1996 IEEE International Conference on Robotics and Automation
Minneapolis, April 1996

Learning Force Control with Position Controlled Robots

Friedrich Lange   Gerhard Hirzinger

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

Abstract

The paper applies a previously presented method for accurate tracking of paths to force control. This approach is very simple since it does not require a joint torque / motor current interface but only a positional interface. It can be applied with elastic end-effectors (sensors) as well as with stiff environments where most elasticity is in the robot joints. In both cases deviations from the desired forces are transfered to positional deviations on joint level. The resulting path can then be controlled with high accuracy by a learned feedforward controller including the influence of the forces. The approach can be applied to the sensing of a contour or to the tracking of a known contour with high speed.

1 Introduction

In industrial applications force sensor signals may be used, like signals of other sensors, for the teach-in of paths in presence of uncertainty. The task is then the detection of surfaces or edges in operations like welding or assembly. In these applications either adjustment of a position is required or tracking of a path with predetermined (low) contact force. Another case where force sensors are needed is e.g. the on-line control of machining operations as grinding where speed has to be adapted in relation to the sensed contact force.

For all such tasks robot controllers have to be provided with sensory interfaces. Control input are modifications of the commanded positions either in cartesian coordinates or in joint space. Direct access to the joint torques or motor currents is not provided for industrial robots. Such path corrections are generated by a controller with a sampling time of typically 10 ms and a delay of several sampling intervals. This means that most proposed schemes for force control (see e.g. [Sal80], [RC81], [Kha87], [Hog87], [AAH88], [Mül93], or [WAN95]) cannot be used without modifications.

A sample task is defined here for experiments on force control: The robot holds a pin which has to move along a planar contour maintaining a constant absolute value of the force vector. During this task the

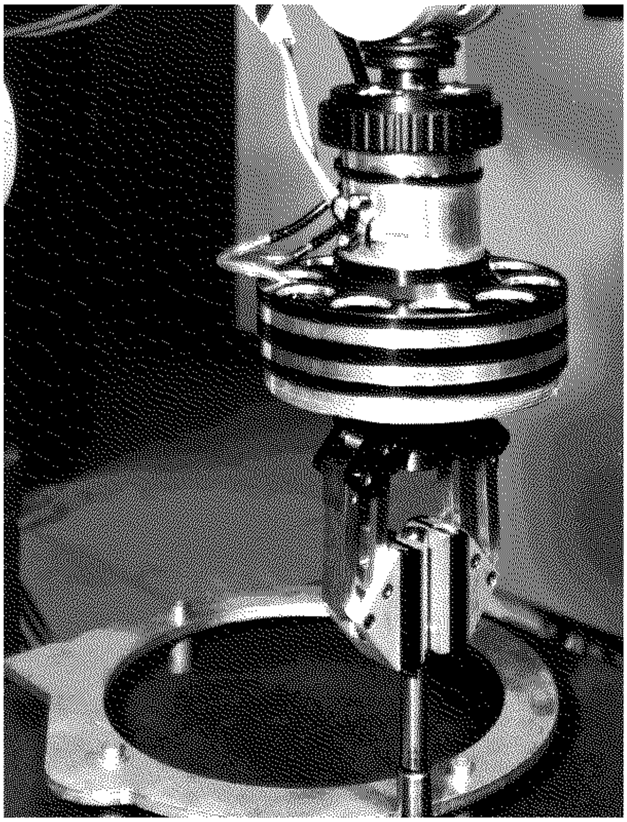

Figure 1: Robot with wrist force-/torque sensor and contour to be tracked

robot has to feel the shape of the contour (Figure 1). Force errors are inevitable since changes in curvature can only be detected by force errors. After sensing, instantaneous reduction of force errors is limited by the maximal accelerations the robot can execute.

A second task can be defined: After the robot has recorded the shape of the contour it is capable to repeat the path. In this case both speed and force accuracy can be increased. The practical use for this task are on one hand problems for which CAD-based teach-in is hardly possible but which are identical for repetitive operation. On the other hand all on-line path planning is identical to this problem if it is performed for a sufficient number of sampling steps in advance. So with a second sensor which can predict the force signal that has to be controlled superior quality can be reached in force control.

In this paper the problem of force control is related to the control of positions (Figure 2). This has two reasons: First, we want to use standard industrial robots with positional interface. Second, not only force values but also other sensor signals, e.g. from laser range finders or vision systems, can be transferred to positional increments which are commanded to the robot. A general controller can be designed which is able to track paths which are planned by multiple sensors. Due to this reason the approach is called multi-sensory control concept. It is a specific way of hybrid control because force information and desired speed have to define the total desired motion in non overlapping subspaces. For the above defined contour following tasks the force controlled subspace is one-dimensional.

For control of forces in stiff systems some authors recommend impedance control or admittance control ([Hog87], [SC93], [PD94], [CGE94]). Those ideas are not applied here since in the force controlled direction we consider only the force error and not the deviation from a desired path. On the other side those methods cannot prevent stability problems due to numerical reasons.

Direct sensor control as in [HL85], [Kha87], or [LH92], where the increments of positional commands are functions of the sensor signals, is not used either, since then learning of the controllers has to be repeated for every new sensor configuration.

Instead, in the multi-sensory control concept all desired movements are expressed by new desired positions. So learning refers only to the tracking of desired paths. [LH94] presents such a controller which after minimal learning is able to guarantee high accuracy for arbitrary positionally defined paths. The disadvantage of this concept is, that all sensors have

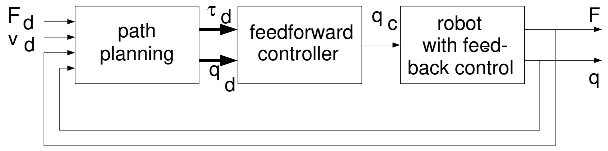

Figure 2: Structure of force control using the multi-sensory control concept: Sensory signals $\mathbf{F}$ yield a desired path $\mathbf{q}_d$ which is controlled by commands $\mathbf{q}_c$. (Thick lines mean data of the current and several subsequent sampling instants)

to be calibrated in positional units, which means the explicit knowledge of elasticity for force control. This will be discussed further in Section 2 which deals with path planning, i.e. the conversion of force signals to a target position for the tool center point (TCP).

A feedforward controller module is required since the standard cascaded feedback controller is only suitable for reducing statical errors. For compensation of dynamical influences some additional calculations are necessary. Section 3 repeats this from [LH94] and [LH96] describing a learning method for accurate execution of positionally defined paths.

The total system will be demonstrated in experiments in Section 4.

2 Transferring sensed forces to desired joint angles

For the task to be solved path planning consists of two steps.

In the first step desired vectors of force and position have to be derived from scalar target values of force and speed. This is explained in Section 2.1.

In the second step the desired joint angles have to be computed from the target vectors of force and position. Roughly spoken, in this step forces are converted to positional increments according to the present elasticity. We distinguish between elastic endeffectors or environments (Section 2.2) and stiff systems where elasticity of the robot joints is predominant (Section 2.3).

2.1 Projection of desired values to the actual contact frame

The force- / torque vector

$$\mathbf{F} = \begin{pmatrix} \mathbf{f} \\ \mathbf{n} \end{pmatrix} \tag{1}$$

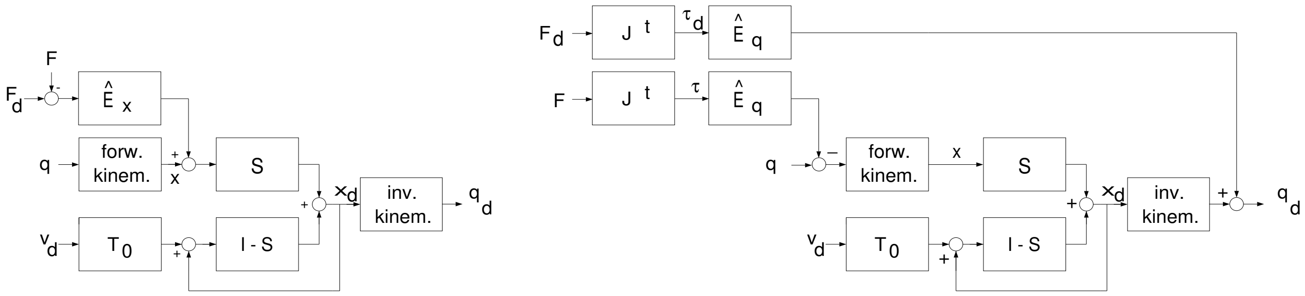

Figure 3: Path planning (without projection of desired values): Left hand side for robots with cartesian elasticity, right hand side for robots with elastic joints

is defined by measured forces and torques expressed with respect to the TCP and with compensated weight. For one-point contact in the TCP as in Figure 1 this means vanishing torques.

In the sample task of Section 1 the desired motion is then given by targets for the absolute value of the force vector $|\mathbf{f}|$ and for the absolute value of speed along the contour $|\mathbf{v}|$, and by the plane of the contour expressed by the normal unit vector $\mathbf{e}_p$.

For convenience additional unit vectors of the force controlled and the velocity controlled directions are defined. The first one has the direction of the actual force vector reduced by the component normal to the plane of the contour since this component can only be caused by noise or friction.

$$\mathbf{e}_f = \frac{\mathbf{f} - \mathbf{e}_p^t \cdot \mathbf{f} \cdot \mathbf{e}_p}{|\mathbf{f} - \mathbf{e}_p^t \cdot \mathbf{f} \cdot \mathbf{e}_p|} \tag{2}$$

$$\mathbf{e}_v = \mathbf{e}_p \times \mathbf{e}_f \tag{3}$$

This yields the 6 dof desired values and the selection matrix which are required in the subsequent sections.

$$\mathbf{F}_d = \begin{pmatrix} |\mathbf{f}_d| \cdot \mathbf{e}_f \\ \mathbf{0} \end{pmatrix} \tag{4}$$

$$\mathbf{V}_d = \begin{pmatrix} |\mathbf{v}_d| \cdot \mathbf{e}_v \\ \mathbf{0} \end{pmatrix} \tag{5}$$

$$S = \begin{pmatrix} \mathbf{e}_f \cdot \mathbf{e}_f^t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{6}$$

2.2 Evaluation of compliant force sensors

For most tasks in force contact the high stiffness of industrial robots is unfavourable. To avoid stability problems artifical compliances are build in between the last link of the robot and the endeffector ([WN79], [Hir93]). In most cases, as in our laboratory setup, this compliance is part of the force- / torque sensor.

Those compliances can be described by coefficients of elasticity in cartesian space expressed by the estimated diagonal matrix $\hat{E}_x$. The robot as well as the environment are regarded to be stiff.

So without a movement in the velocity controlled direction (and without noise) the desired value of the cartesian position is

$$\mathbf{x}_{dF} = \mathbf{x} + \hat{E}_x \cdot (\mathbf{F}_d - \mathbf{F}) \tag{7}$$

If the matrix of elasticity is known exactly one measurement (with low noise) is sufficient to determine the force controlled direction of the desired motion. Otherwise the error will decrease asymptotically if the assumed elasticity is lower than the real one. The determination of the desired motion is convergent if

$$\hat{E}_x < 2 \cdot E_x \tag{8}$$

is valid element by element. Equation (8) assures stability of the whole system if the positional control loop (with the standard cascaded controller) has aperiodical characteristics. So, in contrast to direct force control, neither the elasticity nor the sampling time affects the stability, as long as the elasticity is roughly known and the measurements of force and position are related to exactly the same time instant.

Including the velocity controlled direction yields

$$\begin{aligned} \mathbf{x}_d(k) &= S(k) \cdot \mathbf{x}_{dF}(k) \\ &+ (I - S(k)) \cdot (\mathbf{x}_d(k-1) + \mathbf{V}_d(k) \cdot T_0) \end{aligned} \tag{9}$$

where k is the time step for a sampling time of T_0.

The left hand side of Figure 3 shows the total path planning module. Transformation of the desired cartesian position to the corresponding joint angles is performed by the inverse kinematic transformation instead of the inverse Jacobian since the inverse Jacobian may result in a drift for the directions which belong neither to $\mathbf{e}_f$ nor to $\mathbf{e}_v$ as are translations normal to the specified plane or rotations.

2.3 Evaluation of stiff force sensors

Without a compliant endeffector or environment, the elasticity in the joints becomes dominant. This has to be treated in a different way since elasticity is not collocated. [Mül93] shows that a joint angle vector

$$\mathbf{q}_d = inv(\mathbf{x}_d) + \hat{E}_q \cdot J^t \cdot \mathbf{F}_d \tag{10}$$

has to be commanded if at the desired position $\mathbf{x}_d$ a contact force $\mathbf{F}_d$ is active, if $inv(.)$ denotes the inverse kinematic transformation, J^t is the Jacobian transpose, and $\hat{E}_q$ is the estimated elasticity matrix on joint level.

In the same way the actual joint angles have to be modified before the forward kinematic transformation $kin(.)$.

$$\mathbf{x} = kin(\mathbf{q} - \hat{E}_q \cdot J^t \cdot \mathbf{F}) \tag{11}$$

Including the velocity controlled direction yields

$$\begin{aligned} \mathbf{x}_d(k) &= S(k) \cdot \mathbf{x} \\ &+ (I - S(k)) \cdot (\mathbf{x}_d(k-1) + \mathbf{V}_d(k) \cdot T_0) \end{aligned} \tag{12}$$

This is shown in the right hand part of Figure 3.

Joint elasticities for the first three joints of a Manutec r3 robot are measured in [Tür90]. Considering the resolution of the joint positions this yields an accuracy that cannot be exceeded by this robot when using the positional interface. This accuracy is reached statically due to an integrative part in the standard feedback controller.

$$\begin{aligned} \boldsymbol{\Delta}\tau &= \hat{E}_q^{-1} \cdot \Delta\mathbf{q} \\ &= \begin{pmatrix} 0.47 & & \\ & 0.35 & \\ & & 0.21 \end{pmatrix} \cdot 10^6 Nm/rad \cdot \begin{pmatrix} 25 \\ 12 \\ 44 \end{pmatrix} \cdot 10^{-6} rad \\ &= \begin{pmatrix} 12 \\ 4 \\ 9 \end{pmatrix} Nm \end{aligned} \tag{13}$$

So for an effective arm length of less than $1m$ the above numbers are an upper bound for the reachable force accuracy in N. For a more accurate force control a compliant endeffector has to be used. Joints 4 to 6 are modelled to be stiff in [Tür90] meaning that cartesian forces and torques cannot be controlled independently.

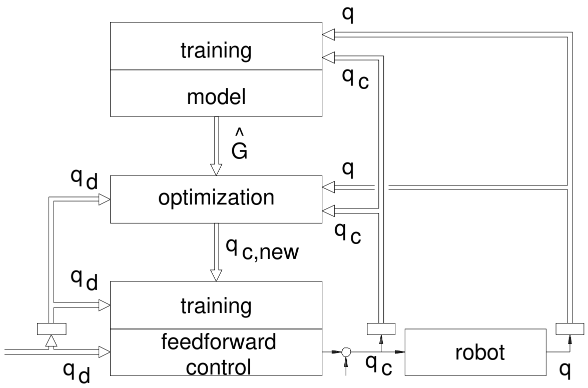

Figure 4: Structure of the learning system in [LH94]

3 Accurate execution of desired paths

[LH94] and [LH96] show a method for compensation of dynamical path errors which are caused by the inability of the robot to follow fast movements accurately and without delay. Such feedforward control requires the knowledge of the desired path at future sampling instants. This method will be summarized in this section.

Dynamical path errors are caused by accelerations of the desired joint trajectories (Section 3.1) and by changes of external forces acting on the robot (Section 3.2). Compensation of those two kinds of influences can assure path accuracy if the path is known in advance (thick lines in Figure 2). This is valid for the off-line defined paths of [LH94] and for the second problem defined in the introduction. For on-line path planning it may occur, however, that the robot looses the optimal path because this path is determined too late. For this case it is advantageous to predict the desired path and / or to feedback the actual path. This is discussed in Section 3.3.

3.1 Learning of optimal feedforward control

Learning of the compensation of path deviations is performed hierarchically in three levels (Figure 4). First, coarse models are built for all joints j, with n_g estimated elements $\hat{g}_{ij}$ of the impulse response function each (q_j is the measured j-th joint angle, q_{cj} is the commanded one).

$$q_j(k) = q_{cj}(k) + \hat{g}_{0j}(k) + \sum_{i=1}^{n_g} \hat{g}_{ij} \cdot (q_{cj}(k-i) - q_{cj}(k)) \tag{14}$$

Then these models are used to modify the commands of the training path so that the control errors are minimized. To do so, for each joint the modifications Δq_{cj} with respect to the previously commanded path q_{cj} are estimated from the control errors $e_j = q_{dj} - q_j$.

$$\begin{pmatrix} \hat{g}_{1j} & & & \\ \cdots & \cdots & & \\ \hat{g}_{n_g j} & & \cdots & \\ & \hat{g}_{n_g j} & \cdots & \hat{g}_{1j} \end{pmatrix} \cdot \begin{pmatrix} \Delta q_{cj}(0) \\ \cdots \\ \cdots \\ \Delta q_{cj}(N-1) \end{pmatrix} = \begin{pmatrix} e_j(1) \\ \cdots \\ \cdots \\ e_j(N) \end{pmatrix} \tag{15}$$

Estimation is preferred to the solution of the system of equations since it generates smoother functions Δq_{cj} if e_j is noisy.

The resulting modified commands can be learned in a feedforward controller being able to control even untrained paths. In [LH96] this parameter adaptive controller is supported by neural nets to be able to compensate nonlinear couplings. The resulting controller for joint j is then

$$\begin{aligned} q_{cj}(k) = q_{dj}(k) + \sum_{i=1}^{n_q} r_{qij} \cdot (q_{dj}(k+i) - q_{dj}(k)) \\ +net_j(\mathbf{q}_d(k), \\ q_{d\hat{j}}(k+1) - q_{d\hat{j}}(k), q_{d\hat{j}}(k+2) - q_{d\hat{j}}(k), .. \\ q_{d\acute{j}}(k+1) - q_{d\acute{j}}(k), q_{d\acute{j}}(k+2) - q_{d\acute{j}}(k), ..) \end{aligned} \tag{16}$$

where $\hat{j}$ and $\acute{j}$ are indices of joints that are seen to influence joint j for the robot on hand. So n_q parameters r_{qij} and one neural net are learned for each joint. The solution with linear part and neural net is chosen because the first line of Equation (16) has to be calculated with high accuracy which is a problem for most neural net architectures. On the other side a multilayer perceptron is adequate for the representation of influences which do not fit to the linear approach.

Equation (16) means that a fictive path $\mathbf{q}_c$ is commanded in order to execute the desired path $\mathbf{q}_d$. The controller parameters r_{qij} can be learned at the installation of the robot and are valid for the whole workspace. [LH94] shows that after iterative learning and control this method yields a reduction of the path errors from $2mm$ to $0.1mm$ for a high speed experiment. The neural nets have to be adapted for each task but are still valid in case of small deviations due to sensor signals.

3.2 Consideration of contact forces

In contrast to data from non tactile sensors, signals of a force-/torque sensor have to be evaluated not only for path planning but also for control. In [LH94] an extension to positional feedforward control of Equation (16) is proposed:

$$q_{cj}(k) = ... + \sum_{i=1}^{n_\tau} r_{\tau ij} \cdot (\tau_{dj}(k+i) - \tau_{dj}(k)) \tag{17}$$

Note that $\tau_d = J^t \cdot \mathbf{F}_d$ is only that part of the desired joint torque vector that is measurable in the sensor. The total value is not important since it changes slowly and statical joint torques are exerted by an integrative part in the standard feedback controller of the robot which is included in the rightmost block of Figure 2.

3.3 Extension of the method for on-line planned paths

If future path information is not available because it is not yet sensed it can be predicted by

$$\mathbf{x}_d(k+i) = \mathbf{x}_d(k) + i \cdot \mathbf{V}_d(k) \cdot T_0. \tag{18}$$

In practice it will suffice to transform only $\mathbf{x}_d(k)$ and $\mathbf{x}_d(k+n_q)$ and interpolate between the resulting joint angle vectors to get the inputs of Equation (16).

In addition the future desired path can be estimated implicitly by

$$q_{cj}(k) = ... + \sum_{i=1}^{n_p} r_{pij} \cdot (q_{dj}(k-i) - q_{dj}(k)) \tag{19}$$

which corresponds to a higher order prediction for constant curvature.

Prediction of the future joint torques is only possible by

$$\tau_d(k+i) = \tau_d(k). \tag{20}$$

For high bandwidth feedback in the standard robot control system no additional feedback is recommended here except for the path planning loop. Such feedback can be advantageous, however, if actual and desired position differ because the desired position has changed. Then the actual joint angles $\mathbf{q}(k)$ have to be adapted to the modified desired joint angles $\mathbf{q}_d(k)$ as fast as possible. This yields a further extension of Equation (16).

$$q_{cj}(k) = ... + \sum_{i=0}^{n_y} r_{yij} \cdot (q_j(k-i) - q_{dj}(k)) + \sum_{i=0}^{n_u} r_{uij} \cdot (q_{cj}(k-i) - q_{dj}(k)) \quad (21)$$

4 Experiments

The experiments are executed with a Manutec r2 robot. According to the remarks at the end of Section 2 a compliant force- / torque sensor (see Figure 1) is used. The sampling time of the feedforward controller is chosen to $T_0 = 16ms$. As a result $n_g = n_q = n_\tau = 10$ is stated. Elasiticity is measured to be about $0.5mm/N$ in the plane of the contour.

For the first problem, tracking the front part of the contour (Figure 1) with a speed of $37.5mm/s$, several approaches are compared. Direct control as in [LH92] can keep the desired force of $10N$ with an RMS error of $0.7N$ after learning of a force controller for the compliance in use (Before learning sensor values cannot be kept within the range of $3N..15N$.). For another sensor learning has to be repeated.

The approach of this paper is able to track the contour with a force error of $1.2N$ working with Equation (9) but without feedforward control. Including positional feedforward control of Equations (16), (18), and (20) reduces the error to $0.9N$. Further improvements can be reached by predicting the shape of the contour by Equation (19), by adding feedback control of Equation (21), or by both. All the same at that desired speed a minimum of $0.7N$ cannot be exceeded since limitations of the robot accelerations prevent all immediate reduction of force errors in case of unexpected curvature.

The second problem is different in the way that the path planning is finished when the path has to be repeated. So the vectors of τ_d and $\mathbf{q}_d$ are available for all sampling instants of interest. This means that the real desired values can be used instead of the predictions according to Equations (18) to (20). In this case path accuracy is not as much restricted by the robot limits. So in contrast to direct force control the same path can be executed again, now with a mean force error of $0.3N$ instead of $0.9N$ during sensing (compare Figure 6 and Figure 5).

Alternatively the speed can be increased according to the limits of the robot. For the whole contour of Figure 1 this means that the time for 1 round is reduced from $26s$ to $3s$ resulting in a top speed of more than $700mm/s$ (see Figure 7).

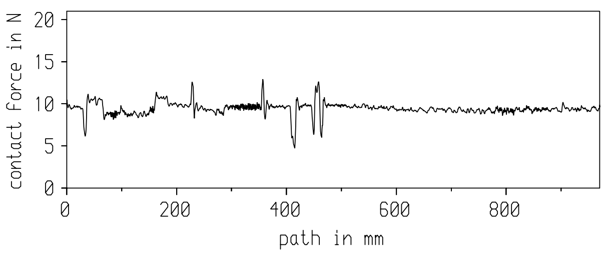

Figure 5: Force history when sensing the contour

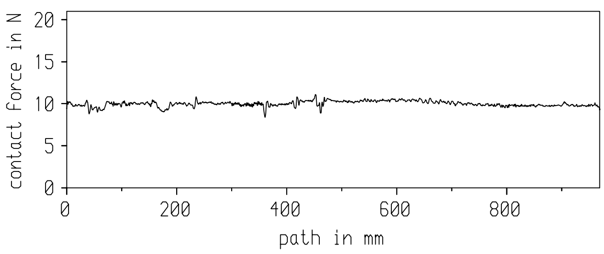

Figure 6: Force history during repetition of the path with the same speed as during sensing

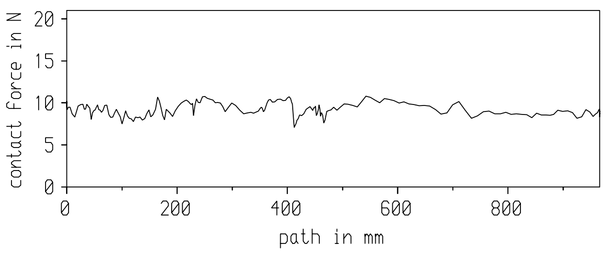

Figure 7: Force history during repetition of the path with a top speed of more than $700mm/s$

The resulting force errors are $1.3N$ when using only the first line of Equation (16), $1.2N$ when force feedforward according to Equation (17) is added, and $0.6N$ if neural nets are provided for the 3rd, the 4th, and the 5th joint. Neural nets for the other joints are not necessary. It should be noted that without feedforward control the contour cannot be tracked at this speed.

This superior behaviour during tracking of the off-line planned paths is valid also for on-line planned paths if the sensor is able to predict the force signal for n_q sampling steps corresponding to $n_q \cdot |\mathbf{v}_d| \cdot T_0 = 6mm$ for Figure 6. For the high speed motion of Figure 7 prediction for $11cm$ is required which at least for vision based contour sensing means no problem. In both cases the sensor signal is different from the force vector to be controlled.

5 Conclusion

The paper demonstrates high accuracy force control applied to a robot with a positional interface. The control error can be minimized, even for movements at maximal speed of the robot, if the path is known in advance. In contrast, if no predictive sensor is available, i.e. if measured and controlled variable are identical, the minimal control error is limited by the maximal accelerations of the robot.

Such high performance results from single learning of the robot dynamics which can be executed during the installation of the robot. No relearning has to be done for changed sensor configurations or different loads since weight and inertia of the load can be treated in the same way as external forces to the TCP. Solely the use of neural nets depends on the problem on hand and training therefore has to be repeated for a different task. A general neural net which acquires only single learning is currently subject of our research.

References

[AAH88] C. H. An, C. G. Atkeson, and J. M. Hollerbach. *Model-Based Control of a Robot Manipulator.* The MIT Press, Cambridge, Massachusetts, 1988.

[CGE94] R. Colbaugh, K. Glass, and A. Engelmann. Decentralized adaptive compliance control of robot manipulators. In *4th IFAC Symposium on Robot Control*, Capri, Italy, 1994.

[Hir93] G. Hirzinger. Rotex - the first robot in space. In *6th Int. Conf. on Advanced Robotics*, Tokyo, Japan, Nov 1993.

[HL85] G. Hirzinger and K. Landzettel. Sensory feedback structures for robots with supervised learning. In *IEEE Int. Conference on Robotics and Automation*, St. Louis, Missouri, March 1985.

[Hog87] N. Hogan. Stable execution of contact tasks using impedance control. In *IEEE Int. Conference on Robotics and Automation*, Raleigh, North Carolina, 1987.

[Kha87] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, RA-3, 1987.

[LH92] F. Lange and G. Hirzinger. Iterative self-improvement of force feedback control in contour tracking. In *IEEE Int. Conference on Robotics and Automation*, Nice, France, April 1992.

[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"*, April 1996.

[Mül93] M. Müller. *Roboter mit Tastsinn.* PhD thesis, Technische Universität Braunschweig, 1993. in German.

[PD94] M. Pelletier and M. Doyon. On the implementation and performance of impedance control on position controlled robots. In *IEEE Int. Conference on Robotics and Automation*, San Diego, California, 1994.

[RC81] M. H. Raibert and J. J. Craig. Hybrid position / force control of manipulators. *Trans. of the ASME*, 102, 1981.

[Sal80] J. K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *IEEE Int. Conf. on Decision and Control*, Albuquerque, New Mexico, Dec. 1980.

[SC93] H. Seraji and R. Colbaugh. Adaptive force-based impedance control. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, Yokohama, Japan, July 26-30 1993.

[Tür90] S. Türk. *Zur Modellierung der Dynamik von Robotern mit rotatorischen Gelenken*, volume 211 of *Fortschritt-Berichte, Reihe 8.* VDI-Verlag, Düsseldorf, 1990. in German.

[WAN95] L. Whitcomb, S. Arimoto, and T. Naniwa. Experiments in adaptive model-based force control. In *IEEE Int. Conference on Robotics and Automation*, Nagoya, Japan, 1995.

[WN79] D. E. Whitney and J. L. Nevins. What is a remote center compliance (rcc) and what can it do? In *9th Int. Symposium on Industrial Robots*, Washington D. C., 1979.