Improvement of trajectory tracking accuracy in serial robotic arms based on feed-forward neural networks per servomotor for position control

Baptiste Toussaint[1], Maxime Raison[1],

[1]Department of mechanical engineering
Polytechnique Montreal
2900, Edouard-Montpetit, Montréal, Canada
[baptiste.toussaint, maxime.raison]@polymtl.ca

EXTENDED ABSTRACT

1 Context

In the field of robotic arms, fast and accurate motion control can be important, with applications involving fast dynamics and trajectories with important accelerations and large movements. Especially, a successful tracking of such trajectories in a safe and reliable way requires either an accurate model of the arm or an aggressive tracking with high-gain feedback, such as a traditional proportional–integral–derivative (PID) controller. Yet, for most robotics arms, an accurate dynamic model can be difficult to obtain, due to some nonlinearities (e.g. friction) and uncertainties (e.g. inertial parameters), in these dynamic systems [1]. The inability of joint servo controllers to address these nonlinearities and uncertainties can lead to a degradation of a trajectory tracking accuracy [2]. This situation is currently accentuated with the attempts to develop 3D printed low-cost robotic arms.

A control architecture for high-speed trajectory tracking was proposed [2-3], by using the advantages of both 1. PID – stable and easy to use – and 2. Neural network (NN) – efficient to estimate the dynamics of systems. This architecture consists in learning the dynamic response of the PID controller by using a NN to predict/correct the errors of the PID (Fig.1).

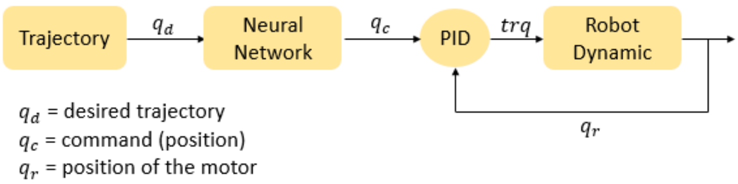


Fig. 1. Control architecture for high-speed trajectory tracking [2-3].

This allows to reduce the errors in position and speed during the trajectory tracking, even without information about the dynamic model [3]. However, no detail was provided to take maximum benefits of this method, forcing one to choose learning parameters experimentally. Especially, there is no guideline about the architecture of the learning system when used as a feedforward controller, with two main architectures: 1. A unique NN for the whole system; 2. One smaller NN for each joint, as well as the usefulness of this method for slower trajectories. The objective of this study is to compare the trajectory tracking accuracy using two architectures – a single NN vs. one NN per servomotor – as feedforward controllers of serial robotic arms. This objective is illustrated in several situations of real serial robotic arms: 3 and 5 DOFs; and trajectory velocities: slow (performed in 1 second), intermediate (2 seconds), and fast (3 seconds).

2 Methods

Using a real 3D-printed robotic arm and a numerical model, we define our problem as follow:

1. Choose a configuration to test. The parameters are the number and position of degrees of freedom (DOF), the duration of the trajectory and the configuration of the NN system.

2. Build a dataset of trajectories that represent the whole workspace of the arm by using a simple PID controller on each joint. Each dataset is composed of five variables for each of the five motors: position q and velocity $\dot{q}$ of the motor at two consecutive timestamps – $q(t)$, $q(t+\delta)$, $\dot{q}(t)$, $\dot{q}(t+\delta)$ – and the command sent during these two timestamps $u(t)$.

3. Train the NN architecture to learn the dynamic model through the response of the PID controller. By giving the NN a lot of data of trajectories, it will learn which command he has to send to each motor to go from a state – $q(t)$, $\dot{q}(t)$ – to the next – $q(t+\delta)$, $\dot{q}(t+\delta)$ – and in an implicit way, the dynamic model of the robotic arm.

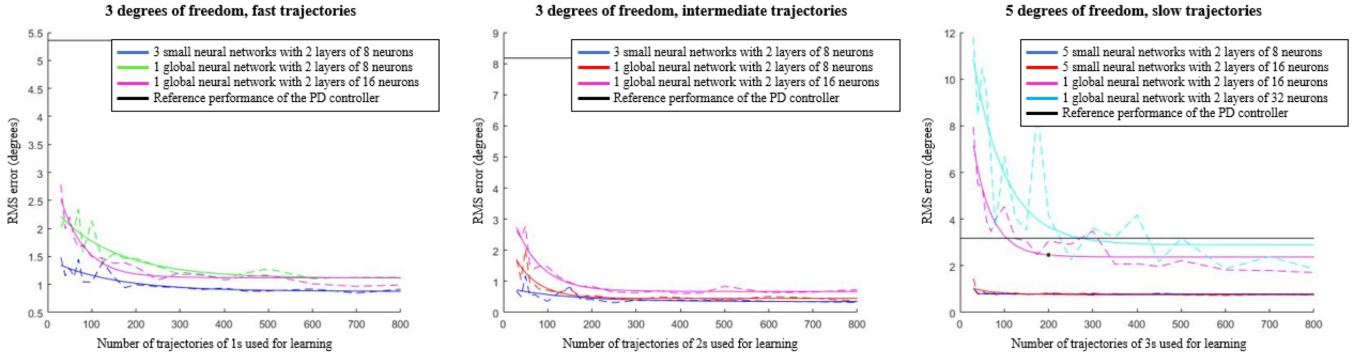
4. Generate a new set of commands for the motors, which consider the response time of the controller to send the requires series of commands to each motor so as these ones execute the desired trajectory.

5. Store the tracking errors for n test trajectories (n = 40, number chosen to have low variation on the results).

6. Repeat steps 2 to 4 for a different number of training trajectories and different NN architectures to compare their effectiveness to track the test trajectories with accuracy.

3 Results

Fig. 2 compares the accuracy – i.e the RMS error in trajectory tracking - for three different configurations between the two studied NN configurations: A. fast trajectories and B. intermediate trajectories for a 3 DOF robotic arm and C. slow trajectories with a 5-DOF serial robotic arm.



A B C

Fig. 2. Performance of trajectory tracking with various NNs, for: A. 3 DOF robotic arm, fast trajectories; B. 3 DOF robotic arm, intermediate trajectories; C. 5 DOF robotic arm, slow trajectories.

Tab. 1 highlights the numerical values of RMS errors among the n test trajectories and the rate of reduction of the errors compared with the reference PD controller. Especially, the first column shows a reduction of tracking errors of at least 75% compared to this reference controller, and an RMS error under 1 degree, whatever the configuration when one NN is used for each servomotor. Finally, Fig 2 shows that the latter architecture allows to get an accuracy close to those obtains with all training data with around 100 and that the use of a global NN requires around 300 trajectories to learn the dynamic response of the PD controller.

Table. 1. Comparison of the trajectory tracking accuracy (RMS error) of NNs for trajectory tracking on two real robot arms

Trajectory velocities (and DOF)	**RMS error with one NN per servomotor**	**RMS error with a global NN**	**RMS error with the reference PD controller**
	In degrees (reduction of tracking errors compared to the reference controller in %)		
Fast (3 DOF)	0.88 (84%)	1.12 (79%)	5.36
Intermediate (3 DOF)	0.33 (96%)	0.45 (94%)	8.17
Slow (5 DOF)	0.76 (76%)	1.71 (46%)	3.18

4 Discussion and conclusion

Fig.2 and Tab.1 recall the interest of the use of a NN as a feed-forward controller, by comparing the accuracy with a standard PD controller. Fig. 1A and 1B showed that the improvement is even more important with fastest trajectories i.e. where the response time of the PD controller has a bigger influence. Further, Tab.1 showed that the use of one NN for each motor demonstrates a reduction of at least 75% of the errors for all tested configurations. Using a NN for each motor allowed a good tracking with an error under 1 degree even when there are few training data available – around 100 trajectories –, whereas the NN for the whole system struggles and requires more data to learn the robot dynamics (around 300 trajectories against 100 for the NNs for each motor), especially with a higher number of DOF – 5 in Fig. 1C. Moreover, even when all the data available is used for training, the use of one NN per servomotor shows better results than the use of a global network. Finally, fig. 1C highlights that the global NN struggle to converge during the training phase and shows a higher variability of the results than the smaller NNs used for each joint.

Next studies will compare these results obtained on two real robot arms with results from a simulated model. The aim is to get more data and use them to better understand why the lack of knowledge on the movements of the other motors doesn't seem to be a limit for the NN used for one servomotor.

References

[1] O. Koç, G.J. Maeda, J. Peters. Optimizing the Execution of Dynamic Robot Movements With Learning Control. IEEE Transactions on Robotics, 35, 909-924, 2019

[2] S. Chen, J.T. Wen. Neural-learning trajectory tracking control of flexible-joint robot manipulators with unknown dynamics. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 128-135), 2019

[3] B. Toussaint, M. Raison. High-speed trajectory tracking on robotic arm by learning the dynamic response of the pid controller with a neural network. Book of Abstracts of the 10th ECCOMAS Thematic Conference on Multibody Dynamics, pages 216–217, 2021