1999 IEEE International Conference on Robotics and Automation
Detroit, Michigan, May 1999

Learning Accurate Path Control of Industrial Robots with Joint Elasticity

Friedrich Lange Gerhard Hirzinger

Institute of Robotics and System Dynamics
Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR)
Oberpfaffenhofen, D-82234 Weßling, Germany
e-mail: Friedrich.Lange@dlr.de

Abstract

An adaptive architecture for feedforward control of industrial robots with standard positional controller is presented. This approach explicitly considers robots with elastical joints. It assumes that the real position of the TCP can be recorded for offline evaluation. Compensation of elasticity is trained in a feedforward controller. Adaptation takes place without any knowledge of the physical system model. The performance of the method is demonstrated in experiments with a 6-axis industrial robot KUKA KR6/1 for which real path errors during full speed motion are reduced by 70 %. Reductions of 50 % can be expected for untrained paths or other robots of the same type.

1 Introduction

Control of elastic robots has been discussed for a number of years (see e.g. [12]). In most cases the arms are assumed rigid, so that elasticity is concentrated in the joints. This seems reasonable for standard industrial robots.

Normally the controller is designed as a feedback controller. Instead, for repetitive control tasks the controller can be implemented as a series of trajectory commands. In this case the standard PD-controller of the industrial system [8] is unchanged.

Another approach is feedforward control. For rigid robots it is well known that accurate trajectory tracking requires a computed torque approach or an equivalent method. This is the same for systems with elastic joints.

In this paper the method of [4] is extended. The standard positional interface of the robot is used, expecting joint commands $\mathbf{q}_c$ and supplying motor positions $\mathbf{q}$ within a sampling time of about 12 ms. The advantage of this approach is, first, that it can be applied to all robots without changes in the hardware, and second, that stabilization and feedback control of the motor positions is left to the standard control system (Figure 2). The algorithm used in this paper focuses on the adaptation of feedforward control parameters.

Most papers assume that the arm positions can be measured [10, 14, 8, 1]. Even the design of observers for full state feedback assumes that the arm positions are measurable and the motor postitions have to be estimated [13, 7]. In contrast, to reach high resolution, common industrial robots measure the angular positions on the motor side of the gear units. Therefore the arm positions have to be computed by an observer which uses the motor positions as inputs. [3] reports that control using only the motor positions and observing the arm positions is possible, but requires a high sampling rate to assure global stability.

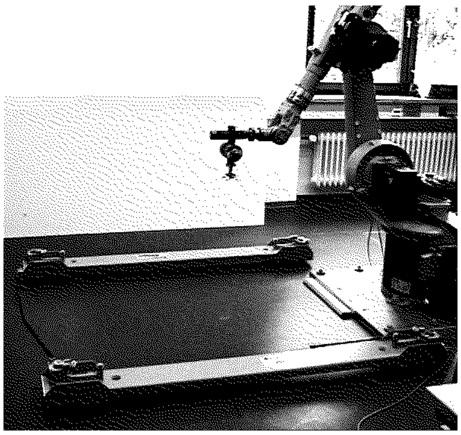

Figure 1: Robot and measuring device [2] in starting position for the test path

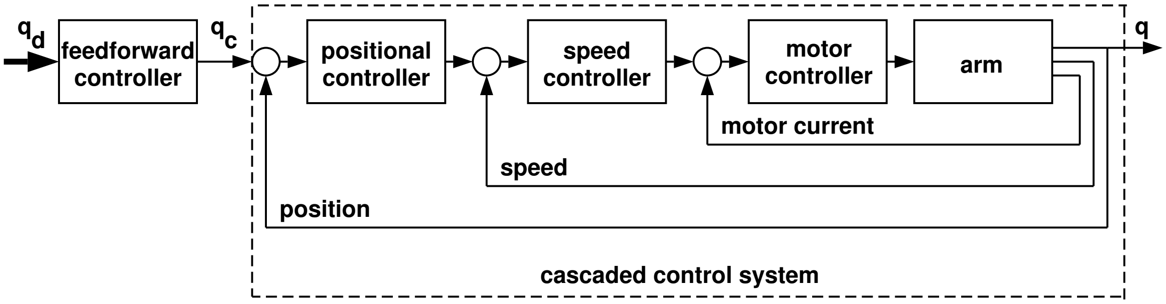

Figure 2: Cascaded positional robot control structure

In contrast we neither have a detailed model of the robot and its feedback control loops nor a high sampling rate. Furthermore industrial robots are not equipped with additional torque sensors as in [11]. So we cannot damp vibrations but only compensate path errors that are caused by elasticity during smooth motion.

For measurement of the robot positions we use a measuring device (see section 3) which can only be evaluated offline. This is no restriction since for feedforward control only the desired path is required. The arm positions are not required after the adaptation.

For the task considered the tool center point (TCP) of the robot has to follow the horizontal path of Figure 3 starting with the configuration of Figure 1. A similar path has already been examined in [5], but without consideration of joint elasiticity. There, the path deviations were reduced by 80 % or more. These path errors were calculated from the motor encoders. The real path error of the TCP could not be measured. Applying an external measuring system, control errors turn out to be somewhat larger (see section 4). In reality path errors are reduced by only 30 %. This proves that elasticity cannot be neglected.

This paper is based on [4, 5]. Therefore section 2 shortly reviews the method. Then elasticity is explicitly considered. In section 4 experiments compare both approaches.

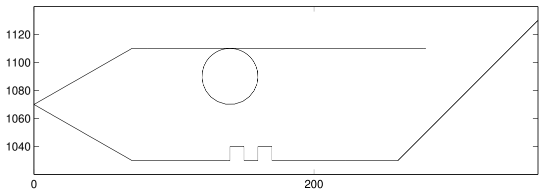

Figure 3: Desired path

2 Learning feedforward control

Learning is performed hierarchically in three levels (see Figure 4). First a coarse model of the robot with its internal feedback control structure (see Figure 2) is built (section 2.1). Then this model is used to modify the commands of the training trajectory such that the control error is minimized (section 2.2). These modifications can be learned in a feedforward controller which thereafter is able to improve tracking of other paths as well (section 2.3). Learning of the optimal commands and of the feedforward controller is repeated iteratively with application of the suboptimal controllers in the meantime. The resulting controller is better than the model. It minimizes differences between actual and desired trajectories with respect to position, orientation, and velocity.

2.1 Identification of a simple model

Experiments show that an I/O model of the robot dynamics can be approximated by linear position independent and decoupled time discrete joint models

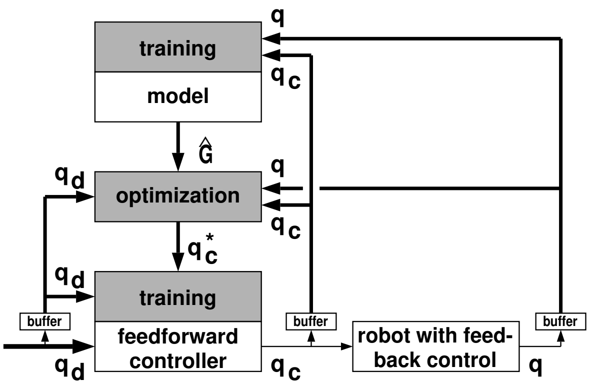

Figure 4: Structure of the learning system

$$q(k+1,j) = q_0(k,j) + \hat{g}_0(k,j) + \sum_{i=1}^{n_g} \hat{g}_i(j) \cdot (q_c(k-i+1,j) - q_0(k,j)) \quad (1)$$

where q_c are the commanded joint angles which effect the measured value q of joint j at time-instant $(k+1)$. q_0 represents the point of lineariziation which may vary over time.

A recursive Kalman filter as a method for linear parameter estimation determines the estimated values $\hat{g}_i(j)$ of the impulse response functions. The theoretical number of elements of this function is infinite, but for asymptotically stable processes it can be limited to e.g. $n_g = 10$ elements that are sufficient when using sampling intervals of 12 ms.

The additional functions $\hat{g}_0(k,j)$ are necessary to represent a bias. To ensure convergence in case of a smooth trajectory it is further necessary to add some stochastical excitation to the commanded values.

2.2 Minimization of the path error

The model of Equation (1) is used to modify the commands of the training path so that the control errors are minimized. This is done in cartesian space since joint errors sometimes compensate each other or influence only less important orientational DOFs of the robot.

Optimal joint commands $\mathbf{q}_c^*$ would result in zero cartesian control error

$$\mathbf{e}_\Phi(k) = \mathbf{\Phi} \cdot \mathbf{J}(k) \cdot (\mathbf{q}_d(k) - \mathbf{q}(k)) \quad (2)$$

which is calculated using the Jacobian $\mathbf{J}$ and the joint control errors. $\mathbf{\Phi}$ is a $(p \times 6)$ selection matrix which extracts those p elements of the cartesian pose for which high precision or accurate velocity is important.

Insertion of $\mathbf{q} = \mathbf{f}(\mathbf{q}_c)$ and $\mathbf{q}_d = \mathbf{f}(\mathbf{q}_c^*)$ from Equation (1) into Equation (2) yields the following system of equations

$$\begin{pmatrix} \mathbf{\Phi J}(1) & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \mathbf{\Phi J}(N) \end{pmatrix} \cdot \begin{pmatrix} \hat{\mathbf{G}}_1 & & & \\ \vdots & \ddots & & \\ \hat{\mathbf{G}}_{n_g} & & \ddots & \\ & \hat{\mathbf{G}}_{n_g} & \cdots & \hat{\mathbf{G}}_1 \end{pmatrix} \cdot \begin{pmatrix} \Delta\mathbf{q}_c(0) \\ \vdots \\ \vdots \\ \Delta\mathbf{q}_c(N-1) \end{pmatrix} = \begin{pmatrix} \mathbf{e}_\Phi(1) \\ \vdots \\ \vdots \\ \mathbf{e}_\Phi(N) \end{pmatrix} \quad (3)$$

with $\mathbf{\Delta q}_c = \mathbf{q}_c^* - \mathbf{q}_c$. $\hat{\mathbf{G}}_i$ is the diagonal matrix of the joint impulse response functions $\hat{g}_i(j)$.

Instead of the solution of the system of equations we prefer estimation of the $\mathbf{\Delta q}_c$ since estimation smoothes if $\mathbf{e}_\Phi$ is noisy. 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 $N \gg n_g$. Convergence is improved if 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 predictions of the models.

2.3 Estimation of the controller parameters

For control, $\mathbf{q}_c^*$ has to be calculated by a feedforward controller which cannot predict future control errors. First of all, linear feedforward filters on joint level are set up to represent $\mathbf{q}_c^*$ in relation to subsequent desired positions

$$q_c(k,j) = q_d(k,j) + \sum_{i=1}^{n_d} r(i,j) \cdot (q_d(k+i,j) - q_d(k,j)). \quad (4)$$

Beyond that, in [5] additional parameters and neural nets are provided for a more precise calculation of $\mathbf{q}_c^*$ in spite of nonlinearities and couplings. But this is not necessary for the experiments of this paper.

The parameters $r(i,j)$ of Equation (4) are estimated using a recursive Kalman filter as well with $n_d = 15$ for our test configuration.

This setup is quite independent from the trained path and therefore well suited for sensor control. Such feedforward control is superior to any algorithm implemented in the robot controller because information about the future path can be provided e.g. by planning a motion sequence from a camera image [6].

3 Consideration of elasticity during training

So far, the method minimizes the difference between the desired path and the assumed position of the TCP. The assumption uses the joint positions that are provided by the robot system. These joint positions are measured on the motor side of the gear units so that in case of elasticity the arm positions are not

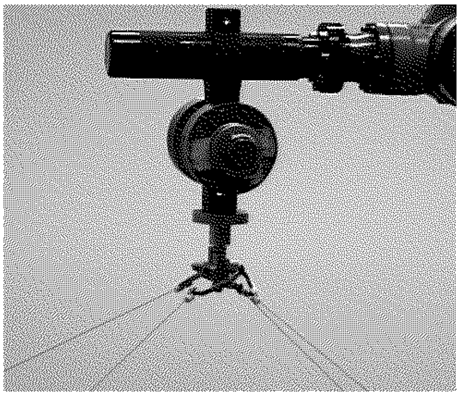

Figure 5: Robot endeffector with fixture for cables of the measuring device [2]

correctly represented. The real position of the TCP has to be measured by an external device.

Up to now it has been further assumed that the position is measurable in all 6 degrees of freedom (DOF). Instead, in simple setups the sensor will measure only 1 DOF as in [6]. In our experiments a 3 DOF cartesian sensing device (Figures 1 and 5) is used. The location of the TCP is calculated by its distance with respect to 4 fixed positions [2].

This measuring device has to be aligned to the robot coordinate system. After calibration an accuracy of about 0.1 mm is reached.

Nervertheless it is difficult to evaluate the measured information in the direction of the motion since the time instant of the measurement will not be identical to that of the measurements of the motor angles (At a speed of 1 m/s a time difference of 1 ms yields a positional error of 1 mm.). Therefore the 3 DOF values are transformed into a time-variant coordinate system of motion (Figure 6) of which only 2 DOF are used so that the direction of motion is excluded. This means

$$\mathbf{x}_{pa}(k) = \mathbf{T}_p(k) \cdot \mathbf{x}_a(k) \tag{5}$$

where $\mathbf{T}_p$ is the transformation to the coordinate system of motion and $\mathbf{x}_a$ is the measured position of the

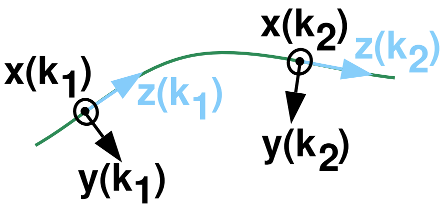

Figure 6: Time-variant coordinate system of motion

TCP in the world system. The definition of the coordinate system of motion is not unique. But this is insignificant if the adaptation is done as in section 2.2 because $\mathbf{\Phi}(k) = \mathbf{T}_p(k)$ changes the left side of Equation (3) as well. So the path error $\mathbf{e}_p$ is minimized

$$\mathbf{e}_\Phi(k) = \mathbf{e}_p(k) = \mathbf{x}_{pa}(k) - \mathbf{x}_{pd}(k) \tag{6}$$

where $\mathbf{x}_{pd}$ is the desired position expressed in the 2 DOF system of motion.

If the measurements are costly, e.g. because of a missing synchronization, it is advisable to record the difference

$$\mathbf{\Delta x}_e(k) = \mathbf{x}_a(k) - \mathbf{x}(k) \tag{7}$$

between the real arm positions $\mathbf{x}_a(k)$ and the positions $\mathbf{x}(k)$ calculated from the robot encoders. These differences will remain constant during adaptation since the elastic deformation is caused by acceleration forces and acceleration will not change seriously during adaptation. So it is sufficient to detect the deformation once and then to add it as an offset in further iterations of learning. After training we propose to repeat the measurements of Equation (7) with the adapted controller and to check if the calculated positions still represent $\mathbf{x}_a(k)$ with the required accuracy. If this is correct, training is finished with two measured trajectories (after the calibration). Otherwise adaptation has to be continued with the new values for the elastic deformation $\mathbf{\Delta x}_e(k)$.

4 Experiments

For the experiments a KUKA KR6/1 industrial robot with the standard industrial controller KUKA KRC1 is used. The experiments are executed at the reference path of Figure 3 with full speed according to the path planning algorithm of the KRC1. It turns out that path deviations occur mainly during the circle which therefore is zoomed for displaying the learning results (Figures 7 to 9).

Small circles seem to be a challenge for control of elastic robots. No doubt, bigger shapes which can be executed at higher speed show bigger path errors but these deviations can normally be compensated by methods as [5] which do not consider elasticity. In contrast during execution of small circles internal and external methods to measure the actual position show substantial differences (Figure 7). Hereby we call a method internal if it measures the motor position and external if the arm positions or the cartesian position of the TCP can be recorded directly.

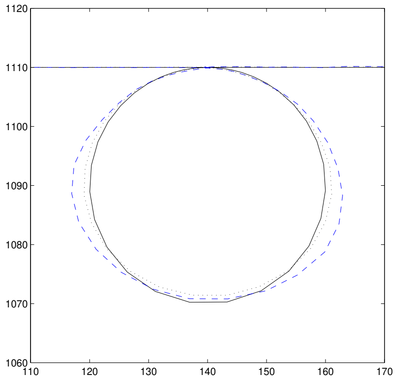

Figure 7: Performance calculated from internal (dotted) or external (dashed) measurements (solid = desired path)

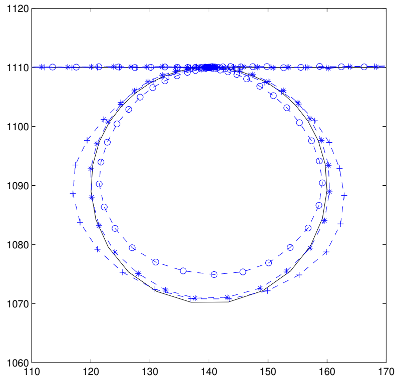

Figure 8: Performance with different methods: ∘ = standard controller, + = universal controller, ∗ = explicitly trained controller, solid = desired path

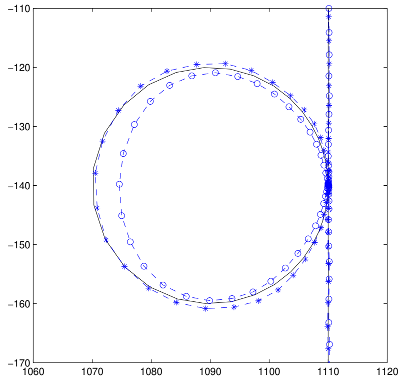

Figure 9: Performance with another robot: ∘ = standard controller, ∗ = explicitly trained controller for the first robot, solid = desired path

Training a controller using only internal measurements tries to enlarge the circle which is executed too small without feedforward control. Figure 8 displays that a universal controller trained without external measurements cannot consider the joint dynamics correctly. In this example the first joint shows perceptible compliance (Figure 7).

Adapting the feedforward controller according to Equations (3) and (4) with the real cartesian control errors (6) yields satisfactory performance (Figure 8). The improvement with respect to the standard robot controller is still evident when executing other paths (Table 1). In this case a path of larger horizontal and vertical circles is executed with higher speed.

Even when using other robots of the same type, path errors are substantially smaller than with the standard controller (Figure 9). Such generalization ability is fundamental for the applicability of the method. By this the necessity of an external measurement system can be restricted. It is not required to install a measurement device whenever the robot path is changed. Instead, either the feedforward controller is maintained or the controller is changed with the path. The latter means that the measurement device is not installed at each robot in the production line but only at one robot in a lab where the new paths are programmed. So the advantage of learning controllers, the adaptivity to the actual process can be exploited optionally.

The differences between the two robots show the reachable accuracy. It cannot be surpassed by any method as long as only measurements of other robots or informations about the nominal physical model are available. Better performance requires data from the robot used, especially measurements of the arm positions, if possible. Otherwise we try to observe the TCP position from recorded internal measurements. But this turns out to be difficult since the internal measurements are hardly affected by different characteristics of the gears (see Table 1). The resulting path errors of the TCP can only be smaller than the compliance if the motors track a fictive path that compensates path deviations due to elasticity. This compensation is sensitive to phase differences. By chance, the phase is favourable with the standard cascaded control system but adverse to training using internal measurements (see Table 1).

5 Conclusion

A learning method has been applied to improve the accuracy of industrial robots at paths for which compliance cannot be neglected. This implies external measurements of the TCP position for which high accuracy can be reached with lengthy calibrations only. Fortunately, such work is required only once since the method combines feedback of the motor positions with learned feedforward to compensate for dynamical delays including compliance. This feedforward controller does not process any online measurements.

In future we will observe the real TCP position, first, to improve the generalization of the method, and second, to allow accurate sensing with a robot mounted sensor, which up to now is the limiting factor for high speed sensor control [6].

Differences between	desired - motor		motor - arm		desired - arm	
	rms	max	rms	max	rms	max
performance before and after training						
standard (cascaded) controller	0.766	5.450	0.261	1.509	0.626	5.075
universal feedforward controller	0.200	1.269	0.342	2.290	0.459	3.225
explicitly learned controller	0.312	1.725	0.303	1.932	0.202	0.853
performance at another path						
standard (cascaded) controller	1.834	2.431	0.599	1.223	1.325	2.200
explicitly learned controller	0.599	1.227	0.615	1.289	0.411	1.284
explicitly learned controller for that path	0.514	1.043	0.617	1.297	0.336	1.097
performance with another robot						
standard (cascaded) controller	0.768	5.473	0.372	1.866	0.584	4.891
expcicitly learned controller	0.309	1.709	0.457	2.334	0.306	1.665

Table 1: Root mean square and maximal cartesian path differences in mm. All measures are calculated with respect to the TCP. Training is not repeated before applying the feedforward controller to the other path or the other robot. Only the explicitly learned controller considers elasticity.

References

[1] A. de Luca and P. Lucibello. Dynamic feedback linearization of robots with elastic joints. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 505–510, Leuven, Belgium, 1998.

[2] Dynalog, Inc. *3D CompuGauge: User's Manual*, 1997. Version 2.7.

[3] M. Jankovic. Observer based control for elastic joint robots. *IEEE Trans. on Robotics and Automation*, 11(4):618–623, 1995.

[4] F. Lange and G. Hirzinger. Learning of a controller for non-recurring fast movements. *Advanced Robotics*, 10(2):229–244, April 1996.

[5] F. Lange and G. Hirzinger. Adaptive minimization of the maximal path deviations of industrial robots. In *European Control Conference ECC'99*, Karlsruhe, Germany, August / September 1999. Already available at www.robotic.dlr.de/Friedrich.Lange/.

[6] F. Lange, P. Wunsch, and G. Hirzinger. Predictive vision based control of high speed industrial robot paths. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 2646–2651, Leuven, Belgium, May 1998.

[7] N. Léchevin and P. Sicard. Observer design for flexible joint manipulators with parameter uncertainties. In *IEEE Int. Conf. on Robotics and Automation*, pages 2547–2552, Albuquerque, New Mexico, April 1997.

[8] J. H. Lee, B. H. Lee, S. M. Lee, and C. Y. Chung. Preshaped trajectory command for fast repetative PTP motion of PD-controlled flexible joint manipulators. In *IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, Sept. 1997.

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

[10] Z. Qu. Input-output robust tracking control design for flexible joint robots. *IEEE Trans. on Automatic Control*, 40(1):78–83, 1995.

[11] J.-X. Shi, A. Albu-Schäffer, and G. Hirzinger. Key issues in the dynamic control of lightweight robots for space and terrestrial applications. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 490–497, Leuven, Belgium, 1998.

[12] M. W. Spong. Modeling and control of elastic joint robots. *Trans. of the ASME, Journal of Dynamic Systems, Measurement, and Control*, 109:310–319, Dec. 1987.

[13] P. Tomei. An observer for flexible joint robots. *IEEE Trans. on Automatic Control*, 35(4):739–743, 1990.

[14] W. Yim. Modified nonlinear predictive control of elastic manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 2097–2102, Minneapolis, Minnesota, April 1996.