



HAL
open science

Neuroevolution with CMA-ES for the tuning of a PID controller of nonholonomic car-like mobile robot

Mohamed Outahar, Eric Lucet

► To cite this version:

Mohamed Outahar, Eric Lucet. Neuroevolution with CMA-ES for the tuning of a PID controller of nonholonomic car-like mobile robot. JDSE 2018 - 3rd Junior Conference on Data Science and Engineering, Sep 2018, Saclay, France. hal-04575330

HAL Id: hal-04575330

<https://hal.science/hal-04575330>

Submitted on 14 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Neuroevolution with CMA-ES for the tuning of a PID controller of nonholonomic car-like mobile robot

Mohamed Outahar and Eric Lucet

CEA, LIST, Interactive Robotics Laboratory, Gif-sur-Yvette, F-91191, France

Abstract. In the field of mobile robotics, finding an optimal control policy is a challenging task. PID controllers have been widely used in the industry. However, tuning a PID controller is not easy, especially to take into consideration the fluctuation in the precision of the perception. We propose a **neuroevolution** algorithm to find **the optimal parameters** of the controller in real time. The controller is tuned by a neural network which is trained by with the **covariance matrix adaption evolution strategy (CMA-ES)** . The neural network takes into account both the error and the uncertainty of the measurement the tuning of the parameters. The level of uncertainty in the measurement is given by the **the covariance matrix of the Kalman filter**.

Keywords: Neuroevolution, Machine learning, Neural network, Gradient-free optimization, Robotics, mobile robot, Control theory, PID controller

1 Introduction

In the last few years, neuroevolution [1] has gained interest in the research community. It has been shown to outperform reinforcement learning algorithms in certain situations where the search space is non-convex and noisy or the gradient is not available [2]. Neuroevolution describes a method to optimize neural networks with evolutionary algorithms. The algorithm is extremely parallelizable and scalable [3]. This document aims to demonstrate a method to automatically tune a PID controller of a car-like mobile robot using neuroevolution. The CMA-ES optimization algorithm was chosen to optimize the neural network. CMA-ES being also noted CMA with ES standing for Evolution Strategy which is a family of algorithms that is loosely based on biological evolution (hence the name). Multiple Evolutionary algorithms exist and they are all based on the same basic steps of population generation, evaluation, selection and reproduction. The CMA-ES has outperformed many algorithms in black box optimization problems [4]. That is why this algorithm has been chosen to tune PID controllers in many cases with promising results [5] [6].

As seen in figure 1, the system is composed of a robot controlled by a PID controller. The state of the robot is observed by an extended Kalman filter (EKF). The EKF provides the state \hat{x} and the corresponding covariance matrix

P. The core concept of the document is to use the covariance matrix with the corresponding error as inputs to a neural network which outputs the parameters of the controller in real time. Both of the CMA-ES blocks are used to define and optimize the neural network in order to adapt the behavior of the robot to the level of uncertainty in the measurements.

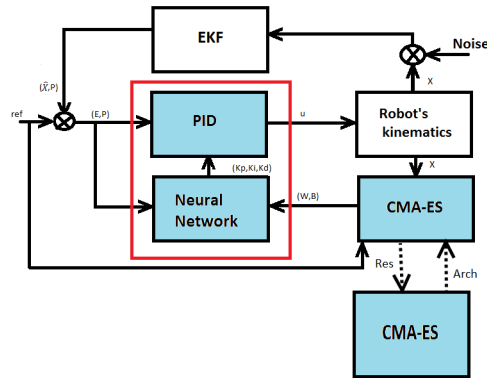


Fig. 1. Control bloc diagram.

2 PID tuning using a neural network

The goal here is to find the optimal parameters K_P , K_I and K_D to control the robot, by taking into consideration the error and the covariance matrix of the EKF. A neural network is used because it offers both adaptability and efficiency.

2.1 PID controller

PID controllers are widely used in the industry. This is due to their reliability and simplicity. PID has shown to have good preferences in multiple cases [7]. The general formula for the PID controller is as follows:

$$C(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{d}{dt} e(t) \quad (1)$$

with $e(t) = \text{actual}(t) - \text{target}(t)$

K_P , K_I and K_D are the proportional, integral and derivative gains respectively. Even though the controller is easy to implement, the tuning of its parameters is not a simple task and is a large area of research [8]. The three actions of proportional, integral and derivative have different and some concurrent effects. For example, the proportional term decreases the rise time while the derivative term increases it, while both are essential for the stability of complex systems. This is the reason of the difficulty of finding optimal gains.

2.2 Neural network

Neural networks are highly connected systems that are used to model complex, non-linear functions. In a simple representation of a neural network, the outputs of the each layer are multiplied by the weights and summed together with the biases and passed through the activation functions. Activation functions are what makes the system capable of modeling non-linear behavior, it can be represented graphically by neurons.

A big part of the progress done in this area is due to the backpropagation algorithm. This algorithm allows the neural network to learn patterns and desired behaviors. However The backpropagation algorithm uses the gradient to optimize the neural network. In this case, the gradient is not available, therefore the backpropagation algorithm can not be used.

3 Neuroevolution

A neural network is used to find the optimal parameters to control the robot efficiently, even with the presence of uncertainty. In traditional neural networks, the backpropagation algorithm is used to update the weights and biases. Here, an evolutionary algorithm is used instead. The choice was made because of the need of exploration in our problem and because neuroevolution is a gradient free method, which reduces execution time by orders of magnitude [2].

3.1 CMA-ES

The CMA-ES is an evolutionary algorithm [9]. it had been used because it outperformed most black box optimization algorithms. The algorithm starts off by generating a population of candidates. Those candidates are evaluated and put in order of fitness. From the top performing candidates, a percentage is selected to regenerate the new population. The new population is again reevaluated and the cycle continues until a termination condition is met. The termination condition is based on number of generations or the resemblance between parents and offspring.

3.2 Objective function

The CMA-ES algorithm takes an objective function as an input, and has the neural network parameters as outputs. the objective function is critical to the performance of the optimization. In our case it will be set to take into consideration the absolute error between the non noisy signals and the reference. In other words, the CMA-ES will tweak the neural network parameters in order to minimize the influence of the noise on the system. This is done to force the neural network to learn to control the system based on the level of noise (EKF's covariance matrix).

4 Results and perspectives

Multiple implementations varying in complexity, were realized for this work. At first a fix PID controller was optimized with the CMA-ES to adapt to fluctuations in the precision of the perception. After this initial phase, a neural network was used to tune a PID controller on line. The neural network was optimized by CMA-ES. The architecture of the neural network was chosen by the user. One of the latest Implementations describes the complete system, where both CMA-ES blocks work to have an optimal system.

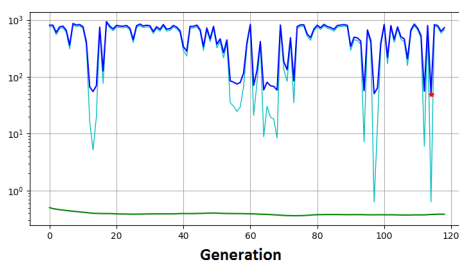


Fig. 2. Evolution of the objective function across generations. The size of the step between generations is displayed in green, the change in the objective function in cyan and the minimum objective function of each generation in blue. The red asterisk is the overall minimum objective function found by CMA-ES [10].

In figure 2 we see the evolution of the objective function throughout the generations. We notice that CMA-ES finds local optima but successfully overcomes them until the fix number of function evaluations has been achieved. As mentioned before the backpropagation algorithm can not be used. Therefore we can not compare the training phase of the two algorithms to evaluate the performance of the developed system. However we can evaluate the performance of the developed system by comparing it to other controllers which are used in these cases. These types of tests have been carried and showed promising results.

References

1. K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.
2. T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, “Evolution Strategies as a Scalable Alternative to Reinforcement Learning,” *ArXiv e-prints*, 2017.
3. X. Zhang, J. Clune, and K. O. Stanley, “On the relationship between the openai evolution strategy and stochastic gradient descent,” *CoRR*, vol. abs/1712.06564, 2017.
4. I. Loshchilov, “Cma-es with restarts for solving cec 2013 benchmark problems,” 06 2013.

5. M. s. Saad, H. Jamaluddin, and I. Mat Darus, "Pid controller tuning using evolutionary algorithms," vol. 7, pp. 139–149, 01 2012.
6. K. Marova, "Using CMA-ES for tuning coupled PID controllers within models of combustion engines," *CoRR*, vol. abs/1609.06741, 2016.
7. Y. Wakasa, S. Kanagawa, K. Tanaka, and Y. Nishimura, "PID Controller Tuning Based on the Covariance Matrix Adaptation Evolution Strategy," *IEEJ Transactions on Electronics, Information and Systems*, vol. 130, pp. 737–742, 2010.
8. B. Doicin, M. Popescu, and C. Patrascioiu, "Pid controller optimal tuning," *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, June 2016.
9. N. Hansen, "The cma evolution strategy: A tutorial," 2010.
10. N. Hansen, "Cma-es source code."