Robot Force/Position Control Combined with ILC for Repetitive High Speed Applications

Herbert Parzer[1(✉)], Hubert Gattringer[1], Andreas Müller[1], and Ronald Naderer[2]

[1] Institute of Robotics, Johannes Kepler University Linz, Altenbergerstraße 69, 4040 Linz, Austria
{herbert.parzer,hubert.gattringer,a.mueller}@jku.at
[2] FerRobotics Compliant Robot Technology GmbH, Altenbergerstraße 69, 4040 Linz, Austria
ronald.naderer@ferrobotics.at
http://www.robotik.jku.at, http://www.ferrobotics.at

Abstract. In this paper an approach for robot force/position control combined with an iterative learning control is proposed. Following high speed force trajectories in different repetitive robotic applications is a challenging field in robotics. Such applications require a desired contact force while following a position/orientation trajectory in the non-force controlled directions. For this a parallel force/position control is suitable, but when it comes to high speed tasks with varying contact stiffness along the trajectory such a method reaches its dynamical limit. The problem can be solved by using the parallel force/position control to learn the trajectory for a slowed down task and correct this trajectory step by step towards the original task speed. When the original task speed is reached an iterative learning control law combined with the force/position control is used to further reduce the force error.

Keywords: Robot force/position control · Iterative learning control · High speed applications · Non-linear contact stiffness

1 Introduction

In many industrial applications manipulators are used to perform repetitive force controlled tasks. Typically, such tasks are polishing, grinding, assembly as well as endurance testing of machine parts. The repetitive nature of such tasks allows for using iterative learning control (ILC) methods, see a brief survey in [2], or adaptive learning feed–forward control, as shown in [8]. While adaptive methods change controller parameters, classical ILC modifies the input of the stabilized system in a serial or parallel way, explained in [3]. Iterative learning control theory is a wide field with different approaches for iteratively minimizing an error between an actual and a desired control signal. These approaches or update laws start from simple proportional (P–type) over "PID–like" types to higher order

© Springer International Publishing AG 2017
A. Rodić and T. Borangiu (eds.), *Advances in Robot Design and Intelligent Control*,
Advances in Intelligent Systems and Computing 540, DOI 10.1007/978-3-319-49058-8_2

types (HOILC), see [1,5]. Such a "PID–like" ILC has typically the form

$$\mathbf{u}_j = \mathbf{u}_{j-1} + \boldsymbol{\Phi}\,\mathbf{e}_{j-1} + \boldsymbol{\Gamma}\,\dot{\mathbf{e}}_{j-1} + \boldsymbol{\Psi} \int \mathbf{e}_{j-1}\mathrm{d}t, \tag{1}$$

where $\boldsymbol{\Phi}$, $\boldsymbol{\Gamma}$ and $\boldsymbol{\Psi}$ are the learn gain matrices, $\mathbf{u}_j$ is the input vector of the system, $\mathbf{e}$ is the error and j the trial index. Further there can be distinguished between *causal* or *non–causal* ILC methods, where the latter ones are able to include zero–phase filtering of the error signal and compensate the delay time of the system. Using such a *non–causal* P–type ILC method often leads to a fast convergence within a few iterations and then to a divergence of the error. This bad transients are explained by Longman in [4] because of the growth of high frequencies in the signals and can be overcome with signal filtering and the introduction of a linear phase–lead compensation. Iterative learning control was also successfully tested for a parallel robot with high speed motions in [1], using P–type as well as PID–type iterative learning.

This paper introduces a novel method of time scaled force/position control and iterative learning control for position controlled robotic manipulators following fast force trajectories.

2 Force/Position Control

Considering a periodic task, where a robot processes or tests the same kind of workpiece in a recurring manner, the end–effector of the robot has to provide a predefined contact force ${}_I\mathbf{f}_d$, given in the inertial frame (I), while following a trajectory along the workpiece. This trajectory is defined by the desired end–effector position ${}_I\mathbf{r}_{E,d}$ and the desired rotation matrix $\mathbf{R}_{IE,d}$ transforming the end–effector frame (E) into the inertial frame. To achieve this goal, a parallel force/position robot control, as suggested in [7], is best suited. Thereby, the force control manipulates a desired end–effector trajectory, of a position controlled system, in such a way that the desired force is reached. The force control law is described by the differential equation $\mathbf{K}_A\,{}_I\ddot{\mathbf{r}}_c + \mathbf{K}_V\,{}_I\dot{\mathbf{r}}_c = {}_I\mathbf{f}_d - {}_I\mathbf{f}$, where $\mathbf{K}_A$ and $\mathbf{K}_V$ denote positive definite controller parameters and the vector ${}_I\mathbf{f}$ represents the measured force. Using the parallel composition ${}_I\mathbf{r}_r = {}_I\mathbf{r}_{E,d} + {}_I\mathbf{r}_c$, with ${}_I\mathbf{r}_c$ as solution of the above differential equation, the reference trajectory ${}_I\mathbf{r}_r$ is calculated. This reference trajectory serves as input of the position controlled robotic system.

3 Iterative Learning Control

However, following a fast desired force trajectory leads to various difficulties. First of all, the exact contact position as well as the contact stiffness is not always known. Further, the stiffness is non–linear and may vary from one point on the surface to another. So, especially for fast trajectories it is difficult to adjust the controller parameter for stable force tracking. Inspired by ILC, the idea is to divide the task into an *on–line* task, where the robot is actually moving, and an *off–line* task, where the computation and trajectory adaptation is done.

3.1 Iterative Force Control

Different to classical ILC methods, the previous mentioned issues are overcome by controlling the task with stretched time by a factor $s_t \geq 1$, depending on the task. In this time stretched *on–line* task the robot is able to follow the desired position and force trajectory safely. So by adding in the actual step j the stored and (*off–line*) non–causal filtered force controller output $F\left[{}_I\mathbf{r}_{c,j-1}\right]$ to the reference trajectory ${}_I\mathbf{r}_r$, the force controller only acts on small force errors, arising from contact damping and system dynamics. The learning law during the time stretched iterations can be written to

$$ {}_I\mathbf{r}_{m,j} = {}_I\mathbf{r}_{m,j-1} + F\left[{}_I\mathbf{r}_{c,j-1}\right], \tag{2} $$

where ${}_I\mathbf{r}_m$ is introduced as memory trajectory. To improve performance, the trajectory from the start to the contact point (and back from the last contact point to the start point) is calculated *off–line* using splines with continuous transition conditions at start and end. With each further iteration the factor s_t is reduced step by step towards 1, representing the original trajectory. Assuming stable force control, the force error at original speed is close to zero and the memory trajectory is trained within a few iterations. Due to the limited dynamics of the force controller further learning will not improve the result. By comparing this method with a P–type ILC one can easily see, that the learning gain $\boldsymbol{\Phi}$ is equal to 1. So for long term learning this is rather aggressive and would lead to an unstable behaviour, as discussed in [4].

3.2 Anticipatory P–type Learning

If it is necessary to further decrease the error a combination of the modified force control learning law from Sect. 3.1 and an anticipatory P–type learning law can be used. Typically such an anticipatory P–type learning law is given with $\mathbf{u}_j(k) = \mathbf{u}_{j-1}(k) + \boldsymbol{\Phi}(k)\,\mathbf{e}_{j-1}(k+1+l)$, where k is the time step and l the linear phase–lead compensation. It is worth to notice, that it is also possible to have a time varying gain matrix $\boldsymbol{\Phi}$. In most cases of standard ILC the signal $\mathbf{u}$ and the error signal $\mathbf{e}$ are from the same type, e.g. joint positions of a robot, and there is no coupling resulting in a diagonal gain matrix. However, controlling contact forces with a position controlled robot system requires as input an end–effector position in the inertial frame, while the output is given by measured forces. So it is obvious that the relationship is defined by the contact stiffness. The contact force can be described by a non–linear function ${}_I\mathbf{f} = {}_I\mathbf{f}\left({}_I\mathbf{r}_E, \mathbf{R}_{IE}\right)$ depending on the end–effector position ${}_I\mathbf{r}_E$ and orientation $\mathbf{R}_{IE}$. Typically there is a desired force at a desired orientation required and so the function of the contact force can be linearized around this point to

$$ {}_I\mathbf{f} \approx \underbrace{{}_I\mathbf{f}\left({}_I\bar{\mathbf{r}}_{E,d}, \mathbf{R}_{IE,d}\right)}_{={}_I\mathbf{f}_d} + \underbrace{\left.\frac{\partial_I\mathbf{f}}{\partial_I\mathbf{r}_E}\right|_{\left({}_I\bar{\mathbf{r}}_{E,d}, \mathbf{R}_{IE,d}\right)}}_{=:\mathbf{C}_{local}\left({}_I\bar{\mathbf{r}}_{E,d}, \mathbf{R}_{IE,d}\right)} \Delta_I\mathbf{r}_E, \tag{3} $$

where ${}_I\bar{\mathbf{r}}_{E,d}$ is the desired end–effector position which will result in zero force error. Using this local contact stiffness matrix $\mathbf{C}_{local}$, which also depends on the orientation, would have a huge impact on the accuracy and also on the stability of the system. So, with additional iterations in the time stretched task and slightly changed desired force vectors, the estimated stiffness matrix $\hat{\mathbf{C}}_{local}$ can be identified from measurements. Substituting the estimated stiffness matrix in (3) a position correction for a measured force error can be calculated to $\Delta_I \mathbf{r}_{m,Pcorr} = \hat{\mathbf{C}}_{local}^{-1}\left({}_I\mathbf{f} - {}_I\mathbf{f}_d\right) = -\hat{\mathbf{C}}_{local}^{-1}\,\Delta_I\mathbf{f}$. By using this in the P–type ILC the position correction

$$\Delta_I \mathbf{r}_{m,Pcorr,j-1}(k) = -\boldsymbol{\Phi}\,\hat{\mathbf{C}}_{local}^{-1}\left(\mathbf{R}_{IE,d}(k)\right)\,F\left[\Delta_I\mathbf{f}_{j-1}(k+1+l)\right] \qquad (4)$$

is derived, where F is again a non–causal zero–phase filter to cutoff high frequencies. In this case the new update law reads in contrast to Sect. 3.1 ${}_I\mathbf{r}_{m,j} = {}_I\mathbf{r}_{m,j-1} + \Delta_I\mathbf{r}_{m,Pcorr,j-1}$. The additional linear phase–lead compensation l is used to in increase the learning bandwidth. Without the knowledge of the local stiffness it would be critical to tune the ILC because a too high estimated stiffness will lead to a very poor convergence while a too low stiffness will cause instabilities, especially when the stiffness changes along the orientation. So, by using the system information of the contact stiffness in the ILC law the convergence is improved and the stability depends on the choice of the gain matrix, the cutoff frequency of the filter and the phase–lead compensator.

3.3 Combined ILC

With the *off–line* position correction from (4) a zero error convergence is not possible due to the filtering of the force error, but with well chosen parameters satisfying force tracking can be achieved. However, one big disadvantage of this method is that the learning control is only able to react on (periodic) disturbances after one trial, whereas the *on–line* force control method, which is used to create the basic memory trajectory, directly reacts on disturbances. In case of disturbances within the bandwidth of the force controller the force error is already in the actual trial reduced. Assuming a periodic disturbance the position correction will cancel out the remaining tracking error. A faster convergence can be achieved by taking the output of the force controller into account, like it is done by the P–type learning. To do this, a further correction of the memory trajectory can be done by

$$\Delta_I \mathbf{r}_{m,Fcorr,j-1}(k) = \boldsymbol{\Phi}\,F\left[{}_I\mathbf{r}_{c,j-1}(k+1+l)\right], \qquad (5)$$

where again the gain matrix, the filter and the phase–lead compensation are used.

All these corrections and controllers for a high speed force control can be summarized in the following statements. First of all, with the force control method and the time stretching a memory trajectory is trained to achieve good force tracking. In the time stretched trajectories the contact stiffness is identified for the ILC and within a few iterations the original speed can be reached. To further improve performance an anticipatory P–type ILC weighted with the contact stiffness is used. Combined learning with the force controller is possible and the whole update law at the original speed is given by

$$_I\mathbf{r}_{m,j} = {}_I\mathbf{r}_{m,j-1} + \Delta_I\mathbf{r}_{m,Pcorr,j-1} + \Delta_I\mathbf{r}_{m,Fcorr,j-1}. \qquad (6)$$

In the next section an example is shown, to clarify the theory above.

4 Example

The following example is based on an experiment published by the authors in [6], where only the time stretching method is introduced. Thereof the gained data are used to build a realistic simulation and so to test the combined ILC method.

4.1 Experiment

For the experiment a six–axis industrial robot (Stäubli TX90L) controlled by decentralized feed–back motor position controllers and by centralized trajectory generation and force control is used. At the end–effector of the robot a curved tool, see Fig. 1, with the radius R is mounted. In the scenario the tool contacts the surface and moves with a smooth roll–over motion from $\beta_d = 25°$ to $\beta_d = -25°$ in $T = 0.75$ s over the surface and back. During the contact a desired force of $_I f_{z,d} = 100 \cos\left(\frac{\pi}{50}\beta_d\right)$ N in z–direction is required. For the time stretched trajectories the scaling factors $s_t = \{16, 16, 4, 1\}$ were used to track the desired force with a maximum force error of less than 3.5%. Figure 2 shows the desired and the measured force with learning force control from Sect. 3.1 (*FCTRL+MEM*, iteration 4) and with pure force control from Sect. 2 (*FCTRL*) at original speed.

4.2 Simulation

To test the above discussed update laws the simulation of the whole system is extended with the new control scheme and a position/orientation depended contact model is implemented. The contact model, see Fig. 3, is estimated by using the data from five measurements (blue lines) of the non–linear stiffness at discrete orientations β. This surface is described by B–splines in the β–Δz–plane, where Δz is the difference between the end–effector point of the robot and the point on the elastic object in inertial z–direction. In Fig. 3 also the on the surface projected desired force $_I f_{z,d}$ (red line) is shown. Considering this, it is clear to see that the local stiffness around the desired force (marked with green) varies along this path. The continuous local stiffness function $c_{local}(\beta)$ is shown in Fig. 4. Figure 5 shows the robot control scheme for the simulation. First there is the *Trajectory* block, where the desired position and orientation for the roll–over

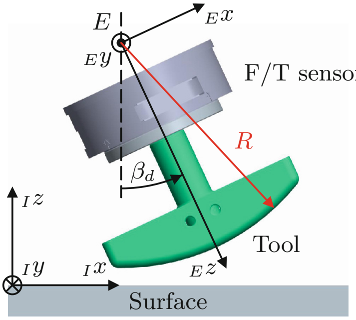

Fig. 1. Setup of the experiment

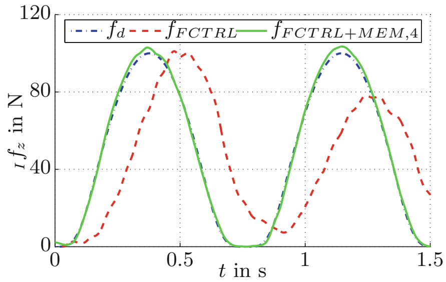

Fig. 2. Measurements from the experiment

motion is calculated. The *Inv. Kin.* block transforms task space coordinates $_I\mathbf{r}_r$ and $\mathbf{R}_{IE,d}$ to joints coordinates $\mathbf{q}_r$. They are passed to the position–controlled non–linear robot model, whose output is the actual end–effector position $_I\mathbf{r}_E$ and orientation $\mathbf{R}_{IE}$. The contact model from Fig. 3 is used to calculate the actual force $_I\mathbf{f}$ and the force controller modifies the desired trajectory, based on the force error. For the iterative learning control the modification as well as the force error is stored *off–line* and the thereof calculated memory trajectory is passed to the system *on–line* in the next trial.

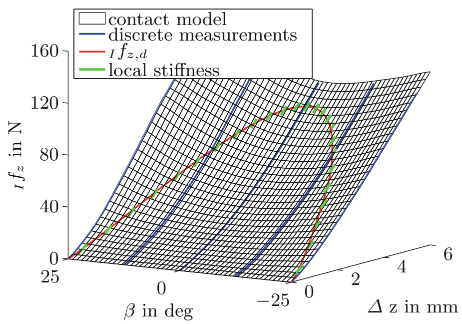

Fig. 3. Calculated contact model

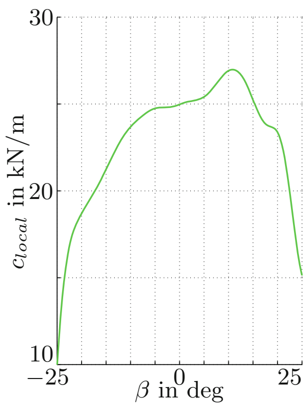

Fig. 4. Local stiffness

To study the learning behaviour the same desired trajectories as in the experiment are used and after 9 iterations a $\cos^2$–shaped periodic disturbance with an amplitude of $_I f_{z,dist} = 20$ N is added. This is done to compare three types of learning. The first type (*ILC*) is only learning with the weighted anticipatory P–type ILC from Sect. 3.2 while for the second type (*ILC+FCTRL*) additionally the force controller is switched on. Finally, the third type (*Combined ILC*) is the

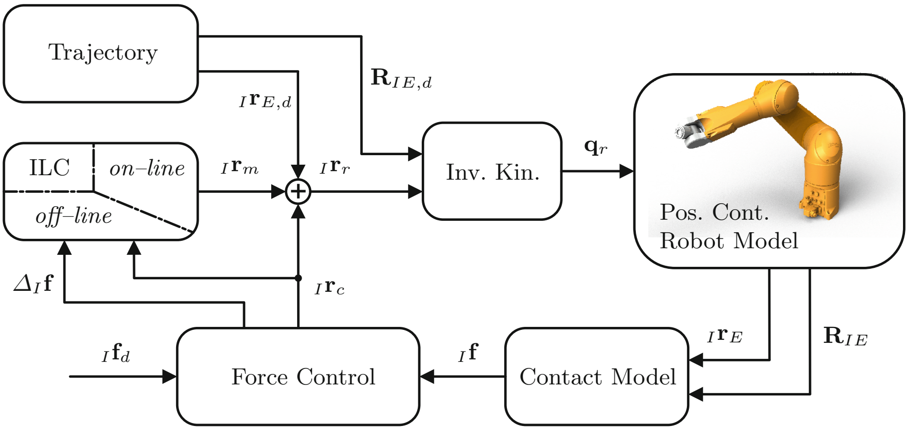

Fig. 5. Robot control scheme in the simulation

combined ILC, where also the output of the force controller is used to further train the memory trajectory, see (6). Identifying the local stiffness (Fig. 4) of the contact model is done during the time stretched iterations by simply following a slight changed desired force trajectory, as it would be done in reality. Figures 6 and 7 shows the convergence of the Euclidean norm of the force error from the first up to the 20th iteration and from the 20th to the 50th iteration, respectively. Thereby the iterations are counted since the original speed is reached. As one can see, additionally using the force controller can reduce the effect of the disturbance can be reduced. By also considering the output of the force controller in the combined learning strategy the convergence rate will be equivalent to that of the pure ILC. Not using this output results in a slower convergence rate of the second type. For all of these three types it is important to notice that

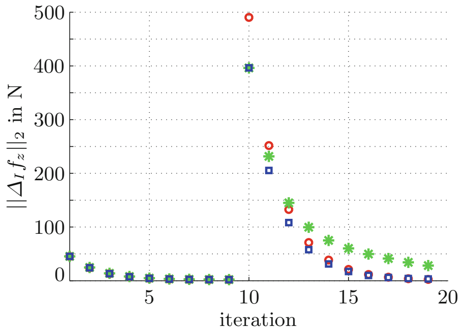

Fig. 6. Error norm (1–20)

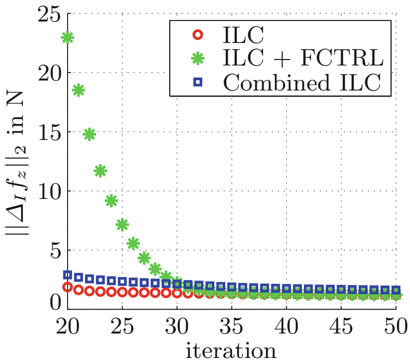

Fig. 7. Error norm (20–50)

if the local stiffness is not identified and estimated constant and too weak the stability of the ILC can not be guaranteed.

5 Conclusion

This paper introduces a promising combination of parallel force/position control for robotic systems and iterative learning control for repetitive high speed trajectories. While the force/position control has its limits in stability for fast force trajectories and varying contact stiffness, the approach of a time scaling iterative learning method shows best results. To further increase accuracy a combined method of anticipatory P–type learning weighted with system information and the force control itself is introduced. Simulation results shows good convergence and robustness, so that further research will focus on implementation and experiments.

Acknowledgment. This work has been supported by the Austrian COMET-K2 program of the Linz Center of Mechatronics (LCM), and was funded by the Austrian federal government and the federal state of Upper Austria.

References

1. Abdellatif, H., Feldt, M., Heimann, B.: Application study on iterative learning control of high speed motions for parallel robotic manipulator. In: 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, pp. 2528–2533. IEEE (2006)
2. Ahn, H.S., Chen, Y., Moore, K.L.: Iterative learning control: brief survey and categorization. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **37**, 1099 (2007)
3. Bristow, D.A., Tharayil, M., Alleyne, A.G.: A survey of iterative learning control. IEEE Control Syst. **26**, 96–114 (2006)
4. Longman, R.W.: Iterative learning control and repetitive control for engineering practice. Int. J. Control **73**, 930–954 (2000)
5. Owens, D.H., Hätönen, J.: Iterative learning control - an optimization paradigm. Ann. Rev. Control **29**, 57–70 (2005)
6. Parzer, H., Gattringer, H., Müller, A., Naderer, R.: Learning robot force/position control for repetitive high speed applications with unknown non-linear contact stiffness. PAMM **15**, 65–66 (2015)
7. Siciliano, B., Villani, L.: Robot Force Control. Springer Science & Business Media, New York (1999)
8. Velthuis, W.J., de Vries, T.J., van Amerongen, J.: Design procedure for a learning feedforward controller. In: Proceedings of the First IFAC Conference on Mechatronic Systems, pp. 955–960 (2000)

http://www.springer.com/978-3-319-49057-1

Advances in Robot Design and Intelligent Control
Proceedings of the 25th Conference on Robotics in
Alpe-Adria-Danube Region (RAAD16)
Rodić, A.; Borangiu, T. (Eds.)
2017, XVII, 649 p. 372 illus., Softcover
ISBN: 978-3-319-49057-1