feed-forward artificial neural network (ANN) with back propagation. These methods were used to attempt to extract useful information of the mappings from parameter space inputs to their respective performances in both simulations and physical robot implementations. Without the assistive model, the GA generated and passed through the simulator 10 child gene sequences per generation. With the assistive model, the GA generated 1000 child genes, and only the 10 genes with the best predicted performances were executed.

Simulator trials consisted of arbitrarily large  $N$ -dimensional Gaussian curves with degrees of added noise (Fig. 1). The curve was centered at  $\mu = \{\pi, \pi, \dots, \pi\}$  with variance  $\sigma_i = i$  (i.e.  $\sigma = \{1, 2, \dots, N\}$ ) (Fig. 1-a). Perceptual noise was added in the form of an error value  $\epsilon$  dynamically generated upon each query to the simulator. Minor noise (Fig. 1-b) was generated as a random value with Gaussian distribution and variance 0.03, while major noise (Fig. 1-c) consisted of random values uniformly distributed in the range of  $[-0.5, 0.5]$ .

Without the presence of noise in the simulator, we could illustrate the benefit of omniscience on an otherwise potentially unknowable system. By using the simulator, itself, as a predictor of its own output we gained a “perfect” model of the simulator as a baseline for optimum convergence, and could likewise compute the first-order derivative of the simulator equation to provide an analytical version of the tangent of the curve’s surface. In contrast, we used the blind guesswork of the unassisted GA implementation as the baseline for expected convergence of a stochastic search. Plotted in Fig. 2 are the expected (i.e. average) performance results of executing the assistive models on a 15-D noiseless simulator. The perfect model and the derived analytical tangent predictably performed the best with measured speedups of  $c0.23$  each. The numerical tangent and ANN surface estimates fared slightly worse at  $c0.43$  and  $c0.57$  respectively.

When noise was added to the system we lost the ability to gauge the perfect model as a filter method. Ensuing plots, however, bear the perfect model curve for comparison. Minor noise (Fig. 3) resulted in a noted performance rank reversal of the gradient descent and ANN models, with the numerical tangent closely following the performance curve of the unassisted GA, and the ANN consistently outperforming both with a measured speedup of  $c0.81$ . Major noise (Fig. 4) saw both the gradient descent and ANN outperform the unassisted GA with speedups of  $c0.2$ , but the rate of convergence for the numerical tangent soon leveled near an optimum value of  $j \approx 0.6$  while the ANN continued to improve slowly to a value of  $j \approx 0.4$ .

Initial physical assembly trials were performed on an aluminum puzzle put together with an ABB IRB-140 industrial robot arm fitted with an ATI Gamma force/torque sensor (Fig. 5). The assembly consisted of a spiral search (stage 1) to engage the circular lip of the puzzle piece followed by a rotational search (stage 2) to engage the five-sided piece profile with the housing. Given the mediocre results

---

Both authors are with the Electrical Engineering and Computer Science Department at Case Western Reserve University, 10900 Euclid Ave., Cleveland, OH, 44106, USA.

\*Contacting Author: Jeremy A. Marvel is with the Electrical Engineering and Computer Science Department, Case Western Reserve University, 10900 Euclid Ave., 321 Glennan Building, Cleveland, OH, 44106, USA. (phone: 1-216-368-4104; email: jeremy.marvel@case.edu)

of the simulator trials, the gradient descent internal modeling method was omitted and only the ANN was utilized. Plotted in Fig. 6 are the raw and averaged performances of the GA execution with and without the assistive model for the second stage. The first three generations of tuning were executed without the internal model, and subsequent generations were run with or without assistance. The individual assisted trials performed as well as or better than the average performance of the unassisted GA, and the average assisted performance had a speedup of  $c0.75$ . The results of both simulations and physical trials suggest that, when omniscience is not possible, even simple models of high-dimensional assembly spaces stand to benefit automated parameter optimization methods.

### REFERENCES

- [1] J. Marvel, *et al.* "Automated Learning for Parameter Optimization of Robotic Assembly Tasks Utilizing Genetic Algorithms." Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics, February, 2009. Pp 179-184.
- [2] O. Gheorghies, H. Luchian, and A. Gheorghies. "A Study of Adaptation and Random Search in Genetic Algorithms." IEEE Congress on Evolutionary Computation, July, 2006. Pp. 2103-10.
- [3] P. Yip, and Y. Pao. "A Guided Evolutionary Computation Technique as Function Optimizer." IEEE World Congress on Computational Intelligence, 1994. Vol. 2. Pp. 628-633.
- [4] N. Yamanobe, *et al.* "Optimization of Damping Control Parameters for Cycle Time Reduction in Clutch Assembly." Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, August, 2005. Pp. 3251-3256.
- [5] S. Chhatpar and M. Branicky. "Localization for Robotic Assemblies with Position Uncertainty." Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, October, 2003. Pp. 2534-2540.

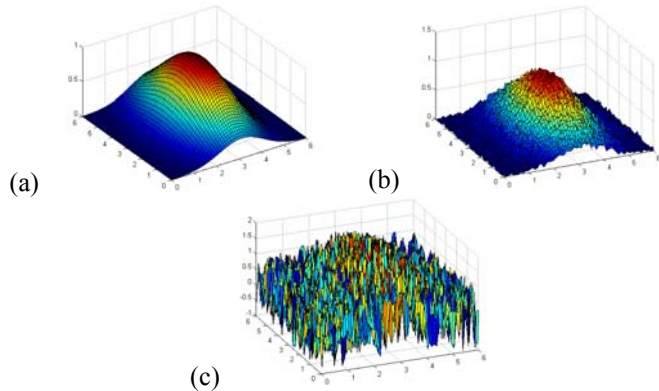


Fig. 1. Low-dimensional representations of the N-dimensional Gaussian curve simulator in (a) a noiseless environment, (b) with minor random noise with Gaussian distribution, and (c) with major random noise uniformly distributed.

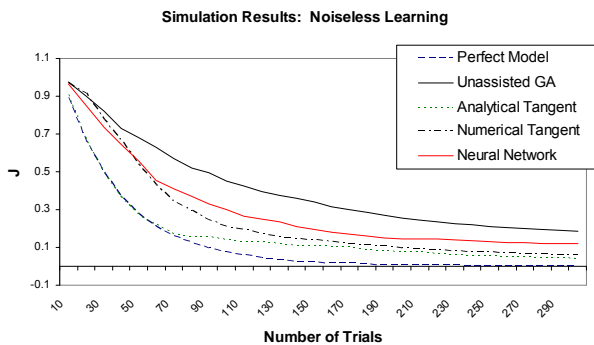


Fig. 2. Averaged results for a noiseless 15-dimensional simulated problem. The value of  $J$  indicates the difference between the curve apogee and the results of the best-performing parameter sequence at the specified trial count.

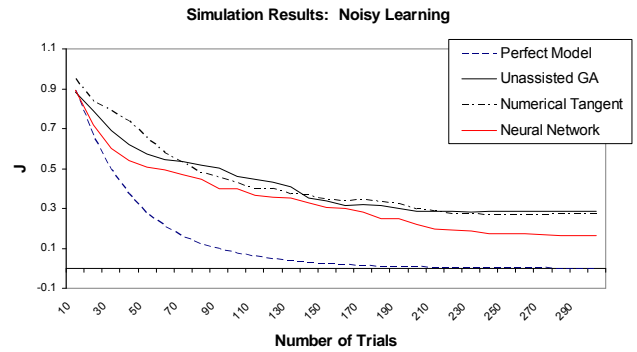


Fig. 3. Averaged enhancement results for a 15-dimensional simulated problem in the presence of minor random noise with Gaussian distribution.

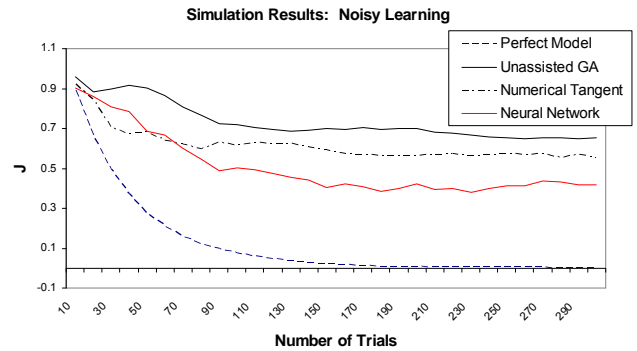


Fig. 4. Averaged enhancement results for a 15-dimensional simulated problem in the presence of massive noise with uniform distribution.



Fig. 5. An ABB IRB-140 industrial robotic arm (left) was configured to assemble an aluminum pentagonal puzzle insert (right).

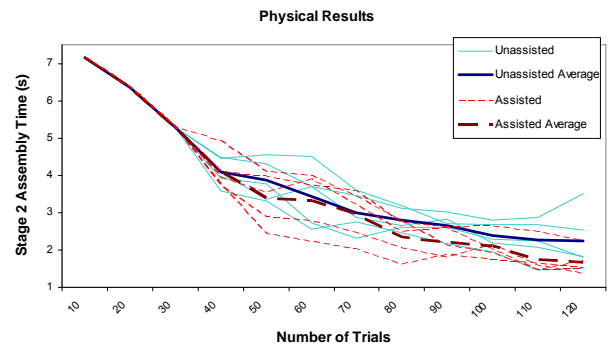


Fig. 6. Raw (light lines) and averaged (dark lines) results for a high-dimensional physical assembly problem with (dashed red) and without (solid blue) the benefit of internal modeling.