Learning to Improve the Path Accuracy of Position Controlled Robots

Friedrich Lange and Gerhard Hirzinger
Institute for Robotics and Systems Dynamics
DLR (German Aerospace Research Establishment)
Oberpfaffenhofen, D-82234 Wessling, Germany
df67@vm.op.dlr.de or fritz@donau.df.op.dlr.de

Abstract

A learning method is presented which improves the dynamic accuracy of conventional industrial robots with integrated position control. The method is based on feedforward control being able to follow off-line programmed trajectories with high speed and negligible pose errors. For learning, the robot has to be moved along a given path. The algorithm then estimates a simple model. This model is used to build up a controller which is able to modify positional commands, thus reducing the positional path error from some millimeters to approximately 0.2 mm for a manutec r2 robot. This improvement is valid also for other, non-trained trajectories. For repetive control of a single path the error will be even less. Measurements of path accuracy are verified using data of a force/torque sensor during tracking a known contour.

1. Introduction

There is a lot of literature about learning control of robots, but most papers deal with improvement of the accuracy of fixed paths by repetitive control (see e. g. [2], [4], or [11]) or with learning to react on sensory signals (see e. g. [3], [5], or [7]).

On the contrary this paper treats the case that the robot is trained only once and should then be able to follow arbitrary paths with high accuray without any further learning. This is important for off-line programmed tasks which take into account the shape of nominal trajectories but normally don't care about the dynamic behaviour of the robot which even may differ among robots of the same type.

In literature this task was dicussed in the past only for robots with joint torque interface (see e. g. [6] or [10]), often assuming knowledge of an inverse robot model (see e. g. [1]). Such an approach in general is not based on positional control which, however, is standard for most existing robots. So we developed a method which easily can be integrated into the normal control system of industrial robots. For our tests it was realized as an external add-on to the robot controller.

As testbed we use an industrial robot manutec r2 (see Figure 1) with the appropriate SIEMENS robot controller RCM3 which allows to send positional joint commands every 8 ms and to measure the corresponding actual values at the same rate.

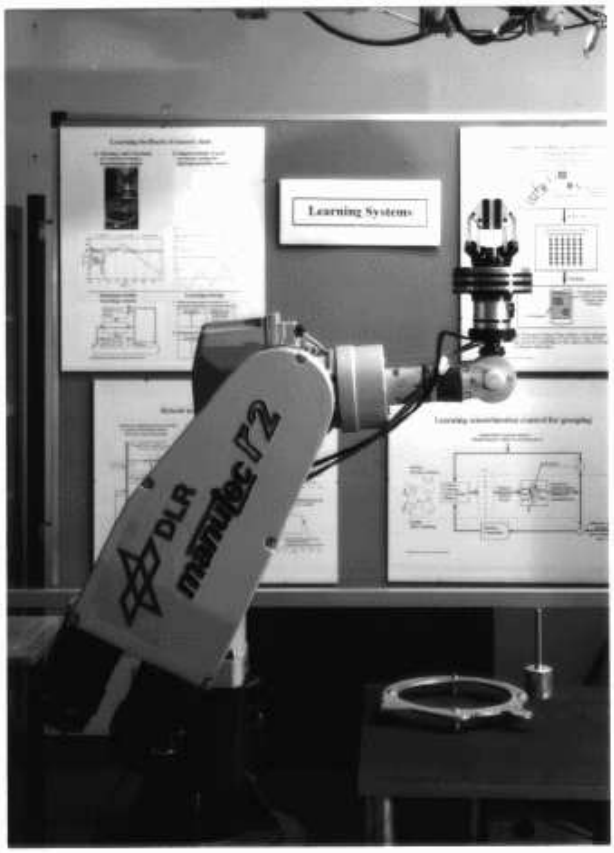

Figure 1. Robot with wrist force sensor in starting position

These commands are the input of cascaded joint controllers which finally move the links to the commanded positions. Unfortunately the delays of the resulting trajectories are different for different joints. This can be

visualized best by plotting the cartesian projections of a commanded path and the resulting actual movements. Figure 2 displays such a sample trajectory, defined by a horizontal and a vertical circular movement of the endeffector without changing the orientation. The starting configuration is shown in Figure 1. The actual path of the vertical movement is distorted because of the high speed of 500 mm/s, whereas in the horizontal circle deviations are mainly harmless trailing errors.

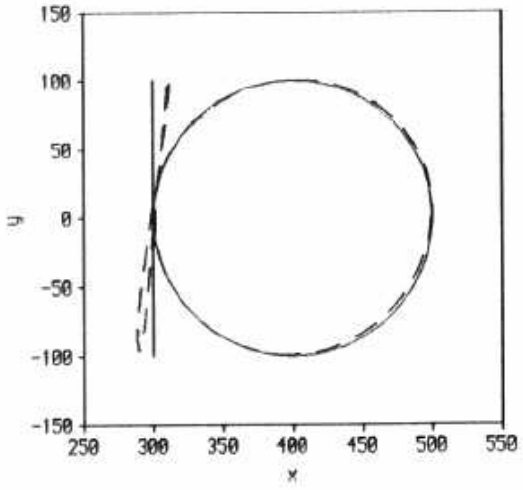

Figure 2. Projection of commanded (desired) and actual path to the xy-plane

The actual positions in Figure 2 are computed using the joint angles measured by the encoders of the motors. So they do not represent the real joint values. Chap. 4. roughly calculates the error.

Chap. 2. explains the method for minimization of the measurable deviations, followed by experimental results in chap. 3..

2. The Learning System

Compensation of path deviations is performed hierarchically in three levels. First a coarse model of the robot is built (sect. 2.1). Then this model is used to modify the commands of the specified path such that the control error is minimized (sect. 2.2). Those modifications can be learned in a feedforward controller being able to correct even untrained paths (sect. 2.3).

Learning is executed on joint level. It turns out that the joints are different but fairly decoupled. So in this chapter we may focus on the treatment of single joints.

2.1 Identification of a Simple Model

Identification of a coarse robot model means learning of a mapping between the positional joint commands $u(k)$ und the measured actual joint positions $y(k)$. The experiments show that a linear decoupled mapping is sufficient for this task so that different but scalar models for the individual joints are appropriate. The approach for each of those models is

$$y(k + n_t + 1) = \hat{g}_0(k) + \sum_{i=1}^{n_g} \hat{g}_i \cdot u(k - i + 1) \qquad (1)$$

with k and n_t denoting the index of the sampling instant and the number of steps until the first reaction in return to a command can be measured, respectively.

A recursive Kalman filter as a sample of a linear parameter estimation method determines the estimated values $\hat{g}_i$ of the impulse response function. The theoretical number of elements of this function is infinite, but for asymptotically stable processes it can be cut. Experiments with our robot yield $n_g = 7$ elements being sufficient when using sampling intervals of 16 ms.

The additional function $\hat{g}_0(k)$ is necessary for the estimation to converge in spite of static differences between $u(k)$ and $y(k)$ due to the limited resolution of the measured values.

To ensure convergence in case of a smooth path it is further necessary to add some stochastical excitation to the commanded values.

2.2 Learning to Follow a Given Path

This model can now be used to modify the positional commands such that the measured values coincide with the desired path. The overall structure is summarized in Figure 3.

The lower part shows the on-line execution of memorized commands. The small blocks left blank collect data of the whole trajectory allowing off-line learning in the rest of the figure. So the vectors represent the scalar values at different time instants.

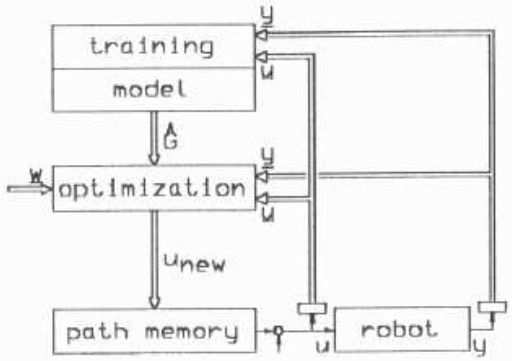

Figure 3. Structure of control (lower part) and learning (upper part)

The optimization procedure determines commands that are appropriate for the control error $e(k) = w(k) - y(k)$ to vanish. To do so, a system of linear equations

$$\begin{bmatrix} \hat{g}_1 & & & \\ \dots & \dots & & \\ \hat{g}_{n_g} & \dots & & \\ & \dots & \dots & \\ & & \hat{g}_{n_g} \dots & \hat{g}_1 \end{bmatrix} \cdot \begin{bmatrix} \Delta u(0) \\ \dots \\ \dots \\ \dots \\ \Delta u(N - n_f - 1) \end{bmatrix} = \begin{bmatrix} e(n_f + 1) \\ \dots \\ \dots \\ \dots \\ e(N) \end{bmatrix} \quad (2)$$

has to be solved, with the required changes of the memorized commands

$$\Delta u(k) = u_{new}(k) - u(k) \quad (3)$$

as unknowns. Instead, the unknowns can be estimated, thus enabling the explicit consideration of noise in the measurements.

This estimation is performed using the inverse Kalman filter (see e. g. [9]), which can be modified to reduce the computing effort to be only proportional to the length N of the trajectory. This yields essential acceleration of the optimization, since $N \gg n_g$.

Estimation and execution of the optimal commands can be repeated iteratively, until performance does not improve any more. Convergence can be improved when the learning parameters of the inverse Kalman filter, as e. g. the assumed noise on the measured values, are determined in advance by comparing real data and the prediction of the model.

2.3 Modification of Commands for Arbitrary Paths

Modifications of the commanded path according to the previous chapter require new learning for every new path or speed. So a parameterized version is necessary for on-line modification of arbitrary trajectories after initial training.

It proves to be sufficient for this modification to train a linear feedforward filter using some subsequently desired positions $w(k + i)$ as input.

$$u(k) = w(k) + \sum_{i=1}^{n_w} r_i \cdot (w(k + n_f + i) - w(k)) \quad (4)$$

For the experiments in chapt. 3. $n_w = n_g = 7$ is used. So with $n_f = 3$ only 10 predicted steps of the path are necessary, making the approach applicable for on-line path planning as well.

This kind of feedforward control is superior to any algorithm implemented in the robot controller, because additional information about the future path is always provided and not restricted to simple, in most cases even linear moves. On the other side the knowledge of the future path can be a restriction, especially for sensor guided movements. This disadvantage can be compensated by using a predictive sensor [12] which in combination with the proposed feedforward control allows to make fast and accurate sensor controlled movements.

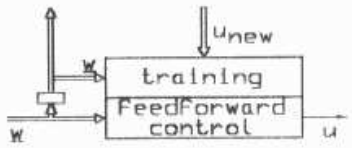

Figure 4. Learning (upper part) and usage (lower part) of feedforward controller

The resulting learning and control structure can be obtained from Figure 3 by replacing the path memory block by Figure 4. Training is performed by estimating the parameters r_i of equation (4) according to the target signals u_{new} using a recursive Kalman filter as for the identification in sect. 2.1. So the adaptation of learning parameters is possible as well.

3. Experimental Results

Experiments with our laboratory setup prove that it is sufficient to execute a slow movement with superposed stochastical excitation of about 200 sampling steps to get a model that is accurate enough.

Thereafter the path specific modification of the commands according to sect. 2.2 converges within few iterations (see Figure 5 for deviations in cartesian space).

The positional pose errors without modification of commands are approximately the joint angles covered in 80 ms. Transferred to cartesian values within 20 iterations the pose error is reduced to about 0.1 mm (see 3rd column in Table 1).

The more important positional path error, i. e. the part of the positional pose error normal to the direction of movement, can be reduced by a factor of 40 for this high-speed trajectory. Accuracy is limited by the resolution of the encoders (about 0.05 mm) and, for high speed movements, by model errors.

Orientational pose error and orientational path error are not regarded, it may be assumed, however, that similar improvements are observable as well, because modifications are learned from joint errors and not from cartesian positions.

The learned feedforward controller cannot compete with the accuracy of real path learning. Nevertheless the error is reduced substantially, even for non-trained paths, if the training path is suitable, as are trajectories planned in cartesian space for learning in joint space. Table 1 and Table 2 show the results after training of the first path of Table 2 (which is not identical with the path of Table 1). The resultant positional pose error is about 0.2 mm. Unfortunately, path deviations can be calculated easily only for circular paths, so that this criterion is dropped in Table 2. However, the circular path of Table 1 proves a reduction of factor 20 for the positional path error of high speed movements.

The inferiority of feedforward control to the path specific modification of commands may be explained by nonlinear effects which are compensated in iterative correction of commands, but which cannot be represented by a linear feedforward filter. Besides, the model is learned only for one path, so it is unsuitable for predicting different reactions at different regions of the working space.

If the accuracy reached by feedforward control is not sufficient, it is possible, of course, to continue learning with path specific modifications of the commands as

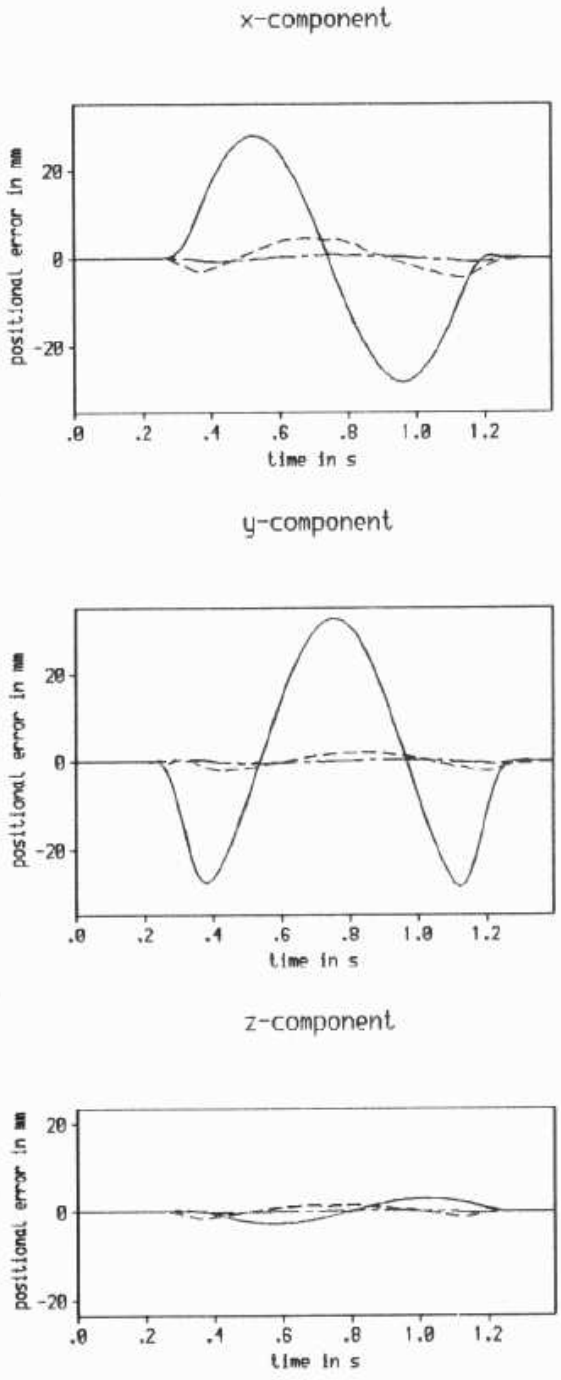

Figure 5. Positional error of horizontal circular path before learning and after the first two iterations

error criterion	without learning	mod. of commands	feedforward control
pose error of joint 1	38.736 mrad	0.109 mrad	0.250 mrad
pose error of joint 2	50.270 mrad	0.054 mrad	0.215 mrad
pose error of joint 3	61.572 mrad	0.075 mrad	0.242 mrad
pose error of joint 4	0.604 mrad	0.103 mrad	0.767 mrad
pose error of joint 5	18.171 mrad	0.093 mrad	0.364 mrad
pose error of joint 6	43.050 mrad	0.252 mrad	0.669 mrad
cartesian pose error	23.091 mm	0.069 mm	0.216 mm
cartesian path error	2.107 mm	0.046 mm	0.114 mm
cartesian trailing error	22.994 mm	0.051 mm	0.183 mm

Table 1. Comparison of different error criteria during execution of a horizontal circle with 375 mm/s

described in chap. 2.. Then the control error can be reduced additionally by a factor of 2.

Concerning the range of paths it can be said that the acceleration limits of the robot are reached, so that faster movements are hardly possible. The two lower paths in Table 2 cover maximal changes of the inertia of joint 1, from small values of x according to the maximal joint speed to an almost singular configuration with big values of x.

Orientation of the gripper was held constant, either upside as in Figure 1, or downside as in Figure 6. So joint 4 was not moved at all, thus disabling any improvement by feedforward control.

4. Verification of Path Measurements

So far, the error reduction only deals with the joint values measured at the motor shafts. To assess the real joint error, sensor signals have to be evaluated measuring the position relative to some fixed objects. In our case a compliant force/torque sensor is used, guaranteeing a high positional resolution.

The gripper is commanded along a known contour as in Figure 6 evaluating the torque values to detect radial path deviations.

Unfortunately, sensible contact forces influence the dynamic behaviour of the robot. So, for the estimation of the error not directly measurable and for further applications with force contact at the endeffector an extension to the described feedforward control is useful.

4.1 Extension to React on External Forces

Extension to the presented feedforward control can be made in different ways. The first idea [7] was to include

path	speed	cartesian pose error	
		without feedforward control	with
hor. and vert. circles, gripper upside	500 mm/s 250 mm/s	39.900 mm 21.523 mm	0.187 mm 0.130 mm
hor. circle, gripper downside	375 mm/s 188 mm/s	23.091 mm 13.546 mm	0.216 mm 0.134 mm
vert. circle, gripper downside	375 mm/s 188 mm/s	24.053 mm 14.126 mm	0.182 mm 0.121 mm
hor. cirlces, gripper upside, from x = 100 mm to x = 650 mm	375 mm/s 188 mm/s	32.074 mm 16.709 mm	0.160 mm 0.123 mm
hor. rectangles, gripper upside, from x = 125 mm to x = 650 mm	375 mm/s 188 mm/s	27.221 mm 16.218 mm	0.185 mm 0.155 mm

Table 2. Pose Errors at Different Paths

Figure 6. Configuration for verification of measured position

feedback control. Experiments with our laboratory setup proved, however, that at a sampling rate of 125 cps further improvements are not possible [8].

The same turns out to be valid for force feedback in the outer loop, provided that the programmed path exerts only small forces as in the experiments of sect. 4.2.

It is, however, possible to include predicted changes of force in the learned feedforward control. This is done just by replacing the feedforward controller of equation (4) by

$$u(k) = w(k) + \sum_{i=1}^{n_w} r_{wi} \cdot (w(k + n_t + i) - w(k)) + \sum_{i=1}^{n_f} r_{fi} \cdot (\hat{f}(k + n_t + i) - f(k)) \quad (5)$$

The predicted forces $\hat{f}(k)$ have to be computed using the analytical description of the path and the contact surface or using a predictive sensor as in [12]. Prediction using the actual sensor values does not give any additional information. It should be mentioned further, that predicted and sensed forces are multiplied by the Jacobian transpose before being used in equation (5).

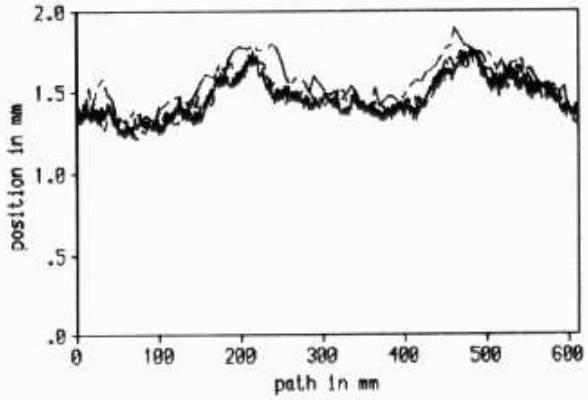

Figure 7. Difference between sensed and measured radial position for velocities of 31 mm/s, 125 mm/s, and 375 mm/s.

4.2 Evaluation of Sensor Data

For a compliant sensor, measured forces/torques represent deviations from the nominal position, the deviations being proportional to the measured values with known elasticity factors, 2.5 N/mm and 0.5 Nm/mm, respectively, with respect to the contact point. These deviations are caused either by external contact forces or by internal effects due to acceleration.

In relation to the path deviations to be verified, the sensor is regarded to be stiff. So the tool center point is assumed to be at the position of the contour, displaced by the sensed deviation. As a result the computed cartesian position is influenced by elasticity in the joints but not in the sensor.

There are two problems left for an accurate path detection. First, the absolute position of the contour to be tracked is unkown. So it is only possible to detect relative path errors, caused by inferior repeatability or by dynamical effects. The static error cannot be determined.

The second problem are forces or torques, sensed due to acceleration of the gripper. They have to be compensated separately, using the approximately known trajectory.

Figure 7 shows the absolute values of the differences between the sensed contour and the path measured in the joints. Friction as well as centrifugal forces in the sensed data are compensated. The graphs are always

no. of test	speed	measured path error	difference between sensed contour and contour of 1st test	total (sensed) path error
1	31 mm/s	0.036 mm	-	-
2	31 mm/s	0.037 mm	0.048 mm	0.046 mm
3	125 mm/s	0.129 mm	0.091 mm	0.173 mm
4	250 mm/s	0.078 mm	0.126 mm	0.138 mm
5	375 mm/s	0.110 mm	0.156 mm	0.161 mm
6	31 mm/s	0.217 mm	0.067 mm	0.263 mm
7	125 mm/s	0.810 mm	0.179 mm	0.959 mm

Table 3. Mean radial path error for different speeds using learned feedforward control (upper part) or the original robot controller (lower part).

positive to guarantee contact. They are not constant because of irregularities in the circular shape of the contour. But those disturbances are the same for all trajectories. The difference between the curves is a measure for the dynamical deviations that are not measurable using the joint values (see 4th column in Table 3).

Comparing the curves yields a distinction between stochastical disturbances of about 0.05 mm and a velocity-dependent part of about 0.1 mm. The stochastical part is caused by the pose repeatability of the robot and by measurement noise. The velocity-dependent part comes from elasticity in the joints yielding compliance because of dynamical forces.

Measured and total deviations are correlated, the mean difference not exceeding 0.1 mm (Table 3). So this is an upper bound for the additional error, not regarded in Table 1 or Table 2, yielding real (total) path errors of about 0.2 mm.

In addition the velocity-dependent disturbance seems to be observable or learnable. To do so, the true position has to be modeled and the real joint control error has to be minimized, not the one calculated by joint measurements. This would require external measurement of the exact six dimensional pose which was not realized in the context of this paper.

5. Discussion

The proposed method on-line reduces measurable control errors for arbitrary paths after initial learning using a suitable trajectory. For our laboratory setup improvements by factors up to 20 are reachable when no contact forces are present. On the other side predictible contact forces can be compensated by feedforward control as well.

Robots with such a learned control behaviour are able to precisely execute off-line programmed paths, without explicit modifications due to dynamical deviations.

For high precision requirements the control error can be further halved by learning modifications of the commands.

These improvements are observed using the internal joint encoders. So for the real joint control error an unknown static deviation and a dynamical one are present, too. The static error is of no importance for sensor aided path planning. The dynamical part does not exceed 0.1 mm. Beyond that it may be learned except for the stochastical noise of less than 0.05 mm, provided that the real control error can be sensed. This sensing setup can be removed after learning, however, because feedforward control does not feed back the actual values.

So far the robot was treated to be essentially decoupled. Existing couplings between the individually controlled joints are observable but are not dominant. For the future we plan to map and compensate these couplings using neural networks.

All experiments were executed using a manutec r2 robot with the corresponding industrial controller. There is no reason to expect some completely different behaviour with another hardware setup. For a direct-drive robot, however, extensions may be desirable to compensate couplings or position dependent motor characteristics. Nevertheless the learning method as it is proposed can be used in this case, too.

6. Bibliography

[1] C. H. An, C. G. Atkeson, and J. M. Hollerbach *Model-Based Control of a Robot Manipulator* The MIT Press, Cambridge, Massachusetts, 1988

[2] S. Arimoto, T. Naniwa, and H. Suzuki *Selective Learning with a Forgetting Factor for Robotic Motion Control* 1991 IEEE Int. Conference on Robotics and Automation, Sacramento, California

[3] H. Asada *Teaching and Learning of Compliance Using Neural Nets: Representation and Generation of Nonlinear Compliance* 1990 IEEE Int. Conference on Robotics and Automation, Cincinnati, Ohio

[4] C. G. Atkeson, E. W. Aboaf, J. McIntyre, and D. J. Reinkensmeyer *Model-Based Robot Learning* 4th Int. Symp. of Robotic Research, Santa Cruz, Aug. 1987

[5] A. G. Barto, R. S. Sutton, and C. W. Anderson *Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems* IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-13, No.5, 1983

[6] D. M. Gorinevsky *Experiments in Direct Learning of Feedforward Control for Manipulator Path Tracking* Robotersysteme 8, 139-147 (1992)

[7] F. Lange and G. Hirzinger *Iterative Self-Improvement of Force Feedback Control in Contour Tracking* 1992 IEEE Int. Conference on Robotics and Automation, Nice, France

[8] F. Lange and G. Hirzinger *Erhöhung der Bahngenauigkeit von positionsgeregelten Robotern* Fachtagung Intelligente Steuerung und Regelung von Robotern, Nov. 1993, Langen, Germany, VDI-Bericht 1094 (in German)

[9] P. S. Maybeck *Stochastic Models, Estimation and Control, Vol. 2* Mathematics in Science and Engineering, Vol 141-2, Academic Press, 1982

[10] W. T. Miller III *Sensor-Based Control of Robotic Manipulators using a General Learning Algorithm* IEEE Journal of Robotics and Automation, Vol. RA-3, No.2, 1987

[11] W. Neubauer, M. Möller, S. Bocionek, and W. Rencken *Learning Systems Behavior for the Automatic Correction and Optimization of Off-line Robot Programs* 1992 IEEE Int. Conference on Intelligent Robots and Systems, Raleigh, NC

[12] G. Pritschow, A. Horn und K. Grefen *Dynamisches Verhalten und Grenzen sensorgeführter Industrieroboter mit vorausblickendem Sensor* Robotersysteme 8, 155-161 (1992) (in German)