Adaptation du comportement sensori-moteur de robots mobiles en milieux complexes

Ecole doctorale Sciences Pour l'Ingénieur

Ashley Hill[1], Eric Lucet[1] et Roland Lenain[2]

1. CEA LIST – LRI, 2. IRSTEA

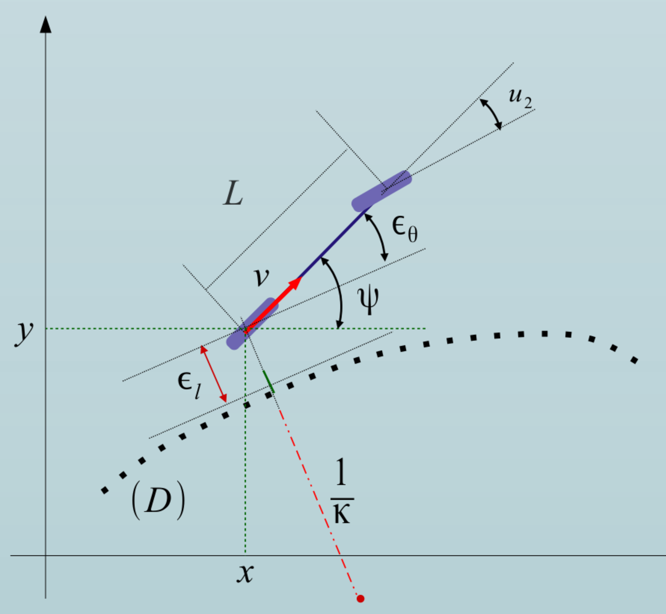
Introduction

Mobile robots are used to accomplish different missions, their sensors being used to correctly and efficiently understand the robot's environment. Those sensors have varying degrees of **certainty in their measurements** depending on the environment and their properties. This **limits the efficiency** of robots using static controllers, as the tuning of their parameters takes into consideration the **nominal behavior** of the robot, which has a negative effect on the robot's efficiency when operating in sub-nominal states. Furthermore, the tuning of these controllers is done to guarantee a higher level of margins, which has a negative effect on robot performance during nominal states. This **compromise reduces the overall performance** of the robot.

The aim of the work is to integrate the **covariance of the measurement noise** and the **trajectory information** into the control policy in order to adjust the robot's behavior to its complex environment, by adjusting the **controller gain**. **Neural networks** have been used, with promising results [2,3,4]; however such methods use small neural networks that are not capable of complex inference, and are using the error or state vector as the basis for the gradient which can cause instability if noisy.

The following Robotic model with the state equations below and the controller defined by u_2 :

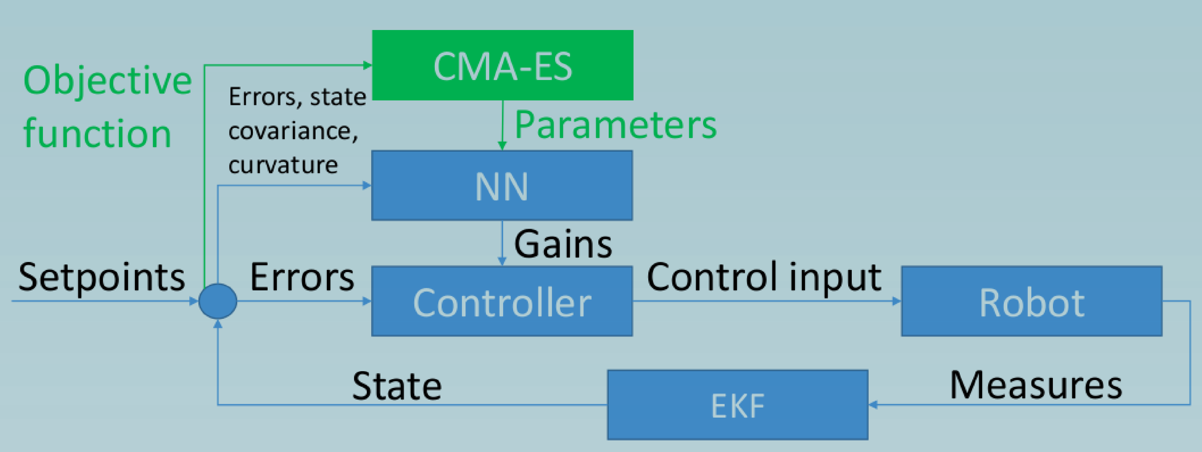
$$\dot{X} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} v\,\cos(\theta) + \alpha_x \\ v\,\sin(\theta) + \alpha_y \\ v\,\frac{\tan(u_2+\alpha_{u_2})}{L} + \alpha_\theta \\ u_1 + \alpha_{u_1} \end{pmatrix}$$



$$u_2 = \arctan\left(\frac{L\,\cos^3\epsilon_\theta}{\alpha}\left(k_\theta(e_\theta) + \frac{\kappa}{\cos^2(\epsilon_\theta)}\right)\right) \text{ with } e_\theta = \tan\epsilon_\theta - \left(\frac{k_l\,\epsilon_l}{\alpha}\right)$$

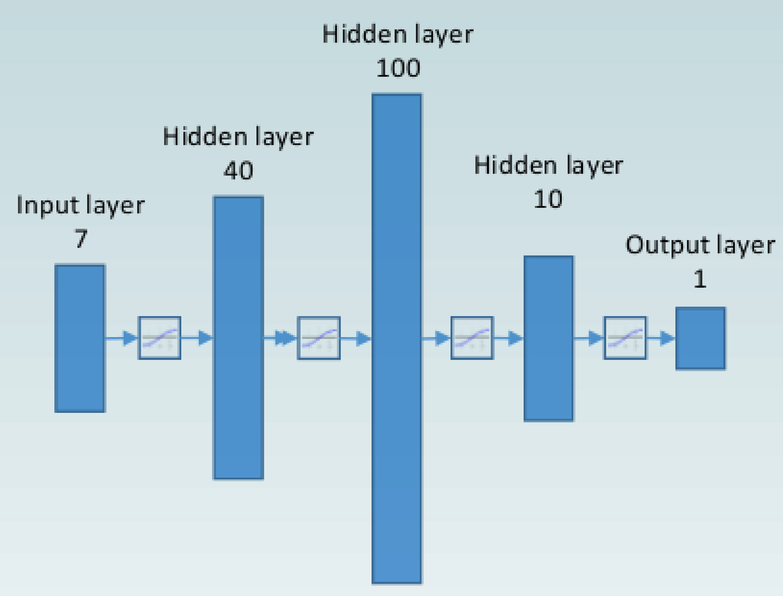
Methods

A new strategy for on-line gains adaptation is proposed :



The controller uses the gains defined by the neural network in order to effectively follow the requested trajectory. The robot's dynamics simulate the behavior of the robot. The noise applied to the measurements mimics real world conditions, and so an **extended Kalman filter** (EKF) is used to observe the state of the robot.

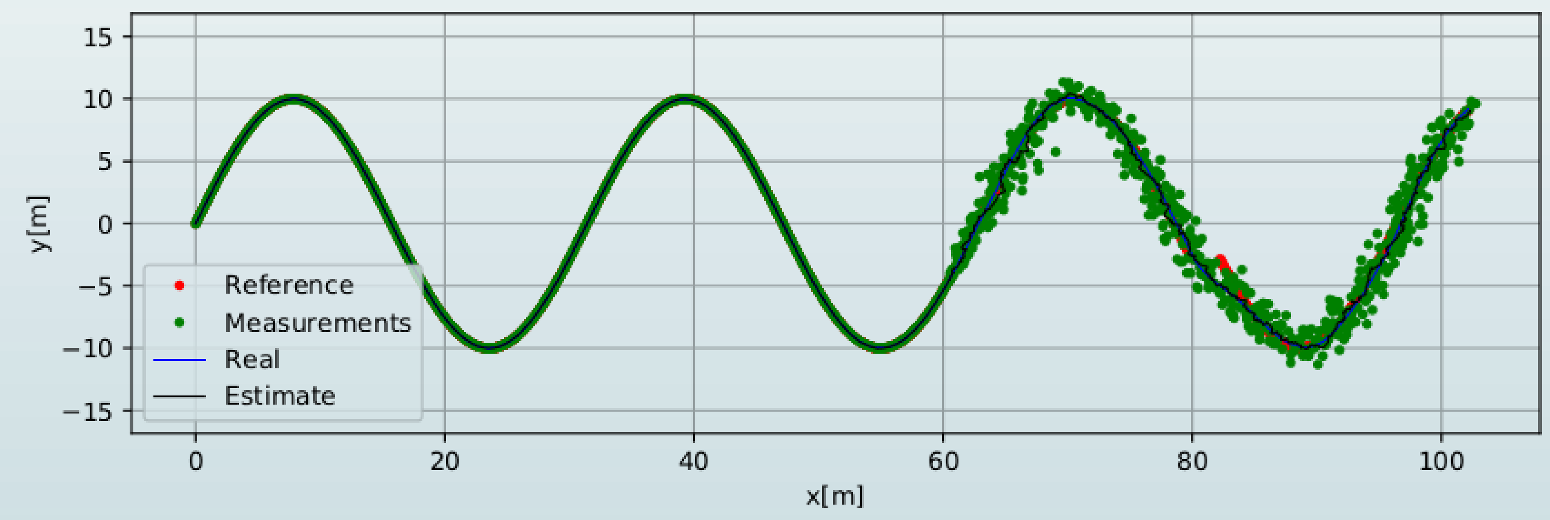
The neural network is trained using CMA-ES [1] as the optimization algorithm for the parameters of the neural network. This is commonly called **Neuroevolution**, and has shown promising results in the past [5]. However for this, objective function must be defined. In our case it is defined as the **minimization** of the lateral error, angular error, and applied steering.

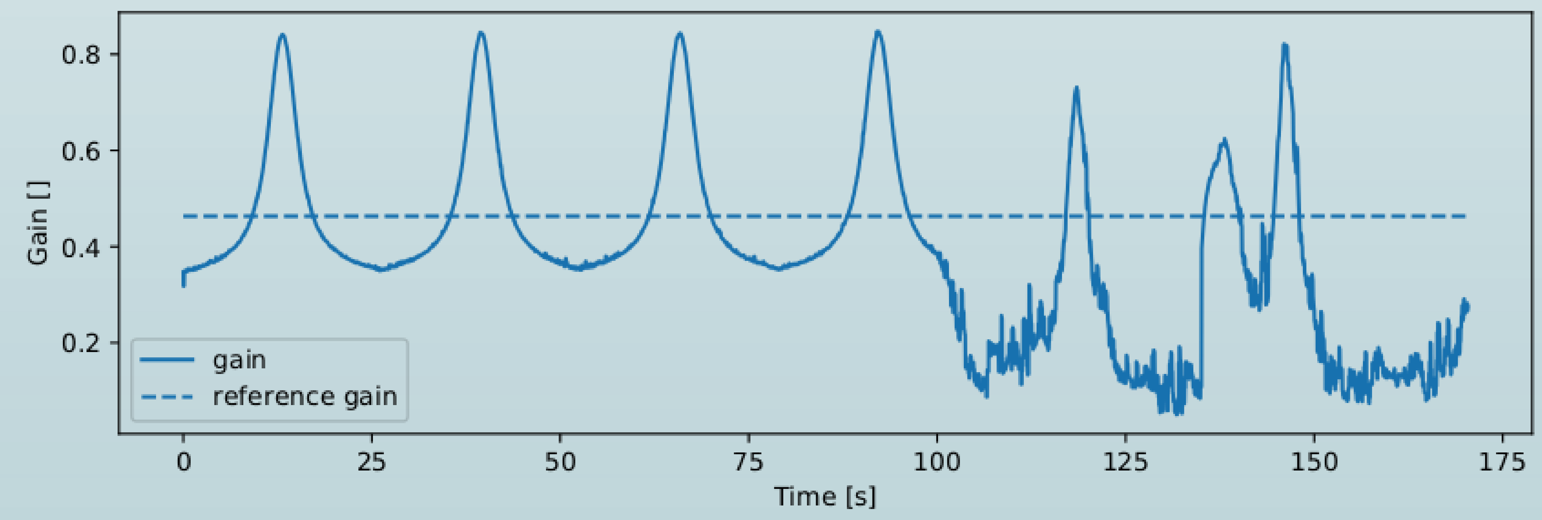


The neural network is given the lateral error, angular error, Kalman covariance matrix, trajectory curvature and current speed as input, in order to determine the optimal gain.

Results

The experiments where done with a **simulation** of the robotic model and state equations, over a few trajectories. Here is an example of the gain when following a sinusoidal path :





A visible **increase** in the gain can be observed when perturbations such as comers occur, and a **reduction** of the mean gain when entering a noisy region were the robot is no longer accurately measuring it's position. This allows more accurate following of the trajectory, while avoiding oscillatory behavior in sub-optimal regions.

A more quantitative measurement is required however to show the general improvement of the method. As such, simulations were run **250 times** over different trajectories, using the proposed method (named **CMA-ES NN**), compared over reinforcement learning methods and a fixed gain method. The comparative metric was the objective function defined in the method (**lower is better**) :

Trajectory	**CMA-ES NN**	SAC	PPO	DDPG	A2C	Fixed gain
Line	**41.60** (± 2.34)	60.99 (± 22.21)	115.11 (± 57.10)	**158.14** (± 4.09)	57.52 (± 15.12)	48.20 (± 2.45)
Sine	**144.76** (± 2.86)	1140.52 (± 3004.65)	**2403.93** (± 2824.32)	309.18 (± 5.79)	280.51 (± 175.61)	151.70 (± 2.83)
Parabola	**98.73** (± 3.39)	191.48 (± 2.43)	389.65 (± 6.34)	**417.99** (± 6.23)	208.80 (± 26.57)	125.29 (± 4.02)
Spline1	**117.66** (± 3.04)	508.24 (± 152.24)	**2864.41** (± 14.31)	272.20 (± 5.27)	542.85 (± 4.47)	142.04 (± 3.45)
Spline2	**147.32** (± 3.18)	**6563.63** (± 3310.81)	3169.80 (± 23.61)	258.85 (± 26.39)	437.22 (± 736.97)	164.94 (± 4.45)

Conclusions

We are currently working on a method of **neuroevolution**, which is used to train a neural network to then tune a controller in real time in order to adapt a robot's behavior to a varying level of precision in the perception. The proposed method has been shown to **improve** the overall performance in the context of mobile robotics when compared with constant gain models or reinforcement learning methods. Furthermore, the proposed method can be used with varying controllers in many different applications, such as navigation in urban landscapes, agricultural application, or even drones. Many possible variants exist of this method that could be put in application, such as variants of the CMA-ES algorithm could be used, or even the possible variants of the objective function for different tasks. First simulation tests have shown the **theoretical validity** of the proposed approach, accounting for sensors noises and low level settling times. Further experimentation with existing adaptive control algorithms, multi-gain controllers, observateurs gain tuning, and **experimentation with real world robots** are envisioned for future works, especially with respect to grip conditions.

Bibliography

1. **Hansen, N. (2016). The CMA evolution strategy: A tutorial.CoRR, abs/1604.00772**
2. **Jalali, L. and Ghafarian, H. (2009). Maintenance of robot's equilibrium in a noisy environment with fuzzy controller. In 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, volume 2, pages 761–766**
3. **Jiang, L., Deng, M., and Inoue, A. (2008). Support vec-tor machine-based two-wheeled mobile robot motion control in a noisy environment. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 222(7):733–743.**
4. **Doicin, B., Popescu, M., and Patrascioiu, C. (2016). Pidcontroller optimal tuning. In2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pages 1–4**
5. **Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I.(2017). Evolution Strategies as a Scalable Alternativeto Reinforcement Learning.ArXiv e-prints.**