International Conference on Artificial Neural Networks, ICANN'95
Paris, France, October 9-13, 1995

Application of Multilayer Perceptrons to Decouple the Dynamical Behaviour of Robot Links

Friedrich Lange and Gerhard Hirzinger
DLR Institute for Robotics and Systems Dynamics
P. O. Box 1116, D-82230 Wessling, Germany
e-mail: Friedrich.Lange@dlr.de

Abstract

The paper extends a previously proposed method for improving the path accuracy of robots. Especially during high speed movements nonlinear couplings between the joints deteriorate and degrade the robot's accuracy. Such couplings cannot be compensated by linear feedforward control. They require additionally a general function approximator as e. g. a multilayer perceptron. The learning system translates I/O data of robot movements to training data for the neural nets. Linear controllers and multilayer perceptrons are combined after training to improve the execution of sensor planned paths which are not explicitely trained before. The method is verified in experiments using a high speed path which originally is sensed by a compliant force- / torque sensor.

1 Introduction

For most industrial robots the accuracy achievable with the standard control system does not exceed some millimeters for off-line programmed paths. But several applications, as laser cutting, require accuracies of at least some tenths of a millimeter even during high speed movements. The deviations are results of statical effects as e. g. incorrect kinematic data and dynamical effects caused by acceleration, centrifugal or coriolis forces. Statical errors are not discussed in this paper because for sensor-based path planning they don't matter if the difference between actual and desired position is sensed.

Concerning dynamical deviations a method is presented in [LH94] to reduce delays due to acceleration forces. These delays can be minimized by linear feedforward control since they are almost decoupled and proportional to the accelerations. In this paper neural nets are added to cope with the remaining effects, the nonlinear couplings. The proposed approach does not need a priori information about the structure of these couplings.

Common approaches for feedforward control require a fixed path which is trained, thus allowing repeated execution of the desired trajectory. On the other hand, for a fixed path no feedforward control is necessary because the path memory can be modified directly as described in another method of [LH94]. The advantage of neural nets comes into play if the path differs from time to time as it is valid for on-line planned paths, which are modified in relation to unexpected sensor data. Hence, on-line path planning of high speed movements is the adequate task for decoupling by neural nets. If the nominal trajectory is trained accurately the learning system HLCR (Hierarchically Learning Control of Robots) presented in this paper allows high speed execution of individually sensed paths without further adaption to the robot dynamics.

Neural nets for control of robots were already proposed, e. g. by [HHM+92], [ARF93], or [YY93]. In those cases control of a fixed path is learned. Other papers are not restricted to fixed paths but do not memorize the learning result. E. g. [KJH+88] estimates a reference trajectory in each time instant leading the robot from the actual position to the desired path. This reference is then tracked by on-line estimation of the optimal control sequence (Predictive Functional Control). This requires a very high computational effort.

2 Learning strategy

The task is to minimize the difference between the cartesian desired path w and the actual path y. The optimal control input u commands a fictive path which is deformed by the robot dynamics in such a way that the resulting actual path coincides with the desired one. For convenience this task is splitted into subtasks which reduce the positional errors of the individual joints.

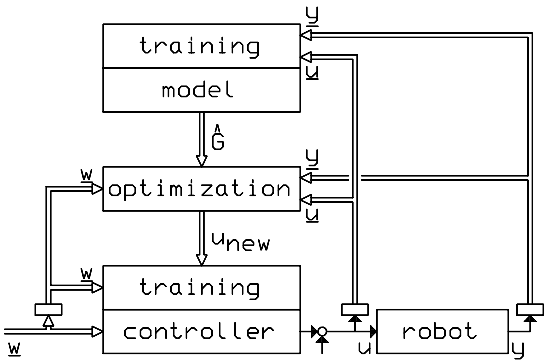


Figure 1: Structure of the learning system (Vectors denote the time history of the corresponding variables)

The basic learning stategy of HLCR is presented in [LH94] and will be repeated here only briefly (see Figure 1). Learning is performed off-line after execution of the training path. At first a simple I/O model $\hat{G}$ of the robot with the internal control system is built. Then the difference between the desired path and the measured path is used to learn a posteriori modifications of the robot commands u. The revised commands u_{new} then act as teacher signals for supervised learning of the feedforward controllers. These controllers weigh subsequent time instants of the desired path w in order to generate accelerations of the robot in time to compensate for dynamical delays. Training is performed as iterative control and learning yielding gradual improvements in path accuracy. If the structure of the controller can represent all dynamical effects and if training is sufficient the controller will be able to reduce path errors of arbitrary paths without relearning the particular movement.

The controller level will be explained here in more detail. In [LH94] a linear approach for each joint is used weighing only subsequent sampling values of the desired trajectory of the particular joint. A linear setup is emphasized because dynamical delays are approximately linear and learning of nonlinear systems needs much more training data.

Additional linear feedforward control of expected contact forces (transformed to the portion $\hat{f}$ of the joint torques by the Jacobian transpose) is proposed in [LH94]. The resulting controller setup can be summarized for each joint j by

$$u_j(k) = w_j(k) + \sum_{i=1}^{n_w} r_{wji} \cdot (w_j(k+i) - w_j(k)) + \sum_{i=1}^{n_f} r_{fji} \cdot (\hat{f}_j(k+i) - f_j(k)) \qquad (1)$$

with learnable weights r_{wji} and r_{fji}. So $n_w + n_f \approx 20$ parameters have to be learned for each joint.

3 Application of neural networks

The setup described above substantially reduces existing path errors. Still, remaining errors exist, due to couplings. In analytical models of robots these errors can be explained by coriolis and centrifugal forces and by the non-diagonal elements of the mass matrix. We propose to extend Equation (1) due to couplings (see Figure 2 for a simpler case). The hybrid approach of linear and neural controllers seems adequate since for the overall system the representational error has to be under 1% which can be reached best using very small output normalization values for the net. Nevertheless two problems remain for robots with 6 joints: The total number of net inputs is immense and the functions are highly nonlinear.

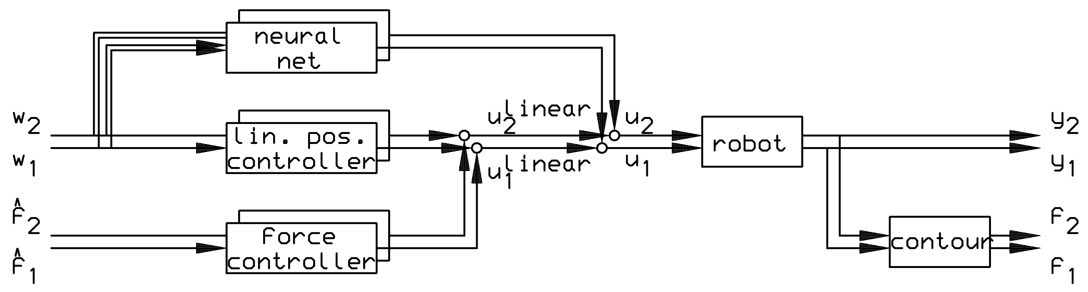


Figure 2: Structure of the controller for a robot with 2 joints

To overcome the first problem a small number of inputs is selected according to the physical effects. For the robot tested, mainly joints 3, 4, and 5 are deteriorated by coupling effects (see Figure 3b). Therefore, it suffices to set up nets for decoupling of these joints (One common net for all 3 joints would require much more weights, at least for a net structure with full interconnection of neighbouring layers.). The net inputs are the absolute values of joints 2 to 5 describing the actual joint geometry, and future values representing accelerations which affect the particular joint. For the joints considered, this yields controller setups of:

$$\begin{aligned} u_3(k) = u_3^{linear}(k) + \quad f_3 \quad &(w_2(k), w_3(k), w_4(k), w_5(k), \\ &w_2(k+1) - w_2(k), \cdots, w_2(k+10) - w_2(k)), \end{aligned} \tag{2}$$

$$\begin{aligned} u_4(k) = u_4^{linear}(k) + \quad f_4 \quad &(w_2(k), w_3(k), w_4(k), w_5(k), \\ &w_1(k+1) - w_1(k), \cdots, w_1(k+10) - w_1(k)), \end{aligned} \tag{3}$$

$$\begin{aligned} u_5(k) = u_5^{linear}(k) + \quad f_5 \quad &(w_2(k), w_3(k), w_4(k), w_5(k), \\ &w_2(k+1) - w_2(k), \cdots, w_2(k+10) - w_2(k), \\ &w_3(k+1) - w_3(k), \cdots, w_3(k+10) - w_3(k)), \end{aligned} \tag{4}$$

where u_j^{linear} is the previously learned function u_j of Equation (1) and $u_j(k)$ is the corresponding component of the teacher signal u_{new} or the control input, respectively. These controller setups restrict the number of net inputs to 14 or 24 real variables.

The second problem is solved by selecting a net architecture which is able to represent arbitrary continuous multiple input single output functions with high accuracy. Multilayer perceptrons with sigmoid activation functions are such general function approximators. In addition they can be trained easily as long as no recurrent weights are provided. Two hidden layers with a sufficient number of neurons each are appropriate to solve the problem. For the experiments the net structure is chosen to be 14-7-3-1 for joints 3 and 4 and 24-8-3-1 for joint 5 resulting in 131 and 231 weights respectively.

Training is performed first for the linear part and then for the nets. As training algorithm an Extended Kalman Filter is used [Lan95]. This method is qualified for problems with up to some hundred weights since a covariance matrix is built that represents couplings between the weights. This almost prevents overwriting of already learned information, at least for small changes in the weights. In contrast to methods from optimization theory [NN91] the Extended Kalman Filter can be used for learning by pattern. Therefore very few epochs are sufficient for learning. In the experiments 10 epochs are used at most before testing the controller and acquiring new training data.

If a training set consists of data of several paths as for the experiments in section 4, the data should be combined to a common training file since otherwise forgetting of information from previous paths is not out of the question when learning from the next ones.

With this learning strategy overfitting is no problem since the set of training data is changed from time to time thus disabling all effects caused by noise. On the other side the new training file itself is sufficient, too, for the representation of the function. This is confirmed by the experiments since not only the error on the actual training set is decreasing but also the error on the previous one. So after all it makes no difference if training is performed with the same data set for 1 or 10 epochs if the total number of epochs is the same. Overfitting can only occur if the number of weights is increased unreasonably with respect to the number of training data in each set or if the number of epochs is not adequate.

Problems will occur independently from the training strategy if a path has to be controlled which is not similar to the trained ones. Then the nets do not correctly represent the dynamical behaviour of the robot. This behaviour is dependent of shape, localization, orientation, and speed of the desired trajectory. So overall generalisation is only possible if a priori information about the physical function to be learned is provided. This yields a parameter adaptive approach as in Equation (1). For joint 4 of the robot tested

$$\begin{aligned} u_4(k) &= u_4^{linear}(k) \\ &+ (sin(w_2(k)) + sin(w_2(k) + w_3(k))) \cdot cos(w_4(k)) \cdot sin(w_5(k)) \\ &\cdot \sum_{i=1}^{n_c} c_{41i} \cdot (w_1(k+i) - w_1(k)) \end{aligned} \tag{5}$$

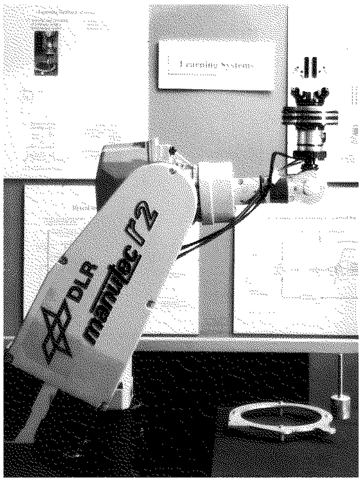
turns out to be a suitable structure with the parameters c_{41i} to be learned. This is the function that is learned by the net, too. But after training of a single path a controller with this equation is globally valid, in contrast to the neural net. It should be stressed that the controller setup requires a priori information which is not available in the general case.

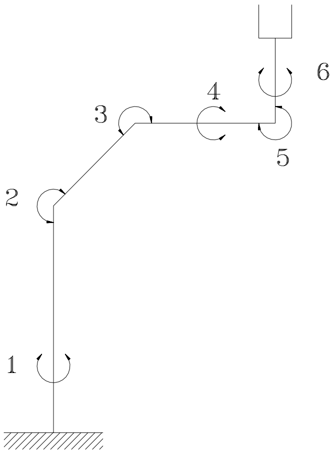
4 Experimental Results

The method is applied to a Manutec r2 industrial robot (Figure 3a) with conventional positional interface. In this case the robot is stabilized by an internal cascaded control system RCM, allowing a common sampling time of 16 ms.

The learning system HLCR can be tested during all high speed movements (see [LH96] for comprehensive experiments). It is demonstrated here for a contour following task (see Figure 3c) for which the path is individually sensed and therefore cannot be learned explicitely before execution.

For industrial applications paths will differ because of tolerances during manifacturing and fixing of the contour. In contrast for the laboratory setup the contour is fixed and always the same. This is the reason for a modified training procedure:





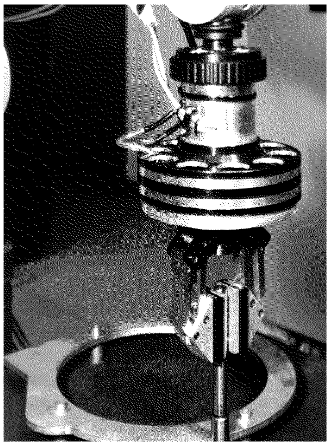


Figure 3a:
Manutec r2 robot in general pose

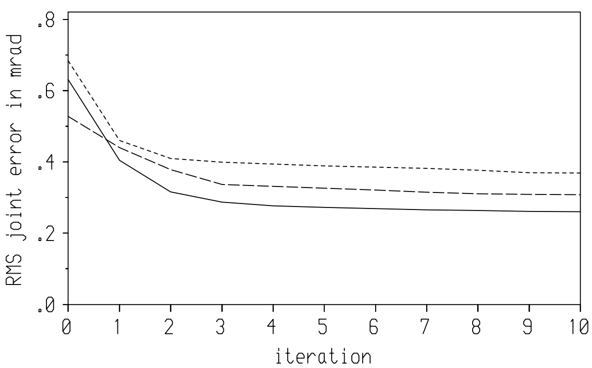
Figure 3b:
Kinematic order of joints

Figure 3c:
Endeffector in starting position at the contour to be followed

Training begins with 10 iterations to build a linear positional controller using a circular path near the contour. Then a linear force controller is trained at the front part of the contour for 10 further iterations. Finally neural nets are trained. These nets are not trained with data from the actual contour but with movements executed without pin and shifted in the 6 cartesian directions by $5mm$ with respect to the previously (at low speed) sensed path. Further displacements are possible but would exceed the tolerances occuring in the industrial task.

For testing of the resulting nonlinear controllers no displacements are necessary anymore. Instead, the contour is sensed again and is then repeated, thereby exerting forces of about $10N$.

Training of the nets is performed using informations of all shifted paths at each time. It is stopped in each case when the net error has been reduced to half or when 10 epochs have been executed. Then the changed nets are applied for control of the nominal path (Figure 4a) and for generating new training data. Within two iterations, requiring less than $90s$ total CPU-time for learning on a R4400 processor, positional errors are reduced substantially. Learning can be



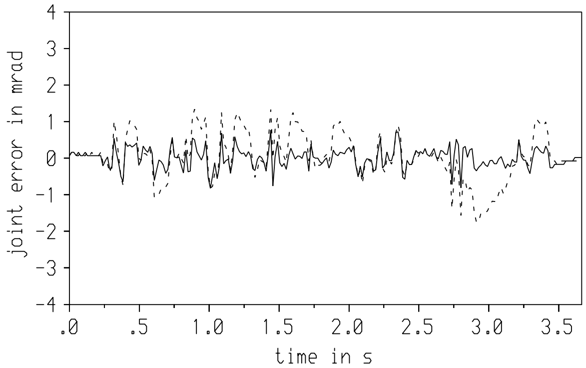


Figure 4a:
Reduction of the RMS joint angle errors by learning when testing the sensed (not explicitely trained) path without pin (solid = joint 3, dashed = joint 4, dotted = joint 5)

Figure 4b:
Error signal of joint during execution of the sensed path with up to 0.7 m/s with contact as in Figure 3c when using a controller with (solid) or without (dashed) neural net

continued resulting in a monotone decrease of the errors in the joint angles. Also when applied to the case with contact the control error is halved with respect to the results of [LH94] (Figure 4b).

5 Conclusion

We presented the learning control system HLCR for improving the path accuracy of robots during high speed movements. Linear feedforward control and neural nets for decoupling turned out to be an efficient combination for executing sensed paths with low path error. Experiments were shown for movements with the highest speed possible, taking into account the acceleration limits of the robot. Unfortunately multilayer perceptrons improve the robot's accuracy only in vicinity of the trained trajectory so that reasonable applications are restricted to on-line modified paths. Future work therefore will concentrate on the construction of general purpose nets which allow global generalisation but which do not require as much a priori information as the approach represented by Equation (5).

References

[ARF93] F. Arai, L. Rong, and T. Fukuda. Asymptotic convergence of feedback error learning method and improvement of learning speed. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Yokohama, Japan, July 26-30 1993.

[HHM+92] M. Höhfeld, J. Hollatz, S. Miesbach, D. Obradovic, B. Schürmann, V. Sterzing, M. Taniguchi, and V. Tresp. Advanced neural control for industrial applications. In M. van der Meer, editor, *Statusseminar Neuroinformatik*, Schloß Maurach, Germany, Oct. 1992.

[KJH+88] K.-B. Kuntze, A. Jacubasch, U. Hirsch, J. Richalet, and Ch. Arber. On the application of a new method for fast and robust position control of industrial robots. In *IEEE Int. Conference on Robotics and Automation*, Philadelphia, Pennsylvania, Apr. 24-29 1988.

[Lan95] F. Lange. Fast and accurate training of multilayer perceptrons using an extended Kalman filter (EKFNet). Internal paper, available by http://www.op.dlr.de/FF-DR_RS/STAFF/friedrich_lange/, Sept. 1995.

[LH94] F. Lange and G. Hirzinger. Learning to improve the path accuracy of position controlled robots. In *IEEE/RSJ/GI Int. Conference on Intelligent Robots and Systems*, München, Germany, Sept. 12-16 1994.

[LH96] F. Lange and G. Hirzinger. Learning of a controller for non-recurring fast movements. *Advanced Robotics, Special Issue on Behavior and Learning*, 1996.

[NN91] S. G. Nash and J. Nocedal. A numerical study of the limited memory BFGS method and the truncated-newton method for large scale optimization. *SIAM Journal on Optimization*, 1(3):358–372, August 1991.

[YY93] T. Yamada and T. Yabuta. Application of learning type feedforward feedback neural network controller to dynamic systems. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Yokohama, Japan, July 1993.