Learning Time Optimal Control of Smart Actuators with Unknown Friction

H. Trogmann * **P. Colaneri** ** **L. del Re** *

* *Institute for Design and Control of Mechatronical Systems, Johannes Kepler University Linz, 4040 Linz, Austria (email:* {hannes.trogmann,luigi.delre}@jku.at)
** *Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano, Italy (email:* colaneri@elet.polimi.it)

Abstract: Active valves are most effective tools to control gas flow in compressors if fast transitions between the open mode and closed mode are needed. Unfortunately, an accurate model including several nonlinear effects and in particular the resistance and gas flow forces is not available, and this prevents the use of standard model based approaches for time optimal control. However, the repetitive nature of the operation of valves suggests the use of learning methods to track a reference in spite of the insufficient information on the control behavior, thus shifting the problem from the search of the time optimal control to the search of the reference corresponding to its solution. To this end, in this paper, a previously proposed algorithm for the iterative determination of the fastest feasible trajectory is analyzed in terms of convergence conditions and applied to the valve model.

1. INTRODUCTION

Compressors are commonly used in a variety of different industrial applications, see e.g. Fig. 1. They rely on the switching of high and low pressure chambers, and the precise and fast control of the transition is critical for their perfomance and efficiency. Indeed, flow control can be achieved in different ways, e.g. by changes of the stroke of the piston, activation or deactivation of waste pockets, use of waste gates and by timing control of the inlet valves of the compression chambers. This last option seems to combine low costs, flexibility and efficiency in the best way.

Fig. 1. An industrial two stage compressor, the two cylinders of stage one can be seen (blue)

To be able to provide such a control possibility it is necessary to use active valves with enough power to hold the port open during the compression phase and to close it in a split second.

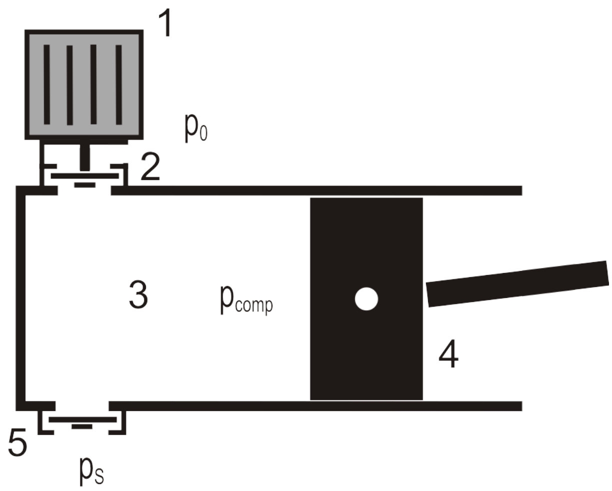

Fig. 2. Actuator on the compressor intake valve, actuator 1, surge valve 2, compression chamber 3, stroke 4 and pressure valve 5. intake pressure p_0, compression pressure p_{comp} and system pressure p_s

As it can be seen in Fig. 2 it is only necessary to equip the intake valve with an actuator, the outtake valve is operated passively by the pressure difference over the valve. As long as the pressure p_s is higher than the pressure in the working chamber the valve remains closed. The valve is operated via the pressure difference $p_s - p_{comp}$, if the difference is negative the valve opens, otherwise it will be closed.

The desired movement of the actuator is shown in Fig. 3. The controlled compression cycle consists of three different phases, namely

I Opening of the inlet valve and intake of the working media,
II Control phase by keeping the inlet valve open, no compression of the media occurs,

III Closing of the inlet valve, opening of the outlet valve and compression of the media.

During the first phase the valve will be opened passively by the compressor and the actuator has to reach the open position until a certain time. After this time point the actuator comes into action by keeping the valve open against the force of the compressor (flow force of the gas). The most critical time point is at the closing of the valve. Indeed the closing time significantly affects the amount of air in the working compartment and the final force on the valve plate.

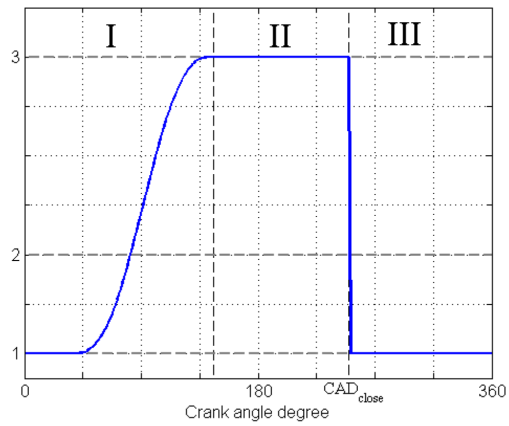

Fig. 3. The different phases during the compression cycle. Actuator end position 1, valve close 2 and valve open 3

There are several critical issues. First, the overall system is strongly nonlinear due to design constraints. Second, the parameters of the actuators but even more of the friction encountered during the operation are unknown and depend on the different operating conditions. During the closing phase an unknown gas flow force acts on the valve plate and this force strongly depends on the closing speed and gap. The intensity of the force itself is comparable with the maximal force of the actuator, and this poses an additional challenge to control the valve.

So the final control challange consists in designing a time optimal control of an unknown, nonlinear system under an unknown disturbance acting only on certain time points during the movement. However, this apparently unsolvable problem is strongly simplified by the fact that the operation of the compressor remains unchanged typically for a longer time, so that learning techniques can be used. A possible solution for this problem is to use an offline optimization based on a simplified mechanical model and to adapt the obtained result to the real system using an Iterative Learning Control (ILC) as presented in Trogmann et al. [2011].

ILC was initially designed to improve the control quality of robotic manipulators, see Arimoto et al. [1984]. However in the last decades scientific groups all over the world have adopted the method and used it for all applications and system classes that fulfil basic requirements, see Chen and Wen [1999]. It has been used for nonlinear non-affine systems, CHI and HOU [2007], constrained linear systems, Chu and Owens [2010], in combination with optimization, Gunnarsson and Norrlf [2001], for highly precise positioning, Barton and Alleyne [2008], nonholomonic mobile robots, Oriolo et al. [1998], calmless valve actuator, Hoffmann and Stefanopoulou [2001], and many more.

ILC, however, is not a time optimal approach. It allows to achieve (perfect) tracking under specific conditions, which are fulfilled in the presented case, the drawback is the necessity of a reference trajectory. Therefore, to achieve time optimality, the "right" reference trajectory must be known. As the design of a time optimal trajectory requires a model, in the model free case such a trajectory could be found via an iterative approach, as already suggested in Trogmann and del Re [2012].

This paper extends the results of the paper Trogmann and del Re [2012] by analyzing the convergence of the trajectory update algorithm, and giving convergence conditions under the assumption that the underlying system is nonlinear and input affine.

In Section 2 the system equations are presented as basis for the simulation model in Section 5. A description of the used method to adapt the desired trajectory is presented in Section 3, followed by convergency conditions for the ILC in Section 4. Conclusion and an outlook can be found in Section 6.

2. APPLICATION

For the used active valve, the already existing passive valve has been used and equipped with a linear actuator. Gearless translational drives have adisadvantage regarding the dynamics of the force density in comparison to rotational actuators with a translative gear. Alternatively to the linear actuator, one can resort to rotational actuators, like permament magnet synchronous machines (PMSM). To translate the rotational movement into a translational one a spindle is used, with self locking capabilities, this means that no translational movement is possible without an input from the rotational part of the gearbox.

The mathematical description of the actuator can be split up into two parts (electrical and mechanical). To avoid the use of angle dependent terms due to the rotation, such motors are normally presented in d/q-coordinates. The electric part is described by

$$\begin{aligned}\frac{d}{dt}i_{sd}(t) &= \frac{1}{L_{sd}}\left[U_{sd}(t) - R_s i_{sd}(t) + \omega_{el}(t) L_{sq} i_{sq}(t)\right]\\ \frac{d}{dt}i_{sq}(t) &= \frac{1}{L_{sq}}\left[U_{sq}(t) - R_s i_{sq}(t) - \omega_{el}(t) L_{sd} i_{sd}(t)\right.\\ &\quad \left.-\omega_{el}(t)\Psi_m\right]\\ M(t) &= \frac{3}{2}p_z\left[\underbrace{\Psi_m i_{sq}(t)(t)}_{synchronous} + \underbrace{(L_{sd}-L_{sq})\, i_{sq}(t)\, i_{sd}(t)}_{reluctance}\right]\end{aligned} \tag{1}$$

with L_{sq}, L_{sd} are the inductances, R_s the resistor, Ψ_m the flux and p_z the number of pole-pairs of the motor. The internal states of the motor are the currents i_{sq}, i_{sd} and the electrical rotational speed $\omega_{el} = \omega \cdot p_z$. The inputs of the system are the two voltages U_{sq} and U_{sd} and the output the mechanical torque M. To complete the model of the actuator, the mechanical equations of the valve plate are presented, i.e.

$$\begin{aligned}\frac{d}{dt}\phi(t) &= \omega(t)\\ \frac{d}{dt}\omega(t) &= \frac{1}{m_{tot}}\left[M(t) - M_f\,\omega(t) - M_{proc}(t)\right]\end{aligned} \tag{2}$$

with ϕ is the angle, ω the rotational speed of the motor, m_{tot} the total mass of the system, M_{proc} the resulting process torque from the valve and M_f the friction force. The last two terms are unknown and an accurate prediction is not possible in real time during normal operation.

For the iterative learning control we need a discrete model, that can be obtained for example using the Forward Euler-discretization method. Letting T_{cycle} be the iteration interval, Δ the sampling interval, and $t_k = k\Delta$ the sampling instants, $0 < k < n$ with $n\Delta = T_{cycle}$, we obtain

$$\begin{aligned} i_{sd}(t_{k+1}) &= i_{sd}(t_k) + \frac{\Delta t}{L_{sd}}\left[U_{sd}(t_k) - R_s i_{sd}(t_k) + \right. \\ &\quad \left. \omega_{el}(t_k) L_{sq} i_{sq}(t_k)\right] \\ i_{sq}(t_{k+1}) &= i_{sq}(t_k) + \frac{\Delta t}{L_{sq}}\left[U_{sq}(t_k) - R_s i_{sq}(t_k) - \right. \\ &\quad \left. \omega_{el}(t_k) L_{sd} i_{sd}(t_k) - \omega_{el}(t_k)\Psi_m\right] \\ M(t_k) &= \frac{3}{2} p_z \left[\underbrace{\Psi_m i_{sq}(t_k)}_{synchronous} + \underbrace{(L_{sd} - L_{sq}) i_{sq}(t_k) i_{sd}(t_k)}_{reluctance}\right] \end{aligned} \tag{3}$$

with the same variables as in (1). The discrete mechanic equations can be written as follows:

$$\begin{aligned} \phi(t_{k+1}) &= \phi(t_k) + \Delta t \cdot \omega \\ \omega(t_{k+1}) &= \omega(t_k) + \frac{\Delta t}{m_{tot}}\left[M(t_k) - M_f(\omega(t_k)) - M_{proc}(t_k)\right] \end{aligned} \tag{4}$$

with the same variables as before. The presented equation use the rotational values, the translational values can be obtained by multplying the values with the ratio of the spindle.

The compressor force is subject to dramatic changes, as can be seen in Fig. 4. The intensity of the force is comparable with the force acting on the actuator, and this fact poses a severe challenge mainly during the breaking phase. An additional problem is that the same force strongly depends on the closing time of the valve, which in turn may change from one iteration to another. As a consequence, the force itself cannot be considered as a periodic disturbance.

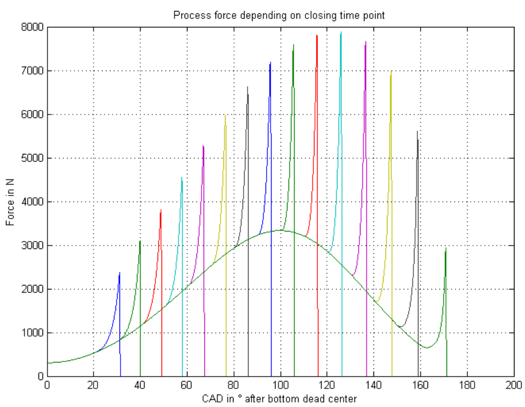

Fig. 4. Dependency of the compressor gas flow force on the closing time point

3. SKETCH OF THE METHOD

As mentioned in Section 2 there are unknown effects, that have to be taken into account to achieve the time optimal movement of the valve. This means that a control method for the proposed application must have the ability to (i) compensate these effects during operation (can be done by the ILC), (ii) create a feasible time optimal trajectory for the ILC.

The importance of the feasibility of the used trajectory will be explained in this section by taking a closer look to the error propagation over the iterations. For the first time the errror propagation for linear systems has be presented in Longman [2000]. We will resort to a slight modification of the algorithm in Trogmann and del Re [2012], that allows to learn a trajectory to suppress the effects of the error propagation of the tracking error and input saturation during and over the iterations. We are interested in the tracking of the motor angle $\phi(t_k)$. With a slight abuse of notation we define the output error propagation during an iteration (the symbol j denotes the j-iteration) as

$$\delta e_j(k) = e_j(k) - e_{j-1}(k) \tag{5}$$

where $e_j(k) = \phi_j(k) - \phi_d(k)$, $\phi_j(k)$ the value at time k of the angle at the $j-th$ iteration, and $\phi_d(k)$ the desired mechanical angle.

3.1 Discrete-time linear systems

The explanation of the error propagation is clear for strictly proper linear systems in discrete-time in the form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Gw(k) \\ y(k) &= Cx(k) \end{aligned} \tag{6}$$

with A,B,C,G the matrices of the system and $w(k)$ a periodic disturbance. The linear property allows the direct calculation of the output or each desired time point k during an iteration, i.e.

$$y(k) = CA^k x(0) + \sum_{i=0}^{k-1} CA^{k-i-1} Bu(i) + \sum_{i=0}^{k-1} CA^{k-i-1} Gw(i) \tag{7}$$

For simpler treatment of the calculation a so called *lifted* vector is created. The lifted vector is obtained by staking in a vector the values of the considered variable (input, ouput, disturbnace) at each time step in one iteration, i.e.

$$\begin{aligned} y_j &= [y_j(1)\ y_j(2)\ \cdots\ y_j(p)] \\ u_j &= [u_j(0)\ u_j(1)\ \cdots\ u_j(p-1)] \\ w_j &= [w_j(0)\ w_j(1)\ \cdots\ w_j(p-1)] \end{aligned} \tag{8}$$

where p is the maximal index for an iteration (and period of $w(k)$). The lifted output can be given a simple formula as follows

$$y_j = Ox(0) + Pu_j + Hw_j \tag{9}$$

where O, P, H are suitable matrices. Notice however that P is a lower triangular Toeplitz matrix containing the Markov coefficient of the system (A,B,C), i.e.

$$P = \begin{bmatrix} CB & & & \\ CAB & CB & & 0 \\ CA^2B & CAB & \ddots & \\ \vdots & \vdots & \ddots & \ddots \\ CA^{p-1}B & CA^{p-2}B & \cdots & CAB\ CB \end{bmatrix}. \tag{10}$$

Under the assumption that the process is repetitive, i.e. $x_{j+1}(0) = x_j(0)$, $w_{j+1} = w_j$, it turns out that disturbance $w(k)$ are equal in each iteration one gets

$$\delta y_j = P\delta u_j \tag{11}$$

and hence, denoting by $e(k)$ the tracking error and e_j its lifted version one has

$$\delta e_j = e_j - e_{j-1} = -\delta y_j \tag{12}$$

so that, using the update law

$$u_j = u_{j-1} + Le_{j-1} \tag{13}$$

the error evolution can be expressed as

$$e_j = (I - PL)\, e_{j-1} = (I - PL)^j\, e_0. \tag{14}$$

In the case of the so called P-type updating law, matrix L is a diagonal matrix with the proportional gain (Φ) as diagonal elements, and hence the error can be computed as

$$e_j(1) = (1 - CB\Phi)\, e_{j-1}(1)$$
$$e_j(2) = (1 - CB\Phi)\, e_{j-1}(2) + CAB\Phi e_{j-1}(1)$$

as so on so forth.

The most important awareness of the error propagation is that one has to chose Φ such that $|1 + CB\Phi| < 1$ as a necessary and sufficient condition for the error going iteratively to zero. Of course this is impossible if $CB = 0$. Notice that normally the condition $CB \neq 0$ is verified since the discrete-time systems is obtained after a discretization of a continuous-time model. The sign of CB is also usually known in real applications. However, in case it were unknown one can easily modify the updating scheme by taking a 2-periodic integrator, i.e. letting L be a 2-periodic diagonal matrix, function of j, with $L_j = \text{diag}\{\Phi, -\Phi, \Phi, \cdots\}$ for j odd and diagonal with alternate elements Φ and $L_j = \text{diag}\{-\Phi, \Phi, -\Phi, \cdots\}$ for j even. Stability is ensured if and only if $|1 - (CB)^2\Phi^2| < 1$, that is always true for $CB \neq 0$ and $|\Phi|$ small enough. For major details on internal model of periodic integrators see Colaneri [1990].

On the other hand, in the case of the so called the PD-type updating law, the L-matrix has a similar form with a secondary diagonal containing the D-component of the update law.

$$L = \begin{bmatrix} \Phi - \frac{\Theta}{T_A} & \frac{\Theta}{T_A} & & 0 \\ & \Phi - \frac{\Theta}{T_A} & \ddots & \\ & & \ddots & \frac{\Theta}{T_A} \\ 0 & & & \Phi - \frac{\Theta}{T_A} \end{bmatrix} \tag{15}$$

It can be seen that the error at time step 1 has an influence on all consecutive errors

$$e_j(1) = \left(1 - CB\hat{\Phi}\right) e_{j-1}(1) + CB\frac{\Theta}{T_A} e_{j-1}(2)$$
$$e_j(2) = \left(1 - CB\hat{\Phi}\right) e_{j-1}(2) + CB\frac{\Theta}{T_A} e_{j-1}(3) + CAB\Phi e_{j-1}$$

with $\hat{\Phi} = \Phi - \frac{\Theta}{T_A}$, and so on. It is apparent that if $CB\left(\Phi - \frac{\Theta}{T_A}\right) < 0$, then the initial error increases whereas it decreases if $CB\left(\Phi - \frac{\Theta}{T_A}\right) > 0$. Notice that matrix $I - PL$ is no longer triangular. However, as shown in Trogmann and del Re [2012], this second method has a better tracking result. The basis of the algorithm is to reduce the initial error at each iteration as well as the errors during the iteration caused by unfeasible trajectory points. In the nonlinear setting, the error propagation cannot be explicitly written. In the next section we mark out sufficient convergence conditions for nonlinear systems in continuous-time.

4. CONVERGENCE CONDITION

The electromechanic system described in Section 2 belongs to the class of nonlinear (actually bilinear) input-affine continuous-time systems. As already said, the iterative learning method is naturally cast in a discrete-time setting, and as such we have considered the Forward-Euler discretization method, leading to a discrete-time input affine nonlinear system of the form

$$\begin{aligned} x(k+1) &= f(x(k)) + g(x(k))\, u(k) \\ y(k) &= h(x(k)) \end{aligned} \tag{16}$$

In the nonlinear setting, the easy procedure discribed in Section 3for linear systems has to be adapted trying to find bounds on the norms of the various signals acting on the loop. Hence, the ILC convergence condition will be derived by showing that

$$\|e_{j+1}\| \leq q_j \|e_j\|, \quad \lim_{j\to\infty} \prod_{i=1}^{j} q_j = 0 \tag{17}$$

where again e_j denotes the lifted vector at iteration j, associated with the tracking error $e(k) = y_d(k) - y(k)$, for k ranging form 0 to the the final horizon time, say N. For the discussion, we notice that

$$f(x_{j+1}(k)) \simeq f(x_j(k)) + \left.\frac{\partial f}{\partial x}\right|_{x_j(k)} (x_{j+1}(k) - x_j(k))$$

and analogously for vectors $h(x)$, $g(x)$ and $l(x)$. Hence, a simple computation shows that

$$\begin{aligned} \delta x_j(k+1) &= A\delta x(k) + B\delta u(k) + G\delta w(k) \\ \delta y_j(k) &= C\delta x_j(k) \end{aligned} \tag{18}$$

where, however, matrices A, B, C, G are trajectory dependent, i.e.

$$\begin{aligned} A &= \left.\frac{\partial f}{\partial x}\right|_{x_j(k)} + \left.\frac{\partial g}{\partial x}\right|_{x_j(k)} u_{j+1}(k) + \left.\frac{\partial l}{\partial x}\right|_{x_j(k)} w_{j+1}(k) \\ B &= g(x_j(k)) \\ C &= \left.\frac{\partial h}{\partial x}\right|_{x_j(k)} \\ G &= l(x_j(k)) \end{aligned} \tag{19}$$

A formula, formally identical to (11) can be obtained if $w(k)$ is periodic, i.e. $\delta e_j = -P\delta u_j$, so that the use of the updating rule $\delta u_j = Le_{j-1}$ leads to

$$e_{j+1} = (I - PL)e_j \tag{20}$$

Matrix $I - PL$ depends on $x_j(k)$, $u_{j+1}(k)$ and $w_{j+1}(k)$, for all k from 0 to N. Notice however that all entries of this matrix are bounded thanks to the common Lipshitz assumption of the functions describing the system.

In the case of the so-called P-Type updating, matrix L is a the identity multiplied by a proportional gain Φ, and $I - PL$ is triangular. The entries on the diagonal can be written as

$$1 - CB\Phi = 1 - \left.\frac{\partial h}{\partial x}\right|_{x_j(k)} g(x_j(k))\Phi$$

so that the design parameter Φ should be chosen to minimize $|1 - \left.\frac{\partial h}{\partial x}\right|_{x_j(k)} g(x_j(k))\Phi|$ over $x_j(k)$. Notice that we can allow Φ to depend on k so that it is possible to select $\Phi(0)$, $\Phi(1)$, $\cdots$ in order to have $|1 - \left.\frac{\partial h}{\partial x}\right|_{x_j(k)} g(x_j(k))\Phi(k)| < \alpha_j < 1$, for each k. This means that the time-varying triangular system (20) is asymptotically stable, so entailing the existence of parameters q_j satisfying (17).

By using instead the so-called PD-type updating law, namely

$$u_{j+1}(k) = u_j(k) + \Phi e_j(k) + \Theta \frac{e_j(k) - e_j(k+1)}{T_A} \quad (21)$$

we end up with a matrix L as in (15). The closed-loop matrix of the system $I - PL$ still depends on $x_j(k)$, $u_{j+1}(k)$ and $w_{j+1}(k)$, for all k from 0 to N, but now we have lost the triangular structure of the matrix (there is just one additional nonzero entries in the elements $(i, i+1)$ whereas the elements $(i, i+k)$, $k \geq 2$ are still zero). However, this structure gives more degrees of freedom and it is possible to work out condition under which there exists parameters $\Theta(k)$ and $\Phi(k)$ such that for each iteration we can obtain a diagonally dominant and contractive matrix $I - PL$, see Trogmann et al. [2011].

A crucial issue comes from the role of the saturated input points. In the case in which the input saturates for a single time step or points, the actual and the next input have zero difference without a zero error. It is then necessary to see what happens with this input in these points. Hence the update rule (21) can be changed as follows

$$u_{j+1}(k) = \text{sat}\left(u_j(k) + \Phi e_j(k) + \Theta \frac{e_j(k) - e_j(k+1)}{T_A}, u_{max}\right) \quad (22)$$

where

$$\text{sat}(u, u_{max}) = \begin{cases} -u_{max} & u < -u_{max} \\ u & -u_{max} \leq u \leq u_{max} \\ u_{max} & u > u_{max} \end{cases}$$

There are in total 4 cases, besides the non saturated case:

$$u_j(k) = u_{max} \bigwedge \Phi e_j(k) + \Theta \frac{e_j(k) - e_j(k+1)}{T_A} > 0 \rightarrow u_{j+1}(k) = u_{max}$$

$$u_j(k) = u_{max} \bigwedge \Phi e_j(k) + \Theta \frac{e_j(k) - e_j(k+1)}{T_A} < 0 \rightarrow u_{j+1}(k) < u_{max}$$

$$u_j(k) = -u_{max} \bigwedge \Phi e_j(k) + \Theta \frac{e_j(k) - e_j(k+1)}{T_A} > 0 \rightarrow u_{j+1}(k) > -u_{max}$$

$$u_j(k) = u_{max} \bigwedge \Phi e_j(k) + \Theta \frac{e_j(k) - e_j(k+1)}{T_A} < 0 \rightarrow u_{j+1}(k) = -u_{max}$$

The consequence of the saturation is that the convergence of the input to the optimal desired input will be delayed. A change of the error can still occur since not all points in the iteration are saturated. In the case of complete saturation, the ILC can converge to a bang bang sequence, whose optimality should be proven through the maximum Pontragyn principle, see Stengel [1994].

5. EXAMPLE - SIMULATION

For the simulation the different parts of the system has been implemented in Matlab/Simulink. As controller an online ILC in the form (21) is used. To obtain the values for $\Phi(k)$ and $\frac{\Theta(k)}{T_A}$ the approximation matrices of the system have been calculated corresponding to (19). For the presented system the matrices B and C are constant and independent from the iteration number. Additional to this, the terms in P, see (10), containing the powers of A are small compared to the other terms (factor 10^{-6}), so that a triangular structure of $I - PL$ is obtained and to satisfy the condition that $\|I - PL\| < 1$ $\Phi(k)$ is set to 5 and $\frac{\Theta(k)}{T_A}$ is set 0.5. To fulfill the requirement that the initial value is the same for each iteration, the simulation is started for each iteration again. There is no post- or preoperation done during two iterations.

The compressor gas flow force is calculated dependendly on the actual status of the system as there are different influences as mentioned before in the paper. In addition this flow force violates a basic requirement of the ILC that the disturbance is equal in each iteration this can only be achieved if the ILC converges to the final trajectory. The effect of the changing force can be seen in Fig. 5 and Fig. 6 as changes during the holding phase around $0.02s$ which induces the oscillations in this period.

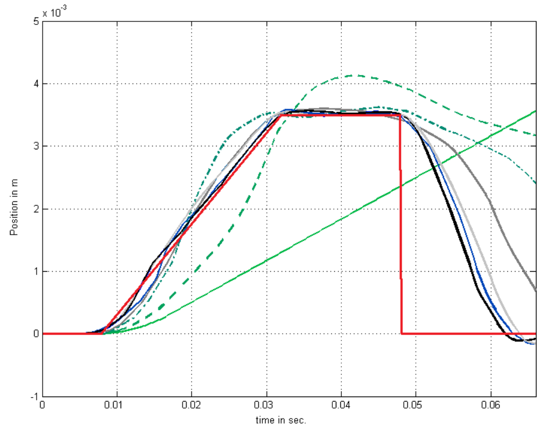

Fig. 5. Output of the system without trajectory adaptation from the first iteration(green) to $j = 2000$ (black). The given reference is drawn in red.

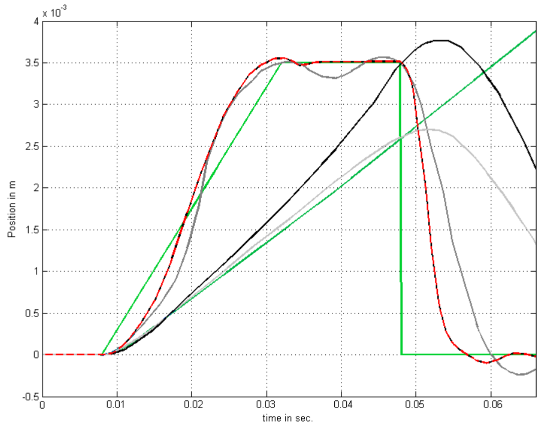

Fig. 6. Output of the system with trajectory adaptation from the first iteration(green) to $j = 2000$ (black dashed). The initial reference is green and the final reference is a red dashed line.

In Fig. 5, 6, the trajectories of the position are plotted. A period of $60ms$ has been selected in order to see what happens at the end of the cycle. The cycle frequenzy is 20Hz which results in a period length of $50ms$. An important measure for the quality of the control is how fast the motor can close the valve. The closing time for this kind of valve is specified for a movement of 2.5mm.

Closing times		
Case	[ms]	%
ILC - without Update	9	145
ILC - with Update	6.8	110
Calculated	6.2	100

So the proposed method is nearly as good as the optimal calculated closing time (with the highest assumed gas force helping and without friction), butdoes not need any model. The more important result is the improvement compared with the standard method. In the presented case the trajectory adaptation allows to reduce the closing time by more than 2ms compared to the standard case (or 35%).

The difference can be appreciated in the error plot as shown in Fig. 7,8

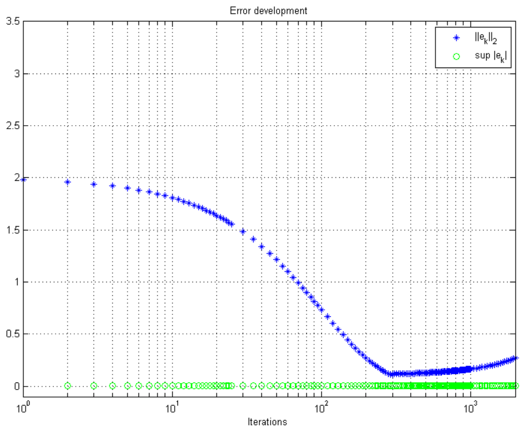

Fig. 7. After a while the error starts sligthly to increase

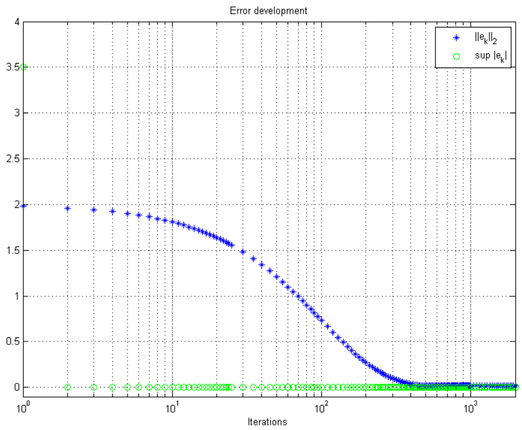

Fig. 8. Longer decrease of the error and lower final values of the error

6. CONCLUSION AND OUTLOOK

In this paper a trajectory update algorithm has been introduced and used in an application. The advantage of the method is that no knowledge about the system is necessary and the adaptation algorithm is a simple method to improve the tracking quality of an ILC. Future work will consider the improvement of the update algorithm by using functions instead of a pointwise adaption of the trajectory. A remaining topic will be the stability of the ILC and the influence of oscillations on the update algorithm during the learning of the trajectory.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the sponsoring of this work by the JKU HOERBIGER Research Institute for Smart Actuators (JHI).

REFERENCES

Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984.

K.L. Barton and A.G. Alleyne. A cross-coupled iterative learning control design for precision motion control. *Control Systems Technology, IEEE Transactions on*, 16(6):1218 – 1231, nov. 2008.

Yangquan Chen and Changyun Wen. *Iterative Learning Control:convergence, robustness and applications*. Lecture notes in control and information sciences. Springer-Verlag, 1999.

Rong-Hu CHI and Zhong-Sheng HOU. Dual-stage optimal iterative learning control for nonlinear non-affine discrete-time systems. *Acta Automatica Sinica*, 33(10):1061 – 1065, 2007.

Bing Chu and David H. Owens. Iterative learning control for constrained linear systems. *International Journal of Control*, 83(7):1397–1413, 2010.

Patrizio Colaneri. Zero-error regulation of discrete-time linear periodic systems. *Systems and Control Letters*, 15(2):377–380, 1990.

Svante Gunnarsson and Mikael Norrlf. On the design of ilc algorithms using optimization. *Automatica*, 37(12):2011 – 2016, 2001.

Woflgang Hoffmann and Anna G. Stefanopoulou. Iterative learning control of electromechanical camless valve actuator. In *Proceddings of the American Control Conference*, 2001.

Richard W. Longman. Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, 73(10):930–954, 2000.

Giuseppe Oriolo, Stefano Panzieri, and Giovanni Ulivi. An iterative learning controller for nonholonomic mobile robots. *The International Journal of Robotics Research*, 17(9):954–970, 1998.

Robert F. Stengel. *Optimal control and estimation*. Constable and Company, 1994.

H. Trogmann and L. del Re. On the time optimal control for nonlinear saturated systems. In *Conference on Decision and Control, 2012 Maui, Hawaii, USA*. IEEE, 2012.

H. Trogmann, H. Waschl, D. Alberer, and B. Spiegl. Time optimal compressor valve soft landing by two step ilc. In *Proceedings of the ASME 2011 Dynamic Systems and Control Conference DSCC 2011*. American Society of Mechanical Engineers, October 2011. Arlington, VA, USA.